

Design, Implementation and Verification of the Attitude Determination and Control Algorithms for the DelFFi Satellites

Morteza Haghayegh



Design, Implementation and Verification of the Attitude Determination and Control Algorithms for the DeIFFi Satellites

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Space Systems Engineering at
Delft University of Technology

Morteza Haghayegh

August 31, 2015



Copyright © Spaceflight
All rights reserved.



Abstract

This thesis is about the design, implementation and verification of the Attitude Determination and Control (ADC) algorithms for the DelFFi satellites. The DelFFi satellites are two satellites that are being developed for the DelFFi mission as part of the Delfi program. In contrast to a passive control system, an active attitude determination and control system has been chosen for the DelFFi satellites in order to fulfil the pointing requirements of the DelFFi mission. The goal of this thesis is to answer the following two research questions:

- What are the performance issues associated with the ADC algorithms in case of nominal, non-nominal and practical conditions?
- What are the sources of errors when running the ADC algorithm on the microcontroller and how can these be mitigated?

The first research question has been answered by developing a spacecraft dynamics and space environment model (also referred to as the simulation environment), designing the ADC algorithms and running the simulation of the ADC algorithms using the simulation environment for varying conditions. The second research question has been answered by verifying the ADC algorithms using a software-in-the-loop test and a test on the processor.

The ADC algorithms consist of the mode determination algorithm and the algorithm of the four modes, namely the detumble mode, velocity-pointing mode, thrust vector control mode and safe mode. The thrust vector control mode will be designed at a later stage of the project.

The analysis of the simulation results of the detumble mode shows that there is a certain optimum measurement update frequency. Moreover, it has been shown that the detumble mode is robust enough to cope with an increase in aerodynamic drag and sensor measurement noise. However, analysis shows that a failure around certain axes will cause the satellite to continue tumbling. An analysis has also been performed on the simulation results of the velocity-pointing mode. It has been found that for a larger inaccuracy than nominal of the IGRF look-up table, the EKF will have poor performance. However, an inaccuracy of the measured moment of inertia does not decrease the EKF performance. Moreover, it has been shown that the velocity-pointing mode is not sufficient for accurate pointing of the thrusters.

Running a software-in-the-loop test and a test on the processor has shown that the C code of the ADC algorithms require some modifications in order to function correctly. These modifications include changes in the coding architecture and rewriting of Simulink functions. These tests have shown that the ADC algorithms can run without errors on the processor after these modifications have been implemented in the code of the ADC algorithms.

Table of Contents

Acknowledgements	xiii
1 Introduction	1
1-1 Background	1
1-2 The Objectives of the Thesis	2
1-3 Research Methodology	2
1-4 Thesis Structure	3
2 Attitude Determination and Control Subsystem of the DelFFi Satellites	5
2-1 Introduction to the DelFFi Mission	5
2-1-1 Delfi Program	5
2-1-2 QB50 Program and the DelFFi Mission	5
2-2 Orbit and Satellite Properties of the DelFFi Mission	6
2-3 Hardware	7
2-3-1 Attitude Sensors	7
2-3-2 Attitude Actuators	9
2-4 Mode Switch Logic	10
2-4-1 Mode Switch Logic Defined by the DelFFi Team	10
2-4-2 Mode Switch Logic Defined for this Thesis	10
2-5 Timing Sequence of the ADCS	13
2-6 Overview of the ADC Algorithms and the Simulation Environment	14
3 Dynamics and Kinematic Models and Enviroment Modelling	15
3-1 Common Spacecraft Reference Frames	15
3-2 Transformation Between Reference Frames	17
3-2-1 Transformation ECI frame to ECEF frame	18
3-2-2 Transformation ECI frame to T frame	18

3-2-3	Transformation ECI frame to O frame	21
3-3	Rotational Kinematics	22
3-3-1	Direction Cosine Matrix	23
3-3-2	Euler Angles	24
3-3-3	Quaternion	25
3-3-4	Kinematic Equation	26
3-4	Euler's Dynamic Equation	27
3-5	Spacecraft Orbit Dynamics	27
3-6	Environment Modelling	28
3-6-1	Model of the Geomagnetic Field	28
3-6-2	Model of the Sun's position	30
3-6-3	Model of the Disturbance Torques	31
3-7	Sensor and Actuator Models	35
3-7-1	Magnetometer Modelling	35
3-7-2	Sun Sensor Modelling	35
3-7-3	Magnetic Torquer Modelling	35
4	Design of the Algorithms	39
4-1	Tumble Parameter Determination Algorithm	39
4-2	Attitude Estimation Algorithm	40
4-3	Control Algorithm	45
4-3-1	Detumble Mode Control	45
4-3-2	Velocity-pointing Mode Control	46
4-4	Summary	48
5	Simulation Results	51
5-1	Detumble Mode	51
5-1-1	Performance Analysis	51
5-2	Velocity-pointing Mode	58
5-2-1	Tuning Parameters	59
5-2-2	Performance Analysis	64
5-3	Requirement Verification and Recommendations	71
6	Testing of the ADC Algorithm	73
6-1	Software Issues	73
6-2	Algorithm Testing	74
6-2-1	Software-in-the-loop Testing	74
6-2-2	Testing the ADC Algorithm on the Processor	81
6-3	Recommended Testing	84
7	Conclusions & Recommendations	87

A DelFFi ADCS Requirements	91
A-1 List of Requirements	91
A-2 ADCS Requirements of the DelFFi Mission	96
B Additional Equations, Derivations and Functions	97
B-1 Arccosine2 Function	97
B-2 Angular Pointing Error and Angular Estimation Error	97
Bibliography	99

List of Figures

1-1	The simulation loop of the ADC algorithm and the simulation environment [1]	3
2-1	Structure of the DelFFi Satellites [2]	6
2-2	Mode switch logic of the attitude modes as defined by the DelFFi team	11
2-3	Mode Switching Logic for the Attitude Determination and Control algorithms of the DelFFi satellites (* = this mode will not be discussed in this thesis)	13
2-4	overview of the ADC algorithms and the simulation environment	14
3-1	ECI reference frame [3]	16
3-2	ECEF reference frame [3]	16
3-3	Sun pointing reference frame	17
3-4	Thrust pointing reference frame	17
3-5	First rotation for the transformation from ECI reference frame to Thrust-pointing reference frame	19
3-6	Second rotation for the transformation from ECI reference frame to Thrust-pointing reference frame	20
3-7	First rotation for the transformation from ECI reference frame to Orbit reference frame	22
3-8	Second rotation for the transformation from ECI reference frame to Orbit reference frame	23
3-9	Third rotation for the transformation from ECI reference frame to Orbit reference frame	24
3-10	Geomagnetic Field represented as a dipole [4]	29
3-11	Eclipse occurrence	31
3-12	Graph of the air density versus the altitude for a varying solar flux [5]	32
3-13	Thruster misalignment overview	34
3-14	Hysteresis curve of a round loop [6]	37
3-15	Hysteresis curve of a flat loop [6]	37

4-1 Overview of the Extended Kalman Filter	42
4-2 Overview of the Attitude Determination and Control Algorithm	49
5-1 Geomagnetic field vector expressed in the B frame for $\omega_0 = 10 \text{ deg/s}$ (Case E)	54
5-2 Geomagnetic field vector expressed in the B frame for $\omega_0 = 60 \text{ deg/s}$ (Case E)	54
5-3 Angular velocity of the satellite with respect to I frame for $\omega_0 = 10 \text{ deg/s}$ (timing sequence = case E)	54
5-4 Angular velocity of the satellite with respect to I frame for $\omega_0 = 60 \text{ deg/s}$ (timing sequence = case E)	54
5-5 Angular velocity of the satellite with respect to I frame (Case L)	57
5-6 Tumble parameter P_{tumble} (Case L)	57
5-7 Angular velocity of the satellite with respect to I frame (Case M1)	58
5-8 Angular velocity of the satellite with respect to I frame (Case M2)	58
5-9 Angular velocity of the satellite with respect to I frame (Case M3)	58
5-10 Angular velocity of the satellite with respect to I frame (Case M4)	58
5-11 Angular velocity of the satellite with respect to I frame (Case M5)	59
5-12 Angular velocity of the satellite with respect to I frame (Case M6)	59
5-13 Angular Estimation Error in [deg] of case A	61
5-14 Angular Estimation Error in [deg] of case B	61
5-15 Angular Estimation Error in [deg] of case C	61
5-16 Angular Estimation Error in [deg] of case D	61
5-17 Angular Estimation Error in [deg] of case E	62
5-18 Angular Estimation Error in [deg] of case F	62
5-19 Angular Estimation Error in [deg] of case G	62
5-20 Angular Estimation Error in [deg] of case H	62
5-21 Angular Pointing Error of case I	63
5-22 Angular Pointing Error of case J	63
5-23 Angular Pointing Error of case K	63
5-24 Angular Pointing Error of case L	63
5-25 Angular Pointing Error of case M	64
5-26 Angular Pointing Error of case N	64
5-27 Angular Estimation Error of a case with nominal conditions	66
5-28 Angular Pointing Error of a case with nominal conditions	66
5-29 Angular Pointing Error of case P	67
5-30 Angular Estimation Error of case Q1	68
5-31 Angular Pointing Error of case Q1	68
5-32 Angular Estimation Error of case Q2	68
5-33 Angular Pointing Error of case Q2	68

5-34 Angular Estimation Error of case R	69
5-35 Angular Pointing Error of case R	69
5-36 Angular velocity of case S	70
5-37 Angular Pointing Error of case S	70
5-38 Angular Pointing Error of case S. Zoomed in on the last thrusting period	71
6-1 Visualisation of the SIL test set-up for case A	75
6-2 Visualisation of the SIL test set-up for case B1	75
6-3 Visualisation of the SIL test set-up for case B2	76
6-4 Visualisation of the SIL test set-up for case C	77
6-5 Visualisation of the SIL test set-up for case D	77
6-6 Rotational rate in case A	77
6-7 Mode determination variable in case A	77
6-8 Angular Estimation Error in case A	78
6-9 Angular Pointing Error in case A	78
6-10 Rotational rate in case B1	79
6-11 Mode determination variable in case B1	79
6-12 Angular Estimation Error in case B1	79
6-13 Angular Pointing Error in case B1	79
6-14 Accuracy of the commanded magnetic dipole moment (case B2)	80
6-15 Rotational rate in case D	81
6-16 Mode determination variable in case D	81
6-17 Hardware-in-the-loop testing of the ADC algorithm	85

List of Tables

2-1	Orbit and satellite properties of the DelFFi mission	7
2-2	Key characteristics of the magnetometers [7]	8
2-3	Key characteristics of the Sun sensors [7]	8
2-4	Key characteristics of the magnetic torquer (MTQ) system [7]	9
2-5	Key characteristics of the reaction wheel system [8]	9
2-6	New ADCS requirements for the DelFFi satellite (* = this mode will not be discussed in this thesis)	12
3-1	Table of the air density for varying solar flux conditions and highest and lowest altitude	32
4-1	List of most commonly used filters in the field of spacecraft attitude estimation.	42
5-1	Overview of the cases for determining the optimal timing sequence interval	52
5-2	Parameters used in the simulations for finding the optimal timing sequence interval	53
5-3	Parameters used for the detumble mode	53
5-4	Results of the simulations of the detumble mode for varying timing sequence intervals.	53
5-5	Overview of the cases for the performance investigation of the Detumble mode. The values used in this table are obtained from the orbit and satellite properties table (table 2-1)	55
5-6	Results of the performance of the detumble mode for non-nominal conditions of the last 15 minutes for total simulation time of 8 orbits using Monte Carlo simulations. All values given in deg/s unless otherwise stated.	57
5-7	Attitude estimation parameter tuning cases	60
5-8	Control gain parameters tuning cases	60
5-9	Filter parameters used on the Delfi-n3Xt mission	60
5-10	Mean values of the Monte Carlo simulations for cases I to N	64

5-11	Mean value of the angular pointing error of the last 10 minutes before the last eclipse (or in case S, before the last thrusting period) for a simulation of 5 orbits	69
6-1	Results of the quaternion obtained from the EKF in case A and in case C	80
6-2	Overview of the Simulink functions used in the ADC algorithms	82
6-3	Angular estimation error [deg] during Sunlit phase between the second last and last eclipse for a Simulink simulation of 7 orbits	83
6-4	Output of algorithm in Simulink using the standard matrix exponential simulink block and an approximation to the matrix exponential function	84
A-1	DeFFi Requirements Specification List Part 1 [9]	92
A-2	DeFFi Requirements Specification List Part 2 [9]	93
A-3	DeFFi Requirements Specification List Part 3 [9]	94
A-4	DeFFi Requirements Specification List Part 4 [9]	95
A-5	ADCS Requirements DeFFi Mission ([9])	96

Acknowledgements

I would like to express my gratitude to dr. Guo, my thesis supervisor, for his guidance and useful critique of this thesis work. Moreover, I would like to thank Nuno Baltazar dos Santos for providing the necessary hardware to run tests on and Jaan Viru for his assistance in using the hardware.

I would also like to thank Xiaoxiang Liu and the DelFFi team for providing useful critique in designing the model.

Furthermore I wish to express my gratitude to Emanuelle de Angelis for his guidance in simulating control algorithms in Simulink during my internship at ALMASpace .

Finally, I would like to thank my parents for their support and encouragement throughout my study.

Delft, University of Technology
August 31, 2015

Morteza Haghayegh

Acronyms

ADC	Attitude Determination and Control
ADCS	Attitude Determination and Control Subsystem
AEKF	Additive Extended Kalman Filter
CDHS	Command and Data Handling Subsystem
CPU	Central Processing Unit
DCM	Direction Cosine Matrix
ECEF	Earth Centered Earth Fixed
ECI	Earth Centered Inertial
EKF	Extended Kalman Filter
IGRF	International Geomagnetic Reference Field
LQR	Linear Quadratic Regulator
MEKF	Multiplicative Extended Kalman Filter
MM	Magnetometer
MTQ	Magnetic Torquer
OBC	On-Board Computer
OF	Orthogonal Filter
PCB	Printed Circuit Board
PD	Proportional Differential
PF	Particle Filter
PIL	Processor-in-the-loop
QUEST	Quaternion Estimator
QFR	Quaternion Feedback Regulator
RW	Reaction Wheel
SGP4	Simplified General Perturbations Model 4
SIL	Software-in-the-loop
SIM	Simulation
SL	Service Layer

SS	Sun Sensor
TLEs	Two Line Elements
TVC	Thrust Vector Control
UKF	Unscented Kalman Filter
VP	Velocity-pointing

Nomenclature

$\dot{\cdot}$	First time derivative of a parameter
$\hat{\cdot}$	Estimate of a parameter
$[() \times]$	Skew-symmetric matrix of a vector
$(\cdot)_0$	Initial value of a parameter
$(\cdot)_\sigma$	Standard deviation of a parameter
$(\cdot)_{avg}$	Average value of a parameter
$(\cdot)_b$	The bias of a parameter
$(\cdot)_B$	A vector expressed in the body fixed reference frame
$(\cdot)_d$	Desired value of a parameter
$(\cdot)_e$	The error of a parameter
$(\cdot)_E$	A vector expressed in the Earth Centered Earth Fixed reference frame
$(\cdot)_{EKF}$	The parameter obtained from the Extended Kalman Filter
$(\cdot)_i$	A parameter at time step i
$(\cdot)_{i+1}$	A parameter at time step i+1
$(\cdot)_I$	A vector expressed in the Earth Centered Inertial reference frame
$(\cdot)_{k/k}$	A parameter at time step k given the parameter at time step k
$(\cdot)_{k+1/k}$	A parameter at time step k+1 given the parameter at time step k
$(\cdot)_{k+1/k+1}$	A parameter at time step k+1 given the parameter at time step k+1
$(\cdot)_m$	The measured value of a parameter
$(\cdot)_{max,eff}$	Effective maximum value of a parameter
$(\cdot)_{max,theoretical}$	Theoretical maximum value of the parameter
$(\cdot)_n$	The noise of a parameter
$(\cdot)_O$	A vector expressed in the orbit fixed reference frame
$(\cdot)_S$	A vector expressed in the Sun Pointing reference frame
$(\cdot)_T$	A vector expressed in the Thrust Pointing reference frame
α	Low-pass filter constant
α	angle
σ_{SS}	Process noise of the Sun sensors
σ_{MM}	Process noise of the magnetometer
σ_q	Processor noise of the quaternion
σ_ω	Process noise of the angular velocity

ϵ	Angle between the ecliptic plane and the equatorial plane
γ_E	Earth's half angular size from the orbit of DelFFi
θ	True Anomaly
θ	Colatitude
ϕ	Longitude
ρ	Air density
ω	Angular velocity
Δ	Difference between two values
Δt	Time to complete one loop
a	Semi-major axis
B	Magnetic field vector
d_y	Length in y direction
d_z	Length in z direction
D_O	Drag expressed in the Orbit fixed reference frame
e	Eccentricity
$e_{allowed}$	Allowed relative error
E	Eccentricity Anomaly
$F_{thruster}$	Force produced by the thruster
F	Jacobian matrix
H	Angular Momentum
H	Observation matrix
I	Identity matrix
i	inclination
J	Moment of Inertia
J_{EKF}	Moment of inertia inserted in the EKF
J_{measured}	Measured moment of inertia
J	Loss function
k	Counting element
k_b	Bdot controller gain
K_d	Derivative gain
K_p	Proportional gain
K_u	Ultimate gain
K	Kalman gain
L	Length of the state vector
$L_{detumbled}$	Detumble limit
L_x	Length of the satellite
m	Magnetic dipole moment
\mathbf{m}_{mtq}	Magnetic dipole moment produced by the magnetic torquers
$\mathbf{m}_{max,eff}$	Maximum effective magnetic dipole moment produced by the magnetic torquers
$\mathbf{m}_{max,theoretical}$	Maximum theoretical magnetic dipole moment produced by the magnetic torquers
M	Mean Anomaly
$\mathbf{M}_{thruster}$	Disturbance torque produced by the thruster around all three axis
n	Orbital speed
P	Tumble parameter
P	Variable gain of the LQR
P_{tumble}	Tumble parameter
P	Estimation error covariance matrix

q	Quaternion
Q	Process noise matrix
Q_c	covariance matrix
r	Vector pointing from center of the Earth to the satellite in meters
r_{cp}	Distance between center of gravity and center of pressure
R	Rotation matrix
R	Sensor noise matrix
R_c	covariance matrix
S	Sun pointing vector
t-t _{ev}	Time since last alignment of Sun direction with the Vernal Equinox given in seconds.
t _{detumbled}	Time that tumble parameter needs to be below detumble limit to ensure satellite is detumbled
t _{actuator}	Period of time where the actuators are on
t _{total}	Total time of one control loop
T	Torque
T_c	Control torque
T_d	Disturbance torque
T_g	Gravity Gradient Torque
T_{mtq}	Torque produced by the magnetic torquers
T _{orb}	Time to complete one orbit
T _s	Settling time
T _u	Oscillation period
T_x	Transformation matrix around the x axis
T_y	Transformation matrix around the y axis
T _{year}	Time in seconds in one year
T_z	Transformation matrix around the z axis
U	Perturbed moment of inertia
u	Control torque output
v	vector
V	Velocity vector
w	Non-negative weights
x	state vector

Chapter 1

Introduction

Within the framework of the Delfi program, Delft University of Technology has been developing two satellites for the DelFFi mission. The objective of the DelFFi mission is to demonstrate formation flying and to perform in-situ measurements in the lower thermosphere. For this mission the Attitude Determination and Control Subsystem (ADCS) will be critical in order to point the thruster and the instruments in the correct direction for formation flying and performing in-situ measurements, respectively. The ADCS of a satellite is the subsystem that provides the capability to determine the current orientation of the satellite and control the actuators to change this orientation in order to reach a desired attitude. In this thesis the attitude determination and control (ADC) algorithms of the DelFFi satellite will be designed and verified. The focus of this thesis will therefore be:

*Design, Implementation and Verification of the Attitude Determination and
Control Algorithms for the DelFFi satellites*

1-1 Background

The Delfi program aims to develop Cubesats at the Delft University of Technology, to provide hands-on education and training of students as well as technology demonstration for Dutch space industry and research and development on the nano-satellite spacecraft bus. Delft University of Technology will develop and launch nano-satellites on a regular basis to ensure continuity of the program as much as possible. The Delfi program started with the design and launch of the Delfi-C³ satellite on April 2008. The main objective of the Delfi-C³ mission is to provide hands-on education and training of students on a real spacecraft project, demonstration of four solar arrays for Dutch Space, demonstration of Sun sensors from TNO, testing of a radio amateur platform of TU Delft, development of a modular nano-satellite spacecraft bus and a distributed ground segment. In order to meet these objectives, the ADCS of the Delfi-C³ did not have to be advanced. It was chosen to use a passive magnetic attitude control system. This system will not actively control the attitude of the spacecraft, but it will

use magnetic hysteresis material and a permanent magnet to limit the rotation rate of the spacecraft. The next project of the Delfi program is the Delfi-n3Xt mission which launched the Delfi-n3Xt satellite in November 2013. One significant improvement of the Delfi-n3Xt compared to the Delfi-C³ is the active ADCS. The active ADCS of the Delfi-n3Xt satellite consists of six Sun sensors, a triple-axis magnetometer, 3 reaction wheels and triple axis magnetic torquer. Active ADCS is mainly important for advanced mission requirements, such as precise imagery of the Earth, thruster pointing for orbit manoeuvres, antenna pointing for communication with ground stations and many more applications. For the DelFFi mission, the heritage of the Delfi-C³ and Delfi-n3Xt will be used.

1-2 The Objectives of the Thesis

The objective of this thesis is to design, implement and verify the attitude determination and control algorithm which will run on the DelFFi satellite. The ADC algorithm will be robust and it will need to have the capability to meet the detumbling requirements or the pointing requirements, depending on the mode. The ADC algorithm will not only be tested during nominal conditions, but also during practical and non-nominal conditions. These practical and non-nominal conditions are high noise in the magnetometer measurements, magnetic torquer failure, increase in the distance between the center of gravity and center of pressure, inaccurate values in the IGRF table, inaccurate determination of the moment of inertia and thruster firing. In order to meet the objective, the following research questions have been formulated:

- What are the performance issues associated with the ADC algorithm in case of nominal, non-nominal and practical conditions?
- What are the sources of errors when running the ADC algorithm on the microcontroller and how can these be mitigated?

1-3 Research Methodology

The design, implementation and verification of the ADC algorithm for the DelFFi satellite can be split up into several parts.

Firstly, the simulation environment will be designed. This simulation environment will provide the space environment model and the model of the spacecraft dynamics to determine the behaviour of the satellite. The space environment models and the model of the spacecraft dynamics will consist of the following:

- Vehicle attitude dynamics
- Orbit dynamics
- Model of the disturbance torques
- Model of the physical world

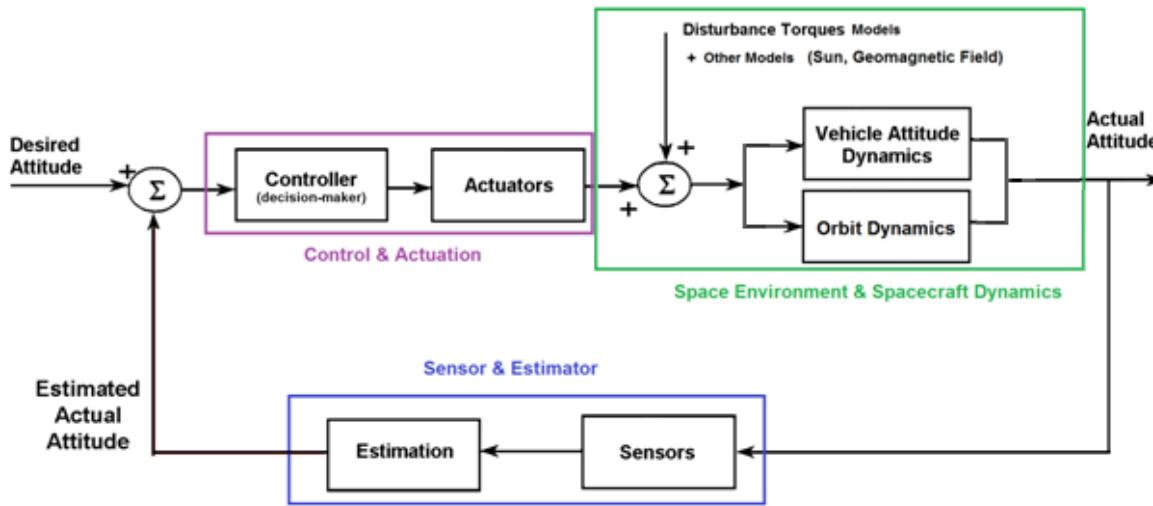


Figure 1-1: The simulation loop of the ADC algorithm and the simulation environment [1]

Using these models the position, velocity and attitude of the satellite will be determined. It should be noted that the simulation environment is not part of the ADC algorithm, but it is essential for simulating the functioning of this algorithm.

Secondly, the ADC algorithm will be designed. A trade-off of the attitude determination and control methods will be conducted to select the most suitable method.

Thirdly, the algorithm will be simulated using the simulation environment. This simulation will be used to check whether the detumbling and pointing requirement have been met. Moreover, the simulation will also run for practical and non-nominal conditions. Figure 1-1 shows one loop of the simulation.

Lastly, the algorithms will be converted to C code and they will be verified using a software-in-the-loop simulation and a test on the processor. The SIL test will verify whether the algorithm meets the detumbling requirement and the pointing requirement. A correct functioning algorithm might still behave differently when running on a processor. Therefore it is necessary to verify the functioning of the algorithm on the processor.

1-4 Thesis Structure

This thesis report will start off by providing a short introduction to the QB50 program and the Delfi program in chapter 2. In addition, this chapter will also provide the ADCS requirements and sensor and actuator characteristics. Furthermore, top-level design aspects of the mode switch logic and the timing sequence of the ADCS will be discussed. Chapter 3 will give the spacecraft dynamics and kinematics model, the common spacecraft reference frames, the transformation matrices between these reference frames and the environment model. The environment model will consist of the geomagnetic field model, Sun's position model and the model of the disturbance torques. In chapter 4 the theory of the attitude

determination and control algorithm will be presented. Simulations of this algorithm will run in the Simulink environment and the results will be presented in chapter 5. Chapter 6 will discuss the software-in-the-loop tests and tests on the processor. Finally, chapter 7 will present the final conclusions and the recommendations. In the conclusion section the research questions will be answered in detail.

Chapter 2

Attitude Determination and Control Subsystem of the DelFFi Satellites

This chapter will give a short overview of the history and the current activities of Delfi program. Moreover, the ADCS hardware and control modes of the DelFFi satellites will be discussed.

2-1 Introduction to the DelFFi Mission

This section will provide a short introduction into the DelFFi project and mission as well as the Delfi program and the QB50 project.

2-1-1 Delfi Program

The Delfi program is a project at the Delft University of Technology. This project is intended to give students hand-on education and training in addition to technology demonstration for the Dutch space industry. The program started with the Delfi-C³ satellite in November 2004 and launched and operated successfully on 28 April 2008 and onwards. Afterwards the Delfi-n3Xt project initiated in November 2007 and launched in November 2013. The mission objective of the Delfi-n3Xt satellite is to demonstrate propulsion, communication systems and ADCS for future mission. This satellite will be used as a heritage for its successor, the DelFFi satellites. To conclude, the main objectives of the Delfi program are education, technology demonstration and nanosatellite bus development.

2-1-2 QB50 Program and the DelFFi Mission

The QB50 project is a unique mission consisting of an international network of 50 nano-satellites for multi-point, performing in-situ measurements in the lower thermosphere and

re-entry research. The scientific objective of the QB50 project is to study the temporal and spatial variations of a number of key constituents and parameters in the lower atmosphere. Delft University of Technology is providing two identical satellites, Phi and Delta, within the framework of the QB50 mission and the Delfi program. These satellites will demonstrate the possibility of formation flying. The satellite platform designed for Delta and Phi is based on the Delfi-n3Xt triple-unit CubeSat with modifications specifically for the DelFFi mission. The structure of the DelFFi satellites is depicted in figure 2-1.

The mission statement of the DelFFi mission is as follows:

"The DelFFi mission shall demonstrate autonomous formation flying and provide enhanced scientific return within QB50 from the 2015 onwards, by utilizing two identical triple-unit Cubesats of TU Delft which further advance the Delfi-n3Xt platform"

2-2 Orbit and Satellite Properties of the DelFFi Mission

Table 2-1 provides the set of parameters of the orbit and the satellite which will be used in all simulations. The orbit parameters have been obtained from an initial estimation [9] of the orbit in which the satellite will be inserted. The satellite parameters have been obtained from estimations based on the Delfi-n3Xt satellite [10] and scaled to the sizing of the DelFFi satellites. The sensor properties have been obtained from the datasheets of the sensors which have been summarised in the DelFFi Design Overview Report [7]. The satellite shall have an initial altitude of 380 km down to at least 200 km [9]. The science sensors will perform in-situ measurements within this altitude range.

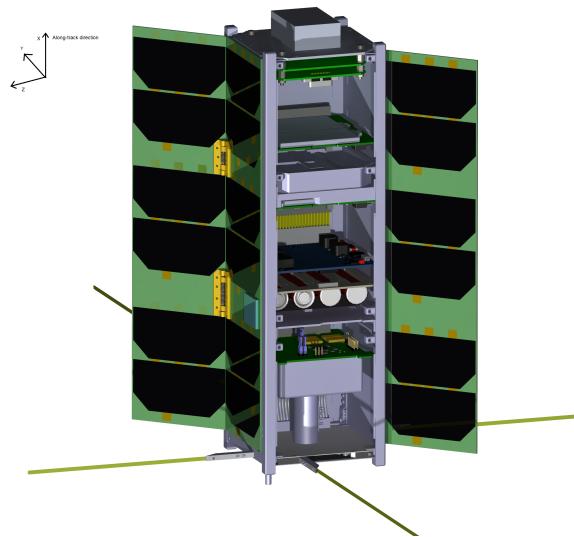


Figure 2-1: Structure of the DelFFi Satellites [2]

Table 2-1: Orbit and satellite properties of the DelFFi mission

Parameter	Value	Unit
Orbit type	Circular	
Eccentricity	0	[\cdot]
Inclination	98.6	degree
Right ascension of the ascending node	-15	degree
Argument of perigee	0	degree
Initial mean anomaly	0	degree
Altitude	200 to 380	km
Satellite inertia	$\begin{bmatrix} 0.017 & 0 & 0 \\ 0 & 0.055 & 0 \\ 0 & 0 & 0.055 \end{bmatrix}$	$\text{kg}\cdot\text{m}^2$
Satellite dimensions	$\begin{bmatrix} 0.1 & 0.34 & 0.1 \end{bmatrix}^T$	m
Center of pressure offset from center of gravity	$\begin{bmatrix} 0.005 & 0.005 & 0.005 \end{bmatrix}^T$	m
Drag coefficient	2.2	[\cdot]
Maximum magnetic torquer dipole moment	$\begin{bmatrix} 0.4 & 0.4 & 0.4 \end{bmatrix}^T$	$\text{A}\cdot\text{m}^2$
Residual magnetic dipole moment	$\begin{bmatrix} 0.005 & 0.005 & 0.005 \end{bmatrix}^T$	$\text{A}\cdot\text{m}^2$
Magnetometer bias	500	nT
Magnetometer noise standard deviation	170	nT
Sun sensor bias	< 3	degree
Sun sensor noise standard deviation	0.4	degree

2-3 Hardware

This section will provide a short overview of the ADCS hardware on-board of the DelFFi satellites. The hardware on-board are [11]

- Two magnetometers (of which one is redundant)
- Six Sun sensors
- Three magnetorquers
- Three reaction wheels

2-3-1 Attitude Sensors

This subsection will discuss the two attitude sensors that the DelFFi satellites will be equipped with, namely the magnetometer and the Sun sensors.

Magnetometer

The magnetometers (MM) that will be used are two Honeywell HMC5883L three-axis fluxgate magnetometer. The characteristics of this type of magnetometer, which is used frequently in mobile phones and game devices, is given in table 2-2 [7]. This magnetometer has been chosen for its low-cost, because it is a Commercial-Off-The-Shelf component.

Table 2-2: Key characteristics of the magnetometers [7]

Section	Parameter	Input
General	Brand/Model	Honeywell HMC5883L
	Communication type	I2C
	Configuration	Triple-axis, orthogonal
Specification	Measurement range	+/- 100 μ T
	Nominal Range In-orbit (scalar)	20 μ - 47 μ T
	Measurement resolution	65 nT
Accuracy	Noise type	white, Gaussian
	Noise level (open field)	170 nT (1σ)
	Bias level	500 nT

Sun Sensors

The DelFFi satellites will each be equipped with 6 Sun sensors (SS) placed on each side of the structure. A casing made of PEEK material provides a square window. A few millimeters below the window, a quadrant photodiode is placed which is read out by circuitry. This allows for shadows to be cast on each of the four photodiodes based on the angle of the Sun. A simple operational amplifier circuit is used to read the output of the photodiodes as a voltage. The voltage output of each photodiode is summed based on rows and columns, and then the values of both rows or both columns are differenced and divided by the sum of the voltages for all four photodiodes. The value of both difference over sums for the rows and columns is compared to a table to determine the angle of the Sun in the sensor frame. The characteristics of these Sun sensors can be found in table 2-3 [7].

