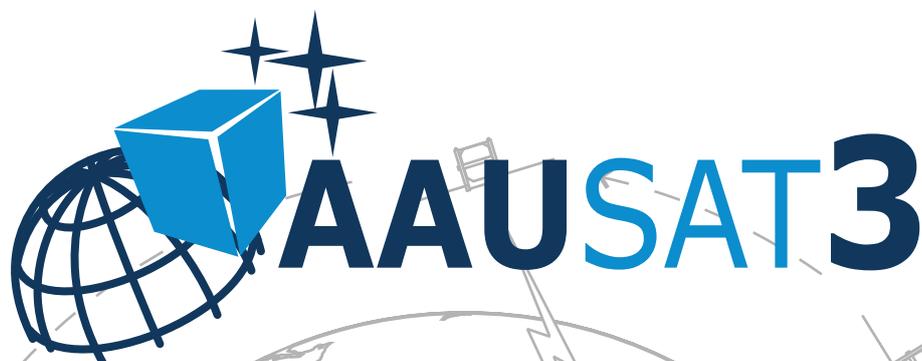


---

# Attitude Determination and Control System for

A large, light gray wireframe globe of Earth is centered on the page. It shows the continents of North and South America. Several dashed lines represent satellite orbits around the globe. Small satellite icons are placed at various points along these orbits. Some satellites have lightning bolts pointing towards the Earth's surface, indicating communication or data links. A larger satellite icon is shown on the left side, with a cone of light or signal directed towards the Earth's surface.

**Master Thesis  
June - 2010**



**AALBORG UNIVERSITY**  
Department of Electronic Systems  
Intelligent Autonomous Systems

---



**Title:**

Attitude Determination and Control System for AAUSAT3

**Theme:**

Autonomous and Reliable Systems

**Specialization:**

Intelligent Autonomous Systems

**Project period:**

Master Thesis,  
From: September 2nd, 2009  
To: June 3rd, 2010

**Project group:**

10gr1035@es.aau.dk

**Group members:**

Kasper Fuglsang Jensen  
Kasper Vinther

**Supervisors:**

Jesper A. Larsen  
Rafael Wisniewski

**Copies:** 5

**Pages:** 131

**Appendix pages:** 52

**Attachments:** 1 CD

**Finished** June 3rd, 2010

**Synopsis:**

This thesis describes the design and analysis of an Attitude Determination and Control System (ADCS) for CubeSats and has been motivated by the need for such a system on AAUSAT3, the next satellite from Aalborg University.

Evaluation of the performance of the designed ADCS is addressed with the introduction of a simulation environment for AAUSAT3, implemented in Matlab Simulink as a library. Effort has been put in keeping the library reusable and it can therefore easily be extended for future satellites.

An ADCS hardware prototype has been developed with the purpose of testing chosen sensors and actuators, which are a 3-axis magnetometer, a 3-axis gyroscope, sun sensors and magnetorquers.

A quaternion implementation of an Unscented Kalman Filter for attitude estimation on CubeSats using a low cost of the shelf sensor setup, is proposed. Emphasis has been put in making the implementation accessible to other CubeSat developers via pseudo code and results indicate, that it is possible to achieve acceptable attitude estimation, even during eclipse, without high precision sensor setups, as long as bias in the sensors are estimated.

It is finally shown, how reliable detumbling can be achieved by use of a B-dot control law that also provides 2-axis attitude stability relative to the local geomagnetic field by use of a permanent magnet. Additionally, ideas for a globally stabilizing attitude acquisition controller using model predictive control theory is given and problems associated with magnetic actuation and nonlinear model predictive control are outlined.



# Preface

This thesis documents the work on the Attitude Determination and Control System (ADCS) for the AAUSAT3 satellite. It is written as part of a Master's program within Intelligent Autonomous Systems at Aalborg University in the Department of Electronic Systems during the period from September 2009 to June 2010.

Citations are made using the IEEE style, e.g. a reference to *Spacecraft Attitude Determination and Control* by J. R. Wertz is given as [1] and references to multiple sources are written as [1, 2, 3]. Equations, figures and tables are numbered according to chapter and location, e.g. second figure in chapter three is numbered 3.2. Functions and file names are written in verbatim font as `main()`. Acronyms, reference frames, general notations and symbols used in the thesis are presented in the nomenclature after this preface.

All the presented algorithms are implemented in Matlab and simulated using Simulink. The simulation environment including design and test files can be found on the appended CD, together with datasheets and this thesis in pdf format.

The authors would like to thank Professor Rafael Wisniewski and Assistant Professor Jesper A. Larsen for supervising. The authors would furthermore like to thank the former students who have been involved in development of the Matlab satellite simulation environment for AAUSAT-II. This work has been a great help, during development of a simulation environment for testing the ADCS algorithms for AAUSAT3.

Aalborg University, June 3rd, 2010

---

Kasper Fuglsang Jensen

---

Kasper Vinther



# Nomenclature

## Acronyms

AAU	Aalborg University.
ACS	Attitude Control System.
ADS	Attitude Determination System.
ADCS	Attitude Determination and Control System.
ADCS1	ADCS subsystem 1.
ADCS2	ADCS subsystem 2.
AIS	Automatic Identification System.
AIS1	AIS subsystem 1.
AIS2	AIS subsystem 2.
CAM	CAMera subsystem.
CAN	Controller Area Network.
CoM	Center of Mass.
CoP	Center of Pressure.
DaMSA	Danish Maritime Safety Administration.
EKF	Extended Kalman Filter.
ESA	European Space Agency.
EPS	Electrical Power Supply subsystem.
FDI	Fault Detection and Isolation.
FoV	Field of View.
FP	Flight Planner.
FPGA	Field-Programmable Gate Array.
GEO	GEostationary Orbit.
GPS	Global Positioning System.
I <sup>2</sup> C	Inter-Integrated Circuit.
ICD	Interface Control Document.
IGRF	International Geomagnetic Reference Model.
JD	Julian Date.

*continued...*

---

LEO	Low Earth Orbit.
LOG	LOGging subsystem.
MCC	Mission Control Center.
MCU	MicroController Unit.
MPC	Model Predictive Control.
NORAD	North American Aerospace Defense Command.
NSO	North Sea Observer.
PCB	Printed Circuit Board.
P-POD	Poly Picosatellite Orbital Deployer.
RAAN	Right Ascension of the Ascending Node.
<i>RK4</i>	Fourth-order Runge Kutta.
S-AIS	Space based AIS.
SDP	Simplified Deep-space Perturbations.
SGP	Simplified General Perturbation.
SoI	Spheres of Influence.
SPI	Serial Peripheral Interface.
SVD	Singular Value Decomposition.
TBD	To Be Decided.
TLE	Two-Line orbital Element set.
TSI	Total Solar Irradiance.
TOMS	Total Ozone Mapping Spectrometer.
UKF	Unscented Kalman Filter.
UHF	Ultra High Frequency subsystem.
VHF	Very High Frequency subsystem.

## Reference Frames

ECI	(i)	Earth Centered Inertial reference frame.
ECEF	(e)	Earth Centered Earth Fixed reference frame.
ORF	(o)	Orbit Reference Frame.
SBRF	(s)	Satellite Body Reference Frame.
CRF	(c)	Controller Reference Frame.

## Notation

The following mathematical notations are used in this thesis.

Vectors are written with bold face and matrices are written with bold face, capital letter and underlined as

**v**

**M**

---

The elements in a vector are listed as

$$\mathbf{v} = (v_1, v_2, v_3) = [v_1 \ v_2 \ v_3]^T \quad (1)$$

The reference frame a vector belongs to is indicated as

$${}^c \mathbf{v}$$

where  $c$  denotes the CRF. Rotations are written

$${}^s_c \mathbf{q}$$

which is a rotation from the CRF to the SBRF described by a quaternion. The transpose is denoted

$$\mathbf{v}^T$$

The complex conjugate is denoted

$$\mathbf{q}^*$$

The inverse of a matrix is written as

$$\underline{\mathbf{M}}^{-1}$$

Unit vectors are typed as

$$\hat{\mathbf{u}}$$

A tilde is the small signal value and a line indicates the operating point value

$$\tilde{i}, \bar{i}$$

Time derivatives are denoted with a dot as

$$\dot{x}$$

## Symbols

Symbol	Description	Unit
$A$	Surface area	$[m^2]$
$A_{mt}$	Area enclosed by magnetorquer	$[m^2]$
$\mathbf{A}_{mt}$	Perpendicular vector to $A_{mt}$	$[m^2]$
$\mathbf{A}(\mathbf{q})$	Rotation matrix defined by a quaternion	$[-]$
$A_{sat}$	Max projected surface area of the satellite	$[m^2]$
$a_s$	Semimajor axis of an ellipse	$[m]$
$\mathbf{B}_E$	Geomagnetic field vector	$[T]$
$\mathbf{B}_{mt}$	Magnetic field vector from magnetorquer	$[T]$
$\mathbf{B}_{pm}$	Magnetic field vector from permanent magnet	$[T]$
$b_s$	Semiminor axis of an ellipse	$[m]$
$C_a$	Aerodynamic drag coefficient	$[-]$
$C_d$	B-dot controller gain	$[-]$
$C_{ra}$	Absorption coefficient	$[-]$
$C_{rs}$	Specular reflection coefficient	$[-]$
$C_{rd}$	Diffuse reflection coefficient	$[-]$
$C_{rk}$	Dimensionless constant	$[-]$
$c$	Speed of light in vacuum	$[m/s]$
$c_f$	Distance from center of ellipse to focus point	$[m]$
$d_m$	Distance between two masses	$[m]$
$E_{ecl}$	Eclipse indication (boolean value)	$[-]$
$E_{kin}$	Rotational kinetic energy of a rigid body	$[J]$
$E_{pot}$	Potential energy	$[J]$
$E_{tot}$	Total energy defined as the sum of kinetic and potential energy	$[J]$
$e_e$	Eccentricity	$[-]$
$\mathbf{F}_{as}$	Force caused by atmospheric drag	$[N]$
$\mathbf{F}_{rs}$	Force caused by solar radiation	$[N]$
$F_{solar}$	Mean integrated energy flux	$[W/m^2]$
$\mathbf{I}_{sat}$	Inertia matrix of the satellite	$[kgm^2]$
$i_o$	Inclination of the orbit plane	$[deg]$
$I_{mt}$	Current through magnetorquer	$[A]$
$J(\cdot)$	Cost function	$[-]$
$\mathbf{L}$	Angular momentum	$[Nm \cdot s]$
$M_a$	Mean anomaly	$[deg]$
$m_a$	Arbitrary mass	$[Kg]$
$m_E$	Mass of the Earth	$[Kg]$
$\mathbf{m}_{ctrl}$	Control magnetic moment vector generated by magnetorquers	$[Am^2]$
$\mathbf{m}_{pm}$	Magnetic dipole moment of the permanent magnet	$[Am^2]$
$\mathbf{m}_{mms}$	Effective magnetic dipole moment of the satellite	$[Am^2]$
$m_{sat}$	Mass of the satellite	$[Kg]$
$n_m$	Mean motion	$[rev/day]$

continued...

<b>Symbol</b>	<b>Description</b>	<b>Unit</b>
$n_w$	Number of windings for magnetorquer	[-]
$\mathbf{m}_{mt}$	Effective magnetic dipole moment of magnetorquer	[ $Am^2$ ]
$\mathbf{N}_{as}$	Aerodynamic torque	[ $Nm$ ]
$N_c$	Control horizon	[-]
$\mathbf{N}_{ctrl}$	Control torque	[ $Nm$ ]
$\mathbf{N}_{des}$	Desired control torque	[ $Nm$ ]
$\mathbf{N}_{dist}$	Disturbance torque	[ $Nm$ ]
$\mathbf{N}_{ext}$	External torque	[ $Nm$ ]
$\mathbf{N}_{gg}$	Gravity gradient torque	[ $Nm$ ]
$\mathbf{N}_{mrs}$	Magnetic residual torque	[ $Nm$ ]
$N_p$	Prediction horizon	[-]
$\mathbf{N}_{pm}$	Torque caused by permanent magnet	[ $Nm$ ]
$\mathbf{N}_{rs}$	Radiation torque	[ $Nm$ ]
$P_{rmf}$	Mean momentum flux	[ $kg/ms^2$ ]
$\underline{\mathbf{P}}_{-x_k z_k}$	State-measurement cross-covariance matrix	[-]
$\underline{\mathbf{P}}_{-z_k z_k}$	Measurement covariance matrix	[-]
$R_{ES}$	Vector from the Earth to the Sun	[ $m$ ]
$R_{Ese}$	Vector from the Earth to the satellite	[ $m$ ]
$R_{mt}$	Resistance in magnetorquer coil	[ $\Omega$ ]
$R_{sS}$	Vector from the satellite to the Sun	[ $m$ ]
$r_{scp}$	Vector from the CoM to CoP of the satellite	[ $m$ ]
$r_{si}$	Radius of Sol	[ $m$ ]
$r_e$	Mean Earth radius	[ $m$ ]
$\underline{\mathbf{S}}(\cdot)$	3 x 3 skew symmetric matrix	[-]
$T_{coil}$	Magnetorquer coil temperature	[ $^{\circ}C$ ]
$T_s$	Sampling time	[ $s$ ]
$t_{acc}$	Time period to detumbled satellite	[ $s$ ]
$t_e$	Time of epoch	[ $days$ ]
$t_p$	Time of perigee passage	[ $days$ ]
$V(\cdot)$	Objective function	[-]
$V_{des}$	Desired voltage over a magnetorquer coil	[ $V$ ]
$\mathbf{V}_{sat}$	Velocity vector of the satellite	[ $m/s$ ]
$\mathbf{v}_{sun}$	Sun vector measurement	[ $m$ ]
$\mathbf{v}_{mag}$	Magnetic field vector measurement	[ $m$ ]
$\mathbf{W}^{(c)}$	Weight in the UKF	[-]
$\mathbf{W}^{(m)}$	Weight in the UKF	[-]
$w_c$	Filter cutoff frequency	[ $rad/s$ ]
$\mathbf{Z}$	Predicted measurement (based on sigma point)	[-]
$\Upsilon$	Vernal equinox	[-]
$\underline{\Phi}$	Model Jacobian matrix (discrete)	[-]
$\boldsymbol{\omega}$	Angular velocity vector ( $\omega_1, \omega_2, \omega_3$ )	[ $rad/s$ ]
$\boldsymbol{\omega}_{init}$	Initial angular velocity	[ $rad/s$ ]
$\omega_p$	Argument of the perigee	[ $deg$ ]

*continued...*

---

<b>Symbol</b>	<b>Description</b>	<b>Unit</b>
$\omega_{req}$	Required angular velocity	$[rad/s]$
$\underline{\Omega}(\cdot)$	4 x 4 skew symmetric matrix	$[-]$
$\Omega_a$	Right ascension of the ascending node	$[deg]$
$\alpha$	Scaling parameter in UKF	$[-]$
$\beta$	Scaling parameter in UKF	$[-]$
$\delta\mathbf{q}$	Error quaternion	$[-]$
$\delta\mathbf{x}$	Error state	$[-]$
$\delta\omega$	Error angular velocity	$[rad/s]$
$\theta_{bpm}$	Angle between $\mathbf{m}_{pm}$ and $\mathbf{B}_E$	$[deg]$
$\theta_{cB}$	Angle between $\mathbf{A}_{mt}$ and $\mathbf{B}_E$	$[deg]$
$\kappa$	Scaling parameter in UKF	$[-]$
$\lambda$	Scaling parameter in UKF	$[-]$
$\mu_0$	Vacuum permeability	$[N/A^2]$
$\mu_E$	Geocentric gravitational constant	$[m^3 s^{-2}]$
$\mu_g$	Gravitational constant	$[m^3 s^{-2}]$
$\rho_a$	Atmospheric density	$[kg/m^3]$
$\nu_a$	True anomaly	$[deg]$
$\chi$	Sigma point	$[-]$

---

## Terminology

ARM7	refers to the 32 [ <i>bit</i> ] Atmel ARM microcontroller AT91SAM7A3.
Apogee	is the point where a satellite reaches the greatest distance to the Earth during its orbit.
Attitude	is the orientation of a satellite in a certain reference frame.
AVR8	refers to the 8 [ <i>bit</i> ] Atmel microcontroller AT91CAN128.
Ecliptic	plane represents Earth's orbit around the Sun. It is inclined at an approximate angle of 23 [ <i>deg</i> ] to the equator.
Eclipse	is the transit of Earth blocking all or some of the Sun's radiation as seen from a satellite.
Epoch	is a moment in time for which the position or the orbital elements of a celestial body are specified.
Geoid	is the equipotential surface which would coincide exactly with the mean ocean surface of the Earth.
Geostationary	is the state where a satellite appears stationary in the ECEF. This is approximately an altitude of 35 900 [ <i>km</i> ] above the Earth's equator and a speed matching the rotation of the Earth.
Mean anomaly	of a satellite, moving with a constant angular speed, is the angle it has traveled since perigee measured in radians.
Nadir	is the direction towards the center of the Earth.
Orbital rate	is the mean angular velocity of a satellite during an orbit around the Earth.
Latitude	is the angular distance north or south of the Earth's equator, measured in degrees.
Longitude	is the angular distance east or west from the prime meridian at Greenwich, measured in degrees.
Pitch, roll, yaw	are the angles describing the attitude of a satellite in a reference frame. Pitch is the rotation about the y-axis, roll refers to rotation about the x-axis and yaw is the rotation about the z-axis.
Perigee	is the point where a satellite reaches the smallest distance to the Earth during its orbit.
System engineering group	is a design and planning group, consisting of members of the AAUSAT3 team.
Vernal Equinox	is where the ecliptic plane crosses the equator going from south to north. Equinox happens twice a year and vernal is the spring equinox (northern hemisphere spring).

---

# Contents

<b>1</b>	<b>AAUSAT3 - Satellite Monitoring of Ships Around Greenland</b>	<b>1</b>
1.1	Satellites at Aalborg University . . . . .	1
1.2	AAUSAT3 Mission Description . . . . .	2
1.3	AAUSAT3 Subsystems and Mechanical Design . . . . .	6
1.4	Motivation for Development of an ADCS . . . . .	10
1.5	Expected Orbit for AAUSAT3 . . . . .	10
1.6	ADCS Requirements . . . . .	11
1.7	Thesis Outline . . . . .	15
<b>2</b>	<b>Satellite and Disturbance Models</b>	<b>17</b>
2.1	Orbit Description . . . . .	17
2.2	Reference Frames . . . . .	22
2.3	Quaternions and Rotations . . . . .	25
2.4	Satellite Equations of Motion . . . . .	27
2.5	Disturbance Models . . . . .	28
2.6	Summary . . . . .	33
<b>3</b>	<b>Hardware and Software Design for the ADCS</b>	<b>35</b>
3.1	Sensor Choice . . . . .	35
3.2	Actuator Choice . . . . .	41
3.3	Permanent Magnet Design . . . . .	43
3.4	Magnetorquer Design . . . . .	48
3.5	Hardware Design . . . . .	54

3.6	Software Design Considerations . . . . .	57
3.7	Summary . . . . .	60
<b>4</b>	<b>Simulink Simulation Environment</b>	<b>61</b>
4.1	Simulink Implementation . . . . .	61
4.2	Orbit Propagators . . . . .	63
4.3	Ephemeris Models . . . . .	65
4.4	Eclipse and Earth Albedo Models . . . . .	66
4.5	Magnetic Field Models . . . . .	70
4.6	Sensor Modeling . . . . .	72
4.7	Actuator Modeling . . . . .	75
4.8	Summary . . . . .	78
<b>5</b>	<b>Attitude Estimation Techniques</b>	<b>79</b>
5.1	Introduction to Attitude Estimation . . . . .	79
5.2	Wahba's Problem and the SVD-Method . . . . .	81
5.3	Extended Kalman Filter . . . . .	82
5.4	Unscented Kalman Filter . . . . .	84
5.5	Summary . . . . .	87
<b>6</b>	<b>Attitude Determination System for AAUSAT3</b>	<b>89</b>
6.1	Introduction of the Quaternion Error State . . . . .	89
6.2	SVD-method Implementation for Attitude Estimation . . . . .	90
6.3	UKF Implementations for Attitude Estimation . . . . .	92
6.4	Discussion on Attitude Determination . . . . .	101
6.5	Summary . . . . .	102
<b>7</b>	<b>Detumble Controller</b>	<b>103</b>
7.1	Aim . . . . .	103
7.2	Control Law and Stability Analysis . . . . .	104
7.3	Implementation for AAUSAT3 . . . . .	106
7.4	Simulation Evaluation . . . . .	110
7.5	Summary . . . . .	112

<b>8</b>	<b>Three Axis Attitude Stabilization with Magnetic Actuation</b>	<b>113</b>
8.1	Satellite Control with Magnetic Actuation . . . . .	113
8.2	Model Predictive Control . . . . .	115
8.3	Linear MPC with Re-Linearization . . . . .	118
8.4	Nonlinear MPC with Time Varying Magnetic Field . . . . .	124
8.5	Summary . . . . .	127
<b>9</b>	<b>Closure</b>	<b>129</b>
9.1	Conclusion . . . . .	129
9.2	Recommendations . . . . .	131
<b>A</b>	<b>Quaternions Defined and Quaternion Algebra</b>	<b>133</b>
A.1	Definitions . . . . .	133
A.2	Addition and Multiplication . . . . .	134
A.3	Quaternion Inverse . . . . .	135
A.4	Rotation of Vectors and Reference Frames . . . . .	135
A.5	Quaternion Error . . . . .	136
<b>B</b>	<b>Inertia Calculation</b>	<b>137</b>
B.1	Inertia Calculation Procedure . . . . .	137
B.2	The Mechanical Structure of AAUSAT3 . . . . .	138
B.3	Mechanical Properties of AAUSAT3 . . . . .	140
<b>C</b>	<b>Derivation of the Satellite Equations of Motion</b>	<b>141</b>
C.1	Satellite Kinematic Equations . . . . .	141
C.2	Satellite Dynamic Equations . . . . .	143
<b>D</b>	<b>Worst Case Disturbance</b>	<b>145</b>
D.1	Maximal Exposed Area of AAUSAT3 . . . . .	145
D.2	Worst Case Gravitational Torque . . . . .	147
D.3	Worst Case Aerodynamic Torque . . . . .	148
D.4	Worst Case Radiation Torque . . . . .	148
D.5	Worst Case Magnetic Residual Torque . . . . .	149
D.6	Total Worst Case Disturbance Torque . . . . .	150

<b>E</b>	<b>Verification of Simulation Environment</b>	<b>151</b>
E.1	Orbit Propagator and Ephemeris Model Test . . . . .	151
E.2	Earth Albedo Model Test . . . . .	153
E.3	Magnetic Field Model Test . . . . .	154
E.4	Equations of Motion Output and Disturbance Models Test . . . . .	157
<b>F</b>	<b>Extended Kalman Filter for Satellite Attitude Estimation</b>	<b>161</b>
F.1	An EKF Implementation for Attitude Estimation . . . . .	161
<b>G</b>	<b>Deviation of Jacobian Matrices</b>	<b>165</b>
G.1	Linearization of Kinematic Equation . . . . .	165
G.2	Linearization of Dynamic Equation . . . . .	168
G.3	Linearized Measurement Model . . . . .	169
G.4	Jacobian Matrices . . . . .	170
<b>H</b>	<b>Interface Control Document and Budgets</b>	<b>171</b>
H.1	Interface Control Document . . . . .	171
H.2	Budgets . . . . .	176
<b>I</b>	<b>ADCS Hardware Schematics</b>	<b>181</b>
<b>J</b>	<b>CD Content</b>	<b>185</b>
	<b>References</b>	<b>186</b>
	<b>Index</b>	<b>194</b>

# List of Figures

1.1	AIS signals received from ships and ground stations during the BEXUS flight. The map is generated with Google Earth. . . . .	6
1.2	Distributed structure of the AAUSAT3 satellite. Each subsystem is described in Subsection 1.3.2. . . . .	6
1.3	The software structure of AAUSAT3. Each MCU has the same layers in order for the subsystem software to be interchangeable. . . . .	7
1.4	Mechanical structure of the AAUSAT3 satellite. The CAM is placed on the opposite side of the GPS antenna. Side panels are only shown for one side. .	7
1.5	Sketch of the ADCS's mission. . . . .	11
1.6	Initial tumbling of CubeSat after ejection from P-POD. . . . .	13
1.7	A satellite tracking a ground station during a pass. $v$ is the velocity of the satellite, $r$ is the orbit height and $\dot{\theta}$ is the angular rate. . . . .	14
2.1	Geometric properties of an ellipse defined by the semimajor axis $a_s$ , the semiminor axis $b_s$ and the distance from the center of the ellipse to a focus point $c_f$ . . . . .	18
2.2	Keplerian orbit about the Earth specified using the modified classical orbital elements. . . . .	20
2.3	TLE format where d is decimal numbers, c is characters, s is symbols and e is the exponent. . . . .	20
2.4	Spheres of influence. $r_{si,1}$ and $r_{si,2}$ are the radii of Region 1 and Region 2 respectively, and $d_m$ is the distance between the two masses $m_{a,1}$ and $m_{a,2}$ . .	22
2.5	The five reference frames used in this thesis. . . . .	23
2.6	Two reference frames rotated relative to each other. . . . .	25

LIST OF FIGURES

---

2.7 Regions of which each external disturbance is dominant for AAUSAT3. The illustrated torques have been calculated in the Matlab file WCDTorque.m, which can be found on the appended CD. . . . . 29

2.8 Satellite modeled as a square plane with the area  $A_{sat}$  and an outward normal parallel to the translational velocity unit vector  $\hat{\mathbf{V}}_{sat}$ . . . . . 31

3.1 Current axes for each of the six photodiodes. The grey box illustrates the satellite and the red arrows illustrate the three largest currents forming the sun vector  $\mathbf{R}_{sS}$  (blue arrow). . . . . 40

3.2 Sketch of a satellite in orbit about the Earth, where both the Earth and the satellite is modeled as dipole magnets. The Earth's magnetic south pole corresponds to the dipoles magnetic north pole (indicated with an N). . . . 43

3.3 (a) Permanent magnet in initial position and accelerating towards the equilibrium point. (b) Permanent magnet aligned with the magnetic field lines  $\mathbf{B}_E$  (equilibrium) with the velocity  $\omega_{pm}$ . . . . . 44

3.4 Hardware blokdiagram for the ADCS with interfaces. Dashed boxes indicate components placed on seperate PCBs. . . . . 56

3.5 State transition diagram for ADCS2 . The diagram is equivalent for ADCS1 , except that ADCS1 cannot go to the POINTING state. State switches are defined by events/guards. A dashed line indicates that the switch can occur in all states and green arrows indicate autonomous state switching. . . 59

4.1 Top hierarchical content of AAUSAT3 block in Simulink. . . . . 62

4.2 Top hierarchical content of sensor and actuator models implemented in Simulink. . . . . 63

4.3 Normed difference in position between SGP4 and GPS data. Data gathered from the Ørsted satellite between 10. and 17. of February 2002 [2]. . . . . 65

4.4 Vectors, angle and distances used to calculate if a satellite is in eclipse. . . . 66

4.5 Illustrative sketch of the reflection of solar irradiance in the common FoV of the satellite and the Sun, known as Earth albedo. . . . . 68

4.6 TOMS mean reflectivity data from 2005. . . . . 69

4.7 The Earth's magnetic field modeled as a dipole magnet. N indicates the north pole of the dipole magnet (the Earth's magnetic south pole). . . . . 70

4.8 Total magnetic field strength contours generated by an IGRF10 model. The data is generated for the 1. of January 2010 in a orbit height of 630 [km]. . 71

4.9 Matlab Simulink magnetometer model. . . . . 73

4.10 Matlab Simulink sun sensor model. . . . . 74

4.11 Matlab Simulink gyroscope model. . . . . 75

4.12	Matlab Simulink permanent magnet model. . . . .	76
4.13	Matlab Simulink models of three magnetorquer coils and drivers. . . . .	76
4.14	Matlab Simulink magnetorquer coil model. . . . .	77
5.1	Block diagram of the Kalman filter. . . . .	80
6.1	Graph 1: Difference between the estimated attitude from the SVD-method and the attitude from the “truth model” (x, y and z are the Euler angles). Graph 2: Sum of elements in covariance matrix calculated according to Equation (5.7). . . . .	97
6.2	Difference between the estimated attitude and the attitude from the “truth model” (x, y and z are the Euler angles). Graph 1: UKF without bias estimation and no bias on measurements. Graph 2: UKF with bias estimation and no bias on measurements. Graph 3: UKF without bias estimation and with bias on measurements. Graph 4: UKF with bias estimation and with bias on measurements. . . . .	98
6.3	Difference between the estimated attitude and the attitude from the “truth model” (x, y and z are the Euler angles). Graph 1: UKF with bias estimation, with bias on measurements and with a small inertia difference. Graph 2: UKF with bias estimation, without $\zeta\mathbf{q}$ and with bias on measurements. . . . .	99
6.4	Difference between the estimated attitude and the attitude from the “truth model” (x, y and z are the Euler angles) for an UKF implementation with bias estimation, with bias on measurements, with sensor displacement and with only one Runge Kutta sub-step. . . . .	99
7.1	Satellite z-axis attitude over Aalborg determined by the Earth’s magnetic field. . . . .	104
7.2	Block diagram of detumble control implementation for a single axis. . . . .	107
7.3	Continuous and discrete state variable filter. . . . .	108
7.4	B-dot filter simulations with a satellite angular velocity of (10,10,10) [deg/s]. . . . .	109
7.5	B-dot simulation results with an initial angular velocity of (10,10,10) [deg/s] and a coil temperature of 30 [C°]. Graph 1 shows the angular velocity during the first four orbits. Graph 2 shows the angular velocity between orbit two and three. Graph 3 shows the angle between the $^s z$ -axis and the magnetic field vector $\mathbf{B}_E$ . . . . .	110
8.1	The local geomagnetic field vector $\mathbf{B}_E$ and the perpendicular plane that contains all the possible control torque vectors $\mathbf{N}_{ctrl}$ . . . . .	114
8.2	Basic principle of Model Predictive Control, where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the control input vector, $N_c$ is the control horizon and $N_p$ is the prediction horizon. . . . .	116

8.3	Performance of linear MPC with re-linearization using thrusters for actuation. Graph 1 shows the Euler angles representing the rotation between the ECI and the SBRF (attitude error). Graph 2 shows the angular velocity of the SBRF relative to the ECI, given in the SBRF. The bottom graph shows the control torque given in the SBRF. . . . .	122
8.4	Projection of the desired control torque $\mathbf{N}_{des}$ onto the plane perpendicular to the local geomagnetic field vector $\mathbf{B}_E$ , giving the possible control torque vector $\mathbf{N}_{ctrl}$ . . . . .	123
B.1	Position of the antenna, the PCBs and the batteries in the stack relative to the z-axis of the SBRF. . . . .	139
B.2	The three types of frame parts used. . . . .	139
B.3	Naming convention for the six sides of the AAUSAT3 satellite. The top is marked by E, while the bottom, marked by F, is concealed. The origin of the SBRF is marked by SBRF. . . . .	140
D.1	Projection of a satellite onto a plane $A_{proj}$ , which is perpendicular to an incident vector $\mathbf{R}_p$ . The area $A_{sat}$ is defined by the the projected axes of the SBRF, where the projection is performed with the matrix $\mathbf{P}_V$ . . . . .	146
E.1	Oblate Earth showing the difference between geocentric and geodetic latitude [3]. . . . .	152
E.2	Orbitron screenshots overlaid with SGP4 and ephemeris model output. Top image shows the satellite, the Sun and the Moon starting position and the bottom image shows the position after the satellite has completed one orbit. . . . .	152
E.3	3D plot of the reflectivity matrix sampled at 20 points along one orbit. . . . .	153
E.4	Reflected irradiance (gray lines) in percentage along an orbit. . . . .	154
E.5	Satellite trajectory (blue) and difference in total magnetic field strength at 12 sample points (red). The contours represent total magnetic field strength in a height of 630 [km] and varies between 18 000 to 48 000 [nT]. . . . .	155
E.6	Difference in total magnetic field intensity between different orders of IGRF. . . . .	156
E.7	Difference in total magnetic field intensity between 13. and 8. order IGRF. . . . .	156
E.8	Angular velocity in the SBRF (top), satellite attitude or orientation of the SBRF relative to the ECI (middle) and satellite trajectory (bottom). . . . .	157
E.9	Magnitude of disturbance torques acting on the satellite. From the Earth (gravity gradient), the Moon, the Sun and zonal harmonics respectively. . . . .	158
E.10	Magnitude of disturbance torques acting on the satellite. Magnetic residual (top), atmospheric drag (middle) and solar radiation with eclipse indication (bottom). . . . .	159

E.11 Total magnitude of disturbance torques. . . . . 160

I.1 Top view of ADCS PCB (not completely soldered yet). . . . . 181

I.2 Top view of ADCS PCB (not completely soldered yet) and GPS module  
(green PCB). . . . . 182

I.3 Bottom view of ADCS PCB (not completely soldered yet). . . . . 182

LIST OF FIGURES

---

# List of Tables

2.1	Common attitude parameters and their characteristics as presented in [1, 4].	26
3.1	Advantages and disadvantages of different reference sensor types, if used in a Cubesat. Arguments are from [1, 4, 5].	37
3.2	Sensor configuration examples on five different satellites. Some sensor types are unknown.	37
3.3	Properties of the HMC6343 magnetometer [6].	38
3.4	Properties of the SLCD-61N8 photodiode [7].	39
3.5	Properties of the IDG-1215 and ISZ-1215 gyroscopes [8, 9].	41
3.6	Specification of the chosen N35 sintered neodymium magnet [10]. The field strength is calculated on-axis of the permanent magnet at a distance of 5 [cm]. The given torque is calculated for a field perpendicular to the magnetic moment of the permanent magnet. The frequencies are calculated using Eq. (3.5), which gives a frequency in three dimensions [x y z].	47
3.7	Required torque from the magnetorquers to control the satellite in a region where the Earth's magnetic field either has a low or high magnetic field intensity. Notice that the detumbling torque is not included in the total torque, because the torque required for tracking is larger than the required torque for detumbling.	50
3.8	Design parameters for one magnetorquer coil without core that produces approximately 200 [nNm] at low geomagnetic field strength (18000 [nT]) perpendicular to coil area. Six of these magnetorquers are needed for the complete system. Total power consumption is for all six coils including driver circuits.	51
3.9	Comparative list of characteristics for magnetorquers with and without a core. "Innovative design" and "tested before" only relates to former satellites made at Aalborg University.	53

LIST OF TABLES

---

3.10	Design parameters for one magnetorquer coil with core that produces approximately 400 [nNm] at low geomagnetic field strength (18000 [nT]) perpendicular to coil area. Three of these magnetorquers are needed for the complete system. Total power consumption is for all three coils including driver circuits. . . . .	53
3.11	Operational states summary. Both ADCS1 and ADCS2 can be turned on at the same time, but only one of them enforces its control on the actuators. Attitude estimation can only be done by ADCS2. . . . .	59
6.1	Pseudo code for quaternion UKF implementation without bias estimation. .	94
6.2	Simulation results for one orbit. Column 1-7: Simulated case. Column 8-11: Largest absolute difference between the estimated attitude and the attitude from the “truth model”. Maximum is taken as the Euler angle (x, y or z) with the largest deviation. Err. 1 is between 0-1000 [s], Err. 2 is between 1001-2100 [s], Err. 3 is between 2101-4095 [s] and Err. 4 is between 4096-6000 [s]. Column 12: Average computation times per iteration for the implemented attitude estimators in Matlab on a single core Pentium 4 (2.8 [GHz], 1 [GB] RAM). . . . .	100
7.1	B-dot simulation results. Detumble time is the amount of orbits the B-dot controller uses to despin the satellite from the initial angular velocity to below ( $\pm 0.3, \pm 0.3, \pm 0.3$ ) [deg/s] (*in the last simulation it was set to ( $\pm 0.4, \pm 0.4, \pm 0.4$ ) [deg/s]). Power consumption is the total average of all coils including coil drivers who consume approximately 26.4 [mW]. . . . .	111
7.2	Angular velocity stability over time with the chosen B-dot control law in the AAUSAT3 simulation environment. All simulations were also performed on a mass symmetrical satellite showing similar results. x,y,z represent the three coils and p is the permanent magnet. . . . .	112
8.1	Average computation time per iteration (mean of 10 iterations) for the implemented nonlinear MPC with time varying geomagnetic field for different horizons in Matlab on a single core Pentium 4 (2.8 [GHz], 1 [GB] RAM). .	126
B.1	Mass and size of each component on AAUSAT3 included in the inertia calculation. Length, width and height are defined along the y-, x- and z-axis respectively. The estimated antenna mass is based on an aluminum prototype, with a large box, where the final version probably will be lighter. . . .	140
B.2	Mechanical properties of AAUSAT3. . . . .	140
D.1	Length of worst case disturbance torque for each environmental disturbance and the total worst case disturbance torque that will affect AAUSAT3. . . .	150
E.1	Total irradiance in 20 samples points along an orbit. . . . .	154

E.2	Difference in total magnetic field strength in 12 samples. . . . .	155
E.3	Standard deviation between the output of different model orders. . . . .	156
F.1	Pseudo code for a quaternion EKF implementation. . . . .	163
H.1	Telecommands for control of the ADCS2. All commands are equivalent for ADCS1. . . . .	173
H.2	Variables stored in EEPROM to save their value during power off/on cycles.	174
H.3	Pin assignment for stack connector. . . . .	174
H.4	Pin assignment for GPS connector. . . . .	175
H.5	Pin assignment for magnetorquer connector. . . . .	175
H.6	Pin assignment for sun sensor connector. Both SPI and I <sup>2</sup> C interface is provided. . . . .	176
H.7	Pin assignment for JTAG daisy chain connector. . . . .	176
H.8	Estimated power consumption for ADCS. Worst case is with maximum actuation on all magnetorquer coils without core. “On time” is the estimated percentage of time the state will be used, and the total is calculated from this value and average consumption. . . . .	177
H.9	Estimated power consumption for other states in ADCS. . . . .	177
H.10	Estimated ADCS prototype cost. . . . .	179



# AAUSAT3 - Satellite Monitoring of Ships Around Greenland

This thesis describes the design and analysis of an Attitude Determination and Control System (ADCS) for CubeSats and has been motivated by the need for such a system on AAUSAT3, the next satellite from Aalborg University.

AAUSAT3's mission, its subsystems and mechanical design and requirements for the ADCS are first introduced, before an outline of the thesis is given at the end of this chapter.

## 1.1 Satellites at Aalborg University

The history of developing satellites at Aalborg University (AAU) began with the assistance in building the first danish satellite, the Ørsted satellite, which was launched in February 1999 [11]. Since then, two satellites have been developed at AAU and launched into space. A short summary of their mission and the current status of the satellites are given here.

**AAU-CUBESAT:** AAU-CUBESAT was the first satellite build by students at AAU. The work on AAU-CUBESAT began in 2001 and finished when the satellite was launched into space in June 2003. The mission of the AAU-CUBESAT was to take pictures of the surface of the Earth, particular of Denmark, using an on-board camera. Connections was established with the satellite, however, problems with the battery capacity meant that only simple telemetry data was downloaded and no pictures were taken. Attempts to contact the satellite was aborted after a while even though the satellite still transmitted signals [12].

**AAUSAT-II:** AAUSAT-II, as the name indicates, is the second CubeSat build at AAU. The work began in the summer of 2003 and ended when the satellite was launched into space in April 2008. The primary mission of the AAUSAT-II satellite was to [13]:

- Establish one-way communication.
- Establish two-way communication.

- Test of an Attitude Determination and Control System (ADCS).
- Measure gamma ray burst from outer space using a gamma ray detector developed by the Danish Space Research Institute (DSRI)

Both one-way and two-way communication has been established to the satellite, however, problems with tumbling makes it difficult to upload and download large data packages from the satellite. For reasons unknown, no gamma rays have been measured. The satellite is furthermore rebooting constantly with a period of 2-5 hours. This is assumed to be caused by a Controller Area Network (CAN) receiver that is overheating. The high rate tumbling can be reduced using the on-board detumble controller, however, this must be switched on manually every time the satellite reboots. The tumbling is worsened further as the satellite seems to actively accelerate its angular velocity. This is assumed to be caused by the wires to the solar panels, which in theory could form a large magnetic residual that interacts with the magnetic field of the Earth [13].

Students from AAU also participated in the construction of the SSETI Express satellite during the same period as the construction of the AAUSAT-II satellite. The SSETI Express satellite was build by 23 groups of students from many countries and was launched into space in October 2005. The satellite suffered from a malfunction in the electric power system and the mission ended only 12.5 hour after the satellite was activated [14].

The work on the third satellite (AAUSAT3) developed at AAU began in 2007. The mission of the AAUSAT3 satellite will be described in the following.

## 1.2 AAUSAT3 Mission Description

The Danish Maritime Safety Administration (DaMSA) task is to make navigation safe in both national and international waters and furthermore they have a vision to make the Danish waters the safest in the world [15]. Currently they are close to achieving this in the Danish waters, however, the surveillance of ships in the waters around Greenland, which also fall under the DaMSA's area of responsibility, is lacking.

The DaMSA wishes to monitor the seas around Greenland using Automatic Identification System (AIS) signals. In Denmark, AIS signals are received by ground stations, however, the ground-based receivers have a limit of almost 100 [km] [16], because transmissions cannot be received beyond line of sight of the ship. This method will therefore not be able to cover the large open seas around Greenland and the many inlets along Greenland's coasts.

One way to solve the problem of receiving AIS signals in large open waters is to use Low Earth Orbit (LEO) satellites with AIS receivers. The DaMSA has already entered into an agreement with exactEarth Ltd., a company providing a space-based AIS (S-AIS) service called exactAIS, to use their services on a paid trial basis [16]. Furthermore the DaMSA proposed the main mission for the AAUSAT3 satellite, namely, to collect S-AIS data from a CubeSat and evaluate the performance.

AAUSAT3 has four other secondary missions relying on four secondary payloads: An Attitude Determination and Control System (ADCS), a camera (CAM), a Global Positioning

System (GPS) receiver and a Field-Programmable Gate Array (FPGA). The secondary missions of the AAUSAT3 satellite is to:

- Test how well the ADCS can control a LEO CubeSat.
- Take a picture of the surface of the Earth for publicity.
- Test the GPS receiver in space.
- Test the FPGA's resilience towards the harsh conditions in space.

In order to evaluate the combined success of the missions for the AAUSAT3 satellite a number of mission success criteria has been made. These will be presented after a small summary of other current initiatives in the use of the AIS.

### **1.2.1 Current Developments within Space-Based Automatic Identification System**

In February 2009 the European Commission launched a project called PASTA MARE [17] with the purpose of making an in depth assessment of the performance of S-AIS. The project is scheduled to be a two year study with the following main tasks [18]:

- Identify regions with high maritime density and develop a digital map in order to define areas where S-AIS may reach its limits.
- Identify potential interference which can affect the AIS signal.
- Determine with which confidence a vessel can be detected in a given area and with a given maritime traffic density. To determine this, the S-AIS data will be compared against terrestrial AIS data and other ship monitoring systems such as the Long Range Identification and Tracking (LRIT) and Vessel Monitoring System (VMS).
- Have the European Space Agency (ESA) to develop and manufacture an improved S-AIS sensor, which can be used to calibrate and validate future S-AIS sensors.
- Summarize conclusion and recommendations for the next step towards S-AIS service.

According to [18] the main issue of receiving AIS signals from space is due to message collision. AIS communication system utilizes a self-organizing principle i.e. ships within signal range to other ships will form self-organizing cells, where the AIS messages are synchronized. A S-AIS receiver has a larger footprint (over 3000 [km] in diameter) and will subsequently receive messages from multiple self-organizing cells, which can cause message collision [16, 18, 19]. Another major issue is saturation of the S-AIS receiver in areas with high maritime density [18].

As mentioned previously, some companies have already launched several satellites into space with AIS receivers. In June 2009 the company COM DEV launched a subsidiary called exactEarth Ltd. [16], which is scheduled to provide S-AIS data in the third quarter

of 2010 [16]. COM DEV began investigating S-AIS in 2004 and launched a demonstration satellite in April 2008, which was able to de-collide a large number of AIS messages [16]. Currently two new satellites with S-AIS receivers is scheduled to be launched in June 2010 and a third in early 2011 [16]. Another company called ORBCOMM launched 6 satellites capable of receiving AIS signals in space in 2008 and is furthermore planning to launch 18 second generation satellites capable of receiving, collecting and forwarding AIS data [20]. A third company called SpaceQuest placed a satellite with an AIS receiver in orbit in 2007 and launched more satellites with advanced AIS payloads in July 2009 [21].

The 18. of November 2009, ESA, as a part of the COLAIS project, had the Columbus module flown to the International Space Station. The Columbus module brought two advanced AIS receivers, which can de-scramble AIS messages and take Doppler shifting into account. The two receivers NORAIS and LUXAIS are developed by the Norwegian Defence Research Establishment (FFI) and the company LuxSpace respectively. The two receivers are activated alternately, two months at a time over a period of two years, in order to demonstrate S-AIS message reception, message decoding and signal sampling [19].

The most ambitious project to deploy global S-AIS ship monitoring is the GLobal AIS and Data-X International Satellite (GLADIS) constellation proposed by the Space Systems Development Department (SSDD)<sup>1</sup>. The GLADIS constellation is proposed to be an international project, where a number of countries/nations together must develop, manufacture and launch 30 nano-satellites into space, which must be able to receive AIS signals, as well as the required number of launch vehicles and satellite dispensers. SSDD suggest that five launch vehicles and five dispensers are necessary to deploy the 30 satellites into space [22].

The U.S. government will provide six of the satellites, as well as one launch vehicle and one dispenser. The U.S. government will also provide specifications/designs for satellites and dispensers. The participating countries/nations can then build or buy as well as control their own satellites. SSDD predicts three possible outcomes of the project proposal [22]:

- Project is canceled due to little interest.
- U.S. launches six satellites into space, but no other countries/nations participate.
- Multiple countries joins the project and all 30 satellites are launched into space.

The current status of the GLADIS constellation project is unknown.

As a final note it is worth mentioning that besides maritime surveillance, AIS has also proven effective as a Search And Rescue Transponder (AIS SART) by the Norwegian company JOTRON. According to [23] the AIS SART contributes to faster and more effective Search and Rescue (SAR) operations.

---

<sup>1</sup>Research and development organization of the Naval Center for Space Technology (NCST) located at the U.S. Naval Research Laboratory in Washington DC.

## 1.2.2 AAUSAT3 Mission Success Criteria

To make the AAUSAT3 mission a complete success both the expectations of the DaMSA and the AAUSAT3 design team must be satisfied. The main expectation for the DaMSA, as mentioned previously, is to evaluate the possibilities of receiving AIS messages from ships via satellite. This, first of all, requires that the AAUSAT3 satellite is able to receive AIS signals from space and if possible to obtain sufficient amounts of data to make area of detection as well as probability of detection analysis [24].

The expectation of the AAUSAT3 design team is stated as a number of mission success criteria. The number of fulfilled criteria indicates the success of the mission, i.e. if all criteria are fulfilled, then the mission is a complete success. The mission success criteria for the AAUSAT3 satellite is summarized here (made by the AAUSAT3 engineering group):

- Two-way communication with the satellite is established.
- The satellite is capable of monitoring all sea around Greenland in one pass.
- The ship traffic data around Greenland can be downloaded via UHF in subsequent passes over Aalborg.
- The satellite can receive GPS coordinates to evaluate received AIS data.
- The satellite can detumble itself and point the antennas in certain directions.
- The satellite is able to take pictures of the surface of the Earth.
- All subsystems are able to operate in space.

Figure 1.1 illustrates the expected outcome of the mission of the AAUSAT3 satellite. The figure shows positions of ships based on real data obtained from a test of AAUSAT3 during the BEXUS flight<sup>2</sup> in October 2009 [25].

---

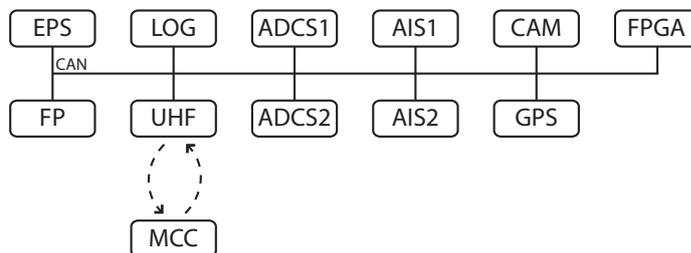
<sup>2</sup>The BEXUS program is a joint project between the Swedish Space Corporation (SSC) and the German Aerospace Center DLR, where students have the opportunity to perform experiments in a maximum altitude of 35 [km] for a duration of two to five hours [25].



*Figure 1.1: AIS signals received from ships and ground stations during the BEXUS flight. The map is generated with Google Earth.*

### 1.3 AAUSAT3 Subsystems and Mechanical Design

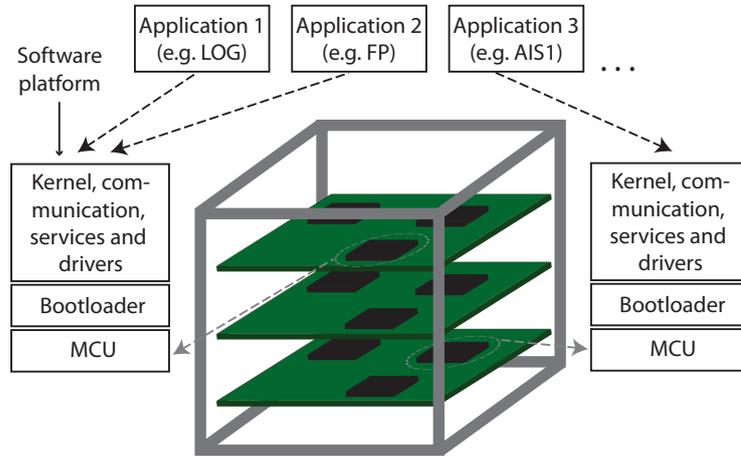
AAUSAT3 consists of multiple subsystems, which, to some extent, can operate independently of each other. Each subsystem is essentially a piece of software, which is needed to perform a specific task such as attitude determination and control. In most cases this piece of software is located on its own MicroController Unit (MCU), which is connected to the necessary hardware. All subsystems are connected through a Controller Area Network (CAN), which is illustrated in Figure 1.2.



*Figure 1.2: Distributed structure of the AAUSAT3 satellite. Each subsystem is described in Subsection 1.3.2.*

Subsystems are developed on different hardware, but they have a common software framework, which ensures a high degree of modularity, flexible development and reconfigurabil-

ity of the subsystems. The modularity, flexibility and reconfigurability of the distributed structure on AAUSAT3 is shown in Figure 1.3.

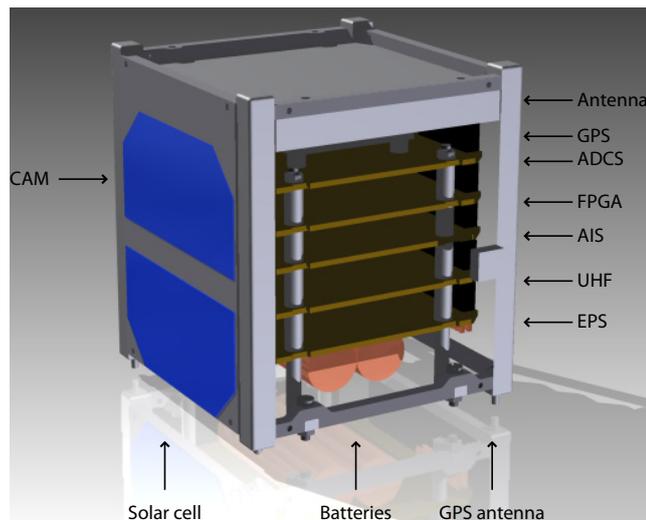


**Figure 1.3:** The software structure of AAUSAT3. Each MCU has the same layers in order for the subsystem software to be interchangeable.

The software layers are exactly the same, except the application layer (top layer) which contains the subsystem software.

### 1.3.1 Mechanical Design

AAUSAT3 is a one unit CubeSat with a length and width of 10 [cm] and a height of 11 [cm], which is the required size for a one unit CubeSat according to the CubeSat standard [26]. The expected mechanical design of AAUSAT3 is presented in Figure 1.4.



**Figure 1.4:** Mechanical structure of the AAUSAT3 satellite. The CAM is placed on the opposite side of the GPS antenna. Side panels are only shown for one side.

AAUSAT3 consists of an outer frame, which supports an inner stack consisting of six Printed Circuit Boards (PCB). AAUSAT3 also has a CAMera (CAM), a GPS antenna and two batteries. Furthermore, the frame supports the deployable antennas in the top of the satellite. The satellite will also have six side panels, which supports the solar cells, sun sensors and temperature sensors.

### **1.3.2 Subsystem Descriptions**

In order for the reader to be familiarize with the different subsystems on AAUSAT3, a small summery of their functionality is given in the following.

#### **Platform with EPS, LOG and FP**

The Electrical Power Supply subsystem (EPS) converts power generated by the solar cells into at suitable voltage used to charge the batteries and furthermore it distributes and monitors power to all the subsystems. The Flight Planner (FP) is essentially a schedule containing planned actions for a given time e.g. when to start recording of AIS data. The FP also maintains the time on the satellite. The LOGging subsystem (LOG) stores log-messages from all subsystems. The messages can then be downloaded from the satellite on request.

#### **UHF**

The Ultra High Frequency (UHF) transceiver subsystem ensures that the satellite is able to receive commands and software from the ground station and that the satellite is able to send data back.

#### **Ground Station/MCC**

The ground station/Mission Control Center (MCC) is not a subsystem on the satellite. The MCC is the link between the satellite and the ground team and is necessary in order to upload commands and software to the satellite as well as downloading data from the satellite.

#### **AIS1 and AIS2**

The hardware based Automatic Identification System 1 (AIS1) and the software based AIS2 subsystems receives AIS messages from ships and stores them on the satellite. The AIS subsystems are able to pack AIS messages received for the same ship in order to lower the amount of data. The AIS data can be downloaded from the satellite on request.

## **CAM**

The CAM takes and stores pictures which can be downloaded from the satellite on request. The CAM has no scientific purpose, but has been included with the intention of creating publicity for AAU and AAUSAT3.

## **GPS**

The GPS is used to determine the position of the satellite in space. The position can be used in the evaluation of AIS data where the position of the satellite for the given data is valuable e.g. to evaluate the effective distance of the AIS monitoring. Furthermore the position is valuable for use in the ADCS e.g. to evaluate the precision of the estimated satellite position.

## **FPGA**

The FPGA do not have any functionality. The FPGA is included to test how resistant it is against the harsh conditions in space.

## **ADCS1 and ADCS2**

The ADCS contains the following aspects:

- **Attitude Estimation:** The ADCS must be able to point the satellite in a given direction, which requires that the ADCS is able to determine the attitude of the satellite.
- **Attitude Control:** As the given attitude of the satellite may not be the wanted attitude, the satellite must be able to control it.

The ADCS1 can be thought of as reduced ADCS, which only performs simple control tasks such as detumbling of the satellite. The ADCS2 subsystem is a fully featured ADCS, which can perform simple control tasks such as detumbling, but also more advanced control task such as pointing control. The purpose of having two ADCS subsystems capable of detumbling the satellite is to ensure robustness and reliability of the ADCS. Hence if one subsystem fails, the other is able to take over.

The ADCS will be discussed more thoroughly throughout this thesis. The ADCS can be split into an Attitude Determination System (ADS) and an Attitude Control System (ACS). The ADS uses sensors and estimation algorithms to determine the attitude of the satellite, while the ACS uses actuators and control algorithms to control the attitude of the satellite.

## 1.4 Motivation for Development of an ADCS

A distinction should be made between orbit determination and control and attitude determination and control. This work deals with the latter of the two. Attitude refers to the orientation/rotation of the satellite in space. It is assumed that the parameters describing the orbit, explained in Section 2.1, are determined beforehand and no attempt will be made to e.g. change the altitude of the satellite, which is orbit control. This is a valid assumption because orbit parameters for all spacecrafts are determined by the North American Aerospace Defence Command (NORAD) and are published in the Two Line Element (TLE) format [27], also described in Section 2.1. This means that the position of the satellites can be estimated directly, at a given time, without measurements from satellite on-board sensors. However, this requires frequent updates of the orbit parameters in order to maintain accuracy over time (see discussion in Section 4.2).

It is desired to be able to control the attitude of AAUSAT3 due to the fact that several applications on board the satellite will benefit from attitude control and the ability to point the satellite with accuracy. The reasons for development of an ADCS are:

- The direction of highest gain for the AIS antenna should be pointed downwards, or in a certain angle relative to the Earth's surface, in order to maximize receipt of signals from ships. Knowing the orientation of antennas also helps in evaluating performance of the AIS receivers.
- The direction of highest gain for the GPS antenna should be pointed up towards the GPS satellites, orbiting in a height of approximately 20 000 [km] [28].
- A camera payload needs pointing accuracy and slow angular velocity to take good pictures.
- The communication between the satellite and a ground station can be optimized, if the satellite antenna tracks the ground station.
- Power generated from the solar panels can be optimized, by maximizing the surface area of the solar panels pointing towards the Sun.

Beyond these reasons, the development of an ADCS also meets the learning requirements stated in the curriculum [29] and will hopefully give results usable for future space missions with Cubesats.

## 1.5 Expected Orbit for AAUSAT3

It is expected that AAUSAT3 will be launched into space by PSLV-C20 (an Indian rocket) in the first quarter of 2011. The rocket will place AAUSAT3 in a sun-synchronous polar orbit, i.e. the AAUSAT3 satellite will have an inclination close to 90 [deg].

Furthermore it is expected that the orbit altitude will be close to 620 [km] and, in general, that the orbital parameters are approximately the same as the AAUSAT-II satellite. Hence TLEs from AAUSAT-II are used for simulation and testing of the ADCS algorithms designed for AAUSAT3.

## 1.6 ADCS Requirements

The motivation described in Section 1.4 is translated into functional requirements and performance requirements for the ADCS. These requirements are used in the development and form the framework for test of algorithms and hardware, making it possible to conclude how well the attitude determination and control problem is solved.

### 1.6.1 ADCS Mission

Figure 1.5 presents the mission of the ADCS for AAUSAT3.

