



SPACEDOT

AcubeSAT

AOCS Design Definition & Justification File

[DDJF_AOCS]

ASAT_DDJF_AOCS_2021-05-17_v2.0
ESA document reference

AcubeSAT-ADC-TU-006
AcubeSAT document reference

Version: 2.0
2021-05-17



FLY YOUR SATELLITE!



Disclaimer

The AcubeSAT project is carried out with the support of the Education Office of the European Space Agency, under the educational [Fly Your Satellite!](#) programme. This document has been authored by university students participating in the AcubeSAT project. The views expressed herein by the authors can in no way be taken to reflect the official opinion, or endorsement, of the European Space Agency.



Authors

Document Role	Project Role	Full Name
Author	AOCS Coordinator	Georgios Savvidis
Author	AOCS Coordinator	Romanos Voulgarakis
Author	AOCS Member	Theodoros Papafotiou
Author	AOCS Member	Vasileios Moustakas
Author	AOCS Member	Emmanouil Mylonas
Author	AOCS Member	Vasiliki Pappa
Reviewer	AOCS Member	Ioannis Kotsakiachidis
Copyeditor	Book Captain	Konstantinos Kanavouras
Copyeditor	System Engineer	Anastasios - Faidon Retselis
Copyeditor	OBDH Member	Athanasios Theocharis

Document Information

Title	AOCS Design Definition & Justification File [DDJF_AOCS]		
Project	AcubeSAT, SpaceDot		
Institution	Aristotle University of Thessaloniki		
Revision Date	2021-05-17	Render Date	2021-11-05 18:45
Status	PUBLISHED	Variant	Web, Public
Documentation Template version	vt1.8	Data Package Template version	vd1.2-dev

Changelog

Date	Version	Details
2021-05-17	2.0 PUBLISHED	<ul style="list-style-type: none">Added residual magnetic dipole compensation (Paragraph 6.2.3.4 - Paragraph 6.2.6.3).Conducted gain tuning (Section 6.3.1).Updated simulations (Section 6.1).Updated budgets (Section 4).



Date	Version	Details
2021-02-21	1.2 PUBLISHED	<ul style="list-style-type: none">Removed Fine Sun sensor throughout the document.Added error metric definition (Section 4.1).Added MEKF (Section 5.4.4).Updated inertia matrix (Section 4.2.1).Added Approximation of Angular Velocity using B-dot metrics (Section 3.3).Updated disturbances (Section 4.2.3).Updated sensor model (Section 6.2.5).Added albedo effect (Paragraph 6.2.3.5 - Paragraph 5.4.4.2).Updated gain tuning (Section 6.3.1).Updated nominal mode analysis with magnetic actuation (Paragraph 6.4.2.10).Updated coarse sun sensor (Paragraph 2.2.1.3).Added sun pointing profile (Section 3.2 - Section 5.6 - Paragraph 6.4.1.3).Added complete simulations (Section 6.4.3).Added redundancies clarifications (Paragraph 2.1.1.1).Updated safe mode dynamics (Paragraph 6.2.4.4).Updated budgets (Section 4).Updated simulation/model sampling times and frequencies (Section 6.2.8).Updated environmental models (Section 6.2.3).Updated nominal scenarios (Section 6.4.1).Updated Worst-case scenarios (Section 6.4.2).
2020-11-12	1.1 PUBLISHED	<ul style="list-style-type: none">Renamed <i>safe</i> to <i>stand-by</i> mode.Added stand-by mode dynamics analysis (Section 4.6).Updated inertia matrix.Updated momentum budget (Section 4.3.2).Added APE, AKE confidence intervals (Section 4.4).Added information about detumbling mode angular velocity thresholds.Added nominal mode analysis with magnetic actuation (Paragraph 6.4.2.10).Added information about in-orbit patching.
2020-10-04	1.0 RELEASED	<ul style="list-style-type: none">Added information regarding technical simulations (??).Added sun sensor details (??).Removed unrequired plots (Section 5.1).Simulations added (Section 5.7).Simulations with the new Inertia matrix added (Section 6.4).



Date	Version	Details
2020-09-27	0.1 RELEASED	Initial release

This document is licensed under a [Creative Commons Attribution 4.0 International License](#). You can find a copy of the license at:
<https://creativecommons.org/licenses/by/4.0/legalcode>



Table Of Contents

Preamble	1
Authors	2
Document Information	2
Changelog	2
List of Abbreviations	8
Introduction	9
Reference Documents	10
1 Quick Facts Table	11
2 Physical Architecture	11
2.1 Hardware	11
2.1.1 Sensors	11
2.1.1.1 Magnetometers	11
2.1.1.2 Gyroscope	12
2.1.1.3 Sun sensors	12
2.1.1.4 Temperature sensor	13
2.1.2 Actuators	13
2.1.2.1 Magnetorquers	13
2.1.2.2 Reaction Wheel	13
2.1.3 AOCS PCB	14
2.1.3.1 CAN transceiver	14
2.1.3.2 MCU	14
2.2 Positions and Orientation	16
2.2.1 Sensors	18
2.2.1.1 Magnetometers	18
2.2.1.2 Gyroscope	18
2.2.1.3 Coarse Sun Sensor	18
2.2.1.4 Temperature Sensors	19
2.2.2 Actuators	19
2.2.2.1 Magnetorquers	19
2.2.2.2 Reaction Wheel	19
2.3 Key interfaces	20
2.3.1 Physical Interfaces	20
2.3.2 Data Interfaces	20
2.4 Justification for the chosen design	20
3 AOCS mode definitions	21
3.1 Mode descriptions	22
3.2 Attitude Profiles	24
3.3 Triggers	25
3.4 AOCS mode sequence	29
4 AOCS Budgets	30
4.1 Error metric definition	30
4.2 Brief spacecraft geometry and orbit characteristics	30
4.2.1 Geometrical characteristics	30
4.2.2 Orbital characteristics	31
4.2.3 Disturbances	31



4.3	Actuator sizing budgets	31
4.3.1	Torque budget	31
4.3.1.1	Detumbling mode	31
4.3.1.2	Nominal mode	34
4.3.2	Momentum budget	37
4.4	Pointing budget	41
4.5	Geolocation budgets	45
4.6	Stand-by mode dynamics analysis	49
4.7	Delta-v budget	49
5	AOCS Control and determination algorithms	50
5.1	Orbit Propagator	50
5.2	On-board ephemeris	56
5.2.1	Sun position	56
5.2.2	Eclipse	59
5.3	Magnetic field model	60
5.4	Sensor processing algorithms	62
5.4.1	Coarse sun sensor filtering	62
5.4.2	Wahba's problem	62
5.4.3	Initial bias estimation	63
5.4.4	Multiplicative Extended Kalman Filter	64
5.4.4.1	Bias model in MEKF	65
5.4.4.2	Albedo compensation	66
5.4.5	Sampling frequency	66
5.4.6	Inputs	66
5.4.7	Outputs	66
5.5	Estimation, guidance and control algorithms	66
5.6	Actuator mapping and command processing algorithms	68
5.7	Control action discretisation	72
6	Simulations	75
6.1	Introduction	75
6.2	Simulation model	76
6.2.1	Reference Frames	76
6.2.1.1	Earth Centered Inertial	76
6.2.1.2	Earth-Centered Earth-Fixed	77
6.2.1.3	Orbit Frame	77
6.2.1.4	NED Frame	77
6.2.1.5	Body Frame	78
6.2.2	Transformation matrices	79
6.2.2.1	ECI to ECEF	79
6.2.2.2	ECI to Orbit	80
6.2.2.3	NED to ECEF	81
6.2.2.4	Orbit to Body	81
6.2.3	Environmental models and assumptions	81
6.2.3.1	Gravitational torque	81
6.2.3.2	Aerodynamic drag	82
6.2.3.3	Solar Radiation Pressure	82
6.2.3.4	Residual Magnetic Torque	83
6.2.3.5	Albedo effect	84

6.2.4	Orbit and attitude kinematic/dynamic propagators	84
6.2.4.1	Orbit propagator	84
6.2.4.2	Kinematic propagator	85
6.2.4.3	Dynamic propagator	85
6.2.4.4	Safe Mode	86
6.2.5	Sensor model	86
6.2.5.1	Magnetometer model	86
6.2.5.2	Gyroscope model	86
6.2.5.3	Coarse sun sensor	87
6.2.6	Actuator model	88
6.2.6.1	Magnetorquers	88
6.2.6.2	Saturation of the Magnetorquers	88
6.2.6.3	Residual Magnetic Dipole Compensation	89
6.2.6.4	Magnetorquers Power	89
6.2.6.5	Reaction Wheel	90
6.2.6.6	Saturation of the Reaction Wheel	91
6.2.6.7	Zero-crossings of the Reaction Wheel	92
6.2.7	SW/HW delays	93
6.2.7.1	SW delays	93
6.2.7.2	HW delays	94
6.2.8	Simulation/model sampling times and frequencies	95
6.2.8.1	Detumbling mode	95
6.2.8.2	Nominal mode	95
6.3	Analyses	96
6.3.1	Gain Tuning	97
6.4	Simulation Performance campaign plan & results	98
6.4.1	Nominal scenarios	98
6.4.1.1	Detumbling	98
6.4.1.2	Nominal	98
6.4.1.3	Sun Pointing	101
6.4.2	Worst-case scenarios	103
6.4.2.1	Detumbling high initial angular velocities	103
6.4.2.2	Unrealistic eclipse duration	104
6.4.2.3	Random estimation spikes in MEKF	107
6.4.2.4	Random measurement spikes from sensors	109
6.4.2.5	Desired quaternion change	109
6.4.2.6	Variation of Deadzone Limits	111
6.4.2.7	Variation of Reaction Wheel Frictions	114
6.4.2.8	Failure of Reaction Wheel during a Time Interval	117
6.4.2.9	Failure of Magnetorquers during a Time Interval	120
6.4.2.10	Nominal mode using only magnetic actuation (N-02)	123
6.4.3	Complete Simulations	123
7	Variables Table	128

List of Abbreviations

ADCS	Attitude Determination & Control Subsystem	LEO	Low Earth Orbit
AKE	Absolute Knowledge Error	MCU	Microcontroller Unit
AOCS	Attitude & Orbit Control System	MEKF	Multiplicative Extended Kalman Filter
APE	Absolute Performance Error	MEMS	Microelectromechanical Systems
CAN	Controller Area Network	MTQ	Magnetorquers
CAN-FD	Controller Area Network Flexible Data-Rate	NED	local North East Down
COMMS	Communications Subsystem	OBC	On-Board Computer
CSS	Coarse Sun Sensor	OBDH	On-Board Data Handling
DDJF	Design Definition Justification File	PCB	Printed Circuit Board
ECC	Error Correcting Code	PD	Proportional Derivative
ECEF	Earth-centered Earth-fixed	RPM	Rounds Per Minute
ECI	Earth-centered Inertial	RT	Radiation-Tolerant
ESA	European Space Agency	RTC	Real-Time Clock
ESD	Electro-Static Discharge	RW	Reaction Wheel
FDIR	Fault Detection Isolation Report	SGP4	Simplified General Perturbations Model 4
FYS	Fly Your Satellite	SPI	Serial Peripheral Interface
GS	Ground Station	SSO	Sun Synchronous Orbit
HW	Hardware	SVD	Singular Value Decomposition
I2C	Inter-Integrated Circuit	SW	Software
IC	Integrated Circuit	TC	Telecommand
IDE	Integrated Development Environment	TLE	Two-Line Element
IGRF	International Geomagnetic Reference Field	TMTC	Telemetry & Telecommand
IO	Input/Output	TT&C	Telemetry, Tracking & Command
JD	Julian Date	UART	Universal Asynchronous Receiver/Transmitter



Introduction

The Design Definition Justification File (DDJF) of AOCS is the document aiming at describing in detail all the aspects regarding attitude control of the satellite for the **Acube-SAT** mission. The mission includes a scientific payload, which is analyzed through photographs, requires a directional patch antenna as explained in [*ASAT_DDjf_PL*](#) and [*ASAT_DDjf_TTC*](#). Consequently, the AOCS is required to achieve nadir pointing during satellite nominal operations in a safe and orderly fashion. This is achieved by using active attitude determination and control techniques.

This document provides information on the satellite determination and control approach with regards to both hardware and software according to ESA guidelines. It also contains the analysis approaches, along with their justification and description, as well as the required documentation, all in compliance with the ESA Policy.

This document will remain open until the later stages of the project and therefore will be updated on a regular basis during the different FYS phases.

Reference Documents

- [1] AcubeSAT Team. *Payload Design Definition & Justification File*. ASAT_DDGF_PL. Version 2.0. May 17, 2021.
- [2] AcubeSAT Team. *TTC Design Definition & Justification File*. ASAT_DDGF_TTC. Version 2.0. May 17, 2021.
- [3] AcubeSAT Team. *OBDH Design Definition & Justification File*. ASAT_DDGF_OBDH. Version 1.2. Feb. 13, 2021.
- [4] AcubeSAT Team. *OBSW Design Definition & Justification File*. ASAT_DDGF_OBSW. Version 1.2. Jan. 22, 2021.
- [5] European Cooperation for Space Standardization. *Control performance*. ECSS-E-ST-60-10C. Nov. 15, 2008.
- [6] James Wertz (editor) David A. Vallado. *Fundamentals of Astrodynamics and Applications, 4th ed. (Space Technology Library)*. 4th ed. Microcosm Press, 2013. ISBN: 1881883183, 9781881883180.
- [7] Xianghua Jia et al. “Eclipse Prediction Algorithms for Low-Earth-Orbiting Satellites.” In: *IEEE Transactions on Aerospace and Electronic Systems* PP (July 2017), pp. 1–1. doi: [10.1109/TAES.2017.2722518](https://doi.org/10.1109/TAES.2017.2722518).
- [8] Drew Compston. “International Geomagnetic Reference Field (IGRF).” In: (2020). MATLAB Central File Exchange. Retrieved August 23, 2020.
- [9] James Forbes and Christopher Damaren. “Geometric Approach to Spacecraft Attitude Control Using Magnetic and Mechanical Actuation.” In: *Journal of Guidance Control and Dynamics - J GUD CONTROL DYNAM* 33 (Mar. 2010), pp. 590–595. doi: [10.2514/1.46441](https://doi.org/10.2514/1.46441).
- [10] Landis Markley and John Crassidis. *Fundamentals of Spacecraft Attitude Determination and Control*. Jan. 2014. ISBN: 978-1-4939-0802-8. doi: [10.1007/978-1-4939-0802-8](https://doi.org/10.1007/978-1-4939-0802-8).
- [11] James Tsui. “Fundamentals of Global Positioning System Receivers: A Software Approach.” In: Oct. 2001, pp. i–xvii. ISBN: 9780471200543. doi: [10.1002/0471200549.fmatter_insub](https://doi.org/10.1002/0471200549.fmatter_insub).
- [12] James R. Wertz Wiley J. Larson. *Space Mission Analysis and Design*. Space Technology Library. Microcosm, 1999. ISBN: 978-1881883104.
- [13] Samir A. Rawashdeh. “CubeSat Aerodynamic Stability at ISS Altitude and Inclination.” In: 2012.
- [14] [MATLAB Earth Albedo Toolbox](#).
- [15] [Total Ozone Mapping Spectrometer-Earth Probe](#).
- [16] Leonardo Regoli et al. “Investigation of a low-cost magneto-inductive magnetometer for space science applications.” In: *Geoscientific Instrumentation, Methods and Data Systems Discussions* 7 (Mar. 2018), pp. 129–142. doi: [10.5194/gi-7-129-2018](https://doi.org/10.5194/gi-7-129-2018).
- [17] Valdemir Carrara and Helio Kuga. “Current and Speed Control Operating Modes of a Reaction Wheel.” In: *Applied Mechanics and Materials* 706 (Jan. 2015), pp. 170–180. doi: [10.4028/www.scientific.net/AMM.706.170](https://doi.org/10.4028/www.scientific.net/AMM.706.170).



1 Quick Facts Table

2 Physical Architecture

2.1 Hardware

The AOCS Physical Architecture involves six sensors, two different types of actuators and one MCU board as detailed below.

2.1.1 Sensors

2.1.1.1 Magnetometers

Two 3-axis magnetometers are used in our AOCS unit. One of them functions as the main sensor in order to acquire high precision geomagnetic field measurements, while the other is equipped only for redundancy purposes, in case a failure occurs. The **RM3100** provided by *PNI* is selected as the precision magnetometer. It consists of three coil sensors (**Sen XY/Z**), which are driven by the MagI2C controller in a separate IC. The second module is provided by *ISIS* and is embedded in the iMTQ magnetorquers board, with an accuracy of $< 3 \mu\text{T}$.

The **RM3100** is utilized consistently, while the second magnetometer, provided by *ISIS* operates periodically, to verify the precision of the **RM3100**, essentially using them simultaneously using the warm-redundancy concept.

Table II: RM3100 - Key parameters

RM3100 - Key parameters	
Noise	15 nT
Sensitivity	13 nT
Dimensions	Sen XY: 6.0 x 2.1 x 2.2 mm Sen Z: 3.0 x 3.0 x 5.75 mm MagI2C: 4.0 x 4.0 x 0.75 mm
Sampling rate	300 Hz
Power	2.2 mW

where **Sen XY/Z** are the sensor coils on the XY and Z axes respectively, and MagI2C is the controller.

Requirements close-out — Review of Design

The above text is the close out reference for requirement:

ADCS-FUN-210

All magnetometers shall have noise with standard deviation less than 20uT.



2.1.1.2 Gyroscope

The **ADXRS453** three-axis MEMS gyroscope is utilized as the sole gyroscope unit, provided by *Analog-Devices*.

Table III: ADXRS453 - Key Parameters

ADXRS453 - Key Parameters	
Angular Random Walk	0.015 - 0.023 °/sec/ $\sqrt{\text{Hz}}$
Bias Instability	16 °/h
Nominal Sensitivity	80 LSB/°/sec
Dimensions	10.3 x 10.4 x 3.5 mm
Vibration Rejection	0.01 °/sec/g
Sampling rate	485 Hz
Power	42 mW

Requirements close-out — Review of Design

The above text is the close out reference for requirements:

ADCS-FUN-230	All gyroscopes shall have rate noise density less than 1 deg/sec/ $\sqrt{\text{Hz}}$.
ADCS-FUN-240	All gyroscopes shall have bias instability less than 60 deg/hr.

2.1.1.3 Sun sensors

Our design uses Coarse sun sensors, which are integrated on the Solar Panels, provided by *EnduroSat*. However, as stated in [paragraph 2.2.1.3](#), one additional coarse sun sensor, the **TSL2540** is utilized, provided by *ams AG*.

Table IV: TSL2540 - Key Parameters

TSL2540 - Key Parameters	
Field of View (2σ)	$\pm 60^\circ$
Dimensions	2 x 2 x 2 mm
Supply voltage	1.8 V
Power	0.3 mW

Requirements close-out — Review of Design

The above text is the close out reference for requirement:

ADCS-FUN-220	All fine sun sensors shall have accuracy better than 5 deg.
--------------	---



No further information about the coarse sun sensors has been provided by EnduroSat.

2.1.1.4 Temperature sensor

Two **MCP9808** temperature sensors are used, provided by *Microchip Technologies*.

Table V: MCP9808 - Key Parameters

MCP9808 - Key Parameters	
Sensing Temperature	-40°C to +125°C
Resolution	0.25°C (typical)
Precision	0.0625°C (typical)
Dimensions	21 x 13 x 2 mm
Operating Voltage Range	2.7V to 5.5V
Operating Current	200 μA

2.1.2 Actuators

2.1.2.1 Magnetorquers

The **ISIS iMTQ** magnetorquer board has been selected, containing three magnetorquers, one for each satellite body frame axis. The board is equipped with two magnetorquer rods and one air-core magnetorquer. They are connected to the ADCS/OBC board and are controlled through the AOCS MCU.

Table VI: iMTQ - Key Parameters

iMTQ - Key Parameters	
Maximum magnetic dipole	0.2 Am ²
Mass	196 g
Dimensions	95.9 x 90.1 x 17 mm
Power	175 mW (Idle) - 1.2 W (Max)

2.1.2.2 Reaction Wheel

The **RW210** reaction wheel is used, manufactured by *Hyperion Technologies*.

Table VII: RW210 - Key Parameters

RW210 - Key Parameters	
Power	< 800 mW (peak)
Maximum Torque	0.1 mN·m
Dimensions	25 x 25 x 15 mm ³
Momentum Storage	1.5, 3.0 or 6.0 mN·m·s

2.1.3 AOCS PCB

2.1.3.1 CAN transceiver

Table VIII: CAN transceiver Key Parameters

TCAN337G - Key Parameters	
Protocol	≤ CAN-FD
Data Rate	≤ 5 Mbps
Protection Features	<ul style="list-style-type: none"> • ESD protection • Over- and undervoltage protection • Babbling idiot protection via timeouts • Fault output pin

A CAN-FD transceiver is necessary to shift the CAN bus voltages and establish the physical architecture of the bus. The one used for the AOCS and the rest of the subsystems is **Texas Instruments TCAN337G**. It is compliant with the CAN-FD bus requirement of the SatNOGS COMMS board, and offers protection features including:

- Overvoltage, undervoltage and overcurrent protection
- Fault detection output
- Babbling idiot protection (via bus timeout detection)

Interfacing between AOCS and OBDH is achieved by the **on-board CAN bus**, which is used to transfer parameters, events, commands, science and logging data, TT&C packets and synchronisation markers/heartbeats. The CAN bus is described in detail in [ASAT_DDJF_OBDH](#).

2.1.3.2 MCU

The **SAMV71Q21RT** radiation-tolerant MCU, manufactured by *Microchip Technologies*, is placed in the ADCS/OBC board to fulfill the computational needs of the AOCS.

Table IX: SAMV71Q21RT - Key Parameters

SAMV71Q21RT - Key Parameters	
Power Supply	3.0V to 3.6V
Maximum operating frequency	300 MHz
Architecture	ARM Cortex-M7
Memory size	<ul style="list-style-type: none"> - 2048 KiB embedded Flash - 384 KiB embedded SRAM

It is a RT microcontroller (which is a drop-in compatible non-RT version) with a high IO count and number of available peripherals.

This model was chosen since it is one of the few commercially available MCU featuring the modern ARM Cortex-M architecture as of 2020, while supporting the performance and interfacing needs of the subsystem. This solution was chosen against a non-RT MCU due to the increased reliability it offers, while still being compatible with any design changes or constraints requiring a non-RT solution, either by AcubeSAT or any users of its open-source design.

Features common to the ARM architecture or specific to the SAMV71Q21 include:

- Dedicated low power modes
- Integrated RTC functionality
- An internal temperature sensor
- Interfaces: I2C, SPI, CAN-FD, UART
- ECC in the embedded flash memory
- 2 independent internal watchdogs. Due to the Radiation Tolerance of this part, no external watchdogs are used.
- Full support for debugging features
- A large degree of hardware & software support, including a dedicated development board ([ATSAMV71-XULT](#)), IDE and configuration tools

The Attitude & Orbit Control System (AOCS) MCU is hosted on the ADCS/OBC board, which is based on the PC/104 standard. Due to a lack of available vertical space, the board hosts both of the aforementioned subsystems, each occupying one side of the PCB, shown in [Figure 1](#).

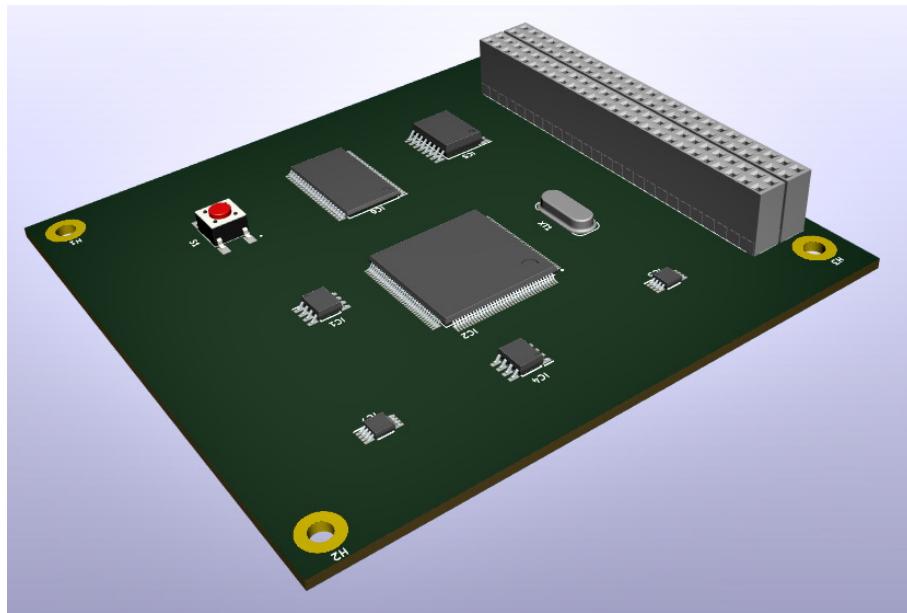


Figure 1: View of an ADCS/OBC board EM iteration

2.2 Positions and Orientation

The placement position of all sensors and actuators is defined relative to the satellite body reference frame, as shown in Fig. 2

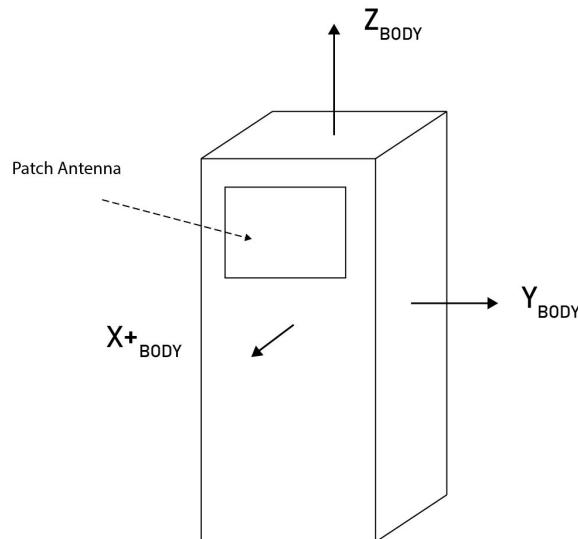


Figure 2: Satellite body reference frame

The position of the ADCS/OBC board and Magnetorquers Board is presented in Figure 3 and the place where the reaction wheel is placed (on top of the payload container) is presented in Figure 4. The whole satellite is shown in Figure 5.

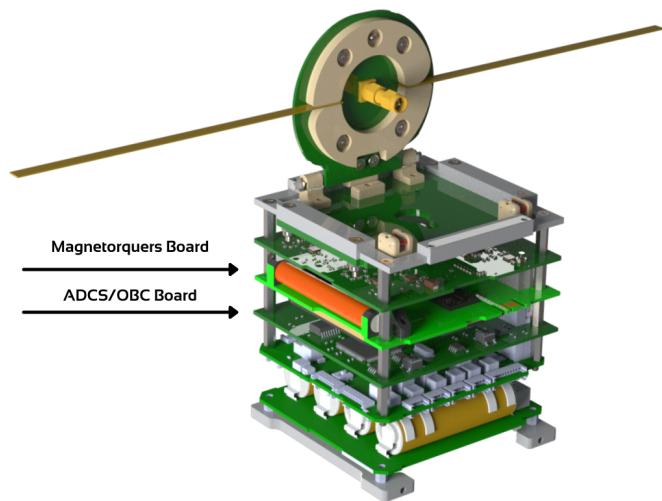


Figure 3: View of ADCS/OBC board and Magnetorquers Board in AcubeSAT

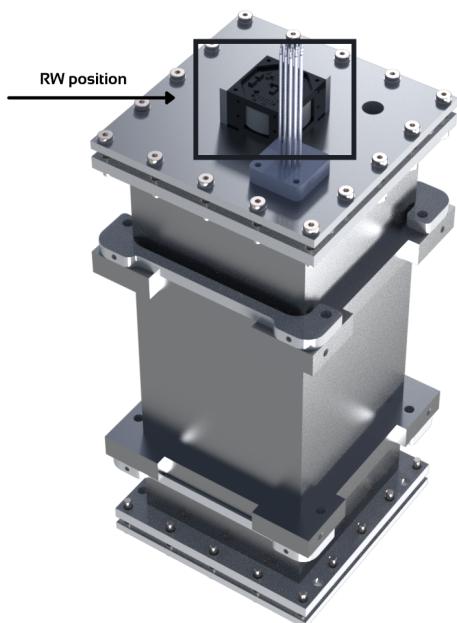


Figure 4: View of Reaction Wheel position in AcubeSAT

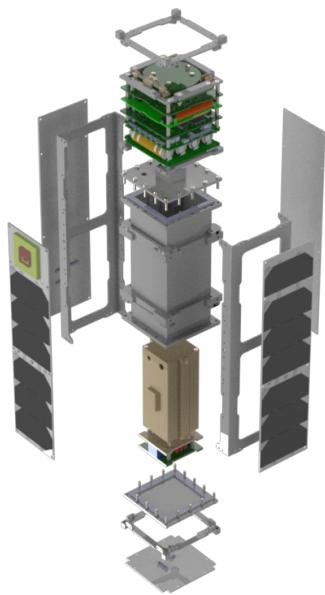


Figure 5: View of AcubeSAT

2.2.1 Sensors

2.2.1.1 Magnetometers

The RM3100 magnetometer is placed on the ADCS/OBC PCB. The Coil Sensors X, Y, Z of the magnetometer are aligned to the corresponding axis of the satellite body frame.

The other magnetometer is integrated on the Magnetorquers board and thus, follows its placement and orientation.

2.2.1.2 Gyroscope

The gyroscope is installed on the ADCS/OBC PCB. Its X, Y, Z axes are aligned to the corresponding ones of the satellite body frame.

2.2.1.3 Coarse Sun Sensor

The coarse sun sensor is comprised of photodiodes integrated on each solar panel. However, solar panels will be placed only on 5 out of 6 sides.

Thus, a photodiode on the +Z face of the satellite body frame is added, in order for the sun vector to be constructed. The additional photodiode is placed on the top of AcubeSAT, right beneath the folded deployable antenna, as shown in [Figure 6](#).

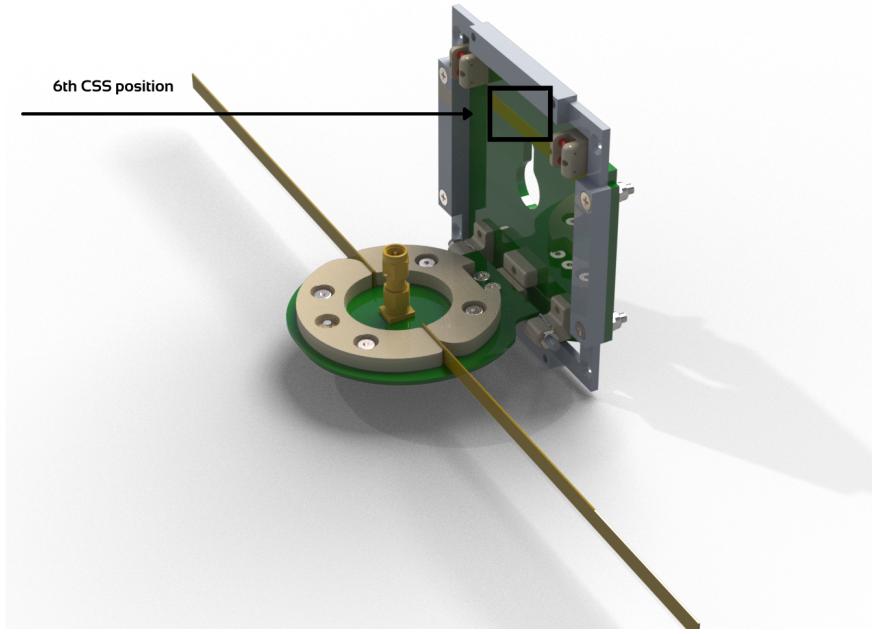


Figure 6: View of 6th CSS position in AcubeSAT

2.2.1.4 Temperature Sensors

The temperature sensors are installed on the ADCS/OBC PCB perpendicular to the corresponding X,Y body frame plane. Their X,Y,Z axes are aligned to the corresponding ones of the satellite body frame.

2.2.2 Actuators

2.2.2.1 Magnetorquers

The iMTQ board integrated below the ADCS/OBC board. The X, Y, Z axes of the board are aligned to the corresponding one of the satellite body frame.

2.2.2.2 Reaction Wheel

The reaction wheel is mounted on top of the payload container. Its rotation axis is perpendicular to the Z axis of the body frame.

2.3 Key interfaces

2.3.1 Physical Interfaces

Regarding the physical interfaces of the AOCS, the communication between AOCS and the rest of the spacecraft is performed through the CAN bus of the PC/104 bus, as specified on the [ASAT_DDJF_OBDH](#). The iMTQ board is connected via the PC/104 bus. The communication with the reaction wheel is performed through a connector placed on the AOCS PCB, directly connected to the reaction wheel itself. The communication with the sun sensor is performed through a connector placed on the AOCS PCB, directly connected to the sun sensor itself. The rest of the components are placed on the PCB, so a direct connection is performed, through the aforementioned protocols.

2.3.2 Data Interfaces

With reference to the data interfaces, data inputs and outputs are controlled by the AOCS MCU. The key interfaces between the AOCS' components and the AOCS MCU are specified in [??](#). The AOCS can send all parameters, events and functions to any subsystem through the CAN bus. It can also communicate TMTC directly to the COMMS subsystem. In case of OBC failure, the AOCS subsystem can also take over the system-level FDIR functionality.

Additionally, AOCS is capable of updating the software in-orbit by TC. The update includes uploading of new software or a software patch. The technical details of software reprogramming can be found in [ASAT_DDJF_OBSW](#). In general, the reprogramming is based on a dual-partition architecture where copies of the old and new software remain in memory to provide fault tolerance. Also, excluding bootloader, the entire MCU software can be updated.

2.4 Justification for the chosen design

AcubeSAT's AOCS physical architecture design approach aims for a low-cost, robust and reliable AOCS system, using components with relatively low power consumption, small size and high accuracy.

Regarding the sensor selection, the main focus was being able to dissipate the satellite angular velocity autonomously, without ground segment support, by utilizing the magnetometer. Specifically, the **magnetometer** is considered reliable, flexible and presents minimal restrictions, due to its capability of measuring the magnetic field with high accuracy, its small size and low power consumption. Owing to the robustness of that sensor, we opted for having a second redundant magnetometer. Additionally, in order to unambiguously define the satellite orientation, measurements from two sensors, namely the magnetometer and the sun sensor, are combined. The **sun sensor** is deployed only during AOCS nominal mode, complementing the magnetometer measurements. A **MEMS gyroscope** is deployed, enabling the satellite's accurate angular velocity definition in attitude determination, during nominal mode.

Regarding actuator selection, we aimed to safely and uninterrupted achieve the desired pointing accuracy using two low-cost and small-size actuators. **Magnetorquers** are considered a reliable and flexible way of attitude control in LEO, where the magnetic field is considered sufficient for the control of the satellite. During the AOCS detumbling mode, the magnetorquers are solely used. However, the torque of the magnetorquers is perpendicular to the aforementioned magnetic field, depriving the controller of one degree of freedom. In order to tackle this, a **reaction wheel** was selected to compensate the ancillary torque demanded to be produced by the magnetorquer on the Z axis of the body frame. This allows the satellite to achieve the desired pointing accuracy in AOCS nominal mode.

We opted for using a MCU separate from the OBDH subsystem, enabling us to achieve simpler data distribution, greater memory storage, greater processing power, while also offering redundancy. The integrated MCU on the *ISIS* MTQ board is not utilized in our design due to lack of software interface, thus it is used exclusively to calculate any required interface between the commanded torque and the MTQ. The MCU is accompanied with its own internal independent watchdog, which ensures that any latent failure in the software will be reliably detected.

3 AOCS mode definitions

In order to satisfy the AOCS requirements the following operating modes were agreed upon:

Table X: AOCS modes & System-level modes

AOCS modes	System-level modes
Stand-by mode	Commissioning / Safe mode
Detumbling mode	Commissioning / Safe / Nominal / Science mode
Nominal mode	Nominal mode

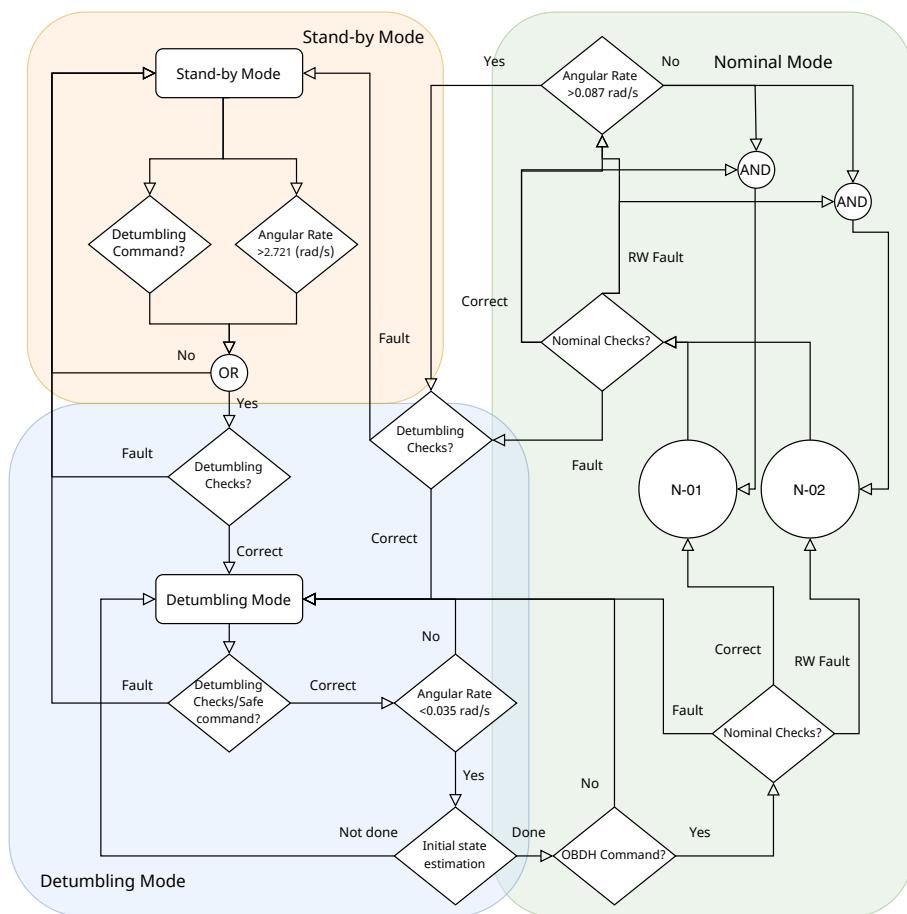


Figure 7: Mode diagram of AOCS

Requirements close-out — Review of Design

The above text is the close out reference for requirement:

ADCS-FUN-060

During Nominal mode, attitude determination and control system (ADCS) shall initiate subsystem Detumbling mode and maintain the spacecraft attitude, after receiving an OBC trigger.

3.1 Mode descriptions

In this section each AOCS mode's purpose is outlined. Descriptions are accompanied by the respective mode's control cycle.

Stand-by mode

The purpose of AOCS stand-by mode is to be a stable state for AOCS to fall back on in case of failure or whenever AOCS needs to be disabled. Sensors can be enabled by

command and their measurements are made available. During this mode, no actuation or determination is taking place.

Detumbling mode

The purpose of the AOCS detumbling mode is to autonomously dissipate and control the satellite's angular velocity. This is achieved with a B-dot controller directly using measurements from the high precision 3-axis magnetometer. In case of magnetometer failure, the redundant 3-axis magnetometer unit integrated in the iMTQ board is used. Actuation is performed solely by the magnetorquers. No satellite orbit propagation, initialized by a TLE file, is performed during this mode.

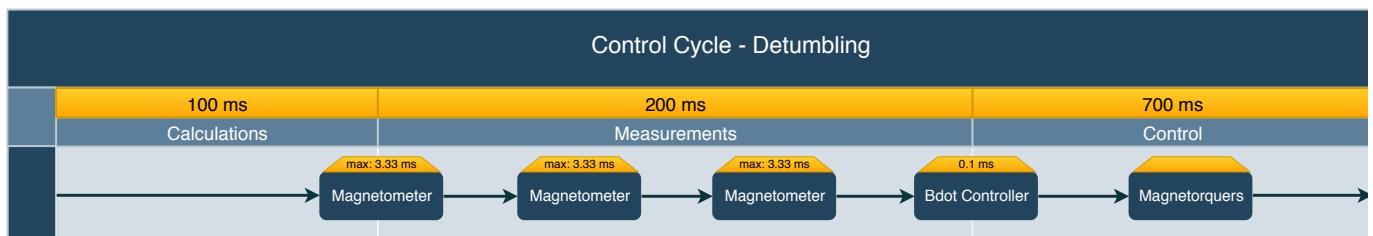


Figure 8: Detumbling mode control cycle

Nominal mode

The purpose of the AOCS nominal mode is nadir-pointing according to the requirements. Nadir-pointing enables the reliable and continuous data transmission via the S-band to the GS. Nadir-pointing is achieved by determining and controlling the satellite's attitude. Attitude is determined by combining sensor measurements from the magnetometer, the sun sensors and the gyroscope. Attitude control is carried out by the magnetorquers and the reaction wheel.