Table 2-3: Key characteristics of the Sun sensors [7]

Section	Parameter	Input
General	Brand/Model	TU Delft μ SS-1
	Communication type	I2C
	Configuration	1 on each satellite face
Specification	Field of view (FOV)	+/- 60 degree
	Power consumption	26 mW idle, 66 mW measuring
	Mass	10 gram
	Maximum read-out frequency	120 Hz
Accuracy	Noise type	white, Gaussian
	Noise level (open field)	0.4 degree (1σ)
	Bias	< 3 degree

2-3-2 Attitude Actuators

This subsection will discuss the two attitude actuators that the DelFFi satellites will be equipped with, namely the magnetic torquers and the reaction wheel system.

Magnetic Torquer

The magnetic torquers (MTQ) consist of two torque rods of mu-metal with coils wound around it for the X and Y axis and a open structure coil with a rod for the Z axis. The characteristics of the magnetic torquer system are given in table 2-4 [7].

Table 2-4: Key characteristics of the magnetic torquer (MTQ) system [7]

Section	Parameter	Input
General	Brand/Model	TU Delft μ MTQ System
	Communication type	Analogue
	Configuration	2 rods and 1 open coild
	Current driving	Pulse-width Modulation (PWM)
Specification	Range	$+/- 0.4 \text{ A} \cdot \text{m}^2$
	Resolution	- / off / +
	Power consumption	10-150 mW

Reaction Wheels

The reaction wheel system (RW) consists of three custom-made reaction wheels. The reaction wheels use brushless direct current motors and a bronze flywheel. A microcontroller located just below the motor on a PCB will drive the motors. The torque commands given by the Attitude Determination and Control algorithm are converted to accelerations of the flywheel by the driver electronics, which are subsequently converted into fast closed-loop control of the current to the motors. The characteristics of the reaction wheel system can be found in table 2-5 [7].

Table 2-5: Key characteristics of the reaction wheel system [8]

Section	Parameter	Input
General	Brand/Model	TU Delft RW+ System
	Communication type	I2C
	Configuration	Trip-axis
Specification	Angular momentum dynamic range	$+/- 1.44 \cdot 10^2 \text{ Nms}$
	Maximum torque at maximum speed	$5 \cdot 10^{-5} \text{ Nm}$
	Power consumption	10-150 mW
	Mass	92 grams

2-4 Mode Switch Logic

This section will discuss the mode switch logic as defined by the DelFFi team and the mode switch logic that will be used for this thesis. These two differ due to changes in the requirements as set by QB50 and scheduling changes that need to be taken into account. These changes will be discussed in this section as well.

2-4-1 Mode Switch Logic Defined by the DelFFi Team

The following attitude modes have been defined by the DelFFi team (appendix A-2):

1. *Detumble Mode*: The detumble mode will be used to detumble the satellite from its initial rotation rate to a desired rotation rate. In this mode the magnetometer and the magnetic torquers will be used.
2. *Nadir-pointing Mode*: The satellite needs to continuously point to the ground for communication with the ground station. This mode is the nominal mode, which means that it will be automatically activated whenever the satellite is detumbled.
3. *Sun-pointing Mode*: The satellite's wings located on the +z direction of the body fixed reference frame occasionally need to point to the Sun for maximum power generation. This mode will be initiated by a command from the ground. For a detailed discussion of the body fixed reference frame and the other reference frames, chapter 3 can be consulted.
4. *Thrust Vector Control Mode (TVC Mode)*: This mode will point the satellite's +x or the -x direction in the velocity vector, such that the thruster can decrease or increase the orbit altitude. This mode will also be initiated by a command from the ground.
5. *Safe Mode*: This mode is initiated in case the ADCS software detects any abnormal conditions. the detumble mode will be started again.

These attitude modes result in a mode switch logic as depicted in figure 2-2.

2-4-2 Mode Switch Logic Defined for this Thesis

For this thesis different attitude modes will be defined. This is because some aspects of the DelFFi mission have been changed. These changes are listed below:

- The antenna system of the satellite will consist of turnstile antennas, which gives a high overall gain and omni-directionality of the antennas [12]. Therefore a nadir-pointing mode is not necessary.
- The QB50 payload will need to point in the velocity vector, as discussed in appendix A-1 requirement MIS-C-M.1. Therefore the nominal attitude mode will be a mode where the payload is pointing in the velocity vector.

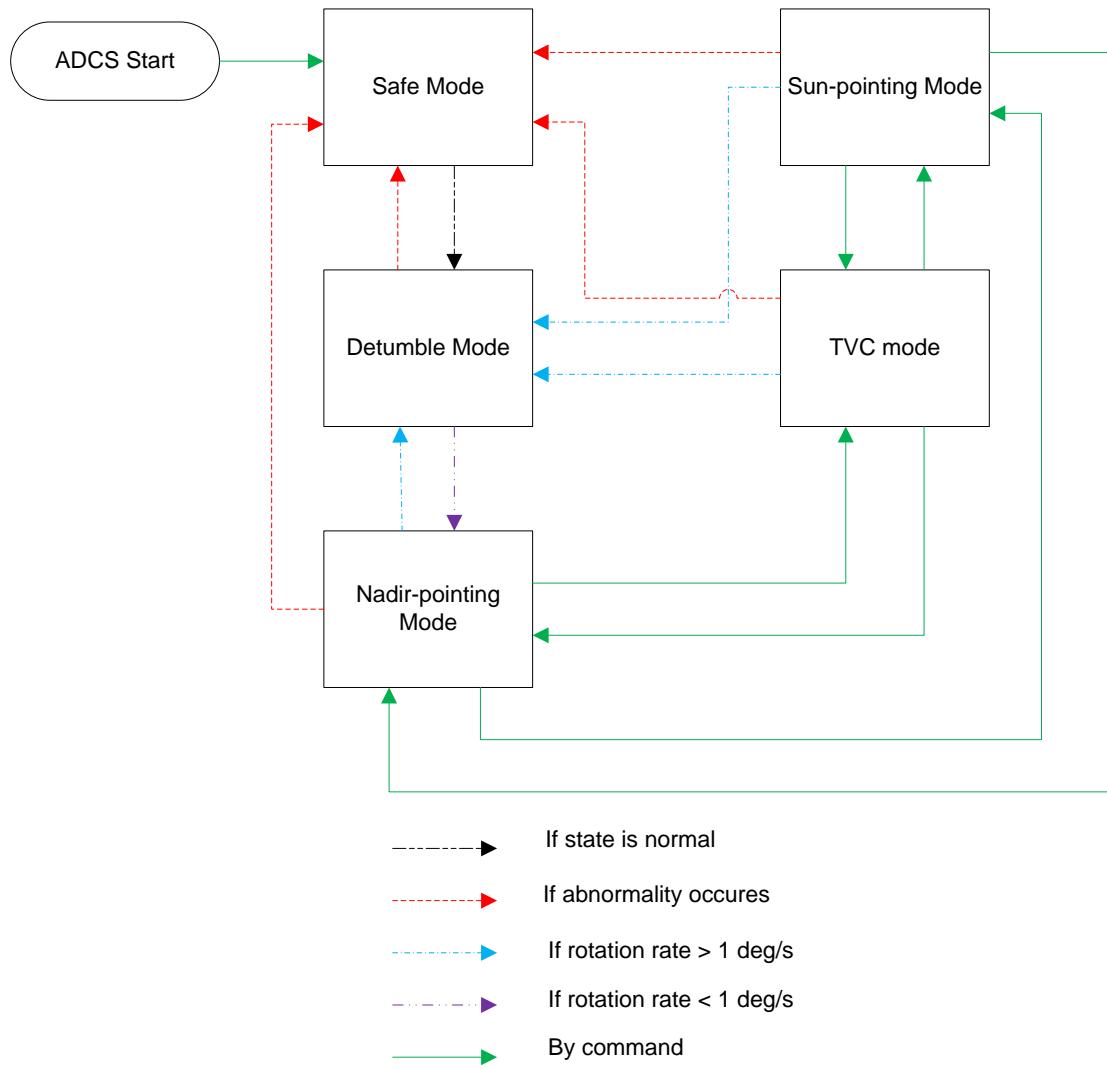


Figure 2-2: Mode switch logic of the attitude modes as defined by the DeIFFi team

- To be able to point the satellite's QB50 payload in the velocity vector at all times, the satellite will not have a Sun-pointing mode. Instead, the Sun-pointing mode will be included in the mode which points the QB50 payload in the velocity vector.

These items result in four attitude modes, namely:

1. *Detumble Mode*: After the ADCS is started, the Detumble mode is initiated. This will happen when the satellite has been jettisoned from the launch pod. The rotational rate threshold for successful detumbling has been set to 1 degree/s, as stated by the requirements (table A-5).
2. *Velocity-pointing Mode (VP mode)*: The payload needs to point in the along-track direction, hence the satellite principle axis needs to point in the along-track direction as well. This will be the nominal mode for the attitude determination and control subsystem. This mode will also include a Semi Sun-pointing algorithm. This algorithm will rotate the satellite around its roll axis to face the Sun for maximum power.
3. *Thrust Vector Control mode (TVC mode)*: The thruster is pointing in the negative x axis of the body-fixed reference frame which is in the along-track axis of the satellite, visualized in figure 2-1. Therefore in order to change the orbit the thruster will need to point in the along-track direction. This mode will use the reaction wheel system for higher pointing accuracies.
4. *Safe Mode*: This mode is initiated in case any abnormal conditions have been detected. All sensors and actuators will be switched off to save power, except for the magnetometer which uses low amount of power. As soon the abnormal condition is solved, the detumble mode will be started again.

Due to scheduling constraints the reaction wheel system of the satellite will not be used in the analysis in this thesis. This therefore excludes the TVC mode from the design in this thesis, because the TVC mode uses the reaction wheel system for higher angular pointing accuracy. It should be noted that the TVC mode will be implemented in the ADC algorithm at a later design stage.

An overview of the newly defined attitude control modes with their respective requirements has been given in table 2-6. These requirements have been set-up in cooperation with the project manager of the DelFFi mission.

Table 2-6: New ADCS requirements for the DelFFi satellite (* = this mode will not be discussed in this thesis)

Mode	Sensors	Actuators	Rotation rate accuracy required	Attitude accuracy required
Detumble mode	MM	MTQ	<1 deg/s	n/a
VP mode	MM, SS	MTQ	<1 deg/s	<10 deg
TVC mode *	MM, SS	MTQ, RW	<1 deg/s	<2 deg
Safe mode	MM	None	n/a	n/a

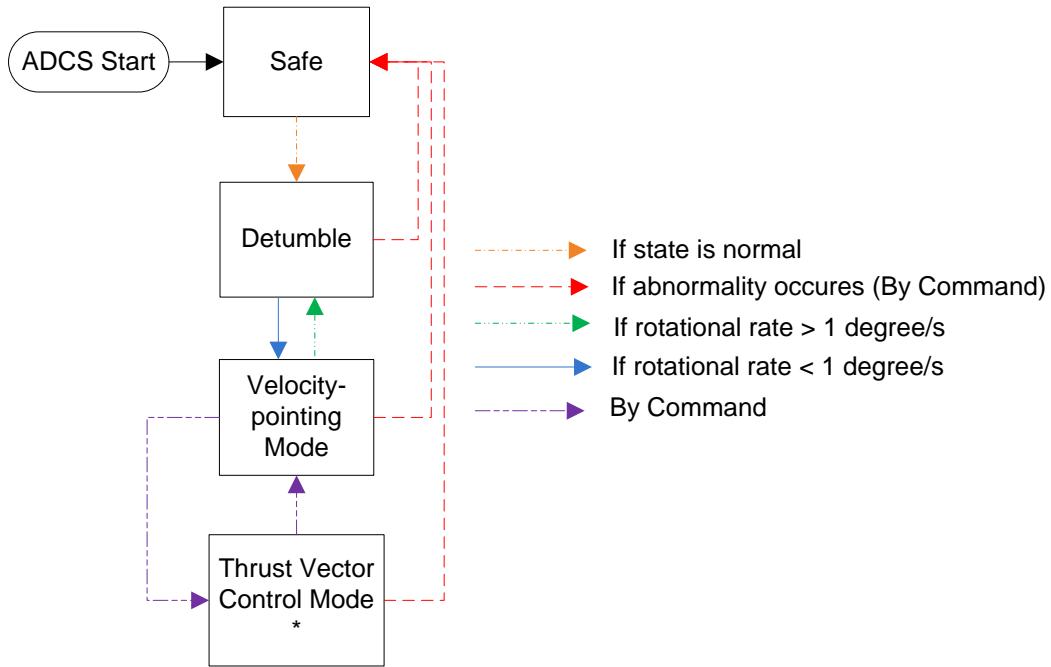


Figure 2-3: Mode Switching Logic for the Attitude Determination and Control algorithms of the DeFFi satellites (* = this mode will not be discussed in this thesis)

2-5 Timing Sequence of the ADCS

The attitude determination and control algorithms will only be a part of the ADCS. The ADCS also needs to read sensor measurements and control the actuators. For each step a certain time is allocated. The sequence of these steps is given as follows:

1. *Sensor measurement*: Here the magnetometer measurements and Sun sensor measurements will be performed. This includes the conversion of the raw Sun sensor data into Sun direction vector and the calculation of the time derivative of the geomagnetic field vector from two consecutive geomagnetic field vector measurements.
2. *Algorithm execution*: This includes the algorithms that determine the mode and the execution of the control mode algorithms.
3. *Actuating magnetic torquer*: The magnetic torquer will be turned on in this step. Depending on the duty cycle provided by the algorithm, the magnetic torquers will be turned on for a specific amount of time. The remaining time the magnetic torquers will be turned off. Nevertheless the time for this step is fixed.
4. *Hold*: This step has been added to account for the magnetic hysteresis of the magnetic torquer. All other activities that are not related to the magnetic torquer can be performed in this step, such as storing housekeeping data.

In chapter 5, the time of each step will be carefully evaluated for the detumble mode. These timings will then also be used for the velocity-pointing mode.

2-6 Overview of the ADC Algorithms and the Simulation Environment

The overview of the ADC algorithms and the simulation environment is given in figure 2-4. This figure gives the data flow of the ADCS where the algorithms and the simulation environment will interact with each other. As can be seen in the figure, there are two control modes, namely the detumble mode and the velocity-pointing mode. The third mode, namely the safe mode, is not depicted in the picture. The safe mode is characterized by a single variable which will indicate whether the algorithm will continue with one of the control modes after mode determination or it will skip the control modes and wait for next loop. Note that the Thrust Vector Control mode has not been displayed in this overview.

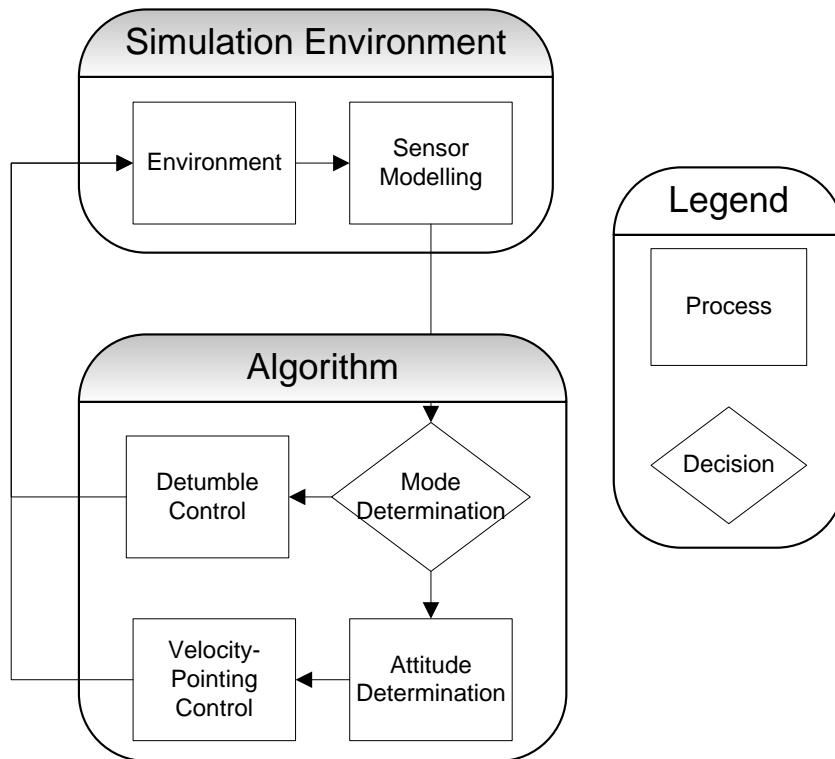


Figure 2-4: overview of the ADC algorithms and the simulation environment

Chapter 3

Dynamics and Kinematic Models and Environment Modelling

This chapter will discuss the spacecraft and the environment model. First the reference frames used throughout this thesis and the reference frame transformation matrices will be discussed which are needed for the discussion of the spacecraft dynamics and environment models.

3-1 Common Spacecraft Reference Frames

This chapter will describe the reference frames that will be used throughout this report.

Earth Centered Inertial Reference Frame

This frame is abbreviated to I (or ECI) reference frame. The origin of the I frame lies in the center of the Earth. It is represented by the x axis pointing from the center of the Earth to the Vernal Equinox. The Vernal Equinox is also called the first point of Aries. It is the point where the Sun's rays are perpendicular to the Earth's surface at the location on Earth which is closest to the Sun at the start of spring. It can also be defined as the point where the ecliptic plane and the equatorial plane cross. The z axis points from the center of Earth to the North along the spin axis of the Earth which goes through the poles of the Earth. The y axis completes the right handed coordinate system. This reference frame has been visualised in figure 3-1.

Earth Centered Earth Fixed Reference Frame

This frame is abbreviated to E (or ECEF) reference frame. This reference frame has its origin in the center of Earth as well. The x axis points from the center of the Earth to the point where the Greenwich Meridian and the equatorial plane cross. The z axis points along the

same axis as the z axis defined in the I reference frame. The y axis completes the right handed coordinate system. This reference frame has been visualised in figure 3-2.

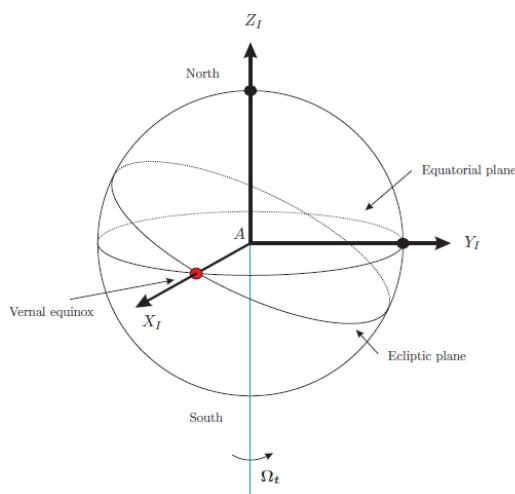


Figure 3-1: ECI reference frame [3]

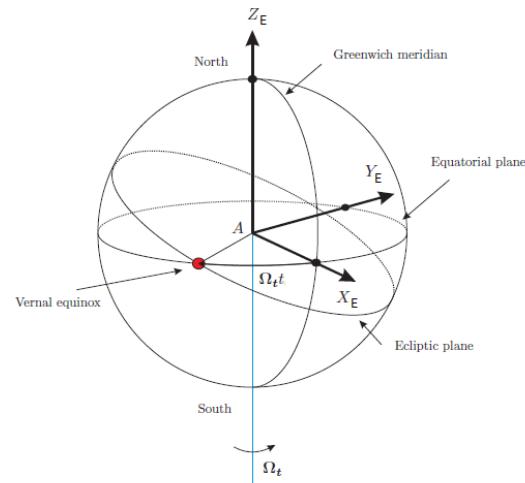


Figure 3-2: ECEF reference frame [3]

Satellite Orbit Fixed Reference Frame

This frame is abbreviated to O reference frame. The origin of this reference frame is the center of mass of the satellite. The y axis is pointing towards the negative orbital plane normal direction. The z axis points towards the nadir direction. The x axis completes the right handed coordinate system.

Satellite Body Fixed Reference Frame

This frame is abbreviated to B reference frame. The origin of this reference frame is located at the center of mass of the satellite. The x axis is directed along the major axis of the satellite. The positive direction is defined as the direction that is opposite to the ideal thrust force. The z axis is pointing from the center of mass of the satellite along the axis perpendicular to the plane where the solar panels are located on the 'wings' of the satellite. The positive z direction is the direction that the solar panels are facing. The y axis completes this right handed coordinate system. This has been visualised in figure 2-1.

Sun Pointing Reference Frame

This frame is abbreviated to S reference frame. The origin of this reference frame is the center of mass of the satellite. The z axis points towards the Sun. The y axis points towards the projection of the y axis of the I frame on the plane perpendicular to the z axis of the S frame. The x axis completes this right handed coordinate system. A visualisation of this reference frame with respect to the Earth-Centred Inertial reference frame is given in figure 3-3.

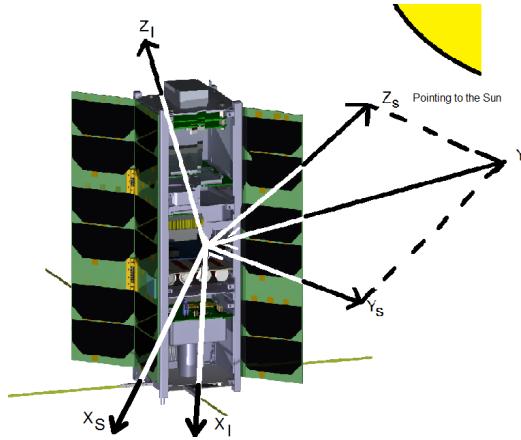


Figure 3-3: Sun pointing reference frame

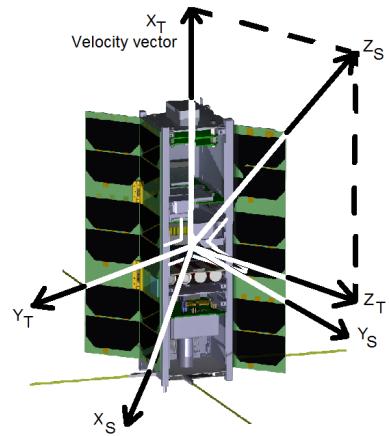


Figure 3-4: Thrust pointing reference frame

Thrust Pointing Reference frame

This frame is abbreviated to T reference frame. This reference frame has its origin at the center of mass of the satellite. The x axis points towards the velocity vector. The z axis points towards the projection of the z axis of the S frame on the plane perpendicular to the x axis of the T frame. The y axis completes this right handed coordinate system. A visualisation of this reference frame with respect to the Sun pointing reference frame is given in figure 3-4.

3-2 Transformation Between Reference Frames

The transformation between reference frames is important for both the simulation environment and using it on the algorithm of the attitude determination and control subsystem. These transformations will specify the attitude of the reference frames with respect to the Earth Centered Inertial reference frame (I frame).

In this section the inverse of the transformation matrices can be performed by taking the transpose of the matrices. This can be explained by the fact that the columns of transformation matrices are linearly independent and therefore a transformation matrix is an orthogonal matrix, as stated by the invertible matrix theorem [13]. In linear algebra, an orthogonal matrix is a square matrix with real values and whose rows and columns are orthogonal unit vectors, also called orthonormal vectors. Multiplying an orthogonal matrix with its transpose gives the identity matrix (equation 3-1, where A is an orthogonal transformation matrix and I is the identity matrix). This leads to the result that the transpose of an orthogonal matrix is equivalent to the inverse of the orthogonal matrix.

$$\begin{aligned} \mathbf{A}^T \mathbf{A} &= \mathbf{A} \mathbf{A}^T = \mathbf{I} \\ \mathbf{A}^T &= \mathbf{A}^{-1} \end{aligned} \tag{3-1}$$

It should be noted that the graphs used to illustrate the transformation of the reference frames in the coming three sections are not showing the vectors to scale.

3-2-1 Transformation ECI frame to ECEF frame

As illustrated in figure 3-2 and explained in section 3-1, the z axis, which is the spin axis of the Earth, of the Earth Centered Earth Fixed (E frame) and Earth Centered Inertial (I frame) reference frame are equal to each other. Rotation around the z axis will be sufficient to align the x and y axis of the I frame with the x and y axis of the E frame. This rotation angle is a function of the time. The E and I frame are aligned once every sidereal day, hence the rotation angle is defined by:

$$\varphi = \Omega_t t = \frac{2\pi t}{T_{sidereal}} \quad (3-2)$$

Where t is the time since the last alignment of the E and I frame and $T_{sidereal}$ of the Earth is equal to 86163.9 seconds. The sidereal time, or the sidereal day length, of the Earth is the time it takes for the Earth to complete one rotation around its axis.

Rotating around the z axis with this angle gives:

$$\mathbf{T}_{EI} = \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-3)$$

This transformation has been used to rotate the vector that specifies the position of the satellite with respect to the center of the Earth expressed in I frame, \mathbf{r}_I , to the E frame, \mathbf{r}_E and to rotate the geomagnetic field vector expressed in the E frame to the I frame, using the transpose.

3-2-2 Transformation ECI frame to T frame

In order to describe the Thrust-pointing frame (T frame) in the I frame the velocity vector of the satellite expressed in the I frame, \mathbf{V}_I , is needed. To align the I frame with the T frame 2 successive rotations will be performed around the x and z axis. The rotation sequence is described as follows:

1. The first rotation around the x axis of the I frame with angle φ rotates the y axis of the I frame to the y axis of the intermediate frame, I_* , to align it with the projection of the negative velocity vector on the yz plane of the I frame (which is also the yz plane of the I_* frame). This is visualised in figure 3-5.

$$\varphi = \arccos 2 \left(\frac{-V_{I,y}}{\sqrt{V_{I,y}^2 + V_{I,z}^2}}, \operatorname{sign}(-V_{I,z}) \right) \quad (3-4)$$

2. The second rotation around the z axis of the I_* frame rotates the x axis of the I_* frame to the x axis of the T frame to align it with the positive velocity vector. This is visualised in figure 3-6 .

$$\theta = \arccos \left(\frac{V_{I,x}}{\|\mathbf{V}_I\|} \right) \quad (3-5)$$

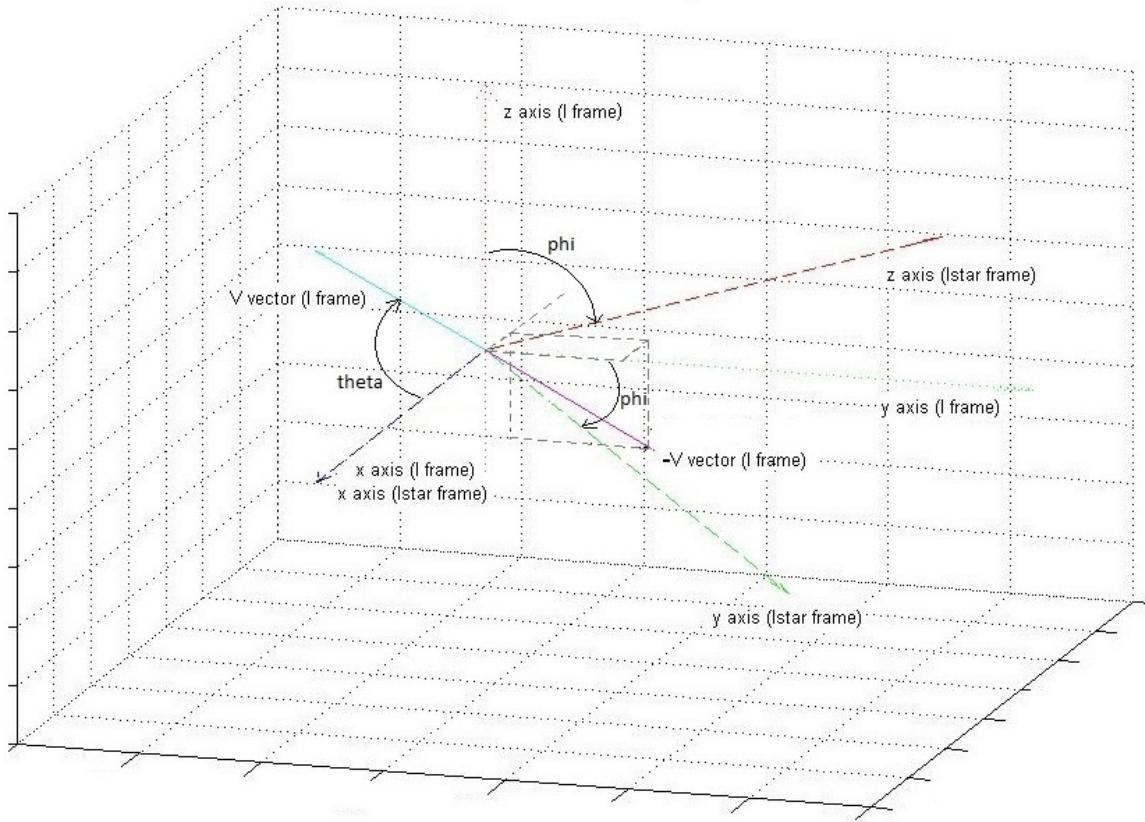


Figure 3-5: First rotation for the transformation from ECI reference frame to Thrust-pointing reference frame

3. The transformation matrix of the I frame to the T frame is then defined as:

$$\mathbf{T}_{TI} = \mathbf{T}_z(\theta) \mathbf{T}_x(\varphi) \quad (3-6)$$

This transformation matrix, also called the Direction Cosine Matrix will be used to determine the direction in which the satellite needs to point its thrusters for changing the orbit.

Semi Sun-pointing during Velocity-Pointing Mode

For the DelFFi it has been decided to align the satellite to the velocity-pointing direction during the nominal mode. This means that the satellite will never be able to fully point to the Sun for an optimal Sun angle which results into maximum power from the solar panels mounted on the satellite. However, it is possible to rotate around the velocity-pointing axis in a way that the solar panels are receiving maximum amount of power. This will not change the velocity-pointing behaviour of the satellite. This technique will be called *Semi Sun-pointing*. The working principle of the Semi Sun-pointing mode is that it changes the desired attitude of the satellite, without changing the velocity-pointing attitude of the satellite. Therefore Semi Sun-pointing will be performed by modifying the desired attitude of the satellite as obtained

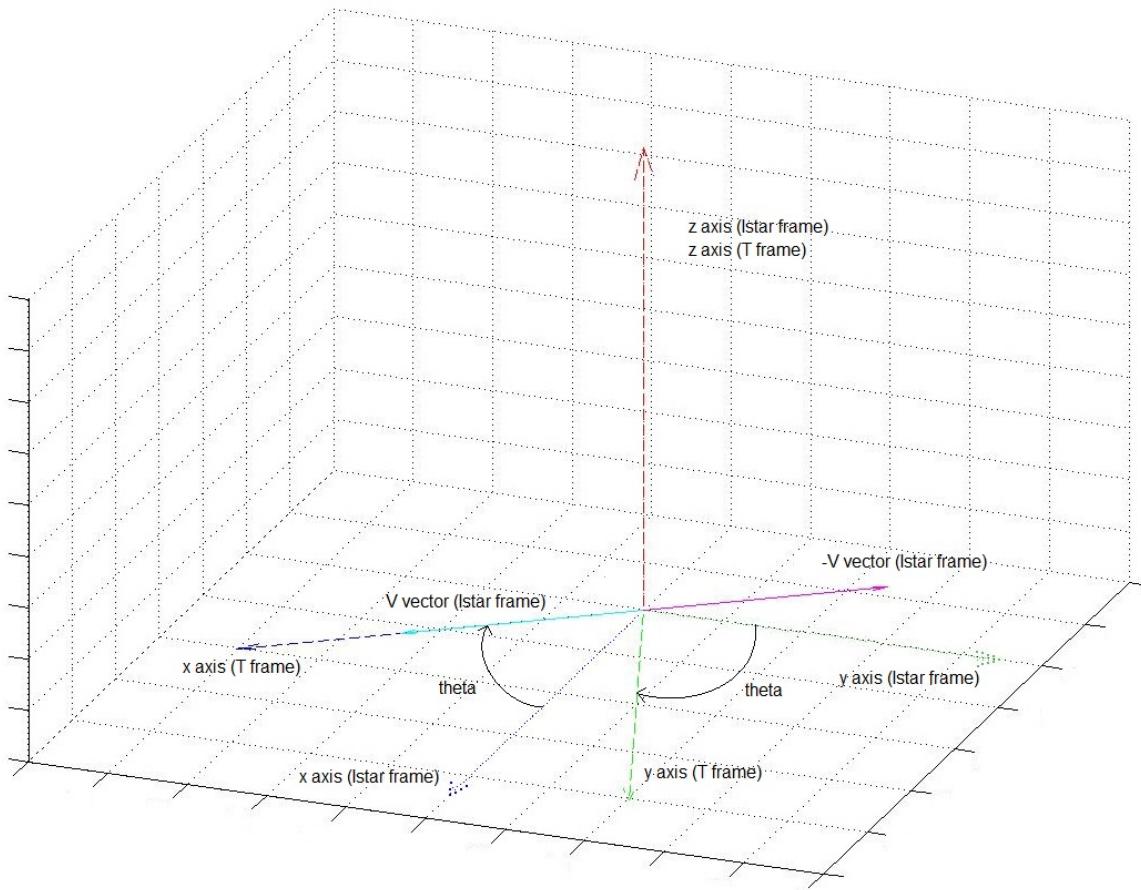


Figure 3-6: Second rotation for the transformation from ECI reference frame to Thrust-pointing reference frame

from equation 3-6. The procedure to change the desired attitude in order to add the Semi Sun-pointing is given as follows:

- First, the Sun vector will be expressed in the thrust pointing reference frame:

$$\mathbf{S}_T = \mathbf{T}_{TI} \mathbf{S}_I \quad (3-7)$$

- Secondly, the rotation angle around the x axis will be calculated such that the z axis of the satellite will point to the Sun vector expressed in the yz plane of the thrust pointing reference frame.

$$\psi = \arccos 2 \left(\frac{S_{T,3}}{\sqrt{S_{T,2}^2 + S_{T,3}^2}}, -\text{sign}(S_{T,2}) \right) \quad (3-8)$$

- Finally, the new rotation matrix will be defined with which the newly defined desired attitude quaternion can be determined.

$$\mathbf{T}_{SI} = \mathbf{T}_x(\psi) \mathbf{T}_{TI} \quad (3-9)$$

Nadir-pointing during Velocity-pointing Mode

In the same way as described in the previous paragraph by rotating around the roll axis for pointing towards the Sun, the satellite can be pointed towards the Earth, while still keeping the velocity vector and the positive x axis of the satellite aligned with each other. The approach is analogous to the approach described in the previous paragraph. The difference is in the reference vector. Instead of the Sun vector, the Earth vector will be used. The Earth vector is the unit vector that points from the Earth to the satellite. This vector can be obtained from the SGP4 model on-board the satellite in the same way as the velocity vector has been obtained.