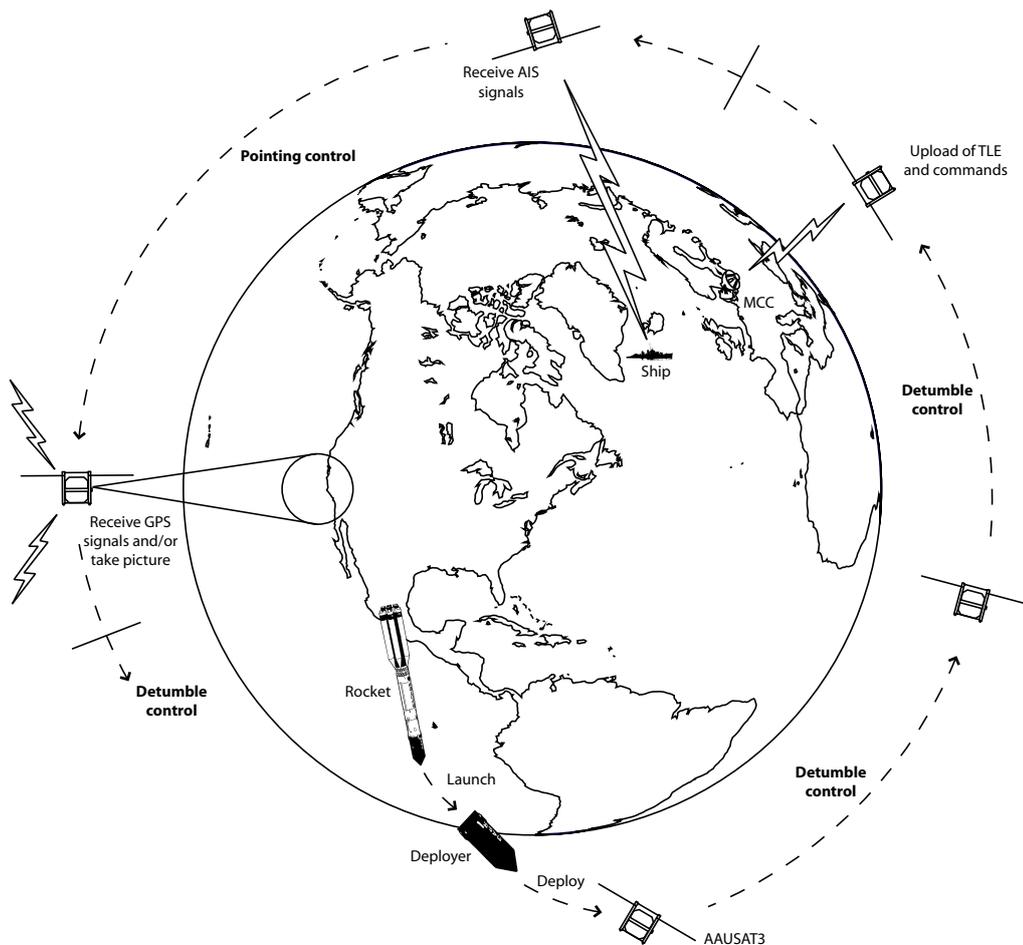


Figure 1.5: Sketch of the ADCS's mission.

As shown in Figure 1.5, the satellite is launched into space with a rocket. The satellite is stored in a Poly-PicoSatellite Orbital Deployer (P-POD), which will deploy the satellite in an altitude of approximately 620 [km]. The deployment sequence can cause the satellite to tumble and the ADCS will therefore initiate a detumble sequence, which will stabilize the satellite. When the satellite is detumbled, a TLE is uploaded from the ground station and the time is synchronized. The ADCS can then use pointing control to receive AIS signals, take pictures of the Earth and receive GPS signals. When pointing control is no longer required the ADCS keeps the satellite detumbled.

## 1.6.2 Functional Requirements

The motivation and the mission description leads to the following functional requirements for the ADCS, with §6 as parent requirement (the numbering comes from the online documentation shared between the members of the AAUSAT3 team [30]).

§6 *The satellite must be able to detumble itself and point antennas and camera in certain directions.*

This defines two tasks to be performed by the ADCS: Detumbling and pointing.

§6.1 *The ADCS hardware must not rely on moving parts.*

This has been determined by the system engineering group and will improve robustness and reliability of the satellite.

§6.2 *The ADCS hardware should be gathered on one PCB (CubeSat size), with the possibility of putting sensors and actuators on the side-panels of the satellite.*

Having a dedicated PCB for ADCS eases the development of the subsystem, since the interface to other subsystems is reduced to a minimum.

§6.3 *The ADCS hardware must comply with the Interface Control Document (ICD), see Appendix H.*

This ensures that the developed hardware is compatible with the rest of the satellite.

§6.4 *The ADCS detumble hardware must be fully redundant for improved reliability.*

It is of high priority that the satellite is capable of detumbling itself, since it is experienced from AAUSAT-II that tumbling increases, when the satellite is not actively trying to detumble itself.

§6.5 *The ADCS must detumble the satellite.*

Referred to as simple control. This gives a more stable communication link, with less high frequency fading of the radio signal (at the time of writing AAUSAT-II tumbles with an approximate velocity of 2.5 [Hz], causing communication problems [13]).

§6.6 *The ADCS must be able to point antennas and camera towards a target.*

Referred to as advanced control. This will potentially optimize data transfer, GPS module operation and render it possible to control what the camera takes pictures of.

§6.7 *The ADCS must be able to detect and accommodate sensor and actuator faults.*  
To further improve robustness and reliability.

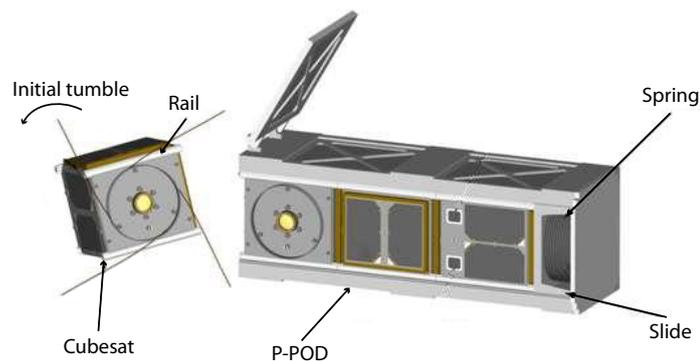
§6.8 *The satellite shall be able to carry out autonomous detumbling.*  
Detumbling shall start 30 to 45 minutes after activation of AAUSAT3.

As mentioned in Section 1.3, ADCS can be split into two subsystems: The ADS part responsible for estimation of the attitude and the ACS part responsible for obtaining a desired reference attitude, given the estimated attitude. These subjects are the main topics of this thesis. Fulfillment of requirement §6.7 is handled by group 10gr832 and is considered out of the scope of this thesis, however, general ideas of how to accommodate faults and design of redundant hardware are presented.

### 1.6.3 Performance Requirements

Performance requirements are important when evaluating the usefulness of different controllers and algorithms and to be able to conclude if the designed systems are performing as required in order to optimize the chances of mission success.

The CubeSat will be deployed from a Poly-PicoSatellite Orbital Deployer (P-POD), with a spring and sliding rail concept, as shown in Figure 1.6.

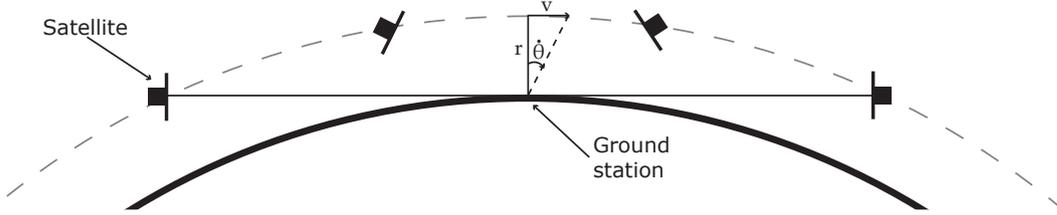


**Figure 1.6:** Initial tumbling of CubeSat after ejection from P-POD.

An initial tumble is to be expected due to the P-POD separation mechanism. In [31] this value was set to  $10 [deg/s]$  for the SSETI satellite. This value will be used, even though the SSETI satellite was not a CubeSat. AAUSAT-II used  $5.73 [deg/s]$  as initial tumble [2]. It has been decided that the ADCS must detumble the satellite from the initial  $10 [deg/s]$  to below  $\pm 0.30 [deg/s]$  in the satellite body fixed axes within three orbits (total detumbled state is approximately  $0.12 [deg/s]$ ). The satellite must additionally be able to recover from spin rates of up to  $720 [deg/s]$  in less than two weeks.

It is decided that the ADCS should be able to track a reference attitude within  $\pm 10 [deg]$  in Euler angles outside eclipse and within  $\pm 15 [deg]$  inside eclipse (it is assumed that the ADS will perform better outside eclipse). Reference attitudes could be ground station tracking or slower maneuvers like constant nadir pointing.

To be able to point the satellite e.g. towards a ground station during a pass, sets requirements on the minimum angular velocity the ADCS can rotate the satellite with. Figure 1.7 illustrates the required rotation of the satellite during a pass.



**Figure 1.7:** A satellite tracking a ground station during a pass.  $v$  is the velocity of the satellite,  $r$  is the orbit height and  $\dot{\theta}$  is the angular rate.

The highest angular rate  $\dot{\theta}$  during the pass will be just above the ground station, where the distance is smallest. Equation 1.1 gives the angular rate in this point.

$$\dot{\theta} = \tan^{-1}\left(\frac{v}{r}\right) \quad (1.1)$$

The lowest expected orbit height  $r$  is set to 600 [km] for AAUSAT3 and the velocity is calculated to be 7.56 [km/s], giving an angular rate of approximately 0.72 [deg/s], which is equivalent to the value stated in [28]. The ADCS should therefore be able to accelerate the satellite from 0 to 0.72 [deg/s] within 385 [s] (time from start of pass to halfway), if assuming that the angular velocity is 0 [deg/s] at the start of the pass and that the start attitude is as shown in Figure 1.7.

The above is gathered in four requirements, which specify the desired performance of the ADCS and should be seen as design guidelines.

§6.5.1 *The ADCS must be able to detumble the satellite from 10 [deg/s] to below  $\pm 0.30$  [deg/s] given in the satellite body fixed axes within 3 orbits.*

§6.5.2 *The ADCS must be able to detumble the satellite from up to 720 [deg/s] to below  $\pm 0.40$  [deg/s] given in the satellite body fixed axes within 2 weeks.*

§6.6.1 *The ADCS must track a reference attitude within  $\pm 10$  [deg] in Euler angels outside eclipse and  $\pm 15$  [deg] inside eclipse, with a 95.4% confidence interval. This requirement only applies for nadir pointing and ground station tracking.*

§6.6.2 *The ADCS must be able to accelerate the satellite from 0 to 0.72 [deg/s] within 385 [s], in order to track a ground station.*

Requirement §6.6.1 holds for both ADS and ACS together. This means that individually they must perform better in order to meet this overall requirement (e.g.  $\pm 5$  [deg]). 95.4% confidence interval is the same as  $\pm 2\sigma$  ( $\sigma$  is standard deviation).

### 1.6.4 Test Specification

A suitable simulation environment should be used to test if the requirements are met. This simulation environment must include all relevant dynamics and disturbances. It must also include simulation of sensors and actuators and it should use an orbit similar to the orbit of AAUSAT-II.

A hardware prototype should also be developed and tested in space like conditions according to the requirements provided by the launch provider. These requirements are, however, not available at the time of writing. The software should then be tested for potential numerical problems on this hardware.

All testing will not be performed within the time limit of this project, but test procedures based on the requirements should be made and posted on the common communication forum [30] shared by the members of the AAUSAT3 team.

## 1.7 Thesis Outline

The rest of this thesis is organized as follows:

**Chapter 2** introduces the Keplerian orbit elements identifying the orbit of the satellite about the Earth, the satellite equations of motion and disturbance models. Different reference frames are also defined along with a definition of rotations and quaternions, which are the used attitude parameterization in this thesis.

**Chapter 3** identifies suitable sensors and actuators for AAUSAT3 and presents the design of the chosen actuators. A prototype PCB for the ADCS is also described together with ideas for ADCS software states.

**Chapter 4** describes the Matlab Simulink simulation environment used to evaluate the performance of the ADCS. The models in this environment are based on the results obtained in Chapter 2 and 3 and extends the work done by former students at Aalborg University, especially members of the AAUSAT-II team.

**Chapter 5** gives an introduction to attitude estimation by discussing different techniques within the field of both deterministic point by point methods and recursive stochastic filters. From these a more thorough description of the SVD-method solving Wahba's problem, the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) are given, with focus on describing the equations and procedures involved in the algorithms.

**Chapter 6** applies UKF theory for satellite attitude estimation using quaternions. This is supported by simulations showing the performance of the ADS on AAUSAT3 under the influence of modelling errors and sensor noise and biases.

**Chapter 7** presents the design of a detumble controller, with the purpose of despinning the satellite and keeping the angular velocity low at all times. The focus of this chapter is stability and robustness, since detumbling is of paramount importance for the mission.

**Chapter 8** discusses three axis satellite attitude stabilization with magnetic actuation. It is investigated how Model Predictive Control (MPC), both in its linear and nonlinear form, can be used for global attitude acquisition.

**Chapter 9** summarizes the thesis with a conclusion and gives recommendations for future work.

# Chapter 2

## Satellite and Disturbance Models

In order to determine and control the attitude of a satellite it is first necessary to model the orbit and the satellite equations of motion, but also to identify and model relevant disturbances that influence the attitude. These models can then be used for design of algorithms and validation of these through simulation.

This chapter starts by giving a description of the different Keplerian orbit elements used to identify the orbit of the satellite about the Earth. Section 2.2 then defines the different reference frames used for defining attitudes and vectors and this is followed by Section 2.3, where Euler parameters (quaternions) are chosen for attitude representation among the many possible parameterizations. The satellite kinematic and dynamic equations of motion are then presented in Section 2.4 and finally Section 2.5 presents the identified disturbance models.

### 2.1 Orbit Description

A system consisting of a satellite orbiting the Earth can be considered as two point masses, which are influenced by their mutual gravitational attraction (validated in the end of this section). It is possible to describe this system using Kepler's three empirical laws of planetary motion, or by using Newton's laws of motion and his law of gravity. These laws apply for all orbital elements, but this section will consider the case of a satellite in orbit about the Earth unless otherwise specified.

#### 2.1.1 Keplerian Orbits

A Keplerian orbit can be described using six orbital elements, which defines the size, shape and orientation of the orbit. The elements mentioned here are referred to as the six classical orbital elements [3] and will be explained in the following.

- Semimajor axis  $a_s$ .

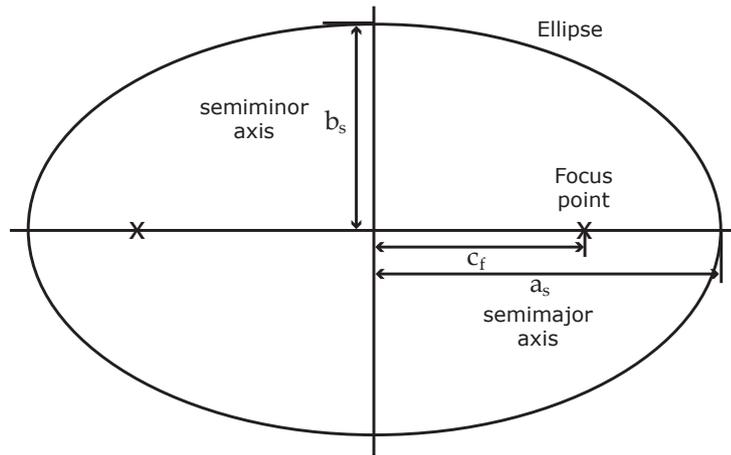
- Eccentricity  $e_e$ .
- Time of perigee passage  $t_p$ .
- Right Ascension of the Ascending Node (RAAN)  $\Omega_a$ .
- Inclination of the orbit plane  $i_o$ .
- Argument of the perigee  $\omega_p$ .

The semimajor axis  $a_s$  and the eccentricity  $e_e$  defines the size and shape of the orbit respectively. The eccentricity is given as

$$e_e = \frac{c_f}{a_s} \quad (2.1)$$

where:

$c_f$  is the distance from the center of an ellipse to a focus point (see Figure 2.1).



**Figure 2.1:** Geometric properties of an ellipse defined by the semimajor axis  $a_s$ , the semiminor axis  $b_s$  and the distance from the center of the ellipse to a focus point  $c_f$ .

From Eq. (2.1) it is easily shown that if  $e_e = 0$  then the shape of the orbit is a circle with a radius of  $a_s$ . If  $0 < e_e < 1$  then the shape of the orbit is an ellipse, where  $a_s$  is the half length of the major axis of the ellipse. For  $e_e > 1$  the shape of the orbit is hyperbolic [3].

The time of perigee passage  $t_p$  is used to pinpoint the position of the satellite at a given time. However,  $t_p$  is often replaced by the mean anomaly  $M_a$ , in a set of modified classical orbital elements. The mean anomaly is the angle traveled by the satellite since perigee (the point where the satellite has the smallest distance to the Earth) assuming that the orbit is circular. The mean anomaly is calculated as follows [1]

$$M_a = 360 \cdot \left( \frac{\Delta t}{P_o} \right) [deg] \quad (2.2)$$

where:

$\Delta t$  is the time traveled since perigee to a time, where the position is known, called time of epoch  $t_e$ .

$P_o$  is the orbital period.

The time of epoch  $t_e$  is measured in the Julian Date (JD) format. The JD is measured in the number of days since noon (12:00 Universal Time (UT)<sup>1</sup>) on January 1, 4713 BC. This year was suggested as starting point by Joseph Scaliger and is the start of the current Julian period of 7980 [*years*] [1]. As an example the date January 15, 2010 at 12:00 UT corresponds to JD= 2 455 212.0 [*days*].

The mean anomaly is of no physical interest, but it can be used to calculate the true anomaly, which is almost similar to the mean anomaly except that the true anomaly applies for elliptic orbits [1].

The RAAN  $\Omega_a$  is the angle measured from the vernal equinox counter clockwise to the ascending node (the point where the satellite crosses the Earth's equatorial plane from south to north).

The inclination of the orbit plane  $i_o$  is the angle measured between the Earth's equatorial plane and the orbital plane. Orbits with  $i_o < 90$  [*deg*] are called prograde orbits while orbits with  $i_o > 90$  [*deg*] are called retrograde orbits. Orbits with  $i_o = 90$  [*deg*] and  $i_o = 0$  [*deg*] are called polar and equatorial orbits respectively.

The argument of perigee  $\omega_p$  is the angle measured between perigee and the ascending node in the direction parallel to the movement of the satellite.

An example of a Keplerian orbit about the Earth is given in Figure 2.2.

### 2.1.2 Two Line Elements

The North American Aerospace Defense Command (NORAD) is a bi-national organization between the United States of America and Canada. One of the functions maintained by NORAD is to monitor man-made objects in space [27]. The data containing information about the satellite's orbit is published in the Two Line Element (TLE) format, which also includes all the six modified orbital elements.

The TLE format from NORAD consists of two lines, each with space for 69 characters. The format is illustrated in Figure 2.3.

The classification field can either be U (unclassified) or C (classified) [32], which means that public published TLE's can only be U.

The ephemeris field is only used for internal analysis. The field is published with the value zero for all TLE's using the Simplified General Perturbation/Simplified Deep-space Perturbation (SGP4/SDP4) orbit propagator model, where the number 4 indicates that it is model order 4. Hence it is up to the user to select the appropriate model. The SGP4 orbit propagator model is used for near-Earth orbits, while SDP4 is used for deep-space orbits.

<sup>1</sup>UT is a modern continuation of the Greenwich Mean Time (GMT), which is the same everywhere on the Earth.



period from launch to the satellite reaches the ascending node [32]. As an example one of the TLE's for AAUSAT-II is [34].

```
1 32788U 08021F 09279.82913737 .00000416 00000-0 59418-4 0 4779
2 32788 97.9326 342.7215 0016322 34.9001 325.3249 14.81792083 77983
```

The TLE is used for orbit propagation models such as the SGP, SGP4 and SGP8 for near-Earth orbits and SDP4 and SDP8 for deep-space orbits. The orbit propagators will be discussed further in Section 4.2.

### 2.1.3 Orbital Perturbation

The ideal Keplerian orbit model assumes that the Earth has a spherically symmetric mass distribution and that the Earth's orbital plane is fixed in space. However these assumptions will cause the ideal Keplerian orbit to have small deviations from the true orbit, which is called orbital perturbations [3].

As the Earth is oblate the equatorial bulge will cause the orbital plane to rotate in the direction of the angular momentum i.e. a satellite in a prograde orbit has an angular momentum pointed towards west. The equatorial bulge will in this case cause the orbital plane to rotate westwards (called regression of nodes) [1]. The effects caused by oblateness perturbations are considerable, but is accounted for in the SGP orbit propagators [35].

For satellites in a higher orbit the effects of the oblateness of the Earth are decreasing, but the effects of perturbations from the Sun's and the Moon's gravitational force are then increasing and hence it may not be appropriate to consider the system (satellite in orbit about the Earth) as a two-body problem.

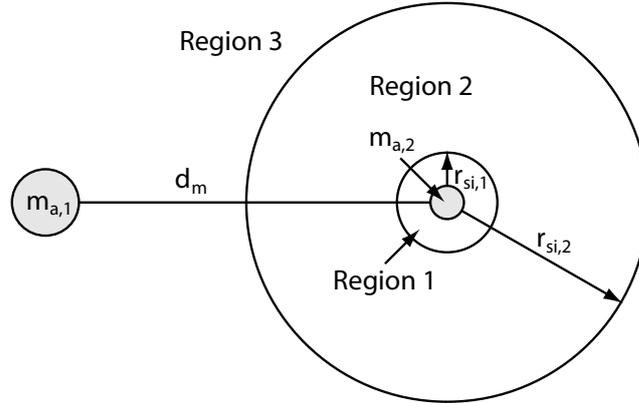
To evaluate whether it is reasonable to consider the above mentioned system as a two-body problem, it is convenient to divide the space into regions called Spheres of Influence (SoI). Figure 2.4 illustrates two masses  $m_{a,1}$  and  $m_{a,2}$ , where  $m_{a,1} \gg m_{a,2}$  with the distance  $d_m$  between them and the space around the masses divided into three regions defined by the radius of  $r_{si,1}$  and  $r_{si,2}$ .

The radii of Region 1 and 2 is defined using a parameter  $\epsilon$  such that the perturbation force due to  $m_{a,1}$  is much less than the perturbation force due to  $m_{a,2}$  times  $\epsilon$ . For  $\epsilon = 0.01$  the perturbation force due to  $m_{a,1}$  is at least 100 times greater than the force due to  $m_{a,2}$  in Region 3 [1].

If the satellite is present within Region 1, the orbit is approximately Keplerian about  $m_{a,2}$ . If the satellite is present within Region 3, the orbit is approximately Keplerian about  $m_{a,1}$ . If the satellite is present within Region 2 both the gravitational forces of  $m_{a,1}$  and  $m_{a,2}$  will be significant and hence the orbit can not be assumed to be Keplerian [1].

Using the SoI between the Earth and the Moon, as well as the SoI between the Sun and the Earth, with  $\epsilon = 0.01$ , it is valid to assume that a satellite in LEO is Keplerian [1].

Another important orbital perturbation, which applies for LEO satellites, is the atmospheric drag. The effect of the atmospheric drag is difficult to predict. The air density changes when the Sun heats up the atmosphere on the day side of the Earth and when it cools down on the



**Figure 2.4:** Spheres of influence.  $r_{si,1}$  and  $r_{si,2}$  are the radii of Region 1 and Region 2 respectively, and  $d_m$  is the distance between the two masses  $m_{a,1}$  and  $m_{a,2}$ .

night side. The atmospheric drag will also increase near perigee where the satellite is closer to the Earth. Furthermore the atmospheric drag causes the satellite to lose altitude [3]. The SGP orbit propagation models take the air density into account, however, not changes in air density since last TLE.

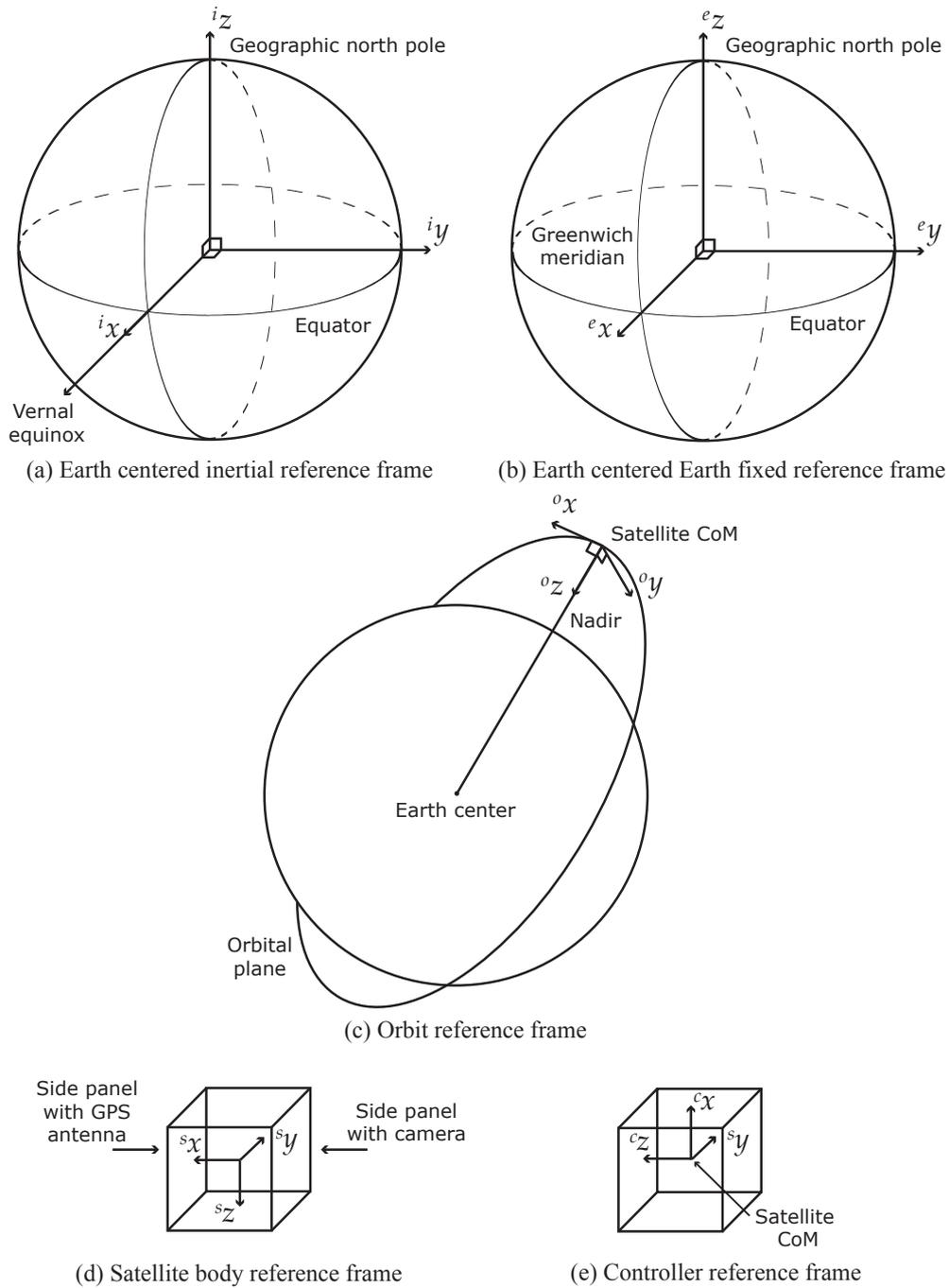
## 2.2 Reference Frames

Reference frames used throughout this thesis are defined in this section. They are introduced in order to describe rotation of rigid bodies and to define vectors in  $\mathbb{R}^3$ . A Reference frame is a descriptive term for a right handed three-dimensional Cartesian coordinate system, described by three mutually perpendicular unit vectors. Multiple reference frames carefully placed eases calculations. Figure 2.5 illustrates the used reference frames.

### 2.2.1 Earth Centered Inertial Reference Frame (ECI)

The motion of the satellite (rigid body) is best described in an inertial reference frame (Newtonian reference frame). Placing this frame in the center of the Earth with the  $x$ -axis going through the point where the vernal equinox and the equatorial plane crosses and the  $z$ -axis through the geographic north pole creates a non-accelerating point of view. An inertial reference frame is identified as a frame where no fictitious forces are present (e.g. centrifugal force) putting the physical laws in their simplest form [36]. So inertial frame is a relative concept. The  $y$ -axis is the cross product of the  $x$ - and  $z$ -axis, thus creating a right handed Cartesian coordinate system and the ECI is depicted in Figure 2.5(a).

The ECI is not a perfect inertial reference frame because of the Earth's orbital motion around the Sun and its rotational motion about itself, both resulting in centripetal accelerations. However these accelerations can be neglected [36].



**Figure 2.5:** The five reference frames used in this thesis.

The Sun and the Moon exerts a gravitational force on the Earth's equatorial bulge causing a torque, which results in a slow rotation of the Earth's spin axis about an imaginary line perpendicular to the ecliptic plane. A complete cycle takes approximately 25 800 [years], which is a rate of 0.014 [deg/year] and the angular radius of the circle is 23.5 [deg]. This

is called precession of the equinoxes and a date is often attached to the vernal equinox to accurately specify its position [1].

The Earth's true spin axis wobbles due to the changing inertial orientation of the Moon's orbit. This wobble (nutation) has an amplitude of 0.0026 [*deg*] and a period of 19 [years] [1] and is considered negligible.

## 2.2.2 Earth Centered Earth Fixed Reference Frame (ECEF)

A frame that is fixed relative to the surface of the Earth is convenient for defining e.g. the magnetic field vectors and position of ground stations. The ECEF is centered in the Earth's center with the x-axis going through the point where the Greenwich meridian crosses the equatorial plane (0 [*deg*] longitude, 0 [*deg*] latitude) and the z-axis going through the geographic north pole. The y-axis is the cross product of the x- and z-axis, again creating a right handed Cartesian coordinate system, see Figure 2.5(b).

## 2.2.3 Orbit Reference Frame (ORF)

The ORF maintains its orientation relative to the Earth and follows the satellite in its orbit. The orientation of the satellite with respect to this system of coordinates is also known as roll, pitch and yaw. The z-axis is always nadir pointing and the x-axis is parallel to the orbit plane and perpendicular to the z-axis. In a circular orbit the x-axis has the same direction as the velocity vector. Again the y-axis is the cross product of the x- and z-axis, see Figure 2.5(c).

Roll is defined as the right handed rotation about the x-axis, pitch is the rotation about the y-axis and yaw is the rotation about the z-axis.

## 2.2.4 Controller Reference Frame (CRF)

The CRF is located in the Center of Mass (CoM) of the satellite with the x-axis defined as the minor axis of inertia and the z-axis defined as the major axis of inertia. The y-axis is the intermediate axis of inertia and also the cross product between the two. This is a body fixed frame and it is used for calculations involving the satellites dynamics, as all products of inertia are eliminated, which is computationally convenient. The axes are also known as the principal axes [3]. This means that the mass distribution of the satellite must be determined in order to derive the inertia matrix and its eigenvalues. The eigenvectors of the largest eigenvalue corresponds to the major axis of inertia and the eigenvector of the smallest eigenvalue corresponds to the minor axis of inertia [3]. The frame is depicted in Figure 2.5(e).

### 2.2.5 Satellite Body Reference Frame (SBRF)

This frame is used to define orientation of ADCS hardware and attitude measurements. The x-, y- and z-axes are chosen to be parallel to the satellite frame structure, with the y-axis pointing in the opposite direction of the camera line of sight and the z-axis pointing in the direction of the batteries on the satellite, see Figure 2.5(d).

To sum up the ECI and the CRF will be used for deriving the equations of motion for the satellite. The ORF is used for control purposes and to describe roll, pitch and yaw of the satellite. Lastly the ECEF is used in the calculation of the geomagnetic field and the SBRF is convenient for describing position and orientation of objects like sensors and actuators.

## 2.3 Quaternions and Rotations

After having defined reference frames for describing position and orientation of objects, it is obvious to discuss rotation of such reference frames, thus making it possible to express the orientation of the objects relative to different viewpoints. This could e.g. be the orientation of the CRF, which is a satellite body fixed frame, relative to the ECI, giving the attitude of the satellite. The rotation example is illustrated in Figure 2.6.

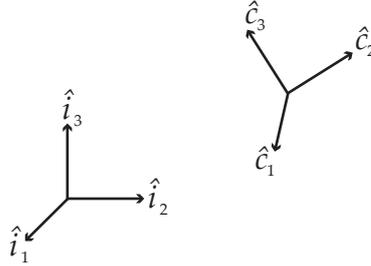


Figure 2.6: Two reference frames rotated relative to each other.

It is important that such a rotation preserves distance and natural orientation of  $\mathbb{R}^3$ , i.e. right-handed reference frames continue to be right-handed reference frames under rotation. This is also consistent with the definition of a rigid body, which is a configuration of points that preserve mutual distance between any two points during movement and rotation. If  $\underline{\mathbf{A}}$  is a transformation matrix then this transformation must be orthogonal and comply with the following constraints [4]

$$\begin{aligned} \underline{\mathbf{A}}^T \underline{\mathbf{A}} &= \underline{\mathbf{1}} \\ \det(\underline{\mathbf{A}}) &= 1 \end{aligned} \tag{2.3}$$

where  $\underline{\mathbf{1}}$  is a 3x3 identity matrix. The space spanned by all the transformation matrices satisfying the stated constraints are denoted  $SO(3)$  also called the special orthogonal group [4]. The matrix  $\underline{\mathbf{A}}$  has nine parameters, but only three are independent (three degrees of

freedom, hence six constraints). However, no three-parameter set can be both global and nonsingular [4] and a list of commonly used attitude parameterizations are presented in Table 2.1.

Representation	Par.	Characteristics	Applications
Rotation matrix (Direction cosine matrix)	9	<ul style="list-style-type: none"> <li>- Inherently nonsingular.</li> <li>- Intuitive representation.</li> <li>- Difficult to maintain orthogonality.</li> <li>- Expensive to store.</li> <li>- Six redundant parameters.</li> </ul>	Analytical studies and transformation of vectors.
Euler angles	3	<ul style="list-style-type: none"> <li>- Minimal set.</li> <li>- Clear physical interpretation.</li> <li>- Trigonometric functions in rotation matrix and kinematic relation.</li> <li>- No convenient product rule.</li> <li>- Singular for certain rotations.</li> </ul>	Theoretical physics, spinning spacecraft and attitude maneuvers. Used in analytical studies.
Axis-azimuth	3	<ul style="list-style-type: none"> <li>- Minimal set.</li> <li>- Clear physical interpretation.</li> <li>- Often computed directly from observations.</li> <li>- No convenient product rule.</li> <li>- Computation of rotation matrix difficult.</li> <li>- Singular for certain rotations.</li> <li>- Trigonometric functions in kinematic relation.</li> </ul>	Primarily spinning spacecraft.
Rodriguez (Gibbs vector)	3	<ul style="list-style-type: none"> <li>- Minimal set.</li> <li>- Singular for rotations near <math>\theta = \pm\pi</math>.</li> <li>- Simple kinematic relation</li> </ul>	Analytic studies.
Quaternions (Euler symmetric parameters)	4	<ul style="list-style-type: none"> <li>- Easy orthogonality of rotation matrix.</li> <li>- Not singular at any rotation.</li> <li>- Linear simple kinematic equations.</li> <li>- No clear physical interpretation.</li> <li>- One redundant parameter.</li> </ul>	Preferred attitude representation for attitude control systems.

**Table 2.1:** Common attitude parameters and their characteristics as presented in [1, 4].

Euler angles, axis-azimuth and Rodriguez (Gibbs vector) parameterizations all suffer from singularities, which is not a problem when working with rotation matrices and quaternions. Even though rotation matrices have a more intuitive representation, quaternions are the preferred attitude representation because of the smaller amount of parameters. Quaternions also have no trigonometric functions in the kinematics and provide a convenient product rule for successive rotations, which makes them computational faster. Among many, [2, 4, 37] uses quaternions as attitude parameters and they are also used in this thesis. The other attitude parameters are not treated further in this thesis, but are explained in e.g. [1].

The quaternion  $\mathbf{q}$  is a hyper complex number composed of a scalar  $q_4$  and a vector  $\mathbf{q}_{1:3}$ , with components spanning  $\mathbb{R}^3$ .

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_{1:3} \\ q_4 \end{bmatrix} = \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 + q_4 \quad (2.4)$$

For any unit quaternion, the operation

$$\mathbf{w} = \mathbf{q}^* \otimes \boldsymbol{\nu} \otimes \mathbf{q} \quad (2.5)$$

may be interpreted as a frame rotation relative to a fixed space of points or vectors  $\boldsymbol{\nu} = [\mathbf{v}^T \ 0]^T$ , or as taking the vector  $\mathbf{v}$  and expressing it in another frame, where the operator  $\otimes$  denotes a quaternion multiplication, which is defined in e.g. [38]. Quaternions are further described in Appendix A.

## 2.4 Satellite Equations of Motion

In the two previous sections reference frames were defined and possible representations of rotations of such frames were discussed. In Appendix C, the time dependent relative orientation between an inertial and a body fixed frame (ECI and CRF), also known as the equations of motion describing the attitude of the satellite, is derived. The kinematic equations of motion relates the time dependent attitude representation and the angular velocity of the satellite, while the dynamic equations of motion describe the relation between angular velocity and external torques acting on the satellite.

Combining the kinematic differential equation (Eq. (C.11)) and the dynamic differential equation (Eq. (C.19)) gives the nonlinear equations of motion of the satellite

$$\begin{aligned} \underbrace{\begin{bmatrix} \dot{\mathbf{q}}(t) \\ \dot{\boldsymbol{\omega}}(t) \end{bmatrix}}_{\dot{\mathbf{x}}} &= \underbrace{\begin{bmatrix} \frac{1}{2} \underline{\boldsymbol{\Omega}}(\boldsymbol{\omega}) \mathbf{q}(t) \\ \underline{\mathbf{I}}_{sat}^{-1} [-\underline{\mathbf{S}}(\boldsymbol{\omega}) (\underline{\mathbf{I}}_{sat} \boldsymbol{\omega}(t)) + \mathbf{N}_{dist}(t) + \mathbf{N}_{ctrl}(t)] \end{bmatrix}}_{f(\mathbf{x}, \mathbf{u}, \mathbf{w}, t)} \\ \underbrace{\begin{bmatrix} \mathbf{q}(t) \\ \boldsymbol{\omega}(t) \end{bmatrix}}_{\mathbf{y}} &= \underline{\mathbf{h}}_{sat} \underbrace{\begin{bmatrix} \mathbf{q}(t) \\ \boldsymbol{\omega}(t) \end{bmatrix}}_{\mathbf{x}} \end{aligned} \quad (2.6)$$

where:

$\mathbf{x} = [q_1 \ q_2 \ q_3 \ q_4 \ \omega_1 \ \omega_2 \ \omega_3]^T$  is the state vector.

$\mathbf{u} = \mathbf{N}_{ctrl}$  is the control torque.

$\mathbf{w} = \mathbf{N}_{dist}$  is the disturbance torque.

$\underline{\boldsymbol{\Omega}}(\boldsymbol{\omega})$  is a 4x4 skew symmetric matrix, defined in Eq. (C.10).

$\underline{\mathbf{S}}(\boldsymbol{\omega})$  is a skew symmetric matrix, defined in Eq. (C.20).

$\underline{\mathbf{I}}_{sat}$  is the diagonal constant inertia matrix about the principal axes.

In this state-variable description the states are the quaternion representing the rotation of the CRF relative to the ECI and the angular velocity of the CRF relative to the ECI, given in the CRF (the angular velocity will be denoted  ${}^c\boldsymbol{\omega}$  in the rest of this thesis). Frame notation is, however, omitted from Eq. (2.6). The states are also the outputs  $\mathbf{y}$  and  $\underline{\mathbf{h}}_{sat}$  is an 7x7 identity matrix, however, sensors and actuators also need to be chosen and modeled, which is done in Chapter 3 and 4.

## 2.5 Disturbance Models

The attitude of a satellite is influenced by a number of different disturbances. These can be both internal disturbances, caused by the satellite itself, and environmental disturbances. In this section the disturbances are also referred to as torques, since the disturbances are affecting the satellite as torques.

The internal disturbances can e.g. be caused by mass expulsion, propellant slosh and movement of mass on board the satellite. Satellites with rocket thrusters for active attitude control or orbital control contains propellant. In case of a leakage in the fuel system, the satellite will experience a disturbance caused by the mass expulsion (same principals as the thrusters). When the satellite uses its thrusters the tank will only be partially filled which can cause the propellant to slosh inside the tank affecting the satellite as an oscillating disturbance. Unintended movement of mass on board the satellite will also affect the satellite as a disturbance (same principals as momentum wheels). However, the above mentioned disturbances are not relevant for the AAUSAT3 satellite, due to the fact that there are no thrusters, propellants or moving parts on board.

The dominant external disturbances are caused by the aerodynamic drag in the upper atmosphere, the magnetic and gravitational field, solar radiation pressure and pressure from impacts of micrometeorites. The last mentioned can however be considered unlikely in orbits around the Earth [1].

The external disturbances have different regions in which they are dominant, which is illustrated in Figure 2.7. These dominant regions are highly dependent on the geometry and mass distribution of satellite under consideration. The relevant external disturbances for AAUSAT3 are described more thoroughly in the following and is based on [1, 3, 39].

### 2.5.1 Disturbance from the Earth's Gravitational Field

A non-symmetrical satellite in orbit about the Earth is subjected to a gravitational torque due to the Earth's non-uniform gravitational field (inverse proportional to the square of the distance to the Earth). It is important to notice that if the gravitational field of the Earth was uniform, no gravitational torque would affect the satellite.

Assuming that the satellite is only influenced by the Earth's gravitational field (justified in Subsection 2.1.3), the satellite consists of a single body and that both the Earth and the satellite are assumed to be two point masses, i.e. they have a symmetrical mass distribution, it is possible to calculate the gravitational force exerted by the Earth on the satellite  $\mathbf{F}_{Egs}$ , using newton's law of gravitation.

$$\mathbf{F}_{Egs} = \frac{Gm_E m_{sat}}{\|\mathbf{R}_{Es}\|^2} \hat{\mathbf{R}}_{Es} \quad [N] \quad (2.7)$$

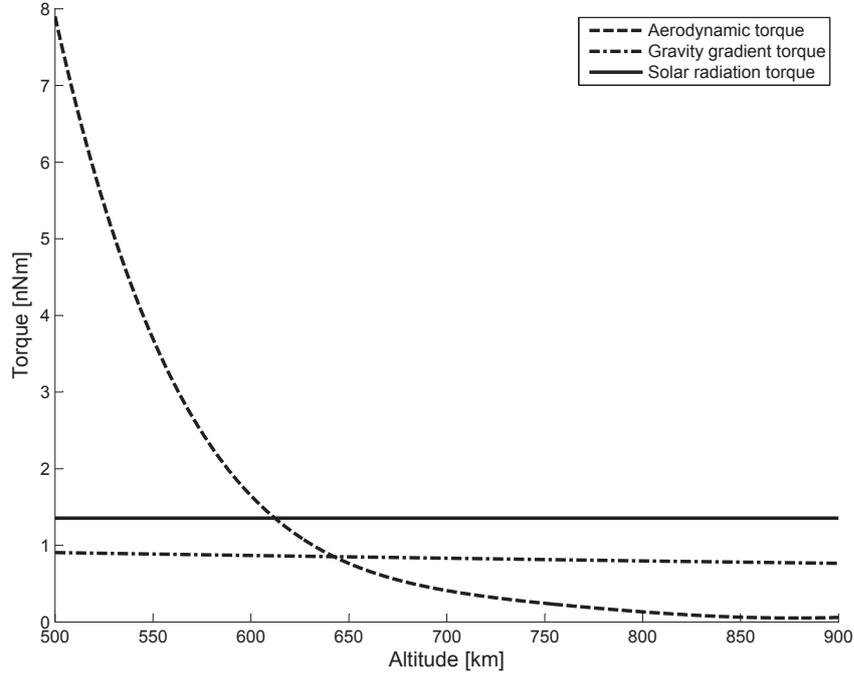
Where:

$G$  is the gravitational constant.

$m_E$  is the mass of the Earth.

$m_{sat}$  is the mass of the satellite.

$\mathbf{R}_{Es}$  is a vector pointing from the Earth's CoM to the satellite's CoM.



**Figure 2.7:** Regions of which each external disturbance is dominant for AAUSAT3. The illustrated torques have been calculated in the Matlab file *WCDTorque.m*, which can be found on the appended CD.

By modeling the satellite as a number of mass elements  $dm_i$ , the gravitational force applied on each individual mass element  $d\mathbf{F}_{Egs,i}$  is [1]

$$d\mathbf{F}_{Egs,i} = \frac{-\mu_E \mathbf{R}_{E,i} dm_i}{\|\mathbf{R}_{E,i}\|^3} \quad [N] \quad (2.8)$$

where:

$\mathbf{R}_{E,i}$  is a vector pointing from the Earth CoM to the element's CoM.

$\mu_E$  is the geocentric gravitational constant, defined as  $\mu_E \equiv Gm_E \quad [m^3/s^2]$ .

$dm_i$  is the mass of one element of the satellite.

The torque about the satellites geometric center  $d\mathbf{N}_{Egs,i}$  applied by the force  $d\mathbf{F}_{Egs,i}$  can then be found as [1]

$$d\mathbf{N}_{Egs,i} = \mathbf{r}_{sge,i} \times d\mathbf{F}_{Egs,i} \quad [Nm] \quad (2.9)$$

where:

$\mathbf{r}_{sge,i}$  is a vector from the satellites geometric center to the element's CoM.

According to [1] the gravity-gradient torque for the entire satellite  $\mathbf{N}_{gg}$  can be found as

$$\mathbf{N}_{gg} = \frac{\mu_E m_{sat}}{\|\mathbf{R}_{Ese}\|^2} (\hat{\mathbf{R}}_{Ese} \times \mathbf{r}_{sgc}) + \frac{3\mu_E}{\|\mathbf{R}_{Ese}\|^3} \int (\mathbf{r}_{sge,i} \times \hat{\mathbf{R}}_{Ese}) (\mathbf{r}_{sge,i} \cdot \hat{\mathbf{R}}_{Ese}) dm_i \quad [Nm] \quad (2.10)$$

where:

$\mathbf{R}_{Ese}$  is a vector pointing from the Earth CoM to the satellite's CoM.

$\mathbf{r}_{sgc}$  is a vector from the geometric center to the CoM of the satellite.

According to [1] this can be simplified to

$$\mathbf{N}_{gg} = \frac{3\mu_E}{\|\mathbf{R}_{Ese}\|^3} \left( \hat{\mathbf{R}}_{Ese} \times \left( \mathbf{I}_{sat} \hat{\mathbf{R}}_{Ese} \right) \right) [Nm] \quad (2.11)$$

where:

$\mathbf{I}_{sat}$  is the moment of inertia tensor for the satellite.

The simplification made in Eq. (2.11) assumes that  $\mathbf{r}_{sgc} = 0 [m]$ , i.e. the geometric center is the same as the CoM for the satellite.

## 2.5.2 Disturbance from Atmospheric Drag

The aerodynamic drag disturbance originates from atmospheric molecules that collide with the surface of the satellite. These impacts are possible to model as elastic, without reflection of molecules. Furthermore, if the velocity of the surface element  $dA_{se}$  is considered equal to the velocity of the satellite's CoM, then the force  $d\mathbf{F}_{ase}$ , exerted on a surface element, is given as [1]

$$d\mathbf{F}_{ase} = -\frac{1}{2} C_a \rho_a \|\mathbf{V}_{sat}\|^2 \left( \hat{\mathbf{n}}_{on} \cdot \hat{\mathbf{V}}_{sat} \right) \hat{\mathbf{V}}_{sat} dA_{se} [N] \quad (2.12)$$

where:

$C_a$  is the aerodynamic drag coefficient.

$\mathbf{V}_{sat}$  is the translational velocity vector of the satellite.

$\rho_a$  is the atmospheric density.

$\hat{\mathbf{n}}_{on}$  is the outward normal vector of the surface element on the satellite.

The aerodynamic drag coefficient  $C_a$  is usually between 1 and 2. A good estimate is  $C_a = 2$  in cases where the aerodynamic coefficient is unknown. The expected atmospheric density  $\rho_a$  at an altitude of 600 [km] is  $1.454 \cdot 10^{-13} [kg/m^3]$ , which is based on the Committee On SPACe Research (COSPAR) International Reference Atmosphere (CIRA 72) empirical atmospheric model [1].

It is important to notice that the atmospheric density is not constant, but varies as a function of e.g. solar activity and whether the air is in sunlight or not [39]. However, the air density will be modeled as constant in this thesis.

Assuming that the satellite can be seen as a plane, with the area  $A_{sat}$  and an outward normal unit vector  $\hat{\mathbf{n}}_{on}$  parallel to the translational velocity unit vector  $\hat{\mathbf{V}}_{sat}$  (see Figure 2.8), then the aerodynamic force  $\mathbf{F}_{as}$  acting on the satellite is given as [1]

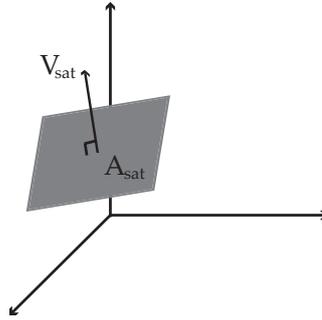
$$\mathbf{F}_{as} = -\frac{1}{2} C_a \rho_a \|\mathbf{V}_{sat}\|^2 \hat{\mathbf{V}}_{sat} A_{sat} [N] \quad (2.13)$$

The aerodynamic torque affecting the satellite  $\mathbf{N}_{as}$  is then given as

$$\mathbf{N}_{as} = \mathbf{r}_{scp} \times \mathbf{F}_{as} [Nm] \quad (2.14)$$

where:

$\mathbf{r}_{scp}$  is a vector from the CoM to the Center of Pressure (CoP) of the satellite.



**Figure 2.8:** Satellite modeled as a square plane with the area  $A_{sat}$  and an outward normal parallel to the translational velocity unit vector  $\hat{\mathbf{V}}_{sat}$ .

### 2.5.3 Disturbance from Solar Radiation

According to [1] the major sources of electromagnetic radiation are direct solar radiation, reflected solar radiation from the Earth and its atmosphere (the Earth's albedo) and radiation from the Earth and its atmosphere. The solar radiation includes all the electromagnetic waves emitted by the Sun and is the dominating source. Hence the solar radiation is generally the only source considered, which also is the case for this thesis.

The Sun also emits particulate radiation (solar wind), which mainly consists of ionized nuclei and electrons. However, the force produced by the solar wind is negligible relative to the solar radiation. Furthermore most of the solar wind is deflected by the Earth's magnetopause [1].

A sufficient method of modeling the force affecting the satellite due to solar radiation, is to assume that the incident radiation is either absorbed, reflected specularly or reflected diffuse. It is also possible to model the incident radiation as a combination of the previous mentioned cases.

The differential radiation force due to absorption  $d\mathbf{F}_{ras}$  on a surface element  $dA_{se}$  is [1]

$$d\mathbf{F}_{ras} = -P_{rmf} C_{ra} \cos(\theta_{son}) \hat{\mathbf{R}}_{sS} dA_{se} \quad [N] \quad (2.15)$$

where:

$P_{rmf}$  is the mean momentum flux acting on the normal to the Sun's radiation.

$C_{ra}$  is the absorption coefficient.

$\hat{\mathbf{R}}_{sS}$  is the a vector from the satellite's CoM to the Sun's CoM.

$\theta_{son}$  is the angel between  $\hat{\mathbf{R}}_{sS}$  and  $\hat{\mathbf{n}}_{on}$ , where ( $0^\circ \leq \theta_{son} \leq 90^\circ$ ).

The mean momentum flux  $P_{rmf}$  is given as

$$P_{rmf} = \frac{F_{solar}}{c} \quad [kg/ms^2] \quad (2.16)$$

where:

$F_{solar}$  is the mean integrated energy flux also known as the solar constant or Total Solar Irradiance (TSI).

$c$  is the speed of light in vacuum.

A TSI of  $1366 \text{ [W/m}^2\text{]}$  has been chosen. This value is the mean value of eight measurements obtained from 8 different satellites in 2005 [40].

The differential radiation force due to specularly reflected radiation  $d\mathbf{F}_{r_{ss}}$  on a surface element  $dA_{se}$  is [1]

$$d\mathbf{F}_{r_{ss}} = -2P_{rmf}C_{rs}\cos^2(\theta_{son})\hat{\mathbf{n}}_{on}dA_{se} \text{ [N]} \quad (2.17)$$

where:

$C_{rs}$  is the specular reflection coefficient.

The differential radiation force due to diffusely reflected radiation  $d\mathbf{F}_{r_{ds}}$  on a surface element  $dA_{se}$  is [1]

$$d\mathbf{F}_{r_{ds}} = P_{rmf}C_{rd}\left(-\frac{2}{3}\cos(\theta_{son})\hat{\mathbf{n}}_{on} - \cos(\theta_{son})\hat{\mathbf{R}}_{sS}\right)dA_{se} \text{ [N]} \quad (2.18)$$

where:

$C_{rd}$  is the diffuse reflection coefficient.

Assuming that the incident radiation is a combination of Eq. (2.15), (2.17) and (2.18), the radiation force for the  $i$ th surface area  $\mathbf{F}_{r_{ts,i}}$  can be calculated as

$$\mathbf{F}_{r_{ts,i}} = \int d\mathbf{F}_{r_{as}} + d\mathbf{F}_{r_{ss}} + d\mathbf{F}_{r_{ds}} \text{ [N]} \quad (2.19)$$

It is common to simplify the irradiated surface of the satellite using simple geometrical elements, e.g. a plane. Assuming that the satellite can be described as a plane with the area  $A_{sat}$ , then the solar radiation force exerted on the satellite  $\mathbf{F}_{rs}$  is given as [1]

$$\mathbf{F}_{rs} = -\left((1 - C_{rs})\hat{\mathbf{R}}_{sS} + 2\left(C_{rs}\cos(\theta_{son}) + \frac{1}{3}C_{rd}\right)\hat{\mathbf{n}}_{on}\right)\cos(\theta_{son})A_{sat}P_{rmf} \text{ [N]} \quad (2.20)$$

where:

$$C_{ra} + C_{rs} + C_{rd} = 1$$

Knowing that the cross-sectional area of  $A_{sat}$  is perpendicular to  $\hat{\mathbf{R}}_{sS}$  i.e.  $\hat{\mathbf{n}}_{on} = \hat{\mathbf{R}}_{sS}$ , it is possible to simplify Eq. (2.20) to

$$\begin{aligned} \mathbf{F}_{rs} &= -\left((1 - C_{rs})\hat{\mathbf{R}}_{sS} + 2\left(C_{rs} + \frac{1}{3}C_{rd}\right)\hat{\mathbf{n}}_{on}\right)A_{sat}P_{rmf} \\ &= -\left(\hat{\mathbf{R}}_{sS} - C_{rs}(\hat{\mathbf{R}}_{sS} - 2\hat{\mathbf{n}}_{on}) + \frac{2}{3}C_{rd}\hat{\mathbf{n}}_{on}\right)A_{sat}P_{rmf} \\ &= -\left(1 + C_{rs} + \frac{2}{3}C_{rd}\right)A_{sat}P_{rmf}\hat{\mathbf{R}}_{sS} \\ &= -C_{rk}A_{sat}P_{rmf}\hat{\mathbf{R}}_{sS} \text{ [N]} \end{aligned} \quad (2.21)$$

The constant  $C_{rk}$  is used to specify the outer material of the satellite. A value of  $C_{rk} < 1$  specifies a translucent material,  $C_{rk} = 1$  specifies a perfect absorbent material (black body) and  $C_{rk} > 1$  specifies a reflecting material, where  $C_{rk}$  is in the range  $0 \leq C_{rk} \leq 2$  [1].

The torque exerted on the satellite  $\mathbf{N}_{rs}$  is then given as [1]

$$\mathbf{N}_{rs} = \mathbf{r}_{scp} \times \mathbf{F}_{rs} \text{ [Nm]} \quad (2.22)$$

### 2.5.4 Disturbance from the Satellite's Magnetic Residual

The residual magnetic field of the satellite primarily originates from:

- Currents in the on-board electronics.
- Eddy-currents<sup>2</sup> induced in the on-board electronics.
- Hysteresis effects in soft ferromagnetic materials.

All these causes create a magnetic moment, from which the currents in the on-board electronics is the dominant source [1].

The magnetic residual torque  $\mathbf{N}_{mrs}$  is caused by the interaction between the magnetic moment of the satellite and the geomagnetic field given as [36]

$$\mathbf{N}_{mrs} = \mathbf{m}_{mms} \times \mathbf{B}_E \quad [Nm] \quad (2.23)$$

where:

$\mathbf{m}_{mms}$  is the effective magnetic dipole moment of the satellite.

$\mathbf{B}_E$  is the Earth's magnetic field.

It is hard to estimate the magnetic moment of a satellite and if compensated for it may be negligible. Lessons learned from the design of AAUSAT-II points out that ill-considered wiring for the solar cell's may cause a large magnetic moment for the satellite e.g. if the wires form a loop and conduct a large current through them.

The FireSat example given in [28] suggest a magnetic moment of 1  $[Am^2]$  for an uncompensated satellite with an average power of 110  $[W]$ . The expected average power of AAUSAT3 is 1  $[W]$ , which is approximately 100 times less than the FireSat. AAUSAT3 should be compensated and hence it is guesstimated that the magnetic moment of AAUSAT3 is 1000 times less than the magnetic moment of the FireSat, which corresponds to a magnetic moment of 0.001  $[Am^2]$ . This value has also been used in [41].

For this thesis it is thereby assumed that the magnetic residual of AAUSAT3 can be modeled by a magnetic dipole moment with a random direction and strength limited by a maximum strength of 0.001  $[Am^2]$ , which corresponds to the strength of a single current conducting loop, with the dimensions  $57 \times 57 [mm]$  given 1  $[W]$  @ 3, 3  $[V]$ .

## 2.6 Summary

In this chapter, the orbit of a celestial body has been described using the Keplerian orbital elements for a two body problem consisting of a satellite in low orbit about the Earth. The description also contains a summary of the TLE format used by NORAD to identify satellite orbits and this format will be used together with an orbit propagator to calculate the satellites position.

<sup>2</sup>Caused by conducting materials exposed to a changing magnetic field.

Five reference frames have been defined for the purpose of specifying e.g. position and attitude of the satellite, together with a presentation of quaternions, which gives a non singular way of representing rotations between reference frames.