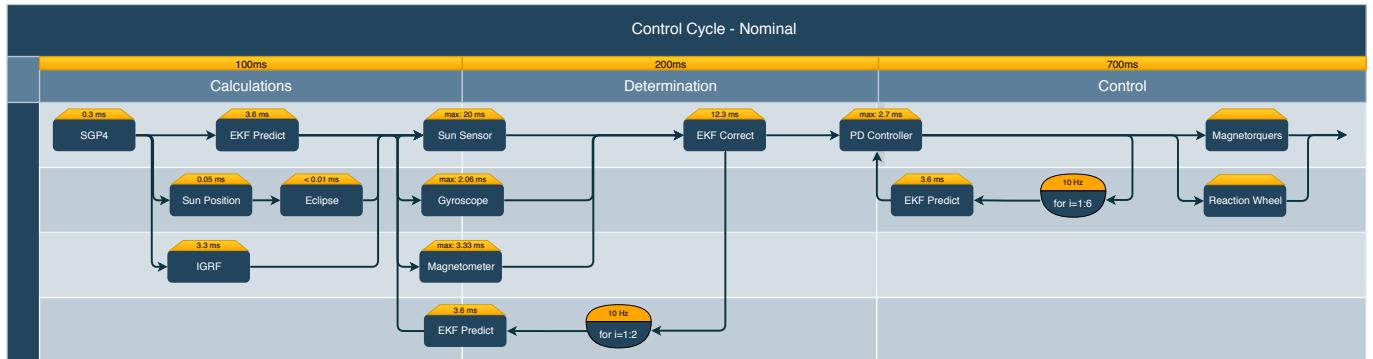


Figure 9: Nominal mode control cycle



Requirements close-out — Review of Design

The above text is the close out reference for requirement:

ADCS-FUN-120

The ADCS shall receive measurements from the subsystem sensors at any ADCS mode autonomously, without ground contact.

3.2 Attitude Profiles

Table XI: AOCS Mode & Attitude Profile

AOCS mode	Attitude Profile
Stand-by mode	S-01
Nominal mode	N-01 / N-02
Detumbling mode	D-01

S-01

This is the default attitude profile of AOCS stand-by mode. Sensor measurements can be made available by command. No actuation is performed.

N-01

This is the default attitude profile of AOCS nominal mode. Attitude determination is accomplished combining measurements from the high-precision magnetometer, the sun sensor and the gyroscope. The satellite orbit is propagated using SGP4, which is frequently initialized by a TLE file. Actuation is carried out using both the magnetorquers and the reaction wheel. The controller used is a PD with a set of 2 constant gains.

N-02

This is an optional attitude profile of AOCS nominal mode which is enabled in case of a reaction wheel failure. Attitude determination is accomplished combining measurements from the high-precision magnetometer, the sun sensor and the gyroscope. The satellite orbit is propagated using SGP4, which is frequently initialized by a TLE file. Actuation is carried out using only the magnetorquers. The controller used is a PD with a set of 2 constant gains.

D-01

This is the default attitude profile of AOCS detumbling mode. No attitude determination is taking place. The only sensor measurements are from the high-precision magnetometer. If the magnetometer fails, the *ISIS* magnetometer unit can be used as an alternative.



Actuation is performed by the magnetorquers. AOCS makes use of a B-dot controller, which is directly fed the magnetometer measurements.

Attitude profiles can be swapped and maintained by TC or by a command from the OBDH unit.

Sun pointing

This is an optional attitude profile of AOCS nominal mode being utilized in specified orbits, for which the power budget does not meet the mission's requirements. The attitude determination and orbit propagator algorithms remain the same as in N-01. A PD controller that aims to achieve sun-pointing is implemented and the commanded torque is split between both the actuators.

Requirements close-out — Review of Design

The above text is the close out reference for requirements:

ADCS-FUN-095	The ADCS shall switch between any attitude profile by trigger, timer or TC.
ADCS-FUN-100	The ADCS shall maintain a specific attitude profile for a given amount of time if requested by TC.

3.3 Triggers

AOCS can switch modes after a command from the OBC unit has been received. AOCS may also switch modes based on the system-level state of AcubeSAT, not solely on internal AOCS triggers.

Requirements close-out — Review of Design

The above text is the close out reference for requirements:

ADCS-FUN-085	The ADCS shall switch between any ADCS mode by trigger, timer or TC.
ADCS-FUN-090	The ADCS shall maintain a specific ADCS mode for a given amount of time if requested by TC.

All transitions are implemented as follows:

Detumbling Mode → Nominal Mode: AOCS switches to AOCS nominal mode when the angular rate of the satellite, which is approximated by the B-dot metrics [Section 3.3](#), decreases to a rate lower than the 0.035 rad/sec on each satellite body frame axes threshold and all corresponding diagnostic checks have finished successfully.



Requirements close-out — Review of Design

The above text is the close out reference for requirement:

ADCS-FUN-070

The ADCS shall switch from subsystem Detumbling to subsystem Nominal mode based on an angular velocity threshold of 0.035 rad/s on the spacecraft's X, Y and Z axes.

Nominal Mode → Detumbling Mode: AOCS switches to AOCS detumbling mode when the angular rate of the satellite exceeds the threshold rate of 0.087 rad/sec on any satellite body frame axis.

Requirements close-out — Review of Design

The above text is the close out reference for requirement:

ADCS-FUN-080

The ADCS shall switch from subsystem Nominal to subsystem Detumbling mode based on an angular velocity threshold of 0.087 rad/s on the spacecraft's X, Y and Z axes.

Detumbling Mode → Stand-by Mode: AOCS switches to AOCS stand-by mode either when the corresponding diagnostic checks have failed, or when the corresponding command from the OBDH unit is received.

Nominal Mode → Stand-by Mode: AOCS switches to AOCS stand-by mode either when the corresponding diagnostic checks have failed, or when the corresponding command from the OBDH unit is received.

Stand-by Mode → Detumbling Mode: AOCS switches to AOCS detumbling mode when the corresponding command from the OBC unit is received, or if the magnitude of the angular rate of the satellite is higher than the 2.72 rad/s threshold, provided that no fault in detumbling mode has been detected.

Requirements close-out — Review of Design

The above text is the close out reference for requirement:

ADCS-FUN-190

The ADCS shall switch from subsystem Safe mode to Detumbling mode when the magnitude of the angular velocity surpasses 2.72 rad/s.

Subsystem safe mode, as defined in the above requirement, corresponds to subsystem stand-by mode

Approximation of Angular Velocity using B-dot metrics

Additionally, during AOCS detumbling mode, where B-dot is enabled, the angular velocity of the satellite needs to be defined in order for the angular velocity threshold, which triggers the transition from AOCS detumbling mode to AOCS nominal mode, to be determined regularly. However, since no sensors, apart from magnetometers, are active in AOCS detumbling mode, the angular velocity of the satellite needs to be ap-

proximated utilizing the B-dot metrics. The aforementioned calculation is performed according to [Equation \(1\)](#).

$$\mathbf{b} \times \boldsymbol{\omega} = \dot{\mathbf{b}} \longrightarrow \text{skew}(\mathbf{b}) \cdot \boldsymbol{\omega} = \dot{\mathbf{b}} \quad (1)$$

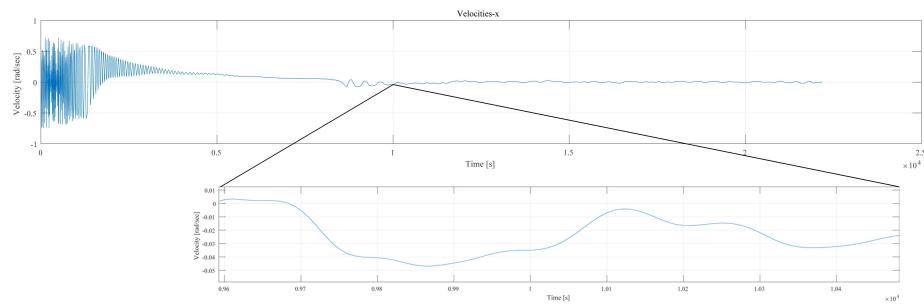
where \mathbf{b} is the magnetic field expressed on the body frame of the satellite and $\boldsymbol{\omega}$ is the angular velocity of the satellite body from the orbit to the body frame, expressed in the body frame. Therefore, if the $\text{skew}(\mathbf{b})$ is an invertible matrix:

$$\boldsymbol{\omega} = \text{skew}(\mathbf{b})^{-1} \cdot \dot{\mathbf{b}} \quad (2)$$

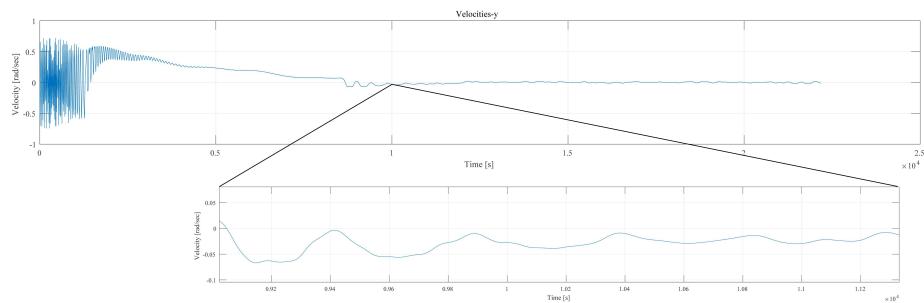
However, the $\text{skew}(\mathbf{b})$ is close to singular, thus MATLAB is unable of performing the operation described in [Equation \(2\)](#). Therefore, the angular velocity can only be approximated, as shown in [Equation \(3\)](#)

$$\boldsymbol{\omega} = \frac{\text{skew}(\mathbf{b}) \cdot (-\dot{\mathbf{b}})}{\mathbf{b}^T \mathbf{b}} \quad (3)$$

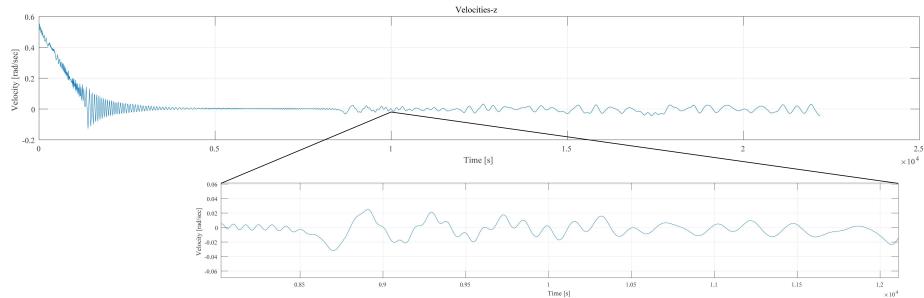
Since the angular velocity is approximated and not accurately calculated, an algorithm is created for verifying that the angular threshold, which triggers the transition from AOCS detumbling mode to AOCS nominal mode, is correctly reached. This algorithm uses the approximated angular velocity as an input and checks the angular rate on all axes during each timestep. If for a defined number of continuous timesteps the threshold is reached, while filtering with a specified number of "allowed" exceptions, the algorithm indicates that AOCS nominal mode shall be initiated, as an output. The actual velocity and the approximation of the velocity are shown on diagrams [Section 3.3](#) and [Section 3.3](#), as well as the time of the AOCS nominal mode first initiation.



(a) X-axis velocity

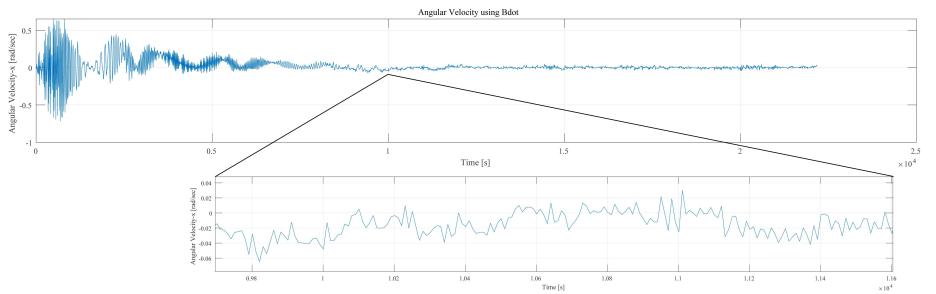


(b) Y-axis velocity

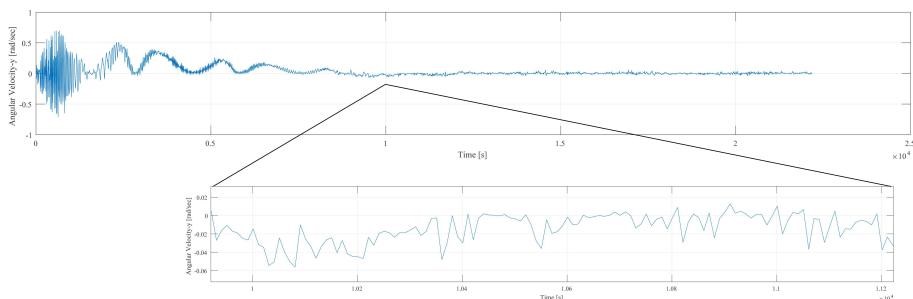


(c) Z-axis velocity

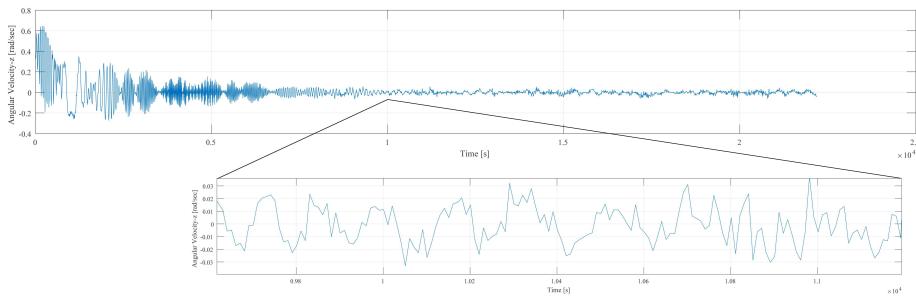
Figure 10: Angular velocity during AOCS Detumbling Mode



(a) X-axis velocity



(b) Y-axis velocity



(c) Z-axis velocity

Figure 11: Angular velocity approximation during AOCS Detumbling Mode

We can observe that the approximation is inaccurate for high velocities, however in the area of interest, namely time-periods where angular velocities are lower than 0.1 rad/sec, the actual angular rate is approximated rather accurately. Consequently, the transition to AOCS nominal mode is triggered at the correct time. *The correct time for the AOCS nominal mode initiation is defined as the time when the actual angular velocity on each satellite body frame axes is lower than the 0.035 rad/sec threshold.*

3.4 AOCS mode sequence

Spacecraft Commissioning mode: AOCS initially operates in stand-by mode. Upon command from OBC, AOCS switches to detumbling mode.



Spacecraft Nominal mode: AOCS initially operates in detumbling mode. If [Detumbling → Nominal trigger](#) is activated AOCS switches to nominal mode. If [Nominal → Detumbling trigger](#) is activated AOCS switches to detumbling mode.

Spacecraft Science mode: AOCS operates in detumbling mode.

Spacecraft Safe mode: AOCS initially operates in detumbling mode. Upon command from OBDH the AOCS can switch to both stand-by or nominal mode.

4 AOCS Budgets

4.1 Error metric definition

In order to correctly characterize the errors present in both knowledge and performance throughout the whole simulation, an appropriate error metric must be defined. This metric is taken to be the median of the absolute value of the raw error vector. 95% confidence intervals are acquired by the bootstrap method. Bootstrap sampling is repeated 10 times, and intervals are calculated using the basic percentile method.

4.2 Brief spacecraft geometry and orbit characteristics

Before deriving the AOCS budgets, it is necessary to remark and define the critical constants related to the geometrical and orbital characteristics of the satellite.

4.2.1 Geometrical characteristics

Initially, some typical values are defined. The values of the principal moments of inertia are $I_p = \begin{bmatrix} 0.04127073921 & 0.041018997570 & 0.00690030456 \end{bmatrix}^T \text{kgm}^2$, while the principal axis of inertia are the rows of the following

$$P = \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} = \begin{bmatrix} -1 & 0.01 & -0.04 \\ 0.01 & 1 & 0.00 \\ -0.04 & 0.00 & 1.00 \end{bmatrix} \quad (4)$$

thus, the inertia matrix is derived as:

$$I = P I_p P^T = \begin{bmatrix} 0.0412 & 0.0000 & 0.0014 \\ 0.0000 & 0.0410 & 0.0000 \\ 0.0014 & 0.0000 & 0.0070 \end{bmatrix} \text{kgm}^2 \quad (5)$$

The center of mass is taken as $C_m = \begin{bmatrix} 0.00415 & 0.00116 & 0.0016 \end{bmatrix}$ m. It is also assumed that the residual magnetic dipole moment is 0.05 Am² for each satellite body reference

frame axis.

4.2.2 Orbital characteristics

Selecting a satellite altitude of $R_s = 500$ km and knowing that the earth mass is equal to $M=5.972 \cdot 10^{24}$ kg and the gravitational constant $G=6.67428 \cdot 10^{-11} \text{ m}^3\text{kg}^{-1}\text{s}^{-2}$, then the satellite angular velocity relative to Earth ω_o , the satellite's velocity in orbit v_s and the orbit period T_o are calculated as:

$$\omega_o = \sqrt{\frac{G \cdot M}{R_s^3}} \quad (6)$$

$$v_s = \sqrt{\frac{G \cdot M}{R_s}} \quad (7)$$

$$T_o = \frac{2\pi}{\omega_o} \quad (8)$$

4.2.3 Disturbances

There are four types of environmental disturbances taken into account:

- Gravitational torque
- Residual magnetic dipole moment
- Aerodynamic drag
- Solar pressure

The disturbances modeling is analyzed in 6.2.3 with the orbital and geometrical characteristics previously defined.

4.3 Actuator sizing budgets

The actuator sizing budget is separated in two sections, the torque and the momentum budget. Thus, both budgets will be initially defined separately.

4.3.1 Torque budget

4.3.1.1 Detumbling mode

During detumbling mode, actuation is based solely on magnetorquers. Therefore, the total torque is calculated by:

$$\mathbf{T}_m = \mathbf{m} \times \mathbf{b} \quad (9)$$

where \mathbf{m} is the magnetic dipole produced by the magnetorquers and \mathbf{b} is the earth magnetic field vector expressed in the satellite body frame.

Due to the cross product, it is evident that a magnetic torque cannot be directly commanded. Instead, by taking into account the current magnitude of the geomagnetic field, the magnetic dipole moment in each magnetorquer is commanded.

Based on the magnetorquer's physical manufacturing, the coils have limited magnetic dipole moment creation capacity. The minimum capacity for which our mission's requirements are feasible needs to be determined, based on our simulations. As seen in the following simulations, the maximum magnetic dipole moment is gradually reduced, in order to determine at what point the satellite will become uncontrollable. The initial angular velocities are set to $\omega_0 = \begin{bmatrix} \frac{\pi}{2} & \frac{\pi}{2} & \frac{\pi}{8} \end{bmatrix}^T \text{ rad s}^{-1}$ and the simulation duration is set to 8 orbits. For the torque budget during the AOCS detumbling mode, the used simulation is detailed in [Section 6.1](#), the simulations are carried out using the SSO orbit TLE, as explained in [Section 5.1](#).

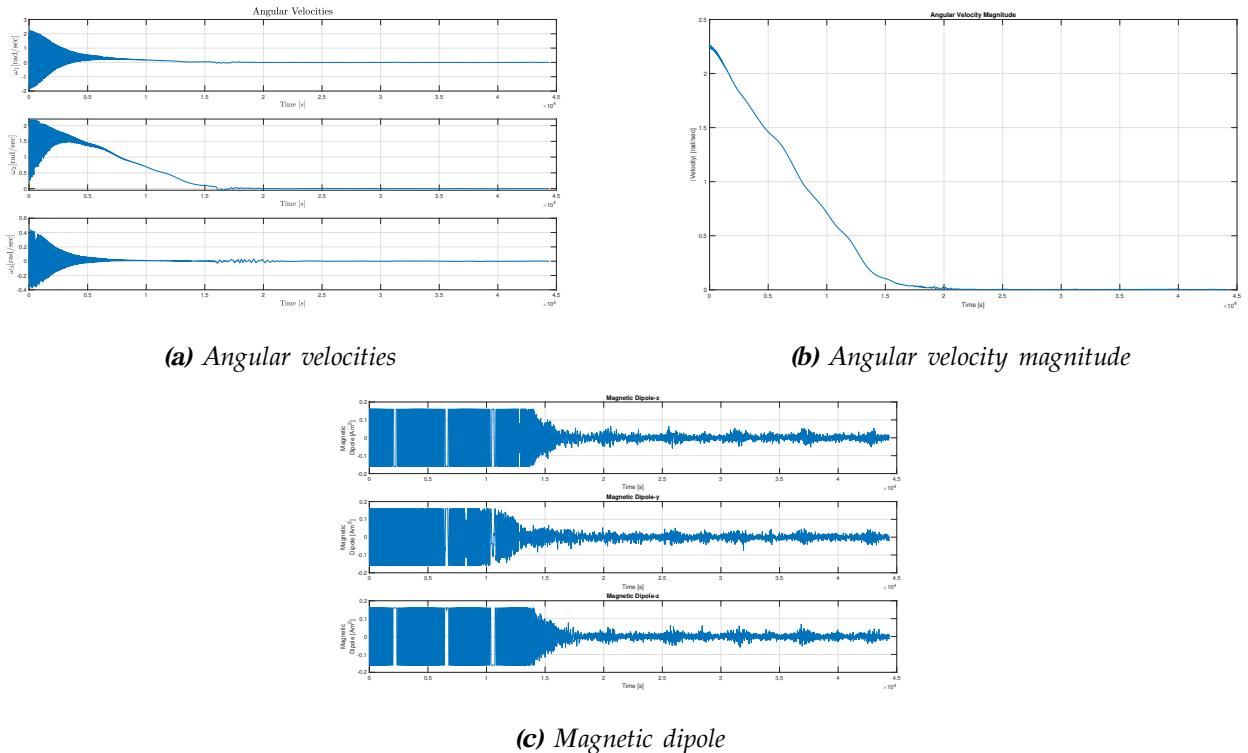
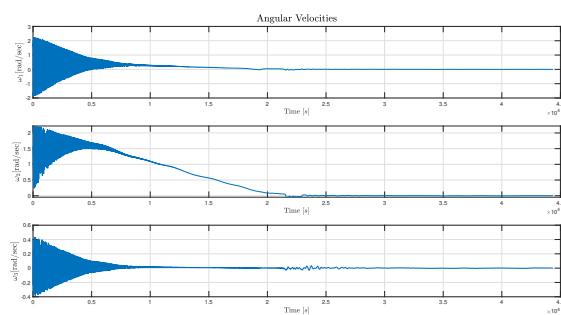
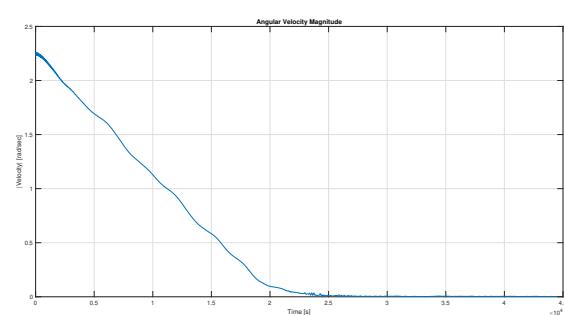


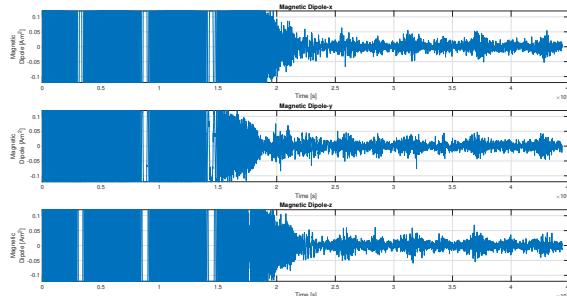
Figure 12: Torque budgets for $m_{\max} = 0.16 \text{ Am}^2$



(a) Angular velocities

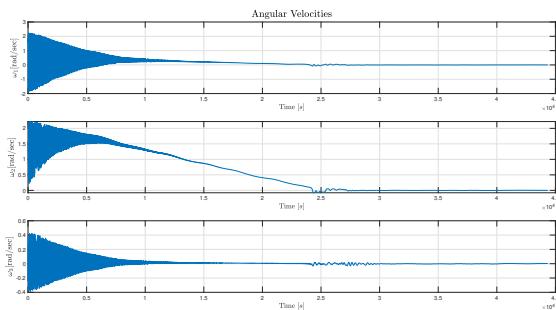


(b) Angular velocity magnitude

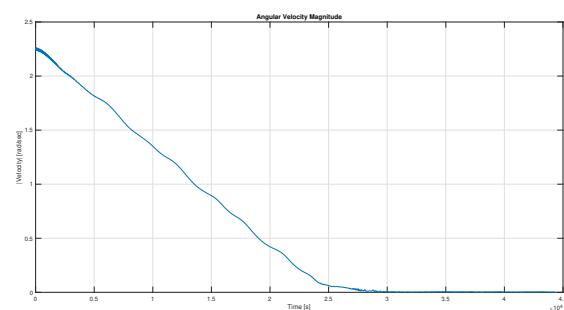


(c) Magnetic dipole

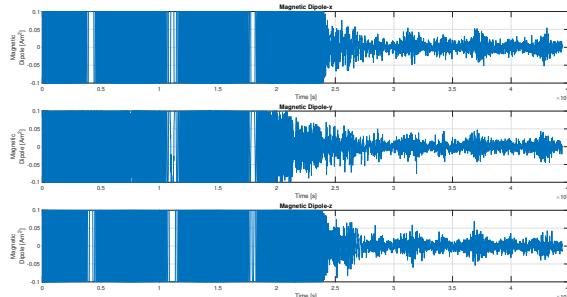
Figure 13: Torque budgets for $m_{\max} = 0.12 \text{ Am}^2$



(a) Angular velocities



(b) Angular velocity magnitude



(c) Magnetic dipole

Figure 14: Torque budgets for $m_{\max} = 0.1 \text{ Am}^2$

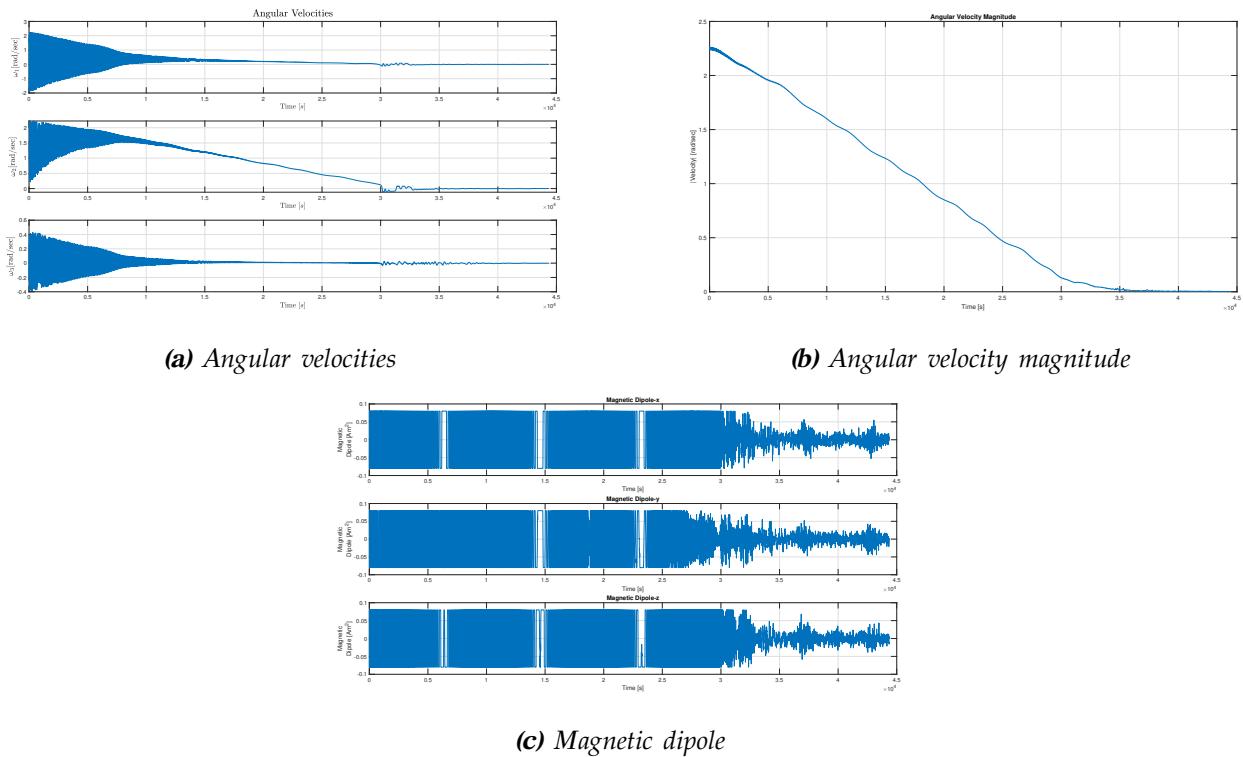


Figure 15: Torque budgets for $m_{\max} = 0.08 \text{ Am}^2$

From Figure 15a we can observe that, for maximum magnetic dipole $> 0.08 \text{ Am}^2$, the magnetorquers are able to detumble the spacecraft, with an initial angular velocity $\omega_0 = \begin{bmatrix} \frac{\pi}{2} & \frac{\pi}{2} & \frac{\pi}{8} \end{bmatrix}^T$, within 8 orbits. On the contrary, if the maximum magnetic dipole is $< 0.08 \text{ Am}^2$, the magnetorquers need more than 8 orbits to detumble the spacecraft, with the same initial angular velocity.

4.3.1.2 Nominal mode

For the derivation of the torque and momentum budget in nominal mode, the attitude determination system was included in the simulations. Also, as explained in Fig. 9, according to our control cycle, for 0.3 sec the actuators do not apply a given torque.

In nominal mode, both the magnetorquers and the reaction wheel are being used cooperatively. To derive the torque budget for the actuators, the following parameters are changed:

- Maximum reaction wheel torque is considered to be $T_{rw} = 10^{-4} \text{ Nm}$
- Maximum magnetic dipole moment produced by each magnetorquer is 0.2 Am^2

The actuator parameters that remain constant throughout the simulations are the reaction wheel axial moment of inertia $J_w = 1.9 \cdot 10^{-6} \text{ kgm}^2$ and the maximum angular rate of the reaction wheel $\Omega_{\max} = 15000 \text{ rpm}$. The desired torque is calculated using a PD

controller as:

$$\mathbf{T}_d = -\text{sign}(n) K_p \mathbf{e} - K_d \boldsymbol{\omega} \quad (10)$$

where K_p, K_d denote as gain matrices, $\boldsymbol{\omega}$ the angular velocities, n the scalar part and \mathbf{e} the vector part of a quaternion expressing the relative rotation from the current state quaternion to the identity quaternion. In the simulations conducted, a maximum value was selected for the maximum reaction wheel torque. In order to achieve pointing the quaternion from orbit to body frame must be equal to the identity quaternion. The simulations run for 1 orbit.

Using maximum $T_{rw} = 10^{-5}$ Nm, the simulation outputs are [Figure 16](#), [Figure 17](#).

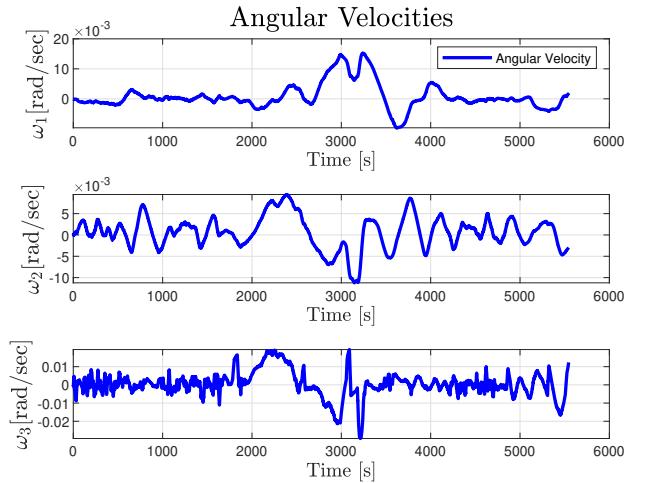
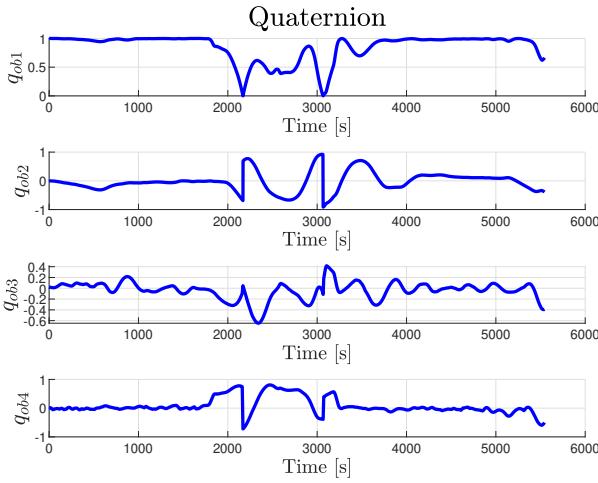


Figure 16: Quaternion from orbit to body frame ($T_{rw} = 10^{-5}$ Nm)

Figure 17: Velocities ($T_{rw} = 10^{-5}$ Nm)

Using maximum $T_{rw} = 5 \cdot 10^{-6}$ Nm, the simulation outputs are [Figure 18](#), [Figure 19](#).

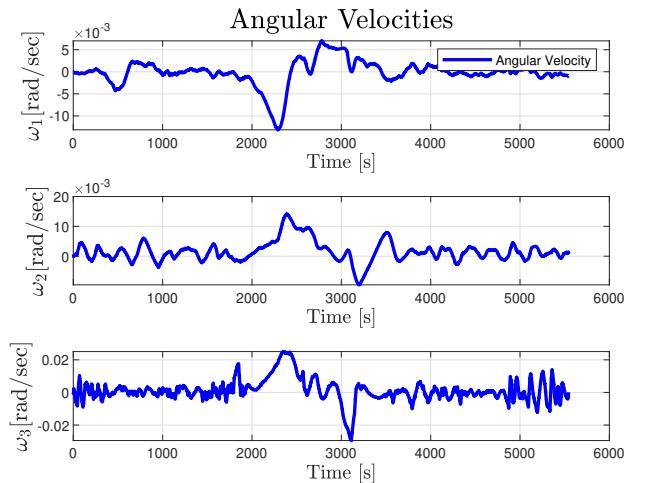
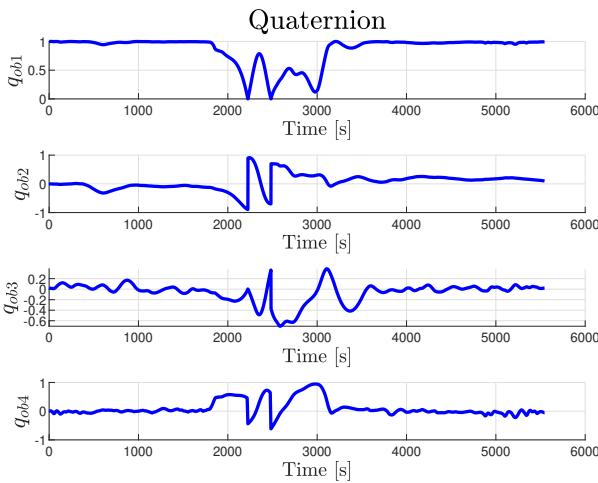


Figure 18: Quaternion from orbit to body frame ($T_{rw} = 5 \cdot 10^{-6}$ Nm)

Figure 19: Velocities ($T_{rw} = 5 \cdot 10^{-6}$ Nm)

Using maximum $T_{rw} = 4 \cdot 10^{-6}$ Nm, the simulation outputs are [Figure 20](#), [Figure 21](#).

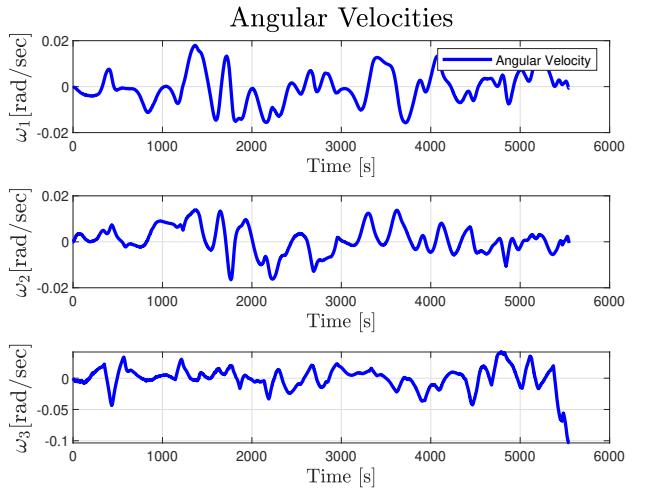
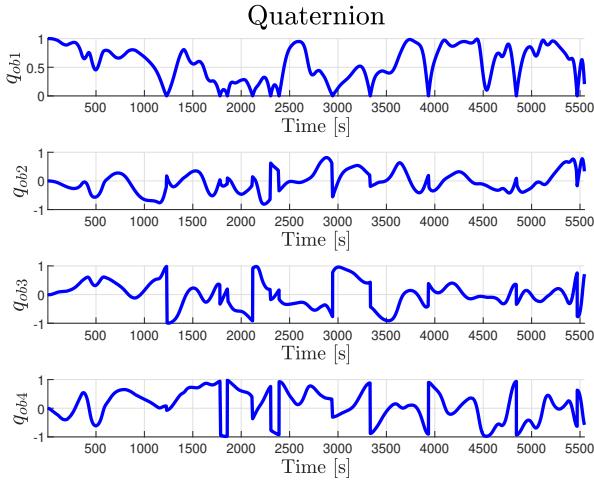


Figure 20: Quaternion from orbit to body frame
 $(T_{rw} = 4 \cdot 10^{-6}$ Nm)

Figure 21: Velocities ($T_{rw} = 4 \cdot 10^{-6}$ Nm)

It appears that for a maximum reaction wheel torque lower than $5 \cdot 10^{-6}$ Nm it is not possible to control the AcubeSAT.

To derive the magnetic torque budget the maximum reaction wheel torque is reset to 10^{-4} Nm, while the other parameters remain unchanged. The magnetic torque is constrained by the saturation of the magnetorquers.

Using a maximum dipole moment of 0.16 Am 2 for each magnetorquer, the simulation outputs are [Figure 22](#), [Figure 23](#).

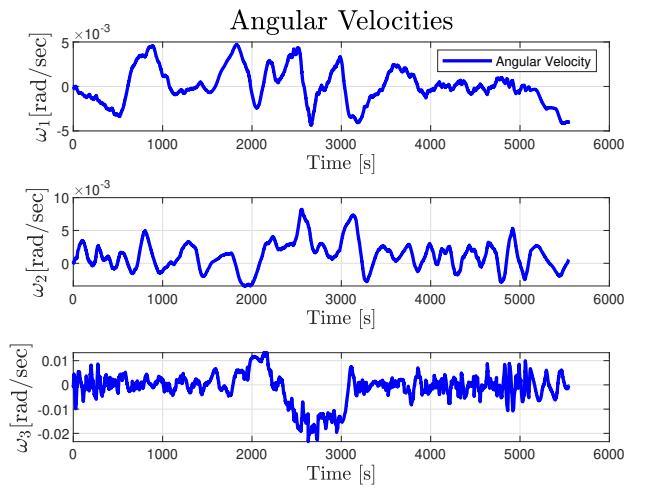
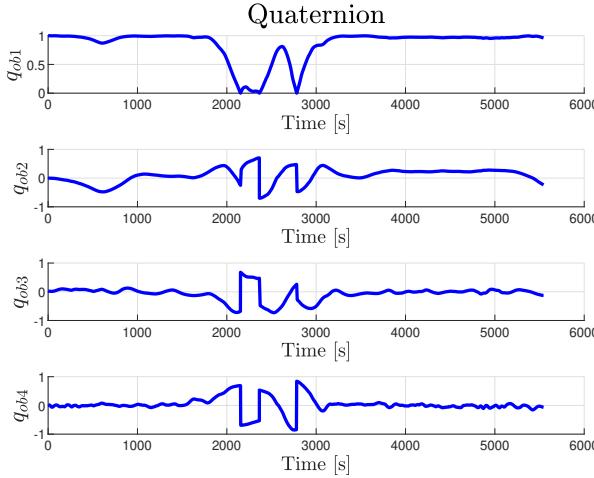


Figure 22: Quaternion from orbit to body frame
 $(M_{max} = 0.16$ Am 2)

Figure 23: Velocities ($M_{max} = 0.16$ Am 2)

Using a maximum dipole moment of 0.14 Am 2 for each magnetorquer, the simulation outputs are [Figure 24](#), [Figure 25](#).

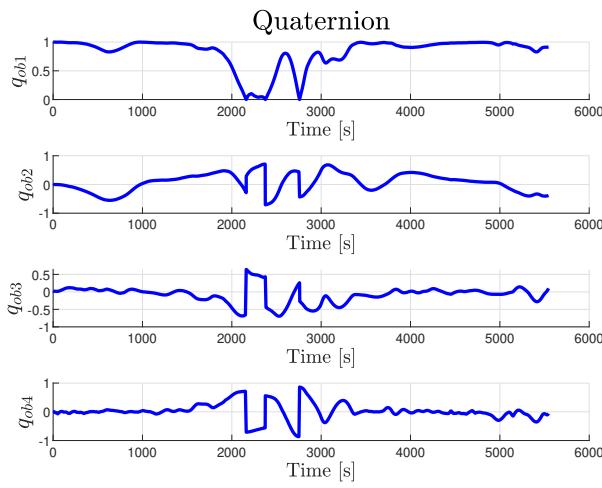


Figure 24: Quaternion from orbit to body frame
 $(M_{\max} = 0.14 \text{ Am}^2)$

Using a maximum dipole moment of 0.12 Am^2 for each magnetorquer, the simulation outputs are [Figure 26](#), [Figure 27](#).