3-2-3 Transformation ECI frame to O frame

In order to describe the orbital frame (O frame) in the I frame, two vectors are needed, the position of the satellite with respect to the center of the Earth expressed in the I frame, \mathbf{r}_I and the velocity vector of the satellite expressed in the I frame, \mathbf{V}_I . To align the I frame with the orbital frame 3 successive rotations will be performed around the z, y and again z axis (323 rotation sequence). The rotation sequence is described as follows:

1. The first rotation around the z axis of the I frame with an angle φ rotates the x axis of the I frame to the x axis of the I_* frame to align it with the projection of the negative position vector, $-\mathbf{r}$ vector, of the satellite on the xy plane of the I frame. This rotation is given in the following equation and a visualisation is given in figure 3-7.

$$\varphi = \arccos 2 \left(\frac{-r_{I,x}}{\sqrt{r_{I,x}^2 + r_{I,y}^2}}, \operatorname{sign}(-r_{I,y}) \right) \quad (3-10)$$

The newly defined frame is denoted as I_* frame. The angle φ is calculated by using the arccos2 function which is defined in appendix B

2. The second rotation around the y axis of the intermediate frame I_* with an angle ϑ rotates the z axis of the I_* frame to the z axis of the I_{**} frame to align it with the negative position vector, $-\mathbf{r}$ vector. This rotation is given in the following equation and a visualisation in figure 3-8.

$$\theta = \arccos \left(\frac{-r_{I,z}}{\|\mathbf{r}_I\|} \right) \quad (3-11)$$

The newly defined frame is denoted as I_{**} frame.

3. The third rotation around the z axis of the intermediate frame I_{**} with an angle ψ aligns the x axis of the O frame with the projection of the velocity vector, \mathbf{V}_I , on the xy plane of the I_{**} frame, defined as \mathbf{V}_{**} . This rotation angle is given in the following equation and a visualisation is given in figure 3-9.

$$\psi = \arccos 2 \left(\frac{V_{**,x}}{\sqrt{V_{**,x}^2 + V_{**,y}^2}}, \operatorname{sign}(V_{**,y}) \right) \quad (3-12)$$

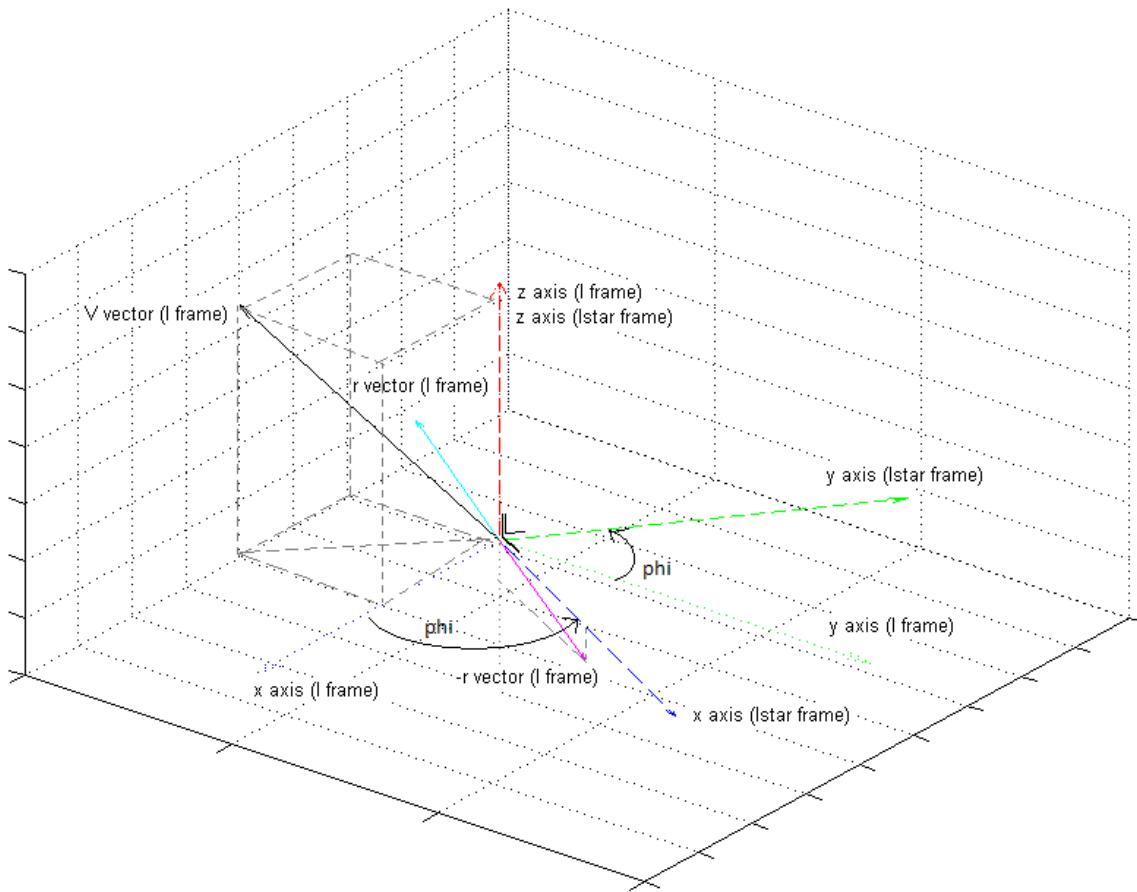


Figure 3-7: First rotation for the transformation from ECI reference frame to Orbit reference frame

Where the vector \mathbf{V}_{**} is defined as:

$$\mathbf{V}_{**} = \mathbf{T}_y(\theta) \mathbf{T}_z(\varphi) \mathbf{V}_I \quad (3-13)$$

4. The transformation matrix of the I frame to the O frame is then defined as:

$$\mathbf{T}_{OI} = \mathbf{T}_z(\psi) \mathbf{T}_y(\theta) \mathbf{T}_z(\varphi) \quad (3-14)$$

This transformation matrix, also called the Direction Cosine Matrix, will be used to simulate the aerodynamic and the gravity gradient disturbance torque.

3-3 Rotational Kinematics

This section will focus on describing the orientation of the satellite that is in a rotational motion. One reference frame can be expressed with respect to another reference frame in multiple ways, namely Direction Cosine Matrix, euler angles and quaternions. In the second

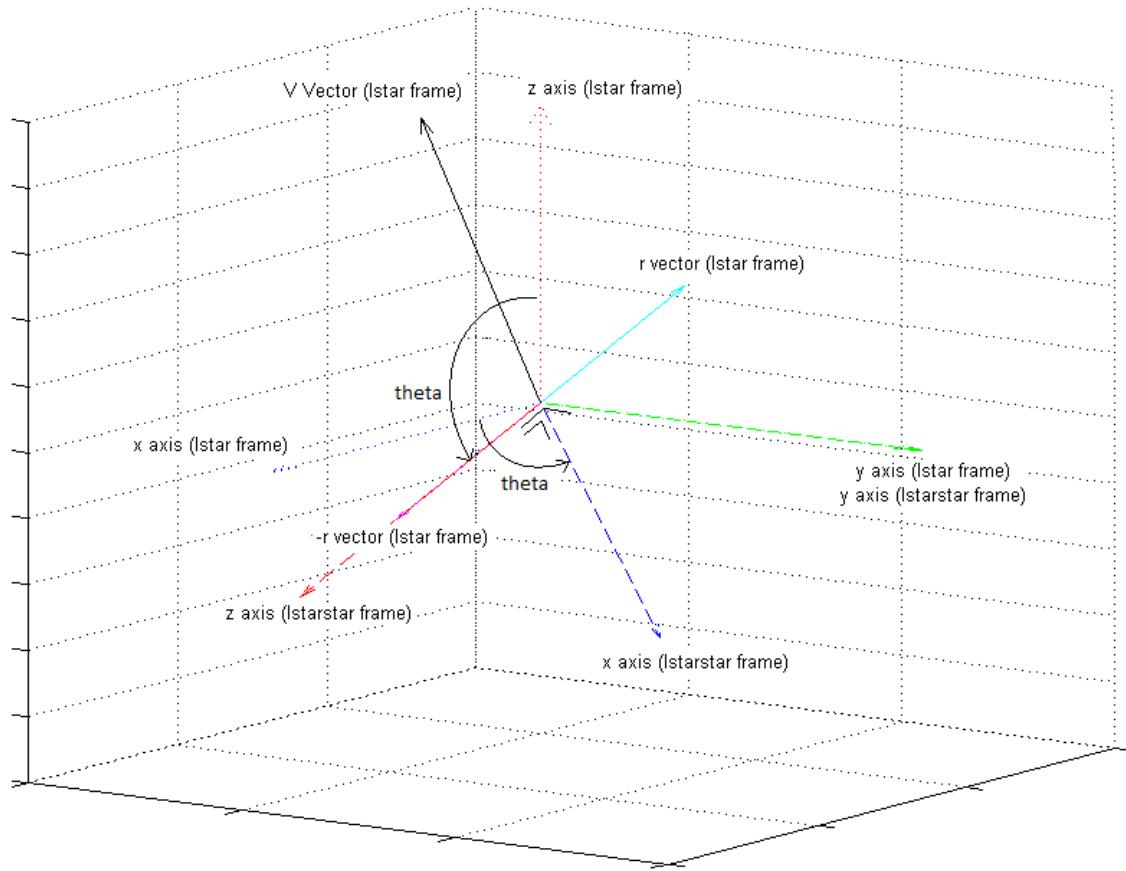


Figure 3-8: Second rotation for the transformation from ECI reference frame to Orbit reference frame

part of this section, the attitude of the satellite will be described when it is under the influence of a rotational motion. This is important for propagating the attitude of the satellite in the attitude simulator code programmed in Matlab/Simulink.

3-3-1 Direction Cosine Matrix

Using the Direction Cosine Matrix (DCM) that describes the rotation from an arbitrary reference frame M to N, a vector that is described in reference frame M can be converted to a vector expressed in reference frame N. This is described in the following equation:

$$\mathbf{v}_N = \mathbf{T}_{MN} \mathbf{v}_M$$

$$\begin{bmatrix} v_{N,1} \\ v_{N,2} \\ v_{N,3} \end{bmatrix} = \begin{bmatrix} T_{MN,11} & T_{MN,12} & T_{MN,13} \\ T_{MN,21} & T_{MN,22} & T_{MN,23} \\ T_{MN,31} & T_{MN,32} & T_{MN,33} \end{bmatrix} \begin{bmatrix} v_{M,1} \\ v_{M,2} \\ v_{M,3} \end{bmatrix} \quad (3-15)$$

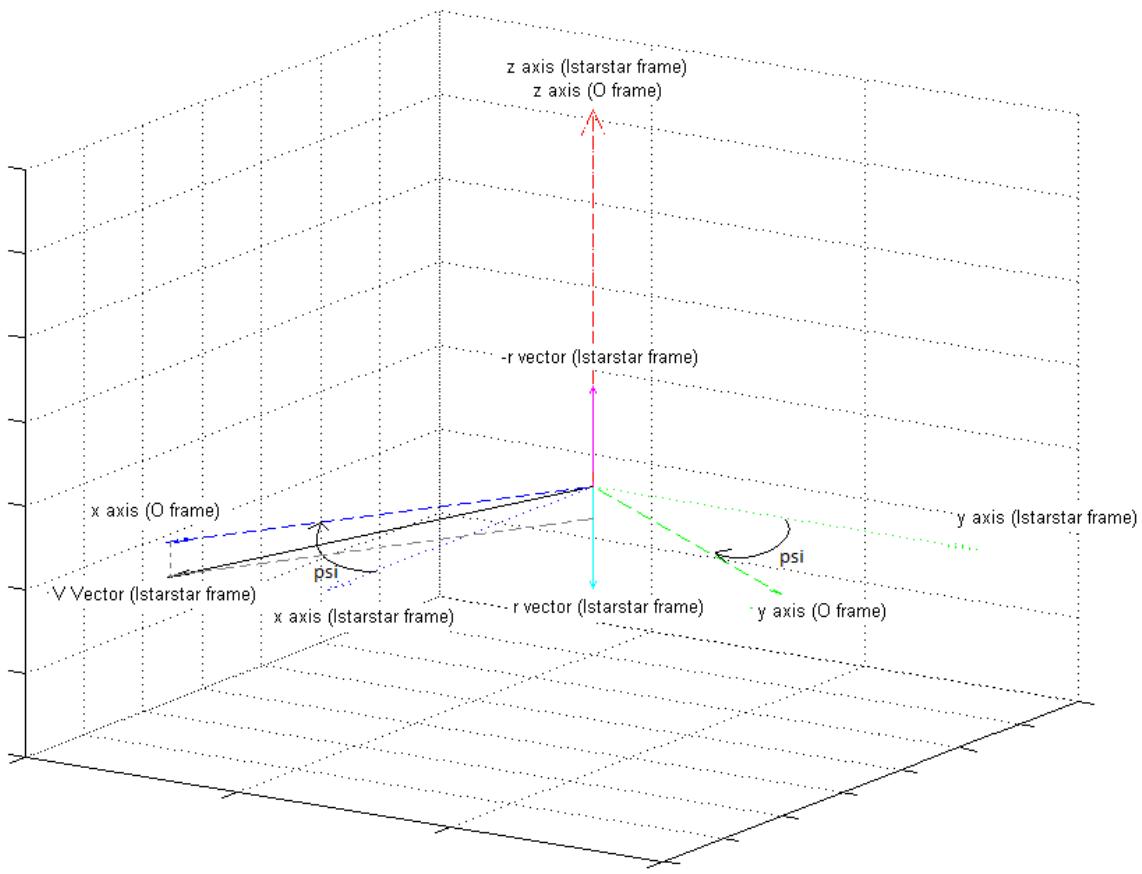


Figure 3-9: Third rotation for the transformation from ECI reference frame to Orbit reference frame

Three elementary rotations respectively about the first, second and third axes (x, y and z respectively) of the reference frame M are described by the following rotation matrices:

$$\mathbf{T}_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi \\ 0 & -\sin \varphi & \cos \varphi \end{bmatrix} \quad (3-16)$$

$$\mathbf{T}_y(\theta) = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (3-17)$$

$$\mathbf{T}_z(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-18)$$

3-3-2 Euler Angles

The rotation from the arbitrary reference frame N to N can also be expressed in terms of Euler angles. Euler angles are three angles which will rotate the reference frame M to N using three

successive rotations with the aforementioned three angles. If the satellite body axis performs three rotations in a 321 rotation sequence, namely yaw, pitch and roll, then the DCM can be obtained from these three Euler angles in the following way:

$$\mathbf{T}_{MN} = \mathbf{T}_x(\varphi) \mathbf{T}_y(\theta) \mathbf{T}_z(\psi) \quad (3-19)$$

In general the rotation sequence can start about any axis, followed by one of the other axis and ended with a rotation about one of the axis which is not the same as the second axis rotation.

3-3-3 Quaternion

It is possible to rotate any reference frame to another reference through only one rotation about a certain defined axis. This axis will be defined as the Euler eigenaxis. One special property of this eigenaxis which follows from its definition is that it has the same direction in both M and N reference frame. The angle with which one reference frame needs to be rotated about the Euler eigenaxis to obtain the other reference frame is denoted as θ . This rotation can be expressed as a quaternion defined as:

$$q = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} e_1 \sin \frac{\theta}{2} \\ e_2 \sin \frac{\theta}{2} \\ e_3 \sin \frac{\theta}{2} \\ \cos \frac{\theta}{2} \end{bmatrix} \quad (3-20)$$

Where $\begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}^T$ is the Euler eigenaxis and θ is the angle with which is rotated about the Euler eigenaxis.

The quaternion parameters are not independent of each other, but are constrained by the relationship:

$$\mathbf{q}^T \mathbf{q} + q_4^2 = 1 \quad (3-21)$$

Another property of the quaternion is that \mathbf{q} and $-\mathbf{q}$ describe the same rotation. The prove of these properties will not be treated in this thesis, but can be found in [14] and [15]. The DCM can be described as a quaternion:

$$\mathbf{T}_{MN} = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix} \quad (3-22)$$

And vice versa:

$$q_4 = \frac{1}{2}\sqrt{(1 + T_{MN,11} + T_{MN,22} + T_{MN,33})}$$

$$\mathbf{q} = \frac{1}{4q_4} \begin{bmatrix} T_{MN,23} - T_{MN,32} \\ T_{MN,31} - T_{MN,13} \\ T_{MN,12} - T_{MN,21} \end{bmatrix} \quad (3-23)$$

3-3-4 Kinematic Equation

For a correct simulation of the attitude, the attitude needs to be propagated based on a certain rotational motion acting on the satellite. Let the body fixed reference frame (frame B) be rotating with respect to the Earth-Centered Inertial reference frame (frame I) with an angular velocity of ω_{BI} . Using equation 3-15, a vector in the B frame can be transformed to the I frame:

$$\mathbf{v}_I = \mathbf{T}_{BI}^T \mathbf{v}_B \quad (3-24)$$

Taking the time derivative of this equation gives:

$$\mathbf{0} = \dot{\mathbf{T}}_{BI}^T \mathbf{v}_B + \mathbf{T}_{BI}^T \dot{\mathbf{v}}_B \quad (3-25)$$

This can be converted to:

$$\mathbf{0} = (\dot{\mathbf{T}}_{BI}^T - \mathbf{T}_{BI}^T \Omega^*) \mathbf{v}_B = \dot{\mathbf{T}}_{BI} \mathbf{T}_{BI}^T + \Omega^* \quad (3-26)$$

Where Ω^* is the skew-symmetric matrix defined as:

$$\Omega^* = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (3-27)$$

Equation 3-26 can be rewritten in:

$$\boldsymbol{\omega} = \begin{bmatrix} \dot{\mathbf{T}}_{21} \mathbf{T}_{31} + \dot{\mathbf{T}}_{22} \mathbf{T}_{32} + \dot{\mathbf{T}}_{23} \mathbf{T}_{33} \\ \dot{\mathbf{T}}_{31} \mathbf{T}_{11} + \dot{\mathbf{T}}_{32} \mathbf{T}_{12} + \dot{\mathbf{T}}_{33} \mathbf{T}_{13} \\ \dot{\mathbf{T}}_{11} \mathbf{T}_{21} + \dot{\mathbf{T}}_{12} \mathbf{T}_{22} + \dot{\mathbf{T}}_{13} \mathbf{T}_{23} \end{bmatrix} \quad (3-28)$$

Substituting equation 3-22 and adding the time derivative of equation 3-21 into equation 3-28, gives:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix} = \begin{bmatrix} 2(\dot{q}_1 q_4 + \dot{q}_2 q_3 - \dot{q}_3 q_2 - \dot{q}_4 q_1) \\ 2(\dot{q}_2 q_4 + \dot{q}_3 q_1 - \dot{q}_1 q_3 - \dot{q}_4 q_2) \\ 2(\dot{q}_3 q_4 + \dot{q}_1 q_2 - \dot{q}_2 q_1 - \dot{q}_4 q_3) \\ 2(\dot{q}_1 q_1 + \dot{q}_2 q_2 - \dot{q}_3 q_3 - \dot{q}_4 q_4) \end{bmatrix} \quad (3-29)$$

Rewriting this equation in matrix form gives:

$$\begin{bmatrix} \boldsymbol{\omega} \\ 0 \end{bmatrix} = 2 \begin{bmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \\ q_1 & q_2 & q_3 & q_4 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} \quad (3-30)$$

And rewriting this equation again gives the Kinematic Equation:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \Omega \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \quad (3-31)$$

Where Ω is defined as:

$$\Omega = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \quad (3-32)$$

3-4 Euler's Dynamic Equation

This chapter will discuss the rotational dynamics of the satellite. This behaviour of the satellite will be simulated in the attitude simulator to determine the rotational motion of the satellite based on a certain torque applied to the satellite. In addition, the rotational dynamics will also be implemented in the Attitude Determination algorithm to predict the rotational motion of the satellite. The angular momentum of a rigid body can be calculated with [14]

$$\mathbf{H} = \mathbf{J}\boldsymbol{\omega} \quad (3-33)$$

Where \mathbf{H} is the angular momentum, \mathbf{J} is the moment of inertia of the DelFFi satellite and $\boldsymbol{\omega}$ is the angular velocity of the satellite's body with respect to the Earth-Centered Inertial reference frame. The inertial time derivative of the angular momentum is:

$$\dot{\mathbf{H}} = \mathbf{T} \quad (3-34)$$

Where \mathbf{T} is the torque exerted on the satellite about its center of mass. This torque consists of all the external torques, namely the disturbance torques and the control torques. The change in angular momentum can also be written as:

$$\dot{\mathbf{H}} = \dot{\mathbf{H}}_I = \dot{\mathbf{H}}_B + \boldsymbol{\omega} \times \mathbf{H} \quad (3-35)$$

Substituting the equation of the angular momentum of a rigid body and the time derivative of the angular momentum of a rigid body (equation 3-33) into the previous equation gives:

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} = \mathbf{T} \quad (3-36)$$

Rewriting this equation gives Euler's Dynamic Equation which describes the rotational dynamics of a rigid body:

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1} (\mathbf{T} - \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}) \quad (3-37)$$

3-5 Spacecraft Orbit Dynamics

The position and velocity of the satellite will be simulated by varying the time t for a fixed semi-major axis a , eccentricity e , inclination i , right ascension of the ascending node Ω and argument of perigee ω . From these variables, the position and velocity vector will be determined as follows [16]

1. The mean anomaly will be computed:

$$M = n \cdot t \quad (3-38)$$

Where n is the orbital speed defined as: $n = \sqrt{\frac{\mu}{a^3}}$

2. Assuming that the eccentricity anomaly E is equal to the mean anomaly M , the true anomaly θ can be computed:

$$\tan\left(\frac{\theta}{2}\right) = \sqrt{\frac{(1+e)}{(1-e)}} \tan \frac{E}{2} \quad (3-39)$$

3. The distance between the Earth's center and the satellite is computed as follows:

$$r = a(1 - e \cdot \cos(E)) \quad (3-40)$$

4. The position components in the inertial reference frame, which define the distance between the satellite and the center of the Earth, are then defined as:

$$\begin{aligned} r_{I,x} &= r ((\cos \Omega \cos \omega_{ap} - \sin \Omega \sin \omega_{ap} \cos i) \cos \theta + (-\cos \Omega \sin \omega_{ap} - \sin \Omega \cos \omega_{ap} \cos i) \sin \theta) \\ r_{I,y} &= r ((\sin \Omega \cos \omega_{ap} + \cos \Omega \sin \omega_{ap} \cos i) \cos \theta + (-\sin \Omega \sin \omega_{ap} + \cos \Omega \cos \omega_{ap} \cos i) \sin \theta) \\ r_{I,z} &= r (\sin \omega_{ap} \sin i \cos \theta + \cos \Omega \sin i \sin \theta) \end{aligned} \quad (3-41)$$

3-6 Environment Modelling

This section will discuss the environment models. Firstly, the model that determines the geomagnetic field vector will be discussed. Secondly, the model that determines the Sun vector will be discussed. Finally, the disturbance torques acting on the satellite will be discussed. It should be noted that due to the low orbit of the satellite, the solar radiation pressure will be considered to be negligible [7].

3-6-1 Model of the Geomagnetic Field

This section will describe the two models that will be used for simulating the geomagnetic field vector. In the next section the geomagnetic field will be modelled as a dipole. This will give sufficient accuracies for use in the simulations. In the section after that the IGRF model will be introduced. This model will be used on-board the satellite as a look-up table to use in the Extended Kalman Filter.

Geomagnetic Field model used for Simulations

For modelling the geomagnetic field vector, the magnetic field will be modelled as a dipole which provides a first order approximation. The dipole model of the geomagnetic field is visualised in figure 3-10. According to the dipole moment the geomagnetic field vector is equal to the negative gradient of the scalar potential function V multiplied by the magnetic permeability of free space, μ_0 . This relation is given in the following equation:

$$\mathbf{B} = -\nabla V \cdot \mu_0 \quad (3-42)$$

The potential function of the geomagnetic field vector can be described by:

$$V(\mathbf{r}) = \frac{1}{4\pi r^3} \mathbf{m} \cdot \mathbf{r}_E \quad (3-43)$$

Where \mathbf{m} is the magnetic dipole moment of the Earth, \mathbf{r}_E is the position in the ECEF frame expressed in spherical coordinates where the potential function will be calculated given in equation 3-44.

$$\mathbf{r}_E = \begin{bmatrix} r \\ \theta \\ \phi \end{bmatrix} \quad (3-44)$$

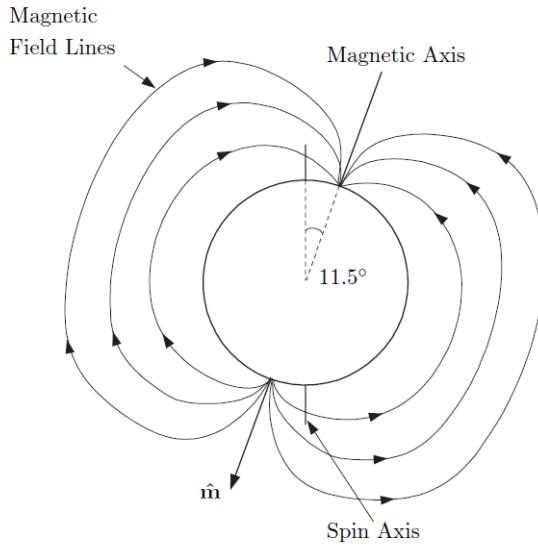


Figure 3-10: Geomagnetic Field represented as a dipole [4]

Where θ is the colatitude measured positive from the North Pole and ϕ is the longitude as measured from the Greenwich meridian.

Inserting equation 3-43 into equation 3-42 gives the geomagnetic field vector expressed in the ECEF frame as spherical coordinates. This equation is given as follows:

$$\begin{bmatrix} B_r \\ B_\theta \\ B_\phi \end{bmatrix} = \frac{\mu_0 m}{4\pi r^3} \begin{bmatrix} 2 \cos(\theta) \\ \sin(\theta) \\ 0 \end{bmatrix} \quad (3-45)$$

Where the magnetic dipole moment of the Earth is assumed to be the same in all the three axes. As can be seen from equation 3-45, the geomagnetic field in the ϕ component of the spherical coordinates is zero. This can be explained by the geomagnetic field being independent of the longitude due to the fact that the geomagnetic field has been modelled using a dipole model. In figure 3-10 this independence between the geomagnetic field and the longitude is visualised.

Converting the values of the geomagnetic field vector from spherical coordinates to cartesian components gives:

$$\mathbf{B} = \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} = \begin{bmatrix} (B_r \cos(\delta) - B_\theta \sin(\delta)) \cos(\phi) - B_\phi \sin(\phi) \\ (B_r \cos(\delta) - B_\theta \sin(\delta)) \sin(\phi) - B_\phi \sin(\phi) \\ B_r \sin(\delta) + B_\theta \cos(\delta) \end{bmatrix} \quad (3-46)$$

Where δ is the latitude measured positive from the North Pole.

Geomagnetic Field model used on-board DelFFi

For the geomagnetic field determination on-board of the satellite the Inertial Geomagnetic Reference Frame (IGRF) data will be used. This model is more accurate than the dipole

model, but this is required for the high demands set by the Kalman Filter as discussed in section 4-2. Since the model of the IGRF is computationally intensive, the data from the IGRF is stored on-board of the satellite using a look-up table. The look-up table will contain as inputs the radius, the longitude and the latitude of the position of the satellite. As an output, the IGRF look-up table provides the x, y and z components of the geomagnetic field expressed in the ECEF frame. The position of the satellite will be computed by the SGP4 orbit propagator using Two-Line Element data.

3-6-2 Model of the Sun's position

The Sun model described in this section will provide the direction of the Sun in the ECI frame (I frame) for any give time since the last alignment of the Sun direction and the Vernal Equinox. The Vernal Equinox is defined as the start of Spring (in the Northern Hemisphere). Similarly, the Vernal equinox is located at the point where the elliptic plane and the equatorial plane intersect. The model of the Sun is given in the following equation [17]:

$$\mathbf{S}_I = \begin{bmatrix} \cos\left(\frac{2\pi(t-t_{ve})}{T_{year}}\right) \\ \cos(\varepsilon) \cdot \sin\left(\frac{2\pi(t-t_{ve})}{T_{year}}\right) \\ \sin(\varepsilon) \cdot \sin\left(\frac{2\pi(t-t_{ve})}{T_{year}}\right) \end{bmatrix} \quad (3-47)$$

Where $t-t_{ve}$ is the time since the last alignment of the Sun direction with the Vernal Equinox given in seconds, T_{year} is the amount of seconds in one year equal to 31557600 seconds, and ε is the angle between the ecliptic plane and the equatorial plane, which is approximated to be 23.440 degrees.

Modelling the Eclipse Periods

For modelling the occurrence of the eclipse periods, the position vector of the satellite, \mathbf{r}_I , and the Sun direction vector, \mathbf{S}_I , need to be known in the ECI frame (I frame). Note that in this model the Sun is assumed that the Sun direction is the same for all positions in the orbit due to the fact that the Sun is assumed to be far away and therefore its rays are received parallel.

1. First the maximum angle between \mathbf{r}_I and \mathbf{S}_I is determined for which the satellite is in eclipse. This is a constant and is given in the following equation:

$$\gamma_E = \arcsin\left(\frac{R_{Earth}}{R_{Earth} + h_{altitude}}\right) \quad (3-48)$$

Where R_{Earth} is the radius of the Earth and $h_{altitude}$ is the altitude of the satellite.

2. Secondly the real angle between the \mathbf{r}_I and \mathbf{S}_I is determined using a dot produc:

$$\alpha = \arccos\left(\frac{\mathbf{r}_I \cdot -\mathbf{S}_I}{\|\mathbf{r}_I\|}\right) \quad (3-49)$$

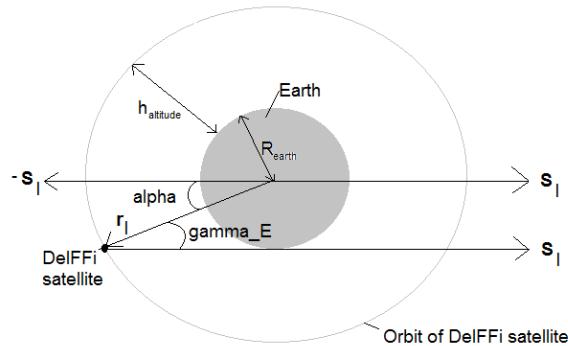


Figure 3-11: Eclipse occurrence

3. The requirement for the occurrence of eclipse is then:

$$\alpha < \gamma_E \quad (3-50)$$

Which can be explained using figure 3-11 where this problem is geometrically visualised. here it can be clearly seen that when α decreases below a constant value γ_E , then the satellite is in eclipse.

3-6-3 Model of the Disturbance Torques

This section will provide the equations for modelling of the disturbance torques.

Aerodynamic Disturbance Torque

The aerodynamic drag force is one of the major contributors to the disturbance torque. It is caused by the fact that there is still some air particles present at the altitude in which the DelFFi satellite is operating. The aerodynamic disturbance torque is determined by calculating the aerodynamic drag force in the negative x direction of the O frame, given in equation 3-51. The surface area of the satellite is considered to be maximum at all times, regardless of the orientation of the satellite with respect to the O frame, equal to $0.34 \text{ m} \cdot 0.1 \text{ m} = 0.034 \text{ m}^2$. The aerodynamic force in the O frame is therefore considered to be constant. ρ is the air density, C_D is the drag coefficient of the satellite, given in table 2-1 and V is the velocity of the satellite, given in equation 3-52, where a is the semi-major axis and T_{orb} is the orbit period. The aerodynamic drag force in the O frame is then projected onto the B frame using the attitude of the satellite with respect to the O frame expressed in the B frame (T_{bo}), given in equation 3-53. Which results in the aerodynamic moment acting on the satellite expressed in the B frame, given in equation 3-54. r_{CP} is the distance between the center of pressure and the center of gravity. This distance is the arm of the aerodynamic moment. It is considered to be maximum at all times, regardless of the orientation of the satellite and is given in table 2-1.

$$\mathbf{D}_O = \frac{1}{2} \rho S V^2 \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \quad (3-51)$$

$$V = \frac{2\pi a}{T_{orb}} \quad (3-52)$$

$$\mathbf{D}_B = T_{bo} \cdot \mathbf{D}_O \quad (3-53)$$

$$\mathbf{M}_{aero,B} = \mathbf{r}_{CP} \times \mathbf{D}_B \quad (3-54)$$

The air density, ρ , is determined using the graph which describes the relation between the altitude, solar flux and the air density, given in figure 3-12. From this graph the following values have been obtained to simulate the minimum, average and maximum values of the air density for the lowest and highest altitude, given in table 3-1.