The dynamic and kinematic equations of motion for the satellite have been identified, which forms the nonlinear state space model, where the states are the attitude of the satellite given by a quaternion representing the rotation of the CRF relative to the ECI and the angular velocity of the CRF relative to the ECI given in the CRF. Lastly models of the aerodynamic, gravity gradient, radiation and magnetic residual disturbances torques are given. The disturbances are expected to be dominated by magnetic residuals onboard AAUSAT3, where the other disturbances torques are approximately 1 [ $nNm$ ] and equal in an orbit height of 620 [ $km$ ]. However, it is important to notice that the worst case magnetic residual is an estimate and measurements should be performed on the satellite before launch.

## Hardware and Software Design for the ADCS

Sensors and actuators are important aspects of attitude determination and control. Models for these components must be part of the simulation environment for realistic evaluation of performance. Since AAUSAT3 is expected for launch in the first quarter of 2011 it is also important to start the development of a hardware prototype, in order to give room for test and revision of design, before making the final flight model.

Suitable sensors and actuators for AAUSAT3 are therefore identified in Section 3.1 and 3.2 respectively, followed by design of a actuators in Section 3.3 and 3.4. Section 3.5 then address the development of a hardware prototype for the ADCS. Lastly Section 3.6 presents some of the preliminary ideas for the software implementation, with identification of possible subsystem states. Development of a hardware prototype and the software design considerations are, however, not the main focus of this thesis and will therefore only be presented briefly.

### 3.1 Sensor Choice

General feedback control has no meaning if sensing is not employed. The attitude of the satellite must be measured and/or estimated in order to control it and a sufficient amount of sensors has to be implemented to get the necessary accuracy on the estimated attitude. This section first discusses different sensor possibilities with examples from other satellites and then concludes with a presentation of the sensors chosen for AAUSAT3.

#### 3.1.1 Sensor Possibilities

Two classes of sensors are often mixed in attitude determination systems [4]:

- Reference sensors
- Inertial sensors

Reference sensors use references such as the Earth, the Sun or the stars and provide vector observations. Often measurements from more than one reference sensor and measurements over time are fused, to overcome the problem of unobservability, where rotation about a measured vector from e.g. a sun sensor will be unknown. Fusion of data over time and combination with another reference sensor such as a magnetometer solves the problem of unobservability and improves the estimate.

Further improvement is obtained by adding an inertial sensor such as a gyroscope, which can provide angular rate relative to an inertial frame. During eclipse the sun sensor cannot provide vector observations, but the gyroscope can help propagate the attitude estimate. However, inertial sensors are limited by noise and bias errors and will not work alone, due to unbounded errors in the attitude estimate over time.

The sensor market is evolving rapidly, where sensors constantly become more precise, smaller, less power consuming and cheaper. A good example is the sensor comparison table found in [28], where e.g. a gyroscope typically weighs between 1-15 [kg] and consumes something between 10-200 [W]. Today's MEMS<sup>1</sup> technology makes it possible to fabricate gyroscopes weighing close to 0.001 [kg], while consuming under 0.025 [W]<sup>2</sup>. The sensor discussion that follows should therefore be seen as a guideline, due to the evolving sensor market and will furthermore not be exhaustive.

The main sensors used for attitude determination are star trackers, horizon sensors, sun sensors, GPS antenna arrays, magnetometers and angular rate gyroscopes. The rate gyroscopes can, as already mentioned, be very small, low power consuming and they are also cheap. Gyroscopes will therefore be a part of the attitude determination system. Reference sensors are also needed and Table 3.1 gives a quick comparison between the sensor types. The concept behind each sensor type is not addressed here, but references can be made to e.g. [1, 4].

### 3.1.2 Satellite Sensor Configuration Examples

The sensor configuration on five different satellites is presented in Table 3.2, before a final sensor choice is made. The satellites are AAUSAT-II [13], Ørsted [4], DTUosat [42], SSETI [31] and NSO [37]. The last satellite is a fictive satellite.

### 3.1.3 Sensors for AAUSAT3

Star trackers and GPS antenna arrays are both very expensive and hard to fit within the size and the power constraints of a CubeSat. The accuracy requirement for the mission does

---

<sup>1</sup>Micro-Electro-Mechanical Systems

<sup>2</sup>See e.g. gyroscopes from InvenSense, Inc.

Sensor type	Advantages	Disadvantages
Star tracker	<ul style="list-style-type: none"> <li>- Very high accuracy.</li> <li>- Lost in space function.</li> </ul>	<ul style="list-style-type: none"> <li>- Dependent on star identification.</li> <li>- Expensive.</li> <li>- Heavy and large.</li> <li>- High power consumption.</li> <li>- Sensitive to spin rates.</li> <li>- No output when pointing towards the Sun.</li> </ul>
Sun sensor	<ul style="list-style-type: none"> <li>- Simple sensor.</li> <li>- Low power consumption.</li> <li>- Analog sun sensors are cheap.</li> </ul>	<ul style="list-style-type: none"> <li>- Coarse accuracy.</li> <li>- Disturbed by Earth Albedo.</li> <li>- No output during eclipse.</li> <li>- Digital 2-axis sun sensors are expensive.</li> </ul>
GPS antenna array	<ul style="list-style-type: none"> <li>- Resistant against spin rates.</li> <li>- Also provides position, velocity and time.</li> </ul>	<ul style="list-style-type: none"> <li>- Expensive.</li> <li>- Large antenna array setup.</li> <li>- High power consumption.</li> </ul>
Horizon sensor	<ul style="list-style-type: none"> <li>- Earth always available.</li> <li>- Simple sensor.</li> </ul>	<ul style="list-style-type: none"> <li>- Coarse accuracy.</li> <li>- Accuracy affected by the Sun and Moon.</li> <li>- Not as widely used as sun sensors.</li> </ul>
Magnetometer	<ul style="list-style-type: none"> <li>- Simple sensor</li> <li>- Low power consumption.</li> <li>- Very cheap.</li> <li>- Very small.</li> <li>- Always available.</li> </ul>	<ul style="list-style-type: none"> <li>- Coarse accuracy.</li> <li>- Magnetic field not completely known.</li> <li>- Affected by on-board electronics.</li> <li>- Only applicable in LEO.</li> </ul>

**Table 3.1:** Advantages and disadvantages of different reference sensor types, if used in a Cubesat. Arguments are from [1, 4, 5].

Satellite	Sensors	Type
AAUSAT-II	Three axis magnetometer	HMC1023
	Six sun sensors (photodiodes)	SLCD-61N8
	Six 1-axis rate gyroscopes	ADXRS401
Ørsted	Three axis magnetometer	From DTU
	Three axis magnetometer	From CNES
	Star camera	From DTU
	Eight Sun sensors	-
DTUsat	Four axis magnetometer	-
	Sun sensors	From DTU
SSETI	Three axis magnetometer	HMR2300
	Two 2-axis sun sensors	From DTU
NSO	Three axis magnetometer	-
	Sun sensors	-
	Rate gyroscopes	-

**Table 3.2:** Sensor configuration examples on five different satellites. Some sensor types are unknown.

furthermore not require the use of e.g. a high precision star tracker. This leaves the choice with magnetometers, sun or horizon sensors.

Magnetometers are more reliable than sun and horizon sensors, because they continue to provide data during eclipse and does not have a limited FoV. Based on Table 3.2 it seems that sun sensors are more often used than horizon sensors. Sun sensors can also be very small and low cost, if photodiodes are used as in the AAUSAT-II case.

A 3-axis magnetometer, measuring the Earth's magnetic field intensity in three axes and a sun sensor setup is chosen for AAUSAT3.

Sun sensors from DTU [43] would be preferable due to their small size, low power consumption and high accuracy, but it was not possible to acquire these within the time frame available. A similar alternative was furthermore not possible to find. Therefore the setup from AAUSAT-II will be reused, where photodiodes (on each side of the CubeSat) measures the intensity of the incoming sun-light. This makes it possible to calculate a sun vector. Reusing the design from AAUSAT-II also saves development time.

Rate gyroscopes are, as concluded in Subsection 3.1.1, also a part of the ADS. The following criteria was used during the sensor component selection process for AAUSAT3:

- Resilience towards the harsh space environment (elaborated in Section 3.5).
- Low power consumption.
- Low weight and size.
- Ease of implementation.
- Accuracy.
- Cost.

This leads to the three sensors presented in the following.

### **Magnetometer: HMC6343**

Honeywell International Inc. has a wide variety of magnetometers. The HMC6343 is chosen for AAUSAT3 mainly based on its small size, low power consumption and ease of implementation. Table 3.3 presents some of the properties of HMC6343.

<b>Property</b>	
Type:	Triaxial digital magnetometer
Range:	$\pm 2$ [G]
Accuracy	3 [deg] heading accuracy (RMS)
Interface:	I <sup>2</sup> C
Power consumption:	15 [mW] @ 3.3 [V]
Sampling frequency:	1, 5 or 10 [Hz]
Size:	9x9x1.9 [mm <sup>3</sup> ] LCC surface mount
Operating temp.:	-40 to 80 [°C]
Other features:	Programmable offset (hard iron calibration) Programmable IIR filter Internal temperature correction

*Table 3.3: Properties of the HMC6343 magnetometer [6].*

The package provides magnetic field intensity measurements in three axes with an I<sup>2</sup>C interface, while consuming only 15 [mW].

It is important to keep the magnetometer away from any ferrous materials such as steel, iron and nickel, since they distort any magnetic fields. It is also important to keep the magnetometer at least a few millimeters away from high current traces ( $>10$  [mA]) [6] and as far away from any permanent magnets and coils (e.g. magnetorquers) as possible. The sensor has a programmable offset, that makes it possible to compensate for any effects caused by magnetized ferrous materials on board the satellite (hard iron calibration).

A sampling frequency of  $10$  [Hz] is four times faster than the fastest tumble rate experienced with AAUSAT-II [13]. This sampling frequency is expected to be enough to detumble the satellite, however, tests should be performed to verify how fast the satellite can tumble without losing stability in the detumble controller.

### Sun sensor: SLCD-61N8

The SLCD-61N8 photodiode is as mentioned reused from AAUSAT-II. Its properties are listed in Table 3.4.

Property	
Type:	Light sensing photodiode
Half angle:	$60$ [deg]
Short circuit current:	$170$ [ $\mu$ A] (typ)
Open circuit voltage:	$0.4$ [V] (typ)
Reverse dark current:	$1.7$ [ $\mu$ A] (max)
Interface:	Analog
Size:	$3.4 \times 1.3$ [ $mm^2$ ] surface mount
Operating temp.:	$-40$ to $125$ [ $^{\circ}$ C]

**Table 3.4:** Properties of the SLCD-61N8 photodiode [7].

The small size of the photodiode makes it possible to fit it in between the solar cells on the side panels.

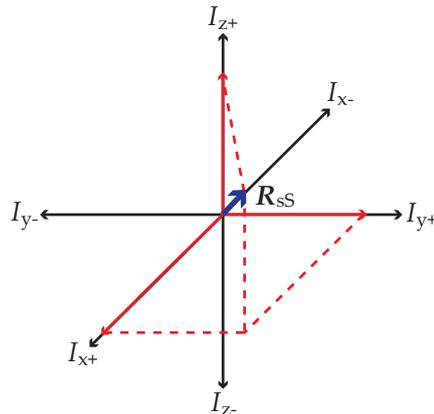
The signal from the light sensing photodiode needs to be amplified, filtered and A/D-converted. The design of a sun sensor PCB is not addressed in this thesis, but the achievable accuracy is guesstimated to be within  $\pm 10$  [deg], thus fairly inaccurate.

The principle behind determination of a sun vector  $\mathbf{R}_{SS}$  from six photodiodes is illustrated in Figure 3.1.

Only three sides of the satellite is illuminated by direct sunlight at any time, however, another major contribution to current in the photodiodes are the Earth albedo, which together can give currents on all sides of the satellite. The three sides with the highest current is used to calculate the sun vector, which is illustrated with red arrows in Figure 3.1.

The dark current<sup>3</sup> for each sensor should be measured beforehand and subtracted from each measurement. Each measurement should furthermore be normalised with the given sensors max current, which again can be determined beforehand. This ensures that potential differences in the sensors are eliminated.

<sup>3</sup>Current from the sensor in the absence of light.



**Figure 3.1:** Current axes for each of the six photodiodes. The grey box illustrates the satellite and the red arrows illustrate the three largest currents forming the sun vector  $\mathbf{R}_{sS}$  (blue arrow).

The design could be improved by having four photodiodes on each side of the satellite. This makes it possible to obtain a sun vector from each side individually, which will improve robustness because only one side is needed compared to three sides. The idea is to place two sensors with a split wall inbetween casting a shadow, which can be used to determine an angle to the sun. Two angles are needed to give a sun vector, thus requiring four photodiodes. This will potentially also improve performance as potential current biases, caused by e.g. temperature fluctuations, are reduced. The photodiodes in this setup will furthermore only be used within their half angles giving a much more linear output (the photodiodes on each side must only cover at least  $\pm 45$  [deg] in order to cover a complete sphere around the satellite).

### Gyroscope: IDG-1215 and ISZ-1215

It was not possible to find a digital gyroscope with a reasonable power consumption. The IDG-1215 two axis and ISZ-1215 one axis analog gyroscopes from InvenSense Inc. are therefore chosen for AAUSAT3. The InvenSense gyroscopes are mainly chosen based on their low size and low power consumption. Their properties are listed in Table 3.5.

The full scale range of  $\pm 67$  [deg/s] is the smallest available in their gyroscope series, providing high sensitivity. This range does not pose a problem because the gyroscope is intended for use in the attitude estimation for the pointing controller, thus when the satellite has been detumbled and is rotating slowly.

An integrated low-pass filter attenuates noise and high frequency components before final amplification. Furthermore an automatic amplification control and an internal temperature sensor ensures constant sensitivity over the specified temperature range of  $-20$  to  $85$  [ $^{\circ}\text{C}$ ] [8, 9]. The build in auto zero function also eliminates the need for an external high-pass filter to reduce the effect of DC bias offset due to temperature.

Property	
Type:	Angular rate gyroscope
Range:	$\pm 67$ [deg/s]
Sensitivity:	15 [mV/deg/s]
Non-linearity	<1% of full scale
Cross-axis Sensitivity	$\pm 1\%$
Interface:	Analog
Power consumption:	21+13.5=34.5 [mW] @ 3.0 [V] (total of both gyroscopes)
Size:	4x5x1.2 [mm <sup>3</sup> ] surface mount
Shock tolerance:	10.000 [g]
Operating temp.:	-40 to 105 [°C]
Other features:	Integrated amplifiers and low-pass filters Auto zero function On-chip temperature sensor

*Table 3.5: Properties of the IDG-1215 and ISZ-1215 gyroscopes [8, 9].*

The gyroscopes are based on MEMS fabrication technology and uses the Coriolis effect and a vibrating proof mass to measure angular rate directly [8]. This means that there is no drift in the measurement due to integration of acceleration.

## 3.2 Actuator Choice

The following presents a summary of different actuators, which are briefly described. Furthermore it is argued whether the given actuator is appropriate for AAUSAT3 or not and finally a presentation of the chosen actuator configuration is given.

### 3.2.1 Actuator Possibilities

**Thrusters:** Thrusters utilize Newtons third law of motion by expulsion of propellant at high velocity in one direction, thereby causing the satellite to be pushed in the opposite direction with an equal force. Thrusters requires additional weight and space on-board the satellite in order to store the propellant used by the thrusters. The amount of propellant, stored on-board the satellite, sets the effective lifetime of the ACS and since a CubeSat is small, then the effective lifetime will be short. Thrusters is hence considered inappropriate as actuators for AAUSAT3.

**Momentum Wheels:** Momentum wheels also utilize Newtons third law of motion by rotating a mass, creating a torque, which affects the satellite in the opposite direction of the rotation. Momentum wheels can also be used as a gyroscope by spinning the mass at a constant velocity, which stabilizes the satellite around the axes perpendicular to the axis of rotation. The momentum wheels includes moving mechanical parts, which rotates at high speed. The momentum wheels are hence considered inappropriate for use on AAUSAT3 according to requirement §6.1.

**Spin Stabilization:** A satellite can be stabilized around two axes by intentionally spinning the satellite rapidly around the third axis utilizing the gyroscopic effect created thereby.

The spinning around one axis is undesirable as pictures taken by the camera payload will become blurry. Dual spin stabilization is a similar approach, where two parts of the satellite spins in opposite direction of each other around the same axis. However, this requires a complex mechanical structure, which is inappropriate for CubeSats in general.

**Gravity Boom:** The gravity-gradient stabilization method utilize that the Earth's gravitational field is non-uniform. The satellite is stabilized around two axes by adding a gravity boom to the satellite or by making it long enough to take advantage of the gravity-gradient. According to [41] it is possible to obtain a precision of  $\pm 5$  [deg], which is sufficient for AAUSAT3, however, the gravity boom must be made expandable in order for the satellite to comply with the one unit CubeSat standard. As an expandable gravity boom include mechanical moving parts it is considered inappropriate for use on AAUSAT3 according to requirement §6.1. A gravity boom will also take up vital space on-board a CubeSat.

**Magnetorquers:** Magnetorquers<sup>4</sup> are a commonly used method to control small satellites in Low Earth Orbit (LEO) (see e.g. [44, 45, 46]). The magnetorquer creates a magnetic field, which will interact with the Earth's magnetic field, causing the satellite to rotate, until the two fields are aligned. The magnetorquer do not include any moving mechanical parts making it a simple and more reliable choice. The main disadvantage of using magnetorquers is that control torques can only be produced in a plane perpendicular to the local geomagnetic field.

**Permanent Magnet:** The permanent magnet works the same way as the magnetorquers, however, the permanent magnet do not require any power to control the satellite and can help ensure two axis stabilization the entire lifetime of the satellite, if the satellite has been detumbled. The main disadvantages of the permanent magnet are that it cannot remove kinetic energy from the satellite, i.e. the permanent magnet cannot remove oscillation from the satellite, and the satellite cannot be controlled in the sense of e.g. pointing the satellite towards a given point on the Earth.

### 3.2.2 Actuator Configuration for the AAUSAT3 Satellite

Based on the analysis in the above only magnetorquers and permanent magnet are appropriate actuators for use on the AAUSAT3 satellite. It is chosen to include both the magnetorquers and the permanent magnet in the ACS. 3-axis stabilization with magnetorquers is possible over time and this subject will be discussed further in Chapter 8. The addition of a permanent magnet adds some desirable advantages:

- **Pointing:** Implemented correctly, the permanent magnet will ensure that the antennas (UHF and VHF) are always pointed downwards towards the Earth over the northern hemisphere, which is advantageous in order to monitor ship traffic around Greenland and to communicate with the ground station in Aalborg.
- **Power reduction:** At normal operation, the permanent magnet will reduce the required power consumption of the ADCS, as the permanent magnet tracks the mag-

---

<sup>4</sup>A magnetorquer is essentially a electromagnetic coil.

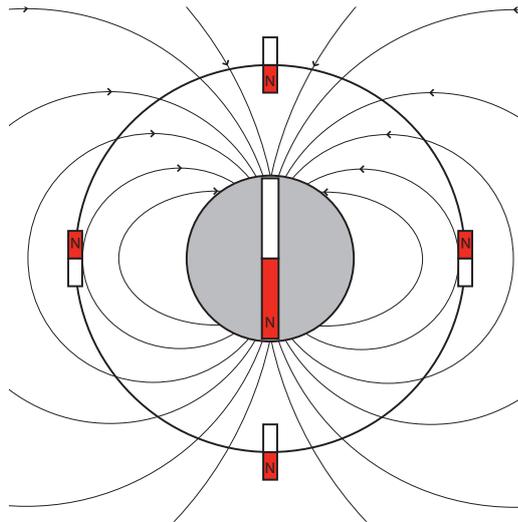
netic field of the Earth, thereby reducing the load on the magnetorquers during de-tumbling.

The expected control precision is dependent of the precision of the Attitude Determination System (ADS) and the algorithms used to control the satellite, however, other satellites such as the DTU-sat and AAU-CUBESAT have made it probable that a control precision of  $\pm 10$  [deg], which is the required precision of the ADCS for the AAUSAT3 satellite (see Section 1.6), is possible using magnetorquers only (based on the summary of ADCS configuration for CubeSats made in [41]).

Both the permanent magnet and the magnetorquers are described more thoroughly in Section 3.3 and 3.4.

### 3.3 Permanent Magnet Design

The attitude of the satellite with a permanent magnet in orbit about the Earth is illustrated in Figure 3.2, where both the satellite and the Earth are modeled as dipole magnets.



*Figure 3.2: Sketch of a satellite in orbit about the Earth, where both the Earth and the satellite is modeled as dipole magnets. The Earth's magnetic south pole corresponds to the dipoles magnetic north pole (indicated with an N).*

The permanent magnet induces an undesired oscillatory motion of the satellite due to the principle of conservation of mechanical energy.

#### 3.3.1 Oscillatory Motion of a Satellite with a Permanent Magnet

To describe the oscillatory motion of a satellite with a permanent dipole magnet located in the Earth's magnetic field, it is assumed, that the system is isolated. Hence, interactions can

only occur between the permanent magnet and the Earth's magnetic field. Furthermore it is assumed that the Earth's magnetic field is homogeneous. The mechanical energy of the system  $E_{tot}$  is then defined as [36]

$$E_{tot} = E_{kin} + E_{pot} \text{ [J]} \quad (3.1)$$

where:

$E_{kin}$  is the kinetic energy of the system.

$E_{pot}$  is the potential energy of the system.

The kinetic energy for a rotational system is [36]

$$E_{kin} = \frac{1}{2} \boldsymbol{\omega}^T \mathbf{I} \boldsymbol{\omega} \text{ [J]} \quad (3.2)$$

where:

$\mathbf{I}$  is the inertia matrix of the system.

$\boldsymbol{\omega}$  is the angular velocity vector of the system.

Considering a case where the initial position of the permanent magnet is misaligned with the Earth's magnetic field by an angle of  $\theta_{bpm}$  (always defined positive), as shown in Figure 3.3(a), then the potential energy stored in the system  $E_{pot}$ , is the difference in potential energy between the potential energy at the initial position  $E_{pot,init}$  and the minimum potential energy  $E_{pot,min} = 0$ , which is defined as when the permanent magnet is aligned with the Earth's magnetic field [36].

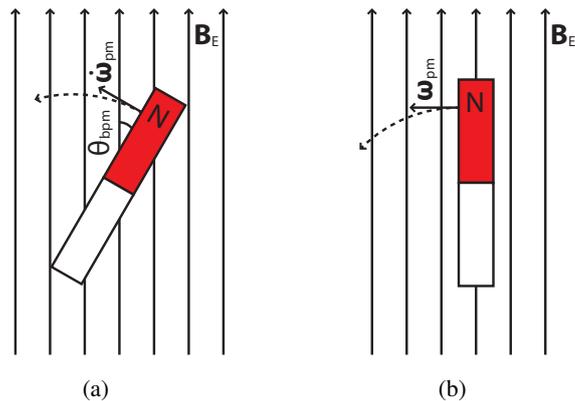
$$E_{pot} = \|\mathbf{m}_{pm}\| \|\mathbf{B}_E\| (1 - \cos(\theta_{bpm})) \text{ [J]} \quad (3.3)$$

Where:

$\mathbf{m}_{pm}$  is the magnetic dipole moment of the permanent magnet.

$\mathbf{B}_E$  is the Earth's magnetic field.

$\theta_{bpm}$  is angle between  $\mathbf{B}_E$  and  $\mathbf{m}_{pm}$ .



**Figure 3.3:** (a) Permanent magnet in initial position and accelerating towards the equilibrium point. (b) Permanent magnet aligned with the magnetic field lines  $\mathbf{B}_E$  (equilibrium) with the velocity  $\boldsymbol{\omega}_{pm}$ .

The torque  $\mathbf{N}_{pm}$  caused by the interaction between the permanent magnet and the Earth's magnetic field (see Equation 3.4) will rotate the permanent magnet to the left with the acceleration  $\dot{\omega}_{pm}$  until it is aligned with the Earth's magnetic field [36].

$$\mathbf{N}_{pm} = \mathbf{m}_{pm} \times \mathbf{B}_E [Nm] \quad (3.4)$$

As the permanent magnet is aligned with the Earth's magnetic field the potential difference of the system  $E_{pot}$  is zero. However, as the energy of the isolated system is conserved ( $\Delta E_{tot} = 0 [J]$ ), the energy is transferred to kinetic energy, meaning that the permanent magnet is moving with an angular velocity  $\omega_{pm}$ , as shown in Figure 3.3(b). When it has passed the equilibrium point it is decelerated until the potential difference of the system is equal to the initial potential difference, which is verified by Eq. (3.3).

For an isolated system the oscillation would continue forever, unless the energy is dissipated through either passive or active methods.

### 3.3.2 Design of Permanent Magnet

In order to design the permanent magnet the following must be taking into consideration:

- **Environment:** The permanent magnet must be suitable for use in space (see Section 3.5) and the size must be small enough to fit inside a CubeSat.
- **Torque:** The torque of the permanent magnet must be considered, as it causes the oscillatory motion described in the above.
- **Magnetic field strength:** The magnetic field strength of the permanent magnet must be considered, as it affects the measurements from the magnetometer.

By assuming that the magnetic moment vector of the permanent magnet and the local geomagnetic field vector are perpendicular in their initial position, it is possible to calculate the characteristic frequency of the oscillatory motion explained in the above as [31]

$$f = \frac{1}{2\pi} \sqrt{\frac{\|\mathbf{m}_{pm}\| \|\mathbf{B}_E\|}{I}} [Hz] \quad (3.5)$$

where:

$I$  is the inertia around the axis of interest.

A large permanent magnet is equivalent to a high characteristic frequency, which is desired to obtain fast magnetic field tracking. However, a high characteristic frequency may decrease the performance of the applications on board the satellite. As the magnetic field strength of the Earth can vary between 18 000 [nT] and 48 000 [nT] during an orbit (see Section 4.5), it is necessary to evaluate the characteristic frequency in both cases.

The magnetic moment of the permanent magnet used in Eq. (3.4) is calculated as [31]

$$\|\mathbf{m}_{pm}\| = \frac{B_i V_{pm}}{\mu_0} [Am^2] \quad (3.6)$$

where:

$B_i$  is the intrinsic induction of the magnetic material (remanence) in tesla.

$V_{pm}$  is the volume of the magnetic material.

$\mu_0$  is the vacuum permeability.

The permanent magnet creates a static magnetic field around the satellite, which is measured by the magnetometer. Hence the strength of the magnetic field must be constrained in order not to exceed the dynamic range of the magnetometer. The strength of the magnetic field evaluated on-axis of the permanent magnet can be calculated as [36]

$$B_{pm} = \frac{\mu_0 \|\mathbf{m}_{pm}\|}{2\pi (l_{dist}^2 + r_{pm}^2)^{3/2}} [T] \quad (3.7)$$

where:

$B_{pm}$  is the magnetic field strength of the permanent magnet at a distance of  $l_{dist}$ .

$l_{dist}$  is the on-axis distance from the permanent magnet.

$r_{pm}$  is the radius of the permanent magnet.

The permanent magnet should not be positioned on-axis with the magnetometer, but evaluating the on-axis field strength will give a worse than real estimate of the field strength.

### 3.3.3 Implementation of Permanent Magnet

There are many types of permanent magnets, from which many are possible choices for AAUSAT3. The choice of a suitable permanent magnet is based on an iterative process, where different magnetic materials and sizes are considered.

The final choice of a suitable permanent magnet for AAUSAT3 is a N35 sintered neodymium magnet, which is chosen based on the following properties [10]:

- **High remanence:** Sintered neodymium magnets have the highest magnetic field strength available. This is preferable as the weight and size of the permanent magnet can be minimized compared to other types of magnetic materials.
- **High coercivity:** A high resistance towards demagnetization is preferable as the permanent magnet will be exposed to various magnetic fields originating from the on board magnetorquers.
- **Suitable operating temperature:** The N35 sintered neodymium magnet has an operating temperature from  $-100$  [ $^{\circ}C$ ] to  $80$  [ $^{\circ}C$ ], which makes it suitable for use in space.
- **Low corrosion resistance:** High corrosion resistance is preferable for use in satellites. However, the surface of the magnet can be protected by coatings with high corrosion resistance e.g. stainless steel.

It is important to notice that the permanent magnet is kept within the operating temperature in order not to corrupt its magnetic properties. Hence placing the permanent magnet on the frame of the satellite will not be advisable.

The final choice of size is a cylindrical shaped magnet with a height of 0.001 [m] and a diameter of 0.002 [m]. The specifications of the chosen permanent magnet is summarized in Table 3.6.

Description	Low geomagnetic field strength	High geomagnetic field strength
Coercivity (typ.):	$892 \cdot 10^3$ [A/m]	$892 \cdot 10^3$ [A/m]
Remanence (typ.):	1.21 [T]	1.21 [T]
Magnetic moment:	0.0030 [Am <sup>2</sup> ]	0.0030 [Am <sup>2</sup> ]
Field strength:	15169 [nT]	15169 [nT]
Torque:	54.45 [nNm]	145.20 [nNm]
Frequency:	[0.0008 0.0008 0.0009] <sup>T</sup> [Hz]	[0.0013 0.0013 0.0015] <sup>T</sup> [Hz]
Period:	1274 [s]	689 [s]

**Table 3.6:** Specification of the chosen N35 sintered neodymium magnet [10]. The field strength is calculated on-axis of the permanent magnet at a distance of 5 [cm]. The given torque is calculated for a field perpendicular to the magnetic moment of the permanent magnet. The frequencies are calculated using Eq. (3.5), which gives a frequency in three dimensions [x y z].

The permanent magnet is placed in a corner of the satellite, which gives a distance of at least 5 [cm] to the magnetometer. As stated in Table 3.6 the on-axis field strength of the permanent magnet, at this distance, is approximately 0.15 [G], which is well below the dynamical range of  $\pm 2$  [G] of the magnetometer (see Section 3.1).

The field intensity  $\mathbf{H}_{dpm}$  required to demagnetize the N35 sintered neodymium magnet is  $892 \cdot 10^3$  [A/m]. The corresponding magnetic field  $\mathbf{B}_{dpm}$  is given by [36]

$$\begin{aligned} \mathbf{B}_{dpm} &= \mu_0 \mathbf{H}_{dpm} \\ &= 1.12 [T] \end{aligned} \quad (3.8)$$

The strongest magnetic source on board the satellite, beside the permanent magnet itself, are the magnetorquers. Hence if the maximum magnetic field generated by the magnetorquers are smaller, than the the magnetic field  $\mathbf{B}_{dpm}$  required to demagnetize the permanent magnet, then the coercivity is sufficiently high for the permanent magnet. The maximum magnetic field generated in the center of one set of magnetorquers can be calculated using Eq. (3.14) in Section 3.4, which gives a maximum magnetic field of 0.12 [mT]. This is significantly smaller, than the coercivity of the N35 sintered neodymium magnet.

To point out the influence of the permanent magnet it is beneficial to convert the characteristic frequency to an angular velocity. The lowest characteristic frequency stated in Table 3.6 corresponds to an angular velocity of 0.33 [deg/s], which is above the angular velocity of 0.12 [deg/s] for a detumbled satellite. Hence the permanent magnet is capable of tracking the geomagnetic field. The chosen design of the permanent magnet is considered as a good compromise between tracking performance and keeping a low bias on the magnetometer.

Furthermore, choosing a bigger permanent magnet is not desirable under pointing/tracking control, where the magnetorquers must overpower the torque from the magnet.

### 3.4 Magnetorquer Design

Magnetorquers are, as stated in Section 3.2, chosen as the primary actuator for AAUSAT3. Three magnetorquers are placed perpendicular to each other in order to provide the possibility of generating a magnetic moment vector with arbitrary direction.

This section first describes general design rules, followed by requirements for the magnetorquers. The section is then concluded with the design of magnetorquers without core and the design of magnetorquers that include an iron core.

#### 3.4.1 General Rules for Design of Magnetorquer

The torque  $\mathbf{N}_{mt}$  exerted on the satellite by the magnetorquer is given as

$$\mathbf{N}_{mt} = \mathbf{m}_{mt} \times \mathbf{B}_E \quad [Nm] \quad (3.9)$$

where:

$\mathbf{m}_{mt}$  is the effective magnetic dipole moment of the magnetorquer.

$\mathbf{B}_E$  is the local geomagnetic field.

The magnetic dipole moment of the magnetorquer is [36]

$$\mathbf{m}_{mt} = n_w I_{mt} \mathbf{A}_{mt} \quad [Am^2] \quad (3.10)$$

where:

$n_w$  is the number of windings for the magnetorquer.

$I_{mt}$  is the current in the magnetorquer.

$\mathbf{A}_{mt}$  is a vector which is perpendicular to the area enclosed by the magnetorquer.

The magnitude of  $\mathbf{A}_{mt}$  is equal to the area enclosed by the magnetorquer and the direction is parallel to the magnetic dipole moment  $\mathbf{m}_{mt}$ .

By inserting Eq. (3.10) into Eq. (3.9) and taking the magnitude yields

$$\begin{aligned} \|\mathbf{N}_{mt}\| &= n_w |I_{mt}| \|\mathbf{A}_{mt}\| \|\mathbf{B}_E\| \sin(\theta_{cB}) \quad [Nm] \\ \|\mathbf{N}_{mt}\| &= n_w |I_{mt}| A_{mt} \|\mathbf{B}_E\| \sin(\theta_{cB}) \quad [Nm] \end{aligned} \quad (3.11)$$

where:

$A_{mt}$  is the area enclosed by the magnetorquer.

$\theta_{cB}$  is the angle between the vector  $\mathbf{A}_{mt}$  and the geomagnetic field vector  $\mathbf{B}_E$ .

Equation 3.11 shows that by maximizing the parameters it is possible to maximize the resulting torque. Hence the following general design rules can then deduced:

- The area enclosed by the magnetorquer  $A_{mt}$  should be as large as possible in order to reduce the required current and number of windings.
- The magnetorquer should consist of a large number of windings  $n_w$ , which also reduce the required current. However, increasing the number of windings adds more weight to the satellite and also takes up more space.
- The current  $I_{mt}$  is preferred to be as small as possible to minimize the power consumption. However reducing the current means that the number of windings must be increased.

The magnitude of the local geomagnetic field  $\|\mathbf{B}_E\|$  and the angle between the vector  $\mathbf{A}_{mt}$  and the local geomagnetic field vector  $\mathbf{B}_E$  is varying over time, however, it is important to notice that the magnitude of the torque is largest when the area spanned by the magnetorquer is parallel to the local geomagnetic field, i.e.  $\theta_{cB} = 90$  [deg].

### 3.4.2 Required Torque from Magnetorquers

The worst case torque generated by the environmental disturbances in space and the permanent magnet, is used to design the magnetorquers, although it may be unlikely that AAUSAT3 will be exposed to the worst case torque in space (all disturbances pointing in the same direction).

In order to control the satellite in any given situation, the magnetorquers must overcome the worst case torque generated by the environmental disturbances and the permanent magnet. Furthermore they must be able to produce the torque required to both detumble and to track a position on the Earth or in space.

As the satellite is exposed to a varying geomagnetic field, the magnetic residual and the permanent magnet will affect the satellite with a varying torque. For this reason a total required torque is calculated for both the highest (48 000 [nT]) and the lowest (18 000 [nT]) magnetic field intensity, which the satellite is expected to be exposed to (see Section 4.5).

The worst case disturbance torque, estimated in Appendix D, is 51.88 [nNm] at high magnetic field intensity and 21.88 [nNm] at low magnetic field intensity.

The design of the permanent magnet is described in Section 3.3, where the torque affecting the satellite is calculated to be 145.20 [nNm] at high magnetic field intensity and 54.45 [nNm] at low magnetic field intensity.

In the requirements specified in Section 1.6, it is stated that the satellite must be detumbled from 10 [deg/s] to below  $\pm 0.30$  [deg/s] within three orbits. The torque required to detumble the satellite  $\mathbf{N}_{det}$  is calculated as

$$\mathbf{N}_{det} = \mathbf{I}_{sat} \dot{\boldsymbol{\omega}}_{det} \text{ [Nm]} \quad (3.12)$$

where:

$\mathbf{I}_{sat}$  is the inertia of the satellite determined in Appendix B.

$\dot{\boldsymbol{\omega}}_{det}$  is the angular acceleration vector required to detumble the satellite.

The angular acceleration needed to detumble the satellite  $\dot{\omega}_{det}$  is approximated by

$$\dot{\omega}_{det} = \frac{\omega_{req} - \omega_{init}}{t_{acc}} \left[ rad/s^2 \right] \quad (3.13)$$

where:

$\omega_{req}$  is the angular velocity of the satellite required for detumbling.

$\omega_{init}$  is the initial angular velocity.

$t_{acc}$  is the time period in which the satellite must be detumbled.

Using Eq. (3.13) and (3.12) and the detumble requirement, yields a required torque of 22.20 [nNm] to detumble the satellite.

In order for the satellite to track a position on the Earth, it is required, to be able to rotate with a angular velocity of 0.72 [deg/s] (see Section 1.6). The required torque to point the satellite can be found using the same approach as for the torque required to detumble the satellite. If assuming that the satellite must be accelerated from an angular velocity of 0 [deg/s], then the satellite requires a torque of 73.22 [nNm] to track a position on the Earth. This torque is multiplied with two, to account for uncertainties and assumptions, giving a torque of 146.44 [nNm].

The calculation of the required torque to detumble and point the satellite is located in the Matlab file `TumblePointTorque.m` on the appended CD.

All the required torques are summarized in Table 3.7, together with the total required torque.

Description	High magnetic field intensity	Low magnetic field intensity
Environment disturbances:	51.88 [nNm]	21.88 [nNm]
Permanent magnet:	145.20 [nNm]	54.45 [nNm]
Tracking:	146.44 [nNm]	146.44 [nNm]
Total:	343.52 [nNm]	222.76 [nNm]

**Table 3.7:** Required torque from the magnetorquers to control the satellite in a region where the Earth's magnetic field either has a low or high magnetic field intensity. Notice that the detumbling torque is not included in the total torque, because the torque required for tracking is larger than the required torque for detumbling.

It must be possible to track a position on the Earth or in space, when the satellite is in a region where the geomagnetic field has a low intensity, because the torque from the magnetorquers are also lowest here. This means that the magnetorquers must at least produce a torque of 222.76 [nNm]. In order to take additional uncertainties into account, e.g. CoM, inertia, magnetic residual disturbance, etc., it is chosen that the magnetorquers must be able to produce 400 [nNm] corresponding to roughly 80% extra torque.

Each of the three magnetorquers are split in two coils, sharing the load, i.e. one magnetorquer is designed to produce a torque of 200 [nNm]. If one coil fails it is still possible to control the satellite with half the torque, which makes the system more robust.

### 3.4.3 Design of Magnetorquers

The magnetorquers are designed through an iterative process, from which the presented design in Table 3.8, is the final result. The design is based on the general design rules stated in Subsection 3.4.1 and the values in Table 3.8 are calculated in the Matlab file `magnetorquer_dimensioning.m`.

Parameter	Value
Coil size:	75x75 [ $mm^2$ ]
Wire Thickness:	0.13 [ $mm$ ]
Windings	250 [-]
Coil mass:	0.053 [ $kg$ ]
Max voltage:	$\pm 1.25$ [ $V$ ] (controlled by PWM duty cycle)
Max current:	15.78 [ $mA$ ]
Actuator on time:	88%
Max power consumption pr. coil:	17.4 [ $mW$ ]
Total power consumption:	134.2 [ $mW$ ]
Coil Discharge time:	0.33481 [ $ms$ ] (99% discharged)
Available time for measurements:	11.67 [ $ms$ ]

**Table 3.8:** Design parameters for one magnetorquer coil without core that produces approximately 200 [ $nNm$ ] at low geomagnetic field strength (18000 [ $nT$ ]) perpendicular to coil area. Six of these magnetorquers are needed for the complete system. Total power consumption is for all six coils including driver circuits.

The area spanned by the magnetorquers are optimized to fit on the inside of the side panels of the satellite.

A copper wire thickness of 0.13 [ $mm$ ] (AWG 36) is chosen, because smaller values seems fragile and this value gives space for 2x250 windings (two coils per axis). The cross sectional area of the coil, when using 500 windings, is approximately 10 [ $mm^2$ ] or 4x2.5 [ $mm^2$ ], if a fill factor of 1.5 is used. The fill factor is used to account for wire insulation (0.02 [ $mm$ ]) and non optimal alignment of the wire when the coil is made. Additional space might be necessary for epoxy, which is why 500 windings is chosen as an upper bound. The chosen copper wire is available at the component shop at Aalborg University.

A buffered H-bridge (SI9988 from Vishay Siliconex [47]) is chosen as driver for the magnetorquer coils. This is the same driver used on AAUSAT-II. It is supplied with 5 [ $V$ ] giving an output range of -5 to 5 [ $V$ ] (it is a full H-bridge). The voltage is controlled by a PWM signal with a typical frequency of 100 [ $kHz$ ] and a variable duty cycle. The two drivers for the same magnetorquer axis are controlled by the same PWM signal, since the AVR8 do not have enough available output pins for six separate PWM signals. A max voltage range of  $\pm 1.25$  [ $V$ ] is chosen, because the voltage is squared in the power calculation, making it a costly parameter. This gives 8192 steps/levels between zero actuation (0 [ $V$ ]) and full actuation (1.25 [ $V$ ]), with the AVR8's PWM resolution of 16 [ $bit$ ].

Another duty cycle is introduced, since the actuator can not be turned on continuously, but has to be turned off during Earth magnetic field measurements. This is referred to as actuator on time in Table 3.8 and it is controlled with the drivers enable pin. This value is increased until the magnetorquer produces the required 200 [ $nNm$ ]. This parameter is cheaper than

the voltage parameter power wise, since it is not squared. 88% duty cycle leaves 11.67 [ms] for measurements, which, according to the datasheet for the HMC6343 magnetometer [6], is enough. 72 bits needs to be transferred according to the datasheet, which takes 0.72 [ms] (@ 100 [kpbs] I<sup>2</sup>C speed) and a 1 [ms] delay has to be added. This is a total of 1.72 [ms] and within the time available. The schematics for the driver circuits are placed in Appendix I.

The worst case resistance of the coil is calculated at a temperature of 85 [°C] and this resistance gives a maximum current through the coil of 15.78 [mA], at maximum actuation. The wire is rated for 40 [mA], which should not pose a problem. The worst case power consumption for the coil is 17.4 [mW] and the driver's worst case power consumption is 5 [mW]. This gives a total power consumption for all six magnetorquers of 134.2 [mW], as an extreme worst case.

The magnetorquer produces a magnetic field  $\mathbf{B}_{mt}$ , which disturbs the magnetometer. The strength of the magnetic field  $B_{mt}$ , at a given distance from the magnetorquers center, is calculated as [36]

$$B_{mt} = \frac{n_w \mu_0 I_{mt} L_{mt}^2}{2\pi \left( l_{dist}^2 + L_{mt}^2/4 \right) \sqrt{l_{dist}^2 + L_{mt}^2/2}} [T] \quad (3.14)$$

where:

- $B_{mt}$  is the magnetic field strength of the magnetorquer.
- $L_{mt}$  is the length of the magnetorquer.
- $I_{mt}$  is the current in the magnetorquer.

At a distance of 30 [mm] the magnetic field strength is approximately 89 321 [nT] for a set of magnetorquers. When the coil is discharge 99%, there is still 893 [nT] left, and the magnetometers should therefore be sampled at the end of the measurement period to allow the coil to discharge more.

The designed magnetorquer coils do not have any core, but introducing a core made of a ferrous material, will enhance the magnetic field produced by the coil considerably. This means that less power is necessary to actuate the satellite. A comparative list is presented in Table 3.9.

The solution with the core could be seen as the little and power saving choice, while the solution without a core could be seen as the secure choice. The permeability of different iron cores varies between typically 700-7600, depending on purity and material composition, and the permeability also varies with temperature. Ferrous materials also have a magnetisation hysteresis<sup>5</sup> that has to be taken into account, where iron has a preferable small hysteresis [36]. This makes a core solution hard to control, but compensation for temperature and tests on the coil revealing the cores permeability, makes it a possible solution. Magnetorquers with iron core are added to the AAUSAT3 satellite as an experiment, but also to give redundant actuators.

<sup>5</sup>Magnetic hysteresis is a kind of "memory" effect, where the ferrous material contains remanent magnetisation after the external field is removed.

Characteristic	With core	Without core
Power consumption:	+	-
Size:	+	-
Weight:	+	-
Placeable on ADCS PCB:	+	-
Innovative design (AAU):	+	-
Linearity over temperature:	-	+
Magnetisation hysteresis:	-	+
Uncertain permeability:	-	+
Small core hard to acquire:	-	+
Tested before (AAU):	-	+

**Table 3.9:** Comparative list of characteristics for magnetorquers with and without a core. “Innovative design” and “tested before” only relates to former satellites made at Aalborg University.

The PWM signals and driver circuit design from the magnetorquers without cores are reused, however the coils are not split in two for each axis, which means that only three drivers are needed. Table 3.10 presents the final design (magnetorquer with iron core), based on the Matlab file `magnetorquer_dimensioning.m`.

Parameter	Value
Core diameter:	10 [mm]
Core length:	10 [mm]
Permeability:	1000
Wire Thickness:	0.13 [mm]
Windings	200 [-]
Coil mass:	0.019 [kg]
Max voltage:	$\pm 1.25$ [V] (controlled by PWM duty cycle)
Max current:	20.0 [mA]
External resistance needed:	62.5 [ $\Omega$ ]
Actuator on time:	27%
Max power consumption pr. coil:	6.75 [mW]
Total power consumption:	35.3 [mW]

**Table 3.10:** Design parameters for one magnetorquer coil with core that produces approximately 400 [nNm] at low geomagnetic field strength (18000 [nT]) perpendicular to coil area. Three of these magnetorquers are needed for the complete system. Total power consumption is for all three coils including driver circuits.

The size of the magnetorquer cores are small enough to fit on the ADCS PCB and winding of a prototype indicated that 200 windings could fit on such a core. As the wire length is considerably shortened, the resistance is also lowered, which means that the current must be limited by an external resistor. Limiting the current to 20 [mA] (half of maximum for wire thickness), requires a resistance of 62.5 [ $\Omega$ ]. This resistor must be able to handle 6.75 [mW] in vacuum.

Even though the magnetorquer with iron core is smaller and has less windings, it still produces a larger magnetic moment than required. This means that the actuator on time can be

lowered to 27%. Comparing the total worst case power consumption reveals that approximately 100 [mW] is saved, if a core is added to the coil.

A permeability of 1000 is suggested for the iron core, which is considered achievable over the whole temperature range. Finding a soft ferrous material that has the specified permeability is left as a future task.

## 3.5 Hardware Design

The expected launch for AAUSAT3 is in the first quarter of 2011. This means that prototype hardware for the ADCS must be finished during the summer 2010, before an extensive test period and delivery of hardware to the launch provider. A prototype is therefore developed in order to have an iterative design procedure, where faults and errors in the design hopefully will be corrected. Making a prototype also opens up for test of sensors and actuators.

### 3.5.1 The Space Environment

Hardware in space is exposed to a various number of degrading effects. The hostile space environment therefore puts stringent requirements on the hardware put in LEO. The following list is based on [48]:

- The temperature differences, between being in direct sun-light and not, are high. Temperature data from the Ørsted satellite showed temperatures between 10 to 40 [°C] inside the satellite and temperatures between -40 to 80 [°C] on the side panels. This data was recorded on January the 9th, 2010 [49].
- The vacuum in space causes some materials (especially plastic) to outgas. If air bobbles are trapped within e.g. components it can also cause them to crack or even explode, due to the high pressure difference. Another effect of vacuum is that heat is not transferred via air convection. This can give heat problems for components, since heat is only transferred to materials in direct contact with the components or via radiation.
- Radiation doses in space are higher than on Earth. Prolonged radiation degrades the performance of components, like an aging effect and can cause "single events", where e.g. bits in memory are flipped due to high energy nucleus hitting the component.
- The hardware is exposed to high g-forces, because of violent shaking during launch. The satellite could also get hit by micro meteorites.
- Ultra violet radiation causes some materials to darken.
- The few atomic oxygen particles that are still present in LEO has shown to be very erosive on plastic and some metals.

The requirements for the prototype can be relaxed a bit in regard to materials used to fabricate the PCB. However, it is still important that the components chosen for AAUSAT3 can withstand the possible temperature range and the vacuum of space. The prototype should therefore be tested in a temperature and vacuum chamber. Shaking and possibly radiation tests should also at least be a part of the test of the final flight model. A detailed list of g-forces and radiation doses a satellite should be able to withstand, can be found in e.g. [28].

### 3.5.2 ADCS Prototype PCB

The heart of every subsystem in the distributed system on AAUSAT3 is a MCU. In order to meet requirement §6.3 stating the ADCS detumble hardware must be fully redundant, both an AT90CAN128 MCU (AVR8) and an AT91SAM7A3 MCU (ARM7) is used in the ADCS, giving the possibility of having two overall power modes: A low power mode with the AVR8 running at 8 [MHz] called ADCS1 and a high power mode with the ARM7 running at 60 [MHz] called ADCS2 . The high power mode gives more computational power for high performance algorithms and the two MCU setup creates a redundant system.

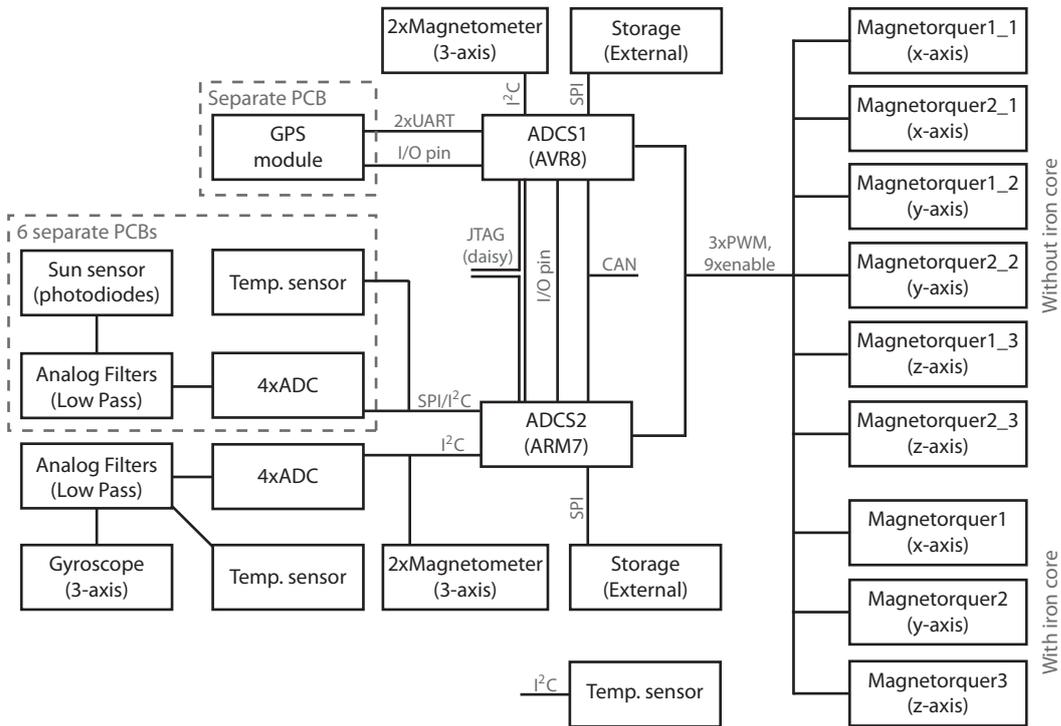
The AVR8 is already used in other subsystems and is well tested, hence a good reason to reuse it. The ARM7 is not used in any subsystem, but it has CAN interface and it is from ATMEL, making it easier to port the software framework for the AVR8 to the ARM7. This choice of MCU has a good balance between ease of implementation, computational power and interfaces. Using the ARM7 can also be seen as a test for future missions, to see how well it works in space, and since it is part of a redundant system, it is not a critical component. The algorithms for the ADCS should be optimized to run on the resources available in the AVR8 and the ARM7 MCUs.

A hardware block diagram with interfaces has been drawn to create an overview of the hardware design for the ADCS, see Figure 3.4. Schematics for the prototype ADCS PCB is placed in Appendix I and on the appended CD.

It has been decided by the system engineering group that the ADCS PCB should interface with the Phoenix GPS module. This interface is controlled by the ADCS1 MCU and the GPS module requires two UART connections and an I/O pin [50].

The digital interfaces in Figure 3.4 are listed in the Interface Control Document (ICD), in Appendix H.1. The ICD also contains a specification of the five power channels assigned to the ADCS and the GPS, which are not shown in the figure. The five power channels are separated as follows:

- ADCS main power: For MCUs, magnetometers, external memory and power injection to GPS antenna.
- ADCS sensor power: For gyroscope and sun sensors.
- ADCS actuator power: For magnetorquers.
- GPS main power: For GPS module.



**Figure 3.4:** Hardware blokdiagram for the ADCS with interfaces. Dashed boxes indicate components placed on seperate PCBs.

- GPS backup power: Instead of GPS backup battery.

Separating the ADCS power in three gives improved fault tolerance towards short circuits, potentially reduced noise levels for sensors and more power control possibilities for the EPS. An idea would be to split the ADCS main power in two, one channel for each MCU. This makes the ADCS even more tolerant towards short circuits, but the EPS must have enough channels available.

The sun sensor photodiodes must be placed on the sides of the satellite and to reduce noise in the measurement, the ADC is placed together with the photodiodes on six separate PCBs, to be mounted on the side panels. A temperature sensor should be placed on each PCB, to be able to reduce the effect of temperature dependence and to provide satellite status. It has not been decided yet if the sun sensor PCBs should use a SPI or a I<sup>2</sup>C interface, but both interfaces are provided by the ADCS2. Attitude estimation and pointing control is handled by the ADCS2 MCU, which is why the ADCS1 does not interface with the gyroscopes and sun sensors.

Each MCU has a separate interface to two 3-axis magnetometers. This redundancy gives robustness towards I<sup>2</sup>C bus-errors, and makes it possible to compare all four magnetometer outputs when both MCUs are turned on. Gyroscopes and sun sensors are not redundant because the primary function of the ADCS is detumbling, which does not require these sensors. Loosing the ability to estimate the attitude and do pointing control is not critical to

the primary mission goal, which is to receive AIS signals from ships.

The nine magnetorquers (driver and coil) in Figure 3.4 are designed in Section 3.4. An important feature on the ADCS PCB is the I/O-pin connection between the two MCUs. Whenever the ADCS2 uses the actuators, it sets this pin high, and before ADCS1 uses the actuators it must sense if the pin is high. If it is high the ADCS1 is not allowed to control the actuators.

The external data storage makes it possible to save ADCS data for a prolonged period of time, e.g. one orbit, for debugging on ground and for experiments on the ADCS. The necessary amount of storage is approximately 1.3 [MB], as specified in Subsection H.2.

The analog gyroscope is sampled with a 16 bit ADC. If calculating with 15 effective bits, this gives an approximate resolution of  $\frac{3.3[V]}{2^{15}} = 0.1 [mV]$ . With a gyroscope sensitivity of 15 [mV/deg/s] it is theoretically possible to determine the angular velocity with a resolution of 0.067 [deg/s]. Choosing a 24 bit ADC would give better resolution, but reducing the potential noise on the power supply below a level that can justify this, is not considered possible.

## 3.6 Software Design Considerations

The software for the ADCS should be implemented based on the following description of desired operational states. In general state transitions are handled by a supervisory system, the EPS and the FP or performed by manual commands from the ground station. Possible telecommands are also indentified and presented in Table H.1 in Appendix H.

### 3.6.1 State Description

Seven states have been identified and accepted by the system engineering group.

- **State: OFF**

Per default the EPS turns on the ADCS1, when there is enough power on the batteries and turns it off again when commanded to or when the battery level is low. A settable variable in the EPS EEPROM determines if the EPS should start ADCS1 or ADCS2 per default. This variable is set by the ADCS supervisory system which monitors the system or from the ground station and storing it in EEPROM saves it between power off/on cycles. It is important that the EPS turns on the ADCS detumbling autonomously to maintain low angular rates of the satellite. On AAUSAT-II detumbling has to be turned on manually and with the frequent rebooting of the system, detumbling commands has to be send to the satellite continuously, otherwise the satellite has shown to increase its angular spin rate [13]. Both ADCS1 and ADCS2 can be turned on at the same time if desired, but only one of them can have control over the actuators at a time. ADCS2 which can also do pointing of the satellite has priority of the actuators and ADCS1 must sense, on an input pin connected to ADCS2, if it is allowed to actuate.

- **State: DIAGNOSTICS**

When starting up the ADCS a check is made whether to perform a diagnostics on the subsystem or not. Per default this is always done and if a sensor or actuator is malfunctioning and the supervisor cannot switch to a working set, the ADCS will set a guard prohibiting the use of actuators, unless in TEST state.

- **State: IDLE/SLEEP**

When the ADCS MCU's are powered up and a diagnostics check has been performed, they will go into IDLE/SLEEP state, where no sampling, computation or actuation is performed. If no commands are received within a certain time period, they will go into low power mode, with the possibility of waking them up individually. If the supervisory system cannot switch to a working set of sensors and actuators, when a fault is detected, a switch to the IDLE state is also performed.

- **State: TEST**

This state gives the possibility of taking manual control over the satellite, used mainly for experimental purposes. A control sequence is sent to the satellite and when entering the TEST state, this sequence is used while data is sampled. Data is raw sensor values, attitude estimate, residuals, control signal, controller state and time stamp. This data gives the possibility of evaluating the performance of the ADCS from the ground station. In TEST state all sensors are sampled and attitude estimation is performed. If the satellite exceeds a tumbling threshold, a state switch to DETUMBLE is automatically performed, to recover the satellite.

- **State: STANDBY**

In this state all sensors are sampled and attitude estimation is performed, but no control is enforced. A switch to this state can happen in the states IDLE, TEST, DETUMBLE and POINTING.

- **State: DETUMBLE**

When the ADCS is started, it automatically enters this state, unless faults were found in DIAGNOSTICS that renders the satellite uncontrollable with magnetorquers. AAUSAT-II has shown to increase its tumbling velocity during periods with an inactive detumble controller. Automatic entry into the DETUMBLE state eliminates the problem with the constant reboot of the main computer on AAUSAT-II, where the detumble controller has to be turned on manually from the ground station after every reboot. If the satellite starts to tumble faster than a predetermined threshold the ADCS will automatically enter DETUMBLE again, if it is in the TEST state or the POINTING state.

- **State: POINTING**

When the satellite is detumbled the ADCS can switch to POINTING state, which is the state taking care of pointing the satellite towards a reference. This reference could be the center of the Earth, the Sun, Aalborg or maybe a place in Greenland. This state requires more advanced controllers that require more computational power and interfaces to the angular rate gyroscopes and sun sensors, which the ADCS1 MCU does not provide. This state is therefore only reachable by ADCS2.