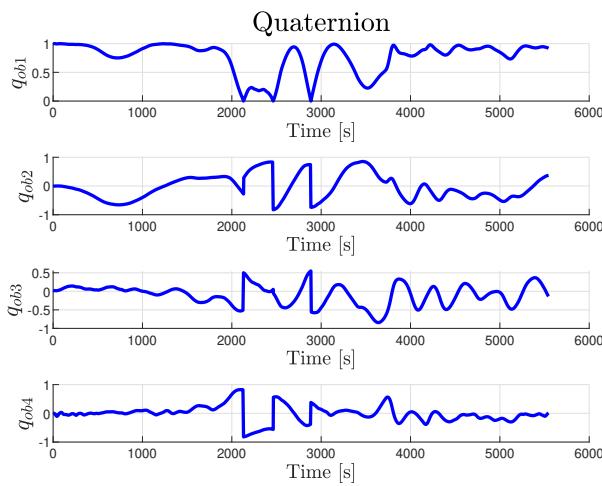


Figure 26: Quaternion from orbit to body frame
 $(M_{\max} = 0.12 \text{ Am}^2)$

It appears that for a maximum magnetic dipole moment less than 0.14 Am^2 the state of the AcubeSAT cannot be controlled.

4.3.2 Momentum budget

The total angular momentum of the satellite is calculated as:

$$\mathbf{H} = I\boldsymbol{\omega} + \mathbf{h} \quad (11)$$

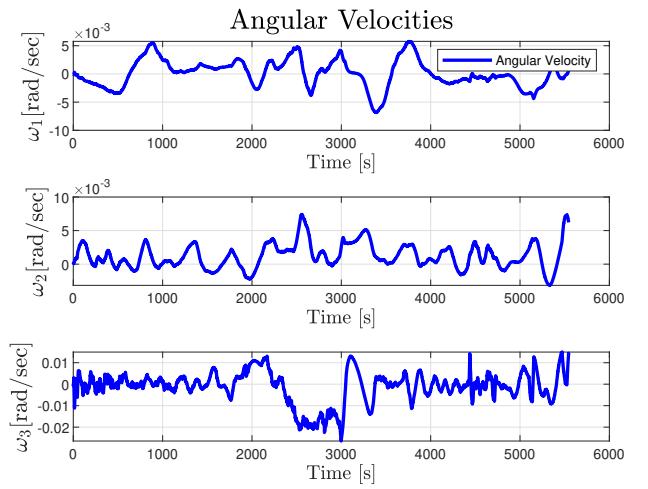


Figure 25: Velocities ($M_{\max} = 0.14 \text{ Am}^2$)

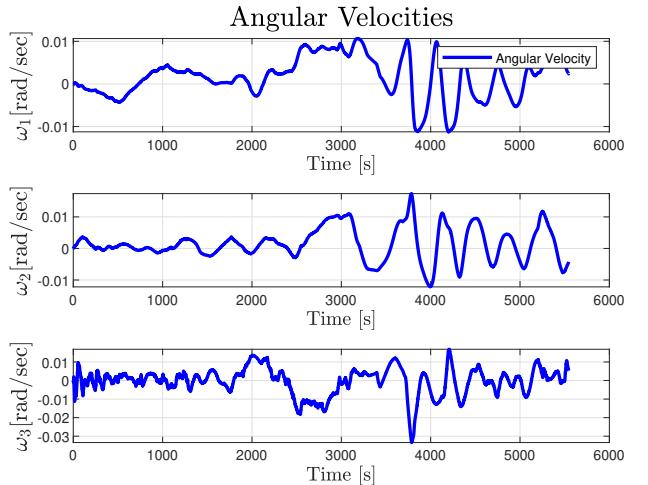


Figure 27: Velocities ($M_{\max} = 0.12 \text{ Am}^2$)

where I is the AcubeSAT inertia matrix, ω is the angular velocity relative to the ECI frame and h is the reaction wheel angular momentum derived from

$$h = J_w \cdot \Omega_{rw} \quad (12)$$

where J_w is the axial moment of inertia of the reaction wheel and Ω_{rw} is its rotation rate. To derive the momentum budget for the actuators the maximum reaction wheel torque is set to 10^{-4} N·m and the maximum dipole moment produced by the magnetorquers to 0.2 A·m 2 , while the following parameters are changed:

- Reaction wheel maximum angular velocity $\Omega_{rw_{max}}$
- Reaction wheel axial moment of inertia J_w

In our reaction wheel model, we actually do not use the maximum rotational speed that the reaction wheel can reach, but a limit after which we start desaturating gradually the wheel. This limit is 5000 rpm less than the maximum rotational speed of the wheel.

From [Equation \(12\)](#) the reaction wheel is dependent on both the axial moment of inertia J_w and the angular velocity Ω_{rw} . Thus, in order to define the budget for each of the two variables the other is predefined as a constant.

At first, in order to derive the angular velocity of the reaction wheel budget, J_w is set to $1.9 \cdot 10^{-6}$ kg·m 2 . Afterwards, J_w is set to $9.5 \cdot 10^{-6}$ kg·m 2 and to $9.5 \cdot 10^{-7}$ kg·m 2 . The simulations run for 2 orbits.

While running the simulations below, we consider the maximum angular velocity of the reaction wheel to be constant, equal to 15000 rpm, thus the maximum momentum storage of the reaction wheel varies proportionally to the J_w according to [Equation \(12\)](#) for $\Omega_{rw} = \text{constant}$.

Using $J_w = 1.9 \cdot 10^{-6}$ kg·m 2 , the simulation outputs are [Figure 28](#), Fig

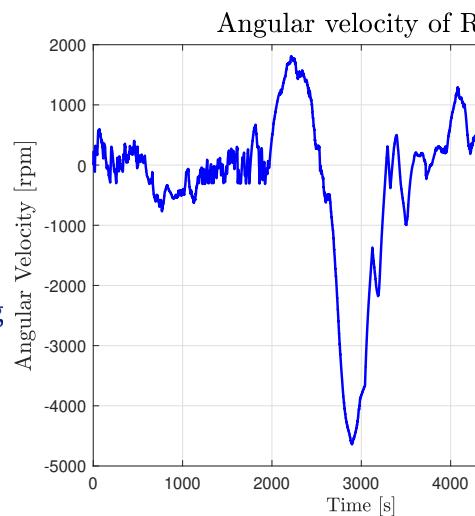


Figure 28: Angular Velocity
($J_w = 1.9 \cdot 10^{-6}$ kgm 2)

Using $J_w = 9.5 \cdot 10^{-7} \text{ kg} \cdot \text{m}^2$, the simulation outputs are [Figure 30](#), [Figure 31](#)

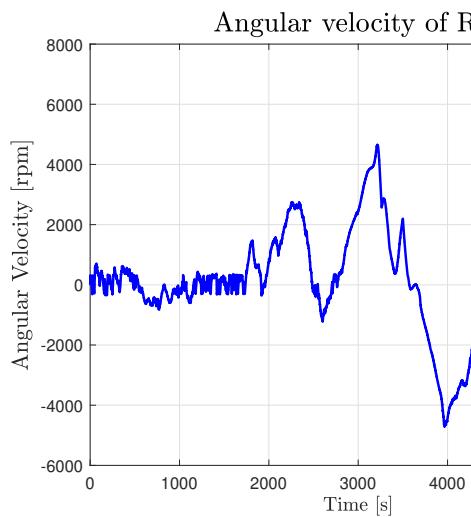


Figure 30: Angular Velocity of Reaction Wheel
($J_w = 9.5 \cdot 10^{-7} \text{ kgm}^2$)

Using $J_w = 3.8 \cdot 10^{-7} \text{ kg} \cdot \text{m}^2$, the simulation outputs are [Figure 32](#), [Figure 33](#)

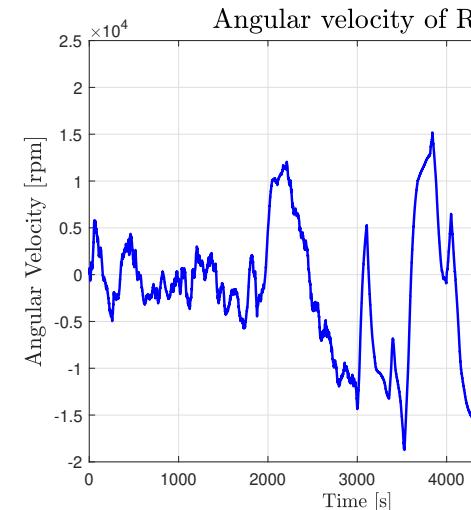
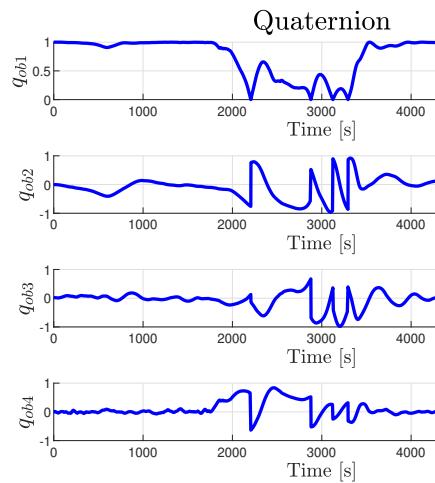


Figure 32: Angular Velocity of Reaction Wheel
($J_w = 3.8 \cdot 10^{-7} \text{ kgm}^2$)

It appears that for a reaction wheel with axial moment of inertia less than $5 \cdot 10^{-7} \text{ kg} \cdot \text{m}^2$ pointing cannot be achieved.

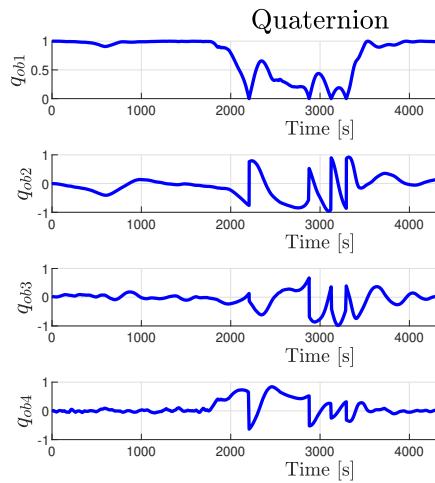
We have to clarify that, in the simulation where we used $J_w = 1.9 \cdot 10^{-7} \text{ kg} \cdot \text{m}^2$, the limit from which we started gradually desaturating the reaction wheel was reduced to 5000 rpm. The nominal limit in our simulations for $J_w = 1.9 \cdot 10^{-6} \text{ kgm}^2$ is 10000 rpm.

Secondly, in order to derive the J_w of the reaction wheel budget, the maximum angular velocity of the reaction wheel, and thus the saturation limit of the wheel, is studied for various values. J_w is defined as a constant $= 1.9 \cdot 10^{-6} \text{ kg} \cdot \text{m}^2$, consequently the maximum momentum storage of the reaction wheel varies proportionally to the $\Omega_{rw\max}$ according to [Equation \(12\)](#) for $J_w = \text{constant}$.



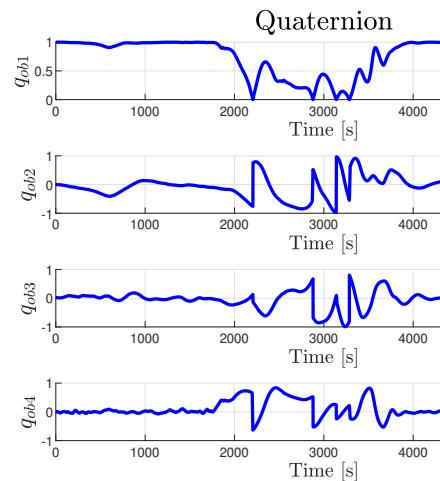
Using $\Omega_{rw_{max}} = 13000$ rpm, the simulation outputs are Figure 34, Fig

Figure 34: Quaternions ($\Omega_{rw_{max}} =$



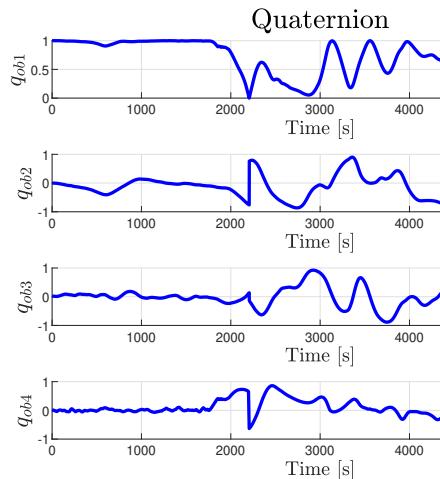
Using $\Omega_{rw_{max}} = 11000$ rpm, the simulation outputs are Figure 37, Fig

Figure 36: Quaternions ($\Omega_{rw_{max}} =$



Using $\Omega_{rw_{max}} = 9000$ rpm, the simulation outputs are Figure 39, Fig

Figure 38: Quaternions ($\Omega_{rw_{max}} =$



Using $\Omega_{rw_{max}} = 7000$ rpm, the simulation outputs are Figure 41, Fig

Figure 40: Quaternions ($\Omega_{rw_{max}} =$

It appears that for a reaction wheel with maximum angular velocity less than 9000 rpm pointing cannot be achieved.

4.4 Pointing budget

In order to derive the AOCS pointing budget, the following errors are calculated:

- Absolute Knowledge Error
- Absolute Performance Error
- Mean Knowledge Error
- Mean Performance Error
- Relative Knowledge Error

- Relative Performance Error
- Drift Performance Error

as described in [ECSS-E-ST-60-10C](#).

For the derivation of the pointing budget in nominal mode and particularly the performance errors, simulations are carried out with the presence of the attitude determination system. The simulations are carried out using the total AOCS simulation.

Absolute Knowledge Error

Absolute knowledge error is the instantaneous value of the knowledge error at any given time. It is expressed as $e_k(t)$. Confidence intervals were calculated by the bootstrap method as mentioned in the error metrics definition [Section 4.1](#). 95% confidence intervals for the median error in each axis are given for SSO in the following table: (errors are in Euler angles and expressed in degrees)

	Z	Y	X
Inside eclipse	[16.4829,18.0426]	[15.5208,16.0531]	[23.7797,24.7498]
Outside of eclipse	[0.1968,0.2027]	[0.1036,0.1066]	[0.2197,0.2256]
Total	[0.3864,0.3918]	[0.2815,0.2863]	[0.4426,0.4500]

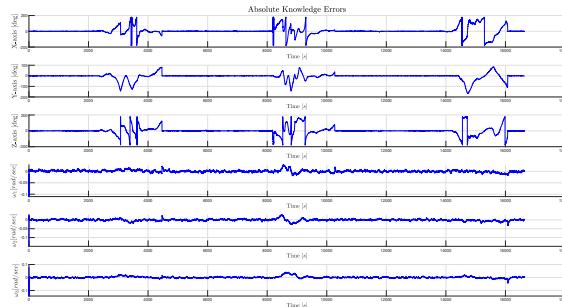


Figure 42: Absolute Knowledge Error in SSO Orbit

Absolute Performance Error

Absolute performance error is the instantaneous value of the performance error at any given time. It is expressed as $e_p(t)$. Confidence intervals were calculated by the bootstrap method as mentioned in the error metrics definition [Section 4.1](#). 95% confidence intervals for the median error in each axis are given for SSO in the following table: (errors are in Euler angles and expressed in degrees)

	Z	Y	X
Eclipse included	[15.1481,15.4128]	[14.032,14.3829]	[32.5623,32.8199]
Eclipse not included	[6.6304,6.8898]	[8.2781,8.4576]	[19.8042,20.0075]
Eclipse and following 500 sec excluded	[6.3973,6.5678]	[7.8937,8.0880]	[19.1527,19.4465]

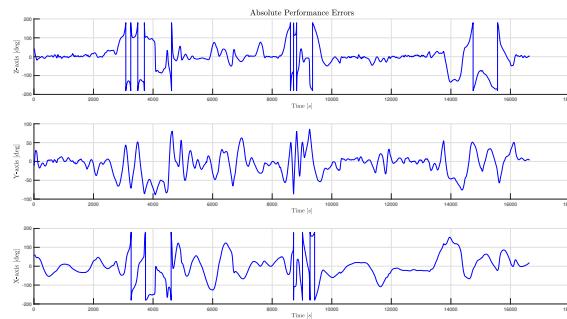


Figure 43: Absolute Performance Error in SSO Orbit

Mean Knowledge Error

Mean knowledge error is the mean value of the knowledge error over a specified time interval. The time intervals are chosen one after the other and are equal to $\Delta_t = 500$ sec.

$$MKE(\Delta_t) = \frac{1}{\Delta_t} \int_{\Delta_t} e_k(t) dt \quad (13)$$

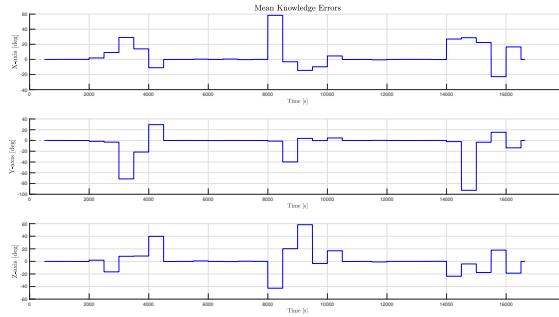


Figure 44: Mean Knowledge Error in SSO Orbit

Mean Performance Error

Mean performance error is the mean value of the performance error over a specified time interval. The time intervals are chosen one after the other and are equal to $\Delta_t = 500$ sec.

$$MPE(\Delta_t) = \frac{1}{\Delta_t} \int_{\Delta_t} e_p(t) dt \quad (14)$$

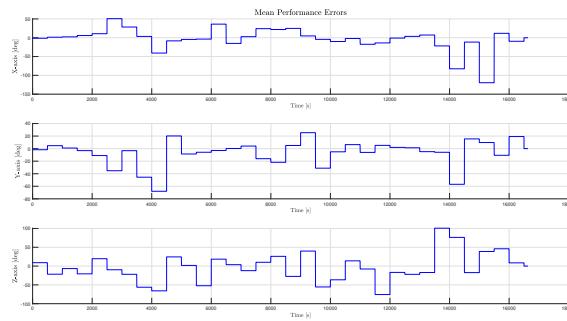


Figure 45: Mean Performance Error in SSO Orbit

Relative Knowledge Error

Relative knowledge error is the difference between the instantaneous knowledge error at a given time, and its mean value over a time interval containing that time. The time intervals chosen are the ones chosen in the mean knowledge error, $\Delta_t = 500$ sec.

$$RKE(t, \Delta_t) = e_k(t) - \frac{1}{\Delta_t} \int_{\Delta_t} e_k(t) dt \quad (15)$$

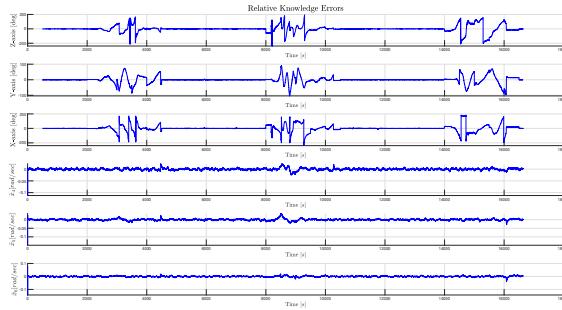


Figure 46: Relative Knowledge Error in SSO Orbit

Relative Performance Error

Relative performance error is the difference between the instantaneous performance error at a given time, and its mean value over a time interval containing that time. The time intervals chosen are the ones chosen in the mean performance error, $\Delta_t = 500$ sec.

$$RPE(t, \Delta_t) = e_p(t) - \frac{1}{\Delta_t} \int_{\Delta_t} e_p(t) dt \quad (16)$$

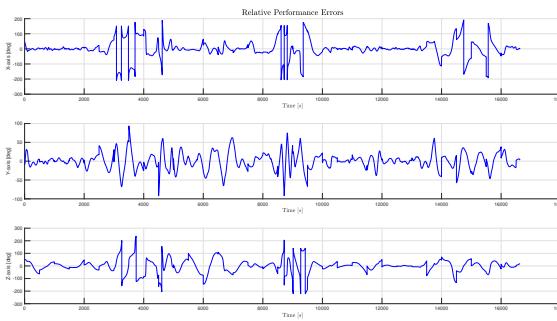


Figure 47: Relative Performance Error in SSO Orbit

Drift Performance Error

Drift performance error is the difference between the means of the performance error taken over two time intervals within a single observation period.

$$DPE(\Delta_{t_1}, \Delta_{t_2}) = \frac{1}{\Delta_{t_1}} \int_{\Delta_{t_1}} e_p(t) dt - \frac{1}{\Delta_{t_2}} \int_{\Delta_{t_2}} e_p(t) dt \quad (17)$$

where $e_p(t)$ is the performance error. For SSO the time periods were selected as $\Delta_{t_1} = \Delta_{t_2} = 250$ sec.

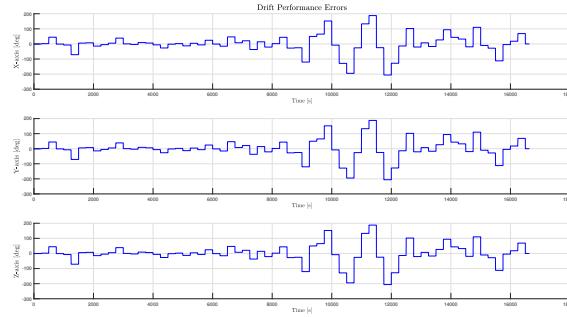


Figure 48: Drift Performance Error in SSO Orbit

We can observe that during eclipse significantly large knowledge errors occur, due to the lack of measurements from the sun sensors. Consequently, the performance errors are skewed for the duration of each eclipse and for a short time period immediately after.

4.5 Geolocation budgets

To acquire a better insight on the geolocation budget and reduce the inaccuracies of the sensor measurements, the total budget is divided into smaller ones, one for each possible sensor error. Thus, a budget will be calculated for each sensor aiming to fulfil the requirements.

Magnetometer detumbling

During detumbling AOCS mode, magnetometer measurements are acquired and utilized by the controller in order to create the desired torque and dissipate the satellite's angular velocity. It is evident that the larger the noise to the magnetometer measurements, the more difficult it is to command the desired torque. Thus, a budget can be calculated regarding the maximum feasible value of the magnetometer noise.

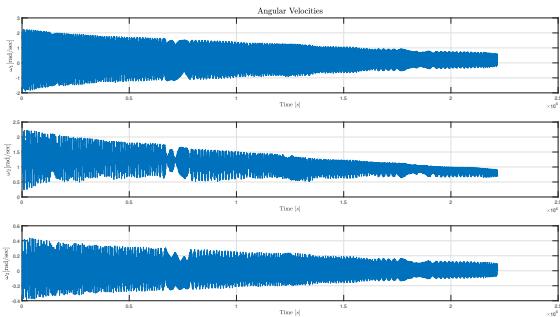


Figure 49: Magnetometer noise budget - Velocities

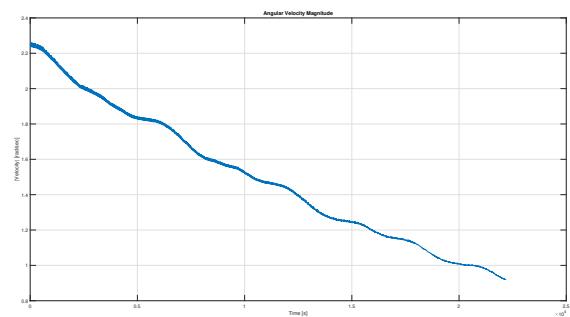


Figure 50: Magnetometer noise budget - Velocities Magnitude

By using the total AOCS simulations, $5.5 \cdot 10^{-6}$ T was found to be the maximum viable noise magnitude of the magnetometer.

Magnetometer nominal

During nominal AOCS mode, magnetometer measurements are acquired and utilized by the MEKF in order to estimate the satellite's attitude. The larger the noise to the magnetometer measurements, the more difficult it is to estimate correctly the satellite's attitude. Thus, a budget can be calculated regarding the maximum feasible value of the magnetometer noise.

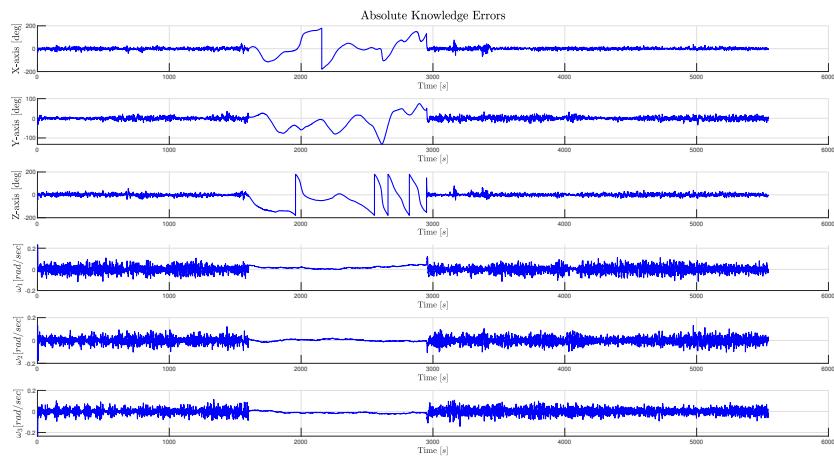


Figure 51: Nominal Magnetometer noise budget - estimation error

Using the total AOCS simulations, 0.42T was found to be the maximum viable noise standard deviation for the magnetometer.

Coarse sun sensor

During nominal AOCS mode, sun sensor measurements are acquired and utilized by the MEKF in order to estimate the satellite's attitude. It is evident that the larger the noise to the sun sensor measurements, the more difficult it is to estimate correctly the satellite's attitude. Thus, a budget can be calculated regarding the maximum feasible value of the sun sensor noise.

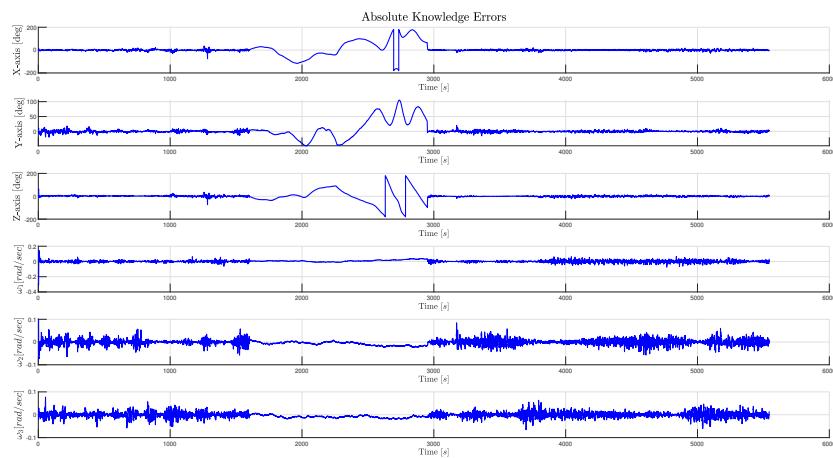


Figure 52: Coarse sun sensor noise budget-estimation error

Sensor noise is proportional to CSS current. The maximum viable constant of proportionality was found to be 0.3.

Gyroscope noise

During nominal AOCS mode, gyroscope measurements are acquired and utilized by the MEKF in order to estimate the satellite's attitude. It is evident that the larger the noise to the gyroscope measurements, the more difficult it is to estimate correctly the satellite's attitude. Thus, a budget can be calculated regarding the maximum feasible value of the gyroscope noise.

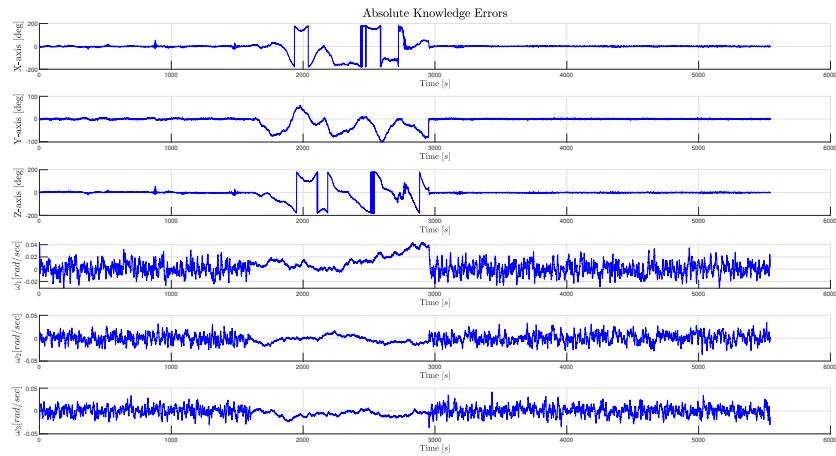


Figure 53: Gyro white Gaussian noise - estimation error

By using the determination simulations, 0.026 rad/sec was found to be a viable noise magnitude for the gyroscope.

Gyroscope bias

During nominal AOCS mode, gyroscope measurements are acquired and utilized by the MEKF in order to estimate the satellite's attitude. It is evident that the larger the bias drift to the gyroscope measurements, the more difficult it is to estimate correctly the satellite's attitude. Thus, a budget can be calculated regarding the maximum feasible value of the gyroscope bias drift.

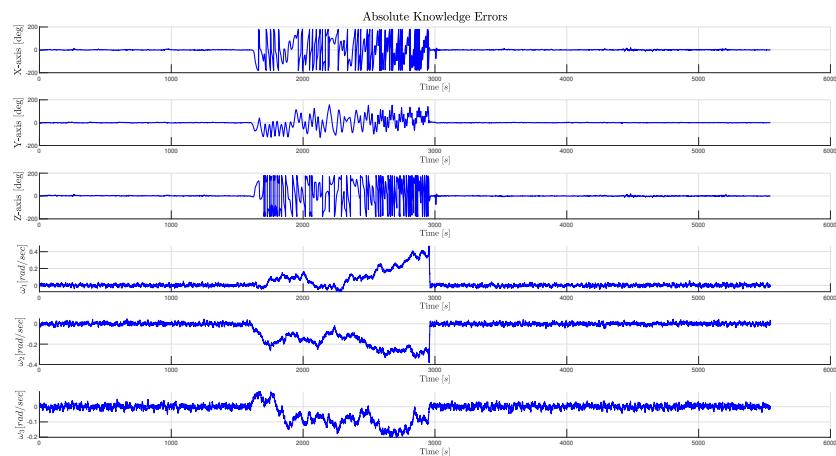


Figure 54: Gyroscope bias budget - estimation error

By using the determination simulations, $7.75 \cdot 10^{-3} \text{ rad/s}$ was found to be a viable bias

drift for the gyroscope.

4.6 Stand-by mode dynamics analysis

In this subsection, a brief analysis of the satellite's dynamics in AOCS stand-by mode is presented. In the following simulation, only disturbance torques have been included, without any form of attitude or angular rate control.

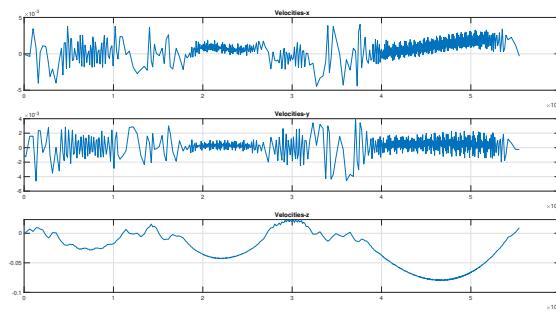


Figure 55: Safe Mode Angular Velocities

As apparent in Fig. 55, the angular velocity can be considered stable for at least 10 orbits.

4.7 Delta-v budget

Delta-v is a measure of the impulse per unit of spacecraft mass that is needed to perform a maneuver. Since there are no orbital maneuvers planned for this mission, the calculation of Delta-v is omitted.



5 AOCS Control and determination algorithms

5.1 Orbit Propagator

AcubeSAT's AOCS is required to determine the satellite's position and velocity autonomously. The SGP4 propagator has been chosen to accomplish this task, due to its relatively high precision, low computational load and widespread application in LEO satellites. The SGP4 propagator is initialized by TLE. In order to use the propagator in the simulations environment, 6 TLE files were created for simulation purposes based on the potential orbit of AcubeSAT. The default SSO TLE is considered to be 600km - 11 PM.

Table XII: SSO TLE, 6 LTAN, 500km

SSO TLE
1 69696U 16025E 23183.00000000 .00002000 00000-0 49111-4 0 6969
2 69696 97.3759 191.5890 0001000 00.0000 000.0000 15.24261762696969

Table XIII: SSO TLE, 9 LTAN, 500km

SSO TLE
1 69696U 16025E 23183.00000000 .00002000 00000-0 49111-4 0 6969
2 69696 97.3759 236.5890 0001000 00.0000 000.0000 15.24261762696969

Table XIV: SSO TLE, 11 LTAN, 500km

SSO TLE
1 69696U 16025E 23183.00000000 .00002000 00000-0 49111-4 0 6969
2 69696 97.3759 266.5890 0001000 00.0000 000.0000 15.24261762696969

Table XV: SSO TLE, 6 LTAN, 600km

SSO TLE
1 69696U 16025E 23183.00000000 .00002000 00000-0 49111-4 0 6969
2 69696 97.7640 191.5890 0001000 00.0000 000.0000 14.89144204204206

Table XVI: SSO TLE, 9 LTAN, 600km

SSO TLE
1 69696U 16025E 23183.00000000 .00002000 00000-0 49111-4 0 6969
2 69696 97.7640 236.5890 0001000 00.0000 000.0000 14.89144204204206

Table XVII: SSO TLE, 11 LTAN, 600km

SSO TLE
1 69696U 16025E 23183.0000000 .00002000 00000-0 49111-4 0 6969
2 69696 97.7640 266.5890 0001000 00.0000 000.0000 14.89144204204206

By using the SGP4 propagator we acquire the following orbit figures.

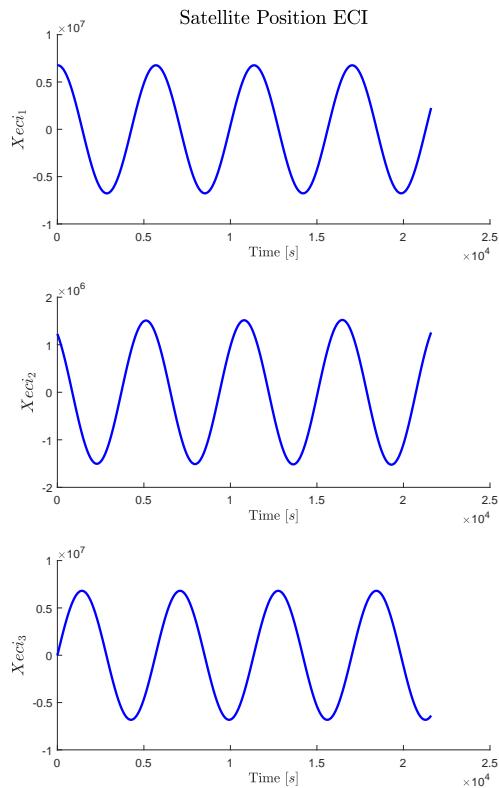


Figure 56: SSO TLE satellite position ECI in 6 hours

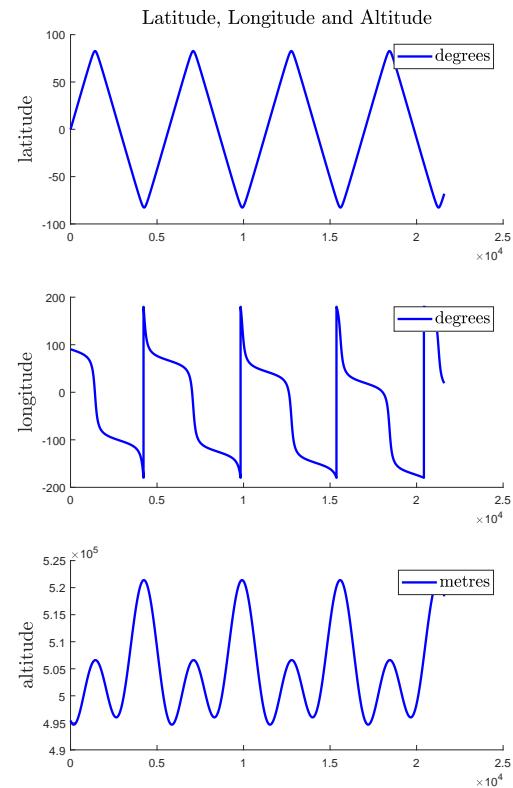


Figure 57: SSO TLE satellite position LLH in 6 hours

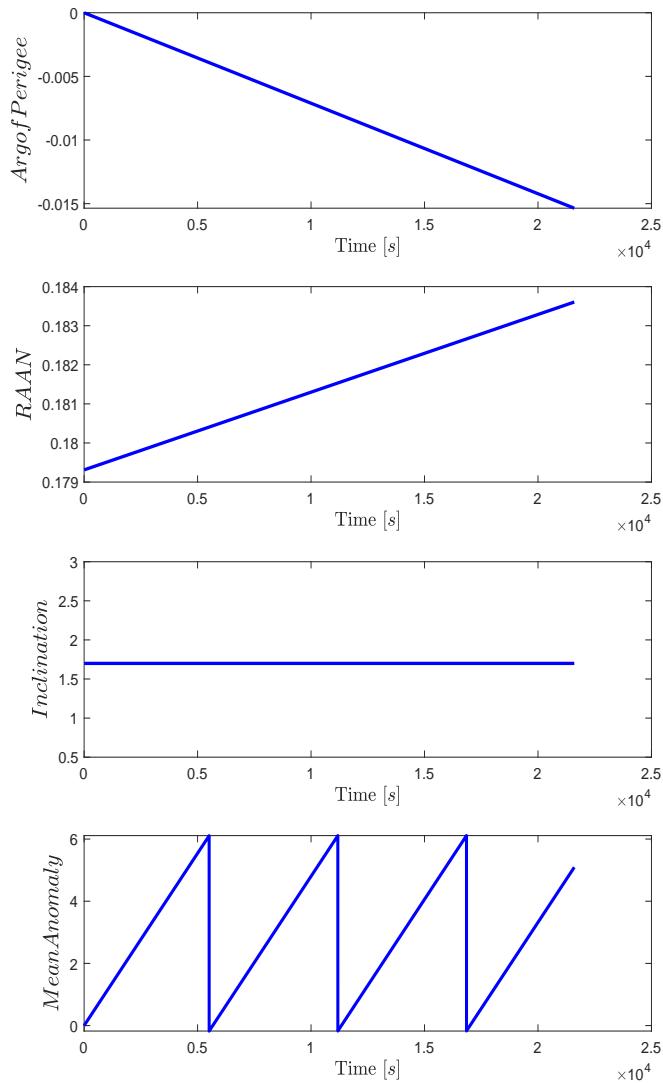


Figure 58: SSO TLE Orbital Parameters in 6 hours

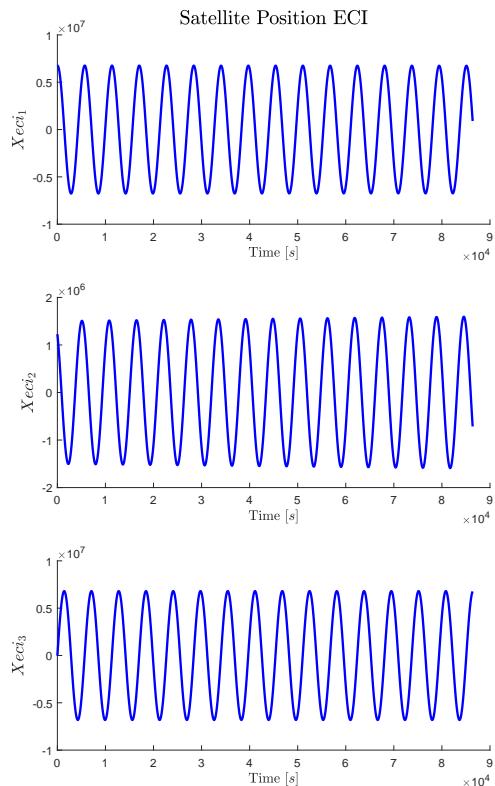


Figure 59: SSO TLE satellite position ECI in 24 hours

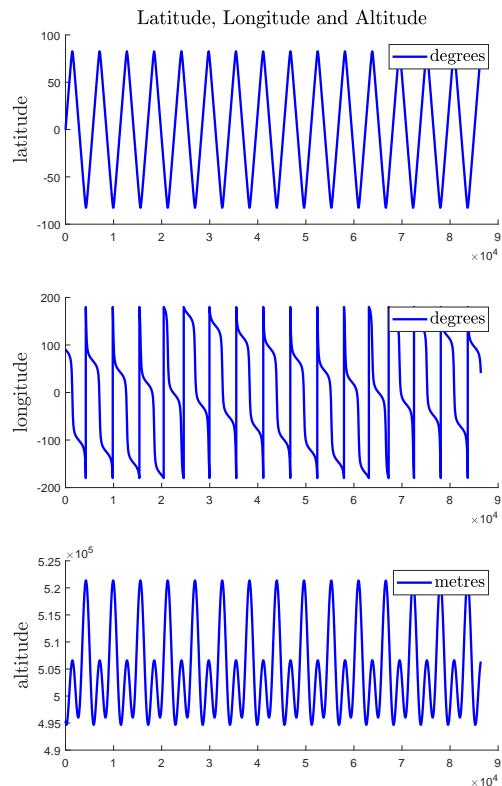


Figure 60: SSO TLE satellite position LLH in 24 hours

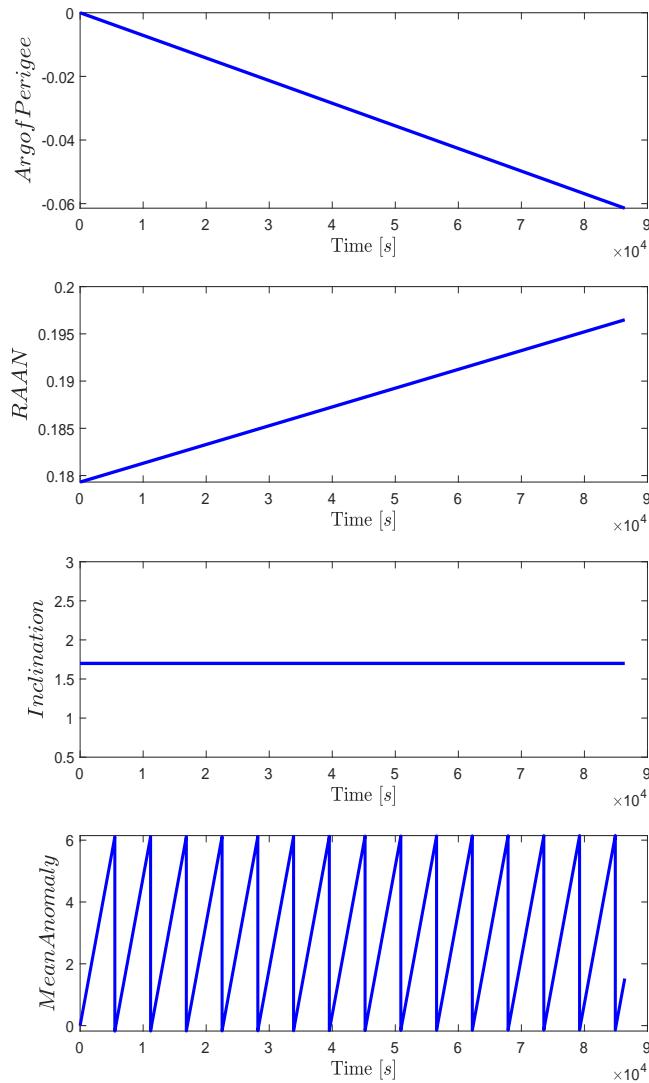


Figure 61: SSO TLE Orbital Parameters in 24 hours

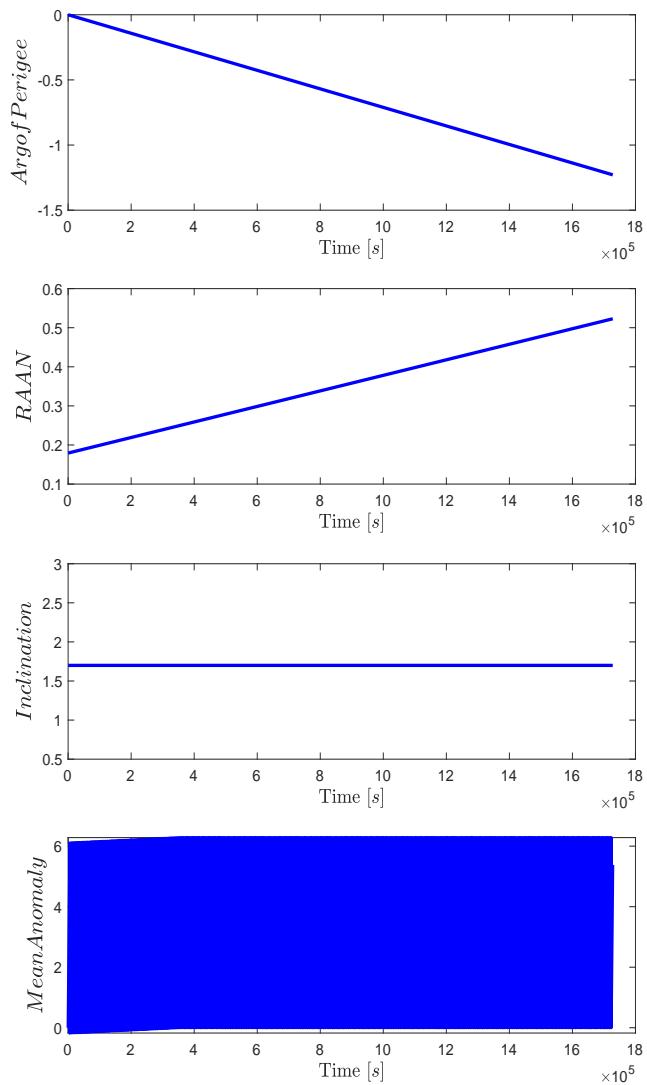


Figure 62: SSO TLE Orbital Parameters in 20 days

Sampling frequency

The orbit propagator will be operated during nominal AOCS mode once every second during the calculations part, as detailed in the control cycle design in [Figure 9](#).