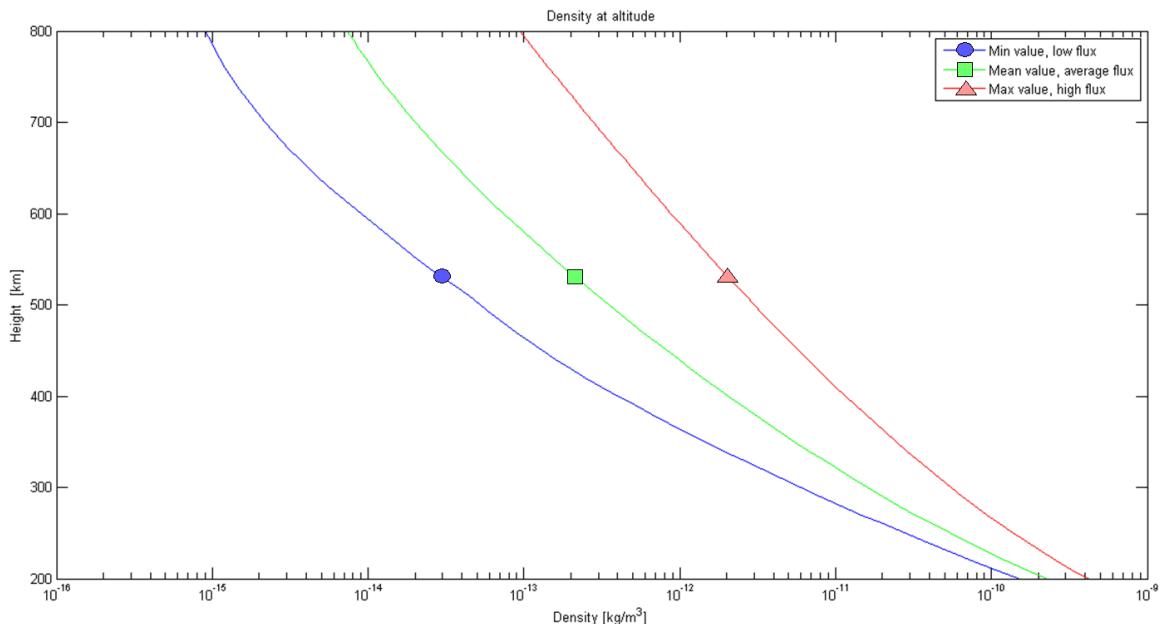


Figure 3-12: Graph of the air density versus the altitude for a varying solar flux [5]

Table 3-1: Table of the air density for varying solar flux conditions and highest and lowest altitude

Altitude [km]	Solar flux	Air density [kg/m³]
380	minimum	$8 \cdot 10^{-12}$
380	average	$9 \cdot 10^{-11}$
380	maximum	$1 \cdot 10^{-10}$
200	minimum	$0.5 \cdot 10^{-9}$
200	average	$1 \cdot 10^{-9}$
200	maximum	$3 \cdot 10^{-9}$

Magnetic Disturbance Torque

The magnetic disturbance torque is caused by the residual magnetic dipole moment originating from the electronics on-board the satellite. This torque can be calculated using the residual magnetic dipole moment and the geomagnetic field vector expressed in the B frame.

This is given in equation 3-55. Where \mathbf{M}_{res} is the magnetic disturbance torque in Nm, \mathbf{m}_{res} is the residual magnetic dipole moment in $A \cdot m^2$, given in table 2-1, and \mathbf{B}_B is the geomagnetic field vector expressed in the B frame.

$$\mathbf{M}_{res} = \mathbf{m}_{res} \times \mathbf{B}_B \quad (3-55)$$

Gravity Gradient Disturbance Torque

The gravity gradient disturbance torque is caused by the fact that the gravity of the Earth is not constant when the satellite is further away from the Earth as a result of the inverse square law. This will result in the satellite aligning its axis of minimum moment of inertia, which is the x axis for the DelFFi satellite as defined in section 3-1, along the nadir vector. The gravity gradient disturbance torque is defined as:

$$\mathbf{T}_g = 3n^2 \mathbf{a}_3 \times \mathbf{I}\mathbf{a}_3 \quad (3-56)$$

Where \mathbf{a}_3 is the 3rd column of the Direction Cosine Matrix of the B frame w.r.t. the O frame given in equation 3-57, n is the orbit angular speed given in equation 3-58 where μ is the gravitational parameter of the Earth and a is the semi-major axis, and \mathbf{I} is the moment of inertia of the satellite given in table 2-1.

$$\mathbf{a}_3 = \begin{bmatrix} T_{OB(1,3)} \\ T_{OB(2,3)} \\ T_{OB(3,3)} \end{bmatrix} \quad (3-57)$$

$$n = \sqrt{\frac{\mu}{a^3}} \quad (3-58)$$

Thruster-axis Misalignment Disturbance Torque

The thruster may be misaligned with respect to the center of gravity. This will cause an additional disturbance torque which needs to be taken into account. The misalignment in the y and z direction will only be considered, an misalignment in the x direction is considered to be negligible compared to the total length of the satellite. In addition, the thruster will also have an angular misalignment. These misalignments are all visualised in figure 3-13. For the simulations the following values of the misalignments will be used [18]:

- $\alpha = 2$ degree
- $d_y = 1$ cm
- $d_z = 1$ cm
- $F_{thruster} = 1.4$ mN

The angular misalignment will direct the thrust in a direction which causes the highest disturbance torque for an investigation of a worst case. Hence, the thrust is directed such that the project of the thrust in the yz plane will be perpendicular to the line connecting the center of gravity and the position of the thrust outlet located on the yz plane.

These misalignments will induce a disturbance torque in all three body-axis about the center of gravity, namely:

$$\begin{aligned} \mathbf{M}_{\text{thruster}} &= \begin{bmatrix} M_{\text{thruster},x} \\ M_{\text{thruster},y} \\ M_{\text{thruster},z} \end{bmatrix} = \begin{bmatrix} -d_z \frac{\sin(\alpha)}{2} F_{\text{thruster}} - d_y \frac{\sin(\alpha)}{2} F_{\text{thruster}} \\ -d_z \cos(\alpha) F_{\text{thruster}} - \frac{L_x}{2} \frac{\sin(\alpha)}{2} F_{\text{thruster}} \\ d_y \cos(\alpha) F_{\text{thruster}} - \frac{L_x}{2} \frac{\sin(\alpha)}{2} F_{\text{thruster}} \end{bmatrix} \\ &= \begin{bmatrix} -0.01 \frac{\sin(2^\circ)}{2} 0.0014 - 0.01 \frac{\sin(2^\circ)}{2} 0.0014 \\ -0.01 \cos(2^\circ) 0.0014 - \frac{0.34}{2} \frac{\sin(2^\circ)}{2} 0.0014 \\ 0.01 \cos(2^\circ) 0.0014 - \frac{0.34}{2} \frac{\sin(2^\circ)}{2} 0.0014 \end{bmatrix} \end{aligned} \quad (3-59)$$

$$\mathbf{M}_{\text{thruster}} = \begin{bmatrix} -0.13959 \\ -2.3837 \\ -2.3623 \end{bmatrix} \cdot 10^{-4} [\text{Nm}] \quad (3-60)$$

Where L_x is the length of the satellite defined in figure 3-13 and given in table 2-1.

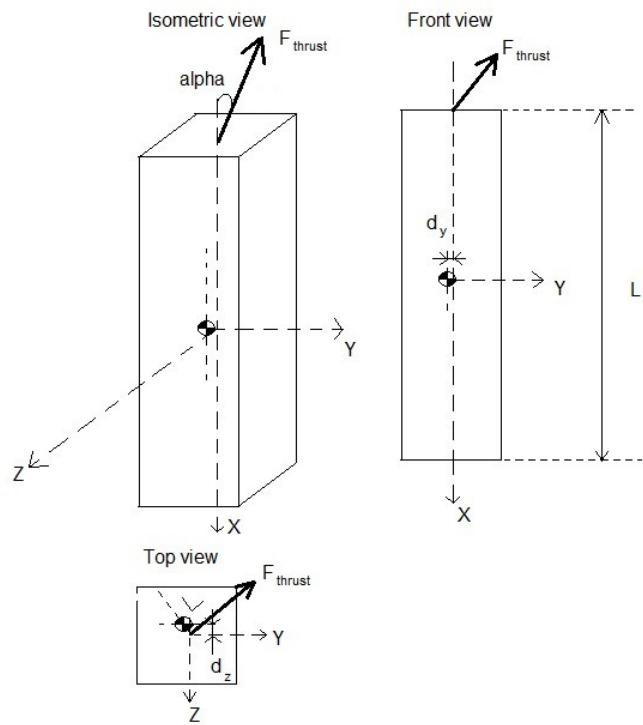


Figure 3-13: Thruster misalignment overview

3-7 Sensor and Actuator Models

This section will discuss the mathematical algorithms that have been used to simulate the behaviour of the sensors due to noise and bias and to simulate the behaviour of the actuators based on a certain input to the driving electronics of the actuators.

3-7-1 Magnetometer Modelling

The magnetometers will be modelled by adding a simple noise and bias. The input for this will be the geomagnetic field vector, \mathbf{B}_B , as determined by the simple dipole model which is explained in subsection 3-6-1. The magnetometer measurements are then modelled as:

$$\mathbf{B}_m = \mathbf{B}_B + \mathbf{B}_n + \mathbf{B}_b \quad (3-61)$$

Where \mathbf{B}_b is the bias the magnetometer and is estimated to be 500 nT in all three axis. The noise, \mathbf{B}_n , is a Gaussian white noise with a standard deviation of 170 nT. These values are taken from chapter 2. The noise value is independent for each axis.

3-7-2 Sun Sensor Modelling

The measurement of the Sun vector will be done by rotating the Sun vector as obtained from the model of the Sun's position as explained in subsection 3-6-2. The rotation will be about the x, y and z axis of the body fixed reference frame with 3 different angles which are independent from each other. These angles are determined using the bias, S_b , which is determined to be 8 degrees and the noise, S_n , of the Sun sensors which has a standard deviation of 0.4 degrees. These values are taken from chapter 2. The angle used for the rotation is given as follows:

$$\text{angle}_i = S_b + S_n \quad (3-62)$$

the bias and the noise used here will be divided by three for each angle, because there will be three angles which are used for the rotation.

Using this angle and the rotations are x, y and z axis, the measured Sun vector can be calculated as follows:

$$\mathbf{S}_m = T_z(\text{angle}_3) * T_y(\text{angle}_2) * T_x(\text{angle}_1) * \mathbf{S}_B \quad (3-63)$$

Where \mathbf{S}_B is the Sun vector as obtained from the model of the Sun's position.

3-7-3 Magnetic Torquer Modelling

Magnetic torquers are very attractive for usage on cubesats due to the fact that they only require electrical energy, unlike thrusters, and do not have moving or rotating parts, unlike reaction wheels. The torque that magnetic torquers generate is simply a cross product between the magnetic dipole moment in the x, y and z direction and the geomagnetic field vector in the x, y and z direction. The cross product results in a torque which is only perpendicular

to the geomagnetic field vector in that specific position in the orbit. The magnetic torquer torque is give by:

$$\mathbf{T}_{mtq} = \mathbf{m}_{mtq} \times \mathbf{B}_m \quad (3-64)$$

Where \mathbf{m}_{mtq} is the magnetic dipole moment in vector form produced by the magnetic torquers and \mathbf{B}_m is the measured geomagnetic field vector.

The magnetic torquers have a maximum magnetic dipole moment that they can produce which is equal to $0.4 \text{ A}\cdot\text{m}^2$. The magnetic torquers designed for the DelFFi mission have three states: on in the negative direction, on in the positive direction and off, therefore an intermediate value of the magnetic dipole momemt can not be produced. In order to still produce a certain torque which requires a certain magnetic dipole moment, the torquers will be run for a certain amount of time. In this way the magnetic torquers can still produce an effective magnetic dipole moment which is equal to the desired magnetic dipole moment. The time the magnetic torquers need to be on to produce a certain effective magnetic dipole moment can be calculated with the following equation:

$$t_{mtq} = \min\left(\frac{\mathbf{m}_{mtq}}{m_{max}}, 1\right) \cdot (t_{actuator}) \quad (3-65)$$

Where \mathbf{m}_{mtq} is the magnetic dipole moment as commanded by the control algorithm, m_{max} is the maximum dipole moment that the magnetic torquers can produce and $t_{actuator}$ is the time that the magnetic torquer can operate during one loop. It is important that the magnetic torquers are not running during the time the magnetometers are running to avoid biased measurements of the magnetometers. The magnetic hysteresis of the magnetic torquers can influence the measurements of the magnetometers time.

Magnetic Hysteresis

The magnetic torquer rods will be magnetized when a current flows through the coil. However, due to magnetic hysteresis the rods will remain magnetized when the current through the coil stops running. A visualisation of this phenomena is given in figure 3-14, where H is the magnetization force and B is the magnetic flux density. As can be seen in the figure, when the magnetization force is zero, that is when there is no current running through the coils, the magnetic flux density is not equal to zero, indicating that the magnetic torquer rods are still magnetized. Another type of the hysteresis curve is given in figure 3-15. This hysteresis curve shows that the magnetic flux density is almost zero when there is no current running through the coils. Both cases, with round loop and flat loop, will be discussed in the analysis of the algorithms.

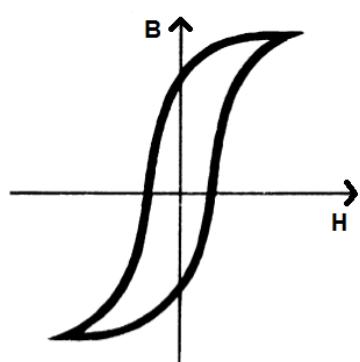


Figure 3-14: Hysteresis curve of a round loop [6]

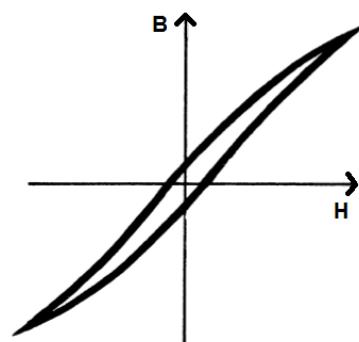


Figure 3-15: Hysteresis curve of a flat loop [6]

Chapter 4

Design of the Algorithms

This chapter will discuss the algorithms which are used to determine the tumble parameter, the attitude and the rotation rate of the satellite.

4-1 Tumble Parameter Determination Algorithm

The tumble parameter will be used to determine whether the satellite has been detumbled. If the satellite is detumbled it will proceed to the velocity-pointing mode which is the nominal mode of the satellite's ADCS. The satellite will be detumbled if the tumble parameter decreases to a threshold, the detumble limit $L_{detumbled}$ for a certain amount of time, $t_{detumbled}$. This is because the tumble parameter often reaches the detumble limit threshold without actually being detumbled due to a time delay in the system caused by the low pass filter. The low pass filter will reduce the noise in the Bdot measurement which is induced by the noise of the magnetometer. The detumble limit $L_{detumbled}$ and $t_{detumbled}$ are determined in section 5-1. The tumble parameter is defined as follows:

$$P_{i+1} = \alpha \cdot |\dot{\mathbf{B}}_{measured}| + (1 - \alpha) P_i \quad (4-1)$$

Where P is the tumble parameter, α is the low pass filter constant, $\dot{\mathbf{B}}_{measured}$ is the time derivative of the measured geomagnetic field vector.

Using this method to determine the tumble parameter and consequently determining whether the satellite has detumbled, is more effective than determining the rotation rate of the satellite using a attitude determination algorithm. The effectiveness of this method is based on the fact that it uses a simple and reliable algorithm to determine whether the satellite is spinning too fast. However, it should be noted that this method does not determine the absolute rotation rate. Instead, the variables $L_{detumbled}$ and $t_{detumbled}$ will be used to determine whether satellite has detumbled. Therefore, these two parameters need to be tuned.

4-2 Attitude Estimation Algorithm

An important part of the ADCS of a spacecraft is the determination of the attitude and the rotational rate. The attitude determination algorithm of the DelFFi mission will use the measurement of the Sun sensors and the magnetometer. These sensors have been chosen by the project management team of the DelFFi mission [7], because of the heritage from the Delfi-n3Xt mission. Using these two sensors, two vectors in the body fixed reference frame can be obtained. In addition there are mathematical models available as discussed in section 3-6-1 and 3-6-2, which will provide two vectors in the Earth-Centered Inertial reference frame. The fundamental goal of an attitude determination algorithm is finding the rotation matrix that describes the rotation from the inertial reference frame to the body fixed reference frame using the measurements and the mathematical model from the sensors, as given in the following equation:

$$\mathbf{v}_B = \mathbf{T}_{BI} \mathbf{v}_I \quad (4-2)$$

Where the subscript B indicates by the body fixed reference frame, I indicates the Earth-Centered Inertial reference frame, v is a vector and T is the rotation matrix.

Due to the unit vector constraint, the unit vector obtained from a measurement or mathematical model only provides two parameters to determine the attitude, while the attitude determination problem requires three independent parameters to determine the attitude. Therefore one vector is not enough (underdetermined) and two vectors is too many (overdetermined). Solving this problem would require (a minimum of) two vectors and to discard one (or more) parameter(s). To avoid discarding parameters, a deterministic attitude determination technique called the TRIAD technique [19] can be applied. This technique will assume that one vector pair of the body and inertial reference frame is correct and the other vector pair is less accurate. In this way all parameters will still be used while still adhering to the attitude determination requirement. Combining this technique with a low-pass filter to filter out the noise of the measurements would result in a suitable attitude determination technique. However, this approach has the disadvantage that it is assumed that one sensor is more accurate than the other. This is a direct implication of the attitude determination problem, therefore an attitude estimation technique will be used instead

Attitude estimation requires the estimation of the spacecraft's orientation and filtering noisy measurements as described in the previous paragraph. There are many different types of filters. In this section the most commonly used filters in the spacecraft attitude estimation field will be discussed. These filters are:

1. Kalman Filter (KF) [20]: A Kalman Filter is an unbiased, minimum-variance linear estimator. The algorithm consists of a recursive method which means that the previous estimated state vector is used to calculate a new estimated state vector. This filter is ideal for tracking the position of an object which is a linear dynamic system.
2. Extended Kalman Filter (EKF) [21]: The motion of a satellite is not a linear system. It is a non-linear dynamical system. The Extended Kalman Filter will linearize the system around the estimated state using the Jacobian matrix, given that the estimation error of the state vector is small. This will extend the Kalman Filter for usage in non-linear dynamic systems.

3. Unscented Kalman Filter (UKF) [22]: A non-linear Kalman filter which is an improvement over the EKF is the Unscented Kalman Filter (UKF). The idea of the UKF method is to produce several sampling points around the current state estimate based on its covariance. This makes the filter applicable to highly non-linear systems.
4. Quaternion Estimator (QUEST) [23]: The QUEST is based on Wahba's problem. Wahba's problem is the problem of finding the rotation matrix \mathbf{R}_{bi} that minimizes the loss function:

$$J(\mathbf{R}_{bi}) = \frac{1}{2} \sum_{k=1}^N w_k \|\mathbf{v}_{kb} - \mathbf{R}_{bi}\mathbf{v}_{ki}\|^2 \quad (4-3)$$

Where J is the loss function, \mathbf{R}_{bi} is the rotation matrix from the inertial reference frame to the body fixed reference frame, w_k are non-negative weights, \mathbf{v} are the measurement and model vectors. The smaller J is, the better the approximation of the rotation matrix.

5. Particle Filter (PF) [24]: Particle filters are especially useful when the process noise and the measurement noise are not normally distributed processes. This type of filter belongs to the class of genetic algorithms, which is a search heuristic that mimics the process of natural selection. Genetic algorithms have a high computational load [25].

Attitude Estimation Technique Selection

An overview of these filters with their main characteristics is given in table 4-1. The filter that will be used in the ADC algorithm must display the following properties:

- Non-linear behaviour of the dynamics and measurement model.
- Low computational load.

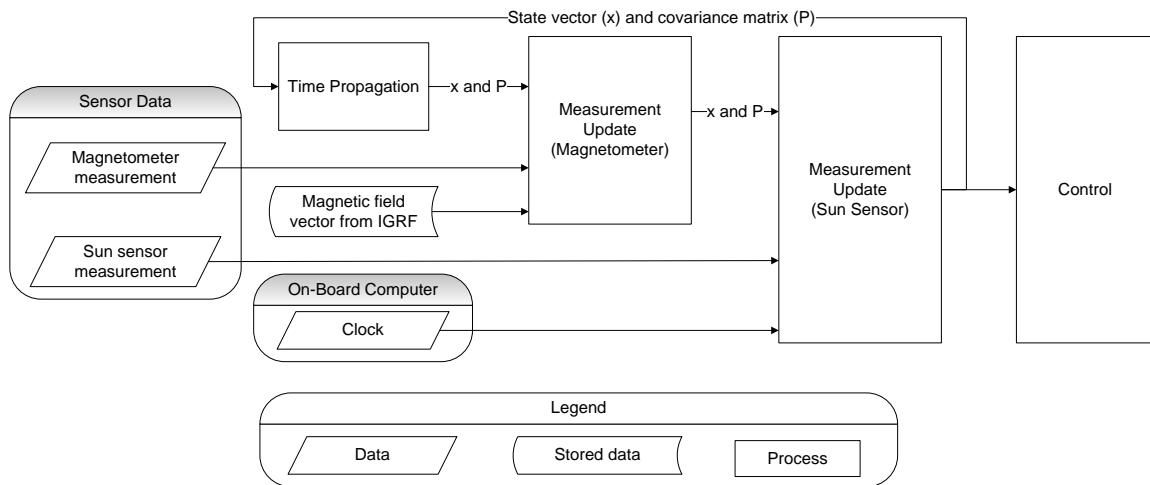
The motion of a satellite is a non-linear system therefore the Kalman Filter and the QUEST are inapplicable, because these two filters require a linear dynamics and measurement model. The filter will be designed for a cubesat, therefore it is important for it to have a low computational load. This results in the Particle Filter being inapplicable. Hence, the filters that are suitable for estimating the attitude are the EKF and the UKF. A comparison by LaViola (2003) [26] between these two filters for the estimation of a motion using quaternions showed that the UKF has similar estimation accuracy, while it has a slightly higher computational load than the EKF. This higher computational load is caused by the fact that the UKF performs $(2 \cdot L + 1)$ integrations while the EKF performs only 1 integration per loop, where L is the length of the state vector. Based on the fact that the EKF has a slightly lower computational load, without a reduction of performance compared to the UKF, the EKF has been selected as the attitude estimation technique.

Extended Kalman Filter Algorithm

This section will describe the Extended Kalman Filter which has been designed for the velocity-pointing mode. A general overview of the EKF algorithm is given in figure 4-1.

Table 4-1: List of most commonly used filters in the field of spacecraft attitude estimation.

Filter	Linearity of the dynamics and measurement models			Computational load	
	Linear	Non-linear	Highly non-linear	Low	High
KF	X			X	
EKF		X		X	
UKF			X	X	
QUEST	X			X	
PF		X			X

**Figure 4-1:** Overview of the Extended Kalman Filter

The filter will consist of 3 phases, namely the time propagation step and the two measurement update steps for both sensors. The state vector which has been defined for this filter is given as a combination of the quaternion and the rotational rate:

$$\mathbf{x} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T \quad (4-4)$$

Where q_i is the quaternion which describes the rotation from the I frame to the B frame and ω_i is the rotation rate of the B frame with respect to the I frame.

Time Propagation Step

In this step a mathematical model [21] will be used to propagate the state from the previous time to the current time in order to obtain the a priori state estimate $\hat{\mathbf{x}}_{k+1/k}$ and the a priori estimation error covariance matrix, $P_{k+1/k}$. The mathematical model that uses the kinematic

equation (equation 3-32) for propagating the quaternion of the state vector is given as:

$$\begin{aligned}\boldsymbol{\omega} &= \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \\ \Omega_k &= \begin{bmatrix} [\boldsymbol{\omega} \times] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix} \\ \Phi_k &= e^{\frac{1}{2}\Omega_k \Delta t} \\ \hat{\mathbf{q}}_{k+1/k} &= \Phi_k \hat{\mathbf{q}}_{k/k}\end{aligned}\tag{4-5}$$

Where the subscript k indicates the timestep, Δt is the time it takes to complete one complete loop.

The mathematical model that uses Euler's dynamic equation for propagating the rotational rate of the state vector is given as:

$$\begin{aligned}\dot{\boldsymbol{\omega}}_{k/k} &= \mathbf{J}^{-1} (\mathbf{T}_c - \boldsymbol{\omega}_{k/k} \times \mathbf{J} \boldsymbol{\omega}_{k/k}) \\ \hat{\boldsymbol{\omega}}_{k+1/k} &= \boldsymbol{\omega}_{k/k} + \dot{\boldsymbol{\omega}}_{k/k} \Delta t\end{aligned}\tag{4-6}$$

Where \mathbf{T}_c is the control torque defined as $\mathbf{m} \times \mathbf{B}_{measured}$, \mathbf{J} is the inertia matrix of the satellite. Note that in the equation of Euler's Dynamic equation, the disturbance torque is left out to reduce the computational intensity of the algorithm.

Next step is to propagate the estimation error covariance matrix \mathbf{P} . The estimation error covariance matrix describes the uncertainty of the estimation of the state vector. Low values for the estimation error covariance matrix on the diagonal of the matrix indicate small uncertainties. The function that describes the change in the non-linear state vector uses equation 3-32 and 3-37. This function is given as follows:

$$f(\mathbf{x}) = \dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\Omega_k \mathbf{q}_{k/k} \\ \mathbf{J}^{-1} (\mathbf{T} - \boldsymbol{\omega}_{k/k} \times \mathbf{J} \boldsymbol{\omega}_{k/k}) \end{bmatrix}\tag{4-7}$$

Linearizing this non-linear function about its nominal values gives the Jacobian matrix of this function:

$$F = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \dot{q}_1}{\partial q_{k/k,1}} & \dots & \frac{\partial \dot{q}_1}{\partial \omega_{k/k,3}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \dot{q}_3}{\partial q_{k/k,1}} & \dots & \frac{\partial \dot{q}_3}{\partial \omega_{k/k,3}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\Omega_k & \frac{1}{2}\Xi_k \\ \mathbf{0}_{[3x4]} & \Theta_k \end{bmatrix}\tag{4-8}$$

Where Ξ_k and Θ_k are defined as:

$$\begin{aligned}\Xi_k &= \begin{bmatrix} q_{k/k,4} & -q_{k/k,3} & q_{k/k,2} \\ q_{k/k,3} & q_{k/k,4} & -q_{k/k,1} \\ -q_{k/k,2} & q_{k/k,1} & q_{k/k,4} \\ -q_{k/k,1} & -q_{k/k,2} & -q_{k/k,3} \end{bmatrix} \\ \Theta_k &= \begin{bmatrix} 0 & \omega_{k/k,3} \frac{J_{yy}-J_{zz}}{J_{xx}} & \omega_{k/k,2} \frac{J_{yy}-J_{zz}}{J_{xx}} \\ \omega_{k/k,3} \frac{J_{zz}-J_{xx}}{J_{yy}} & 0 & \omega_{k/k,1} \frac{J_{zz}-J_{xx}}{J_{yy}} \\ \omega_{k/k,2} \frac{J_{xx}-J_{yy}}{J_{zz}} & \omega_{k/k,1} \frac{J_{xx}-J_{yy}}{J_{zz}} & 0 \end{bmatrix}\end{aligned}\tag{4-9}$$

Using the Jacobian matrix, the state transition matrix Ψ can be determined. The estimation error covariance matrix can then be determined from this state transition matrix as follows:

$$\begin{aligned}\Psi_k &= \expm(F \cdot \Delta t) \\ \mathbf{Q} &= \begin{bmatrix} \sigma_q^2 \mathbf{I}_4 & \mathbf{0}_{[4 \times 3]} \\ \mathbf{0}_{[3 \times 4]} & \sigma_w^2 \Delta t \mathbf{I}_3 \end{bmatrix} \\ \mathbf{P}_{k+1/k} &= \Psi_k \mathbf{P}_{k/k} \Psi_k^T + \mathbf{Q}\end{aligned}\quad (4-10)$$

Where \mathbf{Q} is the process noise matrix, which describes the standard deviation of the process noise for propagating the quaternion (σ_q) and the rotational rate (σ_ω).

Measurement Update Step

The measurement update step uses the magnetometer measurement and, if available, the Sun sensor measurements sequentially using the same measurement update algorithm. This measurement update algorithm [21] uses the measurement vector in the body fixed reference frame as obtained from the sensors, \mathbf{b} , and the reference vector in the inertial reference frame as obtained from the IGRF look-up table or the Sun model, \mathbf{r} . The algorithm for the measurement update step is given as follows:

$$\begin{aligned}\mathbf{s} &= \frac{1}{2}(\mathbf{b} + \mathbf{r}) \\ \mathbf{d} &= \frac{1}{2}(\mathbf{b} - \mathbf{r}) \\ \mathbf{H} &= \begin{bmatrix} -[\mathbf{s} \times] & \mathbf{d} \\ -\mathbf{d}^T & 0 \end{bmatrix} \\ \bar{\mathbf{H}} &= \begin{bmatrix} \mathbf{H} & \mathbf{0}_{[4 \times 3]} \end{bmatrix} \\ \mathbf{R} &= \sigma_{sensor}^2 \mathbf{I}_4 \\ \mathbf{K} &= \mathbf{P}_{k+1/k} \bar{\mathbf{H}}^T (\bar{\mathbf{H}} \mathbf{P}_{k+1/k} \bar{\mathbf{H}}^T + \mathbf{R})^{-1} \\ \hat{\mathbf{x}}_{k+1/k+1} &= (\mathbf{I}_7 - \mathbf{K} \bar{\mathbf{H}}) \hat{\mathbf{x}}_{k+1/k} \\ \mathbf{P}_{k+1/k+1} &= (\mathbf{I}_7 - \mathbf{K} \bar{\mathbf{H}}) \mathbf{P}_{k+1/k} (\mathbf{I}_7 - \mathbf{K} \bar{\mathbf{H}})^T + \mathbf{K} \mathbf{R} \mathbf{K}^T\end{aligned}\quad (4-11)$$

Where \mathbf{H} is the observation matrix, \mathbf{R} is the sensor noise matrix, \mathbf{K} is the optimal Kalman Gain, $\hat{\mathbf{x}}_{k+1/k+1}$ is the a posteriori state estimate and $\mathbf{P}_{k+1/k+1}$ is the a posteriori estimation error covariance matrix

This algorithm will run once with the magnetometer data and once with the Sun sensor data, if the satellite is not in eclipse. Otherwise the measurement update step will run only once with the magnetometer data.

Remarks

- The noise covariances \mathbf{Q} and \mathbf{R} need to be tuned correctly.
- It is important to note that the Kalman Gain is only optimal, if the noise is white and the noise covariance estimations are accurate. An optimal Kalman Gain will result in the best blend between predicted and measured state.

4-3 Control Algorithm

This section will discuss the equations that will be used for attitude control algorithm of the satellite. The actuator that will be used for actuating the control torques is a set of three magnetic torquers. The interaction between these three current-driven magnetic torquers and the geomagnetic field generates a torque. The torques, however, lie in the plane orthogonal to the geomagnetic field vector. Therefore three-axis attitude control is only possible if the satellite passes a significant part of the orbit where the satellite experiences variations in the geomagnetic field vector. Therefore the satellite is only controllable if the orbit is inclined, to ensure variation of the magnetic field.

4-3-1 Detumble Mode Control

During the detumble mode the satellite needs to "follow" a specific field vector. This means that the satellite's attitude will remain fixed with respect to these field lines. Two difference options are possible for the field vector, namely the geomagnetic field vector and the Sun field vector. The algorithm that aligns the satellite's body axis with the geomagnetic field vector using magnetometer measurements is called the Bdot control algorithm. Similarly the algorithm that aligns the satellite's body axis with the Sun field vector using the Sun sensor measurements is called the Sdot control algorithm. The Bdot control algorithm has two advantages over the Sdot control algorithm. Firstly, in the case of the Bdot control algorithm, the commanded magnetic dipole moment can be obtained directly from the calculations. This commanded magnetic dipole moment is used as an input for the actuator driver. In the case of the Sdot control algorithm, the torque first needs to be converted to the commanded magnetic dipole moment. Secondly, the magnetometer measurements are always available, whereas the Sun sensor measurements are not available during eclipse. Therefore the Bdot control algorithm has been chosen as the algorithm in the detumble mode.