### 3.6.2 Operational states summary

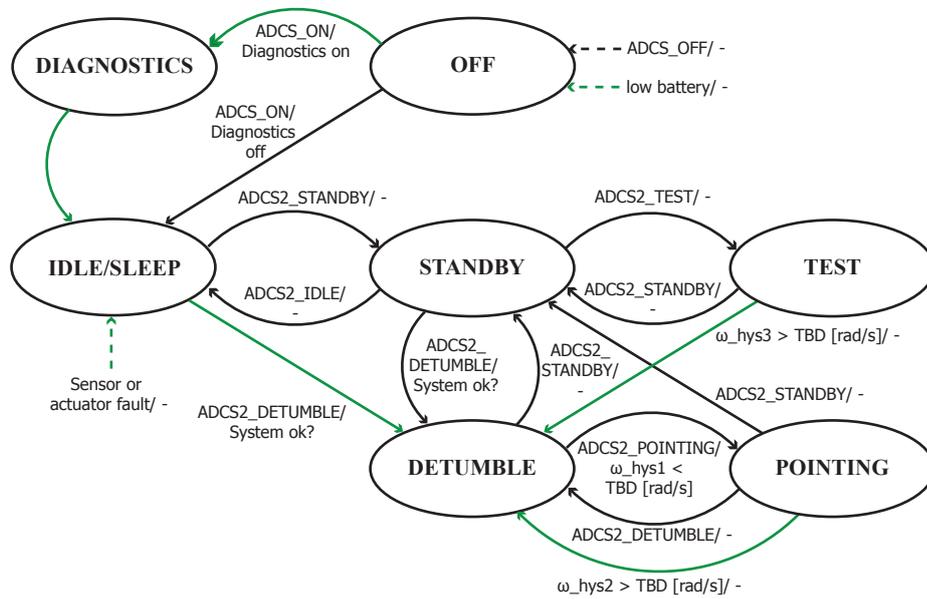
Table 3.11 gives a summary of the properties of each state.

State	Sensor sampling	Attitude estimation	Control
OFF	No	No	No
IDLE	No	No	No
DIAGNOSTICS	Yes	No	Predetermined
STANDBY	Yes	Yes	No
TEST	Yes	Yes	Manual
DETUMBLE	Yes	no	Simple
POINTING	Yes	Yes	Advanced

*Table 3.11: Operational states summary. Both ADCS1 and ADCS2 can be turned on at the same time, but only one of them enforces its control on the actuators. Attitude estimation can only be done by ADCS2.*

### 3.6.3 State Transition Diagram and Telecommands

A state transition diagram has been drawn to illustrate the interconnection between states, see Figure 3.5.



*Figure 3.5: State transition diagram for ADCS2. The diagram is equivalent for ADCS1, except that ADCS1 cannot go to the POINTING state. State switches are defined by events/guards. A dashed line indicates that the switch can occur in all states and green arrows indicate autonomous state switching.*

The telecommands are only accepted if the ADCS is in the right state, otherwise it is ignored. A list of possible telecommands is presented in Table H.1 in Appendix H.

There should also be the possibility of switching between sensor and actuator sets if there is redundancy, both manually and autonomously through a supervisory system running FDI. However, design of FDI is not the main focus in this thesis.

Those variables that can be changed from the ground station, should be saved in EEPROM to eliminate the need for new software upload, which can take a long time over the limited space-link. This will also mean that the new variable values are kept during power off/on cycles. Table H.2 in Appendix H contains a list of potential EEPROM variables.

### 3.7 Summary

The objective of this chapter has mainly been to identify the sensor and actuator setup for AAUSAT3. It was also the objective of this chapter to describe the development of a hardware prototype and possible subsystem states.

The chosen sensor setup consists of a 3-axis gyroscope from Invensense, four 3-axis magnetometers from Honeywell and photodiodes will be used as sun sensors, since a digital 2-axis sun sensor was too expensive for AAUSAT3. The complete sensor setup is estimated to cost less than 2000 DKK, which is considered very cheap. This, however, comes at the cost of precision and accuracy.

A permanent magnet and magnetorquer coils in three perpendicular axes have been chosen for actuation, which means that AAUSAT3 will only have magnetic actuation. The permanent magnet has a magnetic moment of  $0.0030 \text{ [Am}^2\text{]}$ , which is three times the expected worst case magnetic moment of the potential magnetic residuals and the magnetorquer coils have been split in three for redundancy, giving a total of nine coils. Three of the coils will be experimental coils with iron cores, which will be able to produce the same torque at lower power consumption.

Design of a PCB prototype was furthermore presented together with schematics for manufacturing in Appendix I.

## Simulink Simulation Environment

Testing estimators and controllers for spacecrafts is a difficult task. It is possible to emulate weightlessness<sup>1</sup>, vacuum and temperature conditions close to the environment in space. However, this is very expensive and still does not give correct disturbance torques. Total physical replication of the environment in space is not considered as an option for AAUSAT3, but introduction of a realistic simulator working as a “truth model” makes it possible to evaluate the performance of the developed algorithms. It is important that the “truth model” emulates the real system and environment well enough to be able to give good estimates of the performance and robustness of the developed estimator and controller algorithms in space. This chapter describes such a simulator that simulates the orbit and attitude dynamics of the satellite including realistic disturbances. Sensor and actuator models are also introduced, which acts as interfaces between the developed algorithms and the “truth model”.

### 4.1 Simulink Implementation

The simulation environment made for AAUSAT3 is implemented in Matlab as a Simulink library, and is based on an existing Simulink implementation for the NSO satellite [37]. Their implementation was furthermore based on the AAUSAT-II simulation environment made by [2, 45, 51, 52], with an Earth albedo model by [53] and an IGRF model implementation by [4].

Working with Matlab Simulink is considered intuitive and visual, since everything can be represented by reusable blocks, which eases the development and test of algorithms. Furthermore it would be irrational to re-implement the whole simulation environment made by the former groups, because it has taken them a considerable amount of time and because it has been proven to work well. However, some blocks are modified and updated to fit the dimensions and mass of the AAUSAT3 and new sensor and actuator models are introduced.

---

<sup>1</sup>E.g. with zero-G flights or gyro gimbal systems

C code compiled as mex files and Matlab files are used for algorithms such as orbit propagators, magnetic fields models, estimators and controllers in order to ease the transition from a Matlab implementation to a C based implementation on board the satellite. Matlab files can automatically be compiled to C code with Matlab’s compiler.

Figure 4.1 shows the top hierarchical content of the block called AAUSAT3 or “truth model”.

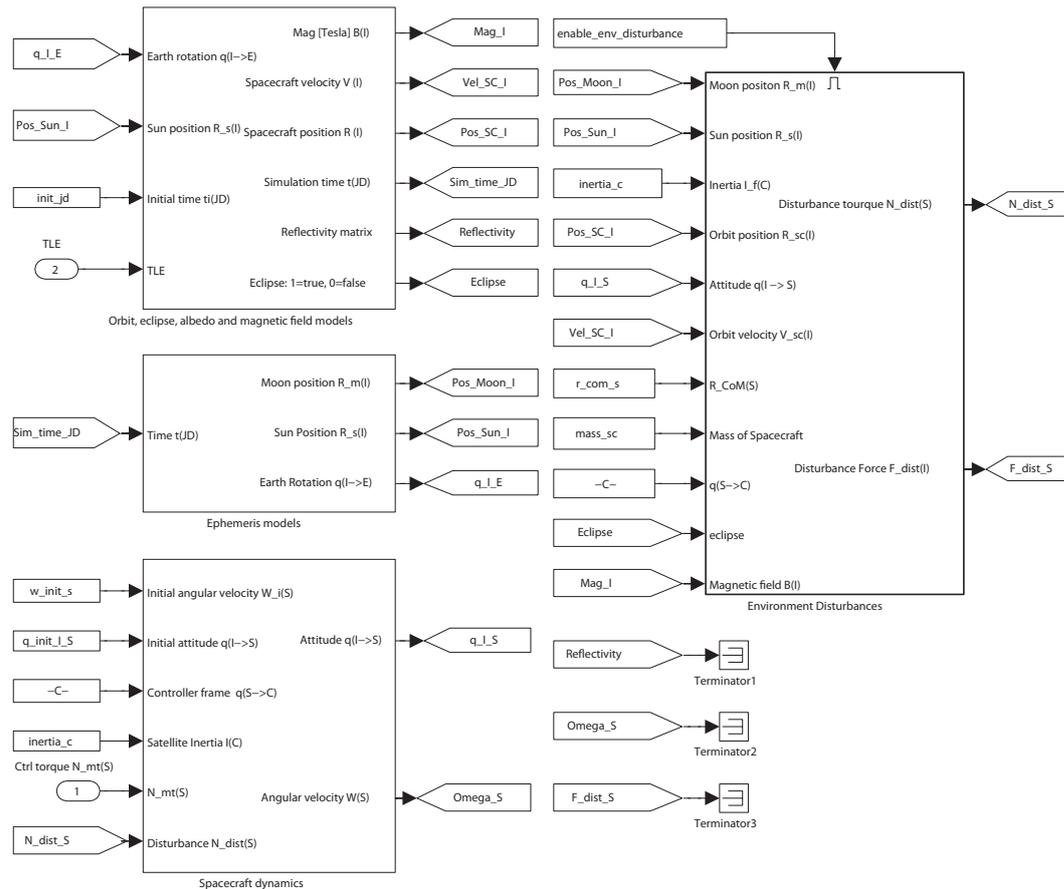


Figure 4.1: Top hierarchical content of AAUSAT3 block in Simulink.

The block containing the orbit, eclipse, albedo and magnetic field models converts an initial time and a TLE into simulation time in Julian Date (JD). This information is used by the orbit propagator, described in Section 4.2, to calculate the position of the satellite in the ECI. The Ephemeris model, described in Section 4.3, also uses the simulation time to calculate the position of the Sun, the Moon and the rotation of the ECEF relative to the ECI.

Knowing the position of the Sun, the satellite and the rotation of the Earth relative to the ECI makes it possible to calculate when the satellite is in eclipse and the Earth albedo affecting the sun sensors, as described in Section 4.4. It is also possible to calculate the Earths magnetic field vector with the magnetic field model, described in Section 4.5.

The spacecraft dynamics block contains the satellite equations of motion derived in Section

2.4 and will not be treated further in this chapter. Furthermore the environmental disturbance block uses the output from the Ephemeris block and the orbit, eclipse, albedo and magnetic field models block to calculate the dominating disturbances acting on the satellite. These disturbances are caused by magnetic residual, solar radiation, atmospheric drag and the Earths gravitational field. The block also calculates gravitational disturbances from the Sun, the Moon and the Zonal harmonics, however, these are very small and considered negligible, see Figure E.9 in Appendix E. A general test of the AAUSAT3 block can also be found in Appendix E.

Figure 4.2 shows the top hierarchical content of the sensor and actuator models.

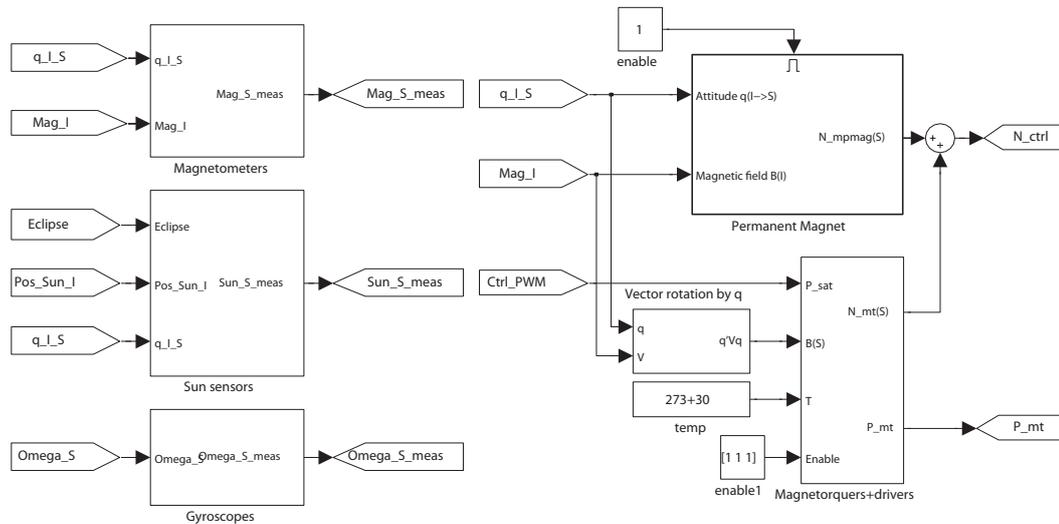


Figure 4.2: Top hierarchical content of sensor and actuator models implemented in Simulink.

The sensor models take some of the outputs from the “truth model” and simulate realistic outputs from magnetometers (x, y and z components of the magnetic field), sun sensors (sun position vector) and gyroscopes (angular rate about the x, y and z axis). These outputs are all in the SBRF and the models are described in Section 4.6. The permanent magnet and magnetorquer plus driver models, providing control torques on the satellite, are described lastly in Section 4.7.

## 4.2 Orbit Propagators

An orbit propagator makes it possible to simulate realistic disturbances. This is obvious when looking at the Earths magnetic field, which varies in strength and direction depending on the location of the satellite, thus giving a changing disturbance torque from the satellites magnetic residual. Additionally, having the satellites position relative to the Earth and knowing where the Sun is, makes it possible to calculate an estimate of the disturbance torque caused by solar radiation and makes it possible to emulate sun sensors.

When evaluating the performance of the pointing controller it is relevant to know the satellite's orbital position, since the controller could be told to point the satellite towards e.g. Aalborg.

An orbit propagator implemented in the software on board the satellite is also important because the attitude estimator needs to know the expected measurements from the sensors, see Sections 5.3 and 5.4.

### 4.2.1 Orbit Propagator Types

A general description of orbits and Two Line Elements (TLE) is presented in Section 2.1. Orbit propagation models such as the SGP, SGP4 and SGP8 for near-Earth orbits and SDP4 and SDP8 for deep-space orbits are mentioned, however, there are many others, see e.g. [54]. SGP is short for Simplified General Perturbations and SDP is short for Simplified Deep-space Perturbations. The word simplified comes from the fact that it is analytical methods that calculate the satellite position at any time instance, without having to numerically propagate the satellite position over small time steps from epoch and integrate these to get a solution, which is time-consuming [55].

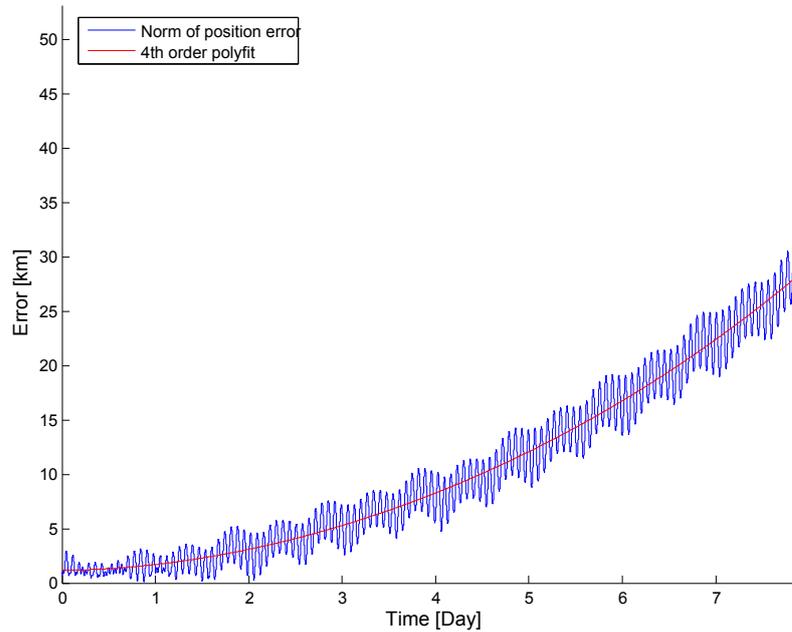
A commonly used orbit propagator for LEO satellites is the SGP4, which was already implemented in the Simulink simulation environment. The SGP4 takes more factors into account than the more simple two body Keplerian orbit propagator, which only considers the force of gravity from the Earth. The factors included are orbital decay from drag, but also effects caused by the Earth's oblateness and gravity from the Sun and the Moon [54].

### 4.2.2 The Simplified General Perturbations Satellite Orbit Model 4

The SGP4 is originally implemented in Fortran code and later wrapped into an S-function. This S-function is then compiled to a mex file in Matlab. It takes a TLE and a JD as inputs and outputs the satellite position in Cartesian coordinates (ECI) and satellite velocity (ECI), at the given JD.

TLEs can both be supplied by NORAD and the Phoenix GPS module on board the satellite. It is important to update the TLE at a regular basis to get higher accuracy of the predicted satellite position. Figure 4.3 shows how the difference between GPS position data from the Ørsted satellite and the SGP4 orbit propagator output increases, as we get further away from the time the TLE was last updated. The error in the GPS data is considered much smaller than the error induced by the SGP4.

After seven days the difference is up to 25 [km]. AAUSAT3 should not rely on TLEs from the phoenix GPS module, since it requires approximately 1 [W] of power to operate and it has to be turned on for about 12-15 [min] when cold-started, before it gets a fix on the GPS satellites and provides TLEs. During this period the GPS antenna must point away from the Earth towards the GPS satellites, thus the ADCS must be functioning correctly first and for that it needs a TLE. However, once the ADCS has been initialized with a TLE from



**Figure 4.3:** Normed difference in position between SGP4 and GPS data. Data gathered from the Ørsted satellite between 10. and 17. of February 2002 [2].

the ground station, it could be possible to autonomously get TLE updates by turning on the GPS module periodically (e.g. once a day or once a week).

As Figure 4.3 indicates the error between the SGP4 output and GPS measurements oscillates with a period of approximately one orbit and a magnitude of 5 [km]. This effect is considered negligible and has not been investigated further in this thesis.

It is important when evaluating the performance of the ADCS to introduce satellite position errors. Apart from the 25 [km] error, accumulated during seven days, a real time clock that is not correctly synchronized, will also give positions errors. 1 [s] corresponds to approximately 7.5 [km].

### 4.3 Ephemeris Models

The rotation of the Earth (rotation from ECI to ECEF) is a prerequisite to be able to determine the geomagnetic field at the satellite. It is also needed to calculate the amount of solar irradiance reflected by the Earth to the satellite (Earth albedo) and to calculate how the satellite should point in order to track reference targets on the Earth's surface.

The quaternion representing the ECI to ECEF rotation is calculated from a JD. This is done by using a fix-point, where the two reference frames were equal (Greenwich meridian transit of the equinox) and calculating how many revolutions the Earth has taken between the fix-point JD to the JD of interest. The last unfinished revolution indicates how much the ECEF is rotated around the z-axis relative to ECI.

Knowing the position of the Sun and the Moon in the ECI makes it possible to calculate gravitational disturbances from them. Again this is also needed to calculate the Earth albedo and to be able to determine if the satellite is in eclipse, which is necessary to emulate sun sensors.

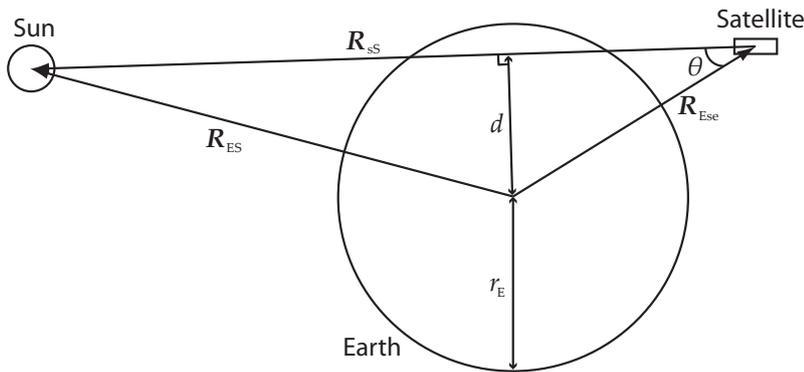
The calculations are placed in Matlab-files, making it easy to implement them in the on-board software for the satellite and again the only input required is a JD. The output is a vector from the Earth to the Sun or the Moon in the ECI.

## 4.4 Eclipse and Earth Albedo Models

Being capable of determining when the satellite is in eclipse and how much irradiance is reflected from the surface of the Earth to the satellite, gives the possibility of making more realistic sun sensor models and more accurate Sun vector measurements. Eclipse indication also helps the attitude estimation, because it is then possible to disregard sun sensor measurements during periods, where the satellite is in eclipse.

### 4.4.1 Eclipse Calculation

The satellite will potentially be in the Earth's shadow, when it is further away from the Sun than the Earth is. Figure 4.4 shows a situation where a satellite is in eclipse. If assuming that the Sun is a point source and the Earth has a uniform radius, it is possible to calculate if the satellite is in eclipse by determining if the distance  $d$  is longer than the radius of the Earth  $r_E$  [56].



**Figure 4.4:** Vectors, angle and distances used to calculate if a satellite is in eclipse.

The distance  $d$  can be determined by Eq. (4.1):

$$d = \sin(\theta) \cdot \|R_{Ese}\| \quad (4.1)$$

where:

$\theta$  is the angle between the vectors  $-\mathbf{R}_{Ese}$  (Earth to satellite vector) and  $\mathbf{R}_{sS}$  (satellite to sun vector).

The vector  $\mathbf{R}_{Ese}$  is determined by the orbit propagator and the vector  $\mathbf{R}_{sS}$  is determined by Eq. (4.2):

$$\mathbf{R}_{sS} = -\mathbf{R}_{Ese} + \mathbf{R}_{ES} \quad (4.2)$$

where:

$\mathbf{R}_{ES}$  is the vector from the center of the Earth to the center of the Sun, determined by the Ephemeris model.

The angle  $\theta$  is determined by taking the inverse cosine of the dot product between the normalised vectors.

$$\theta = \cos^{-1} \left( \frac{\mathbf{R}_{sS}}{\|\mathbf{R}_{sS}\|} \cdot \frac{-\mathbf{R}_{ES}}{\|\mathbf{R}_{ES}\|} \right) \quad (4.3)$$

Rewriting Eq. (4.1) and using (4.2) and (4.3) makes it possible to calculate  $d$  without using trigonometric functions.

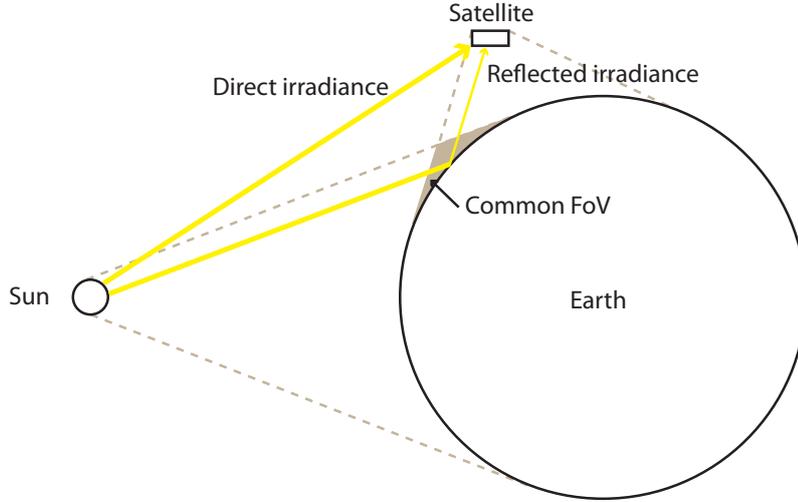
$$\begin{aligned} d &= \sin(\theta) \cdot \|\mathbf{R}_{Ese}\| = \sqrt{1 - \cos^2(\theta)} \cdot \|\mathbf{R}_{Ese}\| \\ &= \sqrt{1 - \left( \frac{-\mathbf{R}_{Ese} + \mathbf{R}_{ES}}{\|-\mathbf{R}_{Ese} + \mathbf{R}_{ES}\|} \cdot \frac{-\mathbf{R}_{ES}}{\|\mathbf{R}_{ES}\|} \right)^2} \cdot \|\mathbf{R}_{Ese}\| \end{aligned} \quad (4.4)$$

The satellite is then in eclipse if two criterias are fulfilled: The distance  $d$  must be smaller than the radius of the Earth  $r_E$  and the angle  $\theta$  must be smaller than  $90$  [deg].

#### 4.4.2 Earth Albedo Model

Some of the sun-light that hits the Earth is reflected. This is called Earth albedo and it contributes to the power induced in solar cells and currents in sun sensors, just like the direct sun-light does.

The Earth albedo model implemented in the simulation environment calculates the amount of Earth albedo arriving at an object in space (e.g. LEO satellite). This is done by first splitting up the surface of the Earth into 180 by 288 cells each having the size  $1$  [deg] latitude and  $1.25$  [deg] longitude. Then the amount of reflected irradiance is calculated for each cell. This value depends on the angle between the sun vector and the cell normal (incident angle) and the angle to the satellite and the cell normal. Only the surface area of the Earth within the common FoV of the satellite and the Sun contributes with reflected irradiance, which is illustrated in Figure 4.5.



**Figure 4.5:** Illustrative sketch of the reflection of solar irradiance in the common FoV of the satellite and the Sun, known as Earth albedo.

Eq. (4.5) gives the reflected irradiance  $E_c(\phi_g, \theta_g)$  for a grid cell, at the distance to the satellite [53].

$$E_c(\phi_g, \theta_g) = \begin{cases} \frac{\rho(\phi_g, \theta_g) E_{AM0} A_c(\phi_g) \hat{\mathbf{r}}_{sun}^T \hat{\mathbf{n}}_c \hat{\mathbf{r}}_{sat}^T \hat{\mathbf{n}}_c}{\pi \|\mathbf{r}_{sat}\|^2} \left[ \text{W/m}^2 \right], & \text{if } (\phi_g, \theta_g) \in (\mathbf{V}_{sun} \cap \mathbf{V}_{sat}) \\ 0, & \text{else} \end{cases} \quad (4.5)$$

Where:

$(\phi_g, \theta_g)$  defines the longitude and latitude of the center point of the grid cell.

$\rho(\phi_g, \theta_g)$  is the reflectivity of the grid cell.

$E_{AM0}$  is the incident irradiance from the Sun (considered a constant of 1367 [W/m<sup>2</sup>]).

$A_c(\phi_g)$  is the cell area.

$\hat{\mathbf{r}}_{sun}$  is the unit vector from the cell center to the Sun.

$\hat{\mathbf{n}}_c$  is the cell normal.

$\mathbf{r}_{sat}$  is the vector from the cell center to the satellite.

$\hat{\mathbf{r}}_{sat}$  is the unit vector from the cell center to the satellite.

$\mathbf{V}_{sun}$  is the set of sunlit cells.

$\mathbf{V}_{sat}$  is the set of cells visible from the satellite.

The implemented model needs a sun vector (ECEF), a satellite vector (ECEF) and outputs a reflectivity matrix containing the reflected irradiance  $E_c(\phi_g, \theta_g)$  values for each cell. Taking the sum of the reflected irradiance  $E_c(\phi_g, \theta_g)$  for all the cells gives the total amount of Earth albedo. However, this does not translate directly to an induced current in the solar panels and sun sensors, since the irradiance is not single directional. Eq. (4.6) gives the total amount of current measured by a sun sensor [53].

$$i_{meas} = \frac{i_{max}}{E_{cal}} \left( \left\{ E_{AM0} \hat{\mathbf{n}}_{ss}^T \hat{\mathbf{r}}_{sun} \right\}_0^\infty + \sum_{\mathbf{v}_{sun} \cap \mathbf{v}_{sat}} \left\{ E_c(\phi_g, \theta_g) \hat{\mathbf{n}}_{ss}^T \hat{\mathbf{r}}_g \right\}_0^\infty \right) [A] \quad (4.6)$$

Where:

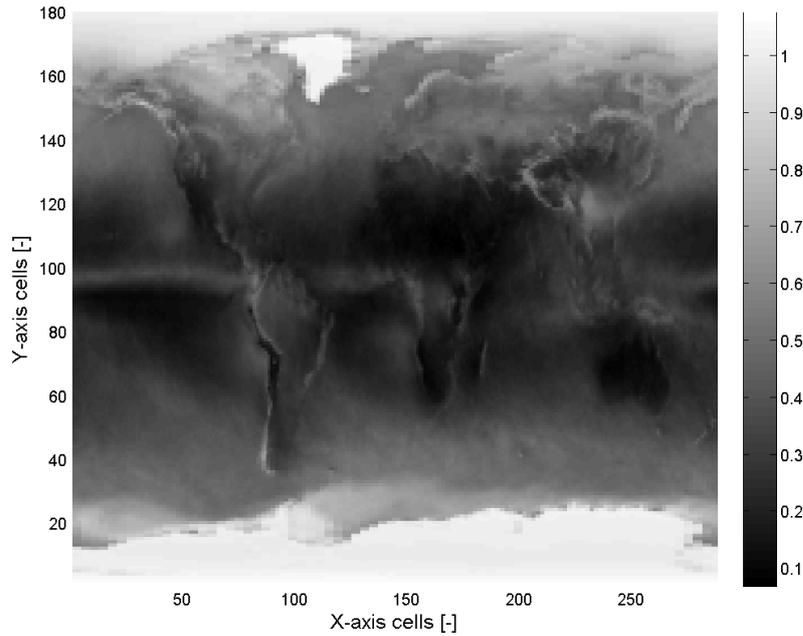
$i_{max}$  is the output current measured, when illuminated with an irradiance of  $E_{cal}$ .

$\hat{\mathbf{n}}_{ss}^T$  is the sun sensor normal.

$\hat{\mathbf{r}}_g$  is the unit vector in the direction of the grid cell.

0 to  $\infty$  comes from limits in the dot product between the unit vectors.

The Earth albedo model needs to know the reflectivity for each grid cell and this is given as Total Ozone Mapping Spectrometer (TOMS) reflectivity data [57], which has been measured by satellites. Dan Bhanderi has gathered TOMS reflectivity data on his homepage [53] for the years 1996 to 2005 and Figure 4.6 shows the mean reflectivity data from 2005.



**Figure 4.6:** TOMS mean reflectivity data from 2005.

Gathering data for a year, averages out the effect of a changing cloud cover. According to [53] there is very little impact in sun sensor currents when using daily reflectivity instead of annual means, hence the mean data is typically applied.

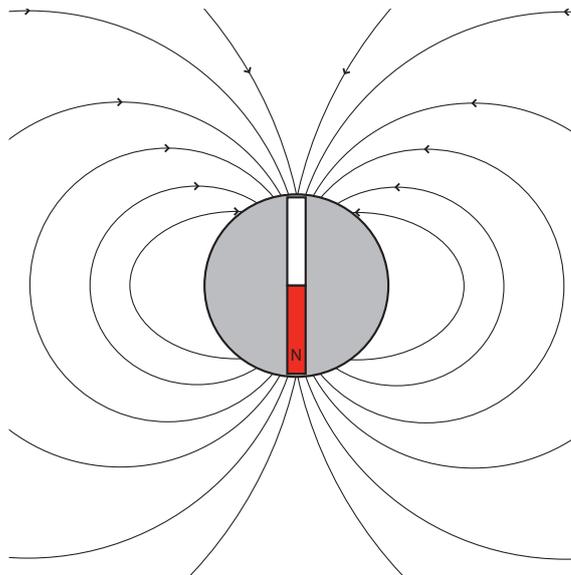
The 180x288 cells make the algorithm computationally heavy and should therefore be avoided in the on-board satellite software, or at least greatly reduced in complexity and accuracy.

## 4.5 Magnetic Field Models

A magnetic field model is essential for emulation of magnetometers and to be able to calculate the disturbance torque generated by the satellites magnetic residual. It also makes it possible to simulate how much torque the permanent magnet and the magnetorquers produce. Lastly a magnetic field model makes it possible to predict magnetometer measurements on board the satellite and predict how the magnetic field varies in the future.

### 4.5.1 The Earth's Magnetic Field

A combination of sources creates the magnetic field surrounding the Earth. More than 90% of this field is generated by the Earth's outer core and this is referred to as the Main Field, which varies slowly [58]. The Main Field can be described as that of a bar magnet with north and south poles residing within the Earth and field lines extending out into space [58]. Figure 4.7 illustrates the main field lines.



*Figure 4.7: The Earth's magnetic field modeled as a dipole magnet. N indicates the north pole of the dipole magnet (the Earth's magnetic south pole).*

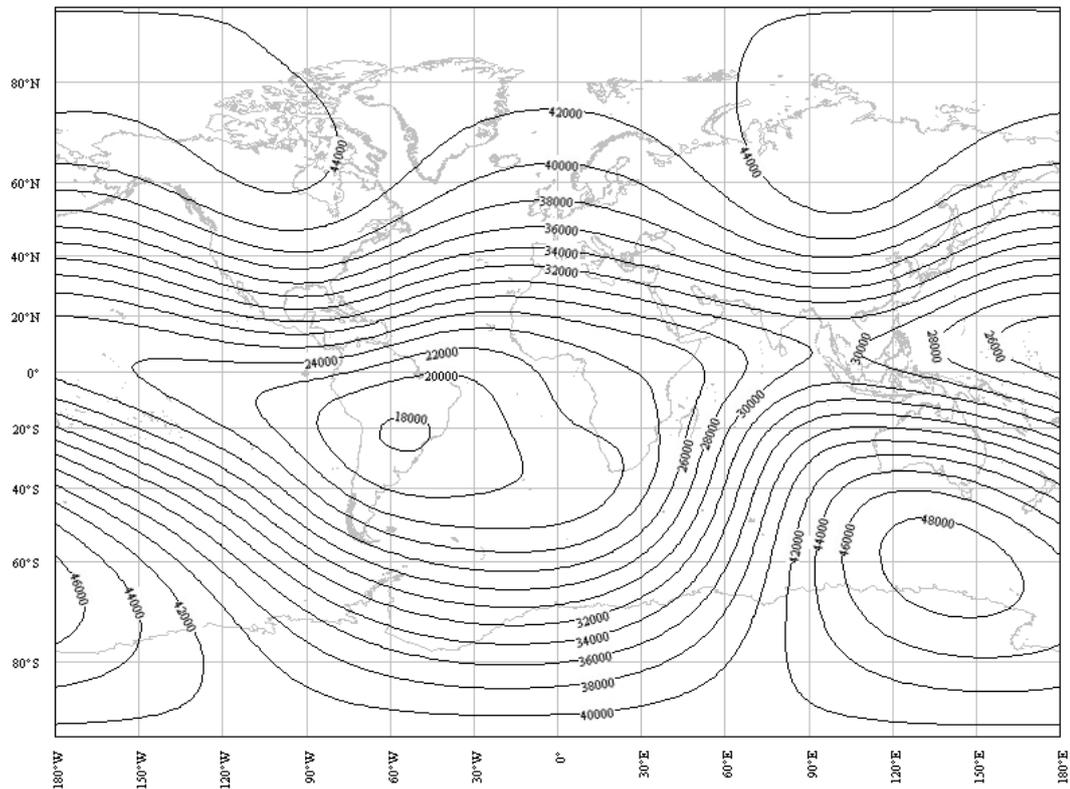
Currents flowing in the ionosphere, the magnetosphere and the Earth's crust cause variations in the intensity of the Earth's magnetic field that changes on a much shorter time scale than the Main Field.

The Main Field can be described by mathematical models such as the International Geomagnetic Reference Field (IGRF) released by the International Association of Geomagnetism and Aeronomy (IAGA) and the World Magnetic Model (WMM) released by the U.S. National Geophysical Data Center (NGDC) and the British Geological Survey (BGS) [58]. The IGRF has been implemented in simulation environment and will be discussed in the following subsection.

## 4.5.2 The International Geomagnetic Reference Field

The IGRF models the Main Field as the negative gradient of a scalar potential, which can be represented by a truncated spherical harmonic series, see [59] for further details. It is based on a collection of contributions from different magnetic field modelers and institutes who gathered magnetic field data from observatories and surveys around the world and from satellites like Champ and Ørsted.

The annual rate of change of the Main Field is also incorporated in the IGRF and the model is assumed to be linear over five-year intervals. The latest model valid for 2005-2010 is called the IGRF 10th generation model (IGRF10). Figure 4.8 shows contours generated by an implementation of the IGRF 10th generation model, that represents total magnetic field strength [60].



**Figure 4.8:** Total magnetic field strength contours generated by an IGRF10 model. The data is generated for the 1. of January 2010 in a orbit height of 630 [km].

The total magnetic field strength is approximately between 60 000 [nT] to 30 000 [nT] at the surface of the Earth and as Figure 4.8 indicates the field has decreased to approximately 48 000 [nT] to 18 000 [nT] in a height of 630 [km].

### 4.5.3 Magnetic Field Model Order and Uncertainties

From 2000 and onwards the IGRF model was extended from degree 10 to degree 13 (harmonics truncated to degree 13), to better reflect the high quality of available measurement data [59]. This gives a total of 210 coefficients and the secular variation coefficients are up to degree 8. The precision of the coefficient are 0.1 [ $nT$ ] for the Main Field and 0.1 [ $nT/year$ ] for the secular variations.

The Simulink implementation was of max degree 10, but has been upgraded to reflect all the coefficients of the IGRF10. A test of the implementation is presented in Appendix E.3.

Uncertainties are introduced from sources not modeled by the IGRF10. These sources are e.g. currents in the magnetosphere, the ionosphere and the Earth's crust. [59] suggests 10-20 [ $nT$ ] RMS as uncertainty in the predicted field, however, larger local uncertainties can occur. [4] cites that the field is predicted to within approximately 100 [ $nT$ ].

The test in Appendix E.3 also showed that an 8th order implementation of the IGRF10 would suffice for the on-board software on the satellite, which reduces the computational time needed. Calculation of the discrepancy between the 13th order model used in the "truth model" and an 8th order model, during 8 orbits, gave a standard deviation of 36.14 [ $nT$ ] and a maximum of approximately 100 [ $nT$ ].

This section has introduced errors coming from unmodeled sources in the IGRF and errors from differences in model order. The sensor will also give a discrepancy between the real and the measured magnetic field, which is treated in Section 4.6.

## 4.6 Sensor Modeling

Sensor models are introduced in order to provide realistic noisy measurement inputs to attitude estimators and controllers during simulations. These models are based on datasheets and estimates, see Section 3.1, and simulates magnetometers, sun sensors and gyroscopes. No actual tests have been performed on the chosen sensors and white Gaussian noise is therefore used and added to the simulated measurements. The real noise might be correlated, however, this has not been incorporated in the models yet. Each of the three sensor models, also shown in Figure 4.2, are described in the following.

### 4.6.1 Magnetometer Model

The "truth model" calculates the magnetic field vector in the ECI. It also calculates the quaternion that represents the rotation from the ECI to the SBRF, which makes it possible to calculate the magnetic field in the SBRF. This corresponds to the magnetic field seen by the 3-axis magnetometer and it is calculated in the first part of Figure 4.9.

The magnetic field vector is then added with a bias vector. The bias is caused by the presence of the permanent magnet and possible magnetic fields created by e.g. onboard electronics. However, most of the static bias can be compensated away in software before the satellite

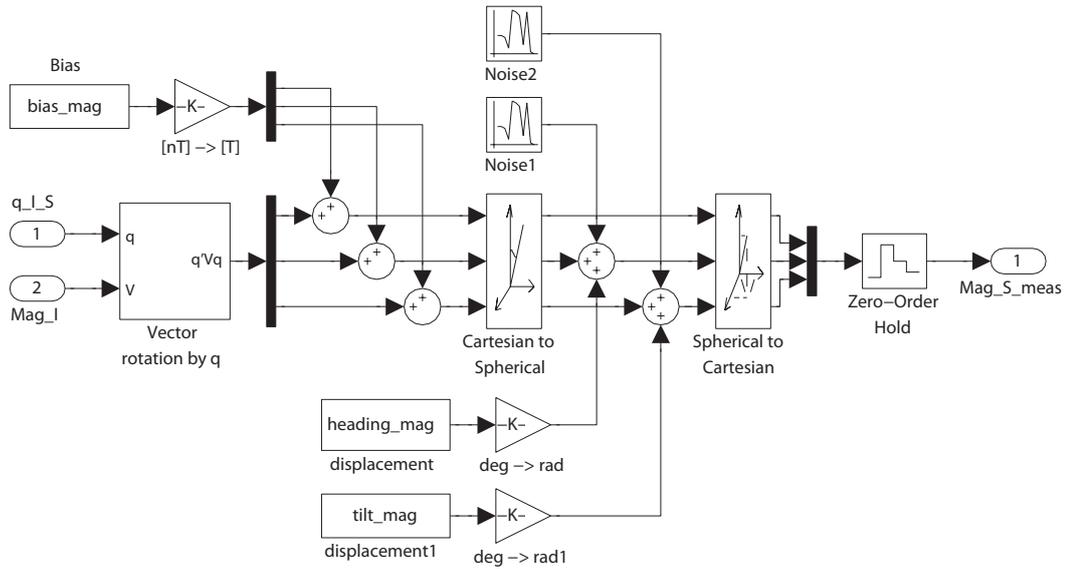


Figure 4.9: Matlab Simulink magnetometer model.

is launched into space and efforts should be made to reduce the magnetic residual of the satellite as much as possible.

The accuracy of the chosen magnetometer is, as stated in Subsection 3.1.3, within 3 [deg] RMS (maximum rating) and the magnetic field vector in the SBRF is therefore converted to spherical coordinates and the two angles  $\theta$  and  $\phi$  are added with Gaussian noise, having a standard deviation of 3 [deg]. Potential disturbances in the magnetic field are considered to be a part of the stated accuracy and temperature dependency is not included in the model, since the chosen sensor has internal temperature compensation.

In addition to the added noise it is also possible to add a constant angle, which simulates sensor displacement (bottom part of Figure 4.9). The heading angle corresponds to a rotation of the sensor on the PCB and the tilt angle corresponds to one of the sides of the sensor being tipped, due to e.g. excessive solder. It is not expected that these two angles will exceed  $\pm 1$  [deg] and if the angle is measurable it would furthermore be possible to compensate in software.

Lastly a zero order hold is added to simulate discrete sampling.

#### 4.6.2 Sun Sensor Model

The “truth model” calculates the position of the Sun in the ECI. It also calculates the quaternion that represents the rotation from the ECI to the SBRF, which makes it possible to calculate the position of the Sun in the SBRF. This corresponds to the Sun vector, measured by a perfect sun sensor and it is calculated in the first part of Figure 4.10.

An angular bias vector is then added to the Sun vector. This is done by converting the Sun vector to spherical coordinates as in the magnetometer model.

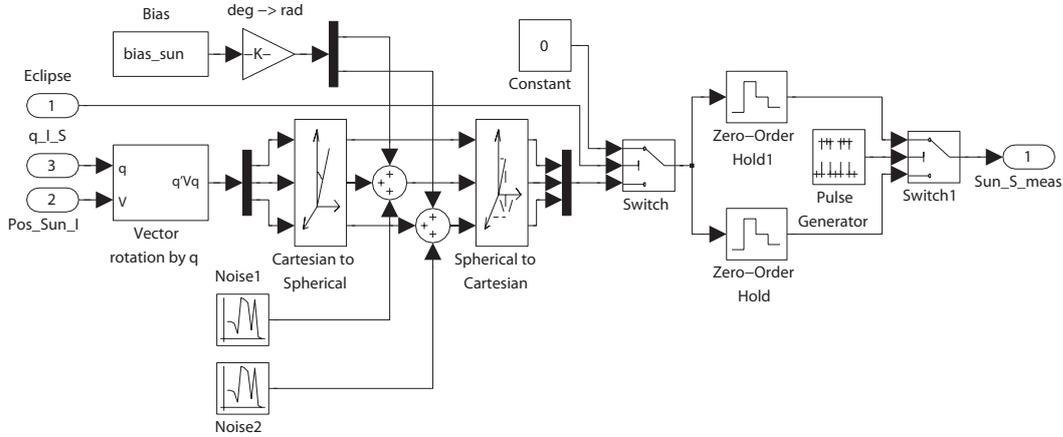


Figure 4.10: Matlab Simulink sun sensor model.

The accuracy of the sun sensor is, as stated in Subsection 3.1.3, guestimated to be within 10 [deg] and the two angles  $\theta$  and  $\phi$  are therefore added with Gaussian noise, having a standard deviation of 3.33 [deg]. This fairly large inaccuracy is used since the Earth albedo effect has not yet been incorporated in the sun sensor model. However, it is expected that a less computationally heavy Earth albedo implementation will be part of the onboard satellite software, which would make it possible to compensate for Earth albedo.

Temperature sensors will be added to all side-panels and introduces the possibility of making temperature compensation. However, temperature compensation is not as important if the Sun vector is calculated by each side-panel individually by multiple photodiodes, see Subsection 3.1.3. Temperature dependency is therefore not added to the sun sensor model.

The output of the sun sensor model is a zero vector during eclipse.

The two zero order hold blocks and the pulse generator makes it possible to both control the discrete sampling frequency, but also to take bursts of samples. This could e.g. be 5 samples, sampled at 10 [Hz], gathered every 100 [s]. However, the bursted sampling is not utilized in this thesis, but could be used on the satellite to save computation time and power, after the attitude estimator has converged or when a lower accuracy is acceptable.

### 4.6.3 Gyroscope Model

The “truth model” directly calculates the satellite angular velocity around the SBRF axes. This corresponds to the angular velocity measured by a perfect three axis gyroscope.

The angular velocity vector is then added with a bias vector, see Figure 4.11. The bias is caused by sensor offset errors due to e.g. temperature fluctuations. However, the chosen sensor does have a built in temperature sensor, providing the possibility for software temperature compensation.

Gaussian noise is also added to each of the vector elements. This noise is expected to have a standard deviation of 0.2 [deg/s]. A more elaborate model could be made, which

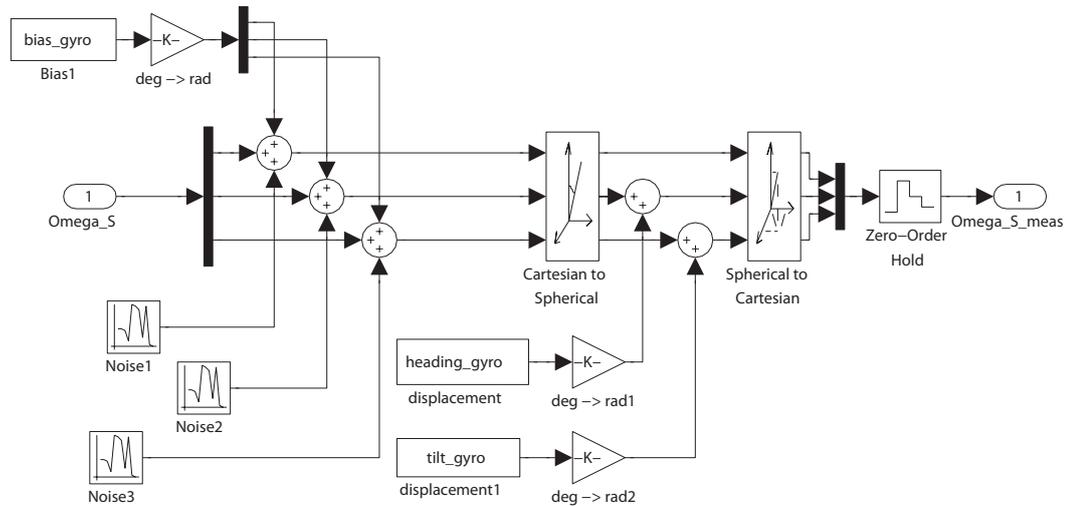


Figure 4.11: Matlab Simulink gyroscope model.

takes potential non-linearities and cross axis sensitivity into consideration, see Subsection 3.1.3. A drift term is not included in the model, because the chosen gyroscope measures the angular velocity directly using the Coriolis effect and therefore has no drift term.

As with the magnetometer model, it is also possible to add a constant angle, which simulates sensor displacement.

Lastly a zero order hold is added to simulate discrete sampling.

## 4.7 Actuator Modeling

Actuator models are introduced in order to translate control outputs into torques acting on the satellite during simulations. Translating control outputs into torques is also usable in attitude estimators as the torque is required in the satellite equations of motion for propagation of the states and the models will therefore be part of the onboard software on the satellite. A permanent magnet and magnetorquer coils, including driver circuits, were designed for AAUSAT3 in Section 3.3 and 3.4 respectively. These designs are translated into Matlab Simulink models in this section and are also shown in Figure 4.2.

### 4.7.1 Permanent Magnet Model

The permanent magnet creates a dipole magnetic moment. This is represented by a vector in the SBRF, which points in the direction from the magnets south pole to its north pole. The torque exerted on the satellite is then calculated as the cross product between this magnetic moment and the Earth's magnetic field given in the SBRF. The permanent magnet model, shown in Figure 4.12, therefore requires the Earth's magnetic field in the ECI and the quaternion representing the rotation from the ECI to the SBRF as inputs.

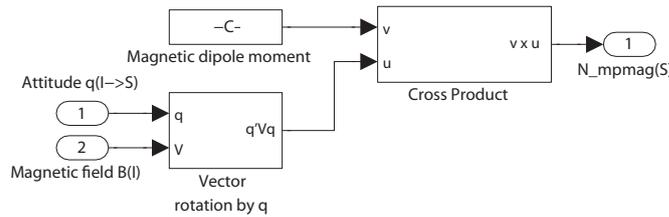


Figure 4.12: Matlab Simulink permanent magnet model.

### 4.7.2 Magnetorquer Model

The magnetorquer model consists of three magnetorquer coils, each having a normal pointing in the direction of the x-, y- and z-axis respectively. These coil models output the torque they exert on the satellite, which is added to form the total torque given in the SBRF. In order to calculate this torque it is necessary to translate the input PWM vector, having the desired dutycycle for each coil driver, into corresponding voltages applied to the coils, as shown in the first part of Figure 4.13.

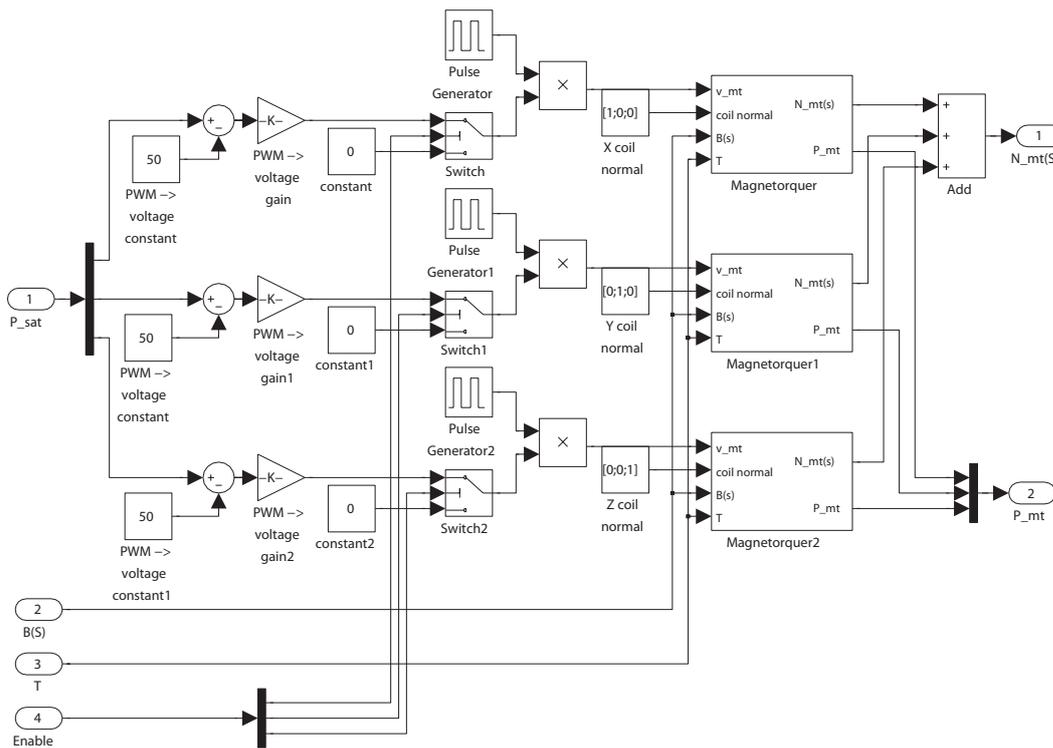


Figure 4.13: Matlab Simulink models of three magnetorquer coils and drivers.

The model then provides the possibility to enable/disable each coil individually and a pulse generator simulates having a duty cycle on the actuation, which is required for periodic actuation and measurement of the Earth’s magnetic field vector.

The content of each of the three magnetorquer coil models are shown in Figure 4.14.

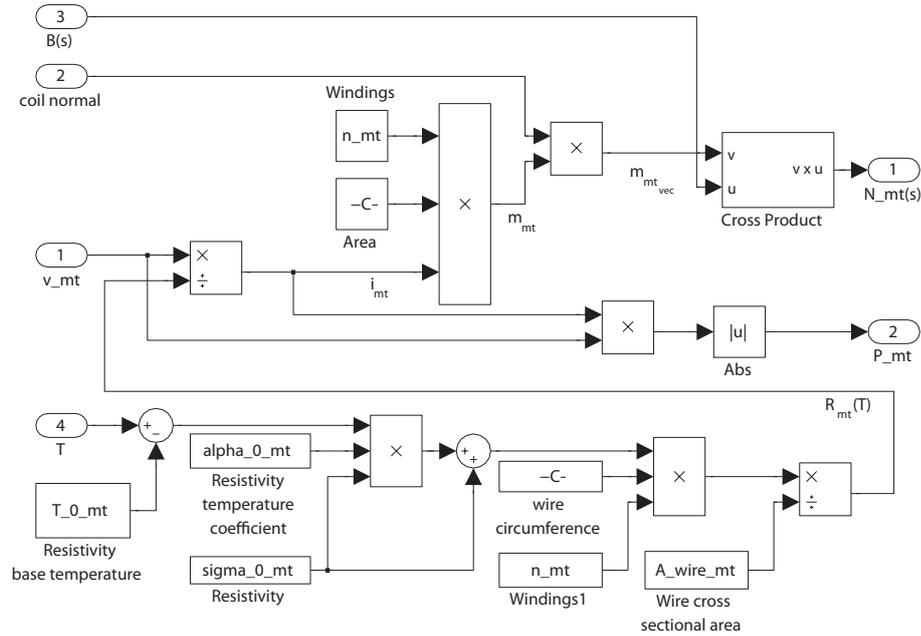


Figure 4.14: Matlab Simulink magnetorquer coil model.

This model first calculates a wire resistance  $R_{mt}(T_{mt})$  based on the magnetorquer coil temperature  $T_{mt}$  in Kelvin, which is given as input. According to [36] the wire resistance can be calculated as

$$R_{mt}(T_{mt}) = \frac{l_{wire}\sigma_{mt}}{A_{wire}} = \frac{l_{wire}\sigma_{0,mt}(1 + \alpha_{0,mt}(T_{mt} - 293))}{A_{wire}} \quad [\Omega] \quad (4.7)$$

where:

$l_{wire}$  is the total length of the wire (windings multiplied by wire circumference).

$A_{wire}$  is the wire cross-sectional area.

$\sigma_{mt}$  is the copper resistivity.

$\sigma_{0,mt}$  is the copper resistivity at 293 [K].

$\alpha_{0,mt}$  is a copper resistivity temperature coefficient.

The input voltage is then divided by the calculated resistance, giving the current in the wire. Multiplying this current with the area enclosed by the wire and the number of windings, gives the magnitude of the magnetic moment generated by the magnetorquer coil. This is then multiplied with the coils normal to create a vector, which gives the torque exerted on the satellite, when the cross-product is taken with the Earth's magnetic field in the SBRF.

Coil inductance introduces a pole with a cutoff frequency at approximately 1500 [Hz], with the design parameters found in Section 3.4. This is far from the 10 [Hz] periodic actuation and measurement and the inductance is therefore considered negligible in the model (coil impedance is negligible at low frequencies).

Lastly power is given as output in order to evaluate performance of controller algorithms.

## 4.8 Summary

This chapter presented the content of the Matlab Simulink simulation environment made for AAUSAT3, which builds on previous work done by other students at Aalborg University. Among many useful Simulink blocks and Matlab-files it first of all contains the satellite equations of motion and the identified disturbances presented in Chapter 2. Secondly it contains an orbit propagator (SGP4), ephemeris models, eclipse and Earth albedo calculation and finally a geomagnetic field model (IGRF10). These functionalities provide the so called "truth model" of the satellite.

Additionally sensor and actuator models have been provided, simulating gyroscopes, sun sensors, magnetometers, permanent magnets, magnetorquer coils and driver circuits.

New simulation models can easily be made, by using the Simulink library containing all the above described functionalities in blocks, including all developed attitude determination and control algorithms. It has been of high importance to keep a high degree of reusability of the library, with the possible intend of reusing it for other satellite projects.

# Attitude Estimation Techniques

This chapter briefly describes some estimation techniques from which the Singular Value Decomposition (SVD)-method solving Wahba's problem and the Unscented Kalman Filter (UKF) are chosen for AAUSAT3. A more thorough description of the SVD-method, the Extended Kalman Filter (EKF) and the UKF are given, with focus on describing the equations and procedures involved in the algorithms. The derivation of the equations are not treated in this thesis.

## 5.1 Introduction to Attitude Estimation

In general, satellite attitude estimation can be seen as a two stage process, where the first stage is to estimate the attitude of the satellite using models of the given system and the second stage is to improve the estimate using sensor measurements from e.g. gyros and reference observations like magnetic field measurements [61].

There are numerous methods to perform attitude estimation, which in general can be separated into two groups; deterministic point by point solutions and filters [4].

Deterministic point by point solutions estimates the attitude from at least two (preferably more) reference vectors observed at a single point in time. The point estimation algorithms are fast and relatively simple, but the requirement of at least two reference vectors constrain the use of these algorithms on satellites with an inadequate number of sensors. A list of different point estimation algorithms and a more thorough description of these can be found in [4, 61]. However, the SVD-method, used to solve Wahba's problem, will be treated in Section 5.2.

The recursive stochastic filters can estimate the attitude more accurate as they also estimate the attitude by combining dynamic and/or kinematic models with measurements obtained by several sensors [4]. As the recursive stochastic filters includes dynamic and/or kinematic models they also have the means of propagating the attitude during eclipse and does not necessarily require two or more reference observations like the deterministic methods. One

of the most well known recursive stochastic filter is the Kalman filter, which is referred to as;

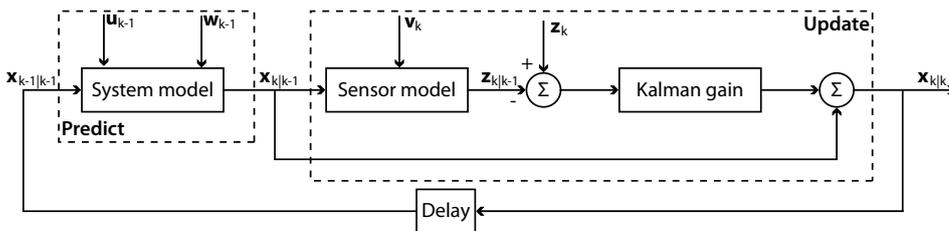
*"... the greatest achievement in estimation theory of the 20th. century." [62]*

The general Kalman filter is a commonly used estimator for applications such as continuous manufacturing processes, aircrafts, ships and satellites [62].