Inputs

The orbit propagator requires the following inputs:

- **Epoch**, for which the propagation will take place.
- **Orbital parameters**, which define the satellite's orbit. These will be initialized based on the TLE file downloaded.

Outputs

The orbit propagator returns the following outputs for the specified epoch:

- **Satellite position**
- **Satellite velocity**
- **Orbital parameters**:
 - Inclination
 - Argument of perigee
 - Right ascension of the ascending node
 - Mean anomaly

5.2 On-board ephemeris

5.2.1 Sun position

AcubeSAT's AOCS is required to determine the sun's position relative to the spacecraft autonomously. The ephemeris modeling as detailed in [\[6\]](#) was used for the AOCS. This modeling is selected due to its relatively high precision and low computational load. Software for the ephemeris modeling can be directly accessed <https://www.celestrak.com/software/vallado-sw.php>.

Analytically, the corresponding algorithm in order to calculate sun's position is:



Algorithm 1: Sun position

- 1 $T_{UT1} = \frac{JD_{UT1} - 2451545.0}{36525};$
 - 2 $\lambda_{M_\odot} = 280.460^\circ + 36000.771 \cdot T_{UT1};$
 - 3 LET $T_{TDB} \cong T_{UT1};$
 - 4 $M_\odot = 357.5291092^\circ + 35999.05034 \cdot T_{TDB};$
 - 5 $\lambda_{ecliptic} = \lambda_{M_\odot} + 1.914666471^\circ \cdot \sin(M_\odot) + 0.019994643 \cdot \sin(2 \cdot M_\odot);$
 - 6 $r_\odot = 1.000140612 - 0.016708617 \cdot \cos(M_\odot) - 0.000139589 \cdot \cos(2 \cdot M_\odot);$
 - 7 $\epsilon = 23.439291^\circ - 0.0130042 \cdot T_{TDB};$
 - 8 $r_\odot^{ECI} = \begin{bmatrix} r_\odot \cdot \cos(\lambda_{ecliptic}) \\ r_\odot \cdot \cos(\epsilon) \cdot \sin(\lambda_{ecliptic}) \\ r_\odot \cdot \sin(\epsilon) \cdot \sin(\lambda_{ecliptic}) \end{bmatrix} AU;$
 - 9 $\alpha_\odot = \arctan(\cos(\epsilon) \cdot \tan(\lambda_{ecliptic}));$
 - 10 $\delta_\odot = \arcsin(\sin(\epsilon) \cdot \sin(\lambda_{ecliptic}));$
-

By utilising the SGP4 propagator described in [Section 5.1](#) for both possible orbits we acquire the following sun ephemeris graphs.

20 days

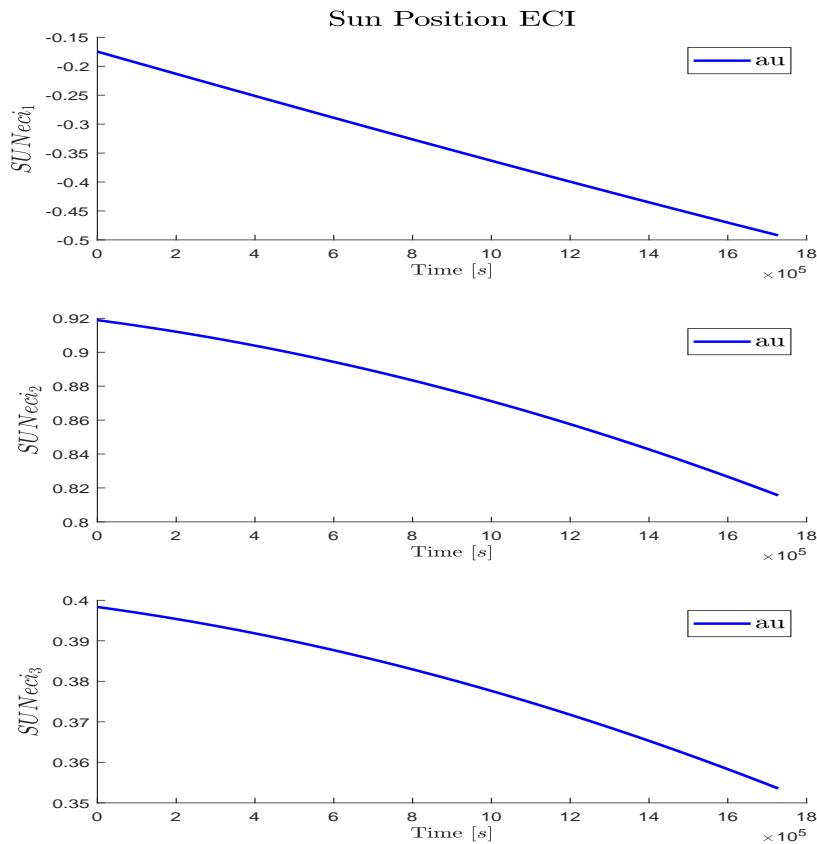


Figure 63: SSO TLE sun position in 20 days

Sampling frequency

The sun position model will be operated during nominal AOCS mode once every 1 second during the calculations part, as detailed in the control cycle design 9.

Inputs

The sun ephemeris requires the following inputs:

- **Time** in JD format for which the calculation will take place.

Outputs

The sun ephemeris returns the following outputs:

- **Sun position:** in the ECI frame for the specified epoch.

5.2.2 Eclipse

While in orbit, the satellite may enter eclipse, this phenomenon has to be calculated in order to define whether or not sun sensor measurements are credible at any moment. The eclipse modeling as developed by Vallado in [6] and [7] was used for the AOCS. This is implemented in the simulations environment and, in the future, may be implemented on-board for further sanity checks to the sensor's measurements.

Analytically the corresponding algorithm in order to calculate eclipse is:

Algorithm 2: Eclipse

```

1  $x_1 = \frac{R_e AU}{R_s + R_e};$ 
2  $\alpha_1 = \pi - \arccos\left(\frac{R_e}{x_1}\right) - \arccos\left(\frac{R_e}{|\mathbf{r}|}\right);$ 
3  $x_2 = \frac{R_e AU}{R_s - R_e};$ 
4  $\alpha_2 = \arccos\left(\frac{R_e}{x_2}\right) - \arccos\left(\frac{R_e}{|\mathbf{r}|}\right);$ 
5  $\alpha = \pi - \arccos\left(\frac{\mathbf{r}_s \cdot \mathbf{r}}{|\mathbf{r}_s| |\mathbf{r}|}\right);$ 
6 if  $\alpha_2 < \alpha < \alpha_1$  then
7    $S = Penumbral;$ 
8 if  $\alpha < \alpha_2$  then
9    $S = Umbral;$ 
10 else
11    $S = No\ Eclipse;$ 

```

By utilising the SGP4 propagator and the ephemeris model as described in [Section 5.1](#) and [Section 5.2](#) for both possible orbits we acquire the following eclipse calculations.

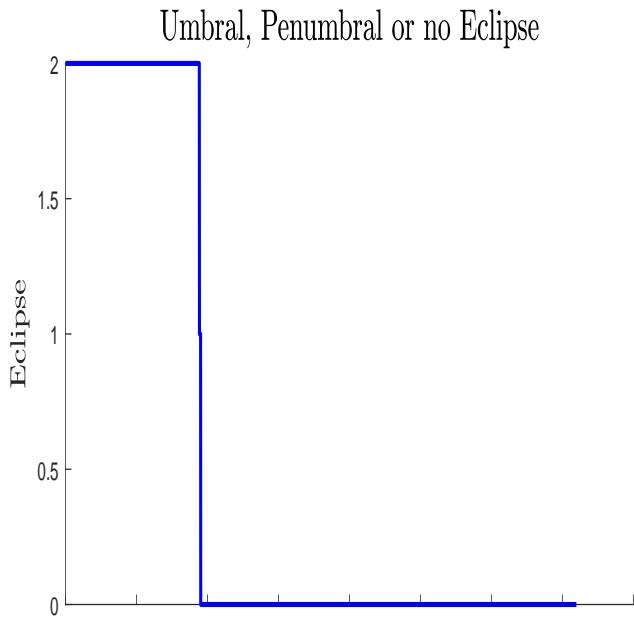


Figure 64: SSO TLE eclipse in 1 hour

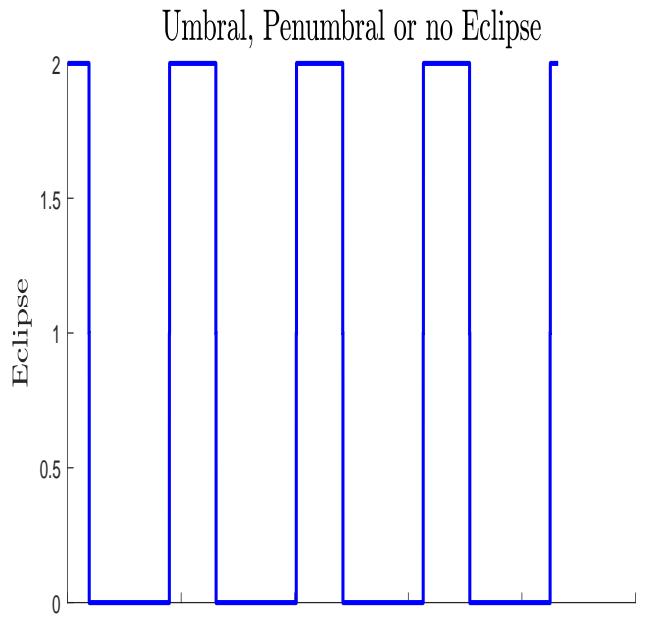


Figure 65: SSO TLE eclipse in 6 hours

Sampling frequency

The eclipse calculation model will be operated during nominal AOCS mode once every 1 second during the calculations part, as detailed in the control cycle design 9.

Inputs

The eclipse model requires the following inputs:

- **Satellite position:** in the ECI frame.
- **Sun position:** in the ECI frame.

Outputs

The eclipse model returns the following outputs:

- **Eclipse:** which defined whether the satellite is in umbral, penumbral or non-eclipse conditions.

5.3 Magnetic field model

AcubeSAT's AOCS is required to determine the magnetic field's vector autonomously. To determine this, the International Geomagnetic Reference Field (IGRF) model is used.

The used IGRF software can be directly accessed [8].

By utilising the SGP4 propagator described in Section 5.1 we acquire the following magnetic field graphs.

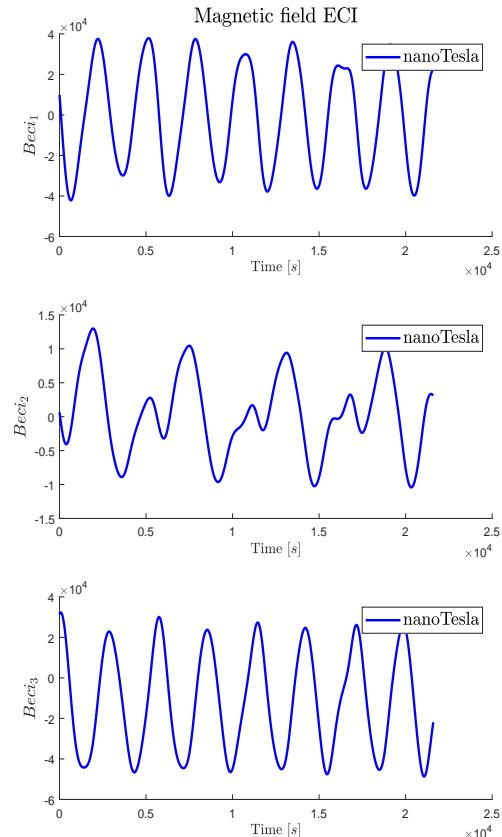
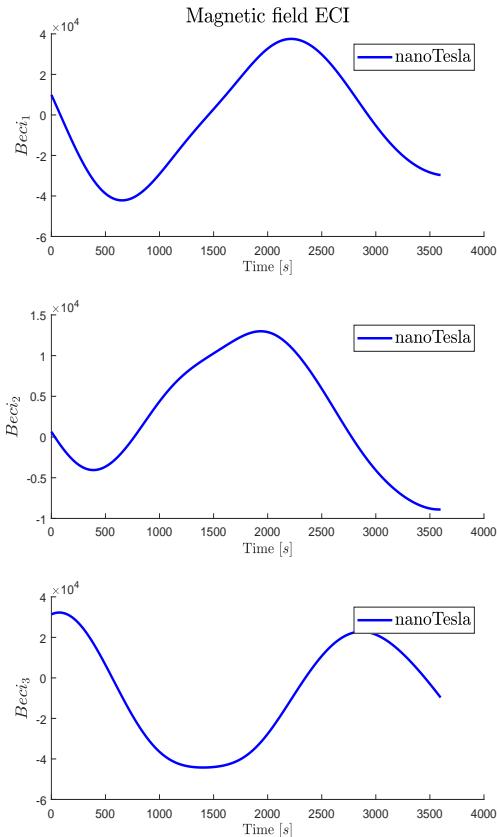


Figure 66: SSO TLE magnetic field in 1 hour

Figure 67: SSO TLE magnetic field in 6 hours

Sampling frequency

The magnetic field model will be operated during nominal AOCS mode once every 1 second during the calculations part, as detailed in the control cycle design 9.

Inputs

The magnetic field requires the following inputs:

- **Time:** for which the calculation will take place.
- **Latitude:** of the satellite for the specified epoch.

- **Longitude:** of the satellite for the specified epoch.
- **Altitude:** of the satellite for the specified epoch.
- **Coordinate system used:** for our integration it is set to 'geodetic'.

Outputs

The magnetic field returns the following outputs:

- **Magnetic field:** in the NED frame for the specified epoch.

5.4 Sensor processing algorithms

The control cycle of each mode requires measurements with a frequency of 10 Hz. However, each sensor has its own internal sampling frequency which is much higher than 10Hz. Thus, an algorithm for the handling of measurements must be chosen. Two options are considered here, the final decision being left for a later stage:

- Measurements are used as-is whenever required by the active mode's control cycle. The rest are discarded.
- Between two consecutive control cycle timesteps, all measurements are kept. The median value of the some measurements closest to the next timestep is calculated and fed to the active controller as the measurement for the next timestep. This calculation filters out random spikes in sensor measurements.

5.4.1 Coarse sun sensor filtering

The CSS output consists of photodiode currents on each side of the satellite. Currents on opposite sides are added signed (+/-) and the sums are then used to create the sun position vector. The vector is then normalized.

5.4.2 Wahba's problem

MEKF requires an initial state estimation in order to start attitude determination. The state vector is defined as follows:

$$\hat{x} = (\mathbf{q}_{1:4}, \boldsymbol{\omega}, \boldsymbol{\beta}) \quad (18)$$

where $\mathbf{q}_{1:4}$ is the ECI to body quaternion, $\boldsymbol{\omega}$ is the angular velocity of the body frame relative to the ECI frame, expressed in the body frame, and $\boldsymbol{\beta}$ is the bias.

A random initial estimation would be inefficient and risky. Thus, a better initial estimate is acquired by solving Wahba's problem, which is stated as such:

Find the orthogonal matrix A with determinant +1 that minimizes the loss function

$$L(A) = \frac{1}{2} \sum_{v_i=1}^N a_i \| \mathbf{b}_i - A \mathbf{r}_i \|^2 \quad (19)$$

where \mathbf{b}_i is a set of N unit vectors measured in a spacecraft's body frame, \mathbf{r}_i are the corresponding unit vectors in a reference frame, and a_i are non-negative weights. In this specific case, the unit vectors are the sun and magnetic field unit vectors, in the body and ECI frames respectively. Both weights are equal to 1. A is the attitude matrix.

To solve Wahba's problem, the singular value decomposition (SVD) algorithm was selected. The entire procedure is presented below:

Algorithm 3: Attitude estimation with SVD

Result: Attitude matrix

```

1  $B = a_1 \cdot \mathbf{sun}_{\text{body}} \cdot \mathbf{sun}_{\text{eci}} + a_2 \cdot \mathbf{mgn}_{\text{body}} \cdot \mathbf{mgn}_{\text{eci}}$  ;
2  $[U, S, V] = \text{svd}(B)$  ;
3  $d = \det(U) \cdot \det(V)$  ;
4 if  $d < 0$  then
5   for  $i = 1 : 3$  do
6     |  $U(i, 3) = -U(i, 3)$ ;
7   end
8  $A = U \cdot V^T$ ;
```

5.4.3 Initial bias estimation

To increase the robustness of the bias estimation, an initial bias estimation is implemented which is fed to the MEKF. In order to calculate this, Wahba's problem is solved in n successive time-steps. By combining these, we can calculate $n - 1$ angular rates $\omega_{k-1,k}$ with $k = 2, 3, \dots, n$ using the following equation:

$$\omega(t) = 2 \frac{d\mathbf{q}(t)}{dt} \mathbf{q}^{-1}(t) \quad (20)$$

where \mathbf{q} is the attitude quaternion. This is a continuous-time equation, however it can be discretized as such:

$$\omega_{k-1,k} = 2 \frac{\Delta \mathbf{q}}{\Delta t} \mathbf{q}_k^{-1} \quad (21)$$

The quaternion derivative is calculated numerically on each time-step. The angular rates $\omega_{k-1,k}$ are then time-averaged in order to obtain a more robust estimate:

$$\omega_{\text{mean}} = \frac{\sum_{k=2}^n \omega_{k-1,k}}{n - 1} \quad (22)$$

Finally the bias is calculated:

$$\hat{\beta}_{\text{initial}} = \omega_{\text{mean}} - \omega_{\text{measured}} \quad (23)$$

5.4.4 Multiplicative Extended Kalman Filter

For the online estimation of attitude and angular velocity, the Multiplicative Extended Kalman Filter (MEKF) has been selected, it is a modification of the standard Extended Kalman Filter. Instead of directly estimating the global attitude quaternion, it estimates the error between the real and estimated global attitude quaternions, and updates the latter accordingly. Two functions are needed for its operation:

- **State Function:** The state function calculates the next global state, consisting of the attitude quaternion and the gyroscope bias. Angular velocity is not propagated, as the satellite's dynamic model accuracy cannot be accurately characterized. Instead, the ground-truth bias-corrected gyro measurement is used (26). The kinematic model equations used are mentioned in Paragraph 6.2.4.2.
- **Measurement Function:** The measurement function calculates measurements based on the estimated state. Magnetometer and sun sensor measurements are implemented as transformations of the reference vectors from ECI to body frame, which is calculated based on the aforementioned estimated state. More specifically:

$$\hat{b}_{meas} = \hat{q}^{-1} \otimes b \otimes \hat{q} \quad (24)$$

$$(25)$$

where b is the magnetic induction in ECI, \hat{q} is the estimated ECI to body attitude quaternion, \hat{b}_{meas} is the estimated magnetic induction in the body frame. The expected sun sensor measurement is acquired by transforming the ECI vector to body frame, and then applying the procedure mentioned paragraph 6.2.5.3, without including random noise.

MEKF uses of the following variables:

- **Covariance Matrix $[P]$:** The covariance matrix expresses the correlation of each state element w.r.t every state element.
- **Q & R Matrices:** The Q and R matrices express the uncertainties in the kinematic/dynamic models and sensor measurements respectively.
- **Jacobian:** The state transition and measurement functions' Jacobian matrices are calculated analytically as per the following:

$$F(\hat{\mathbf{x}}(t), t) = \begin{bmatrix} -[\hat{\omega}(t) \times] & -I_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}$$

$$H_k(\hat{\mathbf{x}}_k^-) = \begin{bmatrix} [A(\hat{q}^-) \mathbf{r}_1 \times] & 0_{3 \times 3} \\ \vdots & \vdots \\ [A(\hat{q}^-) \mathbf{r}_n \times] & 0_{3 \times 3} \end{bmatrix}$$

MEKF requires an initial state estimation before starting its operation. After initialization, it works in a constant cycle, outlined below. $f(x, u)$ is the transition function and F_k is its Jacobian, while in $h(x)$ is the measurement function and H_k its Jacobian.

Algorithm 4: Multiplicative Extended Kalman Filter Predict

```

1 while True do
2    $\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, u_k)$  ;           // Predict the next state
3    $A = \begin{bmatrix} -F_k & Q_{6 \times 6} \\ 0_{6 \times 6} & F_k^T \end{bmatrix}$ 
4    $B = e^{A \cdot dt}$ 
5    $\Phi_k = B_{7:12, 7:12}$ 
6    $Q_s = \Phi_k \cdot B_{1:6, 7:12}$ 
7    $P_{k+1|k} = \Phi_k P_{k|k} \Phi_k^T + Q_s$  ;       // Predict the next error covariance

```

Algorithm 5: Multiplicative Extended Kalman Filter Correct

```

1 while True do
2    $K_k = P_{k|k-1} H_k (H_k P_{k|k-1} H_k^T + R_k)^{-1}$  ;      // Compute the Kalman gain
3    $\epsilon_k = K_k [z_k - h_k(\hat{x}_{k|k-1})]$  ;                  // Calculate error state
4    $q_{error} = \begin{bmatrix} 1 \\ 0.5 \cdot \epsilon_k(1) \\ 0.5 \cdot \epsilon_k(2) \\ 0.5 \cdot \epsilon_k(3) \end{bmatrix}$  ;          // Calculate error quaternion assuming that
                                         // it is close to identity
5    $q_{k|k} = q_{k|k-1} \otimes q_{error}$  ;        // Update global quaternion with error state
6    $b_{k|k} = b_{k|k-1} + \epsilon_k(4 : 6)$  ;        // Update bias with error state
7    $q_{k|k} = q_{k|k} / \text{norm}(q_{k|k})$  ;        // Normalize quaternion
8    $P_{k|k} = (I - K_k H_k) P_{k|k-1}$  ;        // Correct error covariance

```

5.4.4.1 Bias model in MEKF

Bias is assumed to be constant during the predict phase of MEKF:

$$\hat{\beta} = 0 \quad (26)$$

The measurement function is modified to include bias when calculating expected measurements. Bias estimation is automatically updated on the correct phase of MEKF, based on the measurement residual.

$$\tilde{y}_k = z_k - h(\hat{x}_{k|k-1}) \quad (27)$$

MEKF seeks to minimize the difference between the measurement function (which includes bias) and the groundtruth measurements. When these match up, bias has been accurately estimated, thus the above calculation result is sufficient.

5.4.4.2 Albedo compensation

Provided that the Albedo model (paragraph 6.2.3.5) is sufficiently accurate, Albedo calculations are integrated inside the MEKF correct step.

5.4.5 Sampling frequency

The sensor processing algorithms are operated during nominal AOCS mode once every 0.1 second during the determination part, as detailed in the control cycle design (Figure 9).

5.4.6 Inputs

The sensor processing algorithms require the following inputs:

- Magnetometer measurements
- Gyroscope measurements
- Sun sensor measurements

5.4.7 Outputs

The sensor processing algorithms return the following outputs:

- Quaternion: ECI to body.
- Angular velocity: of the body frame with respect to the ECI frame, expressed in the satellite body frame.

5.5 Estimation, guidance and control algorithms

Detumbling mode

During AOCS detumbling mode, the angular velocity of the body frame relatively to the ECI frame, expressed in the satellite body frame, is dissipated. Since attitude determination, and specifically MEKF, is disabled during this mode, angular velocity cannot be estimated directly. In order to solve this issue, the derivative of the magnetic field, \dot{b} , is used. This metric is considered to be a sufficiently proportional to the angular velocity. The controller used is calculated as:

$$m_i = -K \cdot \dot{b}_i \quad (28)$$

where i denotes as the index of each of the three axes and m is the desired magnetic moment applied to the magnetorquers, as detailed in Section 5.6.

On-board magnetic field measurements are acquired through the magnetometers, which subsequently are used to calculate \dot{b} . It is common enough that single sensor measure-



ments are faulty, thus some kind of measurement filtering is considered, as detailed in section [Section 5.4](#).

Nominal mode

During AOCS nominal mode, the satellite is performing **nadir pointing**. Initially, the SGP4 propagator is utilized to estimate the satellite position. These estimations are used by the on-board ephemeris and IGRF models to calculate the sun position and magnetic field vectors respectively.

In order to transition from AOCS detumbling to nominal mode, an initial state estimation is required. This is implemented by solving Wahba's problem for sensor fusion, which, by acquiring magnetic field, sun position values and magnetometer and sun sensor measurements, can estimate the state quaternion. In addition to this, by acquiring consecutive quaternion estimations, we can estimate the state angular velocity. Finally, by combining this estimation with the gyroscope measurements, we can estimate the state bias, thus constructing the required initial state estimation for MEKF, entering nominal AOCS mode.

In nominal mode, the MEKF is operating normally both by correcting and predicting the state estimation. During attitude determination, magnetometer, sun sensor and gyroscope measurements are acquired. These measurements are used by the MEKF to improve its state estimation. Simultaneously, MEKF predicts the next state estimation, with a frequency of 10 Hz, which is used throughout the entirety of the control cycle. During actuation, based on the predicted state estimation, the desired control torque is calculated by a PD controller, with a frequency of 10 Hz. The commanded torque is subsequently applied to the actuators, achieving nadir pointing.

The PD controller, as defined in [Section 5.6](#), requires the quaternion from the orbit to the satellite body frames and the angular velocity of the body frame with respect to the orbit frame expressed in the body frame. In order to acquire these parameters the ECI to body quaternion is rotated with the Orbit to ECI quaternion, as produced by the corresponding rotation matrix:

$$q_{ob} = q_{oi} \otimes q_{ib} \quad (29)$$

where ob , oi , ib depict the orbit to body, orbit to ECI, ECI to body frame rotations. and the required angular velocity is calculated as:

$$\omega_{b,ob} = \omega_{b,ib} - \omega_{b,io} \quad (30)$$

where, for instance, $\omega_{\delta,\epsilon\zeta}$ depicts the angular velocity of the ϵ frame with respect to the

ζ frame, expressed in the δ frame. $\omega_{b,io}$ is $\begin{bmatrix} 0 \\ w_o \\ 0 \end{bmatrix}$.

5.6 Actuator mapping and command processing algorithms

Controllers

Our control design includes two controllers, one B-dot and one PD controller. The first one is used during the AOCS Detumbling Mode to dissipate the kinetic energy of the satellite and the second one during the AOCS Nominal Mode to perform position tracking.

B-dot Controller

B-dot control uses only magnetorquers, as actuators, and only magnetometer, as sensor. The controller is used to create, through magnetorquers, a magnetic dipole in the opposite direction to the derivative of the magnetic field vector \mathbf{b} , estimated with magnetometers data:

$$m_i = -K \cdot \dot{b}_i \quad (31)$$

where i denotes as the index of each of the three axes and \dot{b}_i the temporal derivative of the geomagnetic field (\mathbf{b}) with respect to this axis.

In order to calculate the $\dot{\mathbf{b}}$, we subtract the temporal \mathbf{b}_2 estimation with the previous \mathbf{b}_1 estimation, given by the magnetometer, and the result is divided by the time-period elapsed between those two estimations, which is equal to 0.1 sec.

$$\dot{\mathbf{b}} = \frac{\mathbf{b}_2 - \mathbf{b}_1}{0.1} \quad (32)$$

Afterwards, the magnetic dipole \mathbf{m} is scaled through the [algorithm 6](#).

Algorithm 6: Magnetic dipole scaling

Result: Final dipole given

```

1 for i = 1:3 do
2   if m_i > m_max then
3     |   m_i = m_max;
4   else if m_i < -m_max then
5     |   m_i = -m_max;
6   else
7     |   m_i = m_initial
8 end

```

Finally, the torque given to be produced by the magnetorquers is calculated using the equation

$$\tau_m = \mathbf{m} \times \mathbf{b} \quad (33)$$

Requirements close-out — Analysis

The above document is the close out reference for requirement:

Requirement ID	Requirement Text	Status & Rationale	Reference
ADCS-FUN-180	The de-tumbling shall be performed using only magnetic control without ground contact.	Compliant in Phase C. Theoretical analysis and simulations both use magnetic actuation solely	Section 5.6, Section 6.4

PD Controller

PD control uses both magnetorquers and reaction wheel as actuators and all sensors. The goal is to drive the actual quaternion, which depicts our current orientation, to the commanded and constant quaternion denoted by q_c , as well as stabilizing the satellite upon reaching the desired orientation. The error quaternion is given by

$$\delta q = \begin{bmatrix} \delta q_1 \\ \delta q_{2:4} \end{bmatrix} = q_c^{-1} \otimes q \quad (34)$$

The total $T_{\text{commanded}}$ to be given through our actuators is calculated using the equation below:

$$T_{\text{commanded}} = -\text{sign}(\delta q_1)K_p \delta q_{2:4} - k_d \omega \quad (35)$$

where k_p and k_d are gains matrices and ω is the angular velocity from the satellite body frame to the ECI frame expressed on the body frame.

Sun Pointing PD

Sun pointing PD control utilizes both the actuators. The goal is to drive the -X and +Y sides of the AcubeSAT to face the sun with a 45° angle, as well as stabilizing the satellite upon reaching the desired orientation. Similarly to Eq. (34), the calculation of the error quaternion is described as:

$$\delta q = \begin{bmatrix} q_1 \\ q_{2:4} \end{bmatrix} = \begin{bmatrix} \hat{S} \cdot \hat{P} \\ \hat{S} \times \hat{P} \end{bmatrix} \quad (36)$$

where \hat{S} is the sun vector in body frame and \hat{P} is the unit vector from body frame origin to the center of the edge between -X and +Y sides of the AcubeSAT. The error quaternion is taken from the above quaternion when normalized. In case the angle between \hat{S} and \hat{P} is 180° [Eq. (36)] is omitted and the error quaternion becomes $\begin{bmatrix} 0 & 1 & 1 & 1 \end{bmatrix}^T$. The total $T_{\text{commanded}}$ to be given through our actuators is calculated using the equation below:

$$T_{\text{commanded}} = -\text{sign}(\delta q_1)K_p \delta q_{2:4} - k_d \omega \quad (37)$$

where k_p and k_d are gains matrices and ω is the angular velocity from the satellite body frame to the ECI frame expressed on the body frame.

The parameters utilized by the controllers are shown in [Table XXI](#) in Control Parameters. Due to using a quaternion for attitude orientation representation, ranges between -180° and $+180^\circ$ are fulfilled.

Requirements close-out — Review of Design	
The above text is the close out reference for requirement:	
ADCS-FUN-110	The ADCS shall fulfill the attitude range between -180 to 180 deg on the X, Y and Z axes.

Sampling frequency

The B-dot controller will be operated during detumbling AOCS mode once every 0.1 sec during the controller B-dot part, as specified in the control cycle design [[Fig. 8](#)]. The PD controller will be operated during nominal AOCS mode once every 0.1 sec during the controller PD part, as specified in the control cycle design [[Fig. 9](#)].

Inputs

The controllers require the following inputs:

- **Orbit to body quaternion**
- **Angular velocity of the satellite:** between orbit frame and body frame, expressed in the Body frame.
- **Magnetic field vector:** measured by the magnetometers, expressed in the body frame

Outputs

The controllers return the following outputs:

- **Torques:** to be produced either only by the magnetorquers, when the B-dot controller is enabled, or by the magnetorquers and the reaction wheel, when the PD controller is enabled.

Control Torque allocation

The satellite is equipped with three magnetorquers, each one controlling an axis of the body frame, one reaction wheel and a control system that provides the desired control torque, \mathbf{u} , to be applied to the spacecraft. Since we have also included one reaction wheel in our AOCS design, placed on the Z-axis of the spacecraft, geometric decomposition is

used to distribute the control torque between the magnetorquer and reaction wheel for the Z satellite axis.

The control vector, \mathbf{u} , is decomposed into orthogonal and parallel components with respect to the orientation of the instantaneous magnetic field vector expressed in the satellite body frame:

$$\mathbf{u} = \mathbf{u}_{||} + \mathbf{u}_{\perp} \quad (38)$$

From (38) follows that

$$\tau_m = \mathbf{u}_{\perp} \text{ and} \quad (39)$$

$$\tau_w = \mathbf{u}_{||} \quad (40)$$

The control torque $\mathbf{u}_{||}$ cannot be applied to the system through the magnetorquers, since they produce torques that are perpendicular to the geomagnetic field vector.

The general control signal, the Earth's magnetic field vector, the magnetic torques and the geometrically limited wheel torques to be applied to the spacecraft are shown in Figure 68. The wheels apply torque about some general direction \mathbf{a} . In our case, \mathbf{a} is a fixed unit vector coinciding with the single wheel spin axis.

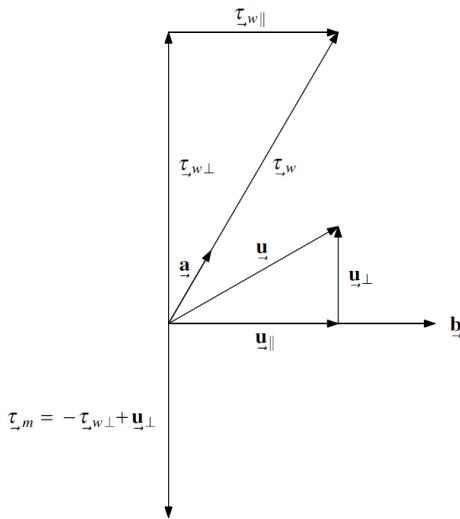


Figure 68: Wheel Torques [9]

Since one reaction wheel is included in our design, only the $\tau_{w||}$ can be applied to the system through the reaction wheel.

Letting $\tau_m = -\tau_{w\perp} + \mathbf{u}_{\perp}$, as shown in Figure 68, and adding τ_w creates the desired control torque:

$$\mathbf{u} = \tau_m + \tau_w = -\tau_{w\perp} + \mathbf{u}_{\perp} + \tau_w = \mathbf{u}_{\perp} + \tau_{w||} = \mathbf{u}_{\perp} + \mathbf{u}_{||} \quad (41)$$

where $\mathbf{u}_{||} = \tau_{||}$.

To calculate the τ_w , we use the relation:

$$\mathbf{u}_{||} = \tau_w \iff \hat{\mathbf{b}}^T \hat{\mathbf{b}} \cdot \mathbf{u} = \hat{\mathbf{b}}^T \hat{\mathbf{b}} \cdot \tau_w \quad (42)$$

where \mathbf{b} is the magnetic field. [Equation \(42\)](#) depends on the number of wheels present. Since there is only one reaction wheel rotating around the Z axis of the satellite body frame, it follows that:

$$\boldsymbol{\tau}_w = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \cdot \boldsymbol{\tau}_z \quad (43)$$

Using [Equation \(42\)](#), we get:

$$\boldsymbol{\tau}_z = \frac{\mathbf{b}^T \mathbf{u}}{b_z} \iff \boldsymbol{\tau}_w = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \frac{\mathbf{b}^T \mathbf{u}}{b_z} \quad (44)$$

In case \mathbf{b} lies in the 112 plane, namely $b_z = 0$, the ability to apply the parallel component of the control vector is lost, thus the system becomes underactuated. However, this underactuation in a particular direction is expected to be instantaneous and therefore not considered a major issue. Having found $\boldsymbol{\tau}_w$, we calculate $\boldsymbol{\tau}_m$ using:

$$\boldsymbol{\tau}_m = -\boldsymbol{\tau}_{w\perp} + \mathbf{u}_\perp = skew(\hat{\mathbf{b}})^T \cdot skew(\hat{\mathbf{b}}) \cdot (-\boldsymbol{\tau}_w + \mathbf{u}) \quad (45)$$

Extending our approach in the control torque allocation, we can write

$$\mathbf{u} = k_{ma}\hat{\boldsymbol{\tau}}_m + k_{wa}\hat{\boldsymbol{\tau}}_w \quad (46)$$

where

$$k_{ma} = \frac{(\hat{\boldsymbol{\tau}}_m^T - (\hat{\boldsymbol{\tau}}_w^T \hat{\boldsymbol{\tau}}_m) \hat{\boldsymbol{\tau}}_w^T) \mathbf{u}}{1 - (\hat{\boldsymbol{\tau}}_w^T \hat{\boldsymbol{\tau}}_m)^2} \quad \text{and} \quad k_{wa} = \frac{(\hat{\boldsymbol{\tau}}_w^T - (\hat{\boldsymbol{\tau}}_m^T \hat{\boldsymbol{\tau}}_w) \hat{\boldsymbol{\tau}}_m^T) \mathbf{u}}{1 - (\hat{\boldsymbol{\tau}}_m^T \hat{\boldsymbol{\tau}}_w)^2} \quad (47)$$

are the magnetic gain and reaction wheel gain, respectively, when the spacecraft is equipped with one wheel and three magnetorquers, and $\hat{\boldsymbol{\tau}}_m$ and $\hat{\boldsymbol{\tau}}_w$ are the unit vectors associated with the magnetorquers and reaction wheel. In the general state of no saturation of the magnetorquers and the Reaction Wheel, we get

$$\mathbf{u} = \boldsymbol{\tau}_m + \boldsymbol{\tau}_w \quad (48)$$

where

$$\boldsymbol{\tau}_m = k_{ma}\hat{\boldsymbol{\tau}}_m \quad \text{and} \quad \boldsymbol{\tau}_w = k_{wa}\hat{\boldsymbol{\tau}}_w \quad (49)$$

which is equivalent with [Equation \(41\)](#).

5.7 Control action discretisation

The control cycle of AcubeSAT during both Nominal and Detumbling Mode is divided into 10 slots of 0.1 sec, in total 1 sec. The usage of each slot depends on the AOCS mode of the satellite.

- **Detumbling Mode:** The first time slot is left empty. During the next 2 time slots the magnetometer measurements are acquired. In the last 7 time slots, the magnetorquers apply the torque, as calculated by the B-dot controller.