The next paragraph explains the Bdot control algorithm.

Bdot Control

The working principle of this technique is to keep the change in the geomagnetic field vector low. This leads to the the satellite having a rotational rate of twice the orbital rate due to the configuration of the geomagnetic field lines. This control algorithm will be simple and computational not intensive, because it does not need attitude determination or estimation algorithms and the control law is simple and efficient. The control itself is only done with magnetic torquers. The Bdot control law is defined as follows:

$$\mathbf{m} = -k_b \cdot \dot{\mathbf{B}}_B \quad (4-12)$$

Where \mathbf{m} is the magnetic dipole moment expressed in $\text{A}\cdot\text{m}^2$, k_b is the controller gain expressed in $\text{A}\cdot\text{m}^2\cdot\text{s}\cdot\text{T}$ and $\dot{\mathbf{B}}_B$ is the time derivative of the geomagnetic field vector in the body fixed reference frame (B frame) expressed in Tesla. The equation for the calculation of the Bdot is given in the following equation:

$$\dot{\mathbf{B}}_B = \frac{\mathbf{B}_{B,i} - \mathbf{B}_{B,i-1}}{\Delta t} \quad (4-13)$$

Where the subscript i denotes the current control step and Δt is the time between the magnetometer measurements of two consecutive control steps. This magnetic dipole moment will induce a torque on the satellite equal to:

$$\mathbf{T} = \mathbf{m} \times \mathbf{B}_B \quad (4-14)$$

Where \mathbf{B}_B is the geomagnetic field vector expressed in the body fixed reference frame in Tesla.

The parameter k_b is a design parameter that needs to be tuned.

4-3-2 Velocity-pointing Mode Control

For the velocity-pointing mode a desired attitude of the satellite will be determined using the velocity vector obtained from the two-line elements (TLEs). This has been explained in section 3-2. The true attitude of the satellite will need to match the desired attitude with a maximum offset of 10 degree, as stated in the ADCS requirements table (table 2-6).

Two control techniques that are widely implemented on the ADCS of spacecraft are the Linear Quadratic Regulator (LQR) and the Proportional-Derivative (PD) controller, as shown in a study by Silani et al. (2005) [27]. These two control techniques will be discussed in the next sections.

LQR

For a general system described in equation 4-15, the LQR control law ([23]) is given in equation 4-16.

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \quad (4-15)$$

$$\mathbf{u}(t) = -\mathbf{k} \cdot \mathbf{x}_e(t) \quad (4-16)$$

Where \mathbf{u} is the control torque, \mathbf{x} is the state vector given in equation 4-4, \mathbf{k} is the optimal gain and the subscript e is the error between the desired and true state vector given by the following two equations:

$$\begin{bmatrix} q_{1e} \\ q_{2e} \\ q_{3e} \\ q_{4e} \end{bmatrix} = \begin{bmatrix} q_{4d} & q_{3d} & -q_{2d} & -q_{1d} \\ -q_{3d} & q_{4d} & q_{1d} & -q_{2d} \\ q_{2d} & -q_{1c} & q_{4d} & -q_{3d} \\ q_{1d} & q_{2d} & q_{3d} & q_{4d} \end{bmatrix} \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \\ \hat{q}_4 \end{bmatrix} \quad (4-17)$$

$$\boldsymbol{\omega}_e = \hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_d \quad (4-18)$$

Where $(\hat{})$ is the variable obtained from the attitude estimation algorithm and the subscript d indicates the desired variable.

The variable k is chosen to minimize the cost function:

$$L = \frac{1}{2} \int_0^{\infty} [\mathbf{x}_e^T \mathbf{Q}_c \mathbf{x}_e + \mathbf{u}^T \mathbf{R}_c \mathbf{u}] dt \quad (4-19)$$

Where Q_c and R_c are covariance matrices. This gives an optimal gain k equal to:

$$\mathbf{k}(t) = \mathbf{R}^{-1}\mathbf{B}(t)^T\mathbf{P}(t) \quad (4-20)$$

And $P(t)$ can be found by solving the Riccati equation given as:

$$\mathbf{PA} + \mathbf{A}^T\mathbf{P} - \mathbf{PBR}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = \mathbf{0} \quad (4-21)$$

The control torque needs to be converted to a magnetic dipole moment of the magnetic torquer. As discussed in Camillo et al. (1980) [28], an accurate method to perform this conversion is by using the following equation:

$$\mathbf{m} = \frac{\mathbf{u} \times \mathbf{B}}{\|\mathbf{B}\|^2} \quad (4-22)$$

PD Control

The PD control law is also based on the system given in equation 4-15. The control law is given as follows ([29]):

$$\mathbf{u}(t) = K_d \mathbf{J} \boldsymbol{\omega}_e + K_p \mathbf{J} \mathbf{q}_e \quad (4-23)$$

Where \mathbf{u} is the control torque, K_d and K_p are scalar control gains, \mathbf{J} is the inertia matrix, $\boldsymbol{\omega}_e$ is the error between the desired and the estimated rotation rate vectors (given in equation 4-18) and \mathbf{q}_e is the first three components of the quaternion (the vector part) that describes the error between the desired and the estimated attitude quaternion (given in equation 4-17). The gains will be determined using tuning techniques. Similarly to the LQR method, the magnetic dipole moment will be calculated from the control torque using equation 4-22.

Attitude Control Technique Selection

The drawback of the LQR method is that solving the Riccati equation is a computationally intensive process that requires numerical integration. However, it is a robust control method, because it gives an optimal gain for varying moments of inertia ([27]). On the other hand, the PD control method is not computationally intensive, because it uses a constant control gain. This results, however, in a non optimal gain where robustness needs to be investigated. Since the PD control uses less computational power and it has been used successfully on a satellite, it has been selected as the attitude control method for the DelFFi satellite. An example of a satellite on which a PD controller has been successfully implemented is the Chinese ZDPS-1A pico-satellite [30]. Moreover, simulations results of the ADCS of the Taiwanese ROCSAT-3 micro-satellite [31] and the Delfi-n3Xt nano-satellite [32] show that this controller can be used to point the satellite in a specific direction. Still, the robustness of the PD controller is a risk and therefore this will be further investigated in chapter 5. As mentioned before, the gains in the PD control need to be tuned. The next section will explain this tuning process.

PD Control Gain Selection

There are various methods to select the gain of the PD control law given in the previous paragraph. This section will explain two gain tuning techniques.

Ziegler-Nicols Tuning Method The Ziegler-Nicols Tuning method [33] is a heuristic tuning method. It is performed by setting the derivative gain to zero. The proportional gain is then increased until it reaches a state where the control torque is oscillating with a constant amplitude. The gain at which this state has been reached is called the ultimate gain, K_u . If K_u is known then proportional gain K_p and derivative gain K_d can be determined as follows:

$$\begin{aligned} K_p &= -0.8 \cdot K_u \\ K_d &= -\frac{K_p T_u}{8} \end{aligned} \quad (4-24)$$

Where T_u is the oscillation period of the control gain for the state where there is a purely oscillating control torque.

Quaternion Feedback Regulator Tuning Method The control gains, namely the proportional gain k and the derivative gain d , can be calculated for optimal performance of the control law as a function of the relative error $e_{allowed}$ that is allowed after the settling time T_s has been reached according to Wie et al. (1989) [34], given in the following two equations:

$$K_d = -2 \frac{\ln(e_{allowed})}{T_s} \quad (4-25)$$

$$K_p = 4 \left(\frac{\ln(e_{allowed})}{T_s} \right)^2 \quad (4-26)$$

The relative error and the settling time need to be tuned for correct performance of the algorithm.

Remarks

- Increasing the proportional gain will result in a faster convergence, but more overshoot.
- Increasing the derivative gain will result in a decrease in overshoot, but slower convergence
- Increasing the gains will result in a better performance, however the limitations of the torquer has to be taken into account. The torquer may not be able to deliver the required control torque for too high gains.

4-4 Summary

This chapter has presented the algorithms of the ADCS of the DelFFi satellite. These are: tumble parameter determination, attitude estimation and attitude control. The tumble parameter will be determined in order to measure the rotation rate of the satellite with respect to the geomagnetic field. This is an efficient technique to determine the behaviour of the satellite. If this value is lower than a certain threshold $L_{detumble}$ for a specific amount of time $t_{detumble}$, then the satellite has been detumbled and the velocity-pointing mode will be triggered. Otherwise the detumble mode will be triggered. In case the flag for the safe mode has been enabled by the OBC, the satellite will enter the safe mode. During the detumble

mode the Bdot control law will be used to detumble the satellite. The velocity-pointing mode will consist of an EKF for estimating the attitude and rotation rate and a PD controller for controlling the attitude. The EKF will propagate the state vector and use the measurements of the magnetometer and the Sun sensors to determine the best estimate of the attitude and the rotation rate. These values will then be fed into the PD controller to determine the magnetic dipole moment of the magnetic torquer in order to point the satellite in a desired direction. The Ziegler-Nicols and the QFR tuning technique will be used to tune the two gains of the PD controller.

Figure 4-2 gives an overview of the control logic.

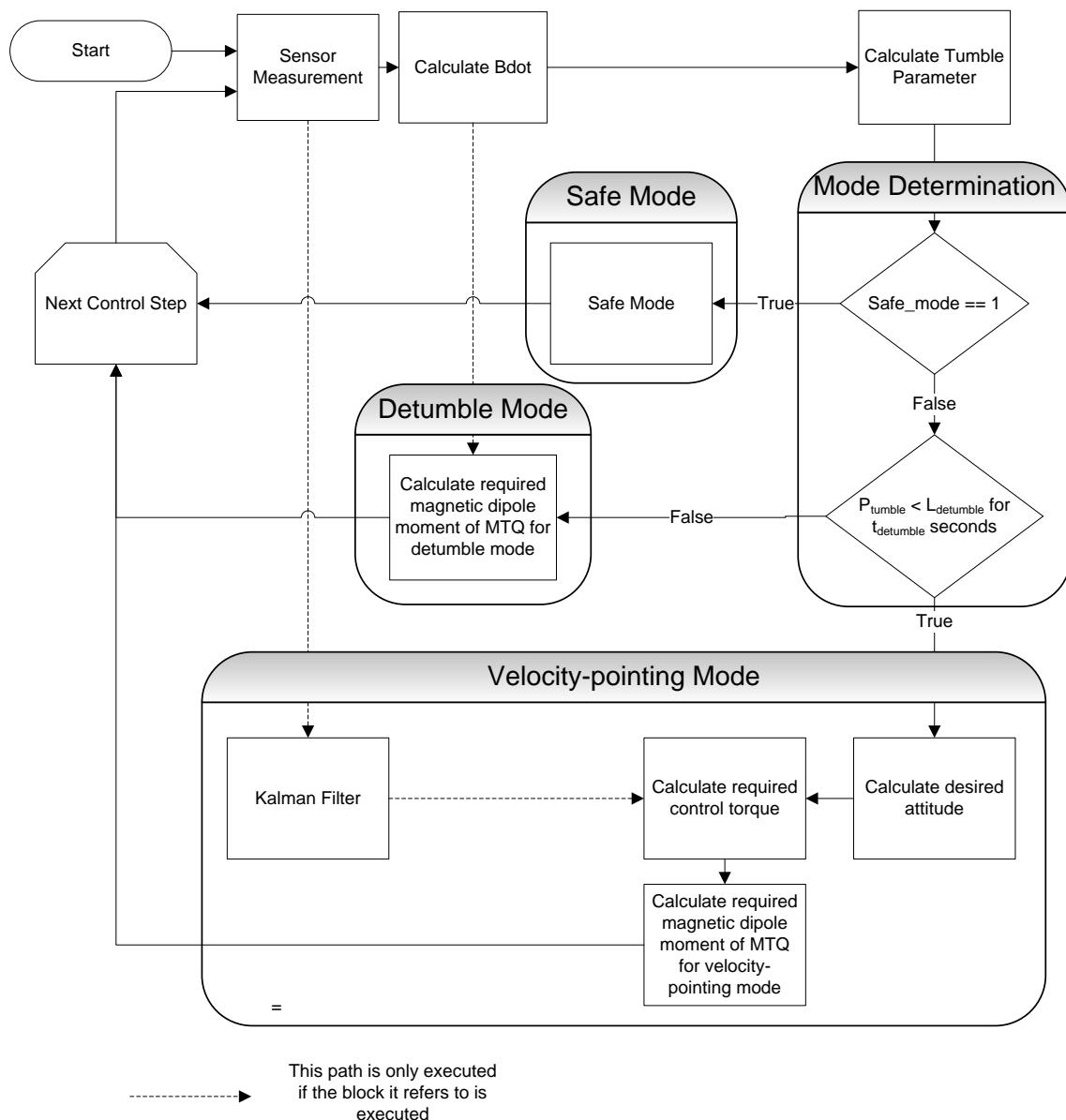


Figure 4-2: Overview of the Attitude Determination and Control Algorithm

Chapter 5

Simulation Results

In this chapter the results of the simulations of the two control modes will be discussed, namely the detumble mode and the velocity-pointing mode (VP mode). For each mode a sensitivity analysis will be performed. Before showing the results, first the modes will be explained.

5-1 Detumble Mode

The control algorithm that will be executed during the detumble mode is the Bdot control algorithm which has been discussed in section 4-3-1. The Bdot controller is designed to lower the time derivative of the geomagnetic field vector to zero or near-zero. This mode will be successfully completed when the tumble parameter, P_{tumble} , is below a certain value called $L_{detumbled}$ for a certain amount of time equal to $t_{detumbled}$ seconds.

5-1-1 Performance Analysis

In this subsection the performance of the Detumble mode will be investigated using varying cases. The investigation of the performance of the Detumble mode will be split up into two parts. The first part will focus on finding the optimal timing sequence interval for a specific initial angular velocity of the satellite's body axis. The second part will focus on investigating non-nominal conditions. These non-nominal conditions are: high noise in the magnetometer measurement, high aerodynamic disturbance torques and failure of a component of the magnetic torquer resulting in loss of torque around one or two axis.

Timing Sequence

The timing sequence of the ADCS has been provided in section 2-5. This section will focus on varying the timing of the four parts of the timing sequence, namely sensor measurement,

algorithm execution, actuating magnetic torquer, and the hold period. Table 5-1 provides a list of the cases that will be used where case A is the timing sequence used on the Delfi-n3Xt satellite [32].

Table 5-1: Overview of the cases for determining the optimal timing sequence interval

Case	Sensor	Algorithm	Actuator	Hold	Total time
A	0.1 s	~ 0 s	1.4 s	0.5 s	2 s
B	0.1 s	~ 0 s	1 s	0.4 s	1.5 s
C	0.1 s	~ 0 s	0.5 s	0.3 s	0.9 s
D	0.1 s	~ 0 s	0.2 s	0.3 s	0.6 s
E	0.1 s	~ 0 s	0.1 s	0.3 s	0.5 s
F	0.1 s	~ 0 s	0.1 s	0.2 s	0.4 s
G	0.1 s	~ 0 s	0.05 s	0.05 s	0.2 s
H	0.05 s	~ 0 s	0.05 s	~ 0 s	0.1 s

The table header consists of the following parameters from left to right:

- Identifier for the used case
- The time it takes to perform one sensor measurement
- The time it takes for the algorithm to be executed
- The time it takes for the actuator to finish actuating
- The time it takes to remove the magnetic hysteresis
- Total time it takes to complete one loop

The satellite is considered to be detumbled if the satellite is rotating with an angular velocity of less than 1 degree per second around all three axes with respect to the inertial reference frame, as defined by the requirements (table A-5). For each case the initial angular velocity will be increased with 5 deg/s per simulation to find the highest initial angular velocity for which the satellite can still be detumbled. For all cases the same magnetometer noise standard deviation and bias will be used. The orbit and satellite parameters used for the simulations are given in table 2-1. The altitude and the solar flux can be found in table 5-2. This results in an air density of $9 \cdot 10^{-11} \text{ kg/m}^3$, as obtained from figure 3-12. The detumble mode parameters can be found in table 5-3. These parameters have been found by first using as an initial estimate of the values from the Bdot control used on the Delfi-n3Xt mission [32] and fine tuning it from those initial estimates. The initial quaternion is $\begin{bmatrix} 0.378 & -0.378 & 0.756 & 0.378 \end{bmatrix}^T$. This is an arbitrary attitude of the satellite that still holds the criteria mentioned in equation 3-21.

Analysis Table 5-4 gives the results of the simulations. The second column gives the initial angular velocity of the body fixed reference frame with respect to the inertial reference frame for which the satellite can still be detumbled. The third column gives the number of orbits that is required in order to detumble the satellite. The ratio actuator time over total time

Table 5-2: Parameters used in the simulations for finding the optimal timing sequence interval

Parameter	Value	Unit
Solar flux	average	
Altitude	380	km

Table 5-3: Parameters used for the detumble mode

Parameter	Value	unit
k_B	28000	A·m ² ·s/T
α_{tumble}	1/1200	-
$P_{tumble,0}$	10^{-6}	T/s

($\frac{t_{actuator}}{t_{total}}$) is a key factor in the convergence rate of the detumbling. This ratio will influence the effective maximum magnetic dipole moment of the magnetic torquer. The theoretical maximum magnetic dipole moment of the magnetic torquer is set to be 0.4 A·m². The effective maximum magnetic dipole moment is calculated in equation 5-1.

$$\mathbf{m}_{max,eff} = \mathbf{m}_{max,theoretical} \cdot \frac{t_{actuator}}{t_{total}} \quad (5-1)$$

From case A until case E this keeps on decreasing which means that the rate of detumbling becomes slower, resulting in a higher number of orbits needed for detumbling. Only in case F it can be seen that this ratio increases resulting in the satellite being detumbled faster. Whether the satellite will be detumbled or remain tumbling depends on the total time it takes to complete one loop, because this determines how fast measurements are taken and how fast the magnetic torquers can change their duty cycles. As can be seen in table 5-1, the satellite can be detumbled for high initial angular velocities of up to 190 deg/s if the magnetometer measurement frequency increases.

The physical explanation for this behaviour of the satellite is the fact that the geomagnetic field vector expressed in the body reference frame changes. By reducing the total loop time, the responsiveness of the ADCS to these changes increases. To show this, the timing sequence

Table 5-4: Results of the simulations of the detumble mode for varying timing sequence intervals.

Case	ω_0	nr. of orbits
A	10 deg/s	1
B	15 deg/s	1.25
C	25 deg/s	1.75
D	35 deg/s	3
E	45 deg/s	6.75
F	55 deg/s	6.25
G	110 deg/s	11.5
H	190 deg/s	31

of case E will be used in the simulation with an initial angular velocity of 10 and 60 deg/s in all three axes. The plots for the geomagnetic field vector expressed in the body reference frame are given in figure 5-1 and 5-2. As has been determined in table 5-1, the satellite will not detumble for an initial angular velocity of 60 degree per second in case E, because the initial angular velocity is too high for the fast changing geomagnetic field vector. However, it can detumble for an initial angular velocity of 10 degree per second, because the geomagnetic field vector changes slowly. The plots for the angular velocity of both cases are given in figure 5-3 and 5-4.

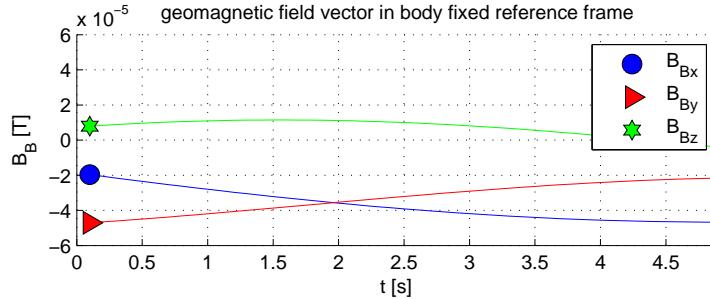


Figure 5-1: Geomagnetic field vector expressed in the B frame for $\omega_0 = 10 \text{ deg/s}$ (Case E)

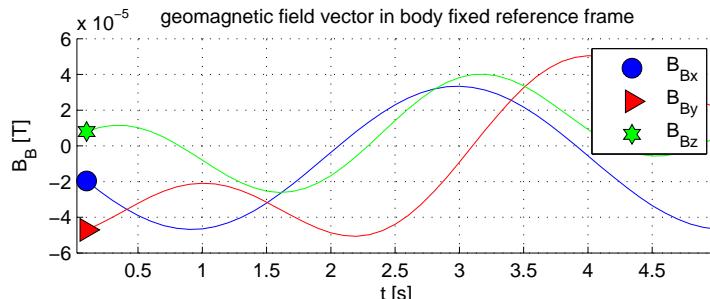


Figure 5-2: Geomagnetic field vector expressed in the B frame for $\omega_0 = 60 \text{ deg/s}$ (Case E)

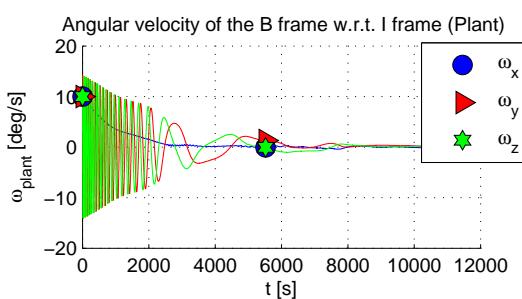


Figure 5-3: Angular velocity of the satellite with respect to I frame for $\omega_0 = 10 \text{ deg/s}$ (timing sequence = case E)

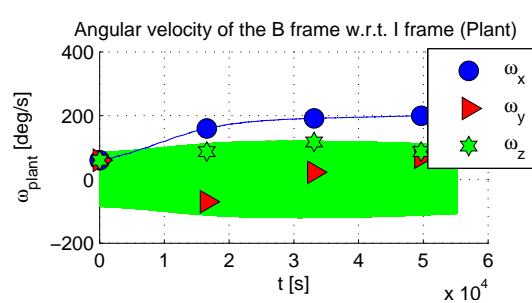


Figure 5-4: Angular velocity of the satellite with respect to I frame for $\omega_0 = 60 \text{ deg/s}$ (timing sequence = case E)

In the QB50 System Requirements issue 6 published in July 2014 [35] the initial rotation rate for which the satellite needed to detumble from was 10 deg/s within 2 days. Case E meets this requirement and in addition also takes into account a margin for an even higher initial

rotation rate which was yet unknown. For the next analysis and the coming chapters the timing sequence of case E will be used in the simulations. In the QB50 System Requirements issue 7 published in February 2015 [36] the initial rotation rate for which the satellite needed to detumble from increased to 50 deg/s within 3 days. In addition, an extra requirement has been set to recover from an initial rotation rate of up to 90 deg/s in case of non-nominal conditions during the jettisoning from the pod. As can be seen in table 5-1, it is possible to meet these new requirements. For this it is, however necessary to determine the hold time to make sure that the magnetic torquers do not affect the magnetometers as the hold time would need to be lowered to decrease the total time to complete one attitude determination and control loop.

Non-nominal conditions

For the simulation of the non-nominal conditions, the timing sequence interval of case E will be used, as discussed in the previous paragraph. In addition, the highest initial angular velocity of this case will be used for which the satellite can still detumble, equal to 45 degree per second around all three axis. These non-nominal cases can be found in table 5-5. The different cases are variations of the following parameters: altitude, solar flux, offset of the center of pressure from the center of gravity, noise of the magnetometer measurements and magnetic torquer failure status. These parameters and other parameters used for this simulation are given in table 2-1 and the parameters specific for the detumble mode are given in table 5-3. The cases I to L will be analysed using the mean angular velocity and its standard deviation using a Monte Carlo simulation. Moreover, the detumble limit $L_{detumbled}$ and detumble time $t_{detumbled}$ have been determined for each case. These two parameters have been explained in section 4-1. The cases M1 to M6 will be analyzed using a single simulation run to determine how the detumble mode performs in case of a failure of the magnetic torquer components causing a malfunction in generating a magnetic dipole moment around one or two axis. It should be noted that for the Monte Carlo simulations the same psuedo random variable sets will be used for each case and for the single simulation run the same random variable will be used for each case.

Table 5-5: Overview of the cases for the performance investigation of the Detumble mode. The values used in this table are obtained from the orbit and satellite properties table (table 2-1)

Case	ω_0	Altitude	Solar flux	r_{cp}	Noise σ	MTQ status
I	[45 45 45]	380 km	average	5 mm	170 nT	on
J	[45 45 45]	200 km	maximum	10 mm	170 nT	on
K	[45 45 45]	380 km	average	5 mm	1700 nT	on
L	[45 45 45]	200 km	maximum	10 mm	1700 nT	on
M1	[45 45 45]	380 km	average	5 mm	170 nT	MTQ _x off
M2	[45 45 45]	380 km	average	5 mm	170 nT	MTQ _y off
M3	[45 45 45]	380 km	average	5 mm	170 nT	MTQ _z off
M4	[45 45 45]	380 km	average	5 mm	170 nT	MTQ _y & MTQ _z off
M5	[45 45 45]	380 km	average	5 mm	170 nT	MTQ _x & MTQ _y off
M6	[45 45 45]	380 km	average	5 mm	170 nT	MTQ _x & MTQ _z off

Case I Case I is the standard case as simulated already in case E. The altitude, solar flux, distance between center of pressure and center of gravity and the MTQ failure status are set to their standard values.

Case J In case J the components that influence the aerodynamic disturbance torque have been changed to simulate the worst case disturbance torques. The parameters that are changed are the altitude and the solar flux, which will increase the air density resulting in higher aerodynamic disturbance torques and the distance between the center of pressure and center of gravity r_{cp} by a factor of 2 which increases the arm of the aerodynamic disturbance force.

Case K In case K the noise standard deviation has been increased to 10 times the normal value.

Case L Case L is a combination of case J and K where both the aerodynamic disturbance torques and the noise have been increased.

Case M In case M the magnetic torque performance will be investigated in case of failure of one or more components in the three axis. For this case a Monte Carlo simulation will not be performed.

Analysis The results of the simulations are given in table 5-6 and in figure 5-7 to 5-12. The following conclusions can be drawn from these results:

1. In a lower altitude the velocity of the satellite and the air density are higher, resulting in a higher aerodynamic drag. However, the geomagnetic field strength also increases in a lower altitude. The results show that decreasing the altitude does not have a significant effect on the detumbling performance.
2. Increasing the standard deviation of the noise reduces the detumbling performance. The mean angular velocity and the standard deviation of the angular velocity are further away from zero in case K and L compared to case I and J where the noise standard deviation is normal.
3. The detumble limit $L_{detumbled}$ and $t_{detumbled}$ depend on the noise of the magnetometer. Therefore these two variables need to be tuned while the satellite is in orbit. For a higher noise standard deviation, the detumble limit $L_{detumbled}$ will be higher and $t_{detumbled}$ will increase. It should be noted, however, that at the start of the simulation the tumble parameter is below the detumble limit $L_{detumbled}$. This is because the method to obtain the tumble parameter P_{tumble} requires the tumble parameter to start at zero and increase to the actual value. Because of this, $t_{detumbled}$ needs to be higher than zero.
4. In case of failure of the magnetic torquer, the satellite will not detumble if it can not produce a torque around the x axis or if it can not produce a torque around both the y and z axes. A failure in the torque around the y or z axis is not a problem, because of the

coupling of the rotational rates. This can be explained with equation 5-2. Expanding this equation gives equation 5-3. Since I_{zz} and I_{yy} are equal to each other, this equation can be rewritten to equation 5-4. This equation shows that there is a coupling between the rotation rates around the y and z axes. From equation 5-4 it can be concluded that a failure of the magnetic torquer around the x axis will cause $T_{c,x}$ to be zero resulting in ω_x not converging and a failure of the magnetic torquer around the y and z axes will remove the control variable in the second and third row of equation 5-4 which results in the coupling of the parts to be ineffective.

$$\dot{\omega} = \mathbf{J}^{-1} (\mathbf{T}_c + \mathbf{T}_d + \omega \times \mathbf{J}\omega) \quad (5-2)$$

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} T_{c,x} + T_{d,x} + \frac{J_{zz}-J_{yy}}{J_{xx}}\omega_y\omega_z \\ T_{c,y} + T_{d,y} + \frac{J_{xx}-J_{zz}}{J_{yy}}\omega_x\omega_z \\ T_{c,z} + T_{d,z} + \frac{J_{yy}-J_{xx}}{J_{zz}}\omega_x\omega_y \end{bmatrix} \quad (5-3)$$

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} T_{c,x} + T_{d,x} \\ T_{c,y} + T_{d,y} + \frac{J_{xx}-J_{zz}}{J_{yy}}\omega_x\omega_z \\ T_{c,z} + T_{d,z} + \frac{J_{yy}-J_{xx}}{J_{zz}}\omega_x\omega_y \end{bmatrix} \quad (5-4)$$

Table 5-6: Results of the performance of the detumble mode for non-nominal conditions of the last 15 minutes for total simulation time of 8 orbits using Monte Carlo simulations. All values given in deg/s unless otherwise stated.

Case	$\omega_{mean,x}$	$\omega_{mean,y}$	$\omega_{mean,z}$	$\omega_{\sigma,x}$	$\omega_{\sigma,y}$	$\omega_{\sigma,z}$	$L_{detumbled}$	$t_{detumbled}$
I	-0.0013	-0.0441	0.0473	0.1159	0.0414	0.1066	$8 \cdot 10^{-7}$ T/s	100 s
J	-0.0049	0.0432	-0.0056	0.0358	0.0448	0.0631	$8 \cdot 10^{-7}$ T/s	100 s
K	0.0583	-0.2435	0.4211	0.3158	0.7925	0.5637	$8 \cdot 10^{-6}$ T/s	8000 s
L	0.0119	-0.0036	0.0052	0.2377	0.2355	0.1886	$8 \cdot 10^{-6}$ T/s	8000 s

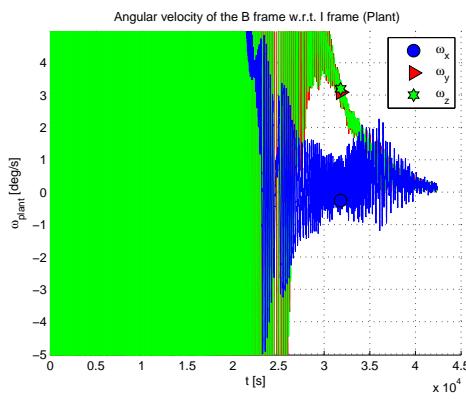


Figure 5-5: Angular velocity of the satellite with respect to I frame (Case L)

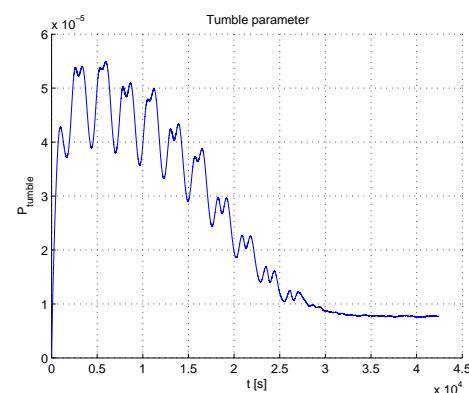


Figure 5-6: Tumble parameter P_{tumble} (Case L)

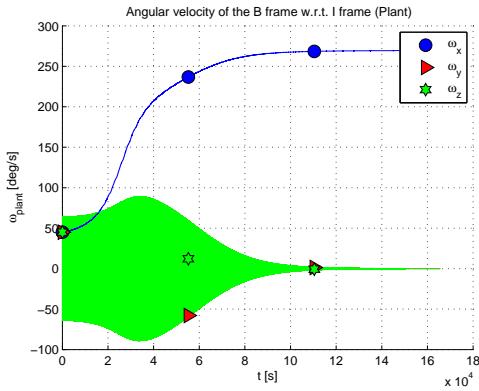


Figure 5-7: Angular velocity of the satellite with respect to I frame (Case M1)

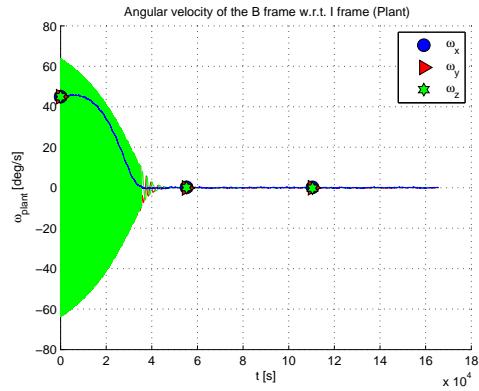


Figure 5-8: Angular velocity of the satellite with respect to I frame (Case M2)

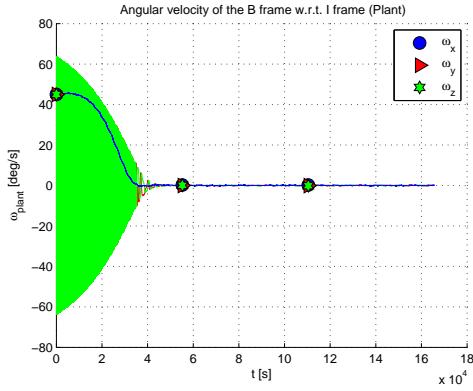


Figure 5-9: Angular velocity of the satellite with respect to I frame (Case M3)

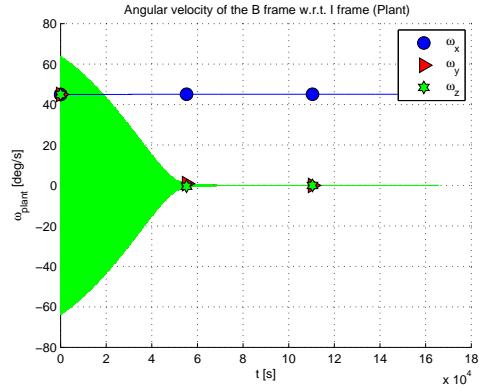


Figure 5-10: Angular velocity of the satellite with respect to I frame (Case M4)

5-2 Velocity-pointing Mode

The velocity-pointing mode has been designed to point in the velocity vector during nominal mode of the satellite operation. The FIPEX instrument on-board of the satellite will need to perform in-situ measurements for which the FIPEX instrument needs to point in the velocity vector. The velocity-pointing mode will be initiated when the detumble mode has achieved successful detumbling as defined in section 5-1. Therefore it can be assumed that when the velocity-pointing mode is initiated, the rotational rate of the satellite is low. The ADC algorithm of the satellite will remain in this mode until the rotational rate of the satellite increases and reaches a value above the threshold defined in the requirements in section 2-4. In the next sections first the attitude estimation and control parameters will be tuned. After that, the velocity-pointing mode will be investigated for four non-nominal and practical conditions. The orbit and satellite parameters that have been used in this section can be found in table 2-1 and 5-2. The timing interval is case E of table 5-1 and the Bdot algorithm parameters are found in table 5-3. The initial rotational rate is equal to 5 degree per second

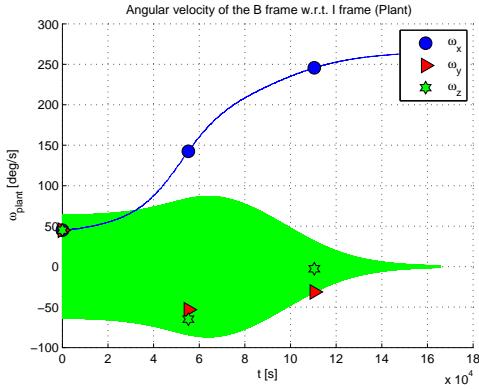


Figure 5-11: Angular velocity of the satellite with respect to I frame (Case M5)

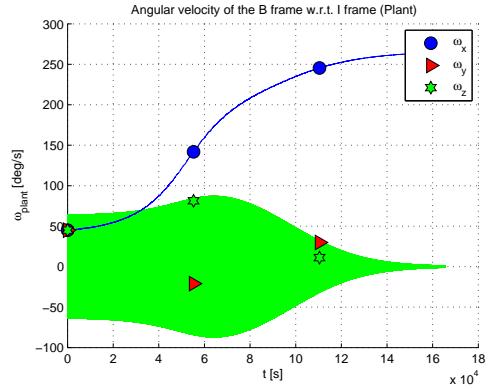


Figure 5-12: Angular velocity of the satellite with respect to I frame (Case M6)

in all three axis and the initial quaternion is $\begin{bmatrix} 0.378 & -0.378 & 0.756 & 0.378 \end{bmatrix}^T$. High values for the initial rotational rate will not be discussed, because it is assumed that the satellite is detumbled during the detumble mode before entering the velocity-pointing mode. The initial quaternion is an arbitrary attitude of the satellite that still holds the criteria mentioned in equation 3-21.