The Kalman filter estimation process is based on two steps:

- **Prediction step:** The previous state estimate  $\mathbf{x}_{k-1|k-1}$  is propagated through the system model to obtain the a priori state estimate of the next sample instance  $\mathbf{x}_{k|k-1}$ .
- **Update step:** The a posteriori state estimate for the current sample instance  $\mathbf{x}_{k|k}$  is calculated as a weighted linear combination of the before mentioned a priori state estimate and a correction factor based on the measurements for the current sample instance  $\mathbf{z}_k$ .

A block diagram of the Kalman filter is illustrated in Figure 5.1, where the predict and update steps are indicated with dashed boxes. As mentioned in the above, the predict step includes the system model, which in this case is influenced by an exogenous input  $\mathbf{u}_{k-1}$  and a process noise  $\mathbf{w}_{k-1}$ . The update step includes the sensor model, which is used to obtain expected measurements  $\mathbf{z}_{k|k-1}$  for the a priori state estimate. In this case the sensor model is influenced by observation noise  $\mathbf{v}_k$ . The real measurement  $\mathbf{z}_k$  is then subtracted from the predicted measurement  $\mathbf{z}_{k|k-1}$  forming residuals, which are multiplied by the Kalman gain to form an update of the a priori state.



*Figure 5.1: Block diagram of the Kalman filter.*

The Kalman filter addresses estimation problems for dynamic systems, which are well described by linear models. For nonlinear systems, such as e.g. a satellite (see Section 2.4), other approaches must be utilized. The following presents some practical approaches for attitude estimation using nonlinear models [62]:

- **Linearization:** By linearizing the nonlinear model it is possible to utilize the Kalman filter on the linearized model. This approach might be sufficient for estimation problems, which are close to linear.

- **Extended Kalman Filtering:** The EKF is a commonly used estimator for problems, which are close to linear. The EKF evaluates the partial derivatives at the estimated state vector value in order to calculate the error covariance matrix and the Kalman gain, while the estimated state itself is propagated using the full nonlinear model.
- **Sampling methods:** Sampling methods approximate the nonlinear transformations of means and covariances using statistical sampling theory. These methods includes, unlike linearization methods, the nonlinear effects on the mean and covariance, however, most of the methods have a considerably higher computational complexity than the Kalman filter.
- **Unscented Kalman Filtering:** The UKF is a special case of a sampling method. The UKF is more computational heavy than the Kalman filter and the EKF, but it handles highly nonlinear systems better.

The above mentioned estimation approaches is considered the most practical, however, other methods exist [62]. The UKF approach for satellite attitude estimation has been used by e.g. [63, 64, 65, 66, 67]. This approach is characterized by a set of sample (sigma) points, which are used to approximate a Gaussian probability distribution, whereas the well known EKF is distinguished by the fact that it uses a linearization of the nonlinear model equations in the working point. The computational load of the UKF is higher than that of the EKF, because all sigma points have to be propagated, but the UKF does not rely on the derivation of Jacobian matrices. Furthermore, the error of the UKF is expected to be lower than that of the EKF, since the sigma points provide a way of estimating the a posteriori mean and covariance accurately to the second order for any nonlinearity [68]. This thesis will focus on the UKF approach, which is described in Section 5.4, however, the EKF approach is first described in Section 5.3, in order to show the differences between the two filters. The descriptions are primarily based on [62, 68, 69].

## 5.2 Wahba's Problem and the SVD-Method

The attitude of a satellite can be determined by unit vector observations, coming from e.g. sun sensors or magnetic field sensors. If two or more vector observations are available in two different coordinate systems, then it is possible to find a special orthogonal matrix  $\underline{\mathbf{A}}$ , representing the rotation between the two coordinate systems, by solving what is known as Wahba's problem. Wahba's problem is a least square optimization problem, where a cost function  $J(\underline{\mathbf{A}})$  is minimized [70]:

$$J(\underline{\mathbf{A}}) = \sum_{i=1}^m a_i \|\mathbf{b}_i - \underline{\mathbf{A}}\mathbf{r}_i\|^2 \quad (5.1)$$

Eq. (5.1) defines the cost function to be minimized, where  $\mathbf{b}$  and  $\mathbf{r}$  are vector observations in two different coordinate systems. The non-negative weight  $a$  can be chosen to be the inverse variance  $\sigma^{-2}$  of the measurement noise [71]. Eq. (5.1) can also be written in a more

convenient form [71, 72]:

$$J(\underline{\mathbf{A}}) = \sum_{i=1}^m a_i - \text{trace}(\underline{\mathbf{A}} \underline{\mathbf{B}}^T) \quad (5.2)$$

where  $\underline{\mathbf{B}} = \sum_{i=1}^m a_i \mathbf{b}_i \mathbf{r}_i^T$ . This reduces the problem to a question of maximizing  $\text{trace}(\underline{\mathbf{A}} \underline{\mathbf{B}}^T)$ .

There are many ways of solving this problem. This thesis focuses on the SVD-method, which is faster than the q-method for two vector observations and considered more robust than other faster methods, such as FOAM and ESOQ [71].

The matrix  $\underline{\mathbf{B}}$  has the singular value decomposition (Hence the name SVD-method):

$$\underline{\mathbf{B}} = \underline{\mathbf{U}} \underline{\mathbf{\Sigma}} \underline{\mathbf{V}}^T = \underline{\mathbf{U}} \text{diag} [\Sigma_{11} \ \Sigma_{22} \ \Sigma_{33}] \underline{\mathbf{V}}^T \quad (5.3)$$

where *diag* refers to a square matrix with zeros outside the main diagonal. The matrices  $\underline{\mathbf{U}}$  and  $\underline{\mathbf{V}}$  are orthogonal and the singular values obey  $\Sigma_{11} \geq \Sigma_{22} \geq \Sigma_{33} \geq 0$  [71]. Then

$$\text{trace}(\underline{\mathbf{A}} \underline{\mathbf{B}}^T) = \text{trace}(\underline{\mathbf{U}}^T \underline{\mathbf{A}} \underline{\mathbf{V}} \text{diag} [\Sigma_{11} \ \Sigma_{22} \ \Sigma_{33}]) \quad (5.4)$$

The maximized trace is obtained by

$$\underline{\mathbf{U}}^T \underline{\mathbf{A}}_{opt} \underline{\mathbf{V}} = \text{diag} [1 \ 1 \ \det(\underline{\mathbf{U}}) \det(\underline{\mathbf{V}})] \quad (5.5)$$

which is consistent with the constraint  $\det(\underline{\mathbf{A}}) = 1$  [71]. This gives the optimal rotation matrix  $\underline{\mathbf{A}}_{opt}$  defined in Eq. (5.6).

$$\underline{\mathbf{A}}_{opt} = \underline{\mathbf{U}} \text{diag} [1 \ 1 \ \det(\underline{\mathbf{U}}) \det(\underline{\mathbf{V}})] \underline{\mathbf{V}}^T \quad (5.6)$$

It is possible to examine how good the estimate  $\underline{\mathbf{A}}_{opt}$  is, by calculating the covariance matrix  $\underline{\mathbf{P}}$  (of the rotation angle error vector). According to [71],  $\underline{\mathbf{P}}$  can be calculated as

$$\underline{\mathbf{P}} = \underline{\mathbf{U}} \text{diag} [(s_2 + s_3)^{-1} \ (s_3 + s_1)^{-1} \ (s_1 + s_2)^{-1}] \underline{\mathbf{U}}^T \quad (5.7)$$

with the three definitions  $s_1 \equiv \Sigma_{11}$ ,  $s_2 \equiv \Sigma_{22}$  and  $s_3 \equiv \det(\underline{\mathbf{U}}) \det(\underline{\mathbf{V}}) \Sigma_{33}$ .  $\underline{\mathbf{P}}$  becomes infinite if the attitude is unobservable. If the satellite only has two sensors (e.g. sun and magnetic field sensor), then the SVD-method fails when the satellite is in eclipse and when the two observations are parallel. This makes the SVD-method unsuitable for continuous attitude estimation. However, it can be used as a sanity-check for filters like the EKF and UKF.

### 5.3 Extended Kalman Filter

For a nonlinear dynamic system given by a state space model (see Eq. (5.8) and (5.9)) with a nonlinear system model  $f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$  and a nonlinear sensor model  $h(\mathbf{x}_k)$ , it is desired to estimate the state of the system.

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \quad (5.8)$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (5.9)$$

Where:

- $\mathbf{x}_k$  is the state vector at time  $k$ .
- $\mathbf{z}_k$  is the measurement vector at time  $k$ .
- $\mathbf{u}_{k-1}$  is observed exogenous inputs.
- $\mathbf{w}_{k-1}$  is the process noise.
- $\mathbf{v}_k$  is the observation noise.

It is assumed that the process noise  $\mathbf{w}_{k-1}$  and the observation noise  $\mathbf{v}_k$  are independent zero mean white Gaussian noises, with a process error covariance matrix  $\underline{\mathbf{Q}}_{k-1}$  and a measurement noise covariance matrix  $\underline{\mathbf{R}}_k$ , respectively.

As for the Kalman filter it is desired to estimate the state of the given dynamical system, using all previous information up to the present time  $k$ . For this, a prediction of the state and error covariance are calculated using previous information up to the time  $k-1$ . The predicted state and error covariance are then updated using information obtained at the present time  $k$  (see Figure 5.1).

### 5.3.1 Predict step

The a priori state estimate  $\mathbf{x}_{k|k-1}$  is calculated using the nonlinear system model, i.e. the previous state  $\mathbf{x}_{k-1|k-1}$  is propagated through the nonlinear system model together with the observed exogenous inputs.

$$\mathbf{x}_{k|k-1} = f(\mathbf{x}_{k-1|k-1}, \mathbf{u}_{k-1}) \quad (5.10)$$

The solution for the above Ordinary Differential Equation (ODE) can be approximated by iterative numerical methods like e.g. Euler and Runge Kutta.

An a priori estimate of the error covariance matrix  $\underline{\mathbf{P}}_{k|k-1}$  is calculated using the previous error covariance matrix  $\underline{\mathbf{P}}_{k-1|k-1}$ .

$$\underline{\mathbf{P}}_{k|k-1} = \underline{\Phi}_{k-1|k-1} \underline{\mathbf{P}}_{k-1|k-1} \underline{\Phi}_{k-1|k-1}^T + \underline{\mathbf{Q}}_{k-1} \quad (5.11)$$

Where:

$\underline{\Phi}_{k-1|k-1}$  is the Jacobian matrix (linearized process function).

The Jacobian matrix contains the first order partial derivatives of the process function  $f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ .

$$\underline{\Phi}_{k-1|k-1} \approx \left. \frac{\partial f_{k-1}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_{k-1|k-1}} \quad (5.12)$$

### 5.3.2 Update step

In the update step the a priori state estimate  $\mathbf{x}_{k|k-1}$  is used to calculate the transformed measurements  $\mathbf{z}_{k|k-1}$  by propagating  $\mathbf{x}_{k|k-1}$  through the sensor model.

$$\mathbf{z}_{k|k-1} = h(\mathbf{x}_{k|k-1}) \quad (5.13)$$

The a priori state estimate is updated using measurements  $\mathbf{z}_k$  obtained at present time  $k$  and the Kalman gain  $\underline{\mathbf{K}}_k$ . The updated state estimate is called the a posteriori state estimate  $\mathbf{x}_{k|k}$ .

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \underline{\mathbf{K}}_k (\mathbf{z}_k - \mathbf{z}_{k|k-1}) \quad (5.14)$$

The Kalman gain  $\underline{\mathbf{K}}_k$  minimizes the the mean squared error between the predicted and measured measurements and is given as

$$\underline{\mathbf{K}}_k = \underline{\mathbf{P}}_{k|k-1} \underline{\mathbf{H}}_k^T (\underline{\mathbf{H}}_k \underline{\mathbf{P}}_{k|k-1} \underline{\mathbf{H}}_k^T + \underline{\mathbf{R}}_k)^{-1} \quad (5.15)$$

The Jacobian matrix  $\underline{\mathbf{H}}_k$  contains the first order partial derivatives of the measurement function  $h(\mathbf{x}_k)$ .

$$\underline{\mathbf{H}}_k \approx \left. \frac{\partial h_k}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_{k|k-1}} \quad (5.16)$$

Finally the estimated error covariance matrix  $\underline{\mathbf{P}}_{k|k-1}$  is updated to obtain the a posteriori estimate of the error covariance  $\underline{\mathbf{P}}_{k|k}$ .

$$\underline{\mathbf{P}}_{k|k} = (\mathbf{1} - \underline{\mathbf{K}}_k \underline{\mathbf{H}}_k) \underline{\mathbf{P}}_{k|k-1} \quad (5.17)$$

## 5.4 Unscented Kalman Filter

The UKF is based on a sigma point sampling method called Unscented TransForm (UTF). Sigma points are a structured set of sample points selected in such way, that the sigma points gives an adequate coverage of the input and output probability distribution.

There are different sampling strategies for the UTF. Some of these are summarized here [62]:

- **Simplex:** The Simplex UTF has the smallest number of samples with a sample size of only  $L + 2$ , or  $L + 1$  if the zeroth sigma point is ignored, where  $L$  is the dimension of the state vector. The small number of samples reduces the computation time of the UKF, however, the statistical performance may not be as good as the following strategies with more samples.
- **Spherical:** The Spherical UTF is a modification of the Simplex UTF, which is design to improve the numerical stability by rearranging the sigma points. The Spherical UTF has the same sample size as the Simplex UTF [69].

- **Symmetric:** The Symmetric UTF has a symmetric sampling about the mean. The sample size of the Symmetric UTF is  $2L + 1$  or  $2L$  because it is possible to drop the zeroth sigma point.
- **Scaled:** The Scaled UTF is a modification of the Symmetric UTF. The Scaled UTF has the same sample size as the Symmetric UTF, but adds the possibility of adjusting samples and weights. This makes the Scaled UTF more robust to higher order nonlinearities beyond second order [62].

For this thesis the Scaled UTF will be utilized due to the extra adjustable scaling parameters, which is advantageous, if the performance of the UKF should be tuned. For the Scaled UTF the sigma points  $(\chi_{k-1|k-1})_i$  are given by

$$(\chi_{k-1|k-1})_0 = \mathbf{x}_{k-1|k-1} \quad (5.18)$$

$$(\chi_{k-1|k-1})_i = \mathbf{x}_{k-1|k-1} + \left( \sqrt{(L + \lambda) \mathbf{P}_{k-1|k-1}} \right)_i, \text{ for } i = 1, \dots, L \quad (5.19)$$

$$(\chi_{k-1|k-1})_i = \mathbf{x}_{k-1|k-1} - \left( \sqrt{(L + \lambda) \mathbf{P}_{k-1|k-1}} \right)_i, \text{ for } i = L + 1, \dots, 2L \quad (5.20)$$

$\mathbf{x}_{k-1|k-1}$  is the state vector at time  $k - 1$  and  $\lambda = \alpha^2(L + \kappa) - L$  is a scaling parameter. The constant  $\alpha$  normally has a small positive value between  $10^{-4} \leq \alpha \leq 1$  and it determines the spread of the sigma points, while the constant  $\kappa$  is normally set to  $3 - L$  [68]. However, for  $\kappa < 0$  the resulting error covariance matrix, under the square root, cannot be guaranteed to be positive definite and  $\kappa$  is therefore often set to 0 [62]. The property of  $\mathbf{P}_{k-1|k-1}$  being positive definite must be satisfied in order to solve  $\sqrt{(L + \lambda) \mathbf{P}_{k-1|k-1}}$  using Cholesky factorization [69]. Possible Cholesky factors could be lower triangular matrix, upper triangular matrix, symmetric (using SVD) or the eigenvector Cholesky factor.

As for the Kalman filter and EKF the procedure for calculating the a posteriori state estimate  $\mathbf{x}_{k|k}$  for UKF consists of a predict and update step (see Figure 5.1).

### 5.4.1 Predict step

The predicted sigma points are calculated by propagating the obtained sigma points in the sigma point matrix  $\underline{\chi}_{k-1|k-1}$ , which are calculated using Eq. (5.18), (5.19) and (5.20), through the nonlinear system model (see Eq. (5.8)).

$$(\chi_{k|k-1})_i = f\left((\chi_{k-1|k-1})_i, \mathbf{u}_{k-1}\right), \text{ for } i = 1, 2, \dots, 2L \quad (5.21)$$

The a priori state estimate  $\mathbf{x}_{k|k-1}$  and the a priori error covariance matrix  $\mathbf{P}_{k|k-1}$  are then calculated using the predicted sigma points.

$$\mathbf{x}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(m)} (\boldsymbol{\chi}_{k|k-1})_i \quad (5.22)$$

$$\underline{\mathbf{P}}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(c)} \left( (\boldsymbol{\chi}_{k|k-1})_i - \mathbf{x}_{k|k-1} \right) \left( (\boldsymbol{\chi}_{k|k-1})_i - \mathbf{x}_{k|k-1} \right)^T + \underline{\mathbf{Q}}_k \quad (5.23)$$

The matrix  $\underline{\mathbf{Q}}_k$  is the process noise covariance. The weights  $W_i^{(m)}$  and  $W_i^{(c)}$ , used to calculate the a priori state estimate and the a priori error covariance matrix, are defined in Eq. (5.24), (5.25) and (5.26).

$$W_0^{(m)} = \frac{\lambda}{L + \lambda} \quad (5.24)$$

$$W_0^{(c)} = \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta) \quad (5.25)$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(L + \lambda)}, \text{ for } i = 1, \dots, 2L \quad (5.26)$$

The scaling parameter  $\beta$  incorporates prior knowledge of the distribution of the state vector  $\mathbf{x}_k$ . According to [68], the optimal value of  $\beta$  for a Gaussian distribution is 2.

As Eq. (5.24), (5.25) and (5.26) indicates, that the weights are independent of time and hence it is possible to calculate them in an initialization step.

### 5.4.2 Update step

The predicted sigma points  $(\boldsymbol{\chi}_{k|k-1})_i$  are propagated through the sensor model (see Eq. (5.9)) in order to obtain the transformed sigma points  $(\mathbf{Z}_{k|k-1})_i$  (predicted measurements).

$$(\mathbf{Z}_{k|k-1})_i = h\left((\boldsymbol{\chi}_{k|k-1})_i\right), \text{ for } i = 0, 1, \dots, 2L \quad (5.27)$$

The transformed measurements  $\mathbf{z}_{k|k-1}$  can then be calculated as

$$\mathbf{z}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(m)} (\mathbf{Z}_{k|k-1})_i \quad (5.28)$$

The a posteriori state estimate  $\mathbf{x}_{k|k}$  is calculated using the measurement vector  $\mathbf{z}_k$ , containing measurements obtained at time  $k$ .

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \underline{\mathbf{K}}_k (\mathbf{z}_k - \mathbf{z}_{k|k-1}) \quad (5.29)$$

The Kalman gain  $\underline{\mathbf{K}}_k$  is calculated as

$$\underline{\mathbf{K}}_k = \underline{\mathbf{P}}_{\underline{x}_k \underline{z}_k} \underline{\mathbf{P}}_{\underline{z}_k \underline{z}_k}^{-1} \quad (5.30)$$

where the measurement covariance matrix  $\underline{\mathbf{P}}_{\underline{z}_k \underline{z}_k}$  and the state-measurement cross-covariance matrix  $\underline{\mathbf{P}}_{\underline{x}_k \underline{z}_k}$  are calculated as in Eq. (5.31) and (5.32) respectively, where the matrix  $\mathbf{R}_k$  is the measurement noise covariance matrix.

$$\underline{\mathbf{P}}_{\underline{z}_k \underline{z}_k} = \sum_{i=0}^{2L} W_i^{(c)} \left( (\mathbf{Z}_{k|k-1})_i - \mathbf{z}_{k|k-1} \right) \left( (\mathbf{Z}_{k|k-1})_i - \mathbf{z}_{k|k-1} \right)^T + \mathbf{R}_k \quad (5.31)$$

$$\underline{\mathbf{P}}_{\underline{x}_k \underline{z}_k} = \sum_{i=0}^{2L} W_i^{(c)} \left( (\mathbf{X}_{k|k-1})_i - \mathbf{x}_{k|k-1} \right) \left( (\mathbf{Z}_{k|k-1})_i - \mathbf{z}_{k|k-1} \right)^T \quad (5.32)$$

Finally the a priori error covariance is updated to obtain the a posteriori error covariance  $\underline{\mathbf{P}}_{k|k}$ , which is used in the next filter iteration to calculate the sigma points.

$$\underline{\mathbf{P}}_{k|k} = \underline{\mathbf{P}}_{k|k-1} - \underline{\mathbf{K}}_k \underline{\mathbf{P}}_{\underline{z}_k \underline{z}_k} \underline{\mathbf{K}}_k^T \quad (5.33)$$

## 5.5 Summary

Among many possible attitude estimation methods, the deterministic SVD-method solving Wahba's problem, the EKF method and the UKF method were presented in this chapter.

The SVD-method solving Wahba's problem fails during eclipse and will thus only be implemented in order to provide a sanity check for the UKF.

Focus will be put on making a suitable UKF implementation for AAUSAT3 in Chapter 6. However, an EKF implementation for satellite attitude estimation is also presented in Appendix F. Simulations on the UKF implementations are furthermore conducted in order to evaluate the performance within computational complexity, accuracy and robustness towards bias-, inertia- and modeling errors, followed by a final choice of implementation for AAUSAT3.



## Attitude Determination System for AAUSAT3

In this chapter the Unscented Kalman Filter (UKF) theory from Chapter 5 is applied on a satellite attitude estimation problem. This includes working with quaternions as part of the state vector, which has advantages but also introduces problems addressed in Section 6.1.

Four attitude estimators have been designed and simulated in the Matlab Simulink simulation environment made for the AAUSAT3 CubeSat, see Chapter 4. The first attitude estimator is a deterministic single point SVD-method implementation solving Wahba's problem, which was addressed in Section 5.2. The second attitude estimator is a quaternion UKF implementation without bias estimation having the state vector

$$\mathbf{x} = \begin{bmatrix} {}^c\mathbf{q}^T & {}^c\boldsymbol{\omega}^T \end{bmatrix}^T \quad (6.1)$$

The third attitude estimator is an UKF implementation that also estimate the bias in a three axis magnetometer and a three axis gyroscope and the fourth attitude estimator is a similar implementation that does not include the quaternion  ${}^c\mathbf{q}$ , which reduces the amount of computations, thus taking advantage of the fact that one unit CubeSats are close to symmetric. The third and fourth UKF implementation have the state vector

$$\mathbf{x} = \begin{bmatrix} {}^c\mathbf{q}^T & {}^c\boldsymbol{\omega}^T & \mathbf{b}_{mag}^T & \mathbf{b}_{gyro}^T \end{bmatrix}^T \quad (6.2)$$

The simulation results are finally discussed in Section 6.4 and a conclusion on attitude estimation is made in Section 6.5.

### 6.1 Introduction of the Quaternion Error State

The orientation (attitude) of the satellite is represented by a unit quaternion that again represents the rotation from the ECI to the SBRF. The quaternion has the advantage that it is

singularity free, but there is a unity constraint on the four parameters. This means that the unit quaternion does not belong to a vector space, but belongs to a sphere in  $\mathbb{R}^4$ , where it is not possible to calculate the mean of a set of quaternions by a weighted sum, since it is not closed for addition and scalar multiplication. The resulting quaternion cannot be guaranteed to be a unit quaternion and thus a valid rotation.

One way of solving this problem is to introduce an error quaternion  $\delta\bar{\mathbf{q}}_{k|k}$  defined by Eq. (6.3) and (6.4), as in e.g. [4, 64, 66, 67].

$$\delta\bar{\mathbf{q}}_{k|k} = \mathbf{q}_{k|k} \otimes \mathbf{q}_{k|k-1}^{-1} \quad (6.3)$$

$$\delta\mathbf{q}_{k|k} = [\delta q_1 \quad \delta q_2 \quad \delta q_3]^T \quad (6.4)$$

The error quaternion  $\delta\mathbf{q}_{k|k}$ , containing the vector part of the four element error quaternion  $\bar{\mathbf{q}}_{k|k}$ , is the update to the predicted quaternion  $\mathbf{q}_{k|k-1}$ , giving the estimated quaternion  $\mathbf{q}_{k|k}$ . By using a multiplicative update step, with the quaternion error state, it is possible to maintain the unity constraint, as in Eq. (6.5) and (6.6).

$$\delta\mathbf{q}_{k|k} = \mathbf{K}_k (\mathbf{z}_k - \mathbf{z}_{k|k-1}) \quad (6.5)$$

$$\mathbf{q}_{k|k} = \left[ \delta\mathbf{q}_{k|k}^T \quad \sqrt{1 - \delta\mathbf{q}_{k|k}^T \cdot \delta\mathbf{q}_{k|k}} \right]^T \otimes \mathbf{q}_{k|k-1} \quad (6.6)$$

The three-element error quaternion is expanded to a four-element quaternion in Eq. (6.6) as in e.g. [4, 67].

## 6.2 SVD-method Implementation for Attitude Estimation

The purpose of this implementation is to provide a secondary attitude estimator, that can provide sanity check for the UKF attitude estimator outside eclipse. Having this single point attitude estimator also makes it possible to provide the UKF attitude estimator with a good initial attitude start guess for faster convergence in an initialization phase.

### 6.2.1 Inputs

The observations given as input to the SVD-method attitude estimator are measurements given in the SBRF, which are a sun vector  ${}^s\mathbf{v}_{sun,k}$  and a magnetic field vector  ${}^s\mathbf{v}_{mag,k}$  and predicted measurements in the ECI, which are a sun vector  ${}^i\mathbf{v}_{sun,k|k-1}$  and a magnetic field vector  ${}^i\mathbf{v}_{mag,k|k-1}$ . To summarize the inputs to the SVD-method are:

- Measured sun vector in the SBRF  ${}^s\mathbf{v}_{sun,k}$ .
- Measured magnetic field vector in the SBRF  ${}^s\mathbf{v}_{mag,k}$ .

- Predicted sun vector in the ECI  ${}^i\mathbf{v}_{sun,k|k-1}$ .
- Predicted magnetic field vector in the ECI  ${}^i\mathbf{v}_{mag,k|k-1}$ .

The measured values comes from sensors on the satellite and the predicted measurements comes from on-board software (Ephemeris, SGP4 and IGRF).

### 6.2.2 Constants

Two constants are used in the SVD-method. These are two non-negative weights, chosen to be the inverse variance of the measurement noise ( $a_1 = \sigma_{sun}^{-2}$  and  $a_2 = \sigma_{mag}^{-2}$ ).

### 6.2.3 Implementation

First step in the SVD-method is to find the matrix  $\mathbf{B}$ , based on the measured and predicted observations and the weights, as in Eq. (6.7). This is also shown in Section 5.2.

$$\mathbf{B} = \sigma_{sun}^{-2} {}^s\mathbf{v}_{sun,k} {}^i\mathbf{v}_{sun,k|k-1}^T + \sigma_{mag}^{-2} {}^s\mathbf{v}_{mag,k} {}^i\mathbf{v}_{mag,k|k-1}^T \quad (6.7)$$

Second step is to perform the singular value decomposition, which is denoted *svd* in Eq. (6.8).

$$svd(\mathbf{B}) = [\mathbf{U}, \mathbf{S}, \mathbf{V}] \quad (6.8)$$

The matrices  $\mathbf{U}$  and  $\mathbf{V}$  are orthogonal and the matrix  $\mathbf{S}$  contains the singular values. Numerical implementations of the *svd* algorithm can easily be found in the literature, see e.g. [73] and on the Internet. Third step is to find the optimal rotation matrix  ${}^s\mathbf{A}_{opt}$  as shown in Section 5.2.

$${}^s\mathbf{A}_{opt} = \mathbf{U} \text{diag} [1 \quad 1 \quad \det(\mathbf{U}) \det(\mathbf{V})] \mathbf{V}^T \quad (6.9)$$

The optimal rotation matrix  ${}^s\mathbf{A}_{opt}$  is lastly converted to a quaternion  ${}^s\mathbf{q}$ , since this is the desired attitude representation. Before calculating the optimal rotation matrix, the covariance matrix  $\mathbf{P}$  is calculated according to Eq. (5.7). The result of Eq. (6.9) is not to be trusted if the sum of the elements in the covariance matrix  $\mathbf{P}$  is larger than 0.1 (value chosen based on simulations), which happens when the observations are close to being parallel or if the satellite is in eclipse. In such cases a quaternion equal to (0,0,0,1) is given as output together with the sum of  $\mathbf{P}$ , which can be used for error checking.

## 6.3 UKF Implementations for Attitude Estimation

The implementation of the three quaternion based UKFs are presented in this section.

### 6.3.1 Inputs

The observations given as input to the UKF attitude estimators are measurements given in the SBRF, which are a sun vector  ${}^s\mathbf{v}_{sun,k}$ , a magnetic field vector  ${}^s\mathbf{v}_{mag,k}$  and angular velocity  ${}^s\boldsymbol{\omega}_k$  and predicted measurements in the ECI, which are a sun vector  ${}^i\mathbf{v}_{sun,k|k-1}$  and a magnetic field vector  ${}^i\mathbf{v}_{mag,k|k-1}$ . The UKF implementation also gets an eclipse indication  $E_{ecl,k}$  (this boolean variable is calculated by an ephemeris model and an orbit propagator), the applied control torque  ${}^s\mathbf{N}_{ctrl,k}$  and finally initialization parameters. To summarize the inputs to the UKF are:

- Measured sun vector in the SBRF  ${}^s\mathbf{v}_{sun,k}$ .
- Measured magnetic field vector in the SBRF  ${}^s\mathbf{v}_{mag,k}$ .
- Measured angular velocity of the SBRF  ${}^s\boldsymbol{\omega}_k$ .
- Predicted sun vector in the ECI  ${}^i\mathbf{v}_{sun,k|k-1}$ .
- Predicted magnetic field vector in the ECI  ${}^i\mathbf{v}_{mag,k|k-1}$ .
- Eclipse indication  $E_{ecl,k}$ .
- Applied control torque including torque from permanent magnet in the SBRF  ${}^s\mathbf{N}_{ctrl,k}$ .
- Initial angular velocity of the SBRF  ${}^s\boldsymbol{\omega}_0$ .
- Initial quaternion representing the rotation of the SBRF relative to the ECI  ${}^s\mathbf{q}_0$ .
- Sample time  $T_s$  indicating time between filter iterations.

### 6.3.2 Constants

There are also constants within the UKF implementation and these are:

- Initial error covariance matrix  $\underline{\mathbf{P}}_0$ .
- Process noise matrix  $\underline{\mathbf{Q}}$ .
- Measurement noise matrix  $\underline{\mathbf{R}}$ .
- Satellite inertia matrix  $\underline{\mathbf{I}}_{sat}$ .
- Quaternion representing the rotation of the CRF relative to the SBRF  ${}^c\mathbf{q}$ .
- Weights  $W_i^{(m)}$  and  $W_i^{(c)}$  for  $i = 0, \dots, 2L$ .

### 6.3.3 Implementations

Table 6.1 presents the implemented UKF in pseudo code. Rotation of vectors  $\mathbf{v}$  with quaternions are written as  $\underline{\mathbf{A}}(\mathbf{q})\mathbf{v}$  and defined in Appendix A.

The SVD-method is utilized in the initialization of the UKF, where an initial quaternion estimate  ${}^s\mathbf{q}_0$  is calculated. This ensures faster convergence of the UKF, but it should be noted that this is only possible when two or more non-parallel vector observations are available. The singular value decomposition algorithm is denoted *svd* in Table 6.1.

It is only necessary to calculate the weights once and this is done in the initialization. These weights are determined by the scaling parameters  $\alpha$ ,  $\beta$  and  $\kappa$ , which are set to  $\sqrt{3}$ , 2 and 0, respectively, which gives weights with a total sum of 1 ( $\sum_{i=0}^{2L} W_i^{(m)} = 1$ ,  $\sum_{i=0}^{2L} W_i^{(c)} = 1$ ). The scaling parameters are tuning parameters and the values chosen are based on filter performance results and the guidelines in Section 5.4.

Calculating sigma points involves finding the matrix square root and an efficient way of calculating this is done by use of a Cholesky decomposition, as described in e.g. [63, 64, 74]. In the implementation a lower triangular Cholesky factorization is used and this is denoted *ch* in Table 6.1. The parameter  $L = 6$  is the number of error states (three quaternion parameters and three angular velocities). This gives a total of 13 scaled symmetric error sigma points distributed around zero, where the first one is zero, as presented in Table 6.1. These error sigma points are then expanded with quaternion multiplication and angular velocity addition, using the previous attitude estimate  $\mathbf{x}_{k-1|k-1}$ .

In the predict step the state is propagated using the non-linear satellite equations of motion defined in Eq. (C.18) and (C.11). This can be done by numerically integrating the continuous time functions over a period of  $T_s$  using a fourth order Runge Kutta implementation as in e.g. [74]. Since using a Runge Kutta method (denoted RK4 in step 1.4 in Table 6.1) involves addition in each sub-step, a normalization of the quaternion is required, in order for it to present a valid rotation. If the time step  $T_s$  is large and the angular velocity is high, it might be necessary to use more sub-steps within the Runge Kutta implementation and/or normalize after each sub-step rather than only after the last sub-step. The predict step has to be repeated for all the sigma points. The a priori full state estimate is then calculated as a weighted sum of all the propagated sigma points, with quaternion normalization, as defined in step 1.5. This approximation of the mean quaternion is also performed in e.g. [64].

A normalization of the Sun and the magnetic field vectors are performed, since it is only the direction of the vectors and not the magnitude that is important for attitude estimation. Sun vector measurements can only be considered valid when the satellite is not in eclipse. The Sun vector measurement and prediction is therefore set equal to each other during eclipse, which is equivalent to stating that there is no discrepancy between the two and thus no update is performed on the state in the update step. The same is the case if there is no new measurements available, which happens when one sensor is sampled slower than each time step  $T_s$ . The state should only be updated by the particular measurement if it has changed since last filter iteration. The normalization of vectors, eclipse check and new measurement check is performed in steps 2.1 - 2.6, where the measurements and predicted measurements are also rotated to the CRF. The predicted measurements are gathered with a weighted sum

<b>Init.:</b>		
0.1	Solve Wahba (SVD):	$\mathbf{B} = \sigma_{sun}^{-2} \mathbf{v}_{sun,k} \mathbf{v}_{sun,k k-1}^T + \sigma_{mag}^{-2} \mathbf{v}_{mag,k} \mathbf{v}_{mag,k k-1}^T$ $svd(\mathbf{B}) = [\mathbf{U}, \mathbf{S}, \mathbf{V}]$ , ${}^s\mathbf{A}_{opt} = \mathbf{U} \mathit{diag}[1 \ 1 \ \det(\mathbf{U}) \ \det(\mathbf{V})] \mathbf{V}^T$
0.2	Calculate ${}^s\mathbf{q}_0$ :	${}^s\mathbf{A}_{opt}$ to ${}^s\mathbf{q}_0$ conversion
0.3	Initialize state:	$\mathbf{x}_{k k} = [({}^s\mathbf{q}_0 \otimes {}^s\mathbf{q})^T \ (\mathbf{A}({}^s\mathbf{q})^s\omega_0)^T]^T$
0.4	Calculate weights:	$W_0^{(m)} = \frac{\lambda}{L+\lambda}$ , $W_0^{(c)} = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta)$ $W_i^{(m)} = W_i^{(c)} = \frac{1}{2(L+\lambda)}$ , $i = 1, \dots, 2L$
0.5	Save:	${}^s\mathbf{v}_{sun,k}$ , ${}^s\mathbf{v}_{mag,k}$ , ${}^s\omega_k$ , ${}^s\mathbf{N}_{ctrl,k}$ go to 2.15
<b>Predict:</b>		
1.1	Error sigma points:	$\delta\chi_{k-1 k-1} = [\mathbf{0}_{6 \times 1} \ - (ch((L+\lambda)\mathbf{P}_{k-1 k-1}))^T \ (ch((L+\lambda)\mathbf{P}_{k-1 k-1}))^T]^T$
1.2	Full sigma points:	$(\chi_{k-1 k-1})_i = \left[ \left[ \left( \delta\chi_{k-1 k-1} \right)_{1-3,i}^T \ \sqrt{1 - \left( \delta\chi_{k-1 k-1} \right)_{1-3,i}^T \left( \delta\chi_{k-1 k-1} \right)_{1-3,i}} \right]^T \right.$ $\left. \otimes {}^c\mathbf{q}_{k-1 k-1} \right)^T \left( {}^c\omega_{k-1 k-1} + \left( \delta\chi_{k-1 k-1} \right)_{4-6,i} \right)^T \right]^T$ , $i = 1, \dots, 2L + 1$ $(\cdot)_{1-3,i}$ = row 1-3, col. $i$
1.3	Rotate control torque:	${}^c\mathbf{N}_{ctrl,k-1} = \mathbf{A}({}^c\mathbf{q}) {}^s\mathbf{N}_{ctrl,k-1}$
1.4	Numerical propagation:	$(\chi_{k k-1})_i = \text{RK4} \left( (\chi_{k-1 k-1})_i, {}^c\mathbf{N}_{ctrl,k-1}, T_s, steps \right)$ , $i = 1, \dots, 2L + 1$
1.5	A priori state estimate:	$\mathbf{x}_{k k-1} = \left[ \left( \frac{\sum_{i=0}^{2L} W_i^{(m)} (\chi_{k k-1})_{1-4,i+1}}{\sum_{i=0}^{2L} W_i^{(m)} (\chi_{k k-1})_{1-4,i+1}} \right)^T \ \left( \sum_{i=0}^{2L} W_i^{(m)} (\chi_{k k-1})_{5-7,i+1} \right)^T \right]^T$
1.6	Full to error state:	$(\delta\chi_{k k-1})_i = \left[ \left( (\chi_{k k-1})_{1-4,i} \otimes {}^c\mathbf{q}_{k k-1}^{-1} \right)_{1-3,i}^T \ \left( (\chi_{k k-1})_{5-7,i} - {}^c\omega_{k k-1} \right)^T \right]^T$ $, i = 1, \dots, 2L + 1$
1.7	Mean error state:	$\delta\mathbf{x}_{k k-1} = \sum_{i=0}^{2L} W_i^{(m)} (\delta\chi_{k k-1})_{i+1}$
1.8	A priori covariance:	$\mathbf{P}_{k k-1} = \sum_{i=0}^{2L} W_i^{(c)} \left( (\delta\chi_{k k-1})_{i+1} - \delta\mathbf{x}_{k k-1} \right) \left( (\delta\chi_{k k-1})_{i+1} - \delta\mathbf{x}_{k k-1} \right)^T + \mathbf{Q}$
<b>Update:</b>		
2.1	Save:	${}^s\mathbf{v}_{sun,k}$ , ${}^s\mathbf{v}_{mag,k}$ , ${}^s\omega_k$ , ${}^s\mathbf{N}_{ctrl,k}$
2.2	Eclipse check:	If in eclipse go to 2.6, else go to 2.3
2.3	New measurement?:	If new vector measurement go to 2.4, else go to 2.6
2.4	Normalize and rotate:	${}^c\mathbf{v}_{sun,k} = \mathbf{A}({}^c\mathbf{q}) \frac{{}^s\mathbf{v}_{sun,k}}{\ {}^s\mathbf{v}_{sun,k}\ }$
2.5	Estimate measurement:	$({}^c\mathbf{v}_{sun,k k-1})_i = \mathbf{A} \left( (\chi_{k k-1})_{1-4,i} \right) \frac{{}^s\mathbf{v}_{sun,k k-1}}{\ {}^s\mathbf{v}_{sun,k k-1}\ }$ , $i = 1, \dots, 2L + 1$ ${}^c\mathbf{v}_{sun,k k-1} = \sum_{i=0}^{2L} W_i^{(m)} ({}^c\mathbf{v}_{sun,k k-1})_{i+1}$ go to 2.7
2.6	Hardcode vector:	${}^c\mathbf{v}_{sun,k} = {}^c\mathbf{v}_{sun,k k-1} = [0 \ 0 \ 0]^T$
2.7	Repeat step 2.3 - 2.6:	For magnetic field vectors ( ${}^c\mathbf{v}_{mag,k}$ , ${}^c\mathbf{v}_{mag,k k-1}$ ) and angular velocities ( ${}^c\omega_k$ , ${}^c\omega_{k k-1}$ )
2.8	Calculate covariances:	$\mathbf{P}_{x_k z_k}$ and $\mathbf{P}_{z_k z_k}$ according to Eq. (5.32) and (5.31)
2.9	Calculate Kalman gain:	$\mathbf{K}_k = \mathbf{P}_{x_k z_k} \mathbf{P}_{z_k z_k}^{-1}$
2.10	Calculate error state:	$\delta\mathbf{x}_{k k} = \mathbf{K}_k (\mathbf{z}_k - \mathbf{z}_{k k-1})$
2.11	Expand quaternion:	${}^c\mathbf{q}_{k k} = \left[ \delta\mathbf{q}_{k k}^T \ \sqrt{1 - \delta\mathbf{q}_{k k}^T \cdot \delta\mathbf{q}_{k k}} \right]^T \otimes {}^c\mathbf{q}_{k k-1}$
2.12	Calculate full state:	$\mathbf{x}_{k k} = [{}^c\mathbf{q}_{k k}^T \ ({}^c\omega_{k k-1} + \delta\omega_{k k})^T]^T$
2.13	A posteriori covariance:	$\mathbf{P}_{k k} = \mathbf{P}_{k k-1} - \mathbf{K}_k \mathbf{P}_{z_k z_k} \mathbf{K}_k^T$
2.14	Rotate and output:	Output = $\left[ ({}^c\mathbf{q}_{k k} \otimes {}^c\mathbf{q}^{-1})^T \ (\mathbf{A}({}^c\mathbf{q}^{-1}) {}^c\omega_{k k})^T \right]^T$
2.15	Repeat:	Go to 1.1 and iterate filter every time step $T_s$

Table 6.1: Pseudo code for quaternion UKF implementation without bias estimation.

in step 2.5 and this result is used in step 2.10.

The UKF implementation with bias estimation is very similar to the pseudo code in Table 6.1. The only difference is that the state vector is expanded by six states (see Eq. (6.2)) and the bias is approximated as being constant during propagation. Furthermore, simulations have shown that the magnetometer bias should only be updated in the update step when both sun and magnetic field vector measurements are available. This means that the UKF implementation should be started outside eclipse the first time it is run and the bias should be saved as a starting guess if the filter goes through on/off cycles. Estimating bias in magnetometer and gyroscope is considered more important than estimating bias in sun sensor measurements, since the permanent magnet and the magnetic fields created by the onboard electronics will create a bias in the magnetometer measurements and the gyroscope has a temperature dependent voltage null-offset. However, adding estimation of sun sensor bias is considered fairly easy, since it is done in the same way as with the other two sensors, but it should be remembered that it is computationally more demanding to add additional states and should therefore be avoided.

The model noise covariance matrix  $\underline{\mathbf{Q}}$ , measurement noise covariance matrix  $\underline{\mathbf{R}}$  and initial error covariance matrix  $\underline{\mathbf{P}}_0$  was set to the following based on sensor noise and iterative testing (*diag* refers to a diagonal matrix):

$$\begin{aligned}\underline{\mathbf{Q}} &= \text{diag} [1 \ 1 \ 1 \ 10 \ 10 \ 10 \ 1 \ 1 \ 1 \ 0.01 \ 0.01 \ 0.01] 10^{-6} \\ \underline{\mathbf{R}} &= \text{diag} [3.4 \ 3.4 \ 3.4 \ 2.7 \ 2.7 \ 2.7 \ 0.012 \ 0.012 \ 0.012] 10^{-3} \\ \underline{\mathbf{P}}_0 &= \text{diag} [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0.1 \ 0.1 \ 0.1 \ 1 \ 1 \ 1] 10^{-3}\end{aligned}$$

The last 6 diagonal elements in  $\underline{\mathbf{Q}}$  and  $\underline{\mathbf{P}}_0$  are only necessary in the UKF implementation with bias estimation.

The model noise covariance matrix should reflect the system noise in order to have consistent and realistic performance. Choosing lower values will increase performance, but this will not reflect the real situation in space where the model is not ideal. Lowering the values is the same as putting more “trust” in the model. The first three diagonal elements in the  $\underline{\mathbf{Q}}$  matrix reflects the expected noise in the kinematic equations (quaternion part), the next three elements reflects the expected noise in the dynamic equations (angular velocity part) and the last six elements reflects the expected noise in the sensor bias states. Less trust has been put in the dynamic equations since all the disturbance torques are not part of the satellite equation of motion. These torques would require a lot of computational power to estimate and they are most likely dominated by the satellites magnetic residual, which is modeled as Gaussian vector noise.

As with the model noise covariance matrix, the measurement noise covariance matrix should reflect the expected noise in the measurements. The first three diagonal elements reflects the expected noise in the sun vector measurement, the next three elements relates to the noise in the magnetic field vector measurement and the last three elements reflects the expected noise in the angular velocity measurements. The sensor setup chosen for AAUSAT3 was simulated by taking the output from the “truth model” and adding Gaussian noise, see Section 4.6. Consideration to power, size and cost of sensors was a key issue when choosing sensors, which is why the performance of the sensors are not in the high end.

The sun sensor measurement is expected to have a deviation from the correct angle of up to  $\pm 10$  [deg] (in both of the spherical coordinates  $\phi$  and  $\theta$ ), which approximately gives a standard deviation of 3.33 [deg] if assuming that 10 [deg] is equal to  $3\sigma$ . This is a standard deviation of  $\sin(3.33) = 0.058$  in each of the Cartesian unit vector coordinates, however, not at the same time. Therefore a variance of  $\sigma^2 = 0.058^2 = 0.0034$  has been chosen. The magnetic field sensor (magnetometer) is expected to have a standard deviation of 3 [deg], which gives a reasonable choice of variance in unit vector coordinates of  $\sigma^2 = (\sin(3))^2 = 0.052^2 = 0.0027$ . The noise in the angular velocity is expected to have a standard deviation of 0.0035 [rad/s], giving a variance of  $\sigma^2 = 0.0035^2 = 0.000012$ . The chosen gyroscope for AAUSAT3 measures the angular velocity directly using the Coriolis effect and therefore has no simulated drift term.

The initial error covariance matrix is a measure of how good a guess the initial state is. Choosing low values in the diagonal is the same as putting “trust” in the initial guess.

### 6.3.4 Simulation Setup and Results

The simulation environment made for AAUSAT3 is described in Chapter 4. The TLE used in the simulations is for the AAUSAT-II satellite, which is in a polar orbit with an inclination of approximately 98 [deg] and an orbit height of approximately 630 [km]. This Low Earth Orbit (LEO) is considered typical for CubeSats.

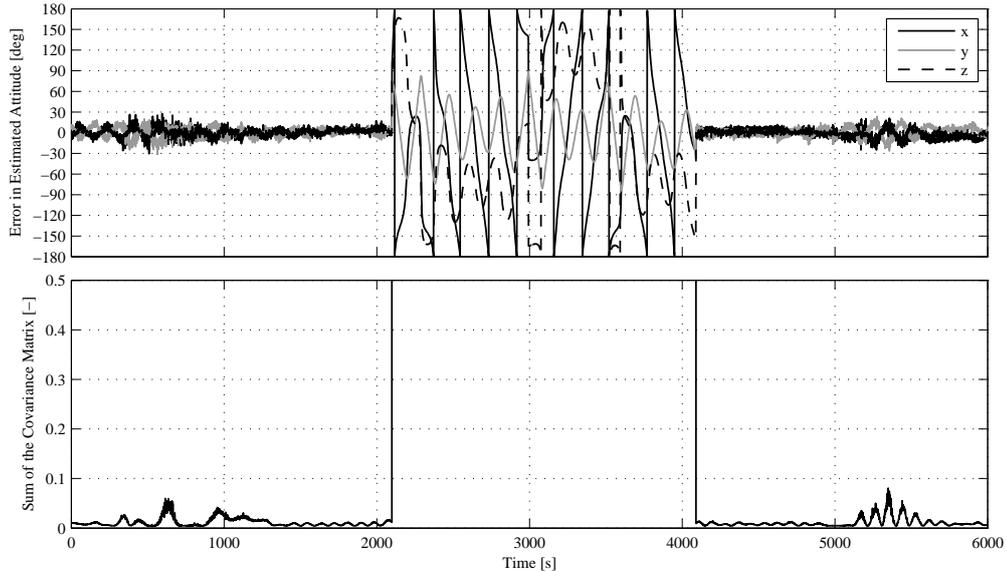
The simulation environment also provides what is considered as satellite onboard software. This software consists of another IGRF10 magnetic field model, implemented with order 8, that, together with a SGP4 orbit propagator provides the predicted magnetic field vector in the ECI. It also provides a sun vector and eclipse indication with ephemeris models and the SGP4. The onboard software does not estimate any of the disturbance torques, however, the torque from a permanent magnet is simulated and given as control input torque to the UKF. This permanent magnet has a magnetic moment of 0.0030 [ $Am^2$ ] and will be part of the ADCS for AAUSAT3.

Attitude estimation is often performed after the satellite has been detumbled and the initial angular velocities in the tests were therefore set to (0.02,0.02,0.02) [rad/s], which is within a realistic bound of what can be expected. The sampling frequency was set to 1 [Hz].

A number of simulations were performed on the four implementations and these are listed in Table 6.2. Realistic sensor biases were applied in some of the simulations. These biases were randomly chosen and different from the ones used during tuning of the filter. The mass property and thus the inertia was changed in the third last simulation so that the center of mass was located approximately 20 [mm] from the geometric center of the satellite, which was just within the CubeSat requirement specified in [75] (center of mass moved from (-0.9,-1.1,-12.0) [mm] to (-0.7,-0.8,20.0) [mm] relative to the geometric center). The quaternion  ${}^c\mathbf{q}$ , that represents the rotation from the SBRF to the CRF, was removed from the implementation in the second last simulation. Ten Runge Kutta sub-steps was used in all the above simulation, but in the last simulation only one sub-step is used and additional sensor measurement errors are introduced in the form of sensor displacement of 1 [deg] in

heading and tilt (the spherical coordinates  $\theta$  and  $\phi$ ) on the magnetometer and gyroscope. This simulation is added to test the robustness of the implementation.

Figure 6.1 shows the results of a simulation of a stand-alone implementation of the SVD-method for approximately one orbit. The SVD-method failed during eclipse (from 2100 to 4090 [s]), which was expected, because the sun vector was set to (0,0,0), thus making the attitude unobservable. The sum of elements in the covariance matrix was also simulated and this graph has been limited to 0.5, but it was infinite during eclipse.



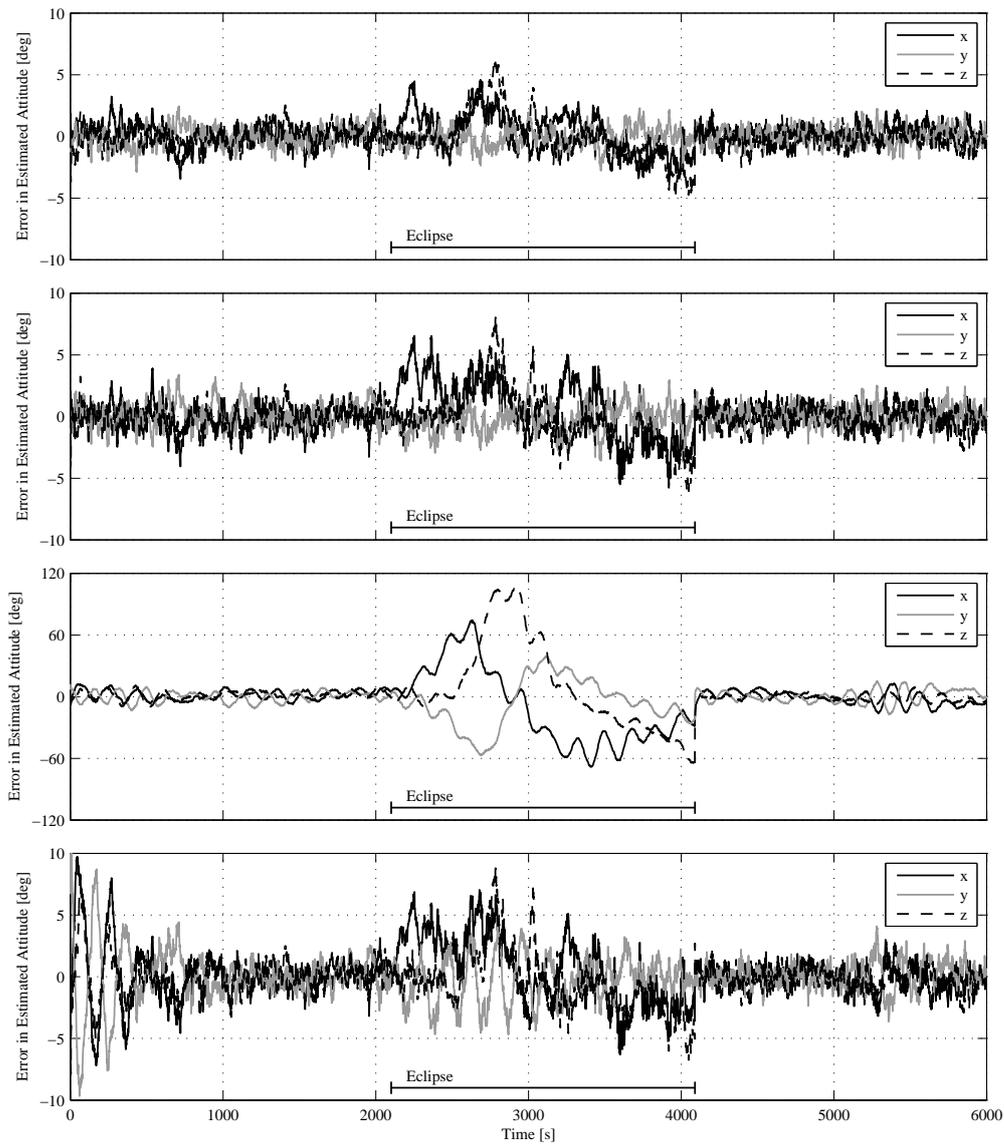
**Figure 6.1:** Graph 1: Difference between the estimated attitude from the SVD-method and the attitude from the “truth model” ( $x$ ,  $y$  and  $z$  are the Euler angles). Graph 2: Sum of elements in covariance matrix calculated according to Equation (5.7).

Figure 6.2 shows the results of a simulation of the implementation of the quaternion UKF with (Graph 2 and 4) and without (Graph 1 and 3) bias estimation for approximately one orbit. There was no measurement bias in the simulations shown on Graph 1 and 2 and a small bias was applied in the simulations shown on Graph 3 and 4.

Figure 6.3 shows the results of a simulation of the implementation of the quaternion UKF with bias estimation for approximately one orbit. Graph 1 is a situation where the inertia in the UKF had a small deviation compared to the one used in the “truth model” and Graph 2 was a situation where all rotations with the quaternion  ${}^c\mathbf{q}$  was removed from the UKF implementation.

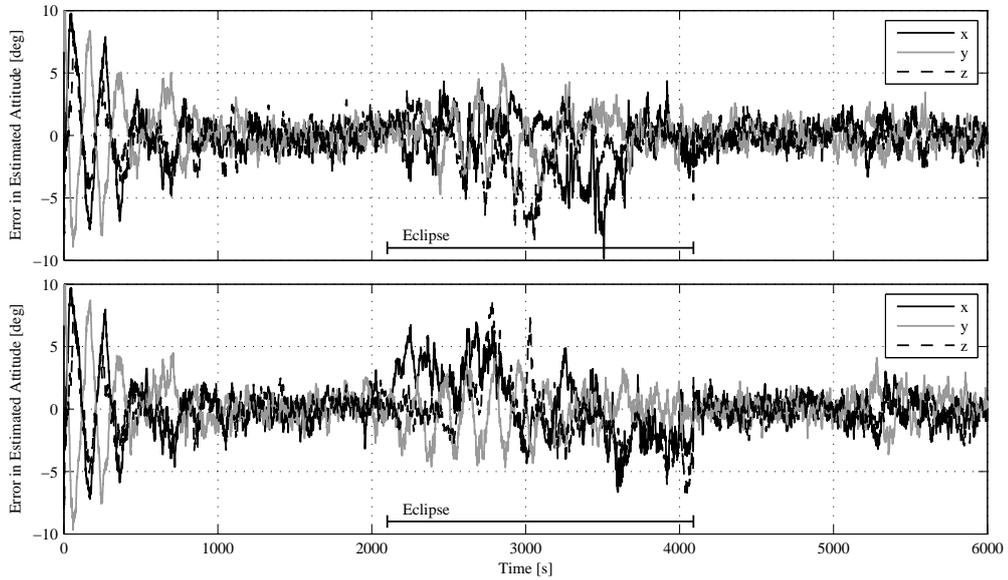
Figure 6.4 shows the results of a simulation of the implementation of the quaternion UKF with bias estimation for approximately one orbit. Only one Runge Kutta sub-step was used in this simulation and the magnetometer and gyroscope sensors were displaced by 1 [deg] in heading and tilt (the spherical coordinates  $\theta$  and  $\phi$ ).

The largest angular deviation has been identified for all simulations and classified into time regions. The first region from 0 to 1000 [s] is defined as the convergence phase. The second

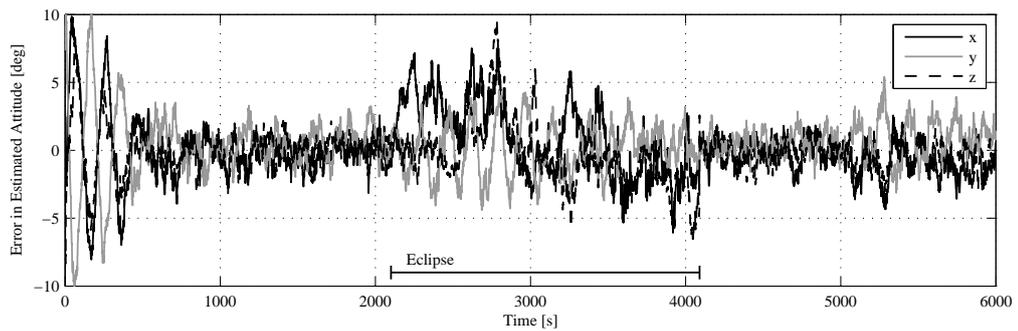


**Figure 6.2:** Difference between the estimated attitude and the attitude from the “truth model” ( $x$ ,  $y$  and  $z$  are the Euler angles). Graph 1: UKF without bias estimation and no bias on measurements. Graph 2: UKF with bias estimation and no bias on measurements. Graph 3: UKF without bias estimation and with bias on measurements. Graph 4: UKF with bias estimation and with bias on measurements.

region is from 1001 [s] to 2100 (until eclipse), the third region is from 2101 to 4095, which is inside eclipse and 5 [s] after and the last region is after eclipse from 4096 to 6000 [s]. Table 6.2 presents these angular deviations.