- **Nominal Mode:** During the first time slot, the SGP4 propagator is used in conjunction with magnetic field calculations, sun position, eclipse and FDIR checks. During the subsequent 2 time slots, measurements are acquired and attitude determination is carried out. In the last 7 time slots, the actuators apply the torque, as calculated from the PD controller in the beginning of each slot.

In the following simulations, the effect of a different duration for the control cycle in both AOCS modes is studied. The number of time slots and the procedures taking place in each of them remain unchanged in every simulation. Hence, the time duration of the time slots in each case will be 10 times shorter than the selected duration of the control cycle, while the sampling frequency for the procedures executed will be the control cycle frequency times 10.

Control cycle of 2 seconds

At first, a longer cycle duration of 2 sec is selected so the cycle frequency becomes 0.5 Hz and the following results are derived.

Detumbling mode

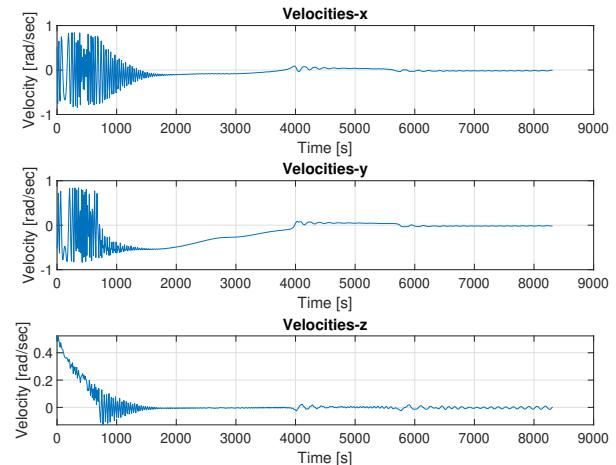
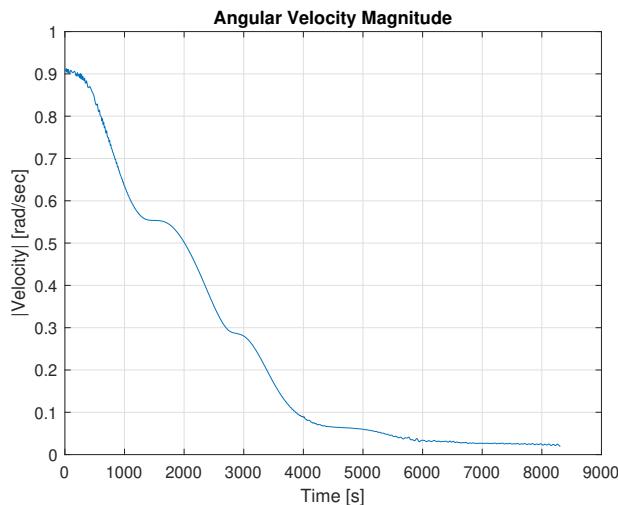


Figure 69: Velocity magnitude - Detumbling cycle duration 2 sec

Figure 70: Velocities - Detumbling cycle duration 2 sec

The updated control cycle design does not seem to affect the controller detumbling capabilities, hence the performance is preserved.

Nominal mode in SSO

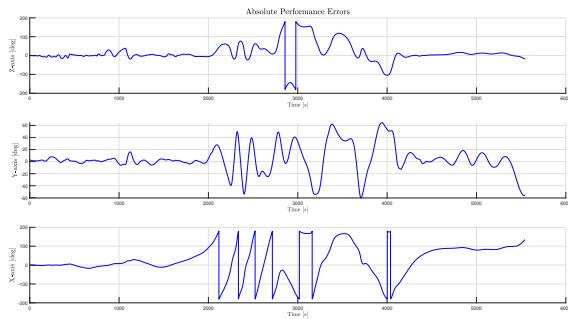


Figure 71: APE - Nominal cycle duration 2 sec

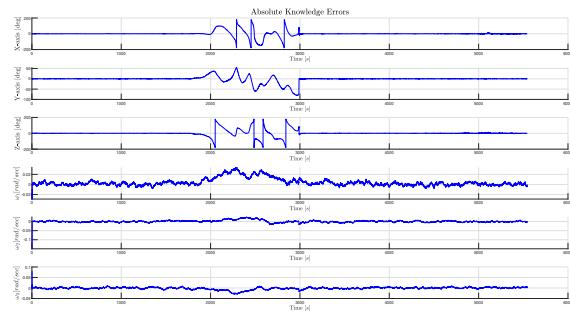


Figure 72: AKE - Nominal cycle duration 2 sec

In the above simulations Section 5.7 we can observe some errors that essentially skew the results.

The updated control cycle design produces considerably larger performance errors.

Control cycle of 0.5 seconds

Next a shorter cycle is selected of 0.5 sec is selected so the cycle frequency becomes 2 Hz and the following results are derived.

Detumbling mode

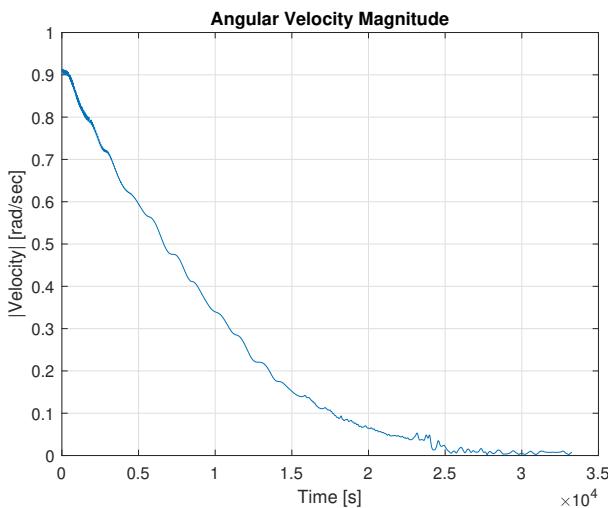


Figure 73: Velocity magnitude - Detumbling cycle duration 0.5 sec

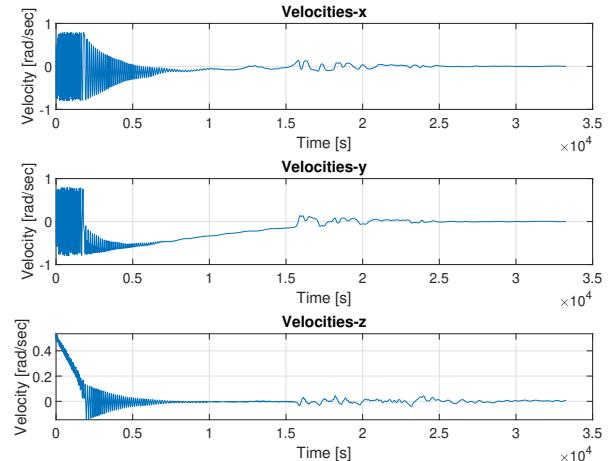


Figure 74: Velocities - Detumbling cycle duration 0.5 sec

The updated control cycle design does not seem to affect the controller detumbling capabilities, hence the performance is preserved.

Nominal mode in SSO

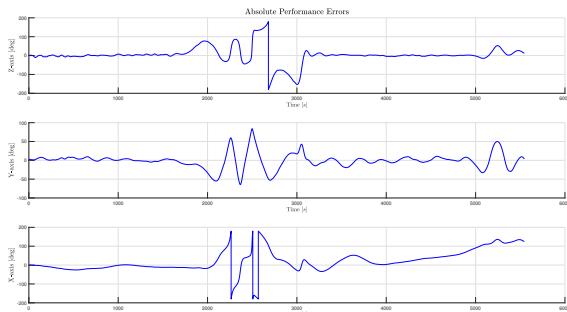


Figure 75: APE - Nominal cycle duration 0.5 sec

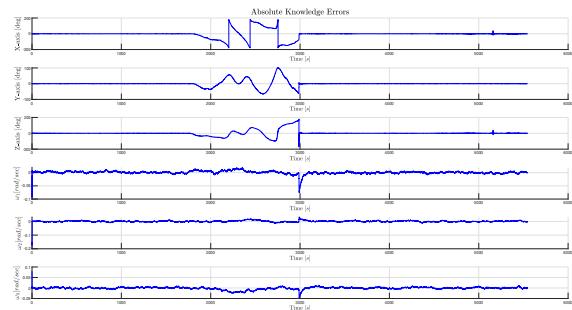


Figure 76: AKE - Nominal cycle duration 0.5 sec

The updated control cycle design appears to enhance the robustness of the system.

Conclusion

It is evident that the duration of the control cycle for AOCS Detumbling mode does not affect the simulation results, as long as the number of timesteps and the procedures that take place in each of them remain unchanged. On the other hand, in AOCS Nominal mode the performance errors seem to improve for a smaller cycle duration. However, a time interval is required for the magnetorquers to discharge, so that magnetometer measurements remain unaffected. Therefore, our cycle duration is selected to be 1 sec.

6 Simulations

6.1 Introduction

For the purpose of presenting the most detailed and precise results possible, 1 distinct simulation is being used:

Total AOCS Simulation

With reference to the Nominal Simulation, the orbit propagator, IGRF and sun position models are utilized, in order to produce and combine sensor measurements, which are in turn used by MEKF. The results are fed to a PD controller to control the satellite.

Also, regarding the detumbling simulation, magnetometer measurements are acquired and utilized by the B-dot controller to calculate the desired torque. The torque is then commanded to the magnetorquers to dissipate the satellite angular velocity.

Results in this total simulation seem correct and sufficient. One issue has emerged from this simulation:

- **Significant error during eclipse:** During eclipse, due to not acquiring reliable coarse sun sensor data, the knowledge error is greatly increased. As a result, pointing during eclipse significantly deteriorates.

Parameter changes

The magnetic dipole moment of the magnetorquers is set to 0.2 Am^2 instead of 0.2 Am^2 because the magnetorquer design may not be precise, thus having lower maximum value, and the magnetorquer performance close to the maximum value may be suboptimal.

All previously presented plots are created by utilizing the aforementioned simulation.

6.2 Simulation model

6.2.1 Reference Frames

In order to represent the kinematic and dynamic attitude of the satellite in a mathematical manner, while in orbit around the Earth, we need first to define a number of reference frames [10].

6.2.1.1 Earth Centered Inertial

The Earth Centered Inertial Frame is an inertial reference frame with the axis origin placed at the center of Earth. The Z axis is aligned with the axis of rotation of Earth. The X axis follows the vernal equinox, while the Y axis completes the orthogonal coordinate system, using the right hand rule.

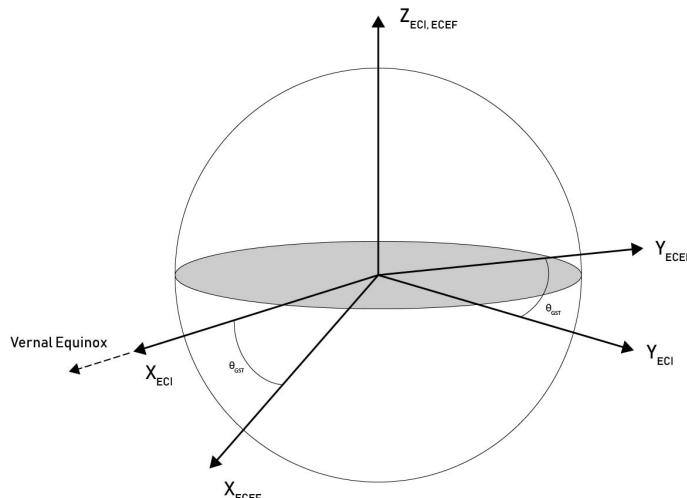


Figure 77: ECI and ECEF Frames

6.2.1.2 Earth-Centered Earth-Fixed

The Earth-Centered Earth-Fixed reference frame is based on the ECI frame principles. The difference lies upon the fact that the X and Y axes rotate along with the rotation of Earth, as shown in [Figure 77](#). The Z axis is aligned with ECI Z axis. The X axis points where the the first meridian intersects with the equator's plane. Lastly, the Y axis completes the orthogonal coordinate system, using the right hand rule.

6.2.1.3 Orbit Frame

The origin of the orbit frame is located on the satellite body. The Z axis follows the velocity vector, tangent to the orbit's trajectory. The X axis points towards nadir, while the Y axis completes the orthogonal coordinate system, using the right hand rule, as shown in [Figure 78](#)

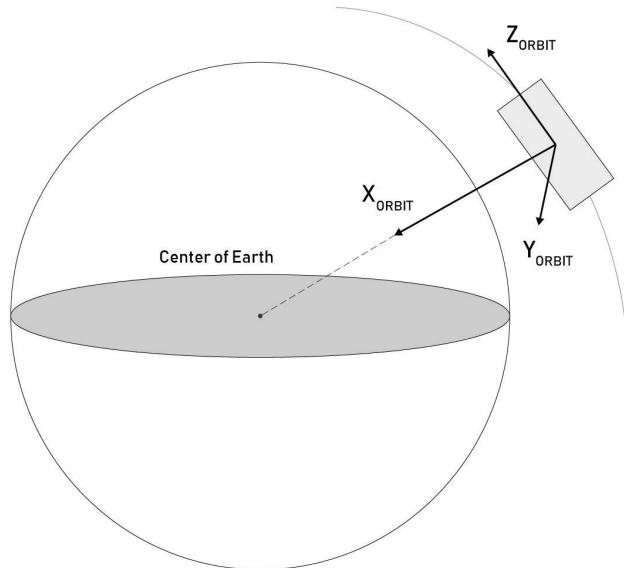


Figure 78: NED Frame

6.2.1.4 NED Frame

The NED reference frame is used mainly to represent the geomagnetic field from the IGRF model. The X axis of this frame points at the true North, while the Z axis points towards nadir. The Y axis completes the orthogonal coordinate system, using the right hand rule, as shown in the [Figure 79](#).

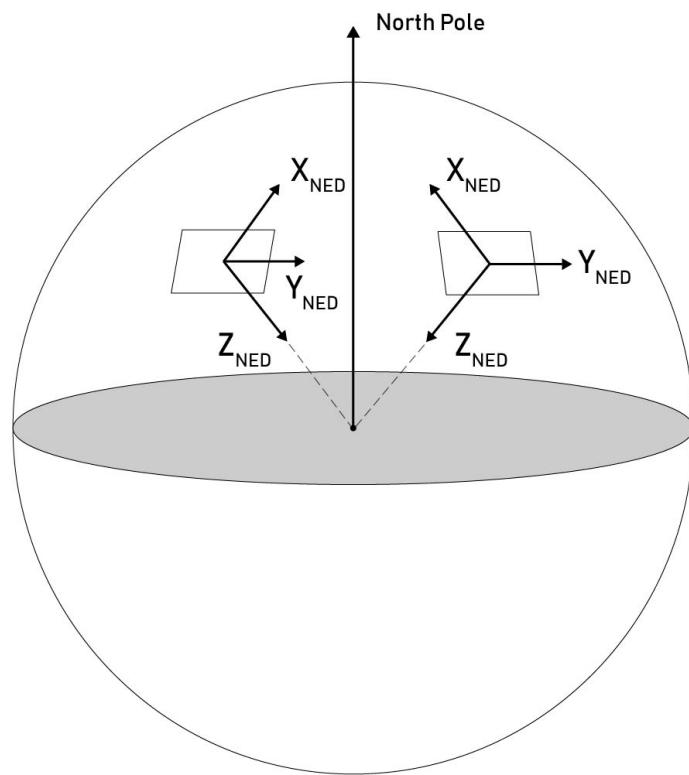


Figure 79: NED Frame

6.2.1.5 Body Frame

The origin of the body frame is located on the geometric center of the satellite. The X+ faces where the patch antenna is placed. The Z+ axis faces towards the deployable antenna. The Y+ axis completes the orthogonal coordinate system, using the right hand rule, as shown in the [Figure 80](#).

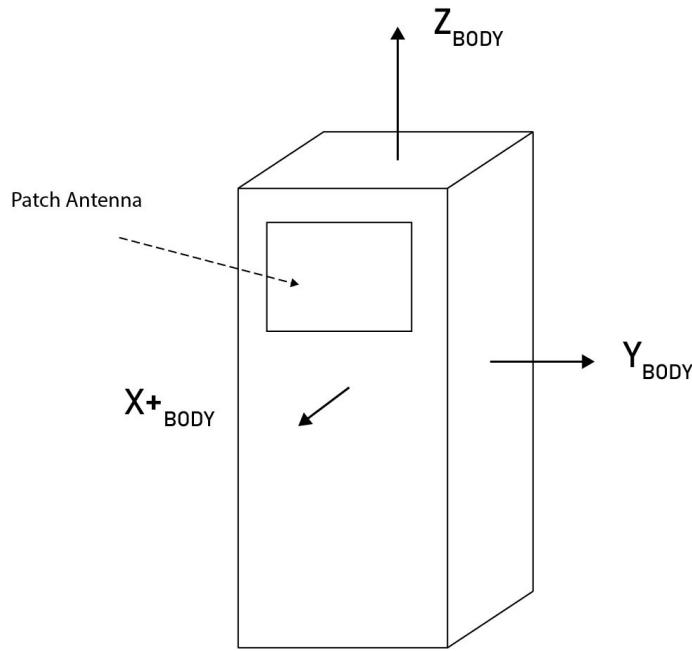


Figure 80: Body Frame

6.2.2 Transformation matrices

Having defined all the necessary reference frames which are to be used for the satellite's orientation representation, it is necessary to define the respective transformation matrices between them.

6.2.2.1 ECI to ECEF

The vectors which are defined in the ECI reference frame can simply be transformed to the ECEF reference frame via a turn with respect to the Z axis. Specifically:

$$\mathbf{R}_{IE} = \begin{bmatrix} \cos(\theta_{GST}) & \sin(\theta_{GST}) & 0 \\ -\sin(\theta_{GST}) & \cos(\theta_{GST}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (50)$$

where θ_{GST} parameter denotes the rotation angle between the two reference frames, which is going to be utilized in the following equation:

$$\theta_{GST} = \lambda_0 + \omega \cdot t \quad (51)$$

where the λ_0 is the longitude, ω is the rotation rate of the Earth, which equals to $\omega = 7.29211586 \cdot 10^{-5}$ rad/sec, and t is the elapsed time.

6.2.2.2 ECI to Orbit

The transformation matrix from the ECI reference frame to the orbit frame requires some orbital elements. Namely:

- Right ascension of the ascending node
- Inclination
- Argument of perigee
- Mean anomaly

All of the above mentioned orbital elements are provided by the SGP4 orbit propagator, as explained [Section 5.1](#). Then, the transformation matrix is defined as:[\[11\]](#)

$$\mathbf{R}_{\text{IO}} = \begin{bmatrix} \cos(\Omega)\cos(\omega) - \sin(\Omega)\cos(i)\sin(\omega) & \sin(\Omega)\cos(\omega) + \cos(\Omega)\cos(i)\sin(\omega) & \sin(i)\sin(\omega) \\ -\cos(\Omega)\sin(\omega) - \sin(\Omega)\cos(i)\cos(\omega) & -\sin(\Omega)\sin(\omega) + \cos(\Omega)\cos(i)\cos(\omega) & \sin(i)\cos(\omega) \\ \sin(\Omega)\sin(i) & -\cos(\Omega)\sin(i) & \cos(i) \end{bmatrix} \quad (52)$$

where Ω is the right ascension of the ascending node, ω is the sum of the argument of perigee and the mean anomaly, whereas i is the inclination. A visual representation of this transformation matrix is:

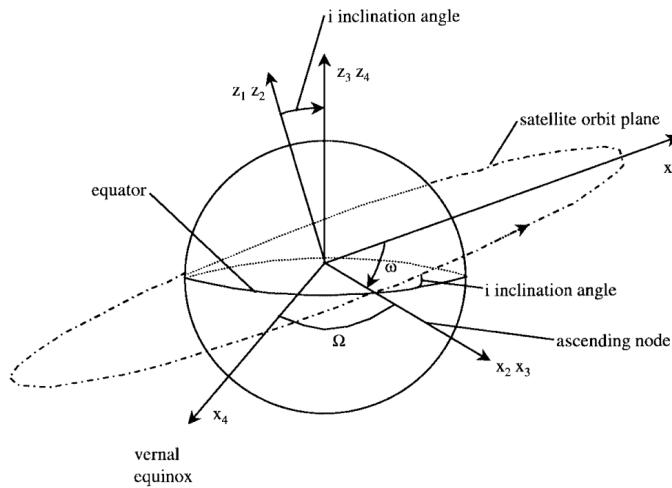


Figure 81: ECI to orbit transformation matrix

Due to the defined orbit frame [Paragraph 6.2.1.3](#), the transformation matrix Fig. 81 rows have to be adjusted. Specifically, the X row sign is changed and the Y and Z rows are swapped.

$$\mathbf{R}_{\text{IO}} = \begin{bmatrix} -\mathbf{R}_{\text{IO},1} \\ \mathbf{R}_{\text{IO},3} \\ \mathbf{R}_{\text{IO},2} \end{bmatrix} \quad (53)$$

6.2.2.3 NED to ECEF

The transformation matrix between NED and ECEF reference frames is required in order to translate the geomagnetic field vector to an inertial reference frame. The transformation matrix requires the following variables:

- Latitude
- Longitude

Then, the transformation matrix is defined as:

$$\mathbf{R}_{\text{NE}} = \begin{bmatrix} -\sin(\phi) \cos(\lambda) & -\sin(\lambda) & -\cos(\phi) \cos(\lambda) \\ -\sin(\phi) \sin(\lambda) & \cos(\lambda) & -\cos(\phi) \sin(\lambda) \\ \cos(\phi) & 0 & -\sin(\phi) \end{bmatrix} \quad (54)$$

where λ is the latitude and ϕ is the longitude.

6.2.2.4 Orbit to Body

The transformation matrix between any reference frame and the body frame is:

$$\mathbf{R}_{\text{OB}} = \mathbf{I} + 2\eta\mathbf{S}(\epsilon) + 2\mathbf{S}^2(\epsilon) \quad (55)$$

where $\mathbf{R}_{\text{BO}} = (\mathbf{R}_{\text{OB}})^T$, η is the quaternion scalar part, ϵ is the quaternion vector part, $\mathbf{S}()$ is the skew matrix and the \mathbf{I} is the identity matrix.

6.2.3 Environmental models and assumptions

While AcubeSAT will orbit around the Earth in LEO, several environmental phenomena will influence both the orientation and the attitude determination system. The most significant torques generated in LEO, as described in [12], [10] and [13], are :

- Gravitational torque
- Aerodynamic drag
- Solar radiation pressure
- Residual magnetic dipole moment

6.2.3.1 Gravitational torque

All non-symmetric, rigid bodies that are inside a gravitational field are subject to gravitational torque [12]. The gravitational torque is calculated as follows:

$$\tau_g = 3 \cdot (\omega_0)^2 \cdot R_{\text{OB},x} \times \mathbf{I} \cdot R_{\text{OB},x} \quad (56)$$

where $R_{\text{OB},x}$ is the first column of the rotation matrix from the orbit to the body reference frame, which is the body frame representation of a nadir-pointing unit vector,

ω_o is the angular velocity of the orbit frame relative to the ECI expressed in the orbit frame, I is the inertia tensor of the satellite. The aforementioned angular velocity is calculated from:

$$\omega_o = \sqrt{\frac{\mu}{R_c^3}} \quad (57)$$

where $\mu = G \cdot M_E$ is the standard gravitational constant of Earth, $G = 6.674 \cdot 10^{-11} \text{ m}^3 \text{kg}^{-1} \text{s}^{-2}$ is the gravitational constant and $M_E = 5.972 \cdot 10^{24} \text{ kg}$ is the total mass of Earth, R_c is the distance between the satellite and the center of Earth (6371km + 500km).

6.2.3.2 Aerodynamic drag

In LEO the external torque produced from the aerodynamic drag is significant and comparable to the gravity gradient disturbance. As the altitude of the orbit decreases, the aerodynamic torque becomes the dominant disturbance torque as the atmospheric density and the satellite velocity increase. From [13] the aerodynamic torque is calculated as

$$\tau_{ad} = \frac{1}{2} \rho C_d A V^2 (\mathbf{u}_v \times (\mathbf{C}_{pa} - \mathbf{C}_m)) \quad (58)$$

where $C_d \in [2, 2.5]$ is the drag coefficient. According to the FYS design specification, the drag coefficient that should be used in for this design is equal to 2, ρ is the atmospheric density, V is the linear satellite velocity and \mathbf{u}_v is the unit linear velocity vector of the satellite, which in this case is the third column of the rotation matrix as stated in the gravity gradient torque. The affected area A is defined as the area projected to the plane perpendicular to \mathbf{u}_v . The center of atmospheric pressure \mathbf{C}_{pa} is calculated as:

$$\mathbf{C}_{pa} = \frac{\int x P(x) dx}{\int P(x) dx} \quad (59)$$

where x is the distance from the origin of the body frame and P is the atmospheric pressure. Assuming that P is constant at every point along the cubesat's surface, the center of atmospheric pressure is taken at the center of the affected area.

In Equation (58) only the torque generated due to the displacement of the atmospheric pressure center relative to the center of mass \mathbf{C}_m is taken into account. The dissipation torque produced by the satellites spin is considered several orders of magnitude smaller than the first and therefore is being neglected.

6.2.3.3 Solar Radiation Pressure

The solar pressure depends on the angle of incidence of the sun rays on the satellite body frame and the reflectance of the cubesat's surface. Since the surface reflectance is unknown, the total torque can be calculated, as a worst-case scenario, by [12]:

$$\tau_{srp} = \frac{F_s}{c} A_s (1+q) \cos(i) \cdot (\mathbf{u}_s \mathbf{C}_{ps}) \quad (60)$$

where F_s is the solar constant (or solar flux density), c is the speed of light, q is the reflection coefficient, which ranges from 0 to 1 and in this case is assumed 0.6, i is the

sun angle of incidence and \mathbf{u}_s is the sun unit vector in body frame, derived from orbit propagator. The affected area A_s is defined as the projected area of the cubesat to the plane perpendicular to \mathbf{u}_s . The center of solar pressure \mathbf{C}_{ps} is assumed to be at the center of the affected area.

6.2.3.4 Residual Magnetic Torque

All electronics, especially the conductive elements, that the satellite uses, produce a magnetic dipole that interacts with the geomagnetic field, producing a disturbance torque. The torque created by the residual magnetic moment is given as:

$$\tau_{rm} = \mathbf{m} \times \mathbf{b} \quad (61)$$

where \mathbf{m} is the magnetic moment of the satellite and \mathbf{b} the geomagnetic field. Note that the total magnetic moment that the satellite produces, excluding the effect of the MTQ, can only be accurately measured after the assembly of the satellite. The residual magnetic torque is modeled as

$$\mathbf{m} = \mathbf{m}_{\text{const}} + \mathbf{m}_{\text{var}} \quad (62)$$

where $\mathbf{m}_{\text{const}}$ and \mathbf{m}_{var} are constant and variable values correspondingly of the residual magnetic dipole moment of the satellite, with

$$\mathbf{m}_{\text{var}} < \frac{\mathbf{m}_{\text{const}}}{9} \quad (63)$$

In the following plots the magnitude of the torques produced from the disturbances is presented for 1 orbit. An initial satellite angular velocity of $[0.035 \ 0.035 \ 0.035]$ rad/sec is selected.

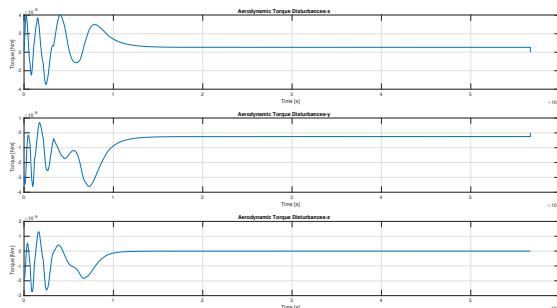


Figure 82: Aerodynamic Torque Disturbances

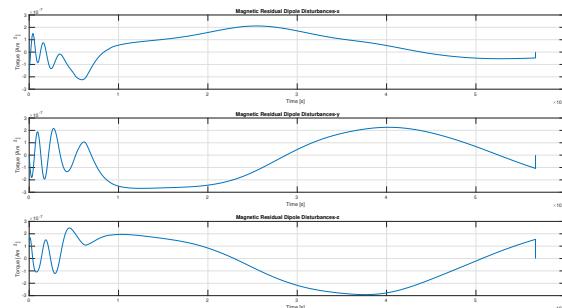


Figure 83: Magnetic Residual Dipole Disturbances

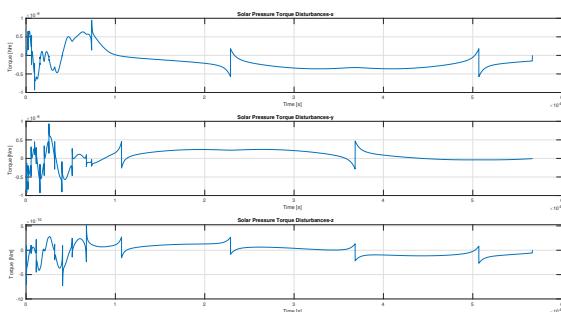


Figure 84: Solar Pressure Torque Disturbances

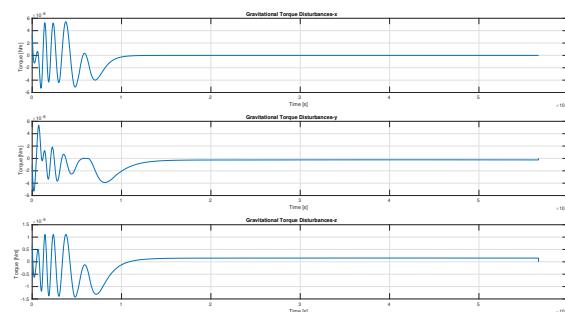


Figure 85: Gravitational Disturbances

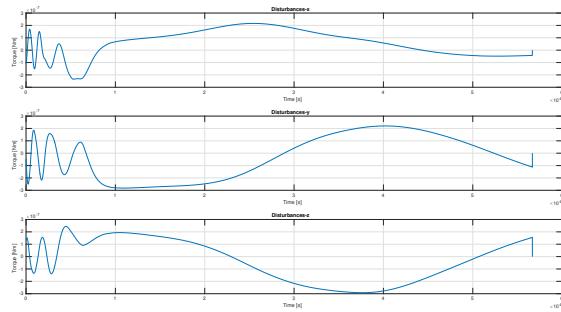


Figure 86: Total Disturbances

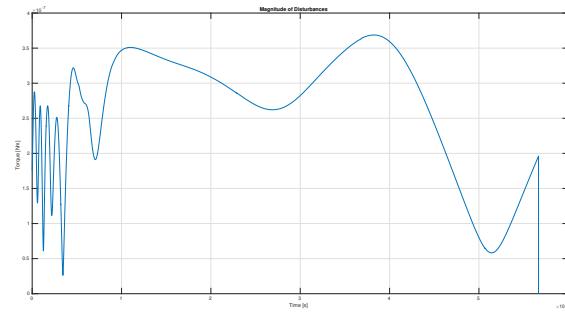


Figure 87: Total Disturbances Magnitude

6.2.3.5 Albedo effect

The sun sensors designed for CubeSat missions generally expect only one light source to exist. In practice, though, this is not the case due to the earth's albedo. The albedo effect is the diffused reflection of solar radiation from a body out of the total solar radiation it receives. The diffused reflected light from the Earth affects the attitude determination system, due to the additional error in the sun sensors. To model the impact of this effect, Earth Albedo Toolbox for MATLAB [14] is utilized. For it to work, reflectivity data from Earth is required. Currently, data from the NASA TOMS project [15] is used. Albedo compensation is included inside MEKF. Coarse sun sensor modeling is described in Paragraph 6.2.5.3.

6.2.4 Orbit and attitude kinematic/dynamic propagators

6.2.4.1 Orbit propagator

As thoroughly explained in Section 5.1, the SGP4 orbit propagator is utilized in order to autonomously determine the satellite's position and orbital parameters at any given point while in-orbit.

6.2.4.2 Kinematic propagator

In order to propagate the kinematic and dynamic attitude of the satellite, we utilized the equations below based on the book "Fundamentals of Spacecraft Attitude Determination and Control" [10]

Kinematics cover those aspects of motion that can be analyzed without consideration of forces or torques. Our kinematic model is described by:

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{q} \odot \boldsymbol{\omega}_B^{IB} = \frac{1}{2} \Xi(\mathbf{q}) \boldsymbol{\omega}_B^{IB} \quad (64)$$

where \mathbf{q} is the quaternion that describes the orientation of the satellite, $\boldsymbol{\omega}$ is the angular velocity of the satellite between orbit frame and body frame expressed in the body frame, and $\Xi(\mathbf{q})$ is equal to

$$\Xi(\mathbf{q}) = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \quad (65)$$

Equation (64) gives $\boldsymbol{\omega}_B^{IB}$ as a function of the quaternion rate as

$$\boldsymbol{\omega}_B^{IB} = 2\Xi^T(\mathbf{q})\dot{\mathbf{q}} \quad (66)$$

6.2.4.3 Dynamic propagator

Dynamics cover those aspects of motion that can be analyzed with consideration of forces or torques. Our dynamic model is described by Equation (67)

$$\dot{\boldsymbol{\omega}}_B^{BI} = (I_B^c)^{-1} [\boldsymbol{\tau}_B^c - \boldsymbol{\omega}_B^{BI} \times (I_B^c \boldsymbol{\omega}_B^{BI})] \quad (67)$$

where $\dot{\boldsymbol{\omega}}_B^{BI}$ is the first derivative of the angular velocity of the spacecraft between orbit frame and Body frame, expressed in the body frame, I_B^c is the inertia tensor expressed in the body frame and $\boldsymbol{\tau}_B^c$ is the total torque applied to the satellite. Each quantity with the superscript c is expressed with respect to the center of mass of the spacecraft. Additionally, the total torque is equal to

$$\boldsymbol{\tau}_{\text{total}} = \boldsymbol{\tau}_{\text{actuators}} + \boldsymbol{\tau}_{\text{disturbances}} \quad (68)$$

where, during AOCS detumbling mode:

$$\boldsymbol{\tau}_{\text{actuators}} = \boldsymbol{\tau}_{\text{magnetic}} \quad (69)$$

(33) and, during the AOCS nominal mode:

$$\boldsymbol{\tau}_{\text{actuators}} = \boldsymbol{\tau}_{\text{magnetic}} + \boldsymbol{\tau}_{\text{rw}} \quad (70)$$

(33), (91). Moreover, the $\boldsymbol{\tau}_{\text{disturbances}}$ consists of the torques described in (6.2.3)

6.2.4.4 Safe Mode

During AOCS safe mode the torque produced by the actuators is equal to zero. The effect that the environmental perturbations have on the satellites orientation and angular velocity during this mode, for 10 orbits, are presented in [Figure 88](#), [Figure 89](#).

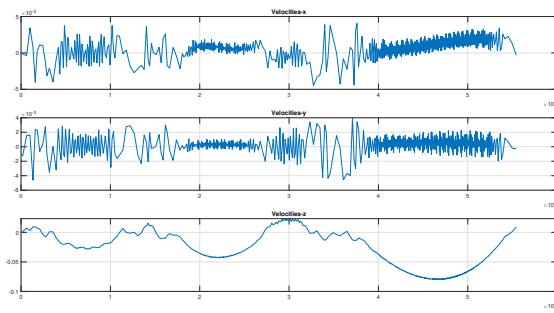


Figure 88: Safe Mode Angular Velocities

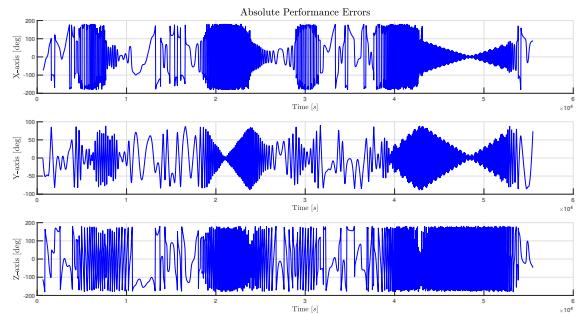


Figure 89: Safe Mode Euler Angles

6.2.5 Sensor model

6.2.5.1 Magnetometer model

Magnetometer noise can be modeled as a combination of bias, scale error and white-noise terms. The RM3100 magnetometer has no inherent bias drift [Section 2](#). Assuming the magnetometer is calibrated, the bias and scale errors are eliminated. Under these assumptions, magnetometer noise is modeled solely as additive Gaussian white noise.

$$\hat{\mathbf{b}} = (I + A) \cdot \mathbf{b} + \boldsymbol{\eta} \quad (71)$$

Where \mathbf{b} is the real value of the magnetic induction, A is the misalignment rotation Matrix, I is the identity matrix and $\boldsymbol{\eta}$ is a Gaussian white noise process. $\boldsymbol{\eta}$ consists of magnetic and thermal noise. Based on [\[16\]](#), the RM3100 sensor was tested by the manufacturer through various scenarios, concluding that the sensor's mean noise value equals to 0 and its standard deviation being close to 20 nT. As a result the magnetometer sensor noise can be modelled as Gaussian white noise with a standard deviation of 20 nT. In our approach the standard deviation value was set to 30 nT as indicated by the component datasheet.

6.2.5.2 Gyroscope model

According to Markley-Crassidis [\[10\]](#), the gyroscope measurement model is given by:

$$\boldsymbol{\omega}(t) = (1 + A)\boldsymbol{\omega}^{\text{true}}(t) + \boldsymbol{\beta}^{\text{true}}(t) + \boldsymbol{\eta}_v \quad (72)$$

$$\dot{\boldsymbol{\beta}}^{\text{true}} = \boldsymbol{\eta}_u \quad (73)$$

where ω is the measured angular velocity in the gyroscope reference frame, ω^{true} the real angular velocity in the same frame, A is the misalignment rotation Matrix, I is the identity matrix, β^{true} the drifting bias of the gyroscope, η_v and η_u are independent zero-mean Gaussian white-noise processes. A more general model could be used including scale factors and misalignments, however they have not yet been modeled, as there is no way of knowing them considering the satellite has not been assembled yet. Discretizing the above model yields the following equations [10]:

$$\omega_{k+1} = (I + A)\omega_{k+1}^{\text{true}} + \frac{1}{2}(\beta_{k+1}^{\text{true}} + \beta_k^{\text{true}}) + \left(\frac{\sigma_v^2}{\Delta t} + \frac{1}{12}\sigma_u^2\Delta t \right)^{\frac{1}{2}} N_{v_k} \quad (74)$$

$$\beta_{k+1}^{\text{true}} = \beta_k^{\text{true}} + \sigma_u\Delta t^{\frac{1}{2}} N_{u_k} \quad (75)$$

where the subscript k denotes the k -th time-step, N_{v_k} and N_{u_k} are zero-mean Gaussian white-noise processes with covariance each given by the identity matrix while σ_v^2 and σ_u^2 are the covariances associated with white-noise and bias respectively.

6.2.5.3 Coarse sun sensor

Noise due to Earth's Albedo is included by utilizing Earth Albedo Toolbox for MATLAB [14], which, given reflectivity data, returns the percentage of sunlight diffused from Earth. The details of implementation are described below:

- Each of the six coarse sun sensors needs to be modeled separately. To that end, six frames are defined, one for every sensor. Four frames are defined by rotating the body frame around the Y axis (0,90,-90,180) $^\circ$. The other two are defined by rotating the body frame around the Z axis (90,-90) $^\circ$
- Albedo radiation is assumed to come from nadir. Nadir vector is acquired in the body frame by rotating the satellite position vector in the ECI frame with the ECI to Body quaternion, and then negating it, to yield the vector pointing from the Body frame to Earth, instead of away from Earth.
- Current induced in the coarse sun sensors is proportional to the cosine of the angle between the vector normal to the sensor and the light source in the sensor frame. Symbolically:

$$i = I_s \cos(\theta) \quad (76)$$

$$\cos(\theta) = \mathbf{source}_{\text{sensor}} \cdot \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \quad (77)$$

If $\cos(\theta) < 0$ then the sensor does not detect light from this source.

- Currents are calculated for each sensor by adding the currents induced directly by sunlight and Earth Albedo. Assuming $I_s = 1$ for direct sunlight, current induced by Earth's Albedo is scaled by the Albedo percentage as calculated from the MATLAB toolbox. Shot noise as described above is added here.

$$i_{\text{sun}} = \mathbf{sun}_{\text{sensor}} \cdot \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \quad (78)$$

$$i_{\text{albedo}} = \mathbf{albedo}_{\text{nadir}} \cdot \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \quad (79)$$

$$i_{\text{total}} = (i_{\text{sun}} + i_{\text{albedo}})(1 \pm 0.01w_k) \quad (80)$$

where w is a Poisson process with $\lambda = 1$.

- Finally, currents in opposite faces are subtracted and the resulting vector normalized, yielding the final sun sensor measurement in the body frame.

$$\mathbf{sun}_{\text{meas}} = \frac{(i_1 - i_4, i_6 - i_5, i_2 - i_3)}{\|(i_1 - i_4, i_6 - i_5, i_2 - i_3)\|} \quad (81)$$

6.2.6 Actuator model

With reference to the models of the actuators, functions, which simulate the operation of the actuators, are generated.

6.2.6.1 Magnetorquers

AcubeSAT uses the *ISIS*' iMTQ ready-to-use board. It consists of a set of two X and Y axis magnetorquer rods and one Z axis air core magnetorquer. The specifications of each type of magnetorquer are in ??.

The magnetorquers generate magnetic dipole moment, \mathbf{m} , which interacts with the earth's geomagnetic field, \mathbf{b} , to induce a torque acting on the satellite $\boldsymbol{\tau}_m$, as described in [Equation \(33\)](#).

Consequently the magnetic moment generated by the magnetorquers is:

$$\mathbf{m} = \frac{\mathbf{b} \times \boldsymbol{\tau}_m}{\mathbf{b}^T \mathbf{b}} \quad (82)$$

6.2.6.2 Saturation of the Magnetorquers

This paragraph is based on the work of Forbes and Damaren [9].