5-2-1 Tuning Parameters

This section will discuss the performance of the attitude estimation algorithm for different process noise and sensor noise. After that the performance of the control algorithm will be discussed using the two control gain tuning methods. In order to decouple the attitude estimation parameter tuning from the control gain parameter tuning, the Bdot control algorithm will be used to control the satellite for the cases A to H, where the attitude estimation algorithm parameters are tuned. The attitude estimation parameters are given in table 5-7. The control gain parameters are given in table 5-8. For tuning the attitude estimation parameters, as an initial estimate the parameters from the Delfi-n3Xt mission have been used [32], which is given in table 5-9. The standard deviation of the process noises σ_q and σ_ω is chosen much higher than the initial estimate. This is because the estimation performance decreases for low values as will be discussed in the next section. For the control gain parameter tuning, two different methods will be used to determine the control gain parameters, namely the Ziegler-Nicols tuning method and the quaternion feedback regulator tuning method. The gains used for the first method are fixed by the ultimate gain K_u and oscillating period T_u , as found in case I. However, the gain used for the second method is determined by setting the relative allowed error $e_{allowed}$ and the settling time T_s . For case J these two parameters have been chosen such that the proportional gain K_p and the derivative gain K_d are similar to the values found in the first method. From this initial estimate, the relative allowed error and the settling time are varied to find the control gain parameters for which the angular pointing error converges to zero as much as possible during the Sunlit phases.

Table 5-7: Attitude estimation parameter tuning cases

Case	σ_{MM} [deg]	σ_{SS} [deg]	σ_q [-]	σ_ω [deg/s]
A	1	1	0.25	0.25
B	20	1	0.25	0.25
C	40	1	0.25	0.25
D	20	0.1	0.25	0.25
E	20	1	0.25	0.25
F	20	10	0.25	0.25
G	20	1	0.025	0.025
H	20	1	2.5	2.5

Table 5-8: Control gain parameters tuning cases

Case	e_{rel}	T_s [s]	K_u	T_u [s]	K_p	K_d
I	-	-	0.003	70	-0.00240	-0.0210
J	0.01	200	-	-	-0.00210	-0.0461
K	0.03	400	-	-	-0.00031	-0.0175
L	0.05	600	-	-	-0.00010	-0.0100
M	0.07	800	-	-	-0.00004	-0.0066
N	0.09	1000	-	-	-0.00002	-0.0048

Analysis cases A to H

The results will be presented as the angular estimation error given in figures 5-13 to 5-20. For these simulations Monte Carlo simulations will be performed. The angular estimation error is defined as the smallest angle between the Euler eigenaxis of the true attitude expressed in the B frame with respect to the I frame and the estimated attitude expressed in the B frame with respect to the I frame. The Euler eigenaxis is explained in subsection 3-3-3 and the algorithm to obtain the angular estimation error is given in appendix B-2. As can be seen in the figures, it is important to choose the standard deviation of the process noise and the sensor noise accurately to increase the performance of the Extended Kalman Filter. The difference in the performance of the filter between cases A to H is negligible during the period where the Sun is detected. This is because, as explained in section 4-2, there are two vectors available, therefore the filter will be accurate. However, the performance decreases in all cases during the eclipse periods, because there is only one vector available. Therefore there is no

Table 5-9: Filter parameters used on the Delfi-n3Xt mission

Parameter	Value
σ_{MM} [deg]	3.82
σ_{SS} [deg]	3
σ_q [-]	0.005
σ_ω [deg/s]	0.00003

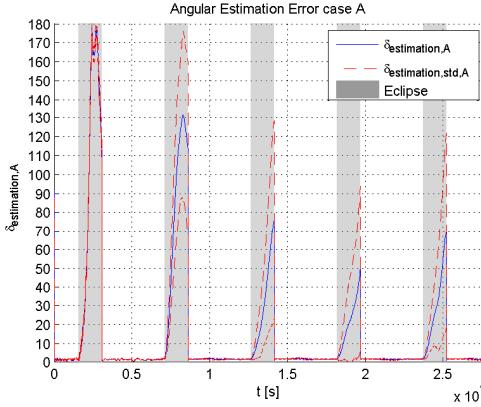


Figure 5-13: Angular Estimation Error in [deg] of case A

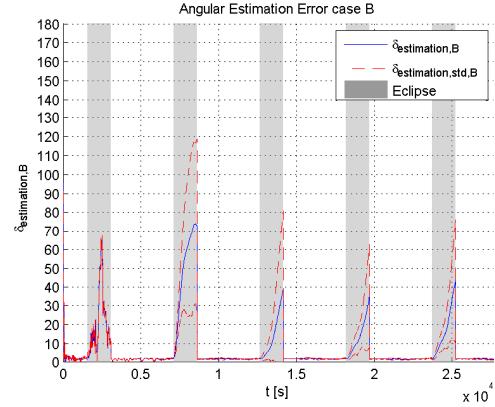


Figure 5-14: Angular Estimation Error in [deg] of case B

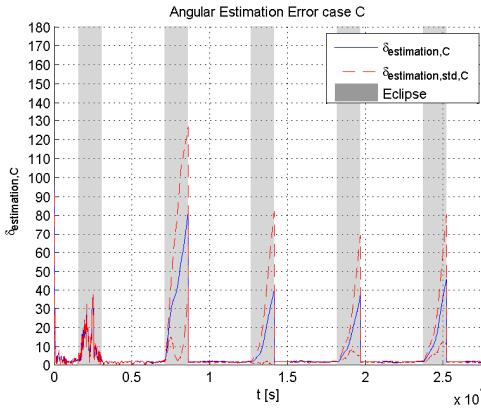


Figure 5-15: Angular Estimation Error in [deg] of case C

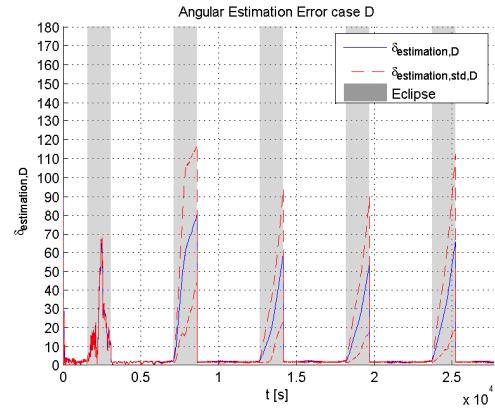


Figure 5-16: Angular Estimation Error in [deg] of case D

large difference between cases D to F, where all parameters are kept constant except for σ_{SS} . Changing σ_{SS} will not have a large influence on the results.

Cases A to C When changing σ_{MM} it can be noticed that there is a certain optimal value for the best angular estimation error. This has been proven to be $\sigma_{MM} = 20$ deg. This is the value that resembles the standard deviation of the noise of the magnetometer best.

Cases D to F As explained before, the standard deviation of the Sun sensor noise will not have a large impact on the results, because the filter is accurate during the phases where the Sun is detected. Even though the changes in the attitude estimation error are small, a σ_{SS} equal to 1 deg gives the lowest angular estimation error during eclipse.

Cases G to H The standard deviation of the quaternion process noise and rotational rate process noise are not changed independently, to simplify the tuning method. However it is expected that when the moment of inertia of the satellite differs from the moment of

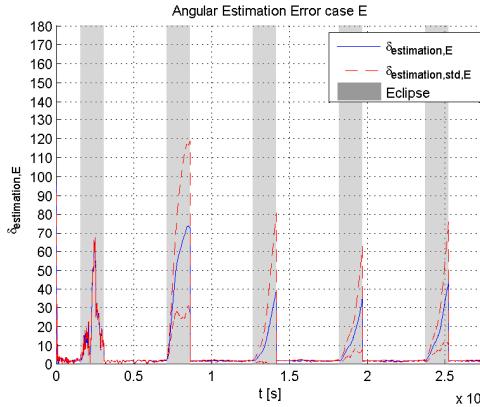


Figure 5-17: Angular Estimation Error in [deg] of case E

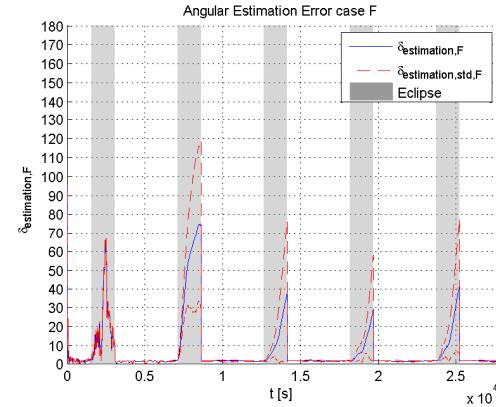


Figure 5-18: Angular Estimation Error in [deg] of case F

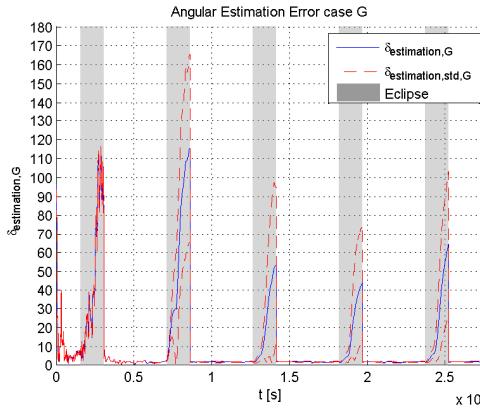


Figure 5-19: Angular Estimation Error in [deg] of case G

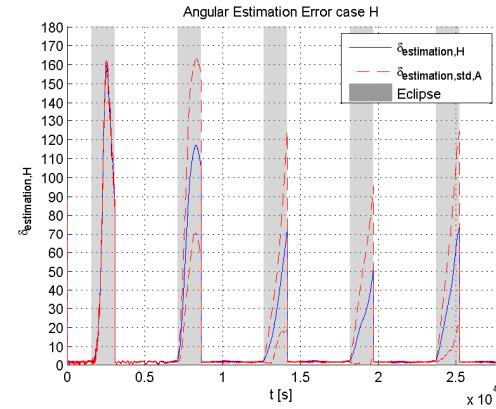


Figure 5-20: Angular Estimation Error in [deg] of case H

inertia inserted in the filter, σ_ω will be higher due to the fact that the rotational rate can be predicted less accurately. Both case G and H have a higher angular estimation error during eclipse compared to case E.

Best case The cases with the lowest angular estimation are D to F. Since there is not a significant difference between these cases, case E has been arbitrarily chosen from these three cases. This case will be used in coming simulations.

Analysis cases I to N

The results of the Monte Carlo simulation are given in figure 5-21 to 5-26. To be able to compare these plots with each other, the mean value of the angular pointing error $\delta_{pointing}$ and the mean value of the standard deviation of the angular pointing error during the last complete orbit and the last eclipse period has been given in a table 5-10. Case L shows the Ziegler-Nicols tuning method and cases J to N show the quaternion feedback regulator

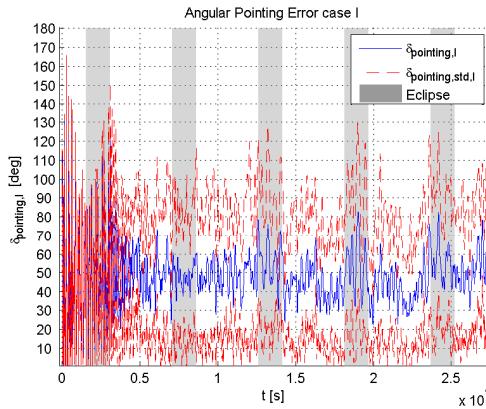


Figure 5-21: Angular Pointing Error of case I

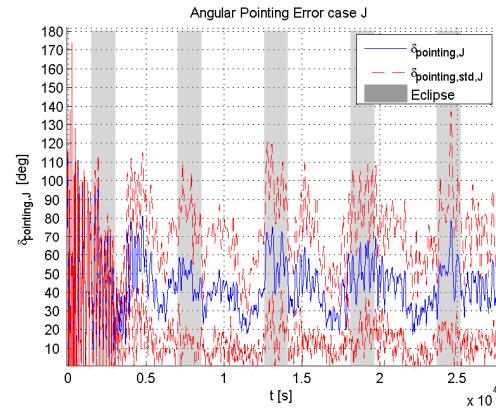


Figure 5-22: Angular Pointing Error of case J

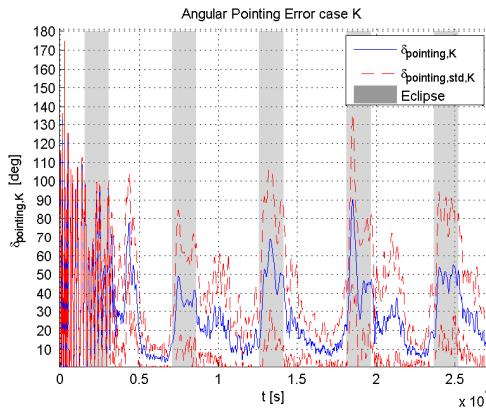


Figure 5-23: Angular Pointing Error of case K

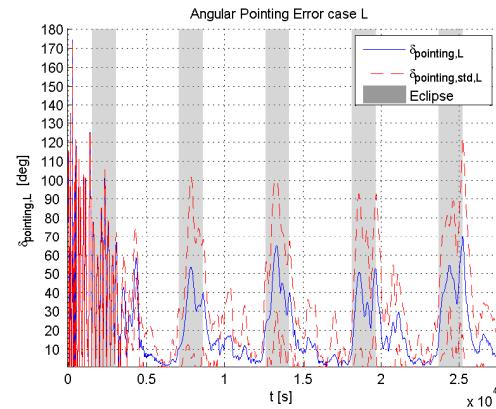


Figure 5-24: Angular Pointing Error of case L

tuning method where the relative allowed error after settling time has been reached e_{rel} has been increased by 0.02 and the settling time T_s has been increased by 200 seconds after each performed case. For the Ziegler-Nicols tuning method the ultimate gain K_u is determined to be 0.003 N·m and the oscillation period of the output of the controller is 70 seconds.

Cases I As can be seen in figure 5-21 and table 5-10, the Ziegler-Nicols tuning method does not converge the angular pointing error to values near zero.

Cases J to N These cases have been tuned using the quaternion feedback regulator tuning method. In these cases the allowed relative error e_{rel} and the settling time T_s have been increased to find the gains which result in a simulation with the lowest angular pointing error. By increasing $e_{allowed}$ and T_s , the ratio $\frac{K_d}{K_p}$ increases as well, which results in more damping. However, too much damping will have a negative effect on the angular pointing error.

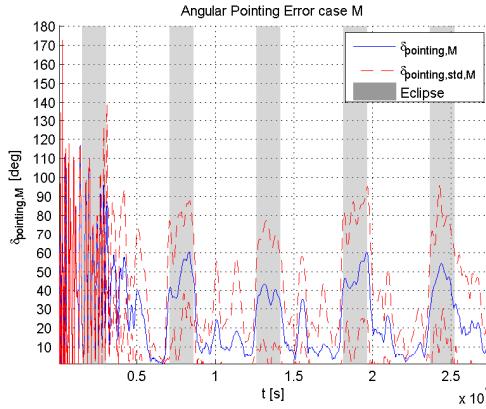


Figure 5-25: Angular Pointing Error of case M

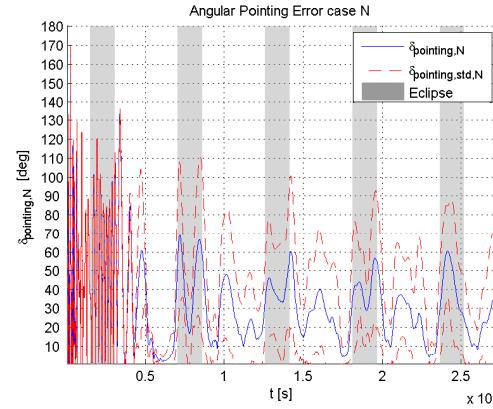


Figure 5-26: Angular Pointing Error of case N

Table 5-10: Mean values of the Monte Carlo simulations for cases I to N

Case	Mean $\delta_{pointing}$ [deg]		Mean of the standard deviation of $\delta_{pointing}$ [deg]	
	Last orbit	Last eclipse	Last orbit	Last eclipse
I	45.1	52.1	25.1	31.2
J	39.0	52.2	28.3	35.4
K	19.3	45.4	18.0	31.2
L	12.7	43.8	14.3	31.5
M	14.3	44.8	15.1	30.7
N	23.1	43.5	20.8	26.9

Best case It can be concluded the ratio $\frac{K_d}{K_p}$ is not optimal for the case of the Ziegler-Nicols tuning method. The Quaternion Feedback Regulator tuning method does have a optimal ratio of $\frac{K_d}{K_p}$, namely case L.

5-2-2 Performance Analysis

In this subsection the performance of the velocity-pointing mode will be investigated using different cases. The simulation parameters are identical to the parameters used in the previous section for determining the filter parameters and control gains. In addition, the filter parameters and control gains of case E and L will be used. All cases will be simulated without the mode determination algorithm except for case S. This means that for all cases except for case S only the velocity-pointing mode will be used. Moreover, the control will be turned off during eclipse due to the higher angular estimation error.

The cases that will be investigated are:

Case P This case will investigate the performance of the velocity-pointing control and attitude determination algorithm where:

- The magnetic torquer does not have any limitations and can produce a high magnetic dipole moment.
- Attitude and rotational rate information is taken directly from the environment model instead of using the extended Kalman filter. This will result in using the true values of the attitude and the rotational rate instead of the estimated values.
- No noise in magnetometer. Even though the magnetometer measurement data is not used for the extended Kalman filter in this case, it is used for the calculation of the magnetic dipole moment of the magnetic torquer for a given control torque.
- No disturbance torques.
- The time to perform one control loop, t_{total} , will decrease from 0.5 s to 0.2 s.

This resembles a case with ideal conditions.

Case Q The IGRF look-up table will be programmed in the service layer of the ADCS software and will be determined using TLEs and propagated using the SGP4 model as explained in subsections 3-5 and 3-6-1. For the purpose of these simulations the IGRF look-up table has been simulated using the real position information obtained from the simulator. In order to simulate the inaccuracy of the IGRF look-up table, the values obtained from the geomagnetic dipole model will be biased in this case. The accuracy of the IGRF look-up table is determined by the resolution of the latitude and the longitude. The bias has been set to be $0.7 \cdot 10^{-5}$ T in all three axis in case Q1. This translates to a jump in the longitude of about 1 degree on average. This is the same IGRF table accuracy as used on the Delfi-n3Xt satellite [10]. It should be noted, however, that this bias is the worst case bias. In reality the IGRF look-up table is discretized, so at the points where the table provides a magnetic field strength the bias will be zero and it will increase when the position of the satellite is such that it is in between two datapoints. In addition, a bias of $1.4 \cdot 10^{-5}$ T in all three axis will also be analyzed (case Q2), which is two times higher than the bias in the Q1 case.

Case R The moment of inertia of the satellite will be measured on the ground, however it can change in orbit due to for example propellant sloshing or failure in deployment of one or both of the solar panel wings. The performance of the satellite will be checked for the case where the moment of inertia of the satellite and the moment of inertia as inserted in the attitude estimation algorithm are different. The difference between these two matrices needs to be small for correct functioning of the time propagation step of the extended Kalman filter as given in equation 4-6 in section 4-2. The moment of inertia which is inserted in the extended Kalman filter will be biased with the following moment of inertia error matrix:

$$\mathbf{U} = \begin{bmatrix} 0.0010 & 0.0014 & 0.0015 \\ 0.0014 & 0.0010 & 0.0013 \\ 0.0015 & 0.0013 & 0.0020 \end{bmatrix} \left[kg \cdot m^2 \right] \quad (5-5)$$

The perturbed moment of inertia U has been obtained by multiplying the expected moment of inertia of the DelFFi satellite given in table 2-1 by a perturbation factor between 0.02 and

0.03. This perturbation factor is the same as used for the analysis of the European Student Earth Orbiter (ESEO) satellite [37]. The moment of inertia which is inserted in the filter is:

$$\mathbf{J}_{EKF} = \mathbf{J}_{measured} + \mathbf{U} \quad (5-6)$$

Case S The thruster will be fired for a 5 minute period before eclipse in each orbit. A maximum of 5 minute is a suggestion by the thermal control subsystem engineer such that the satellite's temperature can be kept within bounds [38]. For these simulations the mode determination algorithm will be turned on such that the ADC algorithm can autonomously determine its control mode.

It should be noted that during nominal conditions, the velocity-pointing mode will not be used for formation flying and hence the thruster will not be used. However, in case the reaction wheels malfunction, this case will show whether the velocity-pointing mode can be used as a back-up solution for performing formation flying.

Analysis cases P to S

Figures 5-27 and 5-28 give the angular estimation error and the angular pointing error of the satellite in nominal conditions. The parameters used for the nominal condition are mentioned in the first paragraph of section 5-2. These plots will be used to compare the cases P to S with a simulation with nominal conditions.

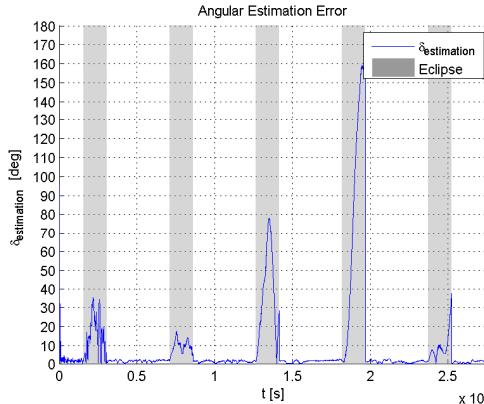


Figure 5-27: Angular Estimation Error of a case with nominal conditions

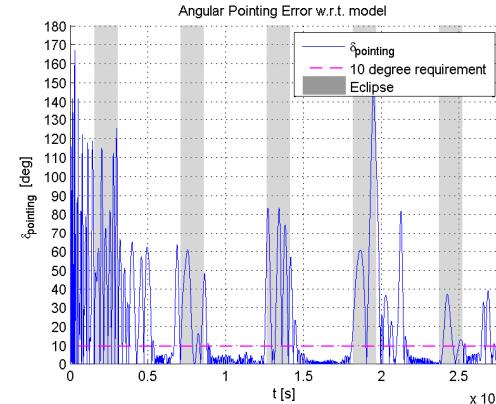


Figure 5-28: Angular Pointing Error of a case with nominal conditions

Case P As can be seen in figure 5-29 and table 5-11, the angular pointing error decreases significantly when the magnetic torquer, magnetometer and attitude estimation algorithm are ideal, when there are no disturbance torques acting on the spacecraft and the time to complete one control loop decreases. When the magnetic torquer does not have any limitations and there are no disturbance torques, the available torque is higher and the required torque is lower so the spacecraft will be able to change the attitude much faster. Increasing the performance of the magnetometer will result in a more accurate determination of the magnetic dipole moment

of the magnetic torquer which is required in the velocity-pointing control, as can be found in equation 4-22. Using the attitude and rotational rate from the environment model instead of the EKF, results in more accurate attitude estimation which leads to more accurate pointing. Decreasing the time to complete one loop, t_{sensor} results in a higher update frequency of the magnetic dipole moment which is being fed into the environment model causing the attitude to be controlled at a higher frequency. It should be noted that there is, just like in all other cases, no control during eclipse, therefore the angular pointing error increases during this period. When the satellite is back in the Sunlit phase of the orbit, it will start the velocity-pointing control again. However, since the attitude and the rotational rate of the satellite are not the same after each eclipse, the time to reduce the angular pointing error differs.

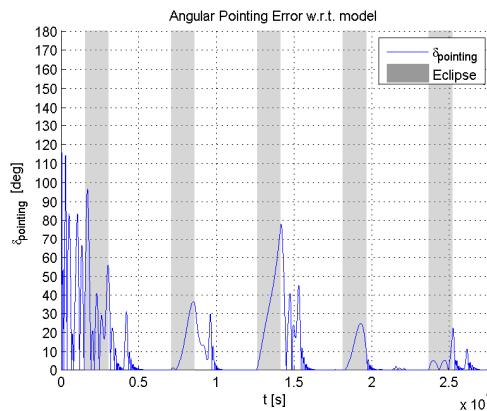


Figure 5-29: Angular Pointing Error of case P

Case Q The angular estimation error of cases Q1 and Q2 are given in figures 5-30 and 5-32 respectively. The angular pointing error of cases Q1 and Q2 are given in figures 5-31 and 5-33 respectively. Comparing these plots with the simulation run with nominal conditions it can be seen that the angular estimation error increases drastically with peaks at 12 degree for case Q1 and 30 degrees for case Q2. Hence a decrease in the accuracy of the IGRF look-up table lowers the performance of the estimation algorithm. The satellite is able to point to the desired attitude with an accuracy of 3.1 degree for case Q1 and 9.4 degree for case Q2 during the last 10 minutes of the last orbit of the simulation, as can be found in table 5-11.

Case R As can be seen in figure 5-34, where the angular estimation error of case R is given, it can be concluded that an inaccurate determination of the satellite's moment of inertia increases the angular estimation error during eclipse. This effect is not observed during the Sunlit periods. This can be explained by the fact that the eclipse periods rely heavily on the time propagation stage where the satellite's moment of inertia is used in the determination of the angular acceleration of the satellite. The Sunlit periods also use the time propagation stage, however unlike the eclipse periods, the Sunlit periods use the Sun direction vector, in addition to the geomagnetic field vector, which results in an attitude estimation using two vectors. The use of two vectors greatly improves the estimation. The control is turned off during eclipse. Therefore this increase in angular estimation error does not have an effect on the angular pointing error.

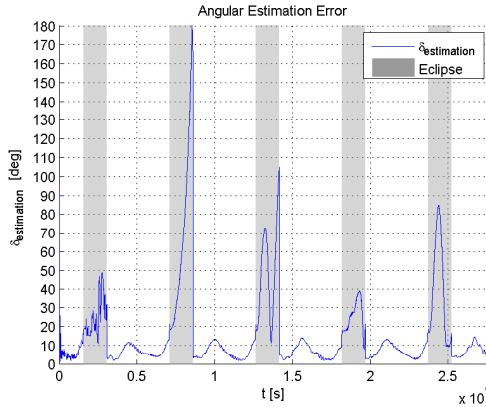


Figure 5-30: Angular Estimation Error of case Q1

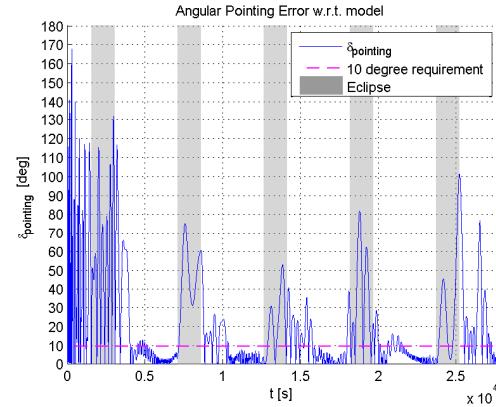


Figure 5-31: Angular Pointing Error of case Q1

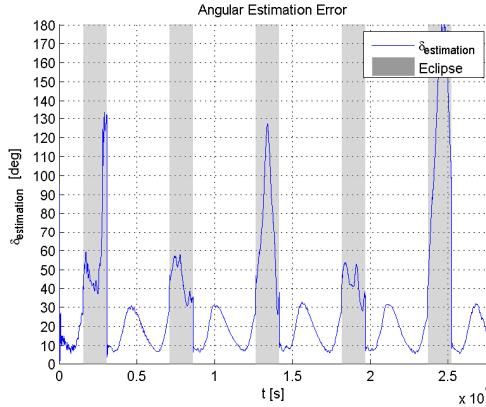


Figure 5-32: Angular Estimation Error of case Q2

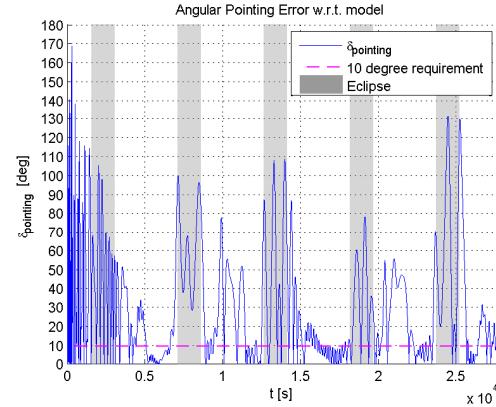


Figure 5-33: Angular Pointing Error of case Q2

Case S Figures 5-36 and 5-37 show the angular velocity and the angular pointing error of cases S. It should be noted that the gaps in the angular pointing error plots are caused by the fact that the detumble mode is turned on during those periods. The detumble mode does not use a desired attitude for control, therefore there is no angular pointing error. The angular pointing error average in the last 10 minutes before the last thrusting period for a simulation of 5 orbits is given in table 5-11. As discussed before, the thruster has been fired for a 5 minute period just before the eclipse when the satellite has been detumbled in each orbit.