**Figure 6.3:** Difference between the estimated attitude and the attitude from the “truth model” ( $x$ ,  $y$  and  $z$  are the Euler angles). Graph 1: UKF with bias estimation, with bias on measurements and with a small inertia difference. Graph 2: UKF with bias estimation, without  $g$  and with bias on measurements.



**Figure 6.4:** Difference between the estimated attitude and the attitude from the “truth model” ( $x$ ,  $y$  and  $z$  are the Euler angles) for an UKF implementation with bias estimation, with bias on measurements, with sensor displacement and with only one Runge Kutta sub-step.

Implementation	Ref. [Fig., Graph]	RK [sub - steps]	Sens. Displ. [deg]	Mag. Bias [nT]	Gyro. Bias [deg/s]	Inertia	Err. 1 [deg]	Err. 2 [deg]	Err. 3 [deg]	Err. 4 [deg]	Time [ms]
Stand-alone SVD-method	(6.1,1)	N/A	(0,0)	(5,1,-3)·10 <sup>3</sup>	N/A	N/A	31.53	17.32	N/A*	26.51	0.22
UKF w.o. bias est.	(6.2,1)	10	(0,0)	(0,0,0)	(0,0,0)	Correct	3.73	2.67	6.10	2.59	61.67
UKF w. bias est.	(6.2,2)	10	(0,0)	(0,0,0)	(0,0,0)	Correct	4.11	3.28	8.04	2.80	125.02
UKF w.o. bias est.	(6.2,3)	10	(0,0)	(5,1,-3)·10 <sup>3</sup>	(0.2,0.2,0.2)	Correct	13.23	11.63	105.82†	17.36	-★
UKF w. bias est.	(6.2,4)	10	(0,0)	(5,1,-3)·10 <sup>3</sup>	(0.2,0.2,0.2)	Correct	12.16	3.26	8.80	4.09	-★
UKF w. bias est.	(6.3,1)	10	(0,0)	(5,1,-3)·10 <sup>3</sup>	(0.2,0.2,0.2)	Incorrect	12.16	3.47	9.85	3.46	-★
UKF w. bias est. (No $\hat{\mathbf{q}}$ )	(6.3,2)	10	(0,0)	(5,1,-3)·10 <sup>3</sup>	(0.2,0.2,0.2)	Correct	12.16	3.26	8.75	4.12	123.84
UKF w. bias est.	(6.4,1)	1	(1,1)	(5,1,-3)·10 <sup>3</sup>	(0.2,0.2,0.2)	Correct	12.90	3.30	9.51	5.41	21.84

\*The SVD-method does not work during eclipse.

†The UKF without bias estimation works very poorly during eclipse, if there is bias in the measurements.

★ The measurement were not performed.

**Table 6.2:** Simulation results for one orbit. Column 1-7: Simulated case. Column 8-11: Largest absolute difference between the estimated attitude and the attitude from the “truth model”. Maximum is taken as the Euler angle (x, y or z) with the largest deviation. Err: 1 is between 0-1000 [s], Err: 2 is between 1001-2100 [s], Err: 3 is between 2101-4095 [s] and Err: 4 is between 4096-6000 [s]. Column 12: Average computation times per iteration for the implemented attitude estimators in Matlab on a single core Pentium 4 (2.8 [GHz], 1 [GB] RAM).

Average computation times was recorded with the `tic-toc` commands in Matlab for the four different implementations and the results are presented in Table 6.2.

## 6.4 Discussion on Attitude Determination

The stand-alone implementation of the SVD-method does provide an attitude estimate that stays within approximately  $\pm 30$  [deg] during the simulated orbit. However, this performance is far from the performance of the quaternion UKF implementation and it does not work during eclipse. The error covariance matrix  $\mathbf{P}$  can be used to evaluate the performance of the SVD-method, where a large value of  $\mathbf{P}$  indicates a poor estimate of the attitude. Graph 2 in Figure 6.1 shows a very large  $\mathbf{P}$  during eclipse and also relatively large values when the satellite has passed the poles, which is where the vector observations are expected to be close to parallel. Calculating  $\mathbf{P}$  thus gives a measure of the confidence in the attitude estimate. Simulations have additionally shown that the quaternion UKF implementation with bias estimation has trouble converging if it is not supplied with a fairly accurate initial quaternion from e.g. the SVD-method. This is due to the chosen parameter tuning of the UKF, which is not too aggressive for better performance after the filter has converged and found a good sensor bias estimate. The tuning could be varied during simulation, however, it can be difficult to know when the filter has converged properly and providing the UKF implementation with an initial quaternion from the SVD-method seems to be more robust. Implementing the SVD-method also introduces the possibility of making sanity checks on the UKF output.

As Graph 3 and 4 in Figure 6.2 indicates, bias in the sensor measurements greatly reduces the performance of the UKF unless the bias is estimated along with the attitude. However, it should be noted that expanding the state vector from 7 to 13 almost doubles the computation time of the filter (see Table 6.2).

As seen in Graph 1 in Figure 6.3, small deviations between the estimated and actual inertia do not affect the performance of the UKF significantly. Larger deviations will affect the performance of the UKF, however the CubeSat standard prescribes that the center of mass must not be further away than 2 [cm] from the geometric center for one unit CubeSats. It has been experienced that the maximum corresponding deviation in the inertia do not affect the performance of the UKF and hence inertia estimation is unnecessary for one unit CubeSats.

The settings of the simulated UKF is set to rely mostly on the obtained measurements as the torques from disturbances have not been included in the satellite equations of motion in the onboard software. Furthermore, it is not expected that the control torque generated by e.g. permanent magnets and magnetorquers is precisely known, which introduces additional uncertainties. This means that the filter property of the UKF is toned down to improve robustness especially during eclipse at the expense of performance outside eclipse. However, this is a matter of tuning.

The main disadvantage of the UKF is the computation time. Taking advantage of the fact that one unit CubeSats are close to symmetric, it is possible to refrain from the rotation between the SBRF and the CRF, as indicated in Graph 2 in Figure 6.3. However, this do not

lower the computation time significantly (125.02 to 123.84 [ms], see Table 6.2). Most of the computation time is used to compute the predict step 1.4 (see Table 6.1). Hence, reducing the number of Runge Kutta sub-steps is the most effective way of reducing the computational time. The last simulation with only one Runge Kutta sub-step used only 21.84 [ms]. This is approximately 1/5 of the 125.02 [ms] stated in Table 6.2, with almost equivalent performance (the slight decrease in performance is mainly caused by sensor displacement). The reason for the almost equivalent performance is expected to be due to the low angular velocity after detumbling.

The largest estimation error outside eclipse of 5.41 [deg], for the quaternion UKF implementation with bias estimation, were encountered in the last simulation, which is considered acceptable, since the sensors were also displaced by 1 [deg] in heading and tilt in addition to the measurements being biased and very noisy.

The implementations proposed here are targeted for CubeSats, however, the results are extendable to larger satellites. Doing so might require the incorporation of some of the disturbance models, such as calculation of the gravity gradient torque, in the onboard satellite equations of motion. This can also be done on CubeSats, but might not be appropriate with the limited computational power often associated with these.

## 6.5 Summary

As mentioned earlier, the focus of this thesis has been to present a robust attitude estimator for low budget CubeSats. To summarize, it has proven important to estimate bias in the sensors, while inertia estimation is unnecessary for small symmetrically shaped satellites, since the diagonal elements in the inertia matrix are almost equal. Simulations of the implemented quaternion UKF with bias estimation have shown, that it is possible to estimate the attitude of the satellite within approximately  $\pm 4$  [deg] outside eclipse and within approximately  $\pm 9$  [deg] in eclipse, with the low cost of the shelf sensor setup chosen for AAUSAT3.

It is recommended that the quaternion UKF with bias estimation and one Runge Kutta sub-step is used for AAUSAT3. This is due to filter computation time, robustness and performance. It is still an open question whether it is necessary to estimate bias in sun sensor measurements, however, a final sun sensor design has not been finished yet.

## Detumble Controller

Two attitude controllers are designed for AAUSAT3. The first is a detumble controller, which purpose is to reduce the angular velocity of the satellite and the second is a pointing controller, which purpose is to track a given reference attitude. Keeping the angular velocity of the satellite low and controlled is of paramount importance for the mission. Firstly, the antennas on AAUSAT3 will have lower gains in certain directions (e.g. dipole antennas have lower gain in the direction parallel to the antenna) and experiences from AAUSAT-II have proven that high angular velocity (it has been estimated to tumble with approximately 2.5 [Hz] [13]) makes communication difficult, due to worsened S/N ratio in certain angles and only short messages like beacon signals are received properly. The interception of AIS signals from ships is also improved by keeping the highest gain of the antenna down towards the ships for as long as possible. Secondly, camera picture quality is also improved when the camera is not spinning at high angular velocity.

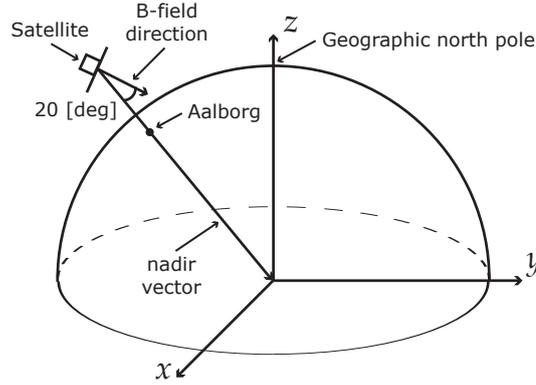
This chapter describes the design of the detumble controller. First, the aim of the detumble controller is given in Section 7.1. Then, a control law is given, followed by a stability analysis in Section 7.2 and a description of the implementation for AAUSAT3 in Section 7.3. Finally, the controller is evaluated based on simulations in Section 7.4.

### 7.1 Aim

The aim of the detumble controller is, as stated in requirement §6.5.1, to reduce the initial spin of the satellite from 10 [deg/s] to below  $\pm 0.3$  [deg/s] in the three SBRF axes within three orbits. Furthermore, the detumble controller must function as a recovery mechanism in situations where e.g. a malfunctioning pointing controller has caused the satellite to spin up.

The satellite is, as stated in Section 3.2, controllable by means of magnetorquers and a permanent magnet. Placing the permanent magnet's magnetic dipole moment in the opposite direction of the satellite's z-axis, will cause the detumbled satellite's z-axis to point in the opposite direction of Earth's magnetic field. This will automatically bring the antennas in

a position where they are closest to the Earth's surface and point the highest gain of the antennas downwards at the North Pole, but also over Greenland and Aalborg. The angle between the direction of the Earth's magnetic field at Aalborg in an orbit height of 630 [km] and a nadir pointing vector is approximately 20 [deg]. Figure 7.1 illustrates the satellite z-axis attitude over Aalborg and a nadir pointing vector. The angle is even smaller at higher elevations.



**Figure 7.1:** Satellite z-axis attitude over Aalborg determined by the Earth's magnetic field.

## 7.2 Control Law and Stability Analysis

A commonly used control algorithm for detumbling of satellites in polar Low Earth Orbits (LEO), is the  $\dot{\mathbf{B}}$  or B-dot controller, see e.g. [44, 76, 77, 78]. This controller is also chosen for AAUSAT3 because of its simplicity. It works by applying a negative gain  $\mathbf{C}_d$  to the change in magnetic field  ${}^s\dot{\mathbf{B}}_E$  measured in the three body axes, thus creating an opposing moment  ${}^s\mathbf{m}_{ctrl}$ , which will dissipate energy and despin the satellite. The control law is

$${}^s\mathbf{m}_{ctrl} = -\mathbf{C}_d {}^s\dot{\mathbf{B}}_E \quad (7.1)$$

This control law only requires a three axis magnetometer in order to find the derivative of the magnetic field and a constant gain. The three elements in the magnetic moment vector  ${}^s\mathbf{m}_{ctrl}$  applies directly to the x-, y- and z-axis magnetorquer coils. The control law causes the satellite to follow the Earth's magnetic field and thus reduces the angular velocity to approximately two revolutions per orbit. A modified control law is used, because of the magnetic moment from the permanent magnet  ${}^s\mathbf{m}_{pm}$ . This control law is stated in [76] and defined as

$${}^s\mathbf{m}_{ctrl} = -\mathbf{C}_d {}^s\dot{\mathbf{B}}_E + {}^s\mathbf{m}_{pm} \quad (7.2)$$

which gives the control torque  ${}^s\mathbf{N}_{ctrl}$  defined as

$${}^s\mathbf{N}_{ctrl} = (-\mathbf{C}_d {}^s\dot{\mathbf{B}}_E + {}^s\mathbf{m}_{pm}) \times {}^s\mathbf{B}_E \quad (7.3)$$

According to e.g. [76], the change in magnetic field can be approximated with (Rotation of vectors  $\mathbf{v}$  with quaternions are written as  $\underline{\mathbf{A}}(\mathbf{q})\mathbf{v}$ ):

$${}^s\dot{\mathbf{B}}_E = {}^s\mathbf{B}_E \times {}^s\boldsymbol{\omega} + \underline{\mathbf{A}}({}^s\mathbf{q}) {}^e\dot{\mathbf{B}}_E \approx {}^s\mathbf{B}_E \times {}^s\boldsymbol{\omega} \quad (7.4)$$

if assuming that the change in magnetic field, caused by a changing orbit position, is negligible compared to the change caused by a rotating satellite (rotating with the angular velocity  ${}^s\boldsymbol{\omega}$ ). This means that it is accepted that a detumbled satellite will follow the Earth's magnetic field and thus rotate with approximately two revolutions per orbit (0.12 [deg/s]), corresponding to the change in magnetic field direction.

Lyapunov's direct method (see e.g. [76]) is used in the following to verify that the proposed control law in Eq. (7.2) does dissipate energy and despin the satellite. An energy function  $E_{tot} = E_{kin} + E_{pot}$  is used as Lyapunov candidate function where disturbance torques have been neglected. The kinetic energy of the satellite  $E_{kin}$  is a function of the angular velocity of the satellite and is defined as

$$E_{kin}({}^s\boldsymbol{\omega}) = \frac{1}{2} {}^s\boldsymbol{\omega}^T {}^s\mathbf{I}_{sat} {}^s\boldsymbol{\omega} \quad (7.5)$$

and the magnetic potential energy of the satellite  $E_{pot}$  caused by the dipole permanent magnet is defined as [36]

$$E_{pot}(\theta_{bpm}) = \|{}^s\mathbf{m}_{pm}\| \|{}^s\mathbf{B}_E\| - {}^s\mathbf{m}_{pm}^T {}^s\mathbf{B}_E = \|{}^s\mathbf{m}_{pm}\| \|{}^s\mathbf{B}_E\| (1 - \cos(\theta_{bpm})) \quad (7.6)$$

if the magnetic field given in the ECEF does not change direction and strength. A change during the orbit does occur, but this change is considered negligible as already mentioned.  $\theta_{bpm}$  is the angle between the local geomagnetic field vector and the magnetic moment vector and Eq. (7.6) was also presented in Section 3.3. This gives a total energy which is equal to

$$E_{tot}({}^s\boldsymbol{\omega}, \theta_{bpm}) = \frac{1}{2} {}^s\boldsymbol{\omega}^T {}^s\mathbf{I}_{sat} {}^s\boldsymbol{\omega} + \|{}^s\mathbf{m}_{pm}\| \|{}^s\mathbf{B}_E\| (1 - \cos(\theta_{bpm})) \quad (7.7)$$

This Lyapunov function is positive definite and the equations of the system have an equilibrium point at  ${}^s\boldsymbol{\omega} = \mathbf{0}$  and  $\theta_{bpm} = 0$ , where the Lyapunov function is 0.

$$E_{tot}(\mathbf{0}, 0) = 0 \quad (7.8)$$

$$E_{tot}({}^s\boldsymbol{\omega}, \theta_{bpm}) > 0 \quad \forall {}^s\boldsymbol{\omega} \neq \mathbf{0}, \forall \theta_{bpm} \neq 0 \quad (7.9)$$

Actually the equations of the system have an unstable equilibrium at  $\theta_{bpm} = 180$ , because the permanent magnet works like a compass needle. However, as with the inverted pendulum, it will not stay in this position.

The derivative of  $E_{tot}$  is (for  $\|{}^s\mathbf{B}_E\|$  assumed to be constant<sup>1</sup>):

$$\dot{E}_{tot}({}^s\boldsymbol{\omega}, \theta_{bpm}) \approx \frac{1}{2} {}^s\boldsymbol{\omega}^T {}^s\mathbf{I}_{sat} {}^s\dot{\boldsymbol{\omega}} + \frac{1}{2} {}^s\dot{\boldsymbol{\omega}}^T {}^s\mathbf{I}_{sat} {}^s\boldsymbol{\omega} - {}^s\mathbf{m}_{pm}^T {}^s\dot{\mathbf{B}}_E = {}^s\boldsymbol{\omega}^T {}^s\mathbf{N}_{ctrl} - {}^s\mathbf{m}_{pm}^T {}^s\dot{\mathbf{B}}_E \quad (7.10)$$

Using Eq. (7.3) and (7.4) it is possible to rewrite Eq. (7.10) to:

$$\begin{aligned} \dot{E}_{tot}({}^s\boldsymbol{\omega}, \theta_{bpm}) &= {}^s\boldsymbol{\omega} \cdot \left( -\mathbf{C}_d {}^s\dot{\mathbf{B}}_E + {}^s\mathbf{m}_{pm} \right) \times {}^s\mathbf{B}_E - {}^s\mathbf{m}_{pm}^T {}^s\dot{\mathbf{B}}_E \\ &= \left( -\mathbf{C}_d {}^s\dot{\mathbf{B}}_E + {}^s\mathbf{m}_{pm} \right) \cdot {}^s\mathbf{B}_E \times {}^s\boldsymbol{\omega} - {}^s\mathbf{m}_{pm}^T {}^s\dot{\mathbf{B}}_E \\ &= -\mathbf{C}_d {}^s\dot{\mathbf{B}}_E^T {}^s\dot{\mathbf{B}}_E + {}^s\mathbf{m}_{pm}^T {}^s\dot{\mathbf{B}}_E - {}^s\mathbf{m}_{pm}^T {}^s\dot{\mathbf{B}}_E \\ &= -\mathbf{C}_d {}^s\dot{\mathbf{B}}_E^T {}^s\dot{\mathbf{B}}_E \\ &= -\mathbf{C}_d \left( {}^s\mathbf{B}_E \times {}^s\boldsymbol{\omega} \right)^T \left( {}^s\mathbf{B}_E \times {}^s\boldsymbol{\omega} \right) \\ &= -\mathbf{C}_d {}^s\boldsymbol{\omega}^T \underline{\mathbf{S}}({}^s\mathbf{B}_E)^T \underline{\mathbf{S}}({}^s\mathbf{B}_E) {}^s\boldsymbol{\omega} \end{aligned} \quad (7.11)$$

where:

$\underline{\mathbf{S}}(\cdot)$  is a skew symmetric matrix representing the cross product operator.

The matrix  $\underline{\mathbf{S}}({}^s\mathbf{B}_E)^T \underline{\mathbf{S}}({}^s\mathbf{B}_E)$  is positive semidefinite, which makes the derivative of the Lyapunov candidate function negative semidefinite as long as the control gain  $\mathbf{C}_d > 0$ , which indicates that the chosen control law does dissipate energy as expected, with  ${}^s\boldsymbol{\omega} = \mathbf{0}$  and  $\theta_{bpm} = 0$  as equilibrium point.

Using the Krasovskii-LaSalle invariance theorem, defined in [79], it is possible to prove that this equilibrium is globally asymptotically stable. If  ${}^s\mathbf{B}_E \times {}^s\boldsymbol{\omega} = \mathbf{0}$  for all  $t > t_0$ , then the vectors are either constantly parallel or  ${}^s\boldsymbol{\omega} = \mathbf{0}$ . Since  ${}^s\mathbf{B}_E$  depends on  ${}^s\boldsymbol{\omega}$  and the orbit position and  ${}^s\boldsymbol{\omega}$  only depends on the dynamic equation of motion. The control torque is equal to zero if  ${}^s\mathbf{B}_E$  and  ${}^s\boldsymbol{\omega}$  are parallel. Then the vectors  ${}^s\mathbf{B}_E$  and  ${}^s\boldsymbol{\omega}$  will not be parallel all the time and it is therefore possible to conclude that the largest invariant set is  ${}^s\boldsymbol{\omega} = \mathbf{0}$  and  $\theta_{bpm} = 0$ .

It is important to notice that the detumble control law utilizes that the changing local geomagnetic field, measured by the satellite, changes during the polar orbit, which ensures that angular velocities about all axes are decreased.

### 7.3 Implementation for AAUSAT3

A discrete B-dot filter and control law is implemented in software and the combined software and hardware block diagram for a single SBRF axis is illustrated in Figure 7.2. The block diagram is identical for the other two axes and the total torque generated by the magnetorquers is the sum of the three control torque vectors  ${}^s\mathbf{N}_{ctrl}$ . The direction of the magnetic moment generated by the magnetorquer is determined by its placement in the SBRF.

<sup>1</sup>The largest change in local geomagnetic field strength is approximately  $(48000 [nT] - 18000 [nT])/0.25 [orbit] = 20 [nT/s]$ , which is assumed much less than potential changes caused by a rotating satellite.

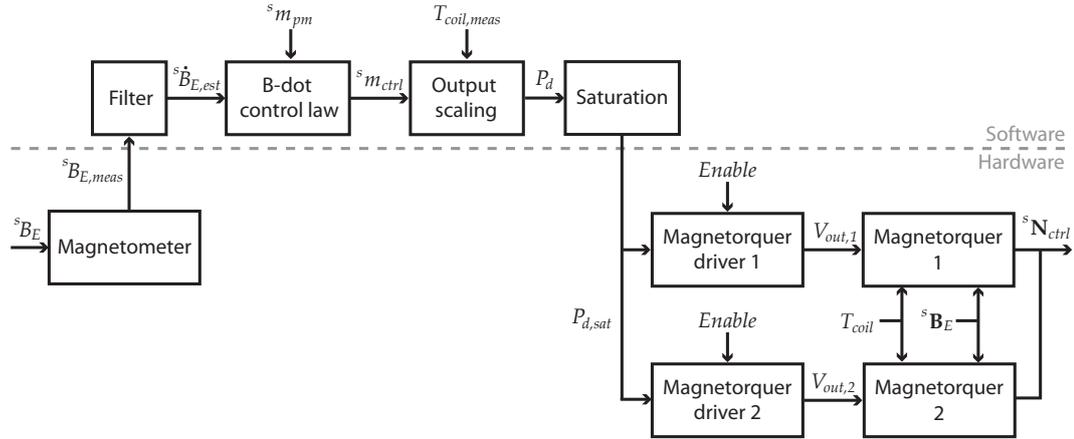


Figure 7.2: Block diagram of detumble control implementation for a single axis.

Each axis has two magnetorquers for redundancy and these are enabled separately. The magnetorquers will only be enabled with a duty-cycle of 88% to ensure periodic measurement of the local geomagnetic field and actuation, see Section 3.4. The magnetorquer drivers are given a PWM signal  $P_{d,sat}$ , with a limited duty-cycle between 37.5% to 62.5%. This ensures that the voltage does not exceed  $\pm 1.25$  [V] (using a 5 [V] supply, see Subsection 3.4.3) and thus keeps the current in the wire within a safety margin. The sampling frequency of the magnetometer is set to 10 [Hz], which is the highest setting for the chosen magnetometer. Additional details on sensor and actuator modeling can be found in Section 4.6 and 4.7 respectively.

The B-dot controller outputs a desired magnetic moment to be produced by the magnetorquers. This magnetic moment is converted to a corresponding PWM signal for the magnetorquer drivers in the output scaling block presented in Figure 7.2. The block first calculates the desired voltage  $V_{des}$  based on the calculated control moment  ${}^s m_{ctrl}$  using Eq. (3.10) from Section 3.4

$$V_{des} = \frac{{}^s m_{ctrl} R_{mt}}{n_w A_{mt}} \quad (7.12)$$

The number of windings  $n_w$  and the area enclosed by the magnetorquer  $A_{mt}$  are constant. However, the Coil resistance  $R_{mt}$  is temperature dependent and the output scaling block has therefore been prepared to accept temperature as input, which allows for estimation of  $R_{mt}$ . Since it is desirable to keep the detumble controller as simple as possible, a constant temperature of 30 [C°] is given as input, to make it independent of temperature measurements. This gives a slight deviation between the desired magnitude of the magnetic moment and the actual magnetic moment generated by the magnetorquers and simulations are therefore performed with actual coil temperatures of 85, 30 and -25 [C°], see Section 7.4. The temperatures 85 and -25 [C°] are the expected worst case temperatures used during design of

the magnetorquers<sup>2</sup>. The voltage is then converted to a PWM signal by first adding 5 [V] to  $V_{des}$  and multiplying this with 10. 0 [V] then equals 50% PWM duty-cycle, 1.25 [V] equals 62.5% PWM duty-cycle and -1.25 [V] equals 37.5% PWM duty-cycle as desired.

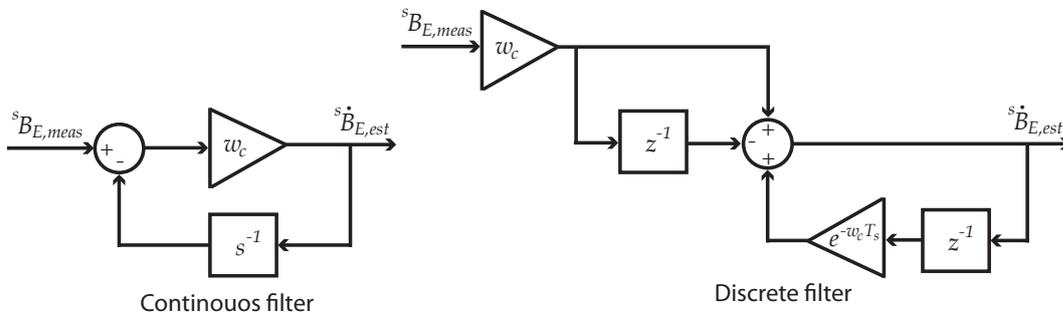
The filter block estimates the derivative of the Earth's magnetic field measured in the SBRF. This is done with a state variable filter, which acts as a high pass filter with variable cutoff frequency. The state variable filter was also used on AAU CubeSat and AAUSAT-II and the transfer function for the continuous filter is

$$\frac{Y(s)}{X(s)} = \frac{w_c s}{s + w_c} = \frac{w_c}{1 + w_c s^{-1}} \quad (7.13)$$

where  $w_c$  is the cutoff frequency. The discrete equivalent is found using Matlab's `c2d` function with the zero order hold method and a sampling time  $T_s$  of 0.1 [s], which gives

$$\frac{Y(z)}{X(z)} = \frac{w_c z - w_c}{z - e^{-w_c T_s}} = \frac{w_c - w_c z^{-1}}{1 - e^{-w_c T_s} z^{-1}} \quad (7.14)$$

Figure 7.3 illustrates the continuous and discrete state variable filter, where it is clear that a very large  $w_c$  will make the  $e^{-w_c T_s}$  term go to zero and a small  $w_c$  will make the current output dependent on the previous output, thus damping the effect of noise in the measurements.  $w_c$  also acts as a gain on the input.

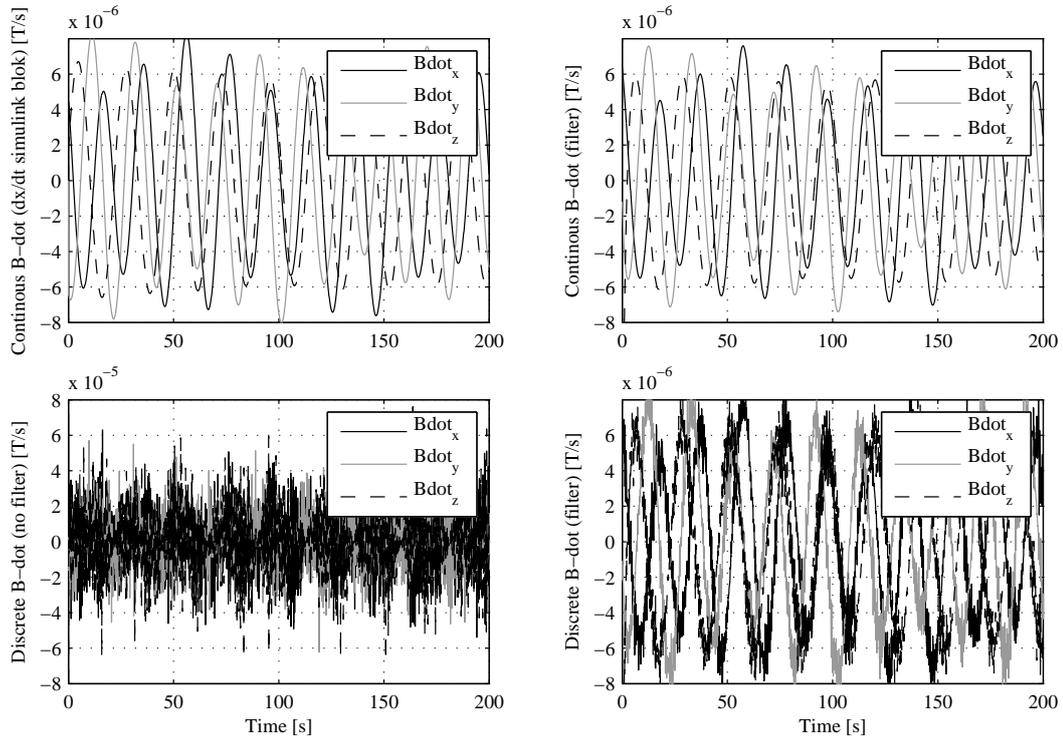


**Figure 7.3:** Continuous and discrete state variable filter.

$w_c$  is set to 0.7 [rad/s] based on experience from AAU CubeSat and AAUSAT-II. Simulations with both the continuous and discrete filter are shown in Figure 7.4 together with numerical simulations without filtering. The measured magnetic field vector was added with white Gaussian noise in the discrete simulations, see Section 4.6.

The simulation results from the two continuous cases are almost similar, except for a small attenuation in the filtered signal. However, the filter property is desirable as indicated by the two discrete simulations with sensor noise, where the numerical differentiation without

<sup>2</sup>Note that the average temperature of 30 [C°] should be updated when a more precise value is known for AAUSAT3



**Figure 7.4:** *B-dot filter simulations with a satellite angular velocity of  $(10,10,10)$  [deg/s].*

filtering deviates considerably from the continuous case without sensor noise, as this noise gives spikes in the discrete differentiation.

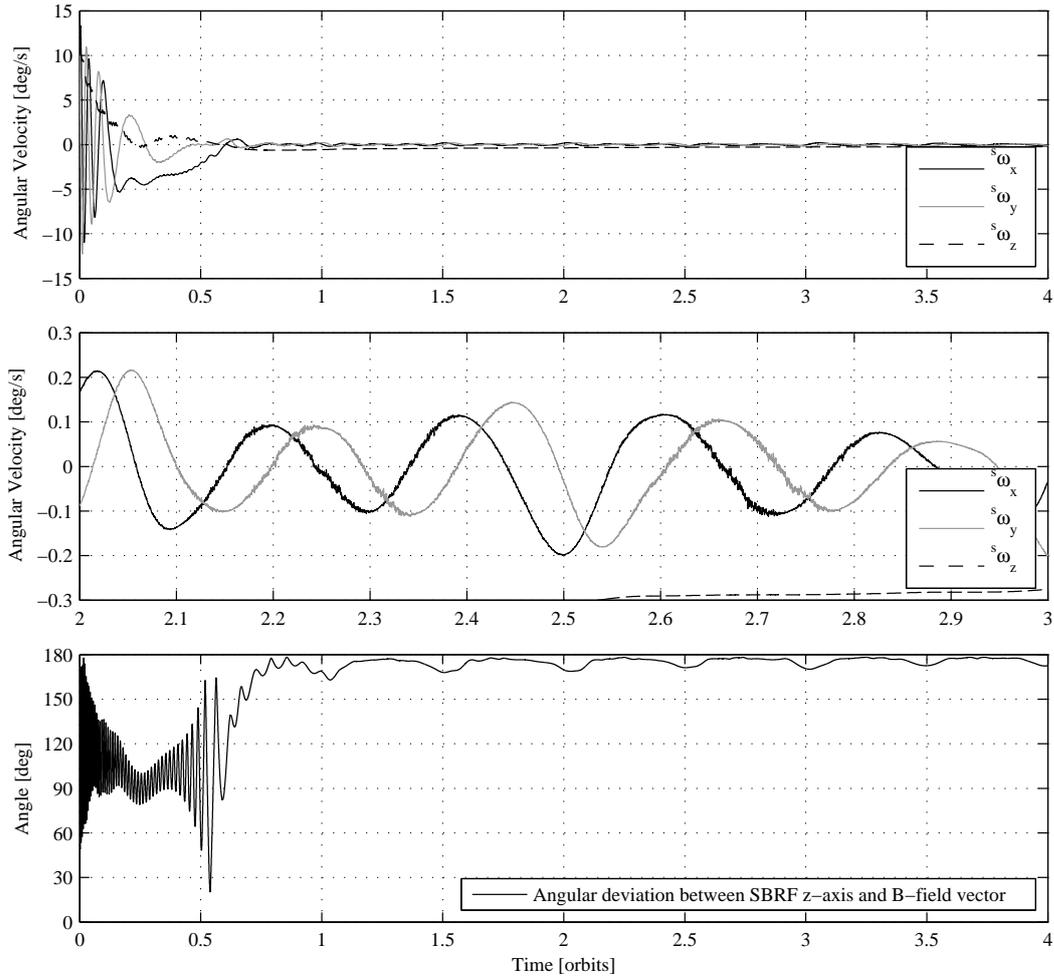
Calculating the derivative of the magnetic field can give problems with outliers in the data. The outliers could be caused by inconsistent sample time in the real time operating system and possible data glitches. In the final implementation for AAUSAT3 it is therefore suggested to add outlier detection and removal before the state variable filter. A method for detection of outliers could be to save  $N$  samples back in time (moving data window) and continuously calculate the sample mean or median of the data and reject the measurement if it is not within a certain threshold, e.g.  $\pm 3$  standard deviations or a predetermined value. More elaborate threshold selection methods are discussed in e.g. [80]. The rejected data could then be replaced by the sample mean, the median or the previous measurement. This means that only outliers are modified, while nominal data is left unaltered.

The B-dot controller gain  $C_d$  is set to  $(6000,6000,6000)$  based on iterative simulation. This gain meets the requirement while keeping the power consumption low, which is also indicated by the simulations presented in Section 7.4.

## 7.4 Simulation Evaluation

The performance of the detumble controller is evaluated in the AAUSAT3 Matlab Simulink simulation environment, where the system in Figure 7.2 has been added.

Figure 7.5 shows the B-dot simulation results with the chosen  $w_c$  and  $C_d$ .



**Figure 7.5:** B-dot simulation results with an initial angular velocity of  $(10,10,10)$  [deg/s] and a coil temperature of  $30$  [C°]. Graph 1 shows the angular velocity during the first four orbits. Graph 2 shows the angular velocity between orbit two and three. Graph 3 shows the angle between the  ${}^s z$ -axis and the magnetic field vector  ${}^s\mathbf{B}_E$ .

The angular velocity is reduced by the B-dot controller (see Graph 1 in Figure 7.5) and stays within  $(\pm 0.3, \pm 0.3, \pm 0.3)$  [deg/s] as required (see Graph 2 in Figure 7.5). The wobble causing the angular velocity to switch between positive and negative values, as shown in Graph 1 and 2, is mostly caused by an asymmetric mass distribution, but also by the permanent magnet. The z-axis wobbles less because the mass is symmetrically distributed

around this axis. Graph 3 shows how the z-axis of the SBRF is pointed approximately 180 [deg] away from the Earth's magnetic field vector, due to the permanent magnet. The angle deviates slightly every half orbit, which is coherent with the satellite passing equator, where the largest change in the direction of the magnetic field vector occurs.

Simulations have shown that the B-dot controller is stable up to initial angular velocities of (500,500,500) [deg/s], which corresponds to a magnitude of 866 [deg/s]. This is more than the required 720 [deg/s]. The sampling frequency is too slow for higher angular velocities. However, the initial angular velocity is expected to be less than 10 [deg/s], which is much slower than the tested limit. At the high angular velocities it is necessary to set  $w_c$  high and it has thus been multiplied with e.g. 10000, which gives a higher gain in the B-dot filter and makes the output dependency on the previous output negligible.

The above results and simulation results at other temperatures are summarized in Table 7.1.

Simulation case			Detumble time	Power consumption [mW]			
$\omega_0$ [deg/s]	$T_{coil}$ [C°]	$w_c$ [rad/s]	$t_{detum}$ [orbits]	Orbit 1	Orbit 2	Orbit 3	Orbit 4
(10,10,10)	85	0.7	2.75	38.4	31.5	31.4	31.3
(10,10,10)	30	0.7	2.54	39.8	32.5	32.4	32.3
(10,10,10)	-25	0.7	1.77	42.1	34.1	33.9	33.9
(500,500,500)	30	7000	11.92*	109.1	109.1	109.1	109.1

**Table 7.1:** B-dot simulation results. Detumble time is the amount of orbits the B-dot controller uses to despin the satellite from the initial angular velocity to below ( $\pm 0.3, \pm 0.3, \pm 0.3$ ) [deg/s] (\*in the last simulation it was set to ( $\pm 0.4, \pm 0.4, \pm 0.4$ ) [deg/s]). Power consumption is the total average of all coils including coil drivers who consume approximately 26.4 [mW].

The slowest detumbling time from (10,10,10) [deg/s] to below ( $\pm 0.3, \pm 0.3, \pm 0.3$ ) [deg/s] is 2.75 [orbits] and experienced when the coil temperature is highest (high temperature equals high resistance, which gives a smaller current and a smaller magnetic moment). This is within the requirement and the initial angular velocity corresponds to a magnitude of 17.3 [deg/s], which is slightly higher than the requirement. The satellite is detumbled within 11.92 [orbits] in the high angular velocity simulation, which is also within the requirement. The power consumption is considered acceptable and drops when the satellite is detumbled, but it will not go lower than 26.4 [mW], which is the power loss in the drivers.

Table 7.2 has been made by extensive simulations and gives an idea about the robustness of the detumble controller. The detumble controller was simulated over a prolonged period to see if it was capable of reducing the angular velocity over time with a reduced set of magnetic actuators. The initial angular velocity in all simulations was (10,10,10) [deg/s].

Losing all coils in one axis will only make detumbling slower and the detumble controller will even be capable of ensuring good radio communication with only a working z-axis coil.

An idea would be to accept a slower detumble time in the startup phase, by using only half of the magnetorquer coils (there are two coils in each axis). This will potentially save the other set of coils and drivers, if malfunctioning software accidentally burns the coils or the drivers, by not saturating the PWM signal appropriately.

Active actuators	Number stable axes				$^s z$ -axis aligned with $^s \mathbf{B}_E$
	0	1	2	3	
0	+				
p	+*				
x			+		
y			+		
z			+		+
xp			+		
yp			+		
zp			+		+
xy				+	+
xz				+	+
yz				+	+
xyp				+	+
xzp				+	+
yzp				+	+
xyz				+	+
xyzp				+	+

\*The permanent magnet alone cannot dissipate energy.

**Table 7.2:** Angular velocity stability over time with the chosen B-dot control law in the AAUSAT3 simulation environment. All simulations were also performed on a mass symmetrical satellite showing similar results.  $x,y,z$  represent the three coils and  $p$  is the permanent magnet.

## 7.5 Summary

The primary task of the ADCS is detumbling. A B-dot control law has been chosen for this task and stability analysis showed that the control law does dissipate energy and detumbles the satellite. Simulations furthermore showed that the satellite was detumbled from 10 to below  $\pm 0.3$  [ $deg/s$ ] in the three SBRF axes within the specified time of three orbits. Simulations also showed that the B-dot control law, using a sample frequency of 10 [ $Hz$ ] on the magnetometer, can detumble the satellite from angular velocities of up to 866 [ $deg/s$ ].

Ideas for the software implementation was given in Section 7.3, which included the recommendation of adding measurement outlier detection and removal for improved robustness and reliability.

Besides providing angular velocity stability, the B-dot controller also provides two axis attitude stability relative to the local geomagnetic field vector, because of the permanent magnet and has shown to be stable even in the absence of one of the three perpendicular magnetorquer coils.

## Three Axis Attitude Stabilization with Magnetic Actuation

According to the requirements, stated in Section 1.6, a pointing controller should be designed with the purpose of tracking reference attitudes. These reference attitudes are ground station tracking and slower maneuvers like orbit frame tracking also referred to as nadir pointing. Nadir pointing is especially interesting, since the GPS patch antenna has to be pointed towards the GPS satellites continuously for up to 12-15 [*min*] during acquisition (cold start).

The detumble controller, designed in Chapter 7, does ensure two axis attitude stability with respect to the local geomagnetic field vector, because of the permanent magnet acting like a compass needle. This automatically ensures that the highest gain of the VHF (AIS) and UHF (COM) antennas points down towards the Earth over the poles, which covers the main mission of the satellite and pointing control or three axis attitude stabilization is therefore not needed in the majority of the time. When needed, it can be divided into two subtasks:

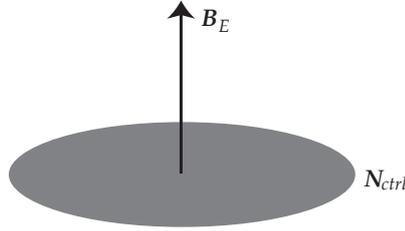
- The most important task is to ensure reception of GPS signals by tracking the orbit frame. This will also point the Camera down towards the Earth.
- A less important task is to do ground station tracking for optimized data transfer during passes.

This chapter introduces problems associated with magnetic actuation and looks into the possibility of using nonlinear Model Predictive Control (MPC) for three axis attitude control.

### 8.1 Satellite Control with Magnetic Actuation

Having only magnetic actuation makes three axis attitude stabilization difficult, since the control torque vector is constrained to lie in a plane perpendicular to the local geomagnetic

field vector, which is illustrated in Figure 8.1.



**Figure 8.1:** The local geomagnetic field vector  $\mathbf{B}_E$  and the perpendicular plane that contains all the possible control torque vectors  $\mathbf{N}_{ctrl}$ .

This means that the satellite is only controllable in two axes at any time instant, i.e. rotation about the two axes that are perpendicular to the local geomagnetic field is controllable whereas the rotation about the axis parallel to the field is uncontrollable. This is also clear by first rewriting the equation giving the control torque  $\mathbf{N}_{ctrl}$  as the cross product between the magnetic moment vector of the actuators  $\mathbf{m}_{ctrl}$  and the local geomagnetic field vector  $\mathbf{B}_E$ .

$$\mathbf{N}_{ctrl} = \mathbf{m}_{ctrl} \times \mathbf{B}_E = \underline{\mathbf{S}}(\mathbf{B}_E)^T \mathbf{m}_{ctrl} = \begin{bmatrix} 0 & B_E(3) & -B_E(2) \\ -B_E(3) & 0 & B_E(1) \\ B_E(2) & -B_E(1) & 0 \end{bmatrix} \mathbf{m}_{ctrl} \quad (8.1)$$

The skew symmetric matrix  $\underline{\mathbf{S}}(\mathbf{B}_E)^T$  has rank two and thus spans  $\mathbb{R}^2$ , which makes it impossible to apply a torque in arbitrary direction. However, the local geomagnetic field changes direction during the polar orbit (or in general inclined orbits) as illustrated in Figure 3.2 in Section 3.3, which makes the satellite three axis attitude stabilizable over time, even though it is not three axis controllable at a single point in time. Stability of systems influenced by an approximately periodic local geomagnetic field has been proven in e.g. [76].

Linear approaches for three axis attitude control exploiting the periodicity of the local geomagnetic field have seen extensive use, see e.g. [76, 81, 82, 83, 84]. However, their use are limited to cases where it is possible to guarantee that the satellite is in close proximity of the reference. The Ørsted satellite utilizes the long extended boom, which gives a gravity gradient torque that helps the satellite stay close to the reference. The gravity gradient torque is, however, very small for an almost mass symmetrical CubeSat. Orbit frame tracking is achieved in [81] using a predictive horizon approach with the linearized satellite model and a periodic approximation of the local geomagnetic field. However, this was only shown for initial attitudes close to the reference and with the help of a momentum wheel.

Another group at Aalborg University (group 10gr832) is working on a constant gain controller based on a linearized model in the reference. The focus of this thesis is therefore devoted to the development of a globally stabilizing controller for attitude acquisition. Examples of globally stabilizing controllers for attitude acquisition with constrained magnetic actuation are sparse. In [76] global stability is achieved both with a sliding mode control

and an energy approach. It is also shown in [85] that conditioned state feedback can give global stability.

Model Predictive Control (MPC) in either its linear or nonlinear form has, to the best knowledge of the authors, not been attempted for three axis attitude stabilization of satellites only controlled by magnetic actuation. MPC inherits optimality from optimal control, while providing the possibility of taking both input and output constraints into consideration in the optimization problem. MPC also provides a flexible objective function and has in its linear form been used for three axis attitude stabilization by e.g. [86, 81], however, not in cases with magnetic actuation only.

In the rest of this chapter it is first shown that finite horizon linear MPC with re-linearization can control the satellite in the case of three independent controls (thruster example) and how this setup fails when only magnetic actuation is available. These results are then extended to nonlinear MPC for time-varying systems, where the requirements to such a controller are identified for the case of attitude acquisition for small satellites with magnetic actuation. Finally, a discussion is presented on possible control solutions for AAUSAT3.

## 8.2 Model Predictive Control

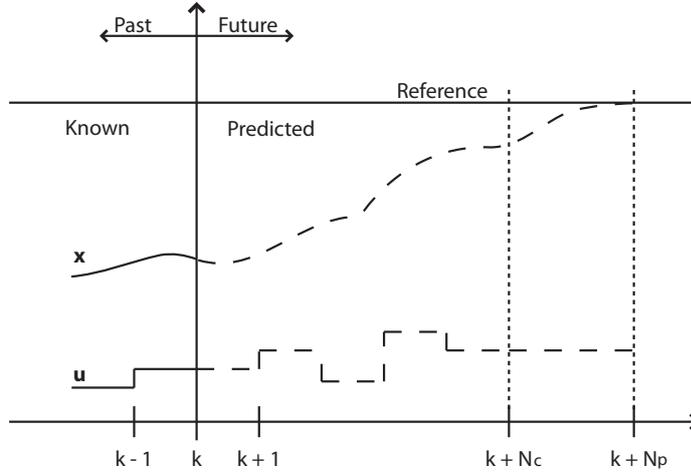
The general idea of MPC is to solve a finite horizon optimal control problem, which may be subjected to constraints. For this thesis the on-line based solution is described. Another approach exist called explicit MPC, where the parameter space is divided into polyhedral partitions, which each contains an optimal state feedback control law. Hence, controlling a system becomes a matter of finding the optimal state feedback control law in a lookup table [86]. The explicit MPC method will, however, not be described further in this thesis.

The main principle of MPC is illustrated in Figure 8.2, where the MPC utilize measurements obtained from a given system, at the present time  $k$ , to predict the future dynamic behavior of the system over a prediction horizon  $N_p$ . Furthermore, a number of inputs  $\mathbf{u}$  up to a control horizon  $N_c \leq N_p$  are determined such that a given open-loop performance objective is optimized.

Under the assumption that there is no disturbances or mismatch between the model and the real system it is possible to apply all inputs obtained at time  $k$  to the system [87]. However, this is in general not possible and hence only the first input is applied to the system. At time  $k + 1$  a new prediction is made over the entire prediction horizon, using measurements obtained at time  $k + 1$ , and a new set of optimal control inputs are determined. For this reason MPC is called receding horizon control.

For the MPC illustrated in Figure 8.2 the input  $\mathbf{u}$  is held constant, at the value determined for time  $k + N_c - 1$ , after the end of the control horizon. Another possibility is to set the input to zero after the end of the control horizon or having  $N_c = N_p$ .

In general, it is desirable to chose a long prediction horizon as well as a long control horizon, but a long horizon will increase the computational demands to solve the optimization problem. The complexity of the optimization problem can be reduced by choosing  $N_c < N_p$ .



**Figure 8.2:** Basic principle of Model Predictive Control, where  $\mathbf{x}$  is the state vector,  $\mathbf{u}$  is the control input vector,  $N_c$  is the control horizon and  $N_p$  is the prediction horizon.

An example of a linear, time invariant system is used in the following, in order to give a more thorough description of the MPC method. The system is presented in discrete time, which will also be the case for the remainder of this Chapter. The discrete linear time invariant state space model is given as

$$\mathbf{x}_{k+1} = \underline{\mathbf{A}}\mathbf{x}_k + \underline{\mathbf{B}}\mathbf{u}_k \quad (8.2)$$

where:

$\mathbf{x}_k$  is the state vector.

$\mathbf{u}_k$  is the control input vector.

It is assumed that the state vector is measurable and  $\underline{\mathbf{C}}$  is equal to the identity matrix. Using the system described in Eq. (8.2) it is desired to have the state track a given reference  $\mathbf{r}_k$  using the cost function  $J(\underline{\mathbf{U}}, \mathbf{x}_k, \mathbf{r}_k)$  defined as

$$J(\underline{\mathbf{U}}, \mathbf{x}_k, \mathbf{r}_k) = \sum_{n=0}^{N_p-1} [(\mathbf{x}_{k+n|k} - \mathbf{r}_{k+n})^T \underline{\mathbf{Q}}(\mathbf{x}_{k+n|k} - \mathbf{r}_{k+n})] + \sum_{n=0}^{N_c-1} [\mathbf{u}_{k+n}^T \underline{\mathbf{R}}\mathbf{u}_{k+n}] \quad (8.3)$$

where:

$\underline{\mathbf{U}} = \{\mathbf{u}_k \ \mathbf{u}_{k+1} \ \dots \ \mathbf{u}_{k+N_c-1}\}$ .

$\underline{\mathbf{Q}}$  is the state cost matrix.

$\underline{\mathbf{R}}$  is the input cost matrix.

Hence, for the current state  $\mathbf{x}_k$  the MPC solves the following objective function

$$V(\mathbf{x}_k, \mathbf{r}_k) = \min_{\underline{\mathbf{U}}} J(\underline{\mathbf{U}}, \mathbf{x}_k, \mathbf{r}_k) \quad (8.4)$$

subjected to

$$\mathbf{x}_{k|k} = \mathbf{x}_k \quad (8.5)$$

$$\mathbf{x}_{k+n+1|k} = \underline{\mathbf{A}}\mathbf{x}_{k+n|k} + \underline{\mathbf{B}}\mathbf{u}_{k+n}, \quad n = 0, 1, \dots, N_p - 1 \quad (8.6)$$

The  $\underline{\mathbf{Q}}$  and  $\underline{\mathbf{R}}$  matrices are tuning parameters just like in a LQ control problem. If  $\underline{\mathbf{Q}}$  is positive semidefinite and  $\underline{\mathbf{R}}$  is positive definite, then it is possible to guarantee existence and uniqueness of a solution to the optimal control problem [88].

One of the key advantages of MPC is that it is relatively easy to include constraints into the optimization problem. In general, most systems have a constraint on the input, e.g. a maximum current of 1 [A]. There may even be constraint in the rate of change for the input, e.g. a valve can not be opened instantaneously. It is also common to have constraints on the states and the output. By including the previous mentioned constraints, the objective function is then subjected to

$$\mathbf{x}_{k|k} = \mathbf{x}_k \quad (8.7)$$

$$\mathbf{x}_{k+n+1|k} = \underline{\mathbf{A}}\mathbf{x}_{k+n|k} + \underline{\mathbf{B}}\mathbf{u}_{k+n}, \quad n = 0, 1, \dots, N_p - 1 \quad (8.8)$$

$$\Delta\mathbf{u}_{k+n} = \mathbf{u}_{k+n} - \mathbf{u}_{k+n-1}, \quad n = 0, 1, \dots, N - 1 \quad (8.9)$$

$$\Delta\mathbf{u}_{min} \leq \Delta\mathbf{u}_{k+n} \leq \Delta\mathbf{u}_{max}, \quad n = 0, 1, \dots, N - 1 \quad (8.10)$$

$$\mathbf{u}_{min} \leq \mathbf{u}_{k+n} \leq \mathbf{u}_{max}, \quad n = 0, 1, \dots, N - 1 \quad (8.11)$$

$$\mathbf{x}_{min} \leq \mathbf{x}_{k+n|k} \leq \mathbf{x}_{max}, \quad n = 1, \dots, N \quad (8.12)$$

Subjected to constraints, the optimal control problem may be infeasible. It is possible to overcome this by introducing slack variables in the constraints, making them soft (soft constraints), or by penalizing undesired behavior using the state cost matrix  $\underline{\mathbf{Q}}$  and the input cost matrix  $\underline{\mathbf{R}}$ .

When looking at stability it is possible to state that feasibility implies stability in an infinite horizon control problem. This is, however, not the case for a finite horizon control problem, as optimality within the finite horizon does not necessarily imply stability, since each optimization does not care about the time after the prediction horizon, which could bring the state into a position where it is not possible to stabilize the system. In [88, 89, 87] it is shown how a terminal equality constraint ( $\mathbf{x}_{k+N_p|k} - \mathbf{r}_{k+N_p} = \mathbf{0}$ ) can ensure stability of the finite horizon control problem, when assuming that this terminal constraint can be met within the finite horizon. It is also shown how this can be relaxed by a terminal constrained set (terminal region) and a terminal cost. Adding the terminal cost matrix  $\underline{\mathbf{P}}$  to the cost function gives

$$\begin{aligned} J(\underline{\mathbf{U}}, \mathbf{x}_k, \mathbf{r}_k) &= \sum_{n=0}^{N_p-1} \left[ (\mathbf{x}_{k+n|k} - \mathbf{r}_{k+n})^T \underline{\mathbf{Q}} (\mathbf{x}_{k+n|k} - \mathbf{r}_{k+n}) \right] + \sum_{n=0}^{N_c-1} \left[ \mathbf{u}_{k+n}^T \underline{\mathbf{R}} \mathbf{u}_{k+n} \right] \\ &+ \left( \mathbf{x}_{k+N_p|k} - \mathbf{r}_{k+N_p} \right)^T \underline{\mathbf{P}} \left( \mathbf{x}_{k+N_p|k} - \mathbf{r}_{k+N_p} \right)^T \end{aligned} \quad (8.13)$$

The last term in the cost function is an approximation of the cost from  $N_p$  to  $\infty$ , which makes the cost function approximate the cost function in the infinite horizon control problem. This term must be chosen appropriately to ensure stability of the closed loop system and a procedure for finding  $\underline{\mathbf{P}}$  is given in [89]. Although the terminal constraint is able to stabilize a given system it may still be undesirable as optimization problems are often easier to solve

if they only have constraints on the input. Fortunately, [88] has shown that it is possible to ensure that a sufficiently large terminal cost implicitly satisfies the terminal constraint in some cases.

Linear MPC is a widely use control method, especially within the process industry, however, many systems are nonlinear [87, 89]. The main difference between linear and nonlinear MPC is that convexity is ensured for liner MPC, which is not the case for nonlinear MPC. Non-convex problems are difficult to solve and it is even more difficult to anticipate how long an optimization step will take to complete. Finally, if a solution is found, it may only be suboptimal [89]. Another issue is that system-theoretic properties of nonlinear MPC, such as stability, is hard to guarantee, whereas the subject is well addressed within linear MPC theory [87].

A common way to overcome the problems of convexity and stability issues, associated with nonlinear MPC, is to re-linearize the nonlinear system model in each operating point and use the linearized model inside the prediction horizon [89].

### 8.3 Linear MPC with Re-Linearization

Linear MPC with re-linearization utilize a linearization of the nonlinear model in the working point to propagate the state vector  $\mathbf{x}_k$ , measured at time  $k$ , to the next time instant  $k + 1$  using the control input solution  $\mathbf{u}_k$  to the optimization problem found using linear MPC. At each time step the nonlinear model is linearized around the measured state vector. The linearized model is then used in the linear MPC to find a solution to the given optimization problem using the current measured state vector as the initial state.

The state vector used in the linear system only contains the vector part of the quaternion, which was also the case in Section 6.1.

$$\mathbf{x}_k = \begin{bmatrix} {}^c\mathbf{q}_{1:3,k}^T & {}^c\boldsymbol{\omega}_k^T \end{bmatrix}^T \quad (8.14)$$

Where:

${}^c\mathbf{q}_{1:3,k}$  is the vector part of the quaternion representing the rotation from the ECI to the CRF.

${}^c\boldsymbol{\omega}_k$  is the angular velocity of the CRF relative to the ECI, given in the CRF.

This means that the quaternion state is propagated by addition and not quaternion multiplication, which is valid as long as  $N_p$  and  $T_s$  are small and the system dynamic is slow (small angular velocities). Finally the permanent magnet is removed from the system as the torque, originating from the permanent magnet, can easily be compensated for in the controller.

The linearized discrete time-invariant state space model for AAUSAT3 is given as (all states are measurable)

$$\mathbf{x}_{k+1} = \underline{\mathbf{A}}\mathbf{x}_k + \underline{\mathbf{B}}\mathbf{u}_k \quad (8.15)$$

The state transition matrix for the discrete time system  $\underline{\mathbf{A}}$  is given as an approximated discretization, where  $\underline{\mathbf{A}} \approx \mathbf{1} + \underline{\mathbf{A}}_c T_s$  and the linearized continuous time model is derived in

Appendix G and summarized here

$$\underline{\mathbf{A}}_c(t) = \begin{bmatrix} -\underline{\mathbf{S}}({}^c\bar{\boldsymbol{\omega}}(t)) & \\ \underline{\mathbf{0}}_{3 \times 3} & {}^c\mathbf{I}_{sat}^{-1}[\underline{\mathbf{S}}({}^c\mathbf{I}_{sat} {}^c\bar{\boldsymbol{\omega}}(t)) - \underline{\mathbf{S}}({}^c\bar{\boldsymbol{\omega}}(t)){}^c\mathbf{I}_{sat}] \end{bmatrix} \quad (8.16)$$

where:

${}^c\bar{\boldsymbol{\omega}}(t)$  is the working point (nominal) angular velocity of CRF relative to the ECI, given in the CRF.