In a practical context, current limitations associated with the magnetic torque rods impose a saturation limit on \mathbf{m} , such that $|m_j| \leq m_{\max}$, where m_{\max} is the saturation limit of each identical torque rod. We extend our geometric method to handle saturation issues associated with magnetic actuation. The actuator configuration on AcubeSAT disallows the desired control, \mathbf{u} , to be fully realized. However, a control torque that is reduced in magnitude as a result of saturation, \mathbf{u}^s , is permissible. Given that \mathbf{u}^s can be applied, it follows that the direction of each control vector be equivalent, that is, $\hat{\mathbf{u}}^T \hat{\mathbf{u}}^s = 1$.

Upon saturation of any one of the torque rods, the total magnetic torque to be applied to the spacecraft is scaled accordingly. Consequently, according to the [Equation \(49\)](#):

$$\boldsymbol{\tau}_m^s = k_{\text{ma}}^s \hat{\boldsymbol{\tau}}_m = \mathbf{b}^{\times T} \mathbf{m}^s \quad (83)$$

where $\mathbf{m}^s = k_{\text{ma}}^s \mathbf{m}'$, $\mathbf{m}' = \frac{\mathbf{m}}{\|\boldsymbol{\tau}_m\|}$ and $k_{\text{ma}}^s = \min(|\frac{m_{\max}}{m'_1}|, |\frac{m_{\max}}{m'_2}|, |\frac{m_{\max}}{m'_3}|)$.

Both τ_m and τ_m^s point in the same direction, but have different magnitudes. It follows that

$$\mathbf{u}^s = k_{ma}^s \hat{\boldsymbol{\tau}}_m + k_{wa}^s \hat{\boldsymbol{\tau}}_w \quad (84)$$

where $k_{wa}^s = \frac{k_{ma}^s \cdot k_{wa}}{k_{ma}}$, owing to the requirement $\hat{\mathbf{u}}^T \hat{\mathbf{u}}^s = 1$. It follows that the torques to be applied by the magnetic actuators and the reaction wheel are

$$\tau_m = k_{ma}^s \hat{\boldsymbol{\tau}}_m \quad \text{and} \quad \tau_w = k_{wa}^s \hat{\boldsymbol{\tau}}_w \quad (85)$$

Keep into consideration that the magnitude of the wheel torques scale so that $\hat{\mathbf{u}}$ and $\hat{\mathbf{u}}^s$ are collinear.

6.2.6.3 Residual Magnetic Dipole Compensation

The main disturbance induced in the satellite is considered to be the magnetic disturbance, generated by the residual magnetic dipole of the satellite electronics and magnetized components. In order to compensate for this disturbance, the constant part of the residual magnetic dipole is estimated as $\hat{\mathbf{m}}$ and set to $[0.048 \ 0.051 \ 0.047]$, which is then subtracted from the magnetic dipole value corresponding to the commanded torque, as

$$\mathbf{m}_{\text{applied}} = \mathbf{m}_{\text{commanded}} - \hat{\mathbf{m}} \quad (86)$$

6.2.6.4 Magnetorquers Power

The magnetic moment \mathbf{m} of each magnetorquer can also be calculated as:

$$m = \mu_t NIA \quad (87)$$

where the units of \mathbf{m} is Am^2 , N is the number of windings, I is the flowing current in the coil and A is the area enclosed by the coil. For torquer rods, we calculate μ_t as $\mu_t = 0.3 \cdot \left(\frac{\text{length}}{\text{diameter}} \right)^{1.5}$. For air core magnetorquers $\mu_t = 1$.

Therefore, the current in each coil is equal to:

$$I_i = \frac{(\mathbf{b} \times \boldsymbol{\tau}_m)_i}{\mu_{t_i} N_i A_i \cdot \mathbf{b}^T \mathbf{b}} \quad (88)$$

where $i = [x, y, z]$ and refers to each magnetorquer.

The current I is calculated using the equation:

$$V_i = I_i \cdot R_i \quad (89)$$

where R_i is the resistance of each magnetorquer. The power P is calculated using the equation:

$$P_i = V_i \cdot I_i \quad (90)$$

6.2.6.5 Reaction Wheel

Torques are produced on the satellite's Z-axis by accelerating or decelerating the reaction wheel. Let the torque demanded by the satellite for the reaction wheel be denoted as τ_w . The τ_w is given by the equation:

$$\tau_w = J_w \cdot \dot{\Omega} + b \cdot \Omega + c \cdot \text{sign}(\Omega) \quad (91)$$

[17] where Ω is the angular velocity of the reaction wheel, J_w is the inertia of the flywheel, b is the viscous friction constant (Nms) and c is the Coulomb friction constant (Nm).

The aforementioned constants are at the moment rough estimations and they will be calculated accurately through experimental procedures after we acquire the component.

No information is provided by the manufacturer regarding flywheel inertia. Consequently, since we are aware of the maximum angular momentum ($= 0.3 \text{ mN} \cdot \text{m} \cdot \text{s}$) and the maximum angular speed of the reaction wheel ($= 15000 \text{ rpm}$), we can calculate the J_w , using the equation

$$h_{\max} = J_w \cdot \Omega_{\max} \Rightarrow J_w = \frac{h_{\max}}{\Omega_{\max}} \quad (92)$$

where h_{\max} is the maximum angular momentum of the reaction wheel.

To calculate the viscous friction coefficient b , a procedure is performed, where the wheel is accelerated up to a given rate, and then left free until it is completely stopped. The solution of differential motion [Equation \(91\)](#) in this situation leads to:

$$\Omega = \Omega_0 \frac{e^{-\beta t} - e^{-\beta t_f}}{1 - e^{-\beta t_f}} \quad (93)$$

where $\beta = \frac{b}{J_w}$, Ω_0 is the initial decay rate and t_f is the decay time. Through least square error fitting, are calculated the b and c constants. The values of Ω_0 , J_w and t_f are known.

Neglecting nonlinear effects present in the conversion from current to torque because of lack of data, it is considered that the torque is linear with the current I :

$$\tau_w = k_m \cdot I \quad (94)$$

where k_m is the motor constant. At the moment it is a rough estimation and will be calculated accurately through experimental procedures, by measuring the steady state of the angular velocity for a given commanded current, after the component is acquired. The result can be modeled using the steady state solution of the equation of motion:

$$k_m \cdot I = J_w \cdot \dot{\Omega} + b \cdot \Omega + c \cdot \text{sign}(\Omega) \quad (95)$$

adjusted after minimizing the mean square error. Since b and c are now known, k_m can be calculated.

Finally, the current I is calculated as:

$$I = \frac{\tau_w}{k_m} \quad (96)$$

The voltage V is calculated using the equation;

$$V = I \cdot R_b \quad (97)$$

where R_b is the resistance of the reaction wheel's armature. The power P is calculated using the equation;

$$P = V \cdot I \quad (98)$$

6.2.6.6 Saturation of the Reaction Wheel

Over time reaction wheels may build up enough stored momentum to exceed the maximum speed limit of the wheel, saturating it. In this case, the magnetorquers are utilized for momentum dumping.

In AcubeSAT, a reaction wheel on the Z satellite body axis of the spacecraft is used, aligned with the principle axis of inertia. The total moment of inertia is calculated as

$$\mathbf{h} = \mathbf{h}_s + \mathbf{h}_w \quad (99)$$

where \mathbf{h}_s is the moment of inertia of the spacecraft (excluding the wheel) in Z axis and \mathbf{h}_w is the moment of inertia of the reaction wheel. In absence of external forces, momentum is preserved and the time derivative of the total momentum is zero, hence:

$$\dot{\mathbf{h}} = 0 \implies \dot{\mathbf{h}}_w + \dot{\mathbf{h}}_s = 0 \implies \dot{\mathbf{h}}_s = -\dot{\mathbf{h}}_w \quad (100)$$

So, if the momentum of the reaction wheel increases, we get the opposite reaction in the change of momentum of the spacecraft. Knowing that:

$$\mathbf{h} = J_w \cdot \boldsymbol{\Omega} \quad (101)$$

where J_w the moment of inertia and $\boldsymbol{\Omega}$ the rotation speed of the wheel. It is evident that accelerating the reaction wheel in one direction results in the spacecraft accelerating in the opposite direction. Precisely:

$$\dot{\omega}_s = -\frac{J_w}{J_s} \cdot \dot{\Omega}_w \quad (102)$$

In a case of reaction wheel saturation, in order to avoid the disturbance of the satellite attitude, desaturation of the wheel is required. By using the magnetorquers, momentum from the wheel is reduced so it has margin to accelerate again, if required. The application of the magnetic torque itself does not slow down the reaction wheel, but it provides margin in the momentum balance to reduce the momentum, and thus the speed, of the reaction wheel:

$$\dot{\mathbf{h}} = \boldsymbol{\tau}_m \quad (103)$$

where $\boldsymbol{\tau}_m$ the magnetic torque.

For the desaturation of the Reaction Wheel, we follow the procedure below:

1. We create $\tau_{m_{\text{added}}}$ such that:

$$\tau_{m_{\text{added}}} = K \cdot \dot{h}_w \quad (104)$$

for some suitable value of K , which indicates the percentage of \dot{h}_w added on the torque given on Z-axis through magnetorquers and is calculated through the method of trial and error.

2. $\tau_{m_{\text{added}}}$ is added to the initial $\tau_{m_{\text{init}}}$, calculated as described in [Equation \(45\)](#)
3. The new magnetic dipole is calculated, using the [Equation \(82\)](#)
4. The new magnetic dipole is scaled not to exceed the limit of the magnetic dipole that each magnetorquer can produce, being $m = 0.2 \text{ Am}^2$
5. The final $\tau_{m_{\text{fin}}}$ is recalculated.
6. From the initial $\tau_{w_{\text{init}}}$ the torque added to the initial $\tau_{m_{\text{init}}}$ is subtracted, namely:

$$\tau_{w_{\text{init}}} = \tau_{m_{\text{fin}}} - \tau_{m_{\text{init}}} \quad (105)$$

Finally, the wheel can spin down without affecting the satellite attitude.

6.2.6.7 Zero-crossings of the Reaction Wheel

The maximum model error occurs either when the wheel starts moving from rest, namely from zero speed, or when the wheel speed crosses the dead-zone and consequently the zero speed. The latter occurs when the reaction wheel needs to reverse its rotational speed direction from clockwise to anticlockwise and vice versa. The dead-zone is the region around zero-speed, where the wheel does not respond accurately to the commanded current and causes a sudden change in the dynamic behavior of the wheel. Regarding the RW210 dead-zone, it is considered to be around the rotational speed range of RPM, according to *Hyperion Technologies*.

Hyperion Technologies has provided us with detailed figures, which depict the functionality of the reaction wheel in the dead-zone for four different angular accelerations that the wheel may have during the dead-zone. Below, we append one of those figures for an angular acceleration = $\pm 50 \text{ rpm/sec}$.

In order to model the performance of the reaction wheel through the zero-crossings, linear approximation is used to depict the behaviour of the wheel, as presented on the aforementioned figures for an angular acceleration of $\pm 25 \text{ rpm/sec}$, $\pm 50 \text{ rpm/sec}$, $\pm 100 \text{ rpm/sec}$ and $\pm 200 \text{ rpm/sec}$. Since, we obtain no figures for angular acceleration between those values, we assumed that the functionality of the wheel will be approached as shown in ??: where the values above are applicable both with positive and with negative sign, satisfying the clockwise and anticlockwise rotation of the reaction wheel.

The required acceleration is scaled based on the given values, with regards to each figure.



6.2.7 SW/HW delays

6.2.7.1 SW delays

We have to clarify that the AOCS software is as of now written only in MATLAB, for utilization in our simulations. The entire software will be written in C programming language, in order to be embedded in our AOCS MCU, during phase D of the programme.

The delays of the AOCS software are initially calculated using MATLAB. Afterwards, in order to estimate the delay on the actual microcontroller without requiring transferring the bulk of the implementation, the delays are multiplied by an experimentally determined constant ($C = 3.65$). In order to calculate this delay, we follow the algorithm:

1. The delay for the B-dot controller, written in MATLAB code, is calculated.
2. The code is rewritten in the C programming language and uploaded to a microcontroller. For our experiments we used an STM32L4R5ZI ARM Cortex-M4 MCU clocked at 72 MHz.
3. The delay for the B-dot controller is calculated on the microcontroller.
4. The constant is equal to

$$C = \frac{\text{Delay on microcontroller}}{\text{Delay in MATLAB}} \quad (106)$$

5. We consider that constant as an estimation value, which indicates the relation between the actual delay on a microcontroller and the delay in total MATLAB code.

The estimated delays are shown in [Table XVIII](#).

Table XVIII: Software Delays

Software delays		
Function	Delay in MATLAB (ms)	Delay on Microcontroller (ms)
B-dot controller	0.023	0.084
SGP4	0.074	0.27
Orbit Propagator		
IGRF	0.13	0.47
Eclipse	0.0014	0.0051
Sun position	0.013	0.047
Wahba	4.047	14.77
Initial bias estimation	4.819	17.59
Kalman correct	3.359	12.26
Kalman predict	0.978	3.57
Quaternion Control Transformation	0.092	0.336
PD controller	0.075	0.274
Saturation of Reaction Wheel	0.567	2.07

6.2.7.2 HW delays

The sampling periods of our components have been considered as the maximum AOCS hardware delays. Regarding our sensors, those delays are actually zero, due to our control cycle, since the measurements are prepared in the output register of the corresponding sensor at the moment we obtain the measurements for utilization in our determination calculations. In respect to AOCS actuators, we obtain no information by the manufacturers, for any possible delays that occur between the signal from the AOCS MCU and the actuation. The maximum hardware delays are shown in [Table XIX](#).

Table XIX: Hardware delays

Hardware delays		
Component	Sampling frequency (Hz)	Maximum delay (ms)
MEMS Gyroscope ADXRS453	485	2.1
Magnetometer RM3100	300	3.3
Coarse Sun Sensor		No data
Magnetorquers		No data
Reaction Wheel		No data

6.2.8 Simulation/model sampling times and frequencies

For the purpose of this mission two different control cycles are designed. The first is used during the detumble mode and the second during the nominal mode. Each cycle lasts for 1 sec and is divided in 10 steps of 0.1 sec. The procedures that need to be executed in each timestep are determined at the beginning of it.

6.2.8.1 Detumbling mode

The first timestep of detumbling is empty. At the beginning of the 2 following timesteps measurements from the magnetometers are collected. These calculations have a time difference of 0.1 sec so the sampling frequency is 10 Hz. Immediately after the last measurement is taken, at the time moment of 0.3 sec, a magnetic dipole moment is calculated to be applied to the magnetorquers for the next 7 timesteps. The time difference between 2 consecutive calculations of the magnetic dipole moment is 0.1 sec so the sampling frequency again is 10 Hz.

6.2.8.2 Nominal mode

The first timestep of nominal mode is reserved for calculations regarding the satellite orbit. More specifically, the models used during this timestep are:

- SGP4 propagator
- Magnetic field model
- Sun position model
- Eclipse model
- Albedo model

The time difference between two consecutive calculations for these models is 1 sec, hence the sampling frequency is 1 Hz. For the following 0.2 sec, the attitude determination system is being used. At the beginning of each timestep the following procedures take place in order:

- Collection of data from sun sensors, magnetometers and gyroscope.
- MEKF correct function.
- MEKF predict function.

These procedures occur, with a time difference of 0.1 sec, for three times, hence the sampling frequency for each of them is 10 Hz. Immediately after the last call of MEKF correct function the control cycle commences. For each of the seven remaining timesteps in the nominal mode, a desired torque is commanded and split between the actuators. The time difference for two consecutive calculations is 0.1 sec, so the sampling frequency is 10 Hz.



Requirements close-out — Analysis

The above document is the close out reference for requirements:

Requirement ID	Requirement Text	Status & Rationale	Reference
ADCS-FUN-150	Kalman Filter and sensor measurement acquisition shall run at a frequency of at least 10 Hz.	Compliant at Phase C since all simulation analyses conducted are with a frequency of 10 Hz	Section 6.3
ADCS-FUN-160	The ADCS control sequence shall run at a frequency of at least 10 Hz.	Compliant at Phase C since all simulation analyses conducted are with a frequency of 10 Hz	Section 6.3

6.3 Analyses

The performance of the satellite in nominal conditions and for the selected parameters' values is shown in [Section 6.4](#).

Table XX: Nominal Parameters

Parameters	Values	Units
Orbits	2	
q_desired	[1, 0, 0, 0]	
Satellite's Initial Angular Velocity	(varying)	rad/sec
B-dot gain	1	
PD Kp gain	$1 \cdot 10^{-5} \cdot \text{diag}([20, 100, 90])$	
PD Kd gain	$1 \cdot 10^{-4} \cdot \text{diag}([90, 90, 90])$	
PD Kp gain during eclipse	$1.2 \cdot K_p$	
PD Kd gain during eclipse	$20 \cdot K_d$	
Q Variance	$0.5 \cdot 10^{-5} \cdot \text{eye}(6, 6)$	
Q Variance during eclipse	6×6 matrix	
R Variance	6×6 diagonal matrix	
R Variance during eclipse	$10^5 \cdot R_{\text{variance}}$	
Magnetorquers Maximum Magnetic Dipole	0.2	Am ²
Reaction Wheel Maximum Torque	$1 \cdot 10^{-4}$	Nm
Reaction Wheel Inertia	$1.9 \cdot 10^{-6}$	Kgm ²
Reaction Wheel Coulomb Friction Constant	$1.9 \cdot 10^{-7}$	Nm
Reaction Wheel Viscous Friction Constant	$1.9 \cdot 10^{-9}$	Nms
Reaction Wheel Desaturation Limit	10000	rpm

The inertia matrix of the satellite is defined in [Section 4.2.1](#).

6.3.1 Gain Tuning

In order to achieve minimum absolute performance errors, gain tuning is performed on PD controller, utilizing the MATLAB Genetic Algorithm toolbox. The proper gains are calculated for all SSO TLEs, corresponding to altitude equal to 500km and 600km and various LTANs. For MEKF, the variance of the process noise (Q) and the measurement noise (R) was tuned manually, as mentioned in [Table XX](#).

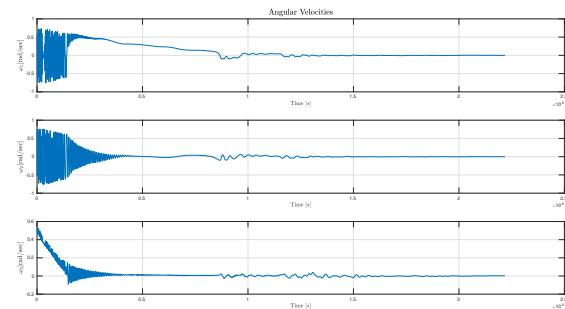
6.4 Simulation Performance campaign plan & results

6.4.1 Nominal scenarios

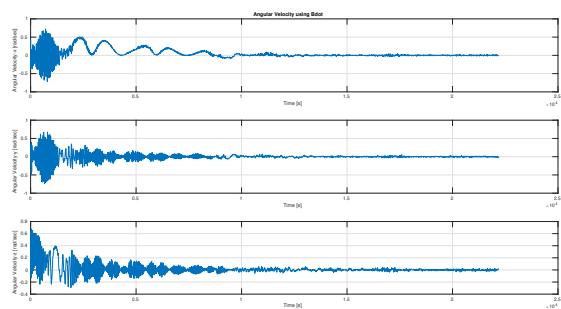
In order to demonstrate the attitude control stability and feasibility of the mission with regards to AOCS, the performance of the satellite in nominal conditions is simulated.

6.4.1.1 Detumbling

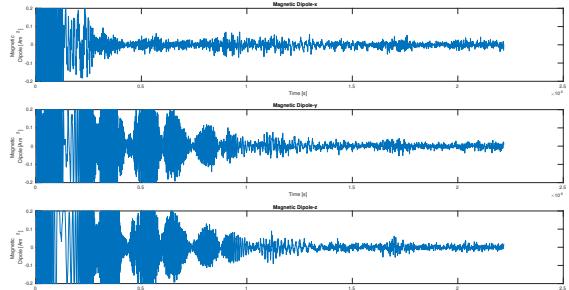
The detumbling simulation is carried out in Total AOCS Simulation, using the updated inertia matrix of the satellite, as shown in [Section 4.2.1](#), from an initial angular velocity of $\begin{bmatrix} \frac{\pi}{6} & -\frac{\pi}{6} & \frac{\pi}{6} \end{bmatrix}$ using the SSO TLE [[Section 5.1](#)] and a duration of 22180 sec, namely 4 orbits.



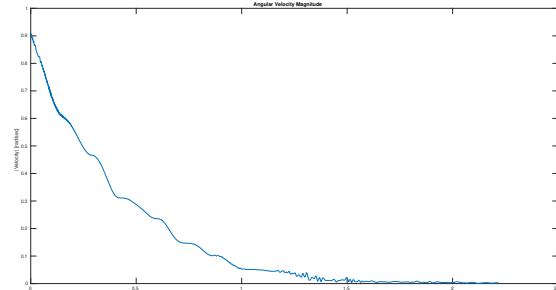
(a) Velocities



(b) Angular Velocity approximation



(c) Magnetic Dipole



(d) Angular Velocity Magnitude

Figure 90: Nominal Detumbling

6.4.1.2 Nominal

The Nominal simulation is carried out in Total AOCS Simulation using the SSO TLEs [[Section 5.1](#)] and the updated inertia matrix of the satellite, as shown in [Section 4.2.1](#). The simulation's duration is 11090 sec, namely 2 orbits.

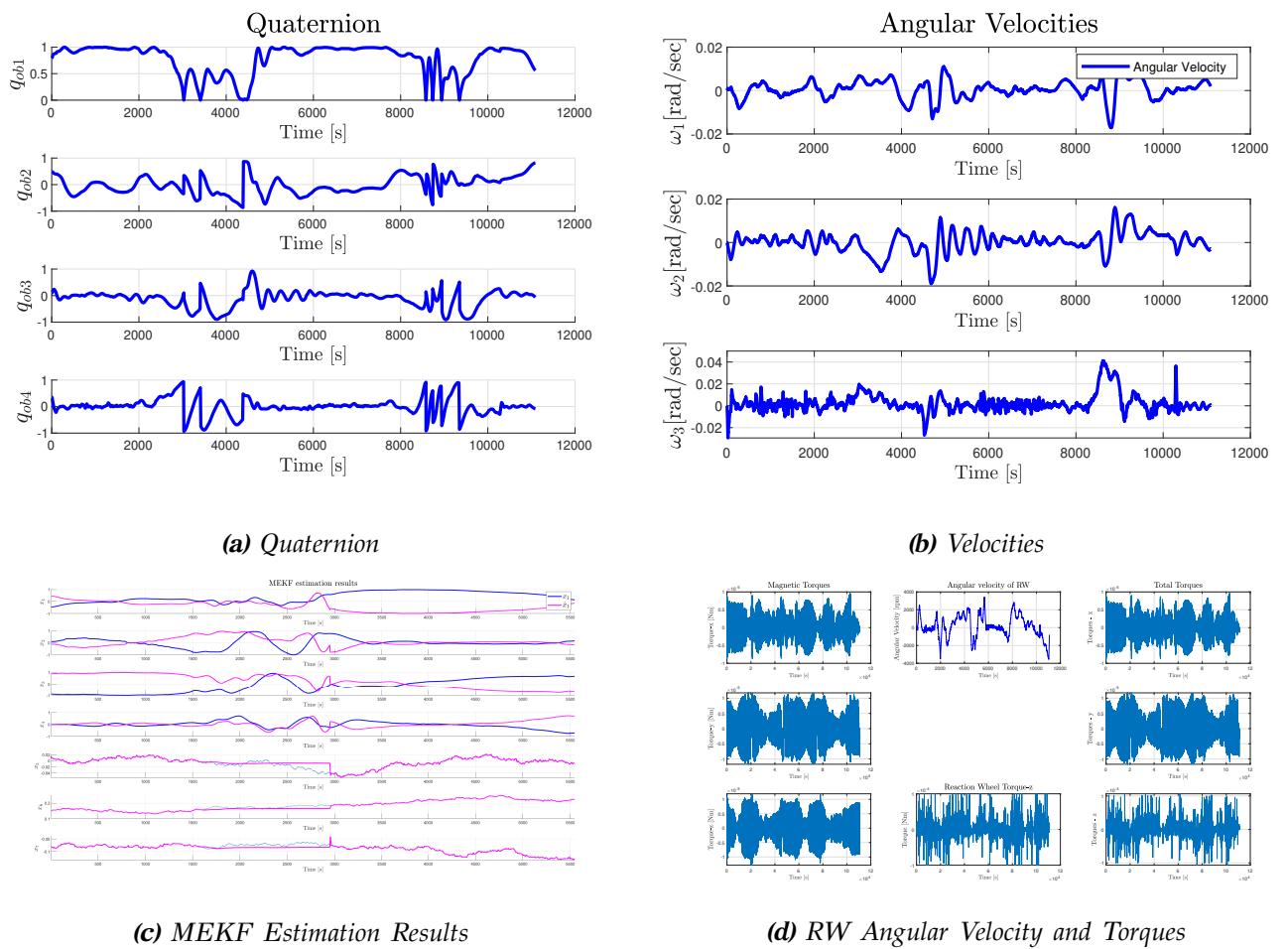
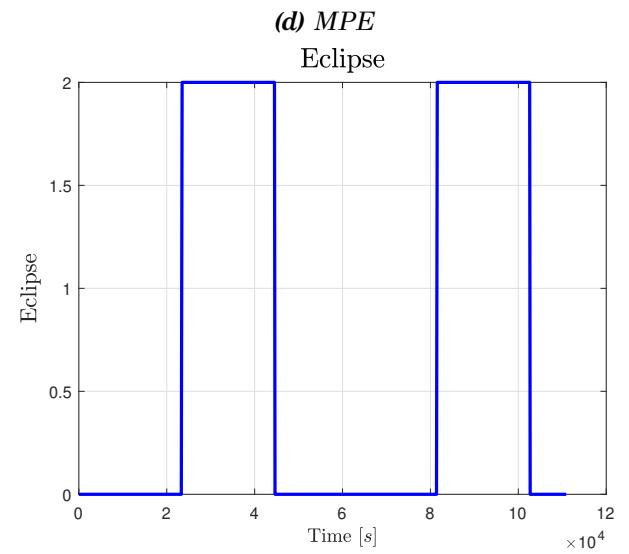
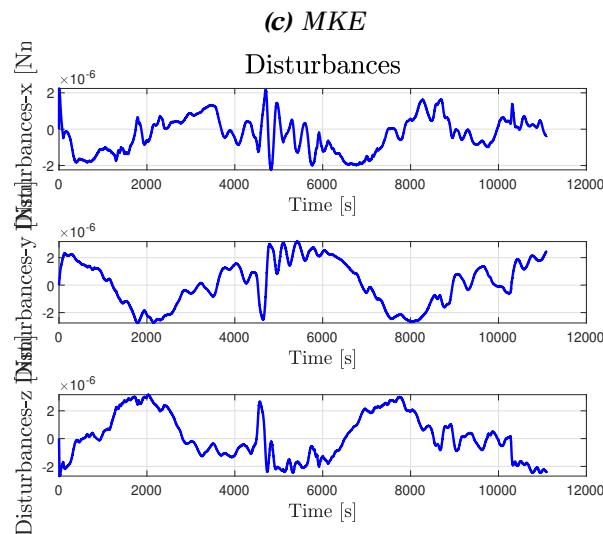
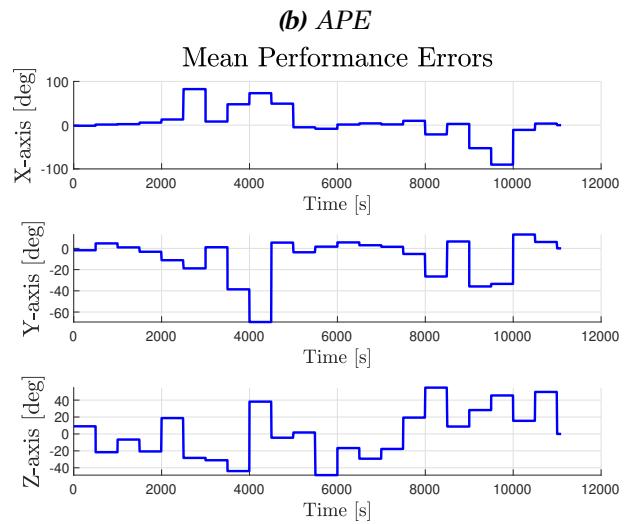
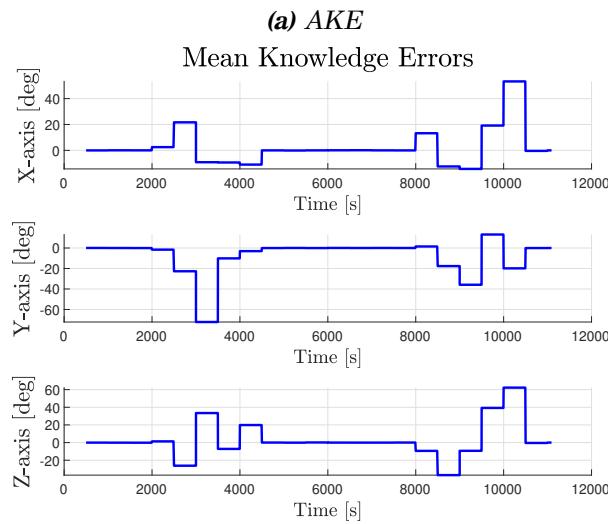
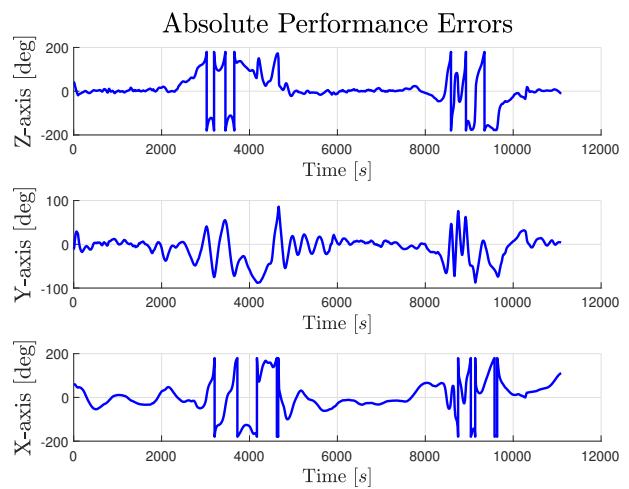
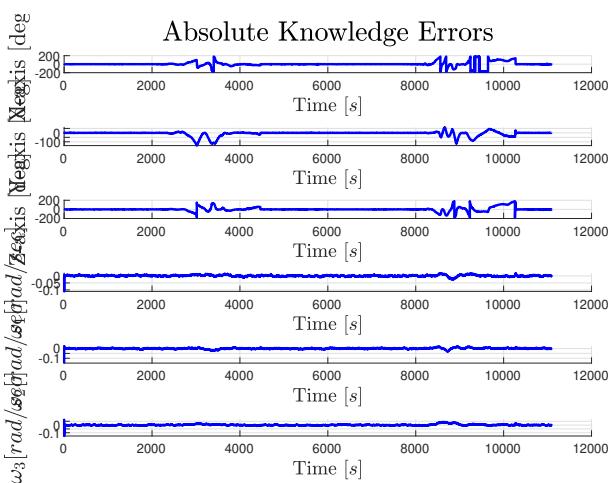


Figure 91: Nominal Mode [1]



(e) Disturbances

(f) Eclipse

Figure 92: Nominal Mode [2]



When eclipse is not present, the pointing requirements, as defined in TS-VCD, are fulfilled. During eclipse the pointing and knowledge are greatly affected.

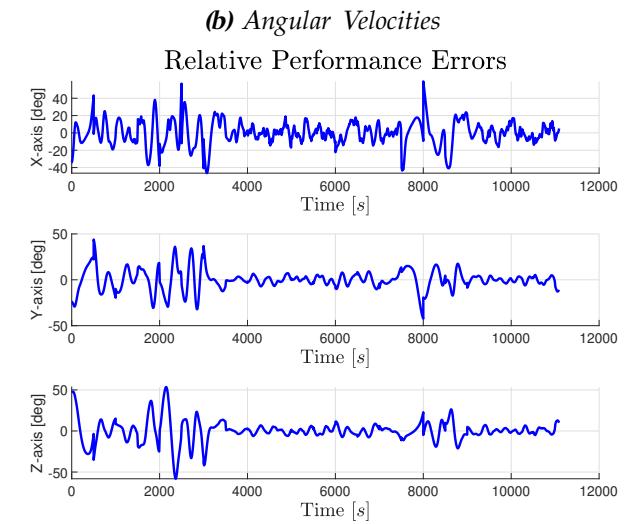
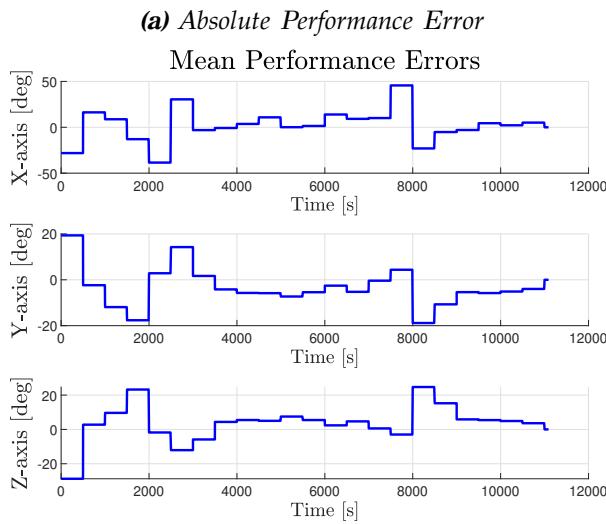
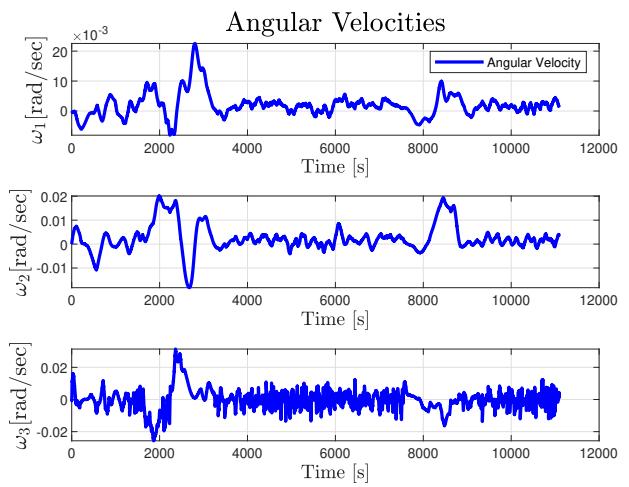
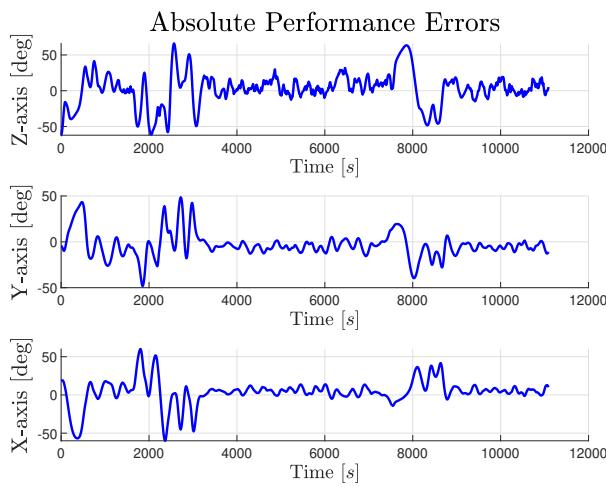
Requirements close-out — Analysis

The above document is the close out reference for requirements:

Requirement ID	Requirement Text	Status & Rationale	Reference
ADCS-FUN-030	Absolute attitude pointing accuracy shall be less than 20 deg on Pitch and Yaw axis during subsystem Nominal Mode.	Compliant at Phase C. Results from MATLAB simulation in attached figure	Fig. 92b (Page 100)
ADCS-FUN-040	Absolute attitude pointing accuracy shall be less than 20 deg on Roll axis during subsystem Nominal Mode.	Compliant at Phase C. Results from MATLAB simulation in attached figure	Fig. 92b (Page 100)
ADCS-FUN-050	Absolute angular rate shall be less than 0.13 rad/s on each axis during subsystem Nominal Mode.	Compliant at Phase C. Results from MATLAB simulation in attached figure	Fig. 91b (Page 99)
ADCS-FUN-130	The ADCS shall have a minimum on-board attitude knowledge performance of 7 deg on the X,Y and Z axes.	Compliant at Phase C. Results from MATLAB simulation in attached figure	Fig. 92a (Page 100)
SYS-FUN-060	The spacecraft shall use an attitude determination and control system to achieve a pointing accuracy within 20 degrees.	Compliant at Phase C. Results from MATLAB simulation in attached figure	Fig. 92b (Page 100)

6.4.1.3 Sun Pointing

In case the AcubeSAT's power budget fails to meet the mission requirements, a sun pointing approach needs to be implemented as shown in [Section 5.6](#). In the following plots the simulation results are presented for 11090 sec, namely 2 orbits, and with SSO TLE [[Section 5.1](#)].



(c) Mean Performance Error

(d) Relative Performance Error

Figure 93: Sun Pointing

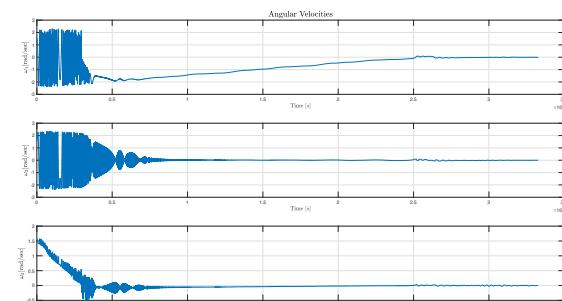
6.4.2 Worst-case scenarios

In order to demonstrate the attitude control stability for AcubeSAT, we simulated the performance of the satellite in worst case scenarios, including non-linearities (e.g. reaction wheel saturation) and variation of assumed values for most modeling and software parameters. Regarding the robustness analysis, we examine through simulations, that the performance of the satellite, in case of unit failures, namely wrong measurements and estimations from sensors, failure of the actuators in orbit. The ability of the control algorithm to detumble from unexpectedly high angular velocities is also demonstrated.

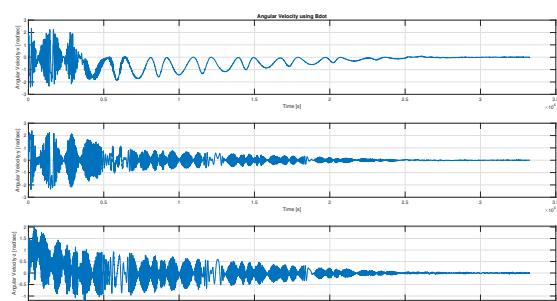
6.4.2.1 Detumbling high initial angular velocities

We performed two simulations with initial angular velocities $\begin{bmatrix} \frac{\pi}{2} & \frac{\pi}{2} & \frac{\pi}{2} \end{bmatrix}$ and $\begin{bmatrix} \frac{2\cdot\pi}{3} & \frac{2\cdot\pi}{3} & \frac{2\cdot\pi}{3} \end{bmatrix}$ correspondingly. The simulations are carried out in Total AOCS Simulation for 33270 sec, namely 6 orbits and use the SSO TLE [Section 5.1].

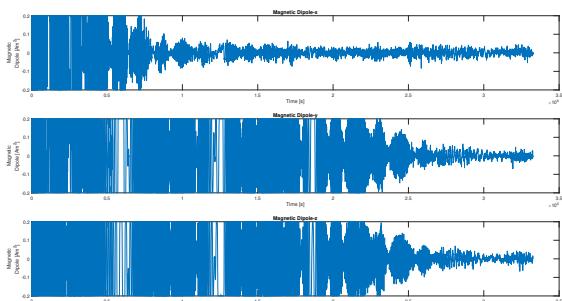
Initial Angular Velocity $\frac{\pi}{2}$ rad/sec on each axis



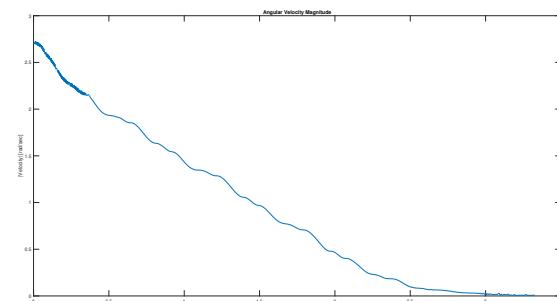
(a) Velocities



(b) Velocities approximation using B-dot metrics



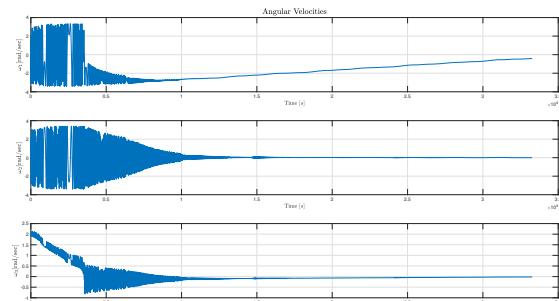
(c) Magnetic Dipole



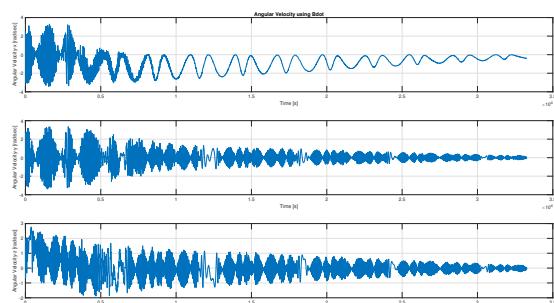
(d) Angular Velocity Magnitude

Figure 94: Detumbling from $\pi/2$ rad/sec on each axis initially

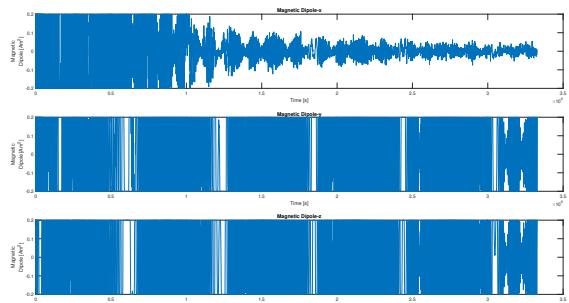
Initial Angular Velocity $\frac{2\pi}{3}$ rad/sec on each axis



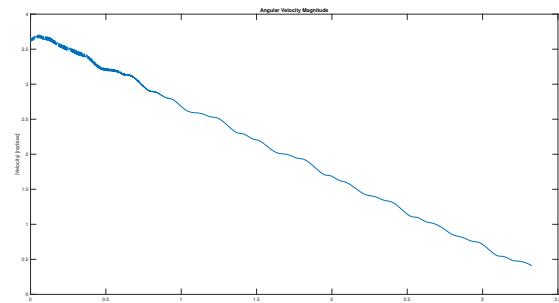
(a) Velocities



(b) Velocities approximation using B-dot metrics



(c) Magnetic Dipole



(d) Angular Velocity Magnitude

Figure 95: Detumbling from $\frac{2\pi}{3}$ rad/sec on each axis initially

We can observe that B-dot controller can detumble the spacecraft within 6 orbits maximum, when initially the angular velocity on each axis is high.