As can be seen in figure 5-36 the satellite will start to tumble right after starting the simulation. After a brief period of running the velocity-pointing mode, the mode determination algorithm will set the control mode to the detumble mode. While the satellite is in the detumble mode, the thruster will not be turned on. When the satellite is detumbled and the thruster is turned on, the plot shows that the satellite immediately starts rotating. This increases the tumble parameter P_{tumble} and hence the detumble mode will be triggered again by the mode determination algorithm. As can be seen from figure 5-37, the satellite is pointing in the velocity vector with a pointing error of 10 degree or less when the thruster is turned

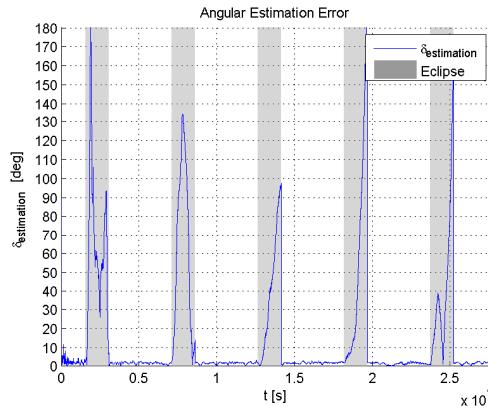


Figure 5-34: Angular Estimation Error of case R

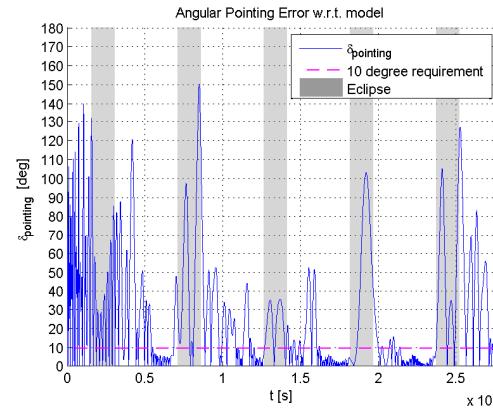


Figure 5-35: Angular Pointing Error of case R

on. This shows that the moment in time when the thruster is turned on (just before eclipse) is sufficient for the velocity-pointing mode to point the satellite in the velocity vector after recovering from the previous thruster firing. Another advantage of thrusting before eclipse is that the eclipse period can be used to reduce the high rotational rates during the detumble mode. Since it is not possible to control during eclipse in the velocity-pointing mode, the detumble mode is ideal to run during eclipse. Figure 5-38 shows the angular pointing error zoomed in on the last thrusting period for a simulation of 5 orbits. Here it can be seen that due to the firing of the thruster, the angular pointing error increases to 52 degree. This angular pointing error of 52 degree is higher than the required 2 degree pointing error given in table 2-6. Thrusting with an angular pointing error of more than 2 degree produces an unreliable thrusting force in the along-track direction. This shows that reaction wheels are essential in keeping the desired attitude. Therefore it can be concluded that firing the thruster for formation flying with only magnetic torquers will not contribute to an accurate orbit manoeuvre. For the purpose of these simulations $t_{detumbled}$ has been set to zero such that the mode determination algorithm has a shorter response time when the satellite needs to enter velocity-pointing mode.

Table 5-11: Mean value of the angular pointing error of the last 10 minutes before the last eclipse (or in case S, before the last thrusting period) for a simulation of 5 orbits

Case	$\delta_{pointing}$ [deg]
Nominal	2.1
P	0.1
Q1	3.1
Q2	9.4
R	3.3
S	8.0

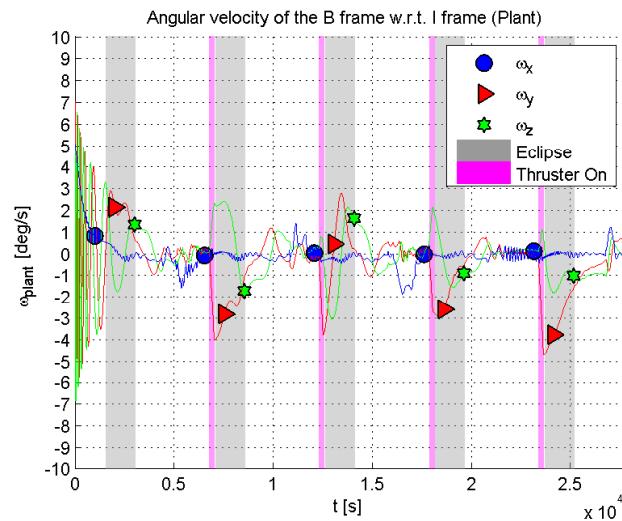


Figure 5-36: Angular velocity of case S

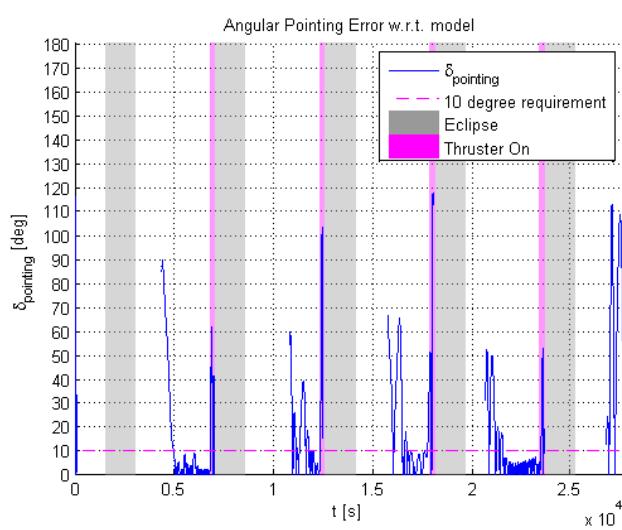


Figure 5-37: Angular Pointing Error of case S

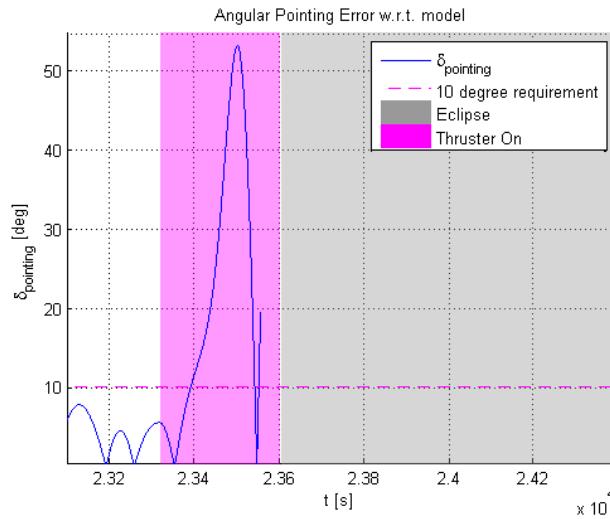


Figure 5-38: Angular Pointing Error of case S. Zoomed in on the last thrusting period

5-3 Requirement Verification and Recommendations

This chapter has presented the simulation results of the detumble mode and the velocity-pointing mode. This section will verify the requirements and provide recommendations.

Detumble Mode

The detumble mode can achieve a rotation rate of less than 1 deg/s in all axes in nominal conditions for an initial rotation rate of up to 45 deg/s in all axes within 6.75 orbits which is equal to 0.4 days. This meets the requirements presented in [35], which dictates that the cubesat shall be able to recover from tip-off rates up to 10 deg/s within 2 days. The 35 deg/s higher initial angular velocity is a result of an uncertainty in the requirements at the start of the design process. In addition, for a different actuator operation time and hold time, it also meets the new requirements set by QB50 for which the satellite needs to recover from tip-off rates of up to 50 deg/s within 3 days for nominal conditions and 90 deg/s in non-nominal conditions.

Velocity-pointing Mode

In this chapter the velocity-pointing mode has been simulated with an initial rotation rate of 5 deg/s and an arbitrary attitude. It has been shown that the satellite can point in the velocity vector with an angular pointing error of less than 10 deg in nominal conditions during at least half of the Sunlit phases (figure 5-28). It is decided to not control the magnetic torquers during eclipse periods, because the angular estimation error increases rapidly during the eclipse periods. Due to the fact that the satellite is not pointing in the desired attitude after eclipse it will take some time to point in this attitude again. The time it takes until the satellite is pointing in the desired attitude differs. The angular pointing error will be below 10 degree during at least the second half of the Sunlit phase. It can be concluded that

the velocity-pointing mode only meets the requirement (requirement MIS-C-M.1 of appendix A-1) of pointing the payload in the velocity vector during the second half of the Sunlit phase.

Recommendations

It is clear that the fact that the requirement can not be met is due to the sensor limitation during eclipse periods. A possible way to solve this issue is by adding a sensor that does not have the same limitation as the Sun sensors. This will result in a higher angular estimation accuracy during eclipse periods. A suggestion would be to include a gyroscope or Earth horizon sensors in the design. The reference vector measured by one of these sensors would replace the Sun direction vector measured by the Sun sensors. With a continuously high angular estimation accuracy, the control algorithm would stay on continuously. This will result in a high angular pointing accuracy for both Sunlit periods and the eclipse periods, thus meeting requirement MIS-C-M.1 (appendix A-1).

It can be concluded that the requirement of pointing in the velocity vector at all times will not be met during eclipse periods and the first half of the Sunlit phase with the current choice of sensors. If the requirement is adjusted to only point in the velocity vector in the second half of the Sunlit phase, then the mission can continue. In this case the scientific payload will be able to perform measurements for the duration of half of the Sunlit phase. This corresponds to about 35% of the orbit. The next chapter will verify the ADC algorithm based on the fact that the mission will continue even with the limited pointing performance.

Chapter 6

Testing of the ADC Algorithm

This chapter will be focussed on testing the ADC algorithm using a software-in-the-loop (SIL) simulation and testing a part of the algorithm on the processor. In addition, before starting with the testing phase, some typical issues in coding will be discussed which has been taken into account in the algorithm. The SIL simulation will be performed to find errors in the algorithm due to the code generation process. The focus of the test on the processor will be to demonstrate whether the algorithm on the processor produces the same outputs as the algorithm in Simulink.

For generating code which is suitable for the flight hardware of the DelFFi mission, it has been chosen to generate C code from the algorithm designed in Matlab/Simulink using the Simulink Embedded Coder™. This method of code generation has become increasingly widely used in generation of code for software which needs to run on critical components, especially in the aerospace industry [39]. The advantage of generating code from a pre-designed algorithm is the re-usability of the code. It is more convenient for a programmer to adjust a code in Simulink and run the simulation to see the implications of the adjustment rather than adjusting a C code.

6-1 Software Issues

The designed algorithm has been specifically designed with the idea that it should work as a black box. It should require a certain input coming from the sensors and generate a certain output which is intended for the actuator with as little interference in the code during execution as possible. The performance of the algorithm may, however deviate from this ideal case. Two issues which can cause this are discussed in this section.

Shared-data problem

When a mode switch occurs during the execution of a control mode, the performance of the algorithm should not be altered compared to a situation where a mode switch did not occur.

This means that the entire algorithm, from the point where a mode has been determined until the end of the algorithm where an output is generated for the input of the actuator, should be atomic in order to prevent the problem of shared-data. A process is called atomic when the process can not be interrupted.

In order to avoid this problem, the control modes and the safe mode will not be initiated by an interrupt. Instead the control mode mode will be determined in the mode determination algorithm. This algorithm will check whether the flag for the safe mode has been enabled to activate the safe mode or activate one of the control modes. Which control mode is activated is based on the mode determination algorithm, as explained in chapter 4. If the flag of the safe mode is enabled during the execution of one of the control modes, the execution will first need to finish before the flag is checked again during the next loop.

Task overrun

Task overruns occur when a processor requires extra processing and as a result requires more time to execute than normally. Hence, a new task might be executed before the previous task has been completed. The problem of task overruns will be solved by using flags in the code. The task overrun flag is enabled at the start of the execution of the code. When this task finishes, the flag is disabled. Before setting the flag, the status of the flag will be checked. If the flag is enabled, meaning that the task is not finished yet, then the algorithm will continue to run until the task has been finished. To prevent the algorithm from deadlocking during the execution of a task, the amount of times the next loop of the algorithm is skipped will be measured. If this value exceeds a certain threshold, then the task overrun will be reset such that the code can start with a new task.

6-2 Algorithm Testing

This section will discuss the software-in-the-loop testing and testing the algorithm on the processor

6-2-1 Software-in-the-loop Testing

Software-in-the-loop testing, also called SIL testing, is a method to test the C code of the algorithm directly inside the simulation program. Testing directly on hardware is usually quite expensive and hardware is mostly available later on in the project phase. Software-in-the-loop tests therefore provide an agile and low-cost solution to testing code. The purpose of the SIL test is to test the code in an inexpensive way to check for bugs associated with C code. These are the type of bugs that do not occur in the Matlab/Simulink language.

The C code of the algorithm is generated using the standard S-function package in Simulink. S-functions provide a fast way to generate C code for SIL testing. This C code is embedded in the S-function Simulink block and can be integrated in the model to replace a part of the code that is written in Simulink. For the software-in-the-loop tests, the following cases will be discussed:

Case A In this case the algorithm in Simulink will be simulated. This case will be used to compare the results of the other cases with results obtained from a simulation of the Simulink code. A visualisation of case A can be found in figure 6-1.

Case B In this case the algorithm converted in separate C code files will be simulated using a software-in-the-loop test. These separate blocks of C code are:

- Mode determination algorithm
- Safe mode algorithm
- Detumble mode algorithm
- Velocity-pointing mode control algorithm
- Extended Kalman filter algorithm. The storage of the state vector and covariance matrix output that is used as an input in the next loop in the filter is coded in Simulink instead of C code. This storage of the output of the filter is done using a memory block which can be found in the package for discrete Simulink blocks in the Simulink toolbox library.

This case will run twice. The first run will be with the C code in the closed loop (case B1). The second run will be with the Simulink code in closed loop and the C code in an open loop (case B2) such that both the results of the Simulink code and the C code can be compared with each other. A visualisation of case B1 and B2 can be found in figures 6-2 and 6-3.

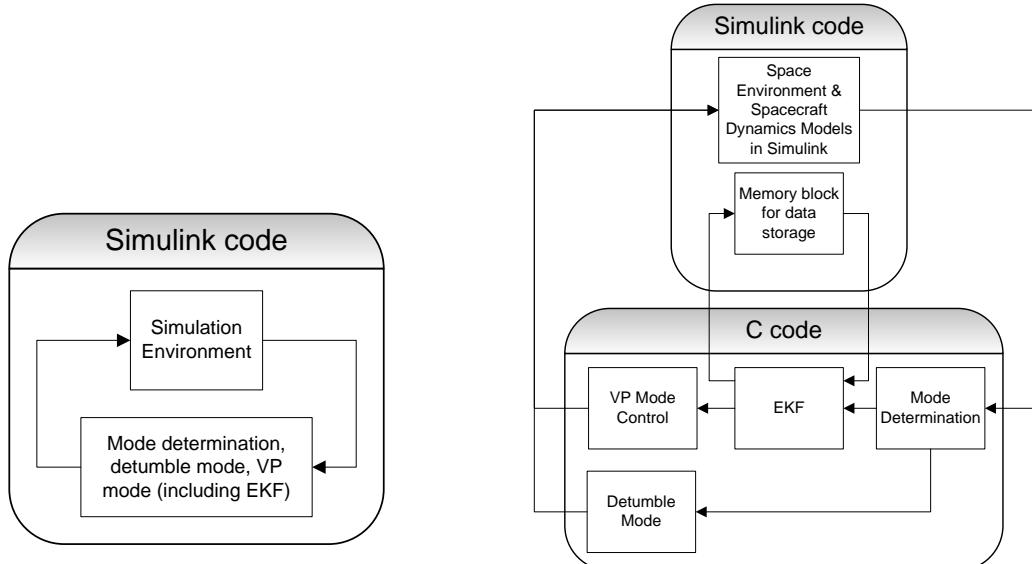


Figure 6-1: Visualisation of the SIL test set-up for case A

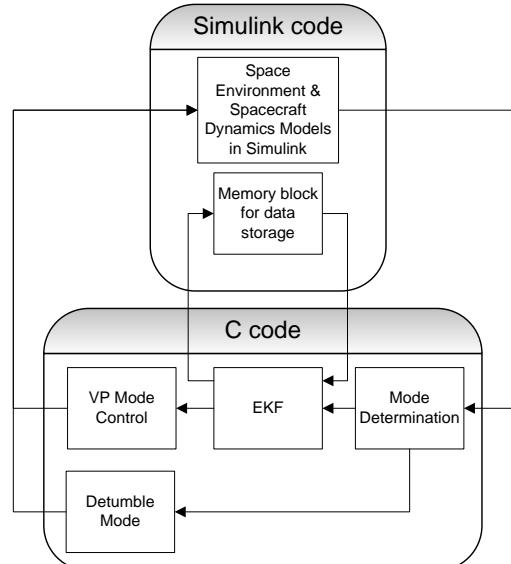


Figure 6-2: Visualisation of the SIL test set-up for case B1

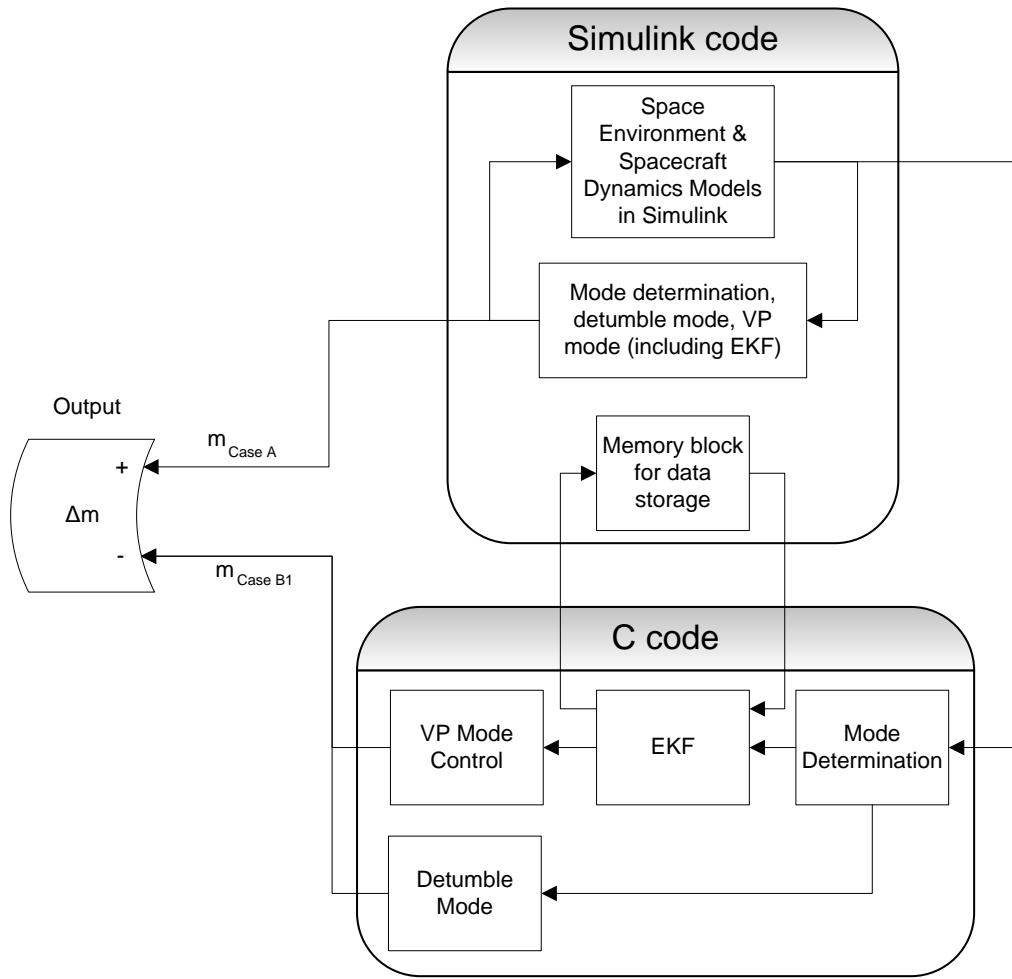


Figure 6-3: Visualisation of the SIL test set-up for case B2

Case C This case is similar to case B1 with the only difference being the fact that the C code for the Extended Kalman Filter also stores the value of the output to be used as an input to the filter in the next loop. This means that the memory block is coded in C instead of using the memory block of the Simulink toolbox library. A visualisation of case C can be found in figure 6-4.

Case D In this case the algorithm is converted in a single C code file and will be simulated using a software-in-the-loop test. A visualisation of case D can be found in figure 6-5.

The parameters for the satellite and orbit properties that have been used in this chapter can be found in table 2-1 and 5-2. The timing interval of the sequence of events of the ADCS is case E of table 5-1 and the Bdot algorithm parameters are found in table 5-3. In addition $L_{detumbled}$ and $t_{detumbled}$ of the mode determination algorithm are set to be $8 \cdot 10^{-7}$ T/s and 100 s respectively, as discussed in chapter 5-1. The initial rotational rate is equal to 5 degree per second in all three axes. In addition the filter parameters and control gains of case E and

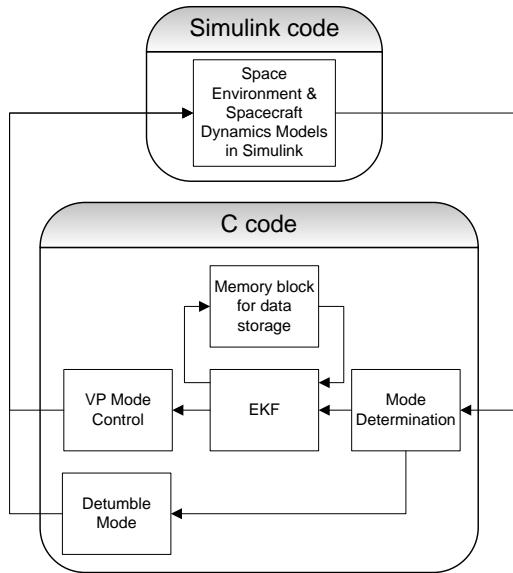


Figure 6-4: Visualisation of the SIL test set-up for case C

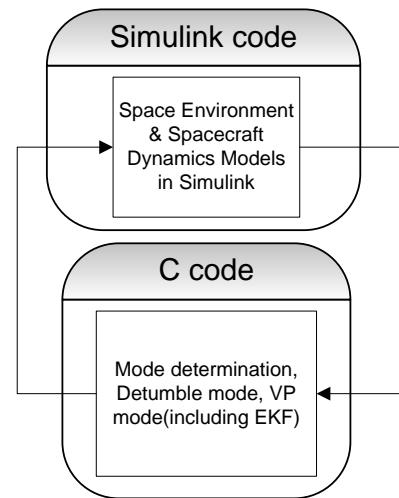


Figure 6-5: Visualisation of the SIL test set-up for case D

L are used, as discussed in chapter 5-2.

Analysis SIL Test

Case A The rotational rate of the spacecraft is given in figure 6-6 and the mode determination variable is given in figure 6-7, which shows which mode is executed. A mode determination variable equal to 0 indicates that the detumble mode is executed and 1 indicates that the velocity-pointing mode is executed. Figure 6-8 and 6-9, show the angular estimation error and the angular pointing error, respectively. This case will be used to compare it with the other cases.

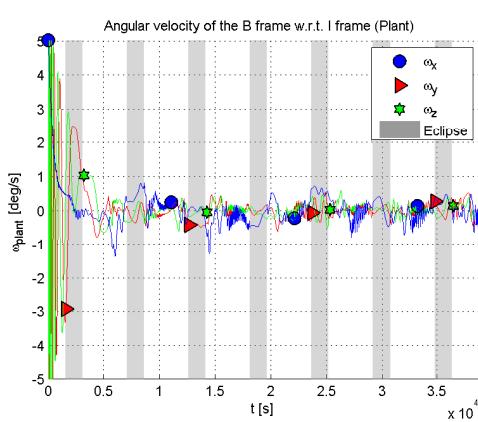


Figure 6-6: Rotational rate in case A

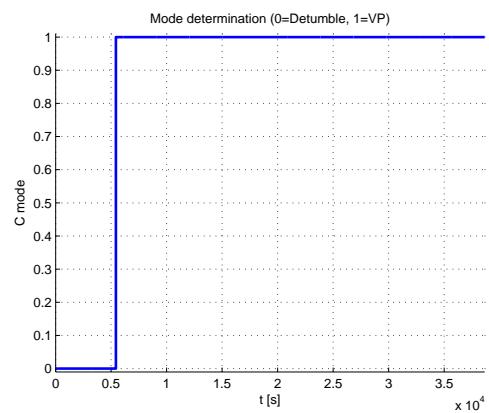


Figure 6-7: Mode determination variable in case A

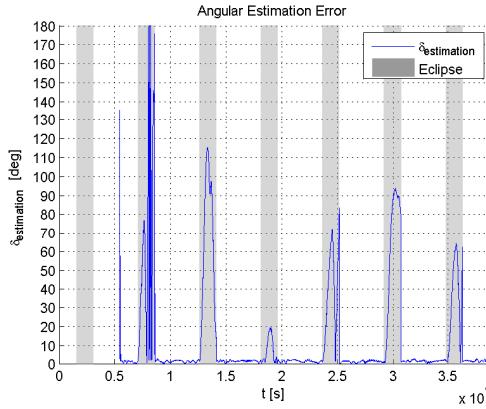


Figure 6-8: Angular Estimation Error in case A

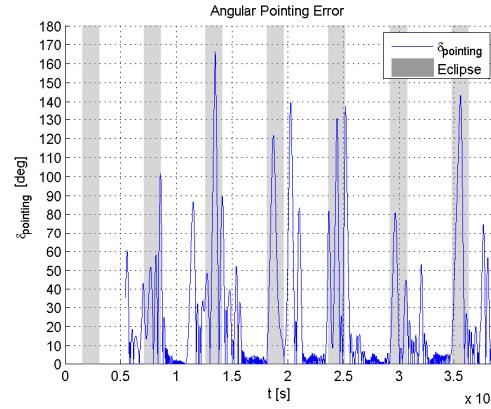


Figure 6-9: Angular Pointing Error in case A

Case B1 The rotational rate, the mode determination variable, angular estimation error and angular pointing error are given in figures 6-10, 6-11, 6-12 and 6-13, respectively. Comparing these to their counter parts in the previous case (case A), it can be concluded that the simulation of this case gives similar results as the simulation of case A during the detumble mode. The behaviour of the algorithm will, however, differ in the velocity-pointing mode, but the inaccuracies are small enough for the algorithm to still control the spacecraft within the bounds of the requirements. The cause of these inaccuracies will be visible in the results obtained in case B2.

Case B2 The purpose of this case is to find the accuracy of the commanded magnetic dipole moment that is generated by the C code of case B1. This C code will be executed in an open loop with the Simulink code of case A as has been depicted in figure 6-3. The accuracy of the commanded dipole moment is given by:

$$\Delta \mathbf{m} = \mathbf{m}_{\text{case_A}} - \mathbf{m}_{\text{case_B1}} \quad (6-1)$$

Where $\mathbf{m}_{\text{case_A}}$ and $\mathbf{m}_{\text{case_B1}}$ are the commanded dipole moment of the simulations of case A and case B1 in open loop, respectively. The plot of this accuracy of the commanded magnetic dipole moment is given in figure 6-14. In this figure the magnetic dipole moment of each axis can not be distinguished separately, however this figure shows that the accuracy of the simulation of the C code of case B1 is high during the detumble mode. This accuracy decreases when the amount of arithmetic operations increases, hence the inaccuracy of the velocity-pointing mode is evident. This is especially the case at the start of the velocity-pointing mode where the angular estimation error is high, as can be seen in figure 6-8. The peak just after the second eclipse in figure 6-14 reaches an order of $10^{-9} \text{ A}\cdot\text{m}^2$. This explains why the simulation results of the velocity-pointing mode of case B1 differ with respect to the simulation results of case A. The inaccuracy is however not high enough to introduce high pointing errors as has been shown in case B1.

Case C The results of the simulation of this case are similar to the results of the simulation of case B1 during the detumble mode. The mode determination algorithm sets the control

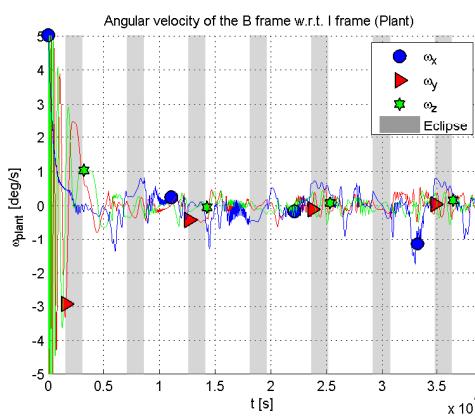


Figure 6-10: Rotational rate in case B1

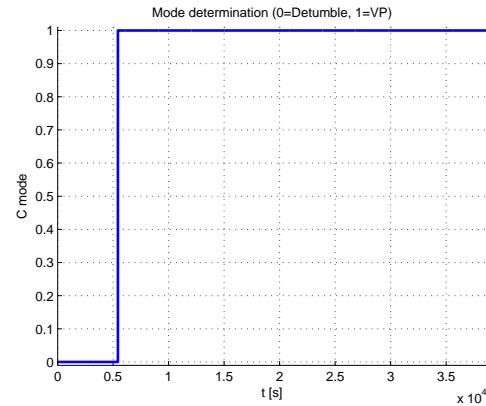


Figure 6-11: Mode determination variable in case B1

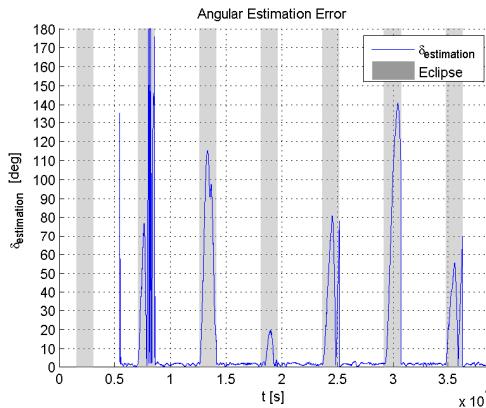


Figure 6-12: Angular Estimation Error in case B1

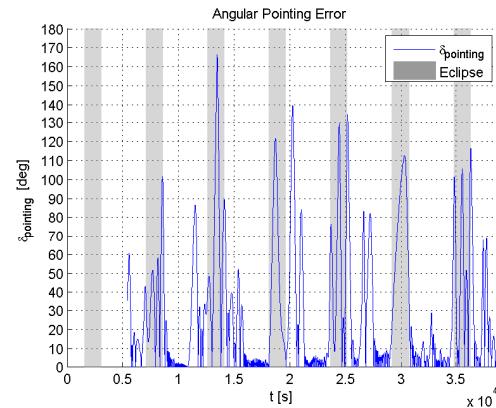


Figure 6-13: Angular Pointing Error in case B1

mode to the velocity-pointing mode after the satellite has been detumbled. However, the EKF can not estimate the attitude correctly during the velocity-pointing mode. It can be seen from table 6-1 that the EKF of case C gives identical results as the EKF of case A in the first loop. This result will be stored and used in the second loop. However, the input to the EKF is incorrect during the second loop in case C, which results in an error in the quaternion. This is caused by an incorrect storage of data on the processor.

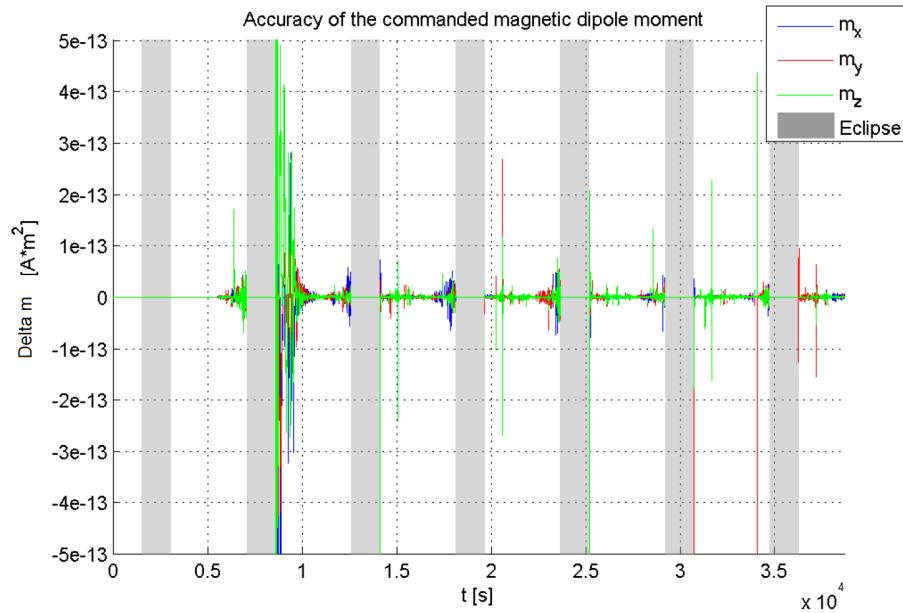


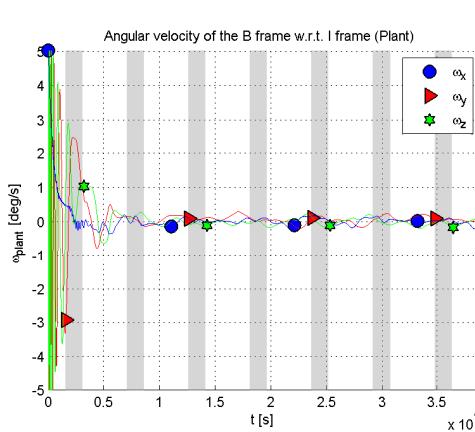
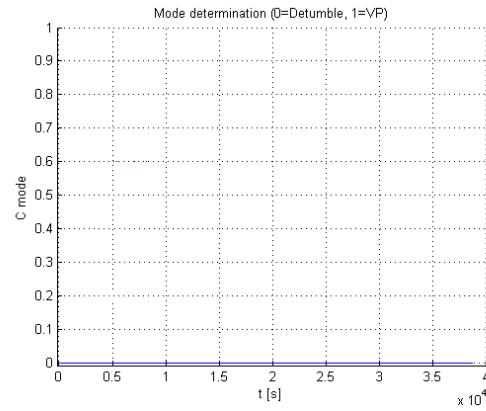
Figure 6-14: Accuracy of the commanded magnetic dipole moment (case B2)

Table 6-1: Results of the quaternion obtained from the EKF in case A and in case C

	q1		q2		q3		q4	
Output EKF	Case A	Case C						
first loop	0.0413	0.0413	0.0289	0.0289	0.0152	0.0152	0.9986	0.9986
secoond loop	0.0920	0.4753	0.0617	0.3425	0.0233	0.1672	0.9936	0.7930

Case D The rotational rate and the mode determination variable of case D can be found in figures 6-15 and 6-16, respectively. In this SIL simulation the velocity-pointing mode will never be initiated as can be seen in figure 6-16. This is caused by the fact that the counter in the mode determination algorithm, which counts the time since the tumble parameter P_{tumble} is lower than $L_{detumbled}$, is reset after each loop which results in the counter never exceeding the required detumble time $t_{detumble}$. The variables $L_{detumbled}$ and $t_{detumble}$ have been explained in section 4-1. In the simulation in case A this counter is only reset when the Simulink block which contains the counter is executed again after not being executed in the previous loop, because this indicates that the detumble mode has been initiated again. Due to the fact that the C code is enclosed in one file instead of multiple files, the processor will not be able to detect whether the code that contains the counter has been executed already. This causes the counter to reset after each loop, hence the ADC algorithm remains in the detumble mode.