$\underline{\mathbf{1}}_{3 \times 3}$  is a 3x3 identity matrix.

${}^c\mathbf{I}_{sat}$  is the inertia matrix of the satellite given in the CRF.

### 8.3.1 Unconstrained Satellite Attitude Acquisition using Thrusters

In the following a linear MPC with re-linearization is implemented in the Matlab Simulink simulation environment made for AAUSAT3 (see Chapter 4), which is able to ensure attitude acquisition when the satellite actuators have full controllability at any time instance. Full controllability is ensured by the use of thrusters. The reference of the satellite is given in the ECI for simplicity of the implementation, however, the reference is easily given in other frames as it is only a matter of rotating the reference into the desired frame.

As a linearized model is used within the linear MPC the prediction horizon  $N_p$ , as well as the sampling time  $T_s$ , must be kept small in order not to make large deviations from the working point. As  $N_p$  is maintained small the control horizon  $N_c$  is set equal to the prediction horizon  $N_c = N_p$ . For the following a prediction horizon of  $N_p = 10$  and a sampling time of  $T_s = 5$  [s] is chosen.

The cost function for a reference tracking controller is given in Eq. (8.13). As  $N = N_c = N_p$  it is possible to simplify Eq. (8.13) to

$$\begin{aligned} J(\underline{\mathbf{U}}, \mathbf{x}_k, \mathbf{r}_k) &= \sum_{n=0}^{N-1} \left[ (\mathbf{x}_{k+n|k} - \mathbf{r}_{k+n})^T \underline{\mathbf{Q}} (\mathbf{x}_{k+n|k} - \mathbf{r}_{k+n}) + \mathbf{u}_{k+n}^T \underline{\mathbf{R}} \mathbf{u}_{k+n} \right] \\ &+ (\mathbf{x}_{k+N|k} - \mathbf{r}_{k+N})^T \underline{\mathbf{P}} (\mathbf{x}_{k+N|k} - \mathbf{r}_{k+N}) \end{aligned} \quad (8.17)$$

The satellite is actuated by thrusters in this implementation and the control input is therefore a torque given in the CRF and the input sequence is denoted  $\underline{\mathbf{U}} = \{\mathbf{N}_k \ \mathbf{N}_{k+1} \ \dots \ \mathbf{N}_{k+N-1}\}$ . This gives an input matrix  $\underline{\mathbf{B}}$  which is equal to

$$\underline{\mathbf{B}} = \begin{bmatrix} \underline{\mathbf{0}}_{3 \times 3} \\ {}^c\mathbf{I}_{sat}^{-1} \end{bmatrix} \quad (8.18)$$

The optimization problem  $V(\mathbf{x}_k, \mathbf{r}_k)$  is given as

$$V(\mathbf{x}_k, \mathbf{r}_k) = \min_{\underline{\mathbf{U}}} J(\underline{\mathbf{U}}, \mathbf{x}_k, \mathbf{r}_k) \quad (8.19)$$

subjected to

$$\mathbf{x}_{k|k} = \begin{bmatrix} {}^c\bar{\boldsymbol{\omega}}_k^T \\ {}^c\bar{\mathbf{q}}_{1:3,k}^T \end{bmatrix} \quad (8.20)$$

$$\mathbf{x}_{k+n+1|k} = \underline{\mathbf{A}}\mathbf{x}_{k+n|k} + \underline{\mathbf{B}}\mathbf{u}_{k+n}, \quad n = 0, 1, \dots, N_p - 1 \quad (8.21)$$

The implementation of the linear MPC with re-linearization using thrusters for actuation is located in the Matlab file `mpc_relin_sdp.m` on the appended CD. The previous mentioned Matlab file can be tested using the Matlab Simulink file `NMPC_N.mdl`, which is also found on the appended CD. The implementation is based on the following procedure:

1. **Setup initial parameters:** Initial parameters such as the cost matrices  $\underline{\mathbf{Q}}$ ,  $\underline{\mathbf{R}}$  and  $\underline{\mathbf{P}}$ , the horizon  $N$ , the sampling time  $T_s$  and the Inertia matrix  ${}^c\mathbf{I}_{sat}$  must be chosen.
2. **Obtain initial states:** The initial state vector, for the given implementation, is measured in the SBRF and rotated to the CRF:  $\mathbf{x}_{k|k} = \left[ \left( {}^s\mathbf{q}_k \otimes {}^c\mathbf{q} \right)_{1-3}^T \quad \left( \underline{\mathbf{A}} ({}^c\mathbf{q})^s \omega_k \right)^T \right]^T$ .
3. **Set the tracking reference:** The tracking reference is given in the SBRF and rotated to the CRF:  $\mathbf{r}_{k|k} = \left[ \left( {}^s\mathbf{q}_{k,ref} \otimes {}^c\mathbf{q} \right)_{1-3}^T \quad \left( \underline{\mathbf{A}} ({}^c\mathbf{q})^s \omega_{ref} \right)^T \right]^T$ .
4. **Linearize nonlinear model in working point:** The continuous time system model  $\underline{\mathbf{A}}_c$  is linearized in the working point, where  $\underline{\mathbf{A}}_c$  is given in Eq. (8.16).
5. **Setup optimization solver:** The implementation relies on the YALMIP<sup>1</sup> toolbox for Matlab. As the optimization problem, given in Eq. (8.19), is a quadratic optimization problem, the `solvesdp` function is setup to use `quadprog` as solver.
6. **Setup optimization variables:** The `solvesdp` function uses its own set of variables, which is constructed using the `sdpvar` function. The implementation relies on the input parameter  $\underline{\mathbf{U}}$  given as  $\underline{\mathbf{U}} = \{ {}^c\mathbf{N}_k \quad {}^c\mathbf{N}_{k+1} \quad \dots \quad {}^c\mathbf{N}_{k+N-1} \}$ .
7. **Setup cost and objective function:** The cost function is constructed in accordance with Eq. (8.17) subjected to Eq. (8.20) and (8.21), but without constraints. The discrete time system, given in Eq. (8.21), is approximated using  $\underline{\mathbf{A}} \approx \mathbf{1} + \underline{\mathbf{A}}_c T_s$ .
8. **Solve optimization problem:** The optimization problem is solved using the `solvesdp` function.
9. **Apply first input:** The optimal input is given in the CRF and is therefore rotated to the SBRF  ${}^s\mathbf{N}_k = \underline{\mathbf{A}} ({}^c\mathbf{q}^{-1}) {}^c\mathbf{N}_k$ . This control torque is then applied to the system.
10. **Repeat 2 to 10 for next time instant  $k + 1$**

The `solvesdp` function, despite its name, solves a number of different programming problems, e.g. linear-, quadratic-, semidefinite- and nonlinear programming problems. The `solvesdp` function essentially provides a standard interface for a number of commercial and free solvers, from which `quadprog` is one of them [90]. The `quadprog` function is Matlab's own build in optimization algorithm, which is able to solve quadratic programming problems given in the form [91]

$$\min_{\mathbf{x}} \frac{1}{2} \mathbf{x}^t \underline{\mathbf{H}} \mathbf{x} + \mathbf{c}^T \mathbf{x} \quad (8.22)$$

<sup>1</sup>YALMIP is a free toolbox for Matlab, which is able to solve both convex and non-convex optimization problems using commercial and free external solvers [90].

where:

$\mathbf{x}$  is a  $n$ -dimensional vector of decision variables.

$\mathbf{H}$  is a  $n \times n$  symmetric matrix containing the coefficients of the quadratic terms in the cost function.

$\mathbf{c}$  is a  $n$ -dimensional vector with the coefficients of the linear terms in the cost function.

which can be subjected to linear inequality and equality constraints

$$\mathbf{Ax} \leq \mathbf{b} \quad (8.23)$$

$$\mathbf{Ex} = \mathbf{d} \quad (8.24)$$

where:

$\mathbf{A}$  is a  $m \times n$  matrix defining the inequalities.

$\mathbf{b}$  is a  $m$ -dimensional vector with the right-hand-side coefficients of the inequalities.

$\mathbf{E}$  is a  $k \times n$  matrix defining the equalities.

$\mathbf{d}$  is a  $k$ -dimensional vector with the right-hand-side coefficients of the equalities.

The state cost matrix  $\mathbf{Q}$ , the input cost matrix  $\mathbf{R}$  and the terminal cost matrix  $\mathbf{P}$  are set to the following values based on iterative testing (*diag* refers to diagonal matrix):

$$\mathbf{Q} = \text{diag} [1 \quad 1 \quad 1 \quad 1000 \quad 1000 \quad 1000] \quad (8.25)$$

$$\mathbf{R} = \text{diag} [1 \quad 1 \quad 1] 10^6 \quad (8.26)$$

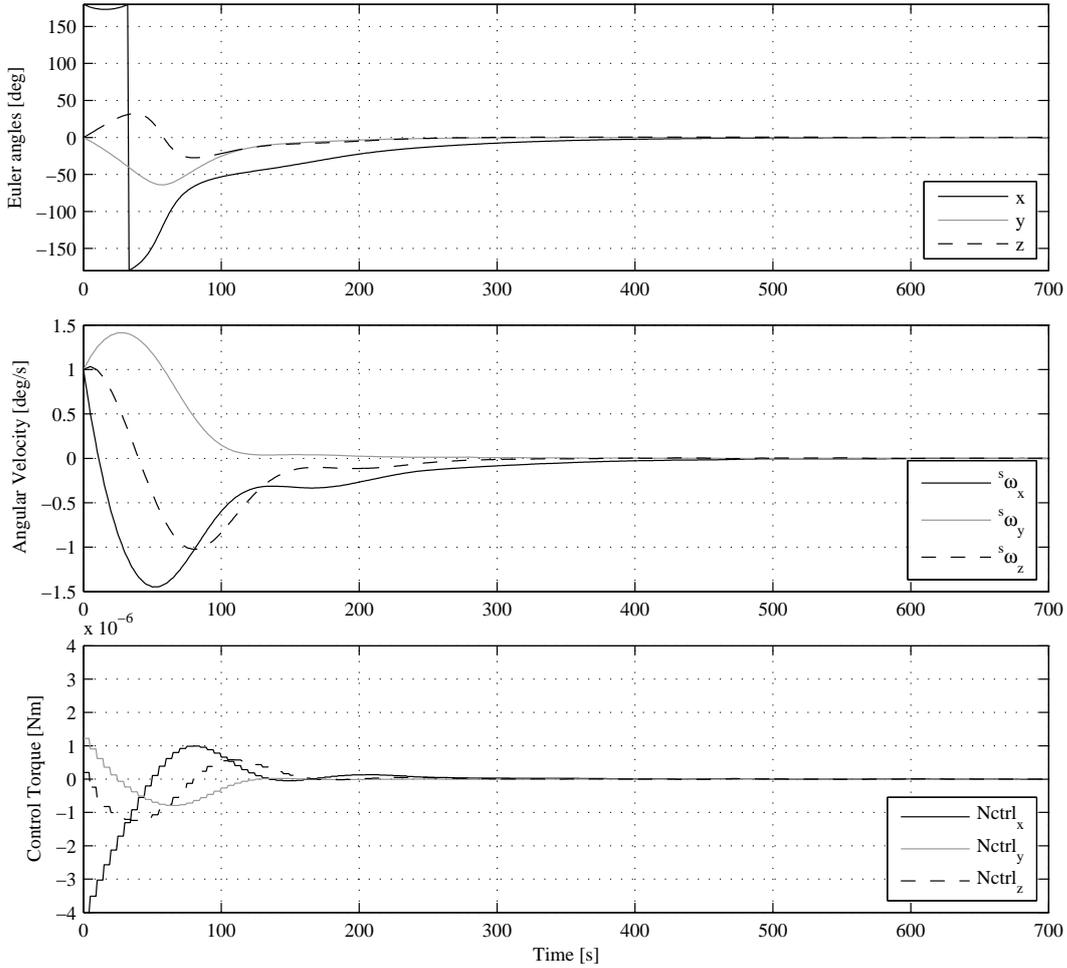
$$\mathbf{P} = \text{diag} [1 \quad 1 \quad 1 \quad 1000 \quad 1000 \quad 1000] 10^2 \quad (8.27)$$

The input is weighted relatively high in order to simulate realistic torques and the implementation has been tested using an initial attitude set to  ${}^s\mathbf{q} = [1 \quad 0 \quad 0 \quad 0]^T$  and an initial angular velocity of  ${}^s\boldsymbol{\omega} = [1 \quad 1 \quad 1]^T$  [deg/s]. The reference attitude was set to  ${}^s\mathbf{q}_r = [0 \quad 0 \quad 0 \quad 1]^T$  and the reference angular velocity was set to  ${}^s\boldsymbol{\omega}_r = [0 \quad 0 \quad 0]^T$  [deg/s]. The result is presented in Figure 8.3 and shows an example of linear MPC with re-linearization that globally stabilizes a satellite actuated with thrusters.

Another implementation based on the method utilized in [85] has also been tested. The method is also describe in [88], where it is referred to as a brute force method. The results of the brute force method is nearly identical with the results obtained using the `solvesdp` function from the YALMIP toolbox. However, the implementation based on the brute force method shows better performance in terms of computation time. Preliminary studies have shown, that the higher computation time, of the implementation using `solvesdp`, is caused by the overhead required by the function. Methods to reduce the overhead are described in the documentation for the YALMIP toolbox [90], however, this has not been investigated further.

### 8.3.2 Constrained Satellite Attitude Acquisition using Magnetic Actuation

The thrusters in the above implementation are, in the following, replaced with magnetic actuation. All the above assumptions still holds using the same horizon  $N = N_c = N_p = 10$  and sampling time  $T_s = 5$  [s]. However, an extra assumptions is made. It is assumed that

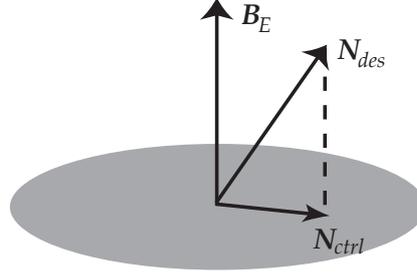


**Figure 8.3:** Performance of linear MPC with re-linearization using thrusters for actuation. Graph 1 shows the Euler angles representing the rotation between the ECI and the SBRF (attitude error). Graph 2 shows the angular velocity of the SBRF relative to the ECI, given in the SBRF. The bottom graph shows the control torque given in the SBRF.

the local geomagnetic field vector measured in the SBRF is constant within the horizon, which is valid for a small  $N$  and  $T_s$ , and if the system dynamic is slow.

It is, as already mentioned, only possible to actuate in the plane perpendicular the local geomagnetic field vector when magnetic actuation is utilized. One possibility is to design a controller, which gives an unconstrained desired control torque  $\mathbf{N}_{des}$  and then project this torque onto the plane, which is perpendicular to the local geomagnetic field vector. This approach has been used in e.g. [76, 81] and is illustrated in Figure 8.4.

The possible control torque vector  $\mathbf{N}_{ctrl}$  can be calculated as [76]



**Figure 8.4:** Projection of the desired control torque  $N_{des}$  onto the plane perpendicular to the local geomagnetic field vector  $B_E$ , giving the possible control torque vector  $N_{ctrl}$ .

$$N_{ctrl} = \underline{S}(B_E)^T m_{ctrl} = \underline{S}(B_E)^T \frac{\underline{S}(B_E)N_{des}}{\|B_E\|^2} [Am^2] \quad (8.28)$$

where:

$\underline{S}(\cdot)$  is a skew symmetric matrix.

$m_{ctrl}$  is the control magnetic moment vector to be applied to the magnetorquers.

The above mentioned property can easily be incorporated as a constraint in MPC, which is one on the main advantages of MPC. The optimization problem given in Eq. (8.19) is thus furthermore subjected to

$${}^c B_E \cdot {}^c u_{k+n} = 0, \quad n = 0, 1, \dots, N-1 \quad (8.29)$$

$${}^c u_{min} \leq {}^c u_{k+n} \leq {}^c u_{max}, \quad n = 0, 1, \dots, N-1 \quad (8.30)$$

where:

$u_{min}$  and  $u_{max}$  are the lower and upper bound on the control input.

Notice that the control input is given as a magnetic moment, hence  ${}^c u_{min} = {}^c m_{ctrl,min}$ ,  ${}^c u_{k+n} = {}^c m_{ctrl,k+n}$  and  ${}^c u_{max} = {}^c m_{ctrl,max}$ . The equality constraint given in Eq. (8.29) ensures that the control input  ${}^c m_{ctrl,k+n}$  is always perpendicular on  ${}^c B_E$ , which is also the most power optimal input.

The previous mentioned implementation has been simulated using the same parameters as for the unconstrained satellite, however, the objective function, given in Eq. (8.19), was furthermore subjected to the constraints given in Eq. (8.29) and (8.30). The results of the simulation showed that the controller were not able to stabilize the satellite, which is expected as the satellite is only controllable around two axes at a time instant. However, as mentioned in Section 8.1 it is possible to stabilize the satellite using the periodicity of the geomagnetic field.

In order to take advantage of the periodicity of the geomagnetic field it is clear that a longer horizon, of preferably an entire orbit, is needed. The re-linearization approach can hence not be utilized as the assumption of a small horizon is invalid. This means that the assumption of a constant geomagnetic field within the horizon is also invalid and that nonlinear MPC with a time varying geomagnetic field within the horizon is necessary to ensure global attitude acquisition using magnetic actuation.

## 8.4 Nonlinear MPC with Time Varying Magnetic Field

The following presents an implementation of a nonlinear MPC with a time varying geomagnetic field, which has been implemented in the Matlab Simulink simulation environment made for AAUSAT3 (see Chapter 4). The objective of the implementation is to ensure attitude acquisition for a satellite using magnetic actuation only.

For the nonlinear MPC implementation the full state is utilized

$$\mathbf{x}_k = \begin{bmatrix} {}^c\mathbf{q}_k^T & {}^c\boldsymbol{\omega}_k^T \end{bmatrix}^T \quad (8.31)$$

The discrete nonlinear time varying model for the satellite is given as (all states are measurable)

$$\mathbf{x}_{k+1} = \underline{\mathbf{A}}_k \mathbf{x}_k + \underline{\mathbf{B}}_k \mathbf{u}_k \quad (8.32)$$

The discrete nonlinear time varying model is approximated using  $\underline{\mathbf{A}} \approx \mathbf{1} + \underline{\mathbf{A}}_c T_s$ , where the continuous nonlinear time varying model  $\underline{\mathbf{A}}_c$  is derived in Appendix C and presented in matrix form here

$$\underline{\mathbf{A}}_c = \begin{bmatrix} \frac{1}{2} \underline{\boldsymbol{\Omega}}({}^c\boldsymbol{\omega}(t)) & \mathbf{0}_{4 \times 3} \\ \mathbf{0}_{3 \times 4} & -{}^c\mathbf{I}_{sat}^{-1} \underline{\mathbf{S}}({}^c\boldsymbol{\omega}(t)) {}^c\mathbf{I}_{sat} \end{bmatrix} \quad (8.33)$$

where:

$\underline{\boldsymbol{\Omega}}({}^c\boldsymbol{\omega}(t))$  is a  $4 \times 4$  skew symmetric matrix, defined in Eq. (C.10).

The discrete time varying input matrix  $\underline{\mathbf{B}}_k$  is given as

$$\underline{\mathbf{B}}_k = \begin{bmatrix} \mathbf{0}_{4 \times 3} \\ {}^c\mathbf{I}_{sat}^T \underline{\mathbf{S}}({}^c\mathbf{B}_{E,k})^T \end{bmatrix} \quad (8.34)$$

The determination of the time varying geomagnetic field requires the information saved in the Two Line Element (TLE) format. The TLE used in the implementation is taken from AAUSAT-II, which is expected to have similar properties and orbital parameters as AAUSAT3 (see Section 1.4). The TLE is utilized in the Simplified General Perturbation (SGP4) algorithm to obtain the position of the satellite. The SGP4 algorithm is implemented in the C-file `sgp4m.c`, located on the appended CD, and returns the position of the satellite given in the ECI. The calculation of the geomagnetic field is then performed using the International Geomagnetic Reference Model (IGRF). The IGRF is implemented in the C-file `igrfm.c` and is located on the appended CD.

The cost function of the nonlinear MPC implementation is the same as for the linear MPC implementation given in Eq. (8.17), where the horizon is  $N = N_c = N_p$ . The objective function is given in Eq. (8.19) and is subjected to the constraints given in Eq. (8.29), (8.30), (8.31) and (8.32).

The implementation of the nonlinear MPC with a time varying geomagnetic field using magnetorquers for actuation is located in the Matlab file `nmpc_constrained_sdp.m` on the

appended CD. The previous mentioned Matlab file can be tested using the Matlab Simulink file `NMPC_m.mdl`, which is also found on the appended CD. The implementation is based on the following procedure:

1. **Setup initial parameters:** Initial parameters such as the cost matrices  $\underline{\mathbf{Q}}$ ,  $\underline{\mathbf{R}}$  and  $\underline{\mathbf{P}}$ , the horizon  $N$ , the sampling time  $T_s$ , the Inertia matrix  ${}^c\mathbf{I}_{sat}$  and the TLE must be chosen.
2. **Obtain initial states:** The initial state vector, for the given implementation, is measured in the SBRF and rotated to the CRF:  $\mathbf{x}_{k|k} = \left[ \left( {}^s\mathbf{q}_k \otimes {}^c\mathbf{q} \right)^T \quad \left( \underline{\mathbf{A}} \left( {}^c\mathbf{q} \right)^s \omega_k \right)^T \right]^T$ .
3. **Set the tracking reference:** The tracking reference is given in the SBRF and rotated to the CRF:  $\mathbf{r}_{k|k} = \left[ \left( {}^s\mathbf{q}_{k,ref} \otimes {}^c\mathbf{q} \right)^T \quad \left( \underline{\mathbf{A}} \left( {}^c\mathbf{q} \right)^s \omega_{ref} \right)^T \right]^T$ .
4. **Setup optimization solver:** The implementation relies on the YALMIP toolbox for Matlab. As the implementation solves a constrained nonlinear optimization problem, the `solvesdp` function is setup to use `fmincon` as solver.
5. **Setup optimization variables:** The `solvesdp` function uses its own set of variables, which is constructed using the `sdpvar` function. The implementation relies on the input parameter  $\underline{\mathbf{U}}$ , given as  $\underline{\mathbf{U}} = \{ {}^c\mathbf{m}_k \quad {}^c\mathbf{m}_{k+1} \quad \dots \quad {}^c\mathbf{m}_{k+N-1} \}$ .
6. **Determine the geomagnetic field vector:** The position of the satellite is obtained with the `sgp4m` function and rotated into the ECEF using the `eci2ecef` function. The geomagnetic field vector  ${}^i\mathbf{B}_E$  is then calculated using the `igrfm` and the `eci2ecef` function and finally rotated to the CRF:  ${}^c\mathbf{B}_E = \underline{\mathbf{A}} \left( {}^c\mathbf{q} \right) {}^i\mathbf{B}_E$ .
7. **Setup cost and objective function:** The cost function is constructed in accordance with Eq. (8.17) subjected to Eq. (8.29), (8.30), (8.31) and (8.32).
8. **Solve optimization problem:** The optimization problem is solved using the `solvesdp` function.
9. **Apply first input:** The optimal input is given in the CRF and is therefore rotated to the SBRF  ${}^s\mathbf{m}_k = \underline{\mathbf{A}} \left( {}^c\mathbf{q}^{-1} \right) {}^c\mathbf{m}_k$ . This magnetic moment is then applied to the system through the magnetorquers.
10. **Repeat 2 to 10 for next time instant  $k + 1$**

The `fmincon` function is a general purpose optimization solver included in Matlab, which is able to solve constrained nonlinear optimization problems, as mentioned in the above. The `fmincon` function is able to find the minimum of a problem given as [91]

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad (8.35)$$

where:

$\mathbf{x}$  is a  $n$ -dimensional vector of decision variables.

$f(\mathbf{x})$  can be a nonlinear function.

which can be subjected to

$$\underline{\mathbf{A}}\mathbf{x} \leq \mathbf{b} \quad (8.36)$$

$$\underline{\mathbf{E}}\mathbf{x} = \mathbf{d} \quad (8.37)$$

$$c(\mathbf{x}) \leq 0 \quad (8.38)$$

$$c_{eq}(\mathbf{x}) = 0 \quad (8.39)$$

where:

$\underline{\mathbf{A}}$  is a  $n \times n$  matrix defining the inequalities.

$\mathbf{b}$  is a  $m$ -dimensional vector with the right-hand-side coefficients of the inequalities.

$\underline{\mathbf{E}}$  is a  $n \times n$  matrix defining the equalities.

$\mathbf{d}$  is a  $m$ -dimensional vector with the right-hand-side coefficients of the equalities.

$c(\mathbf{x})$  and  $c_{eq}(\mathbf{x})$  are optional functions that define constraints.

The discrete approximation of the transition matrix requires a normalization of the quaternion in order to maintain unity, however, it is experienced that the square root operation, used in the quaternion normalization, causes the `solvesdp` function in the YALMIP toolbox to fail. The square root function is a non convex monotonically increasing function, which causes the `solvesdp` function to fail due to a build-in convexity analysis. YALMIP offers another square root function `sqrtn`, which do not require convexity and is solvable using general purpose optimization solvers, such as Matlab's `fmincon` function [90]. However, tests has shown that the `sqrtn` function causes `solvesdp` to crash due to an unknown problem in the solver and hence quaternion normalization is omitted in the following in order to test the remaining part of the implementation.

The implementation has been tested using the same parameters (initial quaternion, initial angular velocity and reference) given in the test of the linear MPC with re-linearization. The sample time was, however, changed to 1 [s], because of the quaternion normalization problems. The horizon were varied from  $N = 2$  to  $N = 128$  to test the computation time for one iteration  $k$  to  $k + 1$  of the implementation. The results are presented in Table 8.1.

Horizon:	Computation time: [s]
2	0.48
4	1.76
8	8.55
16	10.75
32	17.26
64	40.99
128	177.80

**Table 8.1:** Average computation time per iteration (mean of 10 iterations) for the implemented nonlinear MPC with time varying geomagnetic field for different horizons in Matlab on a single core Pentium 4 (2.8 [GHz], 1 [GB] RAM).

The results shows that the implementation of a nonlinear MPC with a time varying geomagnetic field is inappropriate for use in ACS due to the computation time of large horizons.

Simulations also shows that the solver performed only a few prediction steps before terminating, indicating that it could have found a local minimum. It is important to notice that the optimization time for a non-convex system is non-deterministic and hence the computation time could be worse than presented in Table 8.1.

Furthermore, none of the tests were able to globally stabilize the satellite, which is expected for a horizon considerably smaller than one orbit. Due to the large computation time it is not possible to conclude whether or not the given implementation is able to ensure global stability of the satellite. A larger horizon will potentially make the nonlinear MPC able to ensure global stability, however, the complexity of the given implementation is too computational demanding for use in small CubeSat sized satellites with limited computational power, such as AAUSAT3.

Converting the problem into an explicit solution may reduce the complexity and provide a more deterministic computation time compared to the on-line solution, however, the need for faster optimizers seems inevitable. The optimizer may even have to be costume made, in order to exploit the structure of the dynamic system of a satellite, to solve the optimization problem faster.

As mentioned previously, the implementations utilize functions from YALMIP's toolbox, which converts the control problem to comply with the external solvers. This process gives a large computational overhead, which should be avoided in a final implementation. Furthermore the implementation relies on the full model of both SGP4 and IGRF which should be replaced by simpler and less accurate implementations, however, the computation time of these algorithms are much smaller than the computation time spend on the optimization.

The use of quaternion operations such as quaternion multiplication increases the complexity considerably, hence choosing another attitude parameters may be beneficial. The scalar part of the quaternion could alternatively be omitted in the state vector, which removes the difficulties associate with the quaternion operations. However, this can only be used as long as the quaternion does not deviate to much from the initial value, which is hard to guarantee during attitude acquisition.

## 8.5 Summary

This chapter discussed three axis satellite attitude stabilization with magnetic actuation. More precisely the aim was to do global attitude acquisition and investigate whether it would be possible to use MPC in this regard.

A simulating of a linear MPC implementation using re-linearization showed attitude acquisition in a case with thruster actuation, using only a horizon of 10 steps with a sample time of 5 [s]. However, extending this to the constrained case with magnetic actuation showed unstable behavior. Extending the horizon to utilize the periodic geomagnetic field in the orbit was not possible, since the satellite deviates considerably from the working point used in the linearization.

Using a nonlinear MPC implementation with a time varying local geomagnetic field introduced problems associated with nonconvex optimization and showed to be too computationally demanding for reasonable sized horizons. Ideas was finally given on how to improve the implementation.

# Chapter 9

## Closure

This thesis considered a number of aspects related to satellite ADCS and in particular in relation to the AAUSAT3 project. This chapter summarizes the results and conclusions given in the different chapters and gives recommendations for future work.

### 9.1 Conclusion

The following conclusions are made on the accomplishments of this project:

- The Keplerian orbital elements, describing a satellite in orbit around the Earth, has been presented and it was made probable that a satellite in a Low Earth Orbit (LEO) around the Earth can be considered as a two-body problem. The dynamics of a rigid body satellite and the disturbances effecting it have furthermore been describe, where quaternions are the chosen attitude parameterization, because they do not suffer from singularities.
- Different types of sensors and actuators and their advantages/disadvantages have been discussed and the chosen sensors for AAUSAT3 are: Sun sensors, magnetometers and angular rate gyroscopes. Two types of actuators are chosen: Magnetorquers and a permanent magnet. The main properties for both sensors and actuators are that they are small in size, reliable and low cost. An ADCS hardware prototype have been produced adhering to the interface control document and the hardware redundancy requirement. The prototype makes it possible to test the chosen sensors and actuators and to test the developed algorithms on the chosen microcontrollers before the final satellite flight model is made.
- A simulation environment for AAUSAT3 has been implemented in Matlab Simulink based on previous implementations for the North Sea Observer (NSO) satellite and AAUSAT-II. The 4th order Simplified General Perturbations (SGP4) orbit propagator, the International Geomagnetic Reference Field (IGRF) model, the eclipse model and

the Earth albedo model were reused from the previous implementation, however, the IGRF model have been upgraded to the 10th generation (IGRF10). The theory of both the eclipse and the Earth albedo model have been described and it was argued that the complexity of the Earth albedo model should be reduced for implementation on the satellite. Sensor and actuator models for AAUSAT3 have additionally been derived and implemented in the simulation environment. The implementations are gathered in a Simulink library providing reusable blocks and high degree of flexibility, which makes the library usable for future satellites as well.

- The SVD-method solving Wahba's problem, the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) have been described. Focus have been put in making an UKF implementation for AAUSAT3, where the SVD-method have been utilized to give the UKF a good attitude start guess and it was argued that the SVD-method additionally could be utilized for sanity checks of the UKF. Four ADS implementations were made for AAUSAT3: SVD-method, UKF without bias estimation, UKF with bias estimation and UKF with bias estimation and without the rotation from the Satellite Body Reference Frame (SBRF) to the Control Reference Frame (CRF). It have been shown that bias estimation is important and that inertia estimation is unnecessary for small symmetrically shaped satellites. Furthermore, it was shown that omitting rotations from the SBRF to the CRF did not reduce the computation time significantly, however, it have been shown that it is possible to reduce the computation time considerable, by reducing the number of Runge Kutta substeps, without considerable loss of accuracy. Simulations showed that an accuracy of approximately  $\pm 4$  [deg] outside eclipse and approximately  $\pm 9$  [deg] in eclipse can be expected with the low cost of the shelf sensor setup chosen for AAUSAT3.
- The implementation of a B-dot controller has been presented with focus on the implementation method and a stability analysis. Simulations showed that the satellite could be detumbled from 10 [deg/s] to below  $\pm 0.3$  [deg/s] within three orbits. It have furthermore been shown that the satellite can be detumble from angular velocities of up to 866 [deg/s] and that full detumbling is ensured even in the absence of one of the three perpendicular magnetorquer coils. Having a permanent magnet also ensures two axis attitude stability relative to the local geomagnetic field vector. This will ensure that the highest gain of the antennas are automatically pointed downwards at the North Pole, but also over Greenland and Aalborg.
- Different approaches for three axis stabilization using magnetic actuation have briefly been discussed and it was decided that Model Predictive Control (MPC) should be investigated as a possibility for attitude acquisition. A general introduction of the MPC method have therefore been given, followed by a linear MPC implementation with re-linearization. Simulations showed that linear MPC with re-linearization is able to globally stabilize the satellite with thrusters as actuation, however, when the constraints of magnetic actuation is introduced (torques can only be produced perpendicular to the local geomagnetic field) it have not been possible to obtain stability with a short prediction horizon. Ideas for a nonlinear MPC implementation using the periodicity of the geomagnetic field have therefore been presented, but working

with optimization of nonlinear cost functions have proven difficult. The quaternion multiplication, used in the propagation of the quaternion, in the cost function when comparing the state with the reference and in the rotations of the geomagnetic field, significantly increases the complexity of the optimization problem.

The ADCS can, as a concluding remark, detumble the satellite, provide two axis attitude stability relative to the local geomagnetic field and determine the attitude of the satellite. Another group have additionally developed a linear controller for attitude tracking, however, work is still required before global attitude acquisition, addressed in this thesis, is achieved and all requirements are met.

## 9.2 Recommendations

The following recommendations are made based on the experiences gained through this project:

- Nonlinear MPC for satellites with magnetic actuation can ensure global 3-axis stability if a long horizon of e.g. one orbit is used, since this will incorporate the periodicity of the geomagnetic field in the optimization problem. However, the complexity of the optimization problem requires faster solvers or faster microcontrollers. Satellites relying completely on high pointing accuracy should consider using magnetic actuation supported by momentum wheels or a single spin wheel as done in e.g. [81]. Alternatively another control method could be used such as static attitude and rate feedback, which can, according to [85], ensure global stability. It can be questioned whether the requirements for the ADCS are too strict compared to the AAUSAT3 mission and that the requirements could be relaxed, since a maneuver like ground station tracking can be problematic with only magnetic actuation.
- Matlab and Simulink are excellent tools for simulating dynamic systems. However, the main disadvantage of using Matlab and Simulink is that two implementations must be made: One for simulation and one for the microcontroller onboard the satellite. Matlab fortunately includes a function, which converts Matlab files into C-files. This function could be used when implementing the Matlab code on AAUSAT3, however, some of the algorithms have already been written in C-code. Alternatively algorithms should only be written in e.g. C-code and tested directly on the ADCS microcontrollers. Tests could hence be performed using a “hardware in the loop” approach, where the ADCS related algorithms are running on the actual hardware and the satellite environment is simulated in Matlab. An advantage of this approach is that errors caused by the numerical precision is discovered earlier.



# Quaternions Defined and Quaternion Algebra

Quaternions are well suited as rotation operators and eases the deviation of the kinematic equations of the satellite. This Appendix presents quaternion definitions and some of its fundamental algebraic properties. The Appendix is based on [38] and proofs have been omitted, but can be found in this source.

## A.1 Definitions

The quaternion  $\mathbf{q}$  is a hyper complex number composed of a scalar  $q_4$  and a vector  $\mathbf{q}_{1:3}$ , with components spanning  $\mathbb{R}^3$ .

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_{1:3} \\ q_4 \end{bmatrix} = \mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 + q_4 \quad (\text{A.1})$$

The quaternion parameters can also be written  $(q_1, q_2, q_3, q_4)$ . It is an element of  $\mathbb{R}^4$  and mathematically not defined in ordinary linear algebra. The hyper imaginary numbers  $\mathbf{i}$ ,  $\mathbf{j}$  and  $\mathbf{k}$  must satisfy the following conditions

$$\begin{aligned} \mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \\ \mathbf{ij} = \mathbf{k} = -\mathbf{ji} \\ \mathbf{jk} = \mathbf{i} = -\mathbf{kj} \\ \mathbf{ki} = \mathbf{j} = -\mathbf{ik} \end{aligned} \quad (\text{A.2})$$

also named the special products (not dot products). The complex conjugate of the quaternion is equal to

$$\mathbf{q}^* = (-\mathbf{q}_{1:3} + q_4) = -\mathbf{i}q_1 - \mathbf{j}q_2 - \mathbf{k}q_3 + q_4 \quad (\text{A.3})$$

The norm (sometimes called length) of the quaternion is a scalar defined as

$$|\mathbf{q}| = \sqrt{\mathbf{q}^* \mathbf{q}} = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} \quad (\text{A.4})$$

## A.2 Addition and Multiplication

Addition of quaternions follow the associative and commutative laws and the sum is given by adding the corresponding components as

$$\mathbf{q} + \mathbf{p} = \mathbf{i}(q_1 + p_1) + \mathbf{j}(q_2 + p_2) + \mathbf{k}(q_3 + p_3) + (q_4 + p_4) \quad (\text{A.5})$$

It is more complicated to calculate the quaternion product. The commutative law does not apply and the conditions stated in A.2 must be satisfied. Multiplying two quaternions  $\mathbf{q}$  and  $\mathbf{p}$  gives<sup>1</sup>

$$\begin{aligned} \mathbf{q} \otimes \mathbf{p} &= (\mathbf{i}q_1 + \mathbf{j}q_2 + \mathbf{k}q_3 + q_4)(\mathbf{i}p_1 + \mathbf{j}p_2 + \mathbf{k}p_3 + p_4) \\ &= \mathbf{i}^2 q_1 p_1 + \mathbf{i}\mathbf{j}q_1 p_2 + \mathbf{i}\mathbf{k}q_1 p_3 + \mathbf{i}q_1 p_4 \\ &\quad + \mathbf{j}\mathbf{i}q_2 p_1 + \mathbf{j}^2 q_2 p_2 + \mathbf{j}\mathbf{k}q_2 p_3 + \mathbf{j}q_2 p_4 \\ &\quad + \mathbf{k}\mathbf{i}q_3 p_1 + \mathbf{k}\mathbf{j}q_3 p_2 + \mathbf{k}^2 q_3 p_3 + \mathbf{k}q_3 p_4 \\ &\quad + \mathbf{i}q_4 p_1 + \mathbf{j}q_4 p_2 + \mathbf{k}q_4 p_3 + q_4 p_4 \end{aligned} \quad (\text{A.6})$$

By using the defined special products in Eq. (A.2) and rearranging terms, Eq. (A.6) can be rewritten to

$$\begin{aligned} \mathbf{q} \otimes \mathbf{p} &= \mathbf{i}q_1 p_4 - \mathbf{j}q_1 p_3 + \mathbf{k}q_1 p_2 - q_1 p_1 \\ &\quad \mathbf{i}q_2 p_3 + \mathbf{j}q_2 p_4 - \mathbf{k}q_2 p_1 - q_2 p_2 \\ &\quad - \mathbf{i}q_3 p_2 + \mathbf{j}q_3 p_1 + \mathbf{k}q_3 p_4 - q_3 p_3 \\ &\quad \mathbf{i}q_4 p_1 + \mathbf{j}q_4 p_2 + \mathbf{k}q_4 p_3 + q_4 p_4 \end{aligned} \quad (\text{A.7})$$

and using the algebra of matrices and Eq. (A.7), the quaternion product can be put in the following form

---

<sup>1</sup> $\otimes$  is used as a symbol for quaternion multiplication.

$$\begin{aligned}
 \mathbf{q} \otimes \mathbf{p} = \mathbf{r} &= \mathbf{i}r_1 + \mathbf{j}r_2 + \mathbf{k}r_3 + r_4 \\
 r_1 &= q_1p_4 + q_2p_3 - q_3p_2 + q_4p_1 \\
 r_2 &= -q_1p_3 + q_2p_4 + q_3p_1 + q_4p_2 \\
 r_3 &= q_1p_2 - q_2p_1 + q_3p_4 + q_4p_3 \\
 r_4 &= -q_1p_1 - q_2p_2 - q_3p_3 + q_4p_4 \\
 \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} &= \begin{bmatrix} p_4 & p_3 & -p_2 & p_1 \\ -p_3 & p_4 & p_1 & p_2 \\ p_2 & -p_1 & p_4 & p_3 \\ -p_1 & -p_2 & -p_3 & p_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}
 \end{aligned} \tag{A.8}$$

The matrix notation in Eq. (A.8) can be used to calculate the quaternion product.

### A.3 Quaternion Inverse

The inverse of a quaternion is equal to

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{|\mathbf{q}|^2} \tag{A.9}$$

If  $\mathbf{q}$  is a unit or normalized quaternion, the inverse in Eq. (A.9) becomes

$$\mathbf{q}^{-1} = \mathbf{q}^* \tag{A.10}$$

### A.4 Rotation of Vectors and Reference Frames

For any unit or normalized quaternion ( $|\mathbf{q}| = 1$ ) the quaternion is defined as

$$\mathbf{q} = \mathbf{q}_{1:3} + q_4 = \mathbf{u} \sin\left(\frac{\theta}{2}\right) + \cos\left(\frac{\theta}{2}\right) \tag{A.11}$$

where  $\mathbf{u}$  is a unit vector  $|\mathbf{u}| = 1 \Rightarrow u_1^2 + u_2^2 + u_3^2 = 1$ . For any unit quaternion  $\mathbf{q}$ , a rotation of any vector  $\mathbf{v}$  in  $\mathbb{R}^3$  can be calculated as

$$\mathbf{w} = \mathbf{q} \otimes \mathbf{v} \otimes \mathbf{q}^* \tag{A.12}$$

which is a rotation of  $\mathbf{v} = [\mathbf{v}^T \ 0]^T$  to  $\mathbf{w}$  through an angle of  $\theta$  about  $\mathbf{q}_{1:3}$  as the axis of rotation. A rotation of the reference frame relative to the vector  $\mathbf{v}$  through an angle of  $\theta$  about  $\mathbf{q}_{1:3}$  as the axis of rotation is

$$\mathbf{w} = \mathbf{q}^* \otimes \mathbf{v} \otimes \mathbf{q} \quad (\text{A.13})$$

## A.5 Quaternion Error

For control purposes the quaternion  $\mathbf{q}_{err}$  that rotates the body axes to the reference attitude can be calculated as [1]

$$\mathbf{q}_{err} = \mathbf{q}_{est}^{-1} \otimes \mathbf{q}_{ref} = \begin{bmatrix} q_{4,ref} & q_{3,ref} & -q_{2,ref} & q_{1,ref} \\ -q_{3,ref} & q_{4,ref} & q_{1,ref} & q_{2,ref} \\ q_{2,ref} & -q_{1,ref} & q_{4,ref} & q_{3,ref} \\ -q_{1,ref} & -q_{2,ref} & -q_{3,ref} & q_{4,ref} \end{bmatrix} \begin{bmatrix} -q_{1,est} \\ -q_{2,est} \\ -q_{3,est} \\ q_{4,est} \end{bmatrix} \quad (\text{A.14})$$

where  $\mathbf{q}_{est}$  is the estimated attitude quaternion and  $\mathbf{q}_{ref}$  is the attitude reference quaternion.  $\mathbf{q}_{err} = \mathbf{i}0 + \mathbf{j}0 + \mathbf{k}0 + 1$  means that  $\mathbf{q}_{est}$  and  $\mathbf{q}_{ref}$  are equal.

# Appendix B

## Inertia Calculation

This appendix presents the calculation of the inertia of AAUSAT3. Some assumptions are made in order to simplify calculations and the calculated inertia should hence be replaced by more accurate values, from e.g. the program AutoCAD Inventor, when the satellite design is completed. The Center of Mass (CoM) as well as the combined mass of AAUSAT3 are also calculated in this appendix.

### B.1 Inertia Calculation Procedure

To simplify the calculation of the inertia for AAUSAT3, all parts of the satellite is modeled as simple geometric objects, such as solid rectangular cuboid and solid cylinders, with a homogeneous mass distribution, i.e. all the parts have CoM in the Geometric center of Mass (GoM) . The inertia is hence calculated using the following procedure:

1. Calculate CoM for each part of the satellite.
2. Calculate inertia for each part of the satellite.
3. Calculate the combined CoM of the satellite.
4. Calculate the combined inertia of the satellite.

The inertia must be calculated for each principle axis of an object. For the solid rectangular cuboid, the inertia around one axis going through the CoM is [36]

$$I_{x,cu} = \frac{1}{12}m_{cu} (l_{y,cu}^2 + l_{z,cu}^2) \left[ kgm^2 \right] \quad (B.1)$$

$$I_{y,cu} = \frac{1}{12}m_{cu} (l_{x,cu}^2 + l_{z,cu}^2) \left[ kgm^2 \right] \quad (B.2)$$

$$I_{z,cu} = \frac{1}{12}m_{cu} (l_{x,cu}^2 + l_{y,cu}^2) \left[ kgm^2 \right] \quad (B.3)$$

where:

$I_{x,cu}$  is the inertia around the x-axis of the solid rectangular cuboid.

$m_{cu}$  is the mass of the solid rectangular cuboid.

$l_{x,cu}$  is the length of the solid rectangular cuboid along the x-axis.

The inertia of a cylinder is dependent of the axis under consideration. The inertia about the center axis, going through the CoM, is given by Eq. (B.4), while the other axes, also going through the CoM, are given by Eq. (B.5) [36].

$$I_{z,cy} = \frac{m_{cy}r_{cy}^2}{2} \quad [kgm^2] \quad (B.4)$$

$$I_{x,cy} = I_{y,cy} = \frac{1}{12}m_{cy}(3r_{cy}^2 + h_{cy}^2) \quad [kgm^2] \quad (B.5)$$

Where:

$I_{z,cy}$  is the inertia around the z-axis of the cylinder.

$m_{cy}$  is the mass of the cylinder.

$r_{cy}$  is the radius of the cylinder.

$h_{cy}$  is the height of the cylinder.

The combined CoM for the AAUSAT3 satellite is found as [36]

$$\mathbf{r}_{cms} = \frac{\sum_i m_{obj,i} \mathbf{r}_{cmo,i}}{\sum_i m_{obj,i}} \quad [m] \quad (B.6)$$

where:

$\mathbf{r}_{cms}$  is a vector to the combined CoM for the satellite.

$m_{obj,i}$  is the mass of the ith object under consideration.

$\mathbf{r}_{cmo,i}$  is a vector from the origin of a given reference frame to the CoM for the ith object.

By using the parallel-axis theorem, which states that the inertia of a rigid body about any axis parallel to the axis through the body's CoM is given by Eq. (B.7) [36], it is possible to calculate the combined inertia of the satellite.

$$\mathbf{I}_{sat} = \sum_i \mathbf{I}_{obj,i} + m_{obj,i} \mathbf{r}_{obj,i}^2 \quad [kgm^2] \quad (B.7)$$

Where:

$\mathbf{I}_{sat}$  is the combined inertia of the satellite.

$\mathbf{I}_{obj,i}$  is the inertia of the ith object.

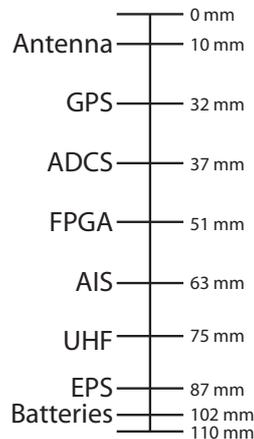
$m_{obj,i}$  is the mass of the ith object.

$\mathbf{r}_{obj,i}$  is distance for each axis through the satellite's CoM to each axis through the ith object's CoM.

## B.2 The Mechanical Structure of AAUSAT3

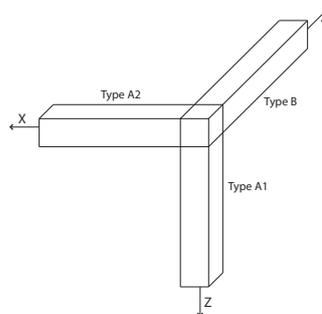
The following presents the mechanical structure of AAUSAT3. It is important to point out that the presented mechanical structure may not be completely in accordance with the actual design, however, it is a reasonable close approximation.

Both the Printed Circuit Boards (PCB), the frame, the antennas, the batteries and the side plates (including the magnetorquers for three of the side plates) are included in the calculation of the inertia. The PCBs are placed in the center of the satellite along the x- and y-axis of the SBRF, while the positions along the z-axis are presented in Figure B.1.



**Figure B.1:** Position of the antenna, the PCBs and the batteries in the stack relative to the z-axis of the SBRF.

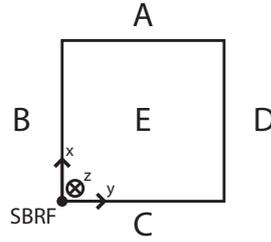
The frame of the satellite is divided into 12 pieces. The 12 pieces are furthermore divided into three types. The type of the frame part is defined by whether the longest side lies along the x-, y- or z-axis (see Figure B.2). The mass of each piece is modeled as 1/12 of the mass of the entire frame. It is also important to notice that the pieces overlaps each other in the corners of the satellite in order to simplify calculations.



**Figure B.2:** The three types of frame parts used.

Three of the side plates each include the mass of one magnetorquer. In order to distinguish between the side plates, a naming convention is introduced. Figure B.3 presents the naming convention used for the six side plates of AAUSAT3.

All parts used in the calculation of the inertia are summarized i Table B.1.



**Figure B.3:** Naming convention for the six sides of the AAUSAT3 satellite. The top is marked by E, while the bottom, marked by F, is concealed. The origin of the SBRF is marked by SBRF.

Component:	Length [mm]:	Width [mm]:	Height [mm]:	Mass [g]:	Comment:
Antenna	100	100	20	403.5	-
GPS	70	46	12	22.5	Estimated
ADCS	87	87	5	50	-
FPGA	87	87	14	30	-
AIS	87	87	12	59.8	-
UHF	87	87	12	37.2	-
EPS	87	87	12	43.2	-
Battery	65	15	15	41.6	Two pieces, cylinders.
Side plate A	100	0.5	110	13.7	-
Side plate B	0.5	100	110	31.4	With magnetorquer.
Side plate C	100	0.5	110	31.4	With magnetorquer.
Side plate D	0.5	100	110	13.7	-
Side plate E	100	100	0.5	13.7	-
Side plate F	100	100	0.5	31.4	With magnetorquer.
Type 1A frame	4.5	4.5	110	7.8	Four Pieces.
Type 2A frame	4.5	100	4.5	7.8	Four Pieces.
Type B frame	100	8.5	8.5	7.8	Four Pieces.

**Table B.1:** Mass and size of each component on AAUSAT3 included in the inertia calculation. Length, width and height are defined along the y-, x- and z-axis respectively. The estimated antenna mass is based on an aluminum prototype, with a large box, where the final version probably will be lighter.

### B.3 Mechanical Properties of AAUSAT3

The calculation of the CoM, the Inertia  ${}^c\mathbf{I}_{sat}$ , the mass and the quaternion that represents the rotation from the SBRF to the CRF is performed in the Matlab file `SatelliteInertia.m` found on the appended CD. The calculated mechanical properties are summarized in Table B.2.

Description	Value	Unit
CoM:	(49.07, 48.91, 42.98,]	[mm]
${}^c\mathbf{I}_{sat}$ :	(0.0017, 0.0022, 0.0022,)	[kgm <sup>2</sup> ]
${}^c\mathbf{q}$ :	(-0.0207, 0.7147, -0.0073, 0.6991)	[-]
Mass:	957.82	[g]

**Table B.2:** Mechanical properties of AAUSAT3.

# Appendix C

## Derivation of the Satellite Equations of Motion

This appendix presents the derivation of the satellite kinematic and dynamic equations of motion.

### C.1 Satellite Kinematic Equations

The kinematic equations of motion can be expressed as a set of first order differential equations specifying the time evolution of the attitude parameters. As stated in Section 2.3 the quaternion parameterization is chosen for the satellite kinematic analysis. This section is based on the deviation made in [1].

Let the quaternion  $\mathbf{q}(t)$  represent the orientation of one reference frame with respect to another reference frame at time  $t$ . In this particular case it is the orientation of the CRF relative to the ECI. For convenience this information has been omitted in the notation. Let another quaternion  $\mathbf{q}(t + \Delta t)$  represent the orientation at time  $t + \Delta t$ . Then

$$\mathbf{q}(t + \Delta t) = \mathbf{q}(t) \otimes \mathbf{q}'(\Delta t) \tag{C.1}$$

where the quaternion  $\mathbf{q}'(\Delta t)$  represents the orientation of  $\hat{c}_1$ ,  $\hat{c}_2$  and  $\hat{c}_3$  at time  $t + \Delta t$  relative to time  $t$ . The unit vectors  $\hat{c}_1$ ,  $\hat{c}_2$  and  $\hat{c}_3$  form the CRF. As defined in Appendix A, the quaternion  $\mathbf{q}'(\Delta t)$  can be written as

$$q'_1(\Delta t) = u_{c_1} \sin\left(\frac{\Delta\Phi}{2}\right) \quad (\text{C.2})$$

$$q'_2(\Delta t) = u_{c_2} \sin\left(\frac{\Delta\Phi}{2}\right) \quad (\text{C.3})$$

$$q'_3(\Delta t) = u_{c_3} \sin\left(\frac{\Delta\Phi}{2}\right) \quad (\text{C.4})$$

$$q'_4(\Delta t) = \cos\left(\frac{\Delta\Phi}{2}\right) \quad (\text{C.5})$$

where  $\mathbf{u}$  is the rotation axis unit vector at time  $t$  and  $\Delta\Phi$  is the rotation in time  $\Delta t$ . Using the quaternion product rule derived in Appendix A in Eq. (C.1) together with the definitions in Eq. (C.2) to (C.5) gives

$$\mathbf{q}(t + \Delta t) = \left( \cos\left(\frac{\Delta\Phi}{2}\right) \mathbf{1} + \sin\left(\frac{\Delta\Phi}{2}\right) \begin{bmatrix} 0 & u_{c_3} & -u_{c_2} & u_{c_1} \\ -u_{c_3} & 0 & u_{c_1} & u_{c_2} \\ u_{c_2} & -u_{c_1} & 0 & u_{c_3} \\ -u_{c_1} & -u_{c_2} & -u_{c_3} & 0 \end{bmatrix} \right) \mathbf{q}(t) \quad (\text{C.6})$$

By using the following small angle theorems [1]

$$\cos\left(\frac{\Delta\Phi}{2}\right) \approx 1 \quad (\text{C.7})$$

$$\sin\left(\frac{\Delta\Phi}{2}\right) \approx \frac{1}{2} \omega \Delta t \quad (\text{C.8})$$

Eq. (C.6) can be rewritten to

$$\begin{aligned} \mathbf{q}(t + \Delta t) &= \left( \mathbf{1} + \frac{1}{2} \underline{\underline{\Omega}}(\omega) \Delta t \right) \mathbf{q}(t) \\ \frac{\mathbf{q}(t + \Delta t) - \mathbf{q}(t)}{\Delta t} &= \frac{1}{2} \underline{\underline{\Omega}}(\omega) \mathbf{q}(t) \end{aligned} \quad (\text{C.9})$$

where  $\underline{\underline{\Omega}}(\omega)$  is the 4x4 skew symmetric matrix ( $t$  has been omitted from the notation but  $\omega$  is a function of time)

$$\underline{\underline{\Omega}}(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix}, \text{ for } \omega \in \mathbb{R}^3 \quad (\text{C.10})$$

Letting  $\Delta t \rightarrow 0$  gives

$$\dot{\mathbf{q}}(t) \equiv \lim_{\Delta t \rightarrow 0} \frac{\mathbf{q}(t + \Delta t) - \mathbf{q}(t)}{\Delta t} = \frac{1}{2} \underline{\underline{\boldsymbol{\Omega}}}(\boldsymbol{\omega}) \mathbf{q}(t) \quad (\text{C.11})$$

The time dependent relationship between two reference frames is described by Eq. (C.11), which is the kinematic differential equations of the satellite. This is the same result as in [1, 3].

## C.2 Satellite Dynamic Equations

The satellite is assumed to be a rigid body without moving parts. A rigid body in space has six degrees of freedom [3]. Three of them are rotational motion and the other three are translational motion defined by the orbit and is not a part of this section.

The dynamic differential equations of motion of the satellite relates the time derivative of the angular momentum and the torques applied to the satellite. In an inertial reference frame Euler's second law states that [3]

$$\dot{\mathbf{L}}(t) = {}^i \mathbf{N}_{ext}(t) \quad (\text{C.12})$$

where  $\dot{\mathbf{L}}$  is the time derivative of the angular momentum and  ${}^i \mathbf{N}_{ext}$  is the external torques. However, it is desirable to express this in the CRF, so it becomes the angular momentum about the center of mass of the satellite. By defining a rotation matrix  ${}^c \underline{\underline{\mathbf{A}}}$  the angular momentum in the CRF is

$${}^c \mathbf{L}(t) = {}^c \underline{\underline{\mathbf{A}}}(t) \dot{\mathbf{L}}(t) \quad (\text{C.13})$$

Using the product rule for differentiation gives

$${}^c \dot{\mathbf{L}}(t) = {}^c \dot{\underline{\underline{\mathbf{A}}}}(t) \dot{\mathbf{L}}(t) + {}^c \underline{\underline{\mathbf{A}}}(t) \ddot{\mathbf{L}}(t) \quad (\text{C.14})$$

According to [4]  ${}^c \dot{\underline{\underline{\mathbf{A}}}}(t) = -{}^c \boldsymbol{\omega}(t) \times {}^c \underline{\underline{\mathbf{A}}}(t)$  and substituting this into Eq. (C.14) gives

$${}^c \dot{\mathbf{L}}(t) = -{}^c \boldsymbol{\omega}(t) \times {}^c \underline{\underline{\mathbf{A}}}(t) \dot{\mathbf{L}}(t) + {}^c \underline{\underline{\mathbf{A}}}(t) \ddot{\mathbf{L}}(t) \quad (\text{C.15})$$

which, by using Eq. (C.12) and (C.13), can be rewritten to

$${}^c\mathbf{L}(t) = -{}^c\boldsymbol{\omega}(t) \times {}^c\mathbf{L}(t) + {}^c\mathbf{N}_{ext}(t) \quad (\text{C.16})$$

Angular momentum of a rigid body is also [3]

$${}^c\mathbf{L}(t) = \underline{\mathbf{I}}_{sat} {}^c\boldsymbol{\omega}(t) \quad (\text{C.17})$$

where  $\underline{\mathbf{I}}_{sat}$  is the diagonal constant inertia matrix about the CRF (the principal axes). Substituting this into Eq. (C.16) gives

$$\begin{aligned} \underline{\mathbf{I}}_{sat} {}^c\dot{\boldsymbol{\omega}}(t) &= -{}^c\boldsymbol{\omega}(t) \times (\underline{\mathbf{I}}_{sat} {}^c\boldsymbol{\omega}(t)) + {}^c\mathbf{N}_{ext}(t) \\ {}^c\dot{\boldsymbol{\omega}}(t) &= \underline{\mathbf{I}}_{sat}^{-1} [-{}^c\boldsymbol{\omega}(t) \times (\underline{\mathbf{I}}_{sat} {}^c\boldsymbol{\omega}(t)) + {}^c\mathbf{N}_{ext}(t)] \end{aligned} \quad (\text{C.18})$$

The torques acting on the satellite are the disturbance torques  ${}^c\mathbf{N}_{dist}(t)$  treated in Section 2.5 and the control torques applied by actuators  ${}^c\mathbf{N}_{ctrl}(t)$ . The angular velocity of the CRF  ${}^c\boldsymbol{\omega}(t)$  is the same  $\boldsymbol{\omega}(t)$  as in the kinematic equations of motion. Removing the reference frame attachment of the variables in the notation for simplicity gives

$$\dot{\boldsymbol{\omega}}(t) = \underline{\mathbf{I}}_{sat}^{-1} [-\underline{\mathbf{S}}(\boldsymbol{\omega})(\underline{\mathbf{I}}_{sat} \boldsymbol{\omega}(t)) + \mathbf{N}_{dist}(t) + \mathbf{N}_{ctrl}(t)] \quad (\text{C.19})$$

where  $\underline{\mathbf{S}}(\boldsymbol{\omega})$  is a skew symmetric matrix defined as<sup>1</sup>

$$\underline{\mathbf{S}}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \text{ for } \boldsymbol{\omega} \in \mathbb{R}^3 \quad (\text{C.20})$$

Eq. (C.19) is the dynamic differential equations of motion for the satellite.