6.4.2.2 Unrealistic eclipse duration

In order to examine the performance of the MEKF while in eclipse, we used in the total simulation with SSO TLE [Section 5.1] higher, unrealistic eclipse durations.

Using eclipse duration of 3000 seconds for SSO 600km - 11PM orbit:

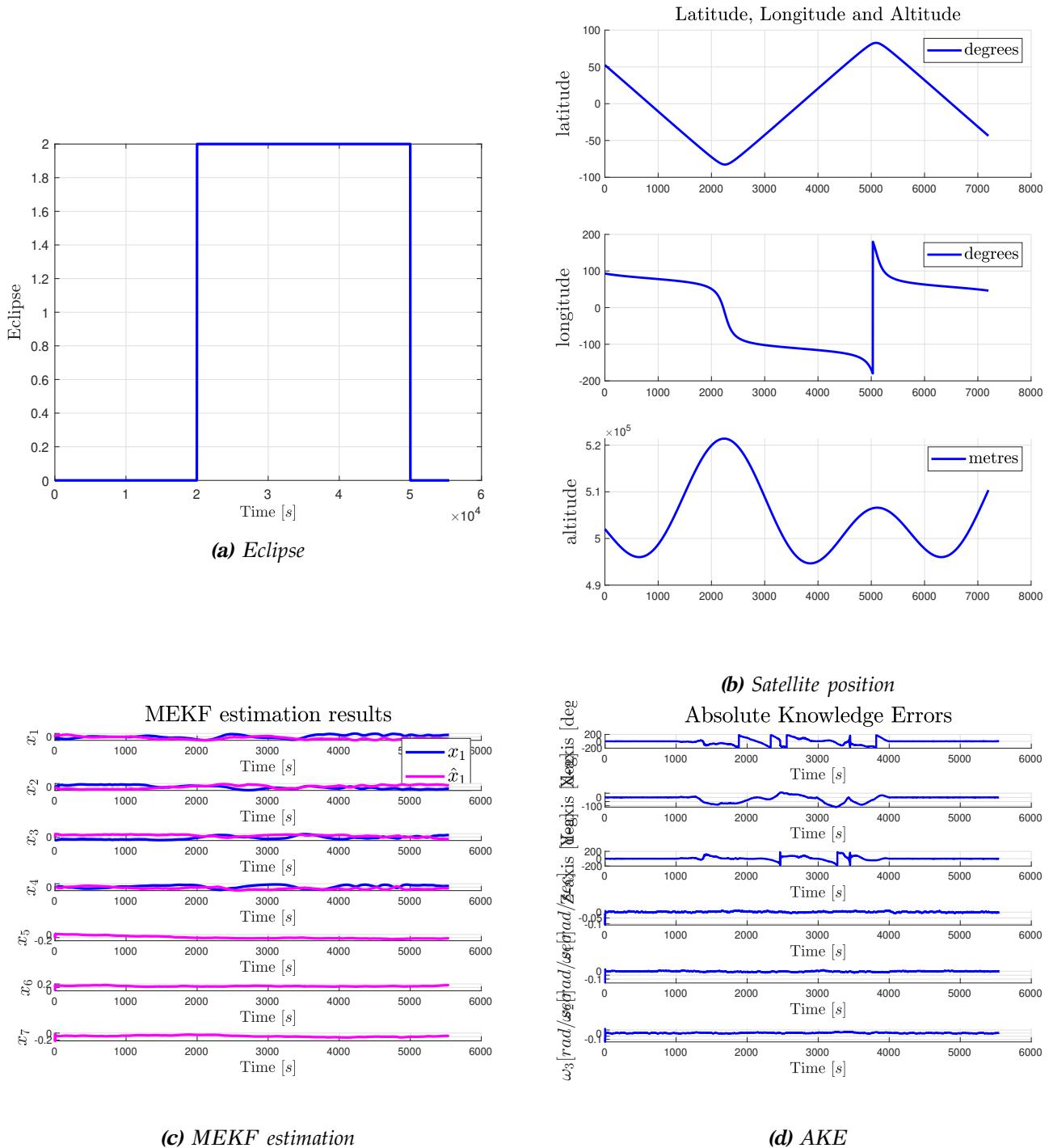
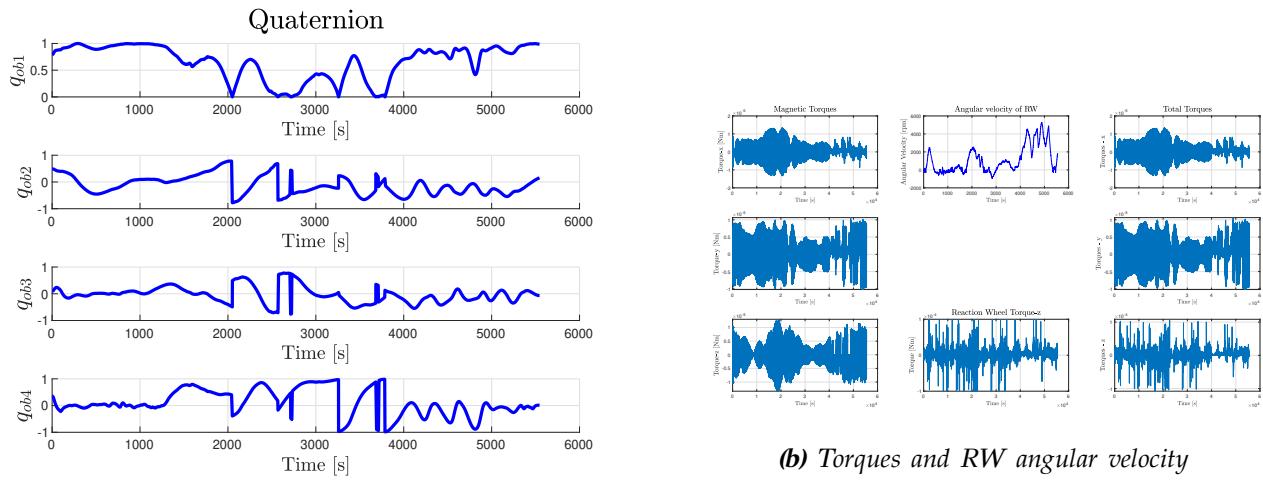
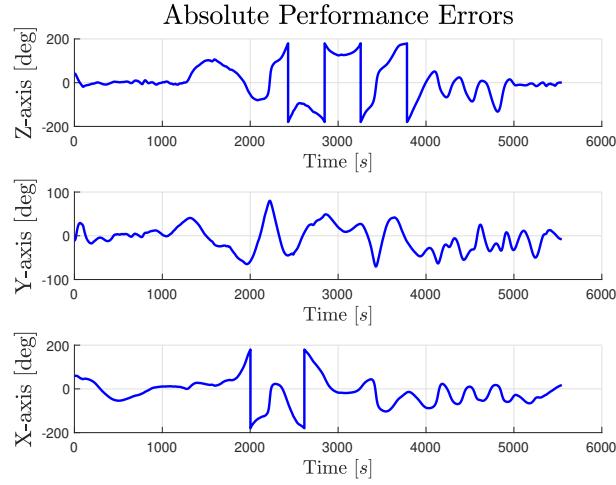


Figure 96: Unrealistic Eclipse - SSO orbit [1]



(a) Pointing quaternion

Figure 97: Unrealistic Eclipse - SSO orbit [2]



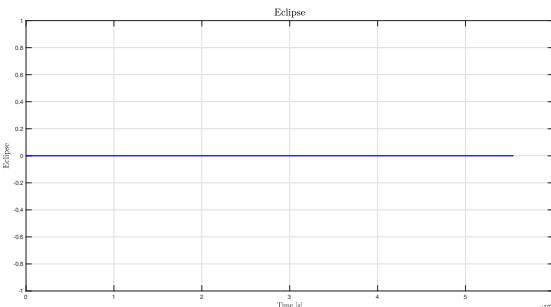
(a) APE

Figure 98: Unrealistic Eclipse - SSO orbit [3]

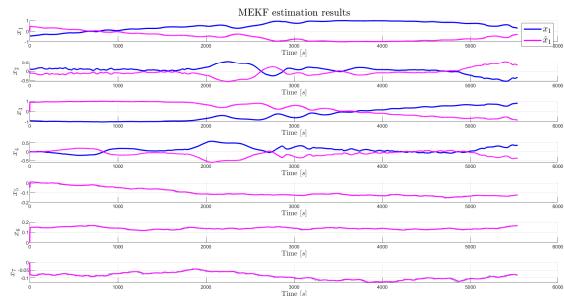
Conclusion

In the simulation [Figure 96](#), [Figure 97](#) and [Figure 98](#) we can observe that even in an unrealistic eclipse duration scenario, both the pointing error [Fig. 96d](#) and estimation [Fig. 98a](#) have relatively small error outside the eclipse and are considered sufficient.

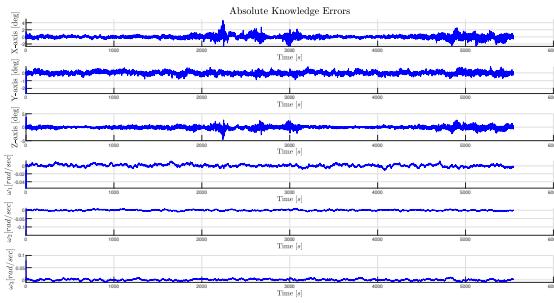
6.4.2.3 Random estimation spikes in MEKF



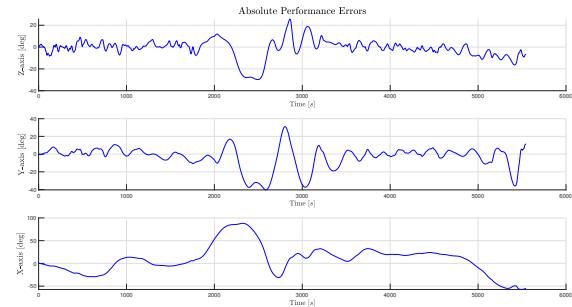
(a) Eclipse



(b) State estimation



(c) AKE



(d) APE

Figure 99: Random estimation spikes [1]

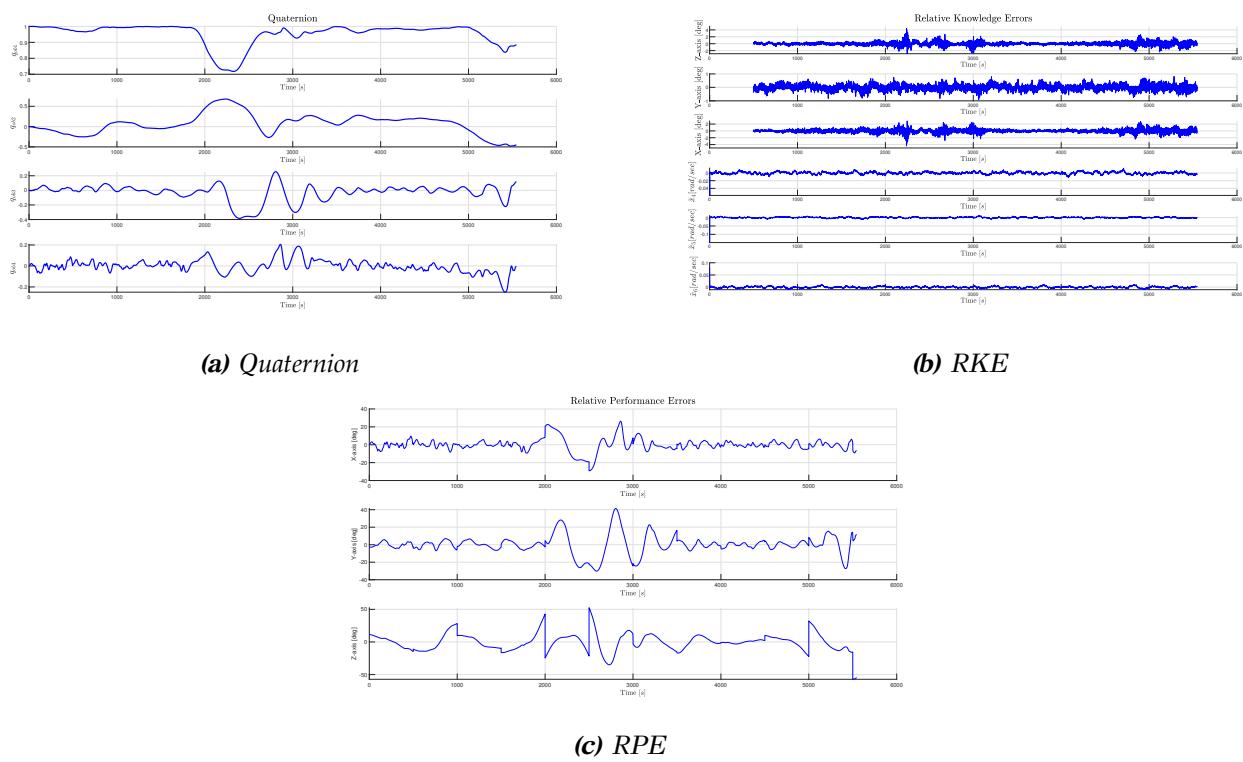
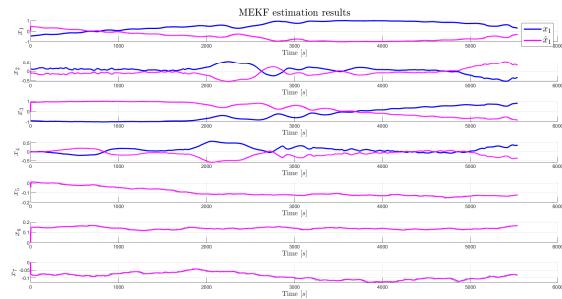


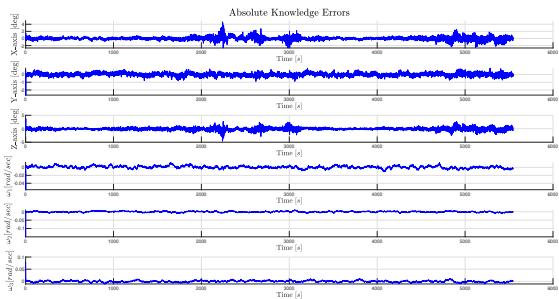
Figure 100: Random estimation spikes [2]

For the simulation, we added 100 spikes in MEKF every 500 seconds. In [Figure 99](#) and [Figure 100](#), we can observe that even with this high frequency of the spikes, MEKF converges with high accuracy to the real value and the absolute knowledge error is minimized in a relatively small time interval.

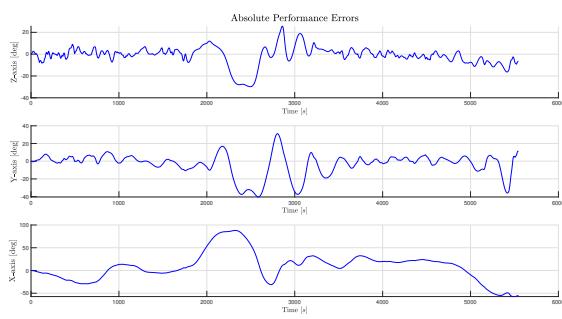
6.4.2.4 Random measurement spikes from sensors



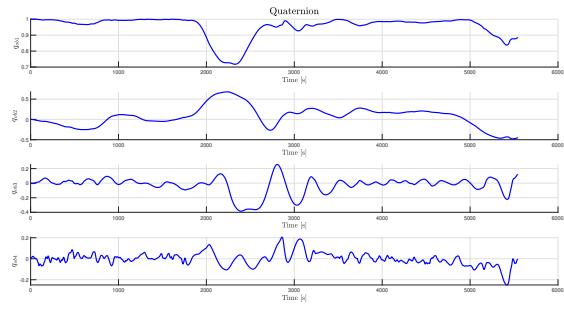
(a) State estimation



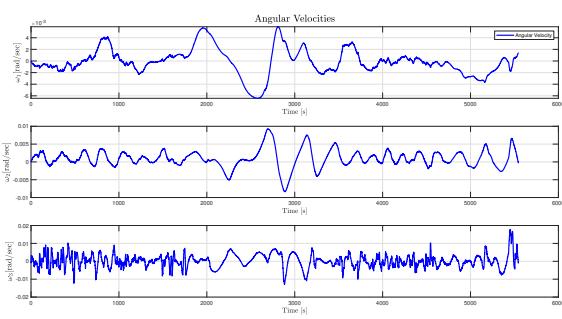
(b) AKE



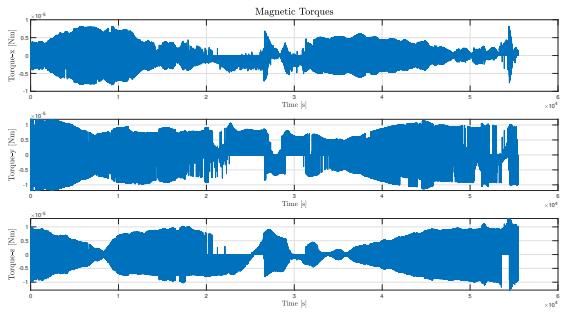
(c) APE



(d) Quaternion



(e) Velocities



(f) Magnetic Torques

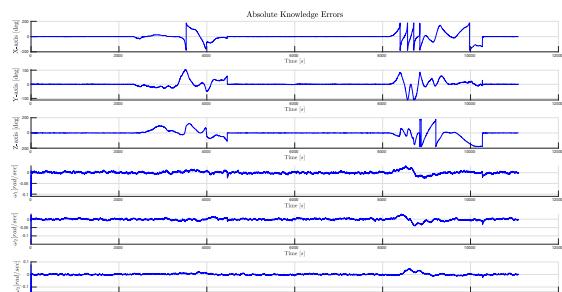
Figure 101: Random measurement spikes

For this simulation, our sensors were assigned to random measurements for 10 seconds every 500 seconds. In Figure 101, we can again observe that MEKF converges with high accuracy to the real value and the pointing error is minimized in relatively small time interval.

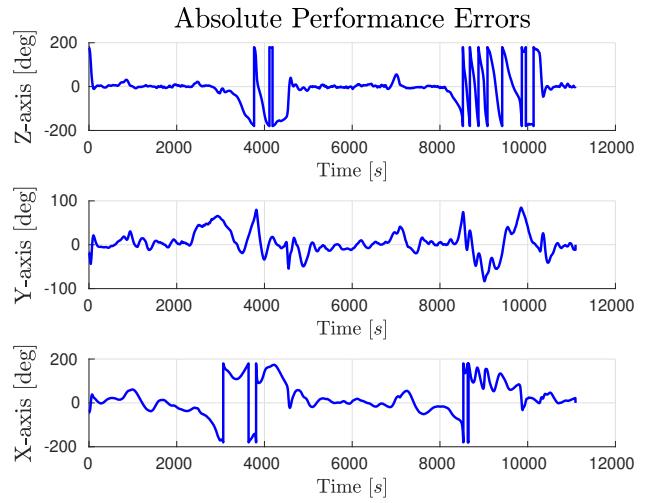
6.4.2.5 Desired quaternion change

Normally for the mission's requirements the desired quaternion would be the identity quaternion $\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$. Here, the possibility for a different satellite orientation is being examined. In order for the implemented PD controller, as presented in Section 5.6,

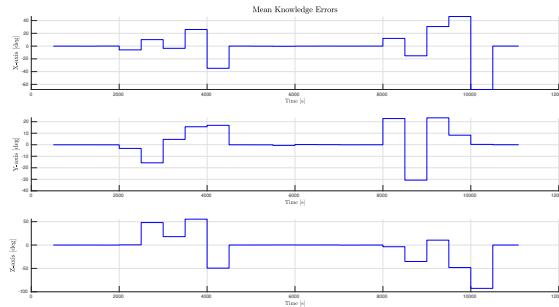
to reach a quaternion other than the identity.



(a) AKE

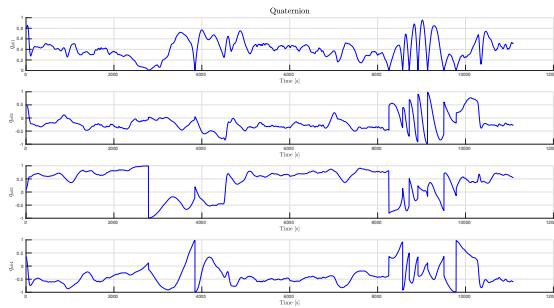


(b) APE



(c) MKE

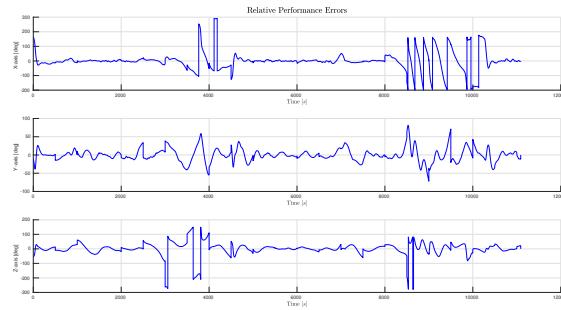
$$\text{Figure 102: } Q_{\text{desired}} = \begin{bmatrix} 0.4 & 0.3 & -0.7 & 0.5 \end{bmatrix} - \text{SSO}[1]$$



(a) Quaternion



(b) MEKF State Estimation



(c) RPE

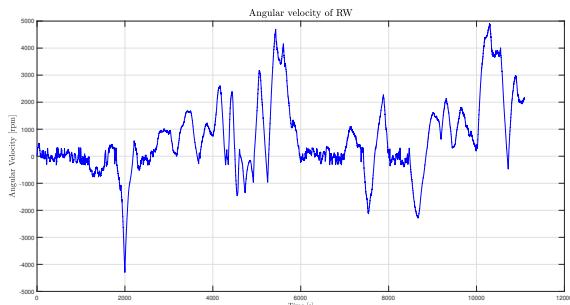
$$\text{Figure 103: } Q_{\text{desired}} = \begin{bmatrix} 0.4 & 0.3 & -0.7 & 0.5 \end{bmatrix} - \text{SSO}[2]$$

As evident by [Figure 102](#) and [Figure 103](#) both the determination and control accuracy is considered sufficient.

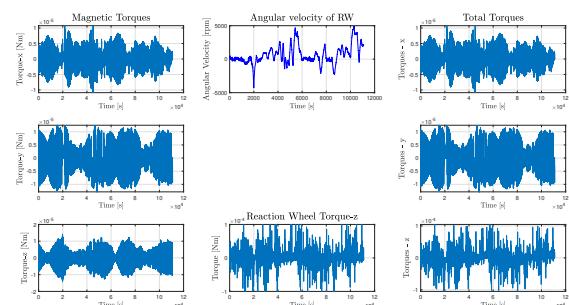
6.4.2.6 Variation of Deadzone Limits

In order to verify the correctness of the reaction wheel dead-zone modeling, we include in the simulations various limits of the deadzone, as mentioned below, and demonstrate the behaviour of the reaction wheel and the controller.

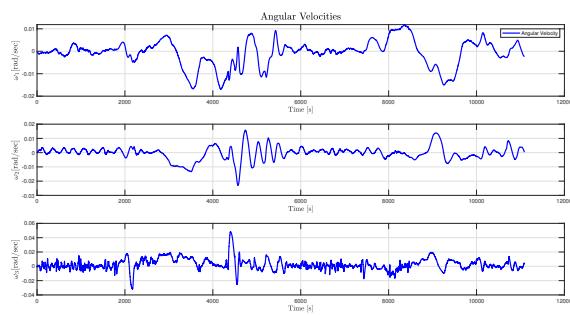
Using Deadzone Limit = ± 300 rpm:



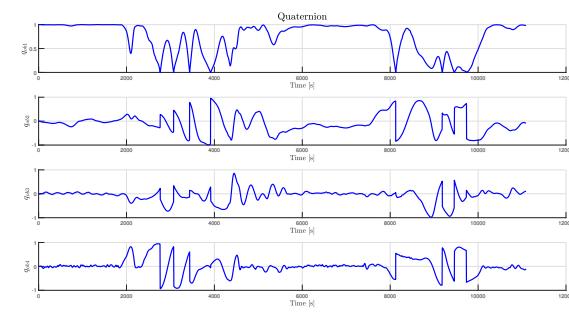
(a) RW Angular Velocity



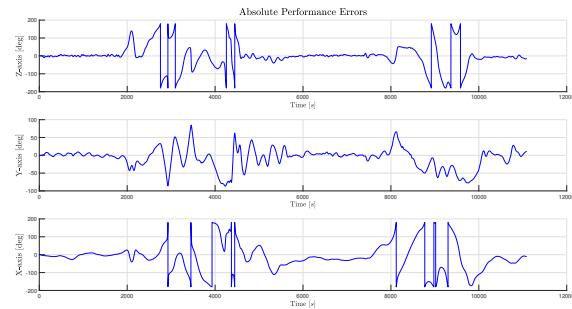
(b) Torques



(c) Velocities



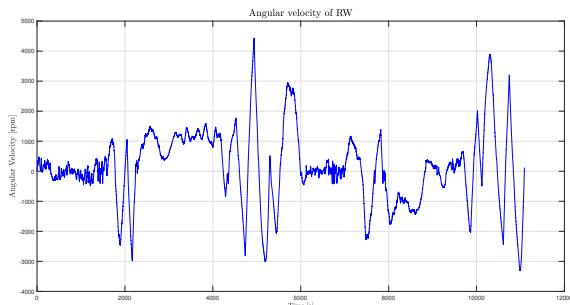
(d) Quaternion



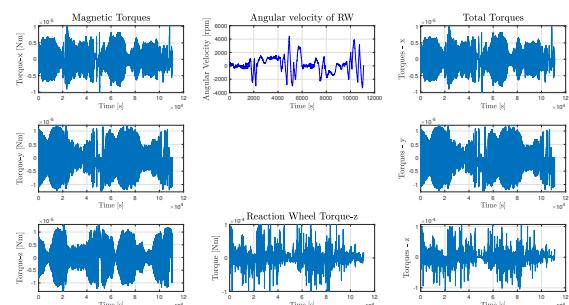
(e) APE

Figure 104: Deadzone Limit = ± 300 rpm

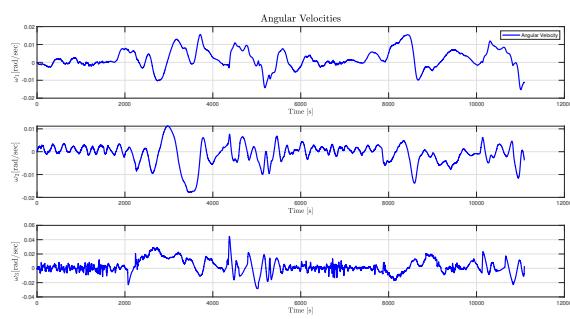
Using Deadzone Limit = ± 400 rpm:



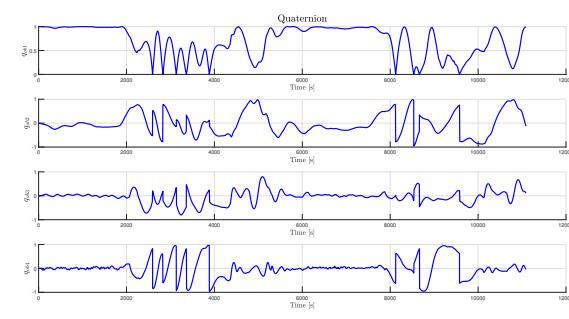
(a) RW Angular Velocity



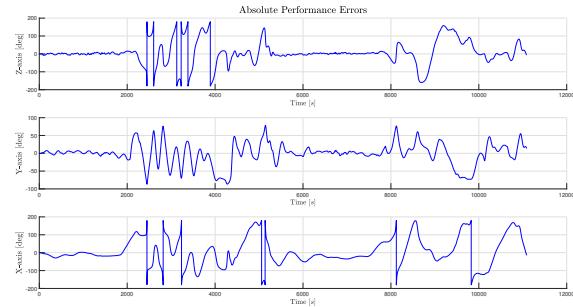
(b) Torques



(c) Velocities



(d) Quaternion



(e) APE

Figure 105: Deadzone Limit = ± 400 rpm

Using Deadzone Limit = ± 500 rpm:

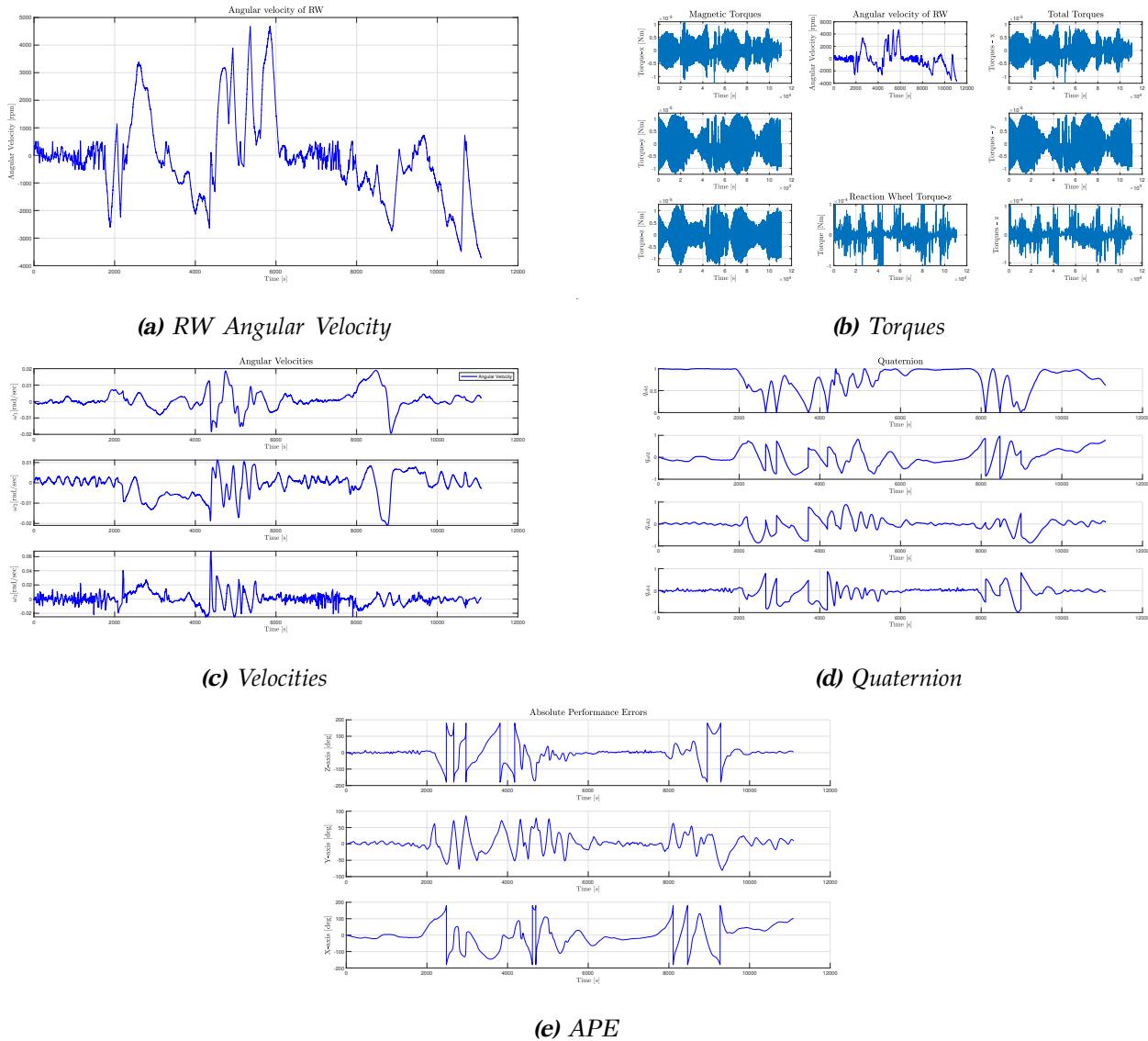


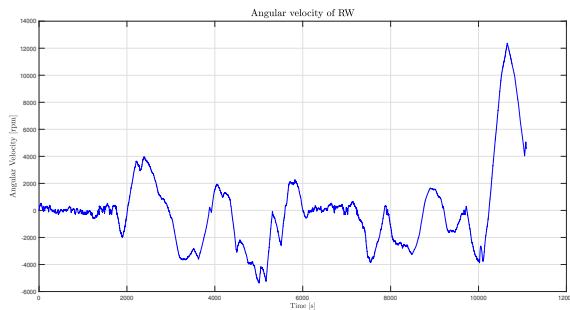
Figure 106: Deadzone Limit = ± 500 rpm

From Figure 104, Figure 105 and Figure 106, we can observe that the behaviour during dead-zone does not affect the pointing accuracy.

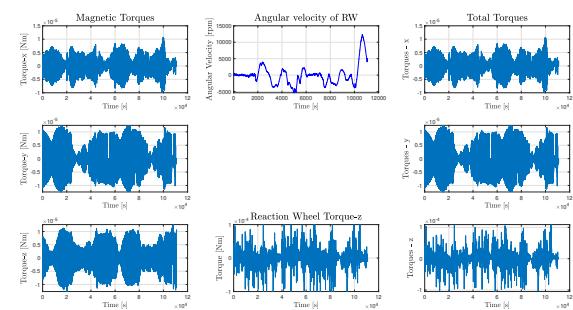
6.4.2.7 Variation of Reaction Wheel Frictions

Since the frictions present in the reaction wheel cannot be specified until the acquisition of the component, we demonstrate the performance of the wheel, using different friction values.

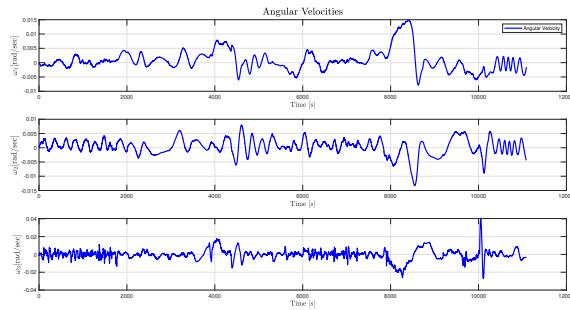
Using $b_{\text{friction}} = 1.9 \cdot 10^{-10} \text{ Nms}$, $c_{\text{friction}} = 1.9 \cdot 10^{-7} \text{ Nm}$ and deadzone limit = $\pm 300 \text{ rpm}$:



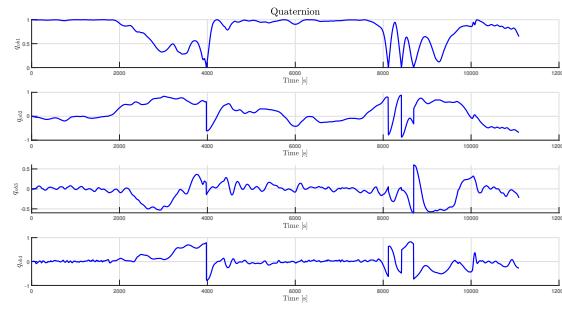
(a) RW Angular Velocity



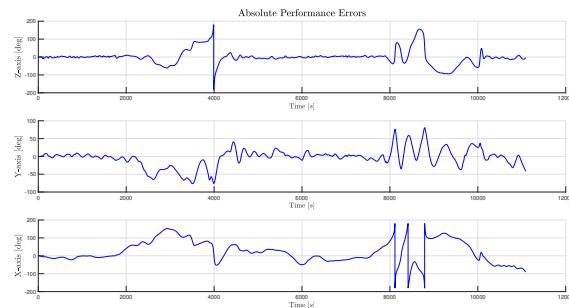
(b) Torques



(c) Velocities



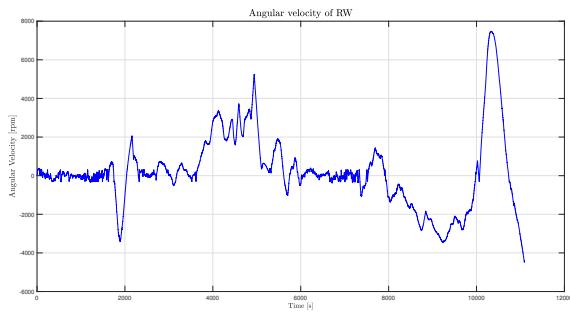
(d) Quaternion



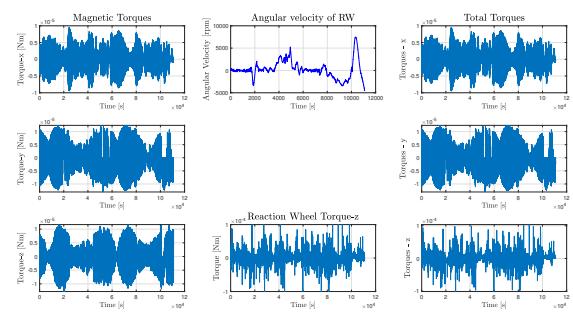
(e) APE

Figure 107: $b_{\text{friction}} = 1.9 \cdot 10^{-10} \text{ Nms}$, $c_{\text{friction}} = 1.9 \cdot 10^{-7} \text{ Nm}$

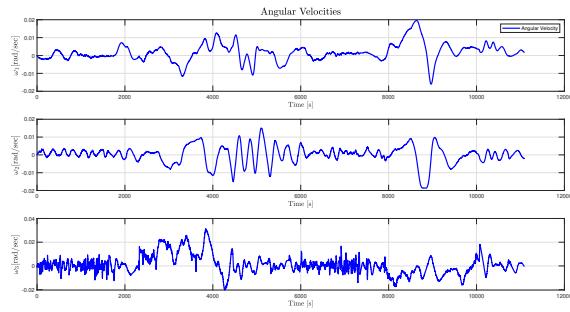
Using $b_{\text{friction}} = 1.9 \cdot 10^{-9} \text{ Nms}$, $c_{\text{friction}} = 9.5 \cdot 10^{-7} \text{ Nm}$ and deadzone limit = $\pm 300 \text{ rpm}$:



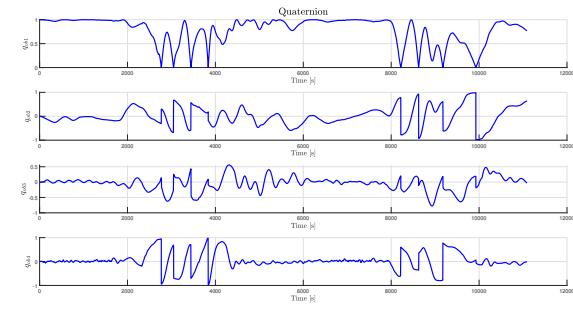
(a) RW Angular Velocity



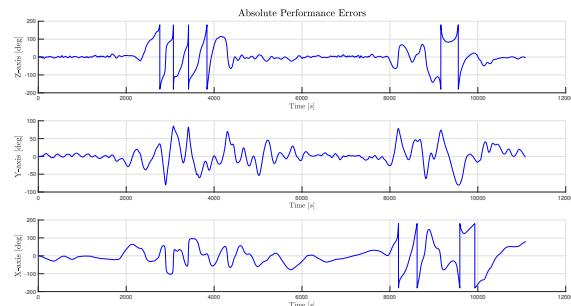
(b) Torques



(c) Velocities



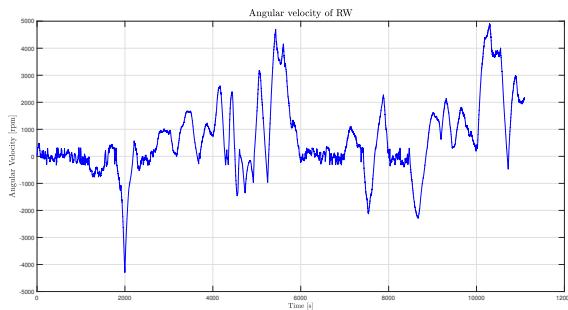
(d) Quaternion



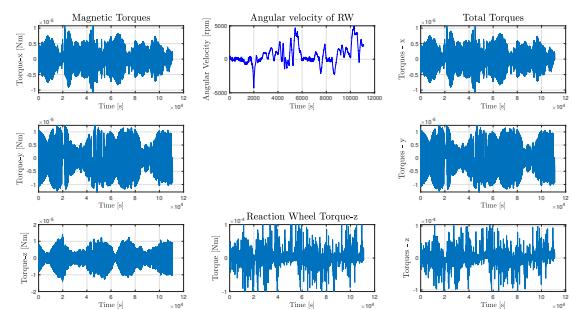
(e) APE

Figure 108: $b_{\text{friction}} = 1.9 \cdot 10^{-9} \text{ Nms}$, $c_{\text{friction}} = 9.5 \cdot 10^{-7} \text{ Nm}$

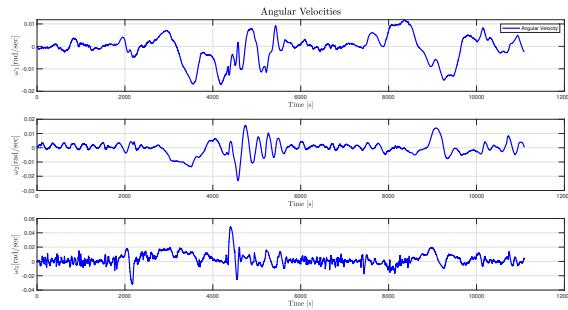
Using $b_{\text{friction}} = 9.5 \cdot 10^{-9} \text{ Nms}$, $c_{\text{friction}} = 1.9 \cdot 10^{-7} \text{ Nm}$ and deadzone limit = $\pm 300 \text{ rpm}$:



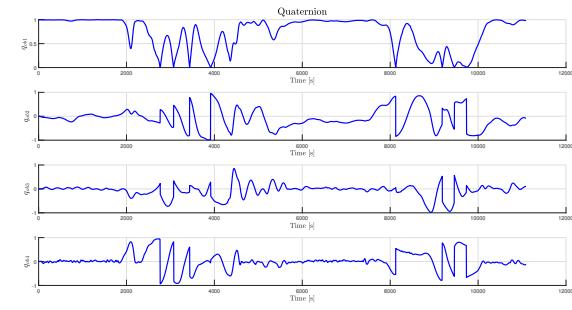
(a) RW Angular Velocity



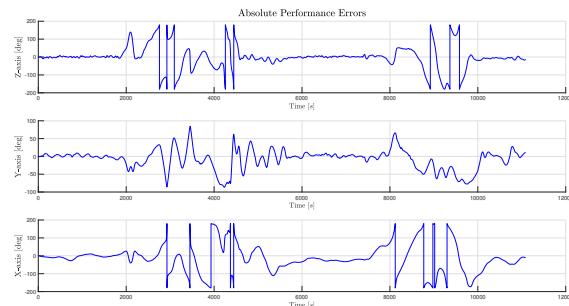
(b) Torques



(c) Velocities



(d) Quaternion



(e) APE

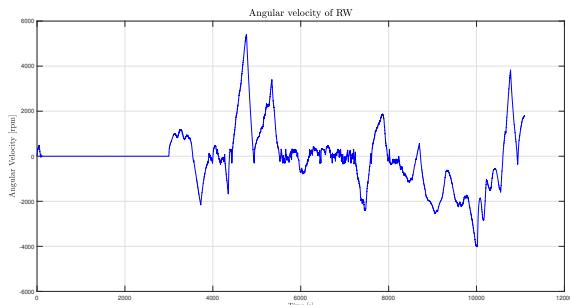
Figure 109: $b_{\text{friction}} = 9.5 \cdot 10^{-9} \text{ Nms}$, $c_{\text{friction}} = 1.9 \cdot 10^{-7} \text{ Nm}$

From Figure 109, Figure 107 and Figure 108, we can observe that under varied friction values the behaviour of the reaction wheel does not affect the pointing accuracy.