Best case Case A is a simulation of the algorithm in Simulink and has been used to compare it to the different cases and will therefore not be considered in the selection. Case B1, C and D have presented three different configurations for the C code of the ADC algorithm. Case B2 has been used to determine the error between the commanded magnetic dipole moment

**Figure 6-15:** Rotational rate in case D**Figure 6-16:** Mode determination variable in case D

between case A and B1 and will therefore not be used in the selection of the best case. Case C and D both showed errors in the code that are caused by the C code generation process. Case B1 showed results similar to case A. Therefore it has been chosen to use the C code of case B1 in the implementation of the algorithm on the processor.

6-2-2 Testing the ADC Algorithm on the Processor

The algorithm that will be used on the processor has been generated using the Simulink Embedded CoderTM. For the generation of the algorithm code a 32-bit Embedded Processor has been chosen as the target hardware. This means that the target hardware is optimized for arithmetic operations involving variables with 32 bits. Since the Simulink simulations use doubles with 64-bit precision, this means that the target hardware will run slower than a 64-bit Embedded processor, however the precision of the variables will be the same as the Simulink simulations. It should be noted that the time to execute the algorithm will not be measured. This is considered to be changing depending on the hardware, therefore it is suggested that the time to execute the algorithm should be measured on the flight hardware.

Hardware

The processor that will be used to test the algorithm should meet the requirements as stated in requirements analysis of the DelFFi mission. Based on these requirements, a trade-off has been performed to determine a suitable processor for the testing of the ADCS algorithm by J. Viru [40]. The hardware that will be used to test the algorithm is the TivaTM C series TM4C1294 Connected LaunchPad Evaluation BoardTM. It uses a TM4C1294NCPDT microcontroller-unit which is a 32-bit, 120 MHz ARM Cortex-M4 CPU with floating point. The development board TM4C1294 Connected LaunchPadTM has a 32-bit CPU. This means that the microcontroller unit is optimized to perform single precision floating point arithmetic operations. As explained before, it is expected that the inaccuracy of the results between the simulations of the algorithm in Simulink and the algorithm on the processor are negligible.

Processor Test Set-up

For the testing of the ADC algorithms on the processor, first an overview of the arithmetic operations used for the algorithms in Simulink is generated. This is given in table 6-2.

Table 6-2: Overview of the Simulink functions used in the ADC algorithms

Operation	Mode determination algorithm	EKF time propagation algorithm	EKF measurement update algorithm (2x)	Detumble mode	Velocity-pointing mode control algorithm
Addition, subtraction	1	2	5	0	11
Gain	0	21	9	1	41
Cross product 3x1 vectors	0	1	0	0	1
Dot product	0	0	2	0	0
Matrix multiplication	0	5	11	0	11
Matrix exponential	0	2	0	0	0
Matrix transpose	0	1	6	0	0
Matrix inverse	0	1	1	0	0
Square-root	0	0	2	0	4
Trigonometric function	0	0	0	0	9

The Cortex-M4 CPU of the board is able to perform basic arithmetic operations as well as trigonometric functions and common math functions such as the square-root function. This is given in the documentation of the Tiva C Series microcontroller [41]. What remains to be investigated are the matrix operations. These matrix operations are used in the time propagation step of the EKF algorithm. Therefore for the processor tests it has been chosen to test the algorithm which performs the time propagation step, because it contains the Simulink blocks which need to be investigated for correct performance. The parameters and satellite properties that will be used for running tests on the processor of the time propagation step algorithm are: the moment of inertia \mathbf{J} is given in table 2-1, the sensor measurement update frequency Δt is 0.5 seconds and the initial quaternion and angular velocity are given as follows:

$$\mathbf{q}_0 = \begin{bmatrix} 0.378 \\ -0.378 \\ 0.756 \\ 0.378 \end{bmatrix} \quad \omega_0 = \begin{bmatrix} 5^\circ \\ 5^\circ \\ 5^\circ \end{bmatrix} = \begin{bmatrix} 0.0873\text{rad} \\ 0.0873\text{rad} \\ 0.0873\text{rad} \end{bmatrix} \quad (6-2)$$

Analysis Processor Testing

Running a the test on the processor, shows that executing the matrix exponential function in C results in an error, while the other matrix operations are executed without errors. Instead of using the converted code of the matrix exponential Simulink block in C, an approximation of the matrix exponential will be used. The matrix exponential function can be defined as follows: let X be an $n \times n$ real or complex matrix, then the exponential of matrix X is the $n \times n$ matrix given by the power series [13]:

$$e^X = \sum_{k=0}^{\infty} \frac{1}{k!} X^k \quad (6-3)$$

To check the performance of the time propagation step algorithm in Simulink using the approximation of the matrix exponential, three simulations will be executed, namely:

- A simulation of the time propagation step using the matrix exponential function of the Simulink toolbox library.
- A simulation of the time propagation step using an approximation of the matrix exponential function for $k = 0,1,2$.
- A simulation of the time propagation step using an approximation of the matrix exponential function for $k = 0,1$.

The performance of the EKF in Simulink with the varying matrix exponential functions are given in table 6-3. For all three cases the angular pointing error is below 10 degree during the second half of the Sunlit phase. Therefore it can be concluded that the pointing requirement is still met in all three cases. The table shows that the angular estimation error increases with 0.2 and 0.3 degree when the matrix exponential function is approximated using $k = 0,1,2$ and $k = 0,1$, respectively. The approximation with $k = 0,1,2$ has an increased computational effort of one matrix multiplication of two 4×4 matrices compared to the approximation with $k = 0,1$ which is insignificant compared to the total amount of 34 matrix multiplications used throughout the algorithm. Therefore the matrix exponential function on the processor will be approximated by taking the power series sum for k is equal to 0,1,2, because it provides a slightly higher accuracy without a high increase in computational intensity.

Table 6-3: Angular estimation error [deg] during Sunlit phase between the second last and last eclipse for a Simulink simulation of 7 orbits

Simulation type	$\delta_{estimation}$ [deg]	$\delta_{estimation}$ [deg]
	average	maximum
matrix exp simulink block	1.5	2.6
approximation of matrix exp ($k=0,1,2$)	1.7	2.6
approximation of matrix exp ($k=0,1$)	1.8	2.7

A test on the processor of the time propagation step algorithm for one loop is shown in table 6-4. The second column shows the output of the time propagation step when running the algorithm in Simulink with the matrix exponential Simulink block. The third column shows

the output of the time propagation step when running the algorithm on the processor with the approximation of the matrix exponential function using $k = 0,1,2$. The fourth column shows the error between the two outputs. There is an error observable in the output of the quaternion which is caused by the approximation of the matrix exponential. This error results in a higher angular estimation error as found with a simulation of the EKF in Simulink (table 6-3). However, the satellite will still be able to meet the pointing requirements as discussed in the previous paragraph.

Table 6-4: Output of algorithm in Simulink using the standard matrix exponential simulink block and an approximation to the matrix exponential function

Variable	Output using matrix exp simulink block	Output using approximation of matrix exp ($k = 0,1,2$)	Error
	Run in Simulink	Run on the processor	
q ₁	0.361234182071317	0.361230220535625	0.000003961535692
q ₂	-0.361234182071317	-0.361230220535625	-0.000003961535692
q ₃	0.780203561328580	0.780209391071250	-0.000005829742670
q ₄	0.361234182071317	0.361230220535625	0.000003961535692
ω_1	0.087300000000000	0.087300000000000	0.000000000000000
ω_2	0.089932809272727	0.089932809272727	0.000000000000000
ω_3	0.084667190727273	0.084667190727273	0.000000000000000

6-3 Recommended Testing

This chapter has focussed on testing the algorithm using a SIL test and a part of the EKF algorithm on the processor. For further testing it is recommended to perform a hardware-in-the-loop (HIL) test of the algorithm on the flight hardware. For this HIL testing the ADC algorithm will run on the flight hardware in a closed loop with the space environment and spacecraft dynamics models in Simulink, as depicted in figure 6-17. The code of case B1 will be used as the ADC algorithms in the HIL tests. The angular pointing error can be used to evaluate the performance of the ADC algorithm. It is important to check whether the time to execute the algorithms does not exceed the allocated time as indicated in table 5-1 which is almost equal to zero. In practice this translates to a few microseconds.

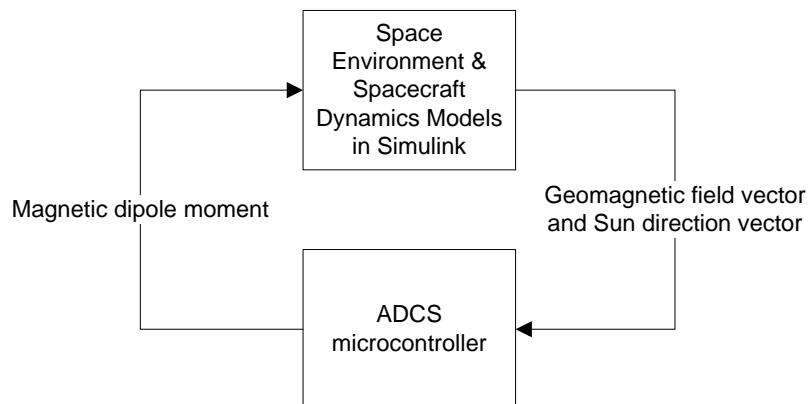


Figure 6-17: Hardware-in-the-loop testing of the ADC algorithm

Chapter 7

Conclusions & Recommendations

This thesis has presented the design of the algorithms for attitude determination and control and implementation and testing of this algorithm. The algorithm has been divided into control modes and one safe mode. The control modes are divided into a detumble mode and a velocity pointing mode. The Bdot controller for the detumble mode will be detumbling the satellite by decreasing the change in the geomagnetic field vector. The velocity-pointing mode will determine the desired attitude and will control the attitude to meet this desired attitude using a PD controller. The velocity-pointing mode will also include an extended Kalman filter (EKF) which will estimate the attitude and the rotation rate of the satellite. These modes will be initiated using a mode determination algorithm. This algorithm uses the tumble parameter and external parameters, such as the parameter which determines whether the safe mode is on or off.

For the purpose of testing the control mode algorithms and the mode determination algorithm, the Simulink model will contain a model of the space environment and the spacecraft dynamics and kinematics.

For determining the performance of the algorithm on the processor, the algorithm will be converted to C code and will be tested using a software-in-the-loop test and a test on the processor which resembles the flight hardware's specifications.

The objective of this thesis is to design, implement and verify the ADC algorithms which will run on the DelFFi satellites. For this objective two research questions have been formulated which will be discussed in the following two paragraphs.

What are the performance issues associated with the ADC algorithm in case of nominal, non-nominal and practical conditions?

In order to answer this question, extensive analysis has been performed on the algorithm for the detumble mode and the velocity-pointing mode. The answer will be therefore split up into two parts, where each part will discuss one of the modes.

Detumble Mode

The detumble mode is influenced by the time to complete one ADCS loop. By reducing the total loop time, the responsiveness of the ADCS improves. In addition, the rate at which the detumble mode converges to a fully detumbled satellite is dependent on the ratio of actuator time over total sampling time. This parameter determines the effectiveness of the magnetic torquers.

From the analysis of the non-nominal conditions it can be concluded that a 10 times increase in the noise of the magnetometer increases the angular velocity, but the satellite can still be detumbled. Furthermore, it can be concluded that decreasing the altitude from 380 km to 200 km does not have a significant effect on the performance of the detumble mode. This shows that even though the disturbance torques increase for a lower orbit, it does not outweigh the effect of the higher geomagnetic field strength which increases the control torque. Moreover, it can be shown that the satellite will not be able to detumble if it can not produce a torque around the x axis or if it can not produce a torque around both the y and z axes. This can occur in the case of a failure of the components of magnetic torquers in one of the axes.

Velocity-Pointing Mode

For the velocity-pointing mode it has been decided to use a PD controller and an EKF. It is, however, important to choose suitable control gain parameters and attitude estimation parameters to improve the performance of the controller and estimator.

From the analysis of the non-nominal conditions it can be concluded that increasing the inaccuracy of the IGRF look-up table decreases the performance of the attitude estimation filter. This inaccuracy is equivalent to a look-up value every 1 degree in longitude and latitude instead of a value for every combination of latitude and longitude. However, for both this inaccuracy and an inaccuracy two times higher it was shown that the satellite can still point in the desired attitude using the velocity-pointing mode. Moreover, it has been concluded that an inaccuracy in the moment of inertia results in minimal changes to the pointing performance. This is because the moment of inertia is only used in the time propagation step algorithm. This time propagation step algorithm is relied upon heavily during the eclipse periods. However, during the Sunlit phases the Sun direction vector is available which improves the estimation performance and therefore the estimation algorithm does not rely on the time propagation step as much as in the eclipse period. In addition, also analysis has been performed for the case that the formation flying concept will be demonstrated during velocity-pointing mode. It has been determined that the best time to fire the thruster is just before eclipse. This is the time where the pointing performance is at its maximum due to the fact that the satellite has been in the Sunlit phase for the longest consecutive amount of time, which results in the Sun direction vector providing the best estimation performance. Furthermore, firing the thruster just before eclipse means that the entire eclipse period can be used to detumble the satellite since it is not possible to start the velocity-pointing mode during eclipse due to the low estimation performance. It has been shown that the reaction wheels are essential for the formation flying concept due to the fact that, for the velocity-pointing

mode where the magnetic torquers are used exclusively, the pointing error increases rapidly when the thrusters are fired. This results in a poor pointing performance of the thrusters for a large part of the thrusting phase.

The velocity-pointing mode will not meet the requirement to point in the velocity vector at all times due to the availability of only one reference vector in eclipse periods. A suggestion to solve this has been given in the recommendations section of this chapter.

What are the sources of errors when running the ADC algorithm on the micro-controller and how can these be mitigated?

Code generation from a pre-designed algorithm is a very attractive means of producing code for a processor, because it results in re-usability of the code. In order to avoid some typical drawbacks with this code, the problems of shared-data problem and task overruns have been solved. The shared-data problem has been solved by designing the Simulink code in a way that the algorithm is atomic. The problem of task overrun has been solved by setting flags at appropriate locations in the code. If a flag has been enabled, meaning that the task is not finished yet, then the code will continue to run and skip the next task until the current task has been finished.

The best coding architecture which has been determined with a software-in-the-loop test is the architecture where all algorithms are separately converted in C code. These algorithms are the mode determination, safe mode, detumble mode, velocity-pointing mode control and the EKF. In addition also the three phases of the EKF, namely the time propagation step and the two measurement update steps, are separately converted in C code. This architecture avoids two issues:

- Simulink memory blocks will be used in the EKF, which are designed to store the value of the output of the filter. By implementing the Simulink memory block manually in the C code instead of converting the memory block directly into C code, the problem of storing values will be solved.
- Generating the entire code into one single C file causes issues in the execution of the counter code. This counter is designed to count the amount of time since the tumble parameter P_{tumble} is below the detumble limit $L_{detumbled}$. When this counter reaches a certain value, equal to $t_{detumbled}$, the mode determination algorithm will determine that the satellite has been detumbled and sets the mode to the velocity-pointing mode. When the velocity-pointing mode is initiated, this counter needs to reset. Converting the entire code into one single C file will cause that this counter code is reset after every loop, even if the control mode is set to the detumble mode. This is because the counter code will not be able to determine if the mode has been switched. By separating the C code into smaller code, the counter will be able to determine that the mode has been switched. This will result in the counter executing without errors.

Now that the coding issues have been solved, it is time to move on to the limitations of the processor. The processor will not have problems with running the basic arithmetic operations as well as the trigonometric and math functions. However, the matrix operation

that calculates the matrix exponential did not run correctly. This issue has been solved by approximating the matrix exponential function with a power series. When approximating this function some inaccuracy is introduced, but this inaccuracy did not reduce the pointing performance of the algorithm.

Recommendations

This section will provide the recommendations on a software and hardware level related to the analysis performed in this thesis.

- The parameters for the mode determination algorithm need to be able to be changed in-situ. This is because these parameters change for a changing noise of the magnetometer measurements.
- To reduce the noise of the magnetometer measurements, the distance between the magnetometer and the magnetic torquers should be as large as possible.
- Additional attitude determination sensors are needed to increase the angular estimation accuracy during eclipse such as a gyroscope or a Earth horizon sensor.
- The ADC algorithms presented in this thesis need to be tested with the flight hardware which still needs to be produced. A hardware-in-the-loop test can be used for this purpose.

Appendix A

DeFFi ADCS Requirements

A-1 List of Requirements

This section of the appendix presents the relevant ADCS requirements for the DeFFi satellite

Table A-1: DelFFi Requirements Specification List Part 1 [9]

Config. Item	Cat.	Nr.	Configuration Item	Description
SYS.1.2.1	P	01	QB50	The Cubesat shall be able to recover from tip-off rates of up to 50 deg/s within 3 days (nominal conditions)
SYS.1.2.1	P	02	QB50	The CubeSats carrying the science sensors shall have an attitude control with pointing accuracy of 10 degree and pointing knowledge of 2 degree from its initial launch altitude at 380 km down to at least 200 km.
MIS	F	01	DelFFi Mission	The mission shall facilitate Autonomous Formation Flying (AFF) payload.
MIS	F	02	DelFFi Mission	The mission shall facilitate standard QB50 payload.
MIS	C	M.1	DelFFi Mission	The QB50 payload shall only be making observations at orbit velocity direction.
MIS		M.2	DelFFi Mission	Both satellites ADCS shall provide three axis stabilization
SAT	C	07	DelFFi Satellite	All satellite systems shall be able to withstand the space environment.
SAT.2.2			ADCS	The Science Unit will be accommodated at one end of the CubeSat, on a 10 mm 10 mm face. The -Z face using the CubeSat reference frame. The vector normal to this face shall be in the spacecraft ram velocity direction. The face shall not be available for solar cells, or for any other subsystem and nothing must forward this face.
SAT.2.2	F	01	ADCS	The ADCS shall perform active attitude determination and control capabilities in three axes.
SAT.2.2			ADCS	The ADCS shall be equipped with coarse attitude sensors for coarse onboard attitude (re-)determination around all spacecraft body axes.
SAT.2.2			ADCS	The ADCS shall integrate the input from the ADCS sensors and generate actuator commands.

Table A-2: DelFFi Requirements Specification List Part 2 [9]

Config. Item	Cat.	Nr.	Configuration Item	Description
SAT.2.2			ADCS	The ADCS shall be able to provide attitude information with no worse than 1 deg accuracy around the various body axes using fine attitude sensors.
SAT.2.2			ADCS	The ADCS shall have actuators capable of providing torques around all spacecraft body axes with sufficient accuracy for data collection of all payloads.
SAT.2.2			ADCS	Attitude and rate estimates shall be determined by the ADCS subsystem in every attitude mode, with input from the AFF payload and ADCS sensors.
SAT.2.2			ADCS	The ADCS shall control the attitude actuators with sufficient frequency to allow the required pointing stability to be obtained.
SAT.2.2			ADCS	Detumbling mode is activated when the S/C rate is above 1 deg/s. The ADCS shall use rate estimates and actuators to reduce the rate below 1 deg/s.
SAT.2.2			ADCS	Coarse attitude acquisition mode is activated when the S/C rate is below 1 deg/s and above 0.5 deg/s. The ADCS shall use rate and angle estimates and actuators to reduce the rate below 0.5 deg/s, and the attitude from any attitude to within 10 deg of the target.
SAT.2.2			ADCS	Safe mode is activated by TC or by on-board autonomy in cases where other modes are considered a hazard to the spacecraft or its operability. It shall provide the safest possible orientation (e.g. a spin around the radial axis, barbecue mode or solar panels sun pointing).
SAT.2.2.1.1	C	01	Magnetometer	The magnetometers must be able to operate in a magnetic field between 19.8 μ T and 46.9 μ T.
SAT.2.2.1.1	F	01	Magnetometer	The magnetometers shall measure their position with respect to the Earth's magnetic field.
SAT.2.2.1.1	F	02	Magnetometer	The magnetometers shall provide measurement data in three axes.
SAT.2.2.1.1	I	01	Magnetometer	The magnetometer shall comply to the interfaces with the read-out electronics and/or microcontroller.
SAT.2.2.1.1	I	02	Magnetometer	The magnetometer shall comply to the interface with the electrical power supply (Vcc of 3.3V preferred).
SAT.2.2.1.1	I	03	Magnetometer	The magnetometer shall not be sensitive to interference from other satellite systems or components.
SAT.2.2.1.1	P	01	Magnetometer	The magnetometers in combination with their read-out electronics must have a minimum accuracy of 3 degrees, i.e. the magnetometers need a minimum resolution of 0.66 μ T.

Table A-3: DelFFi Requirements Specification List Part 3 [9]

Config. Item	Cat.	Nr.	Configuration Item	Description
SAT.2.2.2.1	C	01	Magnetometer	The magnetorquers shall always be operable, except when measuring the Earth's magnetic field using the magnetometers. The magnetic rise and fall time should be within 5ms to allow several measurement per second.
SAT.2.2.2.1	F	01	Magnetic Torquer	At least 3 magnetorquers are required for three-axis active attitude control.
SAT.2.2.2.1	P	01	Magnetic Torquer	Each magnetorquer must deliver a magnetic moment that is variable between at least TBD and TBD A·m ² around each axis.
SAT.2.2.2.2	F	01	Magnetic Torquer	The Reaction Wheel System shall facilitate satellite attitude control about all three principal axes.
SAT.2.2.2.2	F	02	Reaction Wheel System	The Reaction Wheel System shall be able to generate torques about all three principal axes instantaneously.
SAT.2.2.2.2	F	03	Reaction Wheel System	The Reaction Wheel System shall be able to handle commands from the central ADCS microprocessor.
SAT.2.2.2.2	F	04	Reaction Wheel System	The Reaction Wheel System shall generate reliable information about the stored angular momentum among all three principal axes.
SAT.2.2.2.2	F	05	Reaction Wheel System	The Reaction Wheel System shall not have a negative impact on the stability of the satellite.
SAT.2.2.2.2	I	01	Reaction Wheel System	The reactionwheels shall comply to the interface with the electrical power supply (Vcc of 3.3V preferred).
SAT.2.2.2.2	I	02	Reaction Wheel System	The reactionwheels shall comply to the interfaces with the read-out electronics and/or microcontroller.
SAT.2.2.2.2	I	03	Reaction Wheel System	Alignment of each individual reaction wheel shall be better than 2.5 deg with respect to each principal axis.
SAT.2.2.2.2	P	01	Reaction Wheel System	The reactionwheels shall be able to generate a minimum torque of TBD N·m (about each individual axis)
SAT.2.2.2.2	P	02	Reaction Wheel System	The reaction wheels shall be able to store angular momentum over a dynamic range of TBD N·m·s about each individual axis.

Table A-4: DelFFi Requirements Specification List Part 4 [9]

Config. Item	Cat.	Nr.	Configuration Item	Description
SAT.2.2.2.2	P	03	Reaction Wheel System	The reaction wheels shall not generate systemic disturbance torques above $1.0 \cdot 10^{-8}$ N·m.
SAT.2.2.2.2	P	04	Reaction Wheel System	The reaction wheels shall not generate instantaneous disturbance torques above $1.0 \cdot 10^{-7}$ N·m.
SAT.2.2.2.2	P	05	Reaction Wheel System	The Reaction Wheel System shall be able to control the angular momentum about each individual axis with a minimum accuracy of $2.0 \cdot 10^{-7}$ N·m·s (based on a 0.5 Hz ADCS control frequency).
SAT.2.2.2.2	P	06	Reaction Wheel System	The Reaction Wheel System shall be able to determine the angular momentum about each individual axis with a minimum accuracy of $2.0 \cdot 10^{-8}$ N·m·s (based on a 0.5 Hz ADCS control frequency).

The requirement types are abbreviated as follows

- C = Constraint: this requirements puts constraints on a configuration item
- F = Functional: this requirement specifies the functionality of a configuration item
- G = General: this requirement is not belonging to any of the other categories
- I = Interface: this requirement specifies the (type of) interface of a configuration item
- O = Operational: this requirement specifies an operational mode
- P = Performance: this requirement specifies the (minimum) performance of a configuration item

Note that some requirements are numbered with the letter 'M' followed by the number. This indicates that these requirements have not been numbered correctly in the original requirements list and will get a unique identification number for reference purposes.

A-2 ADCS Requirements of the DelFFi Mission

This section gives the ADCS requirements as found in the Requirements document of the DelFFi project [9]. For this thesis the function and the amount of ADCS modes will be changed to meet the changed configuration of the satellite and the changed schedule of the mission. These ADCS requirements can be found in table A-5.

Table A-5: ADCS Requirements DelFFi Mission ([9])

ADCS Modes	Threshold	Control objective	Sensors	Actuators	Processor
De-tumbling	$>1^\circ/\text{s}$	$1^\circ/\text{s}$	MM	MTQ	ARM9
Nadir pointing	$<1^\circ/\text{s}$	$<1^\circ/\text{s}; <10^\circ$	MM, SS	MTQ	ARM9
Sun pointing (optional)	By command	$<1^\circ/\text{s}; <10^\circ$	MM, SS	MTQ	ARM9
Thrust vector control	By command	$<1^\circ/\text{s}; <2^\circ$	MM, SS	RW, MTQ	ARM9
Safe	By command	No	MM	No	ARM9

Appendix B

Additional Equations, Derivations and Functions

B-1 Arccosine2 Function

The acos2 function will be used throughout this report. The acos2 function will extend acos as it can deliver angles ranging from 0 to 360 degrees. An extra variable H is needed, which is equal to the sign of the second component of the angle.

$$\text{acos2}(x, H) = \text{mod}(\text{real}(H \cdot \text{acos}(x)), 2\pi) \quad (\text{B-1})$$

B-2 Angular Pointing Error and Angular Estimation Error

The angular pointing and estimation error is defined as the angle between Euler eigenaxis of two quaternions. The angular error δ between the Euler eigenaxis of two quaternions q_1 and q_2 is then defined as:

$$\begin{aligned}\tilde{q} &= q_2^{-1} \times q_1 \\ \delta &= 2 \cdot \text{acos}(\tilde{q}_4)\end{aligned} \quad (\text{B-2})$$

Bibliography

- [1] Q. Chu, “Spacecraft Dynamics And Control Lecture Notes,” 2012.
- [2] R. Schoemaker, “Robust and Flexible Command and Data Handling on board the DelFFi Formation Flying Mission,” *Master of Science Thesis*, 2014.
- [3] E. Mooij and A. in ’t Veldt, “Flight Dynamics Lecture Notes (3rd year Aerospace Engineering course),” 2011.
- [4] J. Davis and V. Tech, “Mathematical Modelling of Earths Magnetic Field,” 2004.
- [5] Delfi Project Team, “DFF-TUD-TN-1141 [0.3] DelFFi ADCS - Disturbances torques,” 2014.
- [6] Delfi Project Team, “DNX-TUD-TN-0220 [4.5] ADCS - Magnetorquer Design,” 2012.
- [7] J. Guo, “DelFFi CubeSat Design Overview Report.”
- [8] W. Wu, “Design of the Reaction Wheels for the DelFFi Satellite,” *Master of Science Thesis*, 2015.
- [9] J. Guo, “DelFFi Requirements & Configuration Item List,” tech. rep., 2013.
- [10] J. Reijneveld, “Design of the Attitude Determination and Control Algorithms for the Delfi-n3Xt,” *Master of Science Thesis*, 2012.
- [11] J. Guo, “DelFFi System Overview,” 2014.
- [12] J. Verwilligen, “DFF-TUD-TN-xxxx COMMS - EM Interaction,” 2015.
- [13] D. Lay, *Linear Algebra and Its Applications*. 2006.
- [14] B. Wie, *Space Vehicle Dynamics and Control*. 2008.
- [15] J. Wertz, *Spacecraft Attitude Determination and Control*. 1990.
- [16] R. Schwarz, “Keplerian Orbit Elements to Cartesian State Vectors,” *Technical Note*.

- [17] D. Vallado, *Fundamentals of Astrodynamics and Applications*. Space Technology Library, 2001.
- [18] R. Poyck, “Design, manufacturing and characterisation of a water fed CubeSat micro-resistojet,” *Master of Science Thesis*, 2014.
- [19] T. M. A. Habib, “A comparative study of spacecraft attitude determination and estimation algorithms,” *Aerospace Science and Technology*, vol. 26, pp. 211–215, Apr. 2013.
- [20] R. Kalman, “A new approach to linear filtering and prediction problems,” *Research Institute for Advanced Study*, vol. 82, pp. 35–45, 1960.
- [21] D. Choukroun and Y. Oshman, “Novel Quaternion Kalman Filter,” *AIAA Guidance, Navigation and Control Conference*, vol. 42, no. 1, 2006.
- [22] T. Inamori, “In-orbit magnetic disturbance estimation and compensation using UKF in nano-satellite mission,” *AIAA Guidance, Navigation and Control Conference*, no. August, pp. 1–15, 2009.
- [23] Y. Yang, “Spacecraft attitude determination and control: Quaternion based method,” *Annual Reviews in Control*, vol. 36, no. 2, pp. 198–219, 2012.
- [24] M. Afonso, “Particle Filter and Extended Kalman Filter for Nonlinear Estimation: A Comparative Study,” no. March, p. 10, 2008.
- [25] J. L. Crassidis, F. L. Markley, and Y. Cheng, “Survey of Nonlinear Attitude Estimation Methods,” *Journal of Guidance, Control, and Dynamics*, vol. 30, pp. 12–28, Jan. 2007.
- [26] J. J. LaViola, “A comparison of unscented and extended Kalman filtering for estimating quaternion motion,” *Proceedings of the 2003 American Control Conference*, vol. 3, 2003.
- [27] E. Silani and M. Lovera, “Magnetic spacecraft attitude control: a survey and some new results,” *Control Engineering Practice*, vol. 13, pp. 357–371, Mar. 2005.
- [28] P. J. Camillo and F. Markley, “Orbit-averaged behaviour of magnetic control laws for momentum unloading,” *Journal of Guidance and Control*, vol. 3, no. 6, pp. 563–568, 1980.
- [29] S. Beatty, “Comparison of PD and LQR Methods for Spacecraft Attitude Control,” *Technical Note University of New Mexico (US)*, 2012.
- [30] T. Xiang, T. Meng, H. Wang, K. Han, and Z.-H. Jin, “Design and on-orbit performance of the attitude determination and control system for the ZDPS-1A pico-satellite,” *Acta Astronautica*, vol. 77, pp. 182–196, Aug. 2012.
- [31] Y. Jan and J. Chiou, “Attitude control system for ROCSAT-3 microsatellite: a conceptual design,” *Acta Astronautica*, vol. 56, pp. 439–452, Feb. 2005.
- [32] J. Reijneveld and D. Choukroun, “Attitude control system of the Delfi-n3Xt satellite,” *Progress in Flight Dynamics, Guidance, Navigation, Control, Fault Detection, and Avionics*, vol. 6, pp. 189–208, Dec. 2013.

-
- [33] J. Ziegler, N. Nichols, and N. Rochester, “Optimum Settings for Automatic Controllers,” *Journal of Dynamic Systems, Measurement and Control*, vol. 115, no. 2B, pp. 220–222, 1942.
 - [34] B. Wie and H. Weiss, “Quaternion Feedback Regulators for Spacecraft Eigenaxis Rotation,” *Journal of Guidance, Control and Dynamics*, vol. 12, no. 3, p. 375, 1989.
 - [35] QB50, “System Requirements and Recommendations by Von Karman Institute,” no. 6, 2014.
 - [36] QB50, “System Requirements and Recommendations by Von Karman Institute,” no. 7, 2015.
 - [37] E. de Angelis, “Mechanical and Magnetic actuation of the ESEO Satellite,” *PhD. Thesis*, 2013.
 - [38] T. V. Boxtel, “Thermal modelling and design of the DelfFI satellites,” *Master of Science Thesis*, 2015.
 - [39] C. Bunse and C. Peper, “Applying a Model-based Approach for Embedded System Development,” no. Seaa, 2007.
 - [40] J. Viru, “DelfFI ADCS Processor Selection,” *Technical Note*, 2014.
 - [41] M. Morales, “An Introduction to the Tiva C Series Platform of Microcontrollers,” pp. 1–6, 2013.
 - [42] E. Denney, “Certifying Auto-generated Flight Code,” tech. rep.