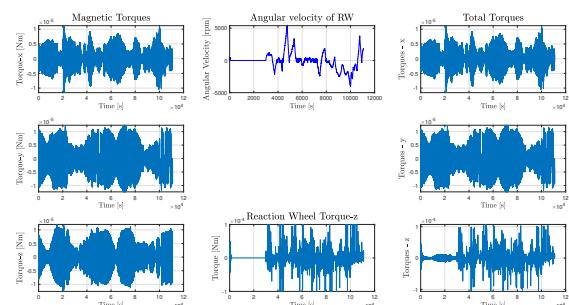
6.4.2.8 Failure of Reaction Wheel during a Time Interval

In order to demonstrate a contemporary failure of the reaction wheel, we disable it for various time intervals. The simulations are carried out for a total time of 11090 sec, namely 2 orbits, and SSO TLE [Section 5.1].

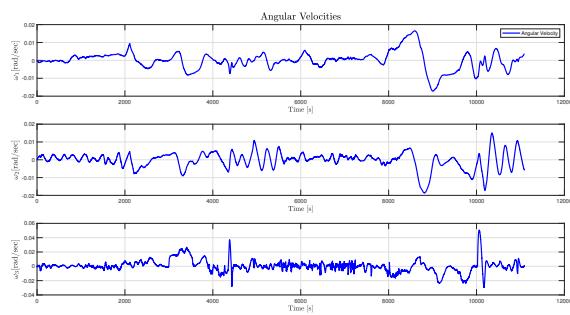
Using Time Interval = [100 - 3000] sec:



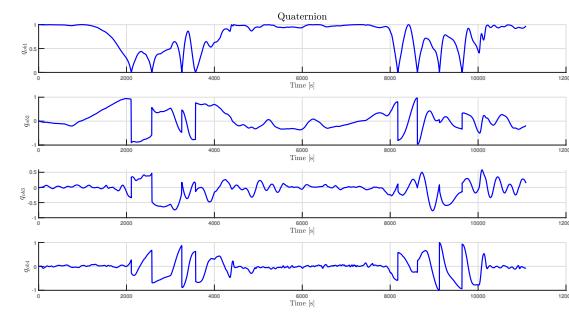
(a) RW Angular Velocity



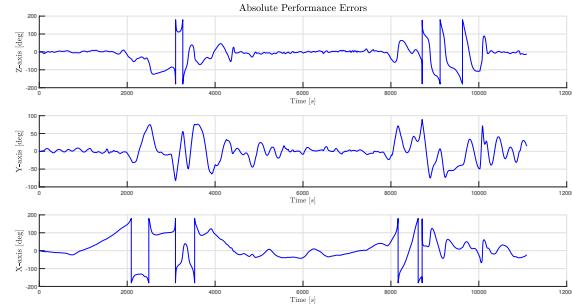
(b) Torques



(c) Velocities



(d) Quaternion



(e) APE

Figure 110: RW_failure (Time Interval = [100 - 3000] sec)

Using Time Interval = [2000 - 5000] sec:

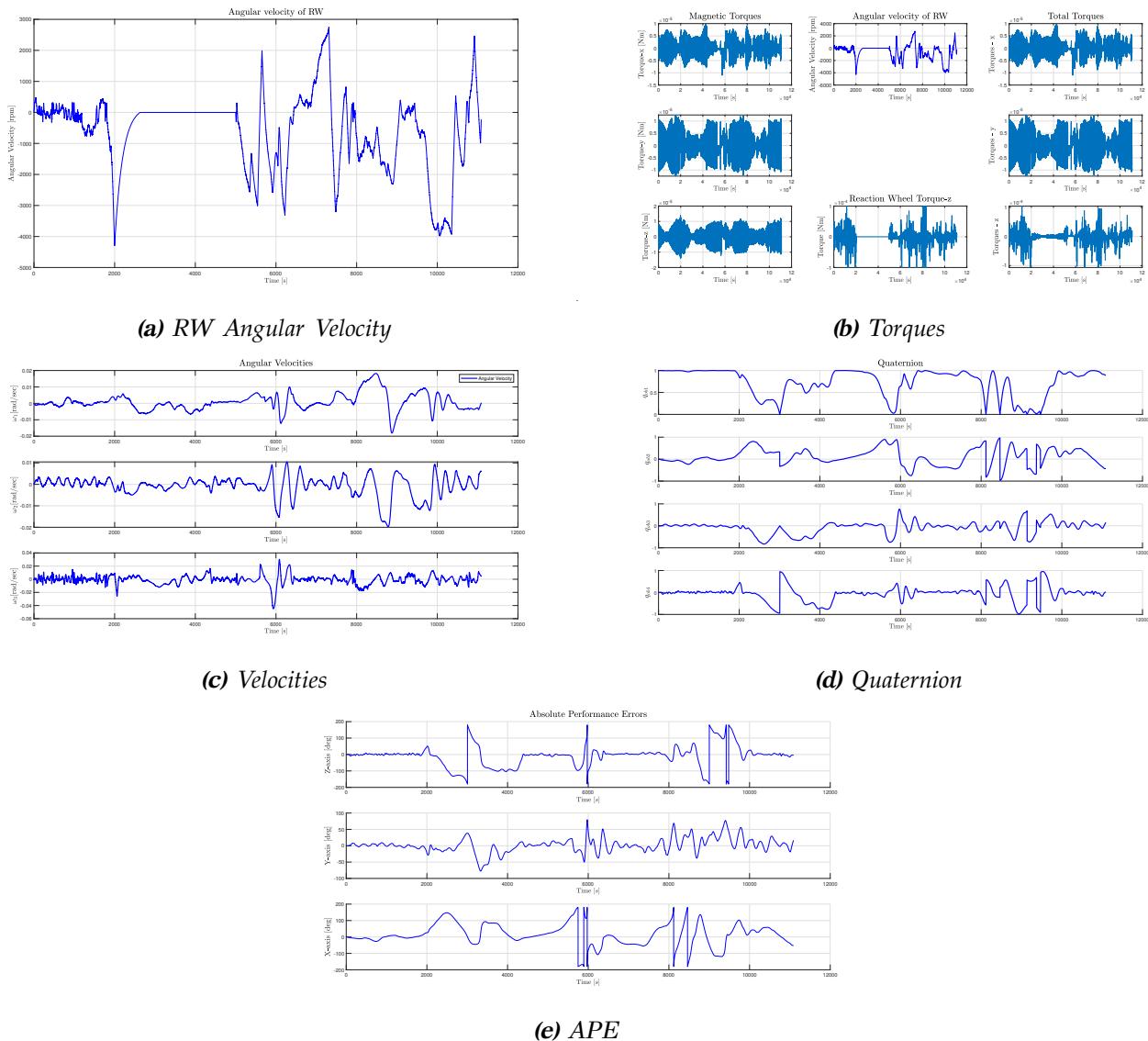


Figure 111: RW_failure (Time Interval = [2000 - 5000] sec)

Using Time Interval = [1000 - 6000] sec:

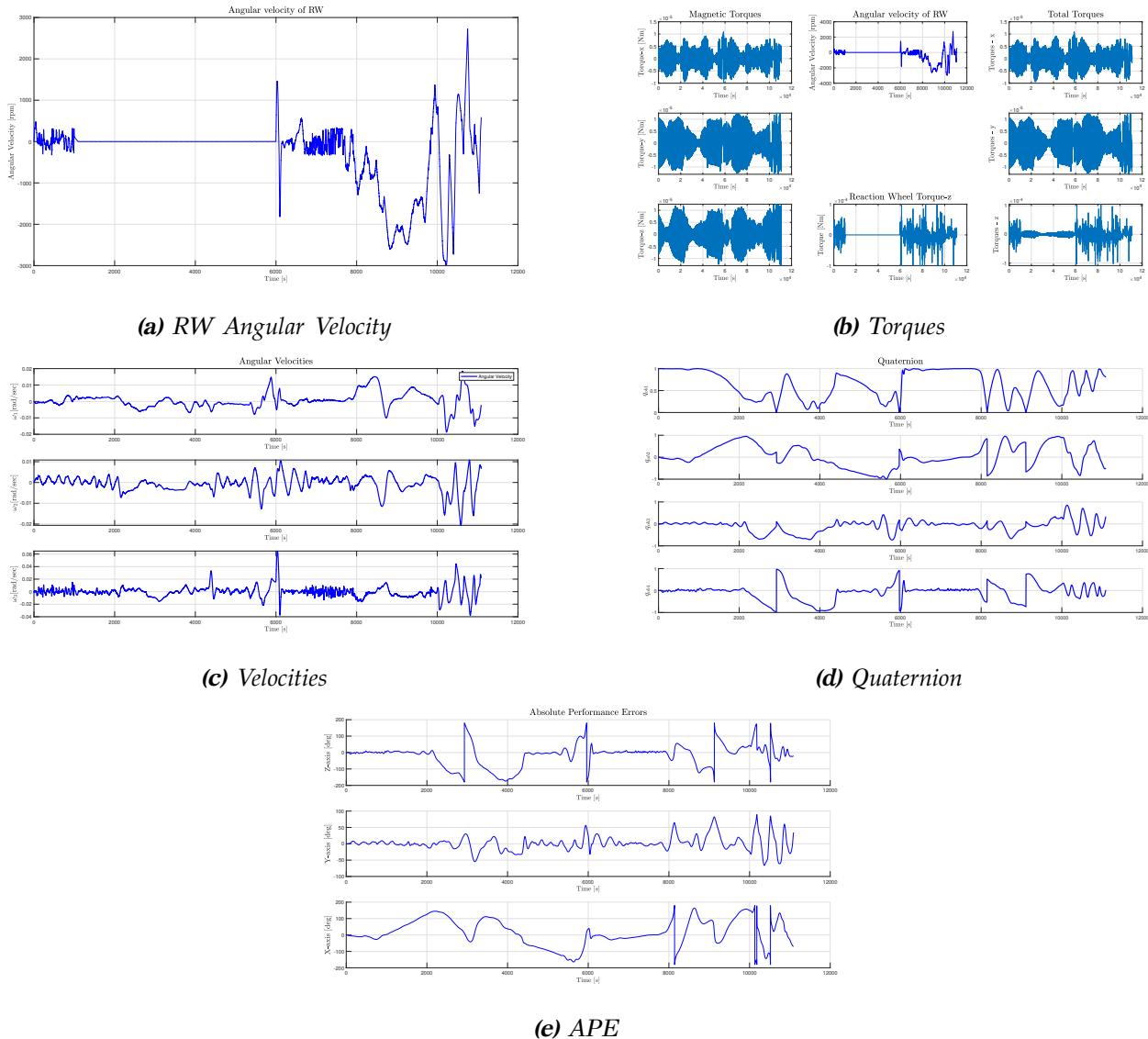


Figure 112: RW_failure (Time Interval = [1000 - 6000] sec)

From [Figure 110](#), [Figure 111](#) and [Figure 112](#), we can observe that, in case of temporary failure of the reaction wheel during nominal mode, although the pointing accuracy of the satellite is affected during the time of the failure, pointing towards nadir is achieved finally -with some errors included.

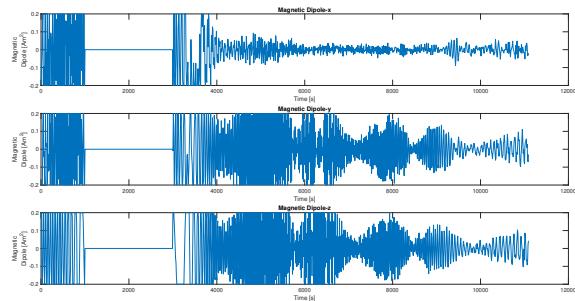
6.4.2.9 Failure of Magnetorquers during a Time Interval

In order to demonstrate a contemporary failure of the magnetorquers, we disable them for various time intervals. The simulations are both carried out using SSO orbits for a total time of 11090 sec, namely 2 orbits and 22180 sec, namely 4 orbits, accordingly, in

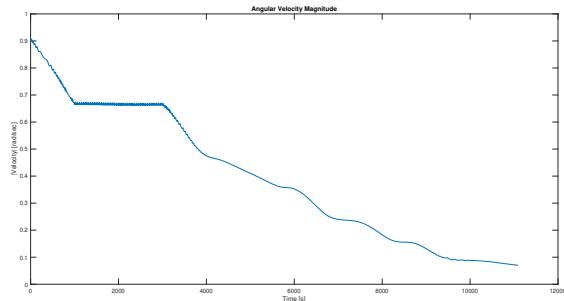
B-dot Simulation. For both simulations the SSO TLE [Section 5.1] is used.

B-dot Simulation

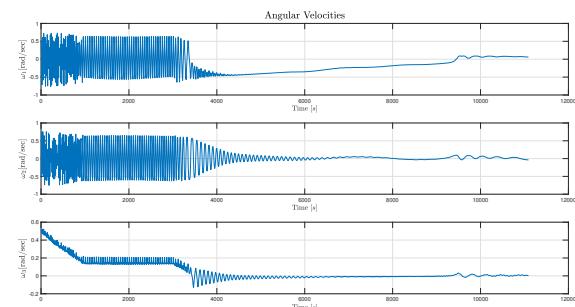
Using Time Interval = [1000 - 3000] sec:



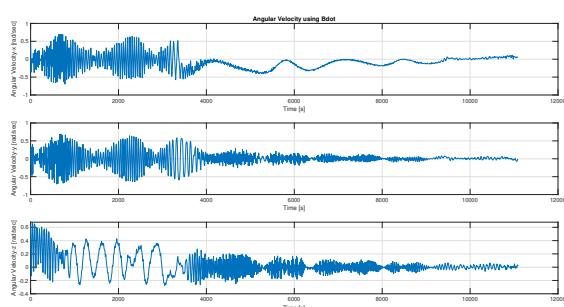
(a) Magnetic Dipole



(b) Angular Velocity Magnitude



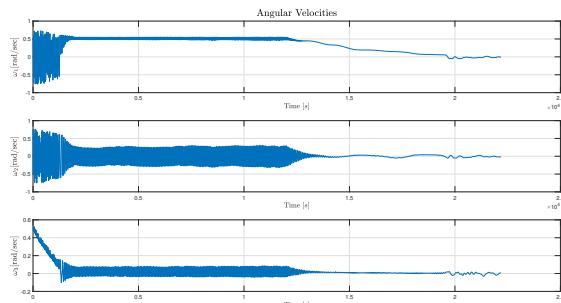
(c) Velocities



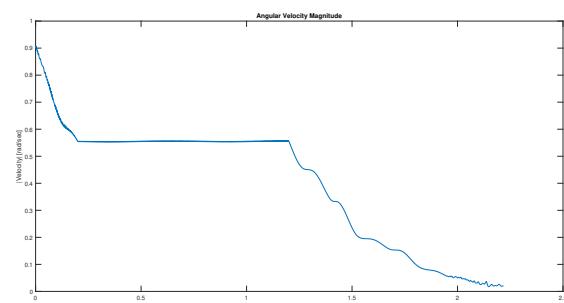
(d) Velocities approximation using B-dot metrics

Figure 113: MTQ_failure during AOCS Detumbling Mode (Time Interval = [1000 - 3000] sec)

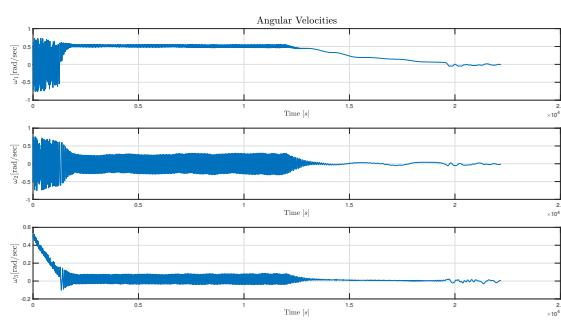
Using Time Interval = [2000 - 12000] sec:



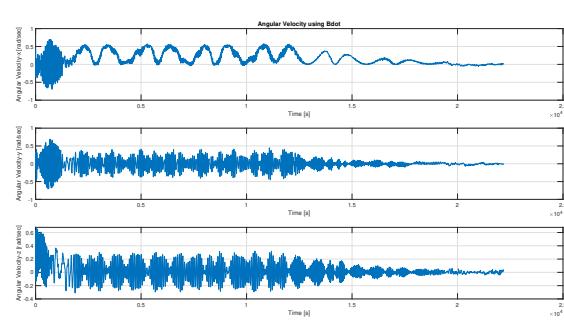
(a) Magnetic Dipole



(b) Angular Velocity Magnitude



(c) Velocities

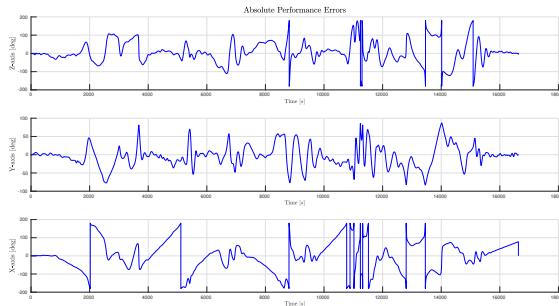


(d) Velocities approximation using B-dot metrics

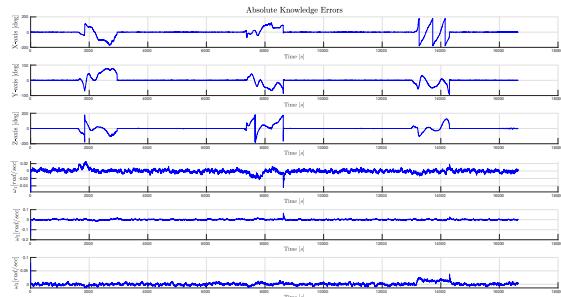
Figure 114: MTQ_failure during AOCS Detumbling Mode (Time Interval = [1000 - 5000] sec)

From [Figure 113](#) and [Figure 114](#), we can observe that, in case of temporary failure of the magnetorquers during AOCS nominal and detumbling mode, although the attitude of the satellite is affected during the time of the failure, the final attitude remains almost intact.

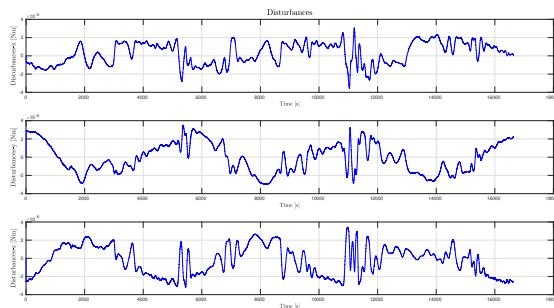
6.4.2.10 Nominal mode using only magnetic actuation (N-02)



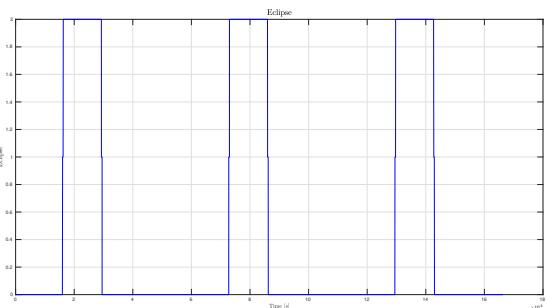
(a) Absolute Performance Errors



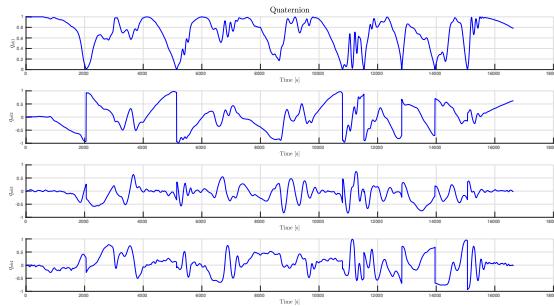
(b) Absolute Knowledge Errors



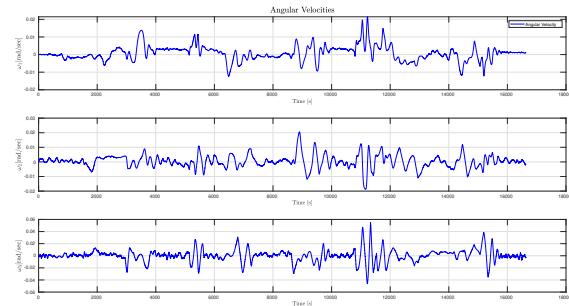
(c) Disturbances



(d) Eclipse



(e) Magnetic Torques



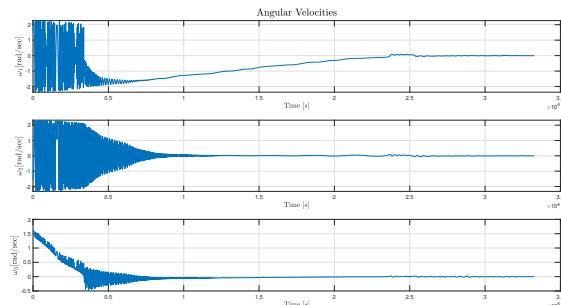
(f) Angular Velocity

As shown in Figure 115a, Figure 115e and Figure 115f, the pointing accuracy of the satellite is highly affected when utilizing only magnetic actuation. However, nadir-pointing is achieved during limited time intervals outside eclipse.

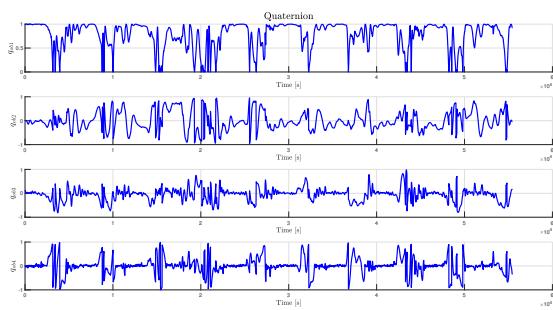
6.4.3 Complete Simulations

In this section we present a Total AOCS Simulation for 2 TLEs. Detumbling Mode starts from an initial angular velocity of $\left[\frac{\pi}{2} \quad -\frac{\pi}{2} \quad \frac{\pi}{2}\right]$, while Nominal Mode starts from the moment the system exits Detumbling Mode and lasts for 10 orbits.

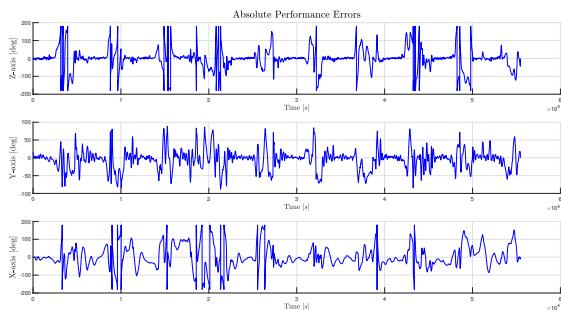
SS0-500-11PM.TLE



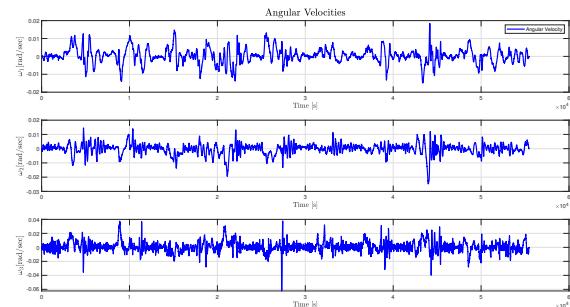
(a) Detumbling Velocities



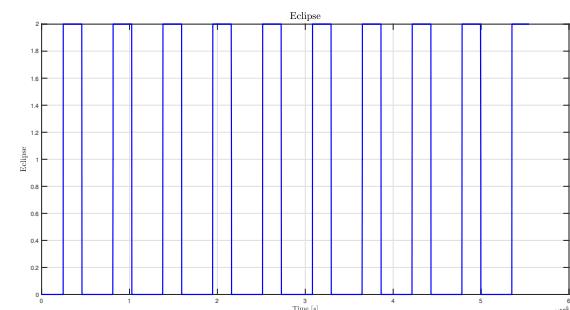
(c) Quaternion



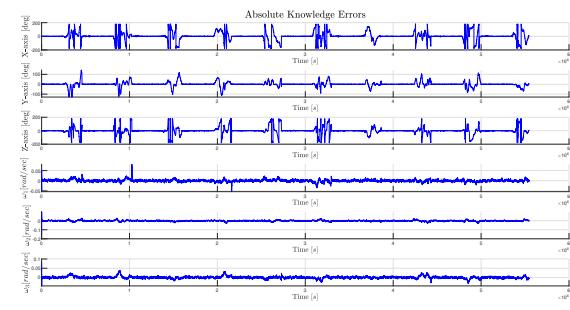
(e) Absolute Performance Error



(b) Nominal Velocities

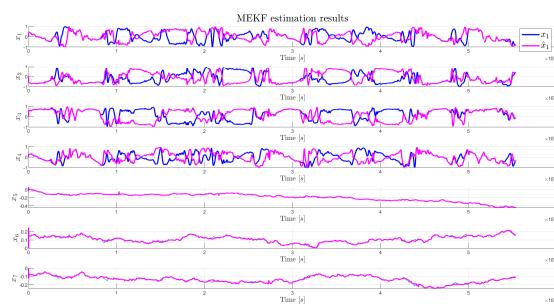


(d) Eclipse

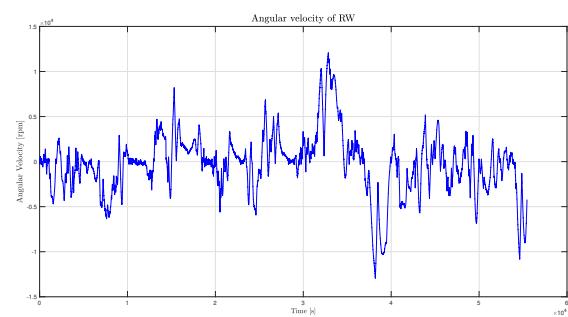


(f) Absolute Knowledge Error

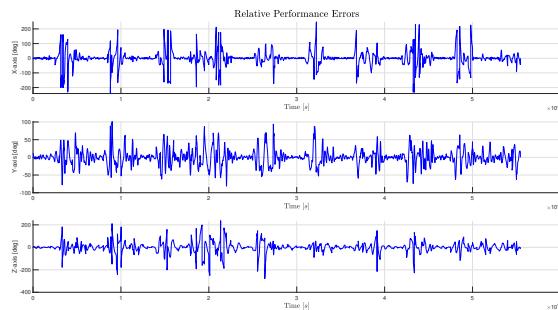
Figure 116: Nominal Scenario 500 km LTAN 11 [1]



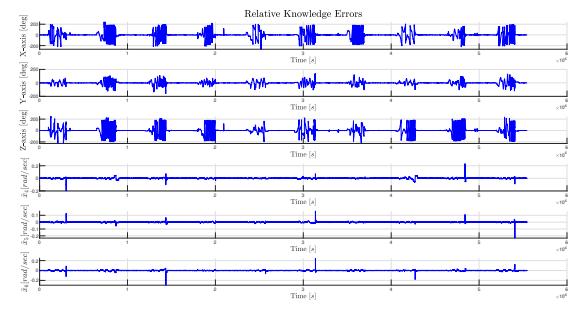
(a) MEKF Estimation Results



(b) RW Angular Velocity



(c) Relative Performance Error



(d) Relative Knowledge Error

Figure 117: Nominal Scenario 500 km LTAN 11 [2]

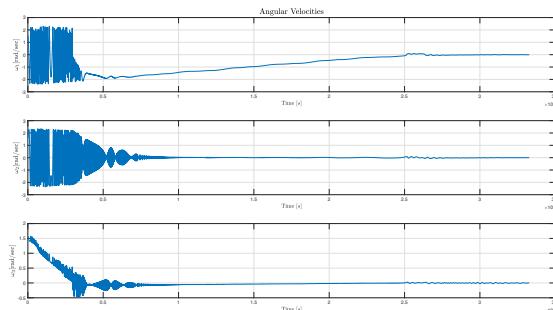
Additionally, 95% confidence intervals for the median Absolute Performance Error in each axis is calculated, as defined in the error metrics [Section 4.1](#).

	Z	Y	X
Eclipse included	[9.0845, 9.2429]	[9.8582, 9.9678]	[31.6750, 31.8750]
Eclipse not included	[3.4068, 3.4501]	[4.4360, 4.4837]	[24.3692, 24.5879]

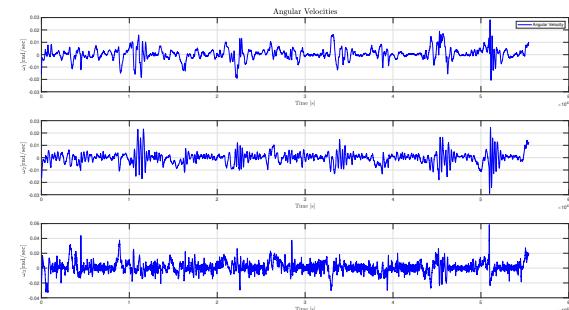
	Percentage of time inside COMMS requirements
Eclipse included	[51.27%]
Eclipse not included	[78.9%]



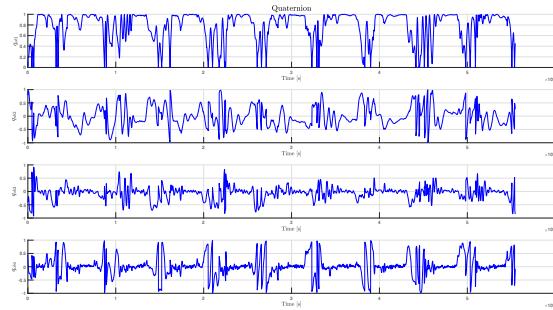
SS0-600-11PM.TLE



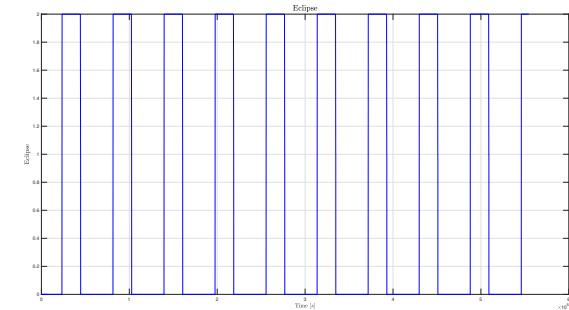
(a) Detumbling Velocities



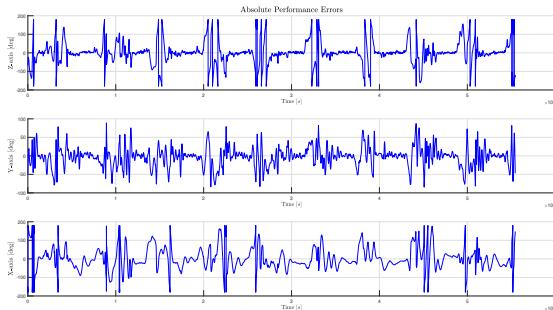
(b) Nominal Velocities



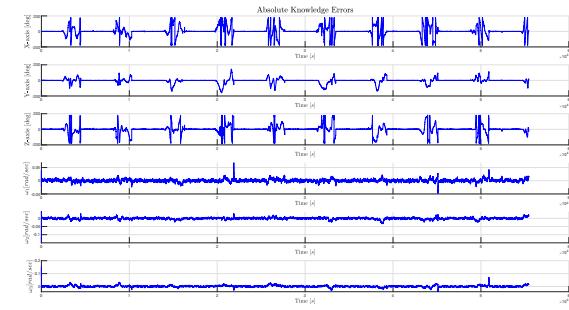
(c) Quaternion



(d) Eclipse

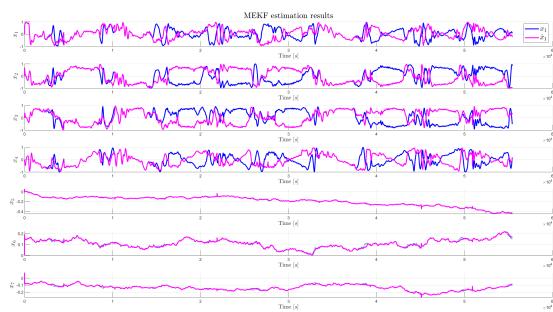


(e) Absolute Performance Error

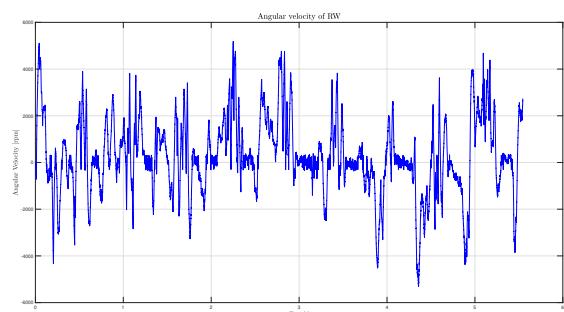


(f) Absolute Knowledge Error

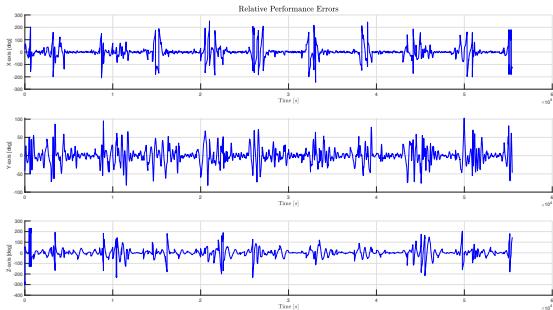
Figure 118: Nominal Scenario 600km LTAN 11 [1]



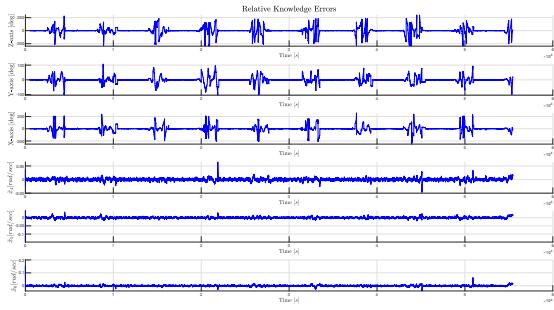
(a) MEKF Estimation Results



(b) RW Angular Velocity



(c) Relative Performance Error



(d) Relative Knowledge Error

Figure 119: Nominal Scenario 600km LTAN 11 [2]

Additionally, 95% confidence intervals for the median Absolute Performance Error in each axis is calculated, as defined in the error metrics [Section 4.1](#).

	Z	Y	X
Eclipse included	[10.9331,11.0726]	[10.5677,10.6756]	[28.1624,28.4132]
Eclipse not included	[5.1791,5.2398]	[6.1340,6.1861]	[24.3075,24.4293]

	Percentage of fulfilling COMMS requirements
Eclipse included	52%
Eclipse not included	79%



7 Variables Table

Table XXI: Variables Table

General Parameters				
Variable	Variable name	Value	Units	Comments
Satellite mass	m	4	Kg	-
Inertia matrix	I	$\begin{bmatrix} 0.0412 & 0.0000 & 0.0014 \\ 0.0000 & 0.0410 & 0.0000 \\ 0.0014 & 0.0000 & 0.0070 \end{bmatrix}$	kg·m ²	-
Center of mass	Cm	[0.00415, 0.00116, 0.0016]	m	-
Earth radius	Re	6371200	m	-
Satellite altitude	Rs	500	km	-
Gravitational constant	G	$6.67428 \cdot 10^{-11}$	m ³ ·kg ⁻¹ ·s ⁻²	-
Earth mass	M	$5.972 \cdot 10^{24}$	kg	-
Satellite angular velocity relative to Earth	w_o	$1.108 \cdot 10^{-3}$	rad/sec	$\sqrt{\frac{G \cdot M}{\text{Radius}^3}}$
Satellite linear velocity in orbit	v_satellite	7616.33	m/sec	$\sqrt{\frac{G \cdot M}{\text{Radius}}}$
Orbit Period	orbit_period	5545	sec	$(2 \cdot \pi) / (w_o)$
Initial Angular Velocity on each axis expressed in Body Frame	vRot0	(varying)	rad/sec	-

Determination Parameters				
Process Covariance matrix	Q	6×6 matrix	-	-
Noise Covariance matrix	R	6×6 diagonal matrix	-	-
Magnetometer Noise	m_noise	$\text{sqrt}([1.83 \cdot 10^{-6}, 1.83 \cdot 10^{-6}, 1.83 \cdot 10^{-6}])$	-	-
Coarse Sun Sensor Noise	css_noise	0.01	-	-
Gyroscope Noise	gyro_noise	0.0026	-	-
Gyroscope Bias Drift	gyro_bias	$7.7570 \cdot 10^{-4}$	-	-

Control Parameters				
Kp gain for PD controller	Kp_gain	[20, 100, 90]· 10 ⁻⁵	-	-
Kd gain for PD controller	Kd_gain	[90, 90, 90]· 10 ⁻⁴	-	-
Kp gain during eclipse for PD controller	$1.2 \cdot Kp_gain$	-	-	
Kd gain during eclipse for PD controller	$20 \cdot Kd_gain$	-	-	
Kp gain for B-dot controller	Kp_gain_bdot	1	-	-
Desired quaternion	x_desired	[1, 0, 0, 0]	-	-
Magnetic field data from magnetometers	B_body	(varying)	T	-
Initial orientation in Quaternion	q	(varying)	-	-

Magnetorquer parameters				
Core Diameter	-	[10, 10, -]	mm	-
Length	-	[70, 70, 70]	mm	-
Width	-	[-, -, 10]	mm	-
Height	-	[-, -, 4]	mm	-
Area enclosed by the coil	A_coils	[0.002199, 0.002199, 0.001566]	mm ²	-
Maximum magnetic dipole	m	[0.2, 0.2, 0.2]	Am ²	-
Wire turns	N_coils	[400, 400, 800]	-	-
Resistance	R_coils	[110, 110, 31.32]	Ohm	-

Reaction Wheel Parameters				
Maximum Torque	-	$1 \cdot 10^{-4}$	Nm	-
Inertia	Jw	$1.9 \cdot 10^{-6}$	Kg·m ²	-
Coulomb friction constant	c_friction	$1.9 \cdot 10^{-7}$	N·m	-
Viscous friction constant	b_friction	$1.9 \cdot 10^{-9}$	N·m·s	-
Deadzone Limit	lim_dz	[-300, 300]	rpm	-
Desaturation Limit	-	10000	rpm	-
Armature Resistance	Rb	10	Ohm	-
Motor Torque constant	Km	$2 \cdot 10^{-4}$	Nm/A	-