---

<sup>1</sup> $t$  has been omitted from the notation, but  $\boldsymbol{\omega}$  is a function of time

## Worst Case Disturbance

This appendix contains the worst case calculations for the gravitational, aerodynamic, radiation and magnetic residual disturbances. The calculations are based on equations derived in Section 2.5 and the vectors are defined in the SBRF, except in the calculation of the gravity gradient torque, where the CRF is used. However, this has been omitted from the notation.

The description of each disturbance contains the used values for the different constants and variables, the obtained result and a reference to a Matlab file, in which the result has been obtained. All results are summarized last in this appendix.

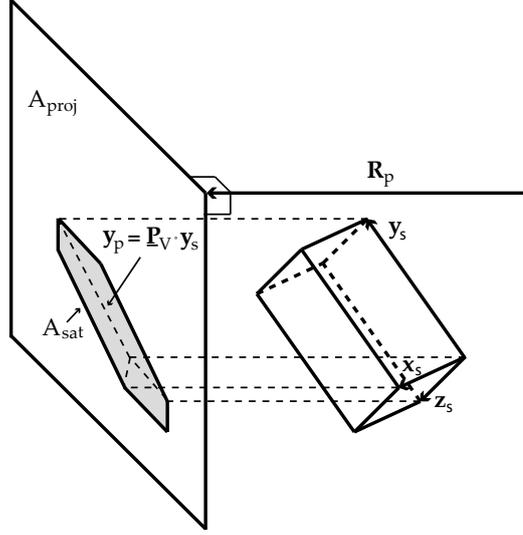
Furthermore, a calculation of the maximum exposed area of the satellite, as seen from a given vector pointing towards the satellite, is included in this appendix.

### D.1 Maximal Exposed Area of AAUSAT3

The radiation and the aerodynamic disturbance torque depends on the exposed area of the satellite. By assuming that the exposed area of the satellite can be seen as a plane perpendicular to a vector  $\mathbf{R}_p$ , which is parallel to the translational velocity vector of the satellite  $\mathbf{V}_{sat}$  (see Subsection 2.5.2) or parallel to a vector from the satellite to the Sun  $\mathbf{R}_{sS}$  (see Subsection 2.5.3), it is possible to calculate the disturbance torques originating from these.

The above mentioned task is illustrated in Figure D.1, where  $\underline{\mathbf{P}}_V$  is the projection matrix used to project the x-, y- and z-axis of the SBRF onto the projection plane  $A_{proj}$ . The length of the x-, y- and z-axis vectors are equal to the dimensions of the satellite (for AAUSAT3 they are 0.1, 0.1 and 0.11 [m] respectively). To find the projection matrix  $\underline{\mathbf{P}}_V$ , it is first necessary to find the projection matrix  $\underline{\mathbf{P}}_R$  that projects the x-, y- and z-axis vectors onto the  $\mathbf{R}_p$  vector.

The projection matrix  $\underline{\mathbf{P}}_R$  is given as [92]



**Figure D.1:** Projection of a satellite onto a plane  $A_{proj}$ , which is perpendicular to an incident vector  $\mathbf{R}_p$ . The area  $A_{sat}$  is defined by the the projected axes of the SBRF, where the projection is performed with the matrix  $\underline{\mathbf{P}}_V$ .

$$\begin{aligned} \underline{\mathbf{P}}_R &= \underline{\mathbf{A}}_p \left( \underline{\mathbf{A}}_p^T \underline{\mathbf{A}}_p \right)^{-1} \underline{\mathbf{A}}_p^T \\ &= \frac{1}{\|\mathbf{R}_p\|^2} \begin{bmatrix} R_{p1}^2 & R_{p1}R_{p2} & R_{p1}R_{p3} \\ R_{p1}R_{p2} & R_{p2}^2 & R_{p2}R_{p3} \\ R_{p1}R_{p3} & R_{p2}R_{p3} & R_{p3}^2 \end{bmatrix} \end{aligned} \quad (\text{D.1})$$

where:

$\underline{\mathbf{A}}_p$  is a  $3 \times 1$  matrix with  $\mathbf{R}_p$  as column vector.

$R_{p1}$  is the first element of the vector.

The y-axis vector  $\mathbf{y}_s$  is projected as an example in Figure D.1 and the projecting of this vector is defined as

$$\begin{aligned} \underline{\mathbf{P}}_V \mathbf{y}_s &= \mathbf{y}_s - \underline{\mathbf{P}}_R \mathbf{y}_s \\ &= \left( \mathbf{1}_{3 \times 3} - \underline{\mathbf{P}}_R \right) \mathbf{y}_s \end{aligned} \quad (\text{D.2})$$

where:

$\mathbf{1}_{3 \times 3}$  is the identity matrix.

Hence the projection matrix  $\underline{\mathbf{P}}_V$  is

$$\begin{aligned}
 \underline{\mathbf{P}}_V &= \underline{\mathbf{1}}_{3 \times 3} - \underline{\mathbf{P}}_R \\
 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{\|\mathbf{R}_p\|^2} \begin{bmatrix} R_{p_1}^2 & R_{p_1}R_{p_2} & R_{p_1}R_{p_3} \\ R_{p_1}R_{p_2} & R_{p_2}^2 & R_{p_2}R_{p_3} \\ R_{p_1}R_{p_3} & R_{p_2}R_{p_3} & R_{p_3}^2 \end{bmatrix} \\
 \underline{\mathbf{P}}_V \|\mathbf{R}_p\|^2 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \|\mathbf{R}_p\|^2 - \begin{bmatrix} R_{p_1}^2 & R_{p_1}R_{p_2} & R_{p_1}R_{p_3} \\ R_{p_1}R_{p_2} & R_{p_2}^2 & R_{p_2}R_{p_3} \\ R_{p_1}R_{p_3} & R_{p_2}R_{p_3} & R_{p_3}^2 \end{bmatrix} \\
 \underline{\mathbf{P}}_V &= \frac{1}{\|\mathbf{R}_p\|^2} \begin{bmatrix} R_{p_2}^2 + R_{p_3}^2 & -R_{p_1}R_{p_2} & -R_{p_1}R_{p_3} \\ -R_{p_1}R_{p_2} & R_{p_1}^2 + R_{p_3}^2 & -R_{p_2}R_{p_3} \\ -R_{p_1}R_{p_3} & -R_{p_2}R_{p_3} & R_{p_1}^2 + R_{p_2}^2 \end{bmatrix} \quad (D.3)
 \end{aligned}$$

The area of the satellite can then be calculated as the norm of the cross product's between the projected SBRF axis vector's  $(\mathbf{x}_p, \mathbf{y}_p, \mathbf{z}_p)$  [92].

$$A_{sat} = \|\mathbf{x}_p \times \mathbf{y}_p\| + \|\mathbf{x}_p \times \mathbf{z}_p\| + \|\mathbf{y}_p \times \mathbf{z}_p\| \quad (D.4)$$

The maximal exposed area of AAUSAT3 is calculated to  $0.0185 \text{ [m}^2\text{]}$ . The calculation of the maximal exposed area is located in both the `AeroDragMax.m` and the `RadiationTorqueMax.m` Matlab file on the appended CD.

It is important to notice that even when the maximal area is given, the worst case calculation may be less than the actual worst case torque due to the inertia, which may contribute to a larger torque at another attitude. The worst case calculations are therefore based on iterations.

## D.2 Worst Case Gravitational Torque

As described in Subsection 2.5.1 the gravity gradient torque  $\mathbf{N}_{gg}$  exerted on the satellite is given as

$$\mathbf{N}_{gg} = \frac{3\mu_E}{\|\mathbf{R}_{Ese}\|^3} (\hat{\mathbf{R}}_{Ese} \times (\underline{\mathbf{I}}_{sat} \hat{\mathbf{R}}_{Ese})) \text{ [Nm]} \quad (D.5)$$

where:

$\mathbf{R}_{Ese}$  is a vector pointing from the Earth CoM to the satellite's CoM.

$\mu_E$  is the geocentric gravitational constant.

$\underline{\mathbf{I}}_{sat}$  is the moment of inertia tensor for the satellite.

The diagonal terms of the principal moment of inertia matrix are calculated in Appendix B. By inserting this and a geocentric gravitational constant with a value of  $3.9860 \cdot 10^{14} \text{ [m}^3/\text{s}^2\text{]}$  into Equation D.5, it is possible to calculate the maximum gravitational torque.

The worst case gravity gradient torque depends on the  $\mathbf{R}_{Ese}$  vector. This vector is varied in order to obtain the worst case disturbance torque, which is done in the Matlab file `GravityTorqueMax.m`, included on the appended CD. Using the previous mentioned variables and constants, a torque of  $0.87 \text{ [nNm]}$  is found.

### D.3 Worst Case Aerodynamic Torque

As described in Subsection 2.5.2 the aerodynamic torque  $\mathbf{N}_{as}$  exerted on a satellite is given as

$$\mathbf{N}_{as} = \mathbf{r}_{scp} \times \mathbf{F}_{as} \text{ [Nm]} \quad (\text{D.6})$$

where:

$\mathbf{r}_{scp}$  is a vector from the CoM to the Center of Pressure (CoP) of the satellite.

$\mathbf{F}_{as}$  is the aerodynamic force exerted on the satellite.

The aerodynamic force exerted on the satellite is given as

$$\mathbf{F}_{as} = -\frac{1}{2} C_a \rho_a \|\mathbf{V}_{sat}\|^2 \hat{\mathbf{V}}_{sat} A_{sat} \text{ [N]} \quad (\text{D.7})$$

where:

$C_a$  is the aerodynamic drag coefficient.

$\rho_a$  is the atmospheric density.

$\mathbf{V}_{sat}$  is the translational velocity vector of the satellite.

$A_{sat}$  is the exposed area of the satellite.

To calculate the maximum torque, an aerodynamic drag coefficient of  $C_a = 2$ , an atmospheric density of  $1.454 \cdot 10^{-13} \text{ [kg/m}^3\text{]}$  (for an altitude of 600 [km]), a satellite velocity of 7.56 [km/s] and a  $\mathbf{r}_{scp} = [0.0009 \ 0.0011 \ 0.0120]^T \text{ [m]}$  is used for the satellite. The translational velocity vector  $\mathbf{V}_{sat}$  is parallel to the outward normal vector  $\hat{\mathbf{n}}_{on}$  for the maximum exposed area of the satellite (see Appendix D.1).

By inserting the above stated values into Eq. (D.7) and afterward solving Eq. (D.6), while iterating through values of  $\mathbf{V}_{sat}$  and  $A_{sat}$ , a worst case torque of 1.65 [nNm] is found.

The calculation of the worst case aerodynamic disturbance is calculated in the Matlab file `AeroDragMax.m` on the appended CD.

### D.4 Worst Case Radiation Torque

As described in Subsection 2.5.3, the radiation torque exerted on the satellite  $\mathbf{N}_{rs}$  is given as

$$\mathbf{N}_{rs} = \mathbf{r}_{scp} \times \mathbf{F}_{rs} \text{ [Nm]} \quad (\text{D.8})$$

where:

$\mathbf{F}_{rs}$  is the solar radiation force exerted on the satellite.

The solar radiation force exerted on the satellite is given as

$$\mathbf{F}_{rs} = C_{rk} A_{sat} P_{rmf} \hat{\mathbf{R}}_{sS} [N] \quad (D.9)$$

where:

$C_{rk}$  is a dimensionless constant in the range  $0 \leq C_{rk} \leq 2$ .

$P_{rmf}$  is the mean momentum flux acting on the normal to the Sun's radiation.

$\mathbf{R}_{sS}$  is the vector from the satellite's CoM to the Sun's CoM.

A mean momentum flux of  $4.5565 \cdot 10^{-6} [kg/ms^2]$  is calculated using Eq. (2.16) in Subsection 2.5.3. By inserting  $C_{rk} = 1.5$  and the before mentioned value of the mean momentum flux into Eq. (D.9), while iterating through values of  $\hat{\mathbf{R}}_{sS}$  and  $A_{sat}$  and afterward solving Eq. (D.8), a maximum radiation torque of  $1.36 [nNm]$  is found.

The procedure for calculation of the worst case radiation torque is the same as for the aerodynamic torque, where the vector  $\hat{\mathbf{R}}_{sS}$  is parallel to the translational velocity vector  $\hat{\mathbf{n}}_{on}$ . The calculation is located in the Matlab file `RadiationTorqueMax.m` on the appended CD.

## D.5 Worst Case Magnetic Residual Torque

As described in Subsection 2.5.4 the magnetic residual torque  $\mathbf{N}_{mrs}$  is given as

$$\mathbf{N}_{mrs} = \mathbf{m}_{mms} \times \mathbf{B}_E [Nm] \quad (D.10)$$

where:

$\mathbf{m}_{mms}$  is the effective magnetic dipole moment of the satellite.

$\mathbf{B}_E$  is the geomagnetic field vector.

The magnitude of the magnetic residual torque  $\mathbf{N}_{mrs}$  is

$$\|\mathbf{N}_{mrs}\| = \|\mathbf{m}_{mms}\| \|\mathbf{B}_E\| \sin(\theta_{mrB}) [Nm] \quad (D.11)$$

where:

$\theta_{mrB}$  is the angle between the magnetic dipole moment and the geomagnetic field vector.

It follows from Eq. (D.11), that the maximal magnetic residual torque is exerted on AAUSAT3, when the magnetic residual is perpendicular to the magnetic field.

A geocentric magnetic flux density of  $48\,000 [nT]$  is used to calculate the worst case magnetic residual torque. This value is the expected maximal magnetic field in a altitude of  $600 [km]$  [60].

By inserting the magnetic flux density of  $48\,000 [nT]$  and an effective dipole moment of  $0.001 [Am^2]$  (see Subsection 2.5.4) into Eq. (D.11), a maximal torque of  $48 [nNm]$  is found.

The calculation of the worst case magnetic residual torque is located in the Matlab file `MagneticResidualTorque.m` on the appended CD.

## D.6 Total Worst Case Disturbance Torque

All the disturbance torques are summarized in Table D.1.

<b>Disturbance:</b>	<b>Torque [nNm]:</b>
Gravitational:	0.87
Aerodynamic:	1.65
Radiation:	1.36
Magnetic Residual:	48.00
Total:	51.88

*Table D.1: Length of worst case disturbance torque for each environmental disturbance and the total worst case disturbance torque that will affect AAUSAT3.*

The calculation of the total worst case disturbance torque is located in the Matlab file `WCDTorque.m` on the appended CD.

# Appendix **E**

## Verification of Simulation Environment

This appendix presents the results of tests performed on the simulation environment for AAUSAT3. The main purposes of these tests are to verify that no errors have been introduced, under the porting of the original simulation environment made for the AAUSAT-II and NSO satellites. More comprehensive tests could have been performed, but only small changes have been introduced and the original simulation environment has already been tested thoroughly by its previous developers [2, 4, 37, 45, 51, 52, 53].

The orbit for AAUSAT3 is expected to be similar to the one for AAUSAT-II and all tests in this appendix are therefore based on an AAUSAT-II TLE from the 17. of November 2009. All simulation files are on the appended CD.

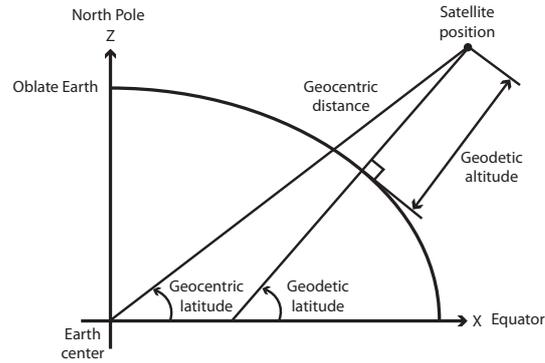
### **E.1 Orbit Propagator and Ephemeris Model Test**

The SGP4 orbit propagator giving the satellite position vector (ECI) and the ephemeris models that gives position vectors for the Sun and the Moon (ECI) have been tested by comparing their output with a simulation in a satellite tracking system called Orbitron<sup>1</sup>.

Before the output from the models can be visually compared with Orbitron, it has to be converted to geodetic latitude and longitude. This is done by first converting the Cartesian position vector into spherical coordinates giving what is equivalent to geocentric longitude, colatitude and distance.  $90 [deg]$  minus the colatitude gives the geocentric latitude. Geocentric latitude and distance can then be used to calculate the geodetic latitude. Figure E.1 illustrates the difference between geocentric and geodetic latitude, which is caused by the oblateness of the Earth. Geocentric longitude is the same as geodetic longitude.

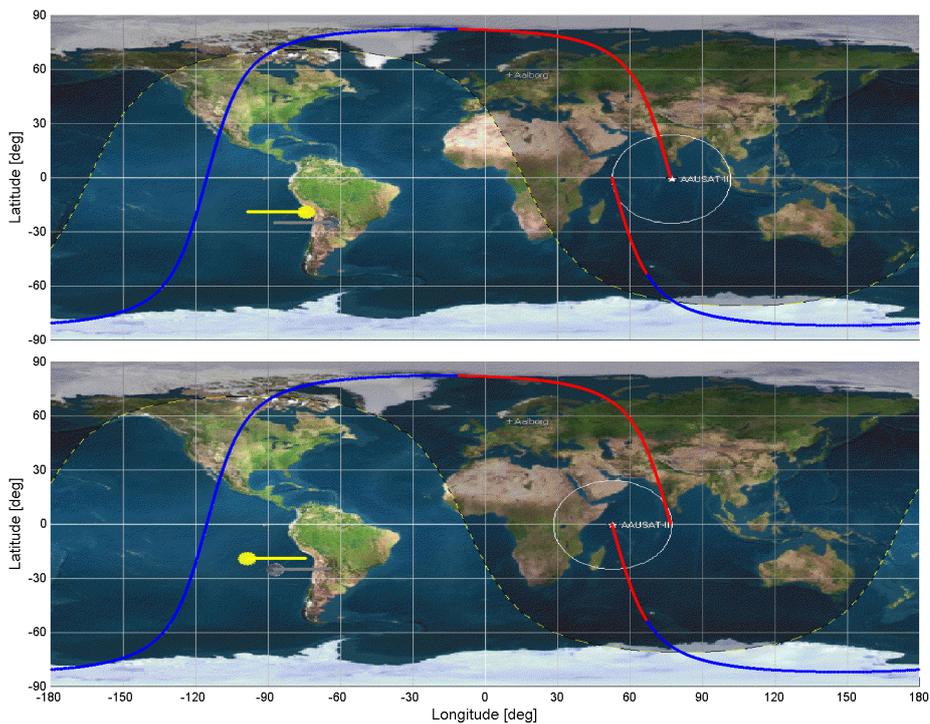
Figure E.2 shows the SGP4 and ephemeris output from the simulation in Matlab on two maps of the World taken from Orbitron. The top image shows the satellite in its starting position (start of the TLE) and the bottom image shows the satellite position after one orbit. The red and blue lines are the propagated trajectory given by SGP4. Orbitron also propa-

<sup>1</sup>Can be downloaded at: <http://www.softpedia.com/get/Others/Miscellaneous/Orbitron.shtml>



**Figure E.1:** Oblate Earth showing the difference between geocentric and geodetic latitude [3].

gates the satellite trajectory and draws this data as a line, but this line is hidden under the simulated SGP4 data, indicating a match between the orbit propagators. The white ring illustrates the FoV of the satellite and is drawn by Orbitron.



**Figure E.2:** Orbitron screenshots overlaid with SGP4 and ephemeris model output. Top image shows the satellite, the Sun and the Moon starting position and the bottom image shows the position after the satellite has completed one orbit.

The maps are overlaid with a shadow, showing which part of the Earth, that is illuminated by the Sun. Orbitron also shows the position of the Sun and the Moon, and the starting

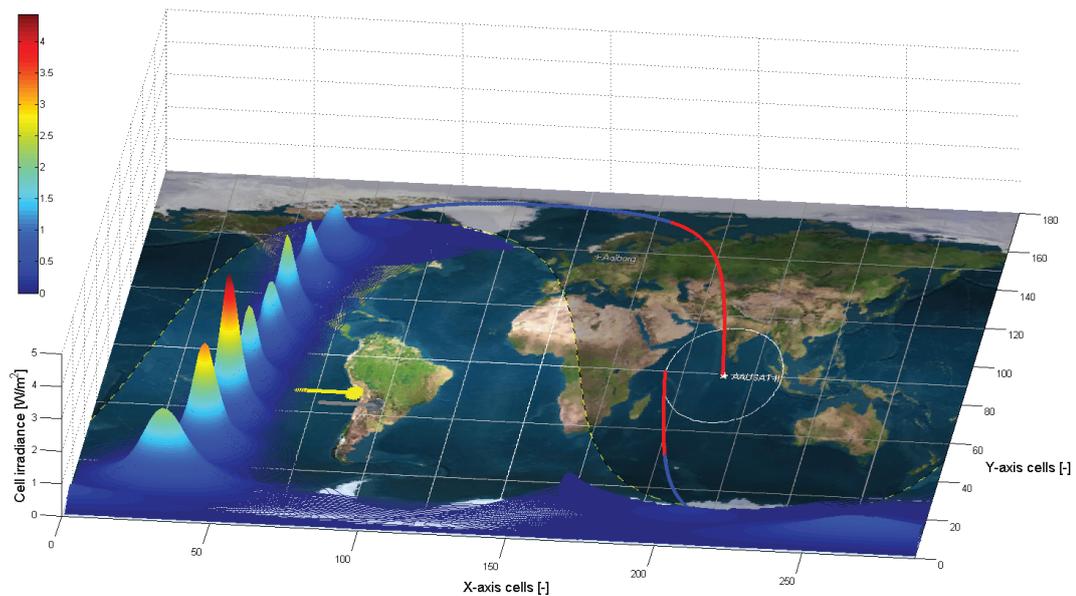
and ending positions coincide with the propagated trajectory calculated by the ephemeris models (yellow and gray line).

The red part of the satellite trajectory indicates that the satellite was in eclipse during this period. This information is given by the eclipse model and it also coincides with the prediction from Orbitron.

It is important to notice, that no guarantee have been made to whether the output from Orbitron is correct or not. For more rigorous tests with e.g. comparison of SGP4 and GPS data from the Ørsted satellite see [2].

## E.2 Earth Albedo Model Test

The output from the Earth albedo model is a  $180 \times 288$  reflectivity matrix containing reflected irradiance hitting the satellite. One cell corresponds to  $1 [deg]$  latitude and  $1.25 [deg]$  longitude. Only the common quantity of cells within the satellites FoV and the Sun's FoV has values different from zero. Figure E.3 shows a 3D plot of the reflectivity matrix for 20 sample points evenly distributed along the orbit also shown in Figure E.2.



**Figure E.3:** 3D plot of the reflectivity matrix sampled at 20 points along one orbit.

As Figure E.3 indicates, there is no reflected irradiance within the part of the Earth that is not illuminated by the Sun, illustrated with the shadow. However, the shadow in the figure is only correctly positioned at the start of the orbit, because the Sun moves along the yellow trajectory.

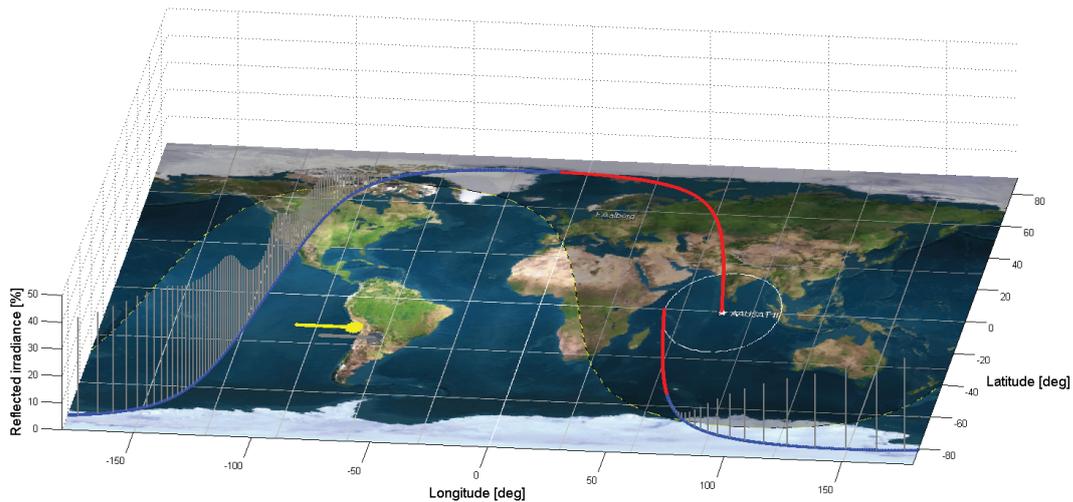
The highest amount of irradiance is obtained when the satellite and the Sun are closest to each other near the equator. The total amount of irradiance in percentage in the 20 sample points are calculated by summing all the cells and taking the percentage relative to the

average solar irradiance of  $1366 \text{ [W/m}^2\text{]}$  (see Subsection 2.5.3). The result is presented in Table E.1.

Sample nr.	Total irradiance [%]	Sample nr.	Total irradiance [%]
1 (start of orbit)	0	11	17.52
2	0	12	18.87
3	0	13	34.39
4	0	14	36.76
5	0	15	42.43
6	0.03	16	29.99
7	2.93	17	4.64
8	14.55	18	0.05
9	14.79	19	0
10	19.02	20 (end of orbit)	0

*Table E.1: Total irradiance in 20 samples points along an orbit.*

During the orbit up to 42.4% of the irradiance hitting the surface of the Earth is reflected up to the satellite. Figure E.4 shows the reflected irradiance in percentage for 200 sample points along the orbit.

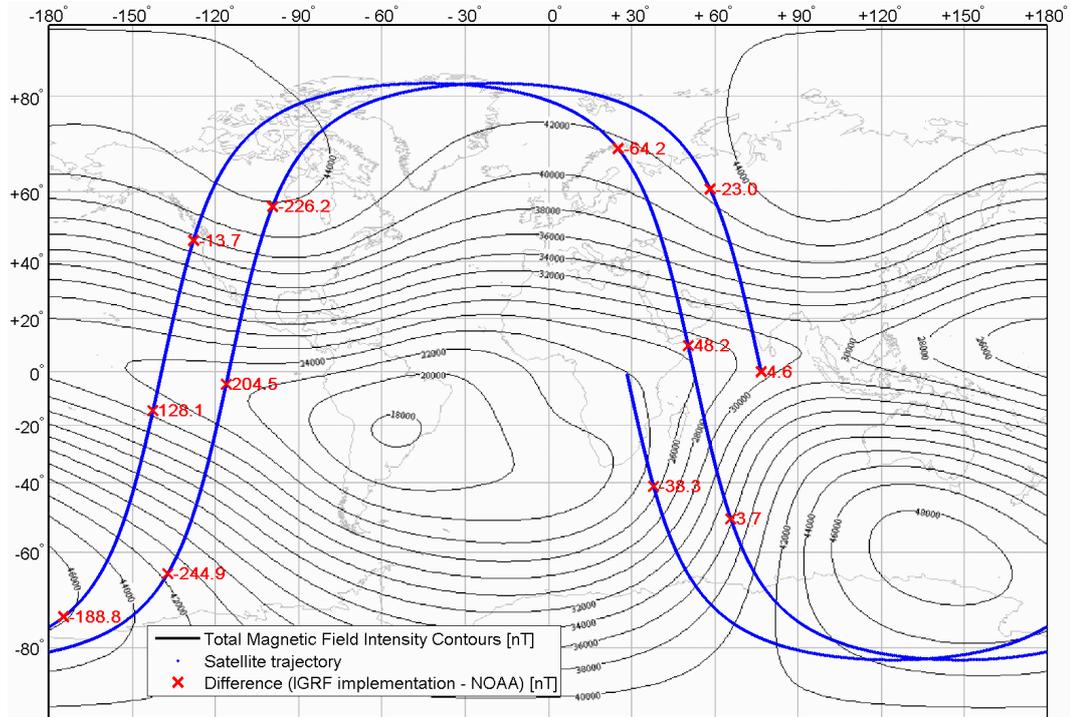


*Figure E.4: Reflected irradiance (gray lines) in percentage along an orbit.*

### E.3 Magnetic Field Model Test

The original IGRF model implementation had a maximum order of 10, but the 10th generation IGRF, for 2005-2010, has a maximum order of 13. The model implementation was therefore updated with more coefficients. In order to test the the implementation, 12 samples of total magnetic field was taken along two orbits and compared with an online implementation of the 10th generation IGRF [58]. The difference in total magnetic field (geometrically the length of the magnetic field vector) is plotted with red numbers in Figure

E.5. The contours representing total magnetic field strength has been generated with a third program [60].



**Figure E.5:** Satellite trajectory (blue) and difference in total magnetic field strength at 12 sample points (red). The contours represent total magnetic field strength in a height of 630 [km] and varies between 18 000 to 48 000 [nT].

The difference in total magnetic field strength is summarized in Table E.2.

nr.	1	2	3	4	5	6	7	8	9	10	11	12
[nT]	4.6	-23.0	-226.2	204.5	-244.9	3.6	48.2	-64.2	-13.7	128.1	188.8	-38.3

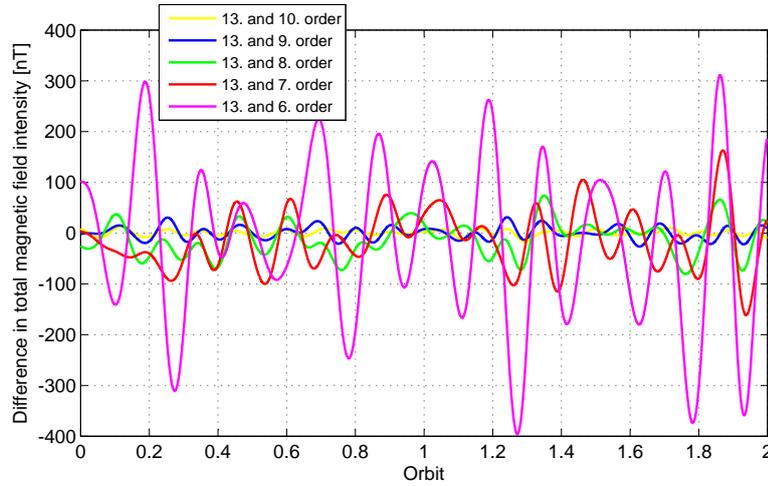
**Table E.2:** Difference in total magnetic field strength in 12 samples.

There does not seem to be a pattern in the deviation and it is assumed that the difference is caused by a difference in the implementation. However, the deviation is relatively small compared to the magnetic field varying between 18 000 to 48 000 [nT] and it will not be addressed further in this thesis.

The difference in total magnetic field intensity between different orders of IGRF has been plotted in Figure E.6, in order to be able to choose an appropriate IGRF model order for the software on board the satellite.

The standard deviation between different model orders is presented in Table E.3.

The deviation is increasing rapidly when going below 8th order, which is also the chosen order in [4]. The difference between a 13th and an 8th order implementation has been plotted for 8 orbits in Figure E.7.

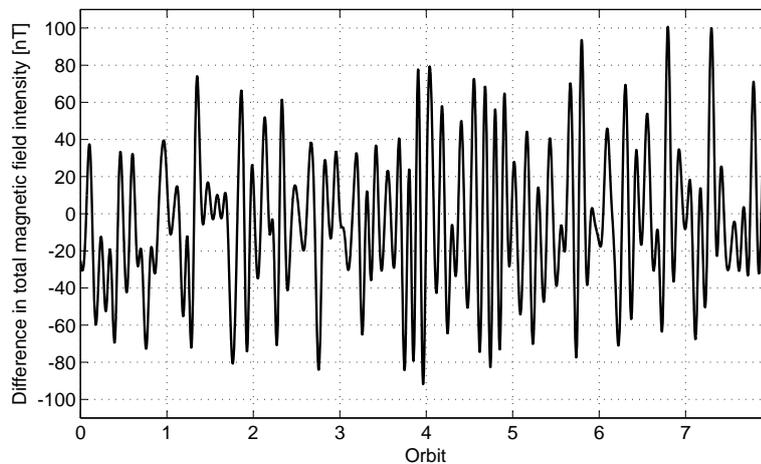


**Figure E.6:** Difference in total magnetic field intensity between different orders of IGRF.

Order comparison	13. - 10.	13. - 9.	13. - 8.	13. - 7.	13. - 6.
Standard deviation [nT]	5.34	12.36	33.36	56.16	156.31

**Table E.3:** Standard deviation between the output of different model orders.

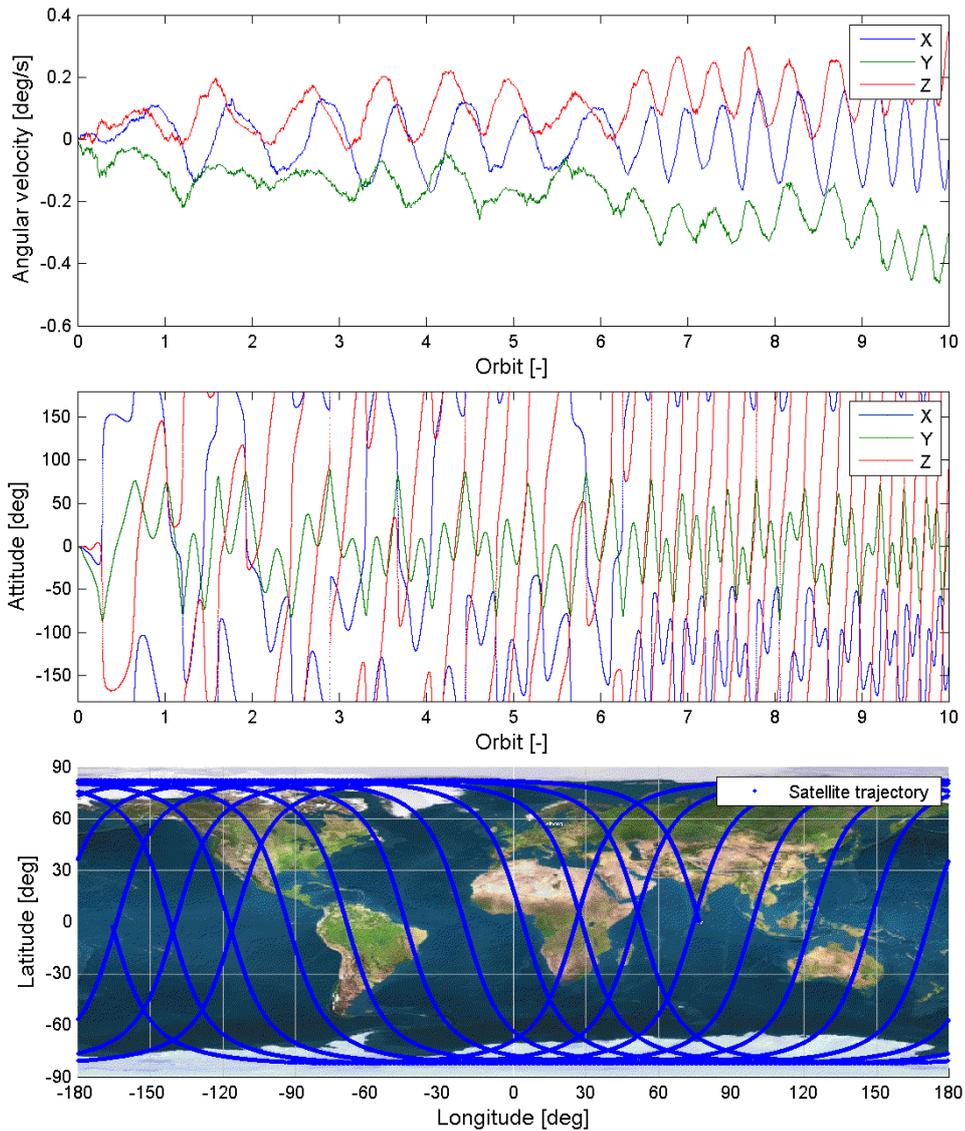
This gives a standard deviation of 36.14 [nT] with a mean of -5.07 [nT]. The minimum and maximum difference in Figure E.7 is -91.8 [nT] and 100.65 [nT] respectively, which is considered acceptable. It is important to remember that the 13. order implementation does not necessarily give the correct magnetic field, due to variations that are not modeled. This could e.g. be magnetic storms giving contributions of perhaps 1000 [nT] and more [58].



**Figure E.7:** Difference in total magnetic field intensity between 13. and 8. order IGRF.

## E.4 Equations of Motion Output and Disturbance Models Test

Simulation of 10 orbits with disturbances, an initial angular velocity of  $(0,0,0)$  [rad/s] and no control input, gives the output shown in Figure E.8.

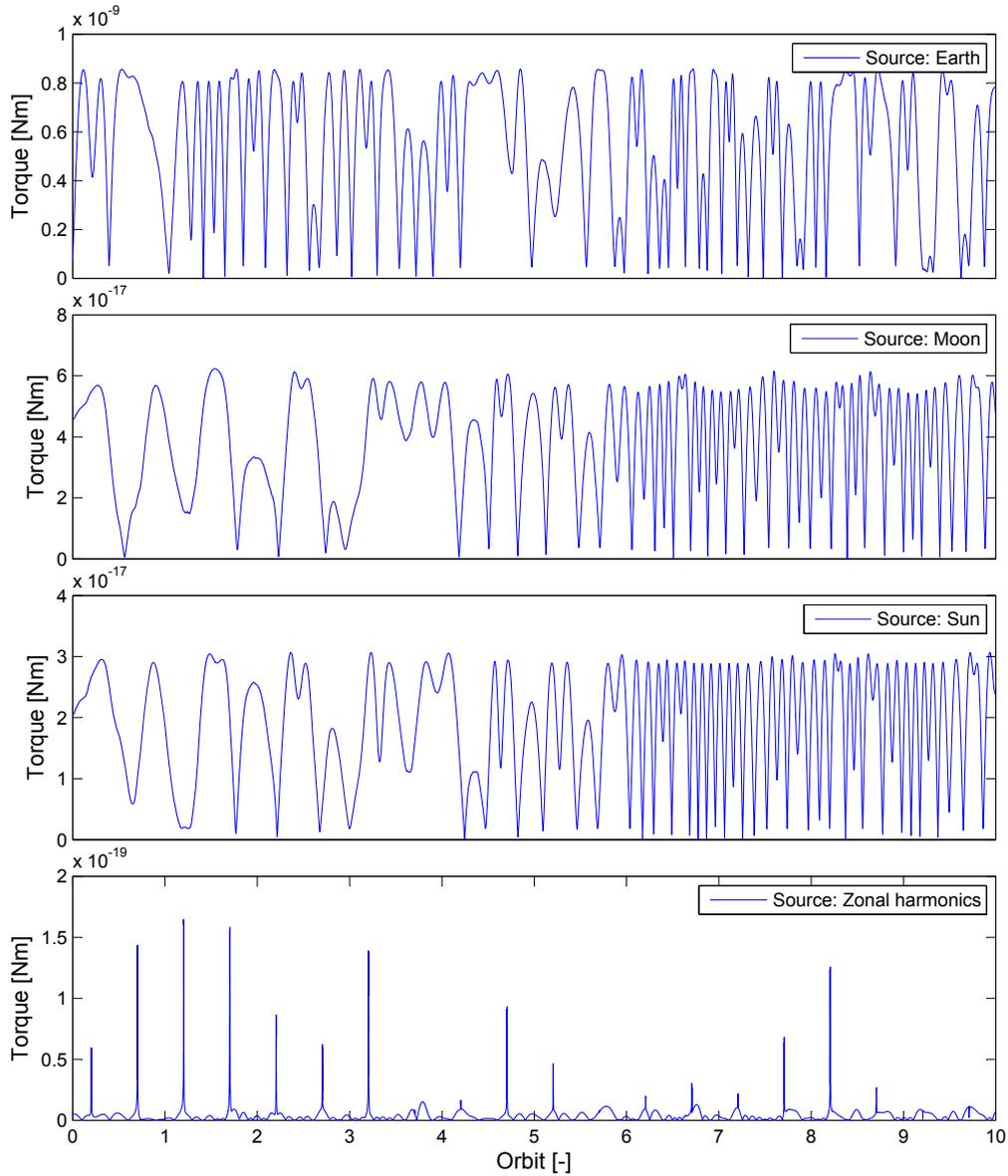


**Figure E.8:** Angular velocity in the SBRF (top), satellite attitude or orientation of the SBRF relative to the ECI (middle) and satellite trajectory (bottom).

The angular velocity starts out in  $(0,0,0)$  [rad/s] and varies slightly due to the disturbances. This gives an attitude that changes over time, within an interval of  $\pm 180$  [deg]. It is clear, that a satellite without control, will tumble because of disturbances. If the disturbances are removed, the attitude does not change during all 10 orbits.

As the bottom plot in Figure E.8 shows, 10 orbits covers most of the surface of the Earth, giving a good representative sample space of possible disturbance magnitudes.

The gravitational disturbances on the satellite from the Earth, the Sun and the Moon are plotted in Figure E.9.

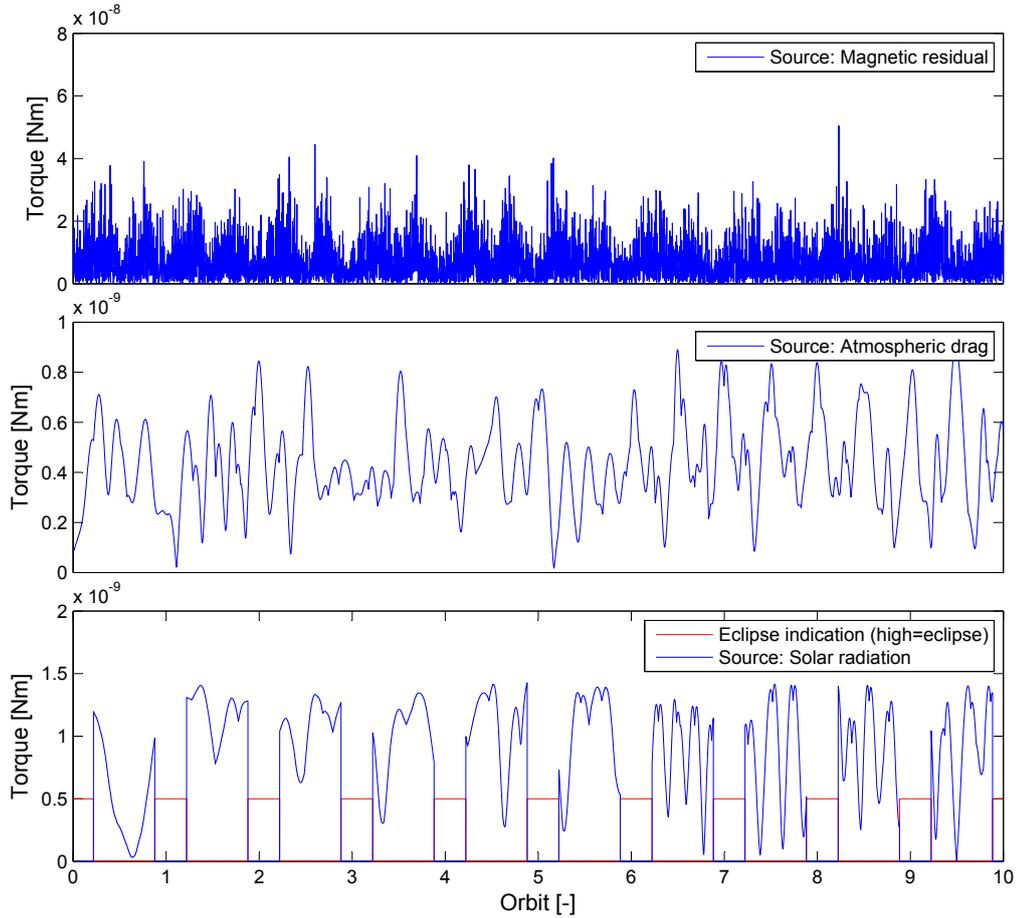


**Figure E.9:** Magnitude of disturbance torques acting on the satellite. From the Earth (gravity gradient), the Moon, the Sun and zonal harmonics respectively.

The disturbance torques from the Sun and the Moon are approximately  $10^8$  times smaller than the gravity gradient torque caused by the Earth and the zonal harmonics are approximately  $10^{10}$  times smaller. These torques are therefore considered negligible.

The maximum gravity gradient torque during the 10 orbits was  $0.86 \cdot 10^{-9}$  [Nm]. This is close to and under the worst case torque of  $0.87 \cdot 10^{-9}$  [Nm], calculated in Appendix D.

The disturbance torques from the magnetic residual caused by electronics on board the satellite, atmospheric drag and solar radiation are plotted in Figure E.10.



**Figure E.10:** Magnitude of disturbance torques acting on the satellite. Magnetic residual (top), atmospheric drag (middle) and solar radiation with eclipse indication (bottom).

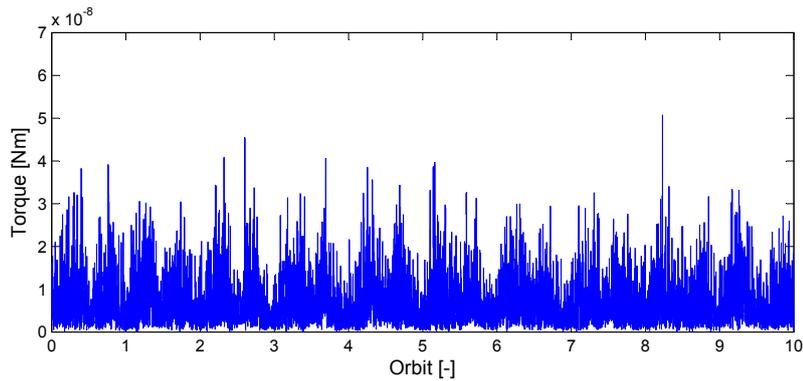
The torque caused by the magnetic residual experiences two peaks every orbit, corresponding to a higher magnetic field around the poles. The maximum torque during the 10 orbits was  $50.43 \cdot 10^{-9}$  [Nm]. This is again close to the worst case torque of  $48.00 \cdot 10^{-9}$  [Nm], calculated in Appendix D.

The atmospheric drag the satellite experiences during the 10 orbits, gave a maximum disturbance torque of  $0.91 \cdot 10^{-9}$  [Nm]. This was with an air density of  $5 \cdot 10^{-14}$  [kg/m<sup>3</sup>] for an orbit height of approximately 630 [km] [1]. The calculated worst case atmospheric drag is  $1.65 \cdot 10^{-9}$  [Nm], see Appendix D. There is a small deviation, but this is caused by the fact, that the worst case was calculated in a height of 600 [km], giving a higher air density

and thus a higher torque.

The eclipse indication shows, that there is no disturbance torque from the solar radiation, when the satellite is in eclipse, which is expected. The maximum torque during the 10 orbits was  $1.43 \cdot 10^{-9}$  [Nm], which is close to the worst case torque of  $1.36 \cdot 10^{-9}$  [Nm] calculation in Appendix D.

The magnitude of the sum of the vector disturbance torques (gravity gradient, magnetic residual, atmospheric drag and solar radiation) is plotted in Figure E.11.



*Figure E.11: Total magnitude of disturbance torques.*

During the 10 orbits the maximum torque was  $50.75 \cdot 10^{-9}$  [Nm], and this torque is dominated by the magnetic residual.

# Appendix **F**

## Extended Kalman Filter for Satellite Attitude Estimation

In this appendix the EKF theory from Chapter 5 is applied on a satellite attitude estimation problem. This includes working with quaternions as part of the state vector, which has advantages but also introduces problems addressed in Section 6.1.

### F.1 An EKF Implementation for Attitude Estimation

The EKF implementation for attitude estimation has the quaternion representing the rotation from the ECI to the CRF and the angular velocity of the CRF relative to the ECI as states.

$$\mathbf{x} = \left[ {}^c\mathbf{q}^T \quad {}^c\boldsymbol{\omega}^T \right]^T = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad \omega_1 \quad \omega_2 \quad \omega_3]^T \quad (\text{F.1})$$

The error state is defined as

$$\delta\mathbf{x} = \left[ \delta\mathbf{q}^T \quad \delta\boldsymbol{\omega}^T \right]^T = [\delta q_1 \quad \delta q_2 \quad \delta q_3 \quad \delta\omega_1 \quad \delta\omega_2 \quad \delta\omega_3]^T \quad (\text{F.2})$$

According to [4] the continuous time Jacobian matrix  $\underline{\mathbf{F}}(t)$  for the error state can be defined as

$$\underline{\mathbf{F}}(t) = \begin{bmatrix} -\underline{\mathbf{S}}({}^c\bar{\boldsymbol{\omega}}(t)) & \frac{1}{2}\mathbf{1}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & {}^c\mathbf{I}_{sat}^{-1} \left[ \underline{\mathbf{S}}({}^c\mathbf{I}_{sat} \quad {}^c\bar{\boldsymbol{\omega}}(t)) - \underline{\mathbf{S}}({}^c\bar{\boldsymbol{\omega}}(t)) {}^c\mathbf{I}_{sat} \right] \end{bmatrix} \quad (\text{F.3})$$

where:

${}^c\bar{\boldsymbol{\omega}}(t)$  is the working point (nominal) angular velocity.

$\mathbf{1}_{3 \times 3}$  is a 3x3 identity matrix.

${}^c\mathbf{I}_{sat}$  is the inertia matrix of the satellite given in the CRF.

This Jacobian matrix is based on a linearization of the satellite equations of motion defined in Eq. (2.6), except for the disturbance torque, which is assumed to be zero in the Jacobian. The deviation is also placed in Appendix G. Using the results in [4] it is also possible to express the discrete time Jacobian measurement model for the error state as (deviation placed in Appendix G).

$$\underline{\mathbf{H}}_k = \begin{bmatrix} 2\underline{\mathbf{S}}(\underline{\mathbf{v}}_{sun,k|k-1}) & \underline{\mathbf{0}}_{3 \times 3} \\ 2\underline{\mathbf{S}}(\underline{\mathbf{v}}_{mag,k|k-1}) & \underline{\mathbf{0}}_{3 \times 3} \\ \underline{\mathbf{0}}_{3 \times 3} & \underline{\mathbf{1}}_{3 \times 3} \end{bmatrix} \quad (\text{F.4})$$

Where:

$\underline{\mathbf{v}}_{sun,k|k-1}$  is the predicted sun vector measurement in the CRF.

$\underline{\mathbf{v}}_{mag,k|k-1}$  is the predicted magnetic field vector measurement in the CRF.

### F.1.1 Inputs

The inputs to the EKF implementation are the same as those for the UKF, defined in Subsection 6.3.1.

### F.1.2 Constants

The constants within the EKF implementation are the same as those for the UKF, defined in Subsection 6.3.2, without the array of weights.

### F.1.3 Implementation

Table F.1 presents the implemented EKF in pseudo code. Rotation of vectors  $\mathbf{v}$  with quaternions are written as  $\underline{\mathbf{A}}(\mathbf{q})\mathbf{v}$  and defined in Appendix A.

The steps in F.1 are similar to those in Table 6.1 in Subsection 6.3.3 and will not be further described here, except for step 1.3 where the discrete Jacobian of the model is found using the discretization approximation  $e^{\underline{\mathbf{A}} \cdot T_s} \approx \underline{\mathbf{1}} + \underline{\mathbf{A}} \cdot T_s$  valid for small steps [93].

Simulating the quaternion EKF is left as a future task, if the satellite should have both an EKF and an UKF implementation in the onboard software for improved robustness or if the UKF turns out to be too computationally demanding for the chosen MCU.

<b>Init.:</b>		
0.1	Initialize state:	$\mathbf{x}_{k k} = \left[ ({}^s\mathbf{q}_0 \otimes {}^c\mathbf{q})^T \quad (\mathbf{A}({}^c\mathbf{q})^s\boldsymbol{\omega}_0)^T \right]^T$
0.2	Save:	${}^s\mathbf{v}_{sun,k}, {}^s\mathbf{v}_{mag,k}, {}^s\boldsymbol{\omega}_k, {}^s\mathbf{N}_{ctrl,k}$ go to 2.14
<b>Predict:</b>		
1.1	Rotate control torque:	${}^c\mathbf{N}_{ctrl,k-1} = \mathbf{A}({}^c\mathbf{q}) {}^s\mathbf{N}_{ctrl,k-1}$
1.2	Numerical propagation:	$\mathbf{x}_{k k-1} = \text{RK4}(\mathbf{x}_{k-1 k-1}, {}^c\mathbf{N}_{ctrl,k-1}, Ts, steps)$
1.3	Calculate discrete Jacobian:	$\mathbf{\Phi}_{k-1 k-1} = \mathbf{1}_{6 \times 6} + Ts \cdot \mathbf{F}(\mathbf{x}_{k-1 k-1})$ where $\mathbf{F}(\mathbf{x}_{k-1 k-1})$ is defined in Eq. (F.3)
1.4	A priori error covariance:	$\mathbf{P}_{k k-1} = \mathbf{\Phi}_{k-1 k-1} \mathbf{P}_{k-1 k-1} \mathbf{\Phi}_{k-1 k-1}^T + \mathbf{Q}$
<b>Update:</b>		
2.1	Save:	${}^s\mathbf{v}_{sun,k}, {}^s\mathbf{v}_{mag,k}, {}^s\boldsymbol{\omega}_k, {}^s\mathbf{N}_{ctrl,k}$
2.2	Eclipse check:	If in eclipse go to 2.5, else go to 2.4
2.3	New measurement?:	If new Sun vector measurement go to 2.4, else go to 2.5
2.4	Normalize and rotate:	${}^c\mathbf{v}_{sun,k} = \mathbf{A}({}^c\mathbf{q}) \frac{{}^s\mathbf{v}_{sun,k}}{\ {}^s\mathbf{v}_{sun,k}\ }$ ${}^c\mathbf{v}_{sun,k k-1} = \mathbf{A}({}^c\mathbf{q}_{k k-1}) \frac{{}^s\mathbf{v}_{sun,k k-1}}{\ {}^s\mathbf{v}_{sun,k k-1}\ }$ go to 2.6
2.5	Hardcode vector:	${}^c\mathbf{v}_{sun,k} = {}^c\mathbf{v}_{sun,k k-1} = [0 \ 0 \ 0]^T$
2.6	Repeat step 2.3 - 2.5:	For magnetic field vectors ( ${}^c\mathbf{v}_{mag,k}, {}^c\mathbf{v}_{mag,k k-1}$ ) and angular velocities ( ${}^c\boldsymbol{\omega}_k, {}^c\boldsymbol{\omega}_{k k-1}$ )
2.7	Calculate Jacobian:	$\mathbf{H}_k = \begin{bmatrix} 2\mathbf{S}({}^c\mathbf{v}_{sun,k k-1}) & \mathbf{0}_{3 \times 3} \\ 2\mathbf{S}({}^c\mathbf{v}_{mag,k k-1}) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{1}_{3 \times 3} \end{bmatrix}$
2.8	Calculate Kalman gain:	$\mathbf{K}_k = \mathbf{P}_{k k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k k-1} \mathbf{H}_k^T + \mathbf{R})^{-1}$
2.9	Calculate error state:	$\delta \mathbf{x}_{k k} = \mathbf{K}_k (\mathbf{z}_k - \mathbf{z}_{k k-1})$
2.10	Expand quaternion:	${}^c\mathbf{q}_{k k} = \left[ \delta \mathbf{q}_{k k}^T \quad \sqrt{1 - \delta \mathbf{q}_{k k}^T \cdot \delta \mathbf{q}_{k k}} \right]^T \otimes {}^c\mathbf{q}_{k k-1}$
2.11	Calculate full state:	$\mathbf{x}_{k k} = \left[ {}^c\mathbf{q}_{k k}^T \quad ({}^c\boldsymbol{\omega}_{k k-1} + \delta \boldsymbol{\omega}_{k k})^T \right]^T$
2.12	A posteroi covariance:	$\mathbf{P}_{k k} = (\mathbf{1}_{6 \times 6} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k k-1}$
2.13	Rotate and output:	Output = $\left[ ({}^c\mathbf{q}_{k k} \otimes {}^c\mathbf{q}^{-1})^T \quad (\mathbf{A}({}^c\mathbf{q}^{-1}) {}^c\boldsymbol{\omega}_{k k})^T \right]^T$
2.14	Repeat:	Go to 1.1 and iterate filter every time step $T_s$

Table F.1: Pseudo code for a quaternion EKF implementation.



## Deviation of Jacobian Matrices

The EKF implementation needs a linearized small signal version of the non-linear satellite equations, derived in Section 2.4. The satellite equations are summarized in Equation G.1, where the reference frame notation is omitted for simplicity<sup>1</sup>.

$$\begin{bmatrix} \dot{\mathbf{q}}(t) \\ \dot{\boldsymbol{\omega}}(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \underline{\boldsymbol{\Omega}}(\boldsymbol{\omega}(t)) \mathbf{q}(t) \\ \underline{\mathbf{I}}_{sat}^{-1} [-\underline{\mathbf{S}}(\boldsymbol{\omega}(t)) (\underline{\mathbf{I}}_{sat} \boldsymbol{\omega}(t)) + \mathbf{N}_{dist}(t) + \mathbf{N}_{ctrl}(t)] \end{bmatrix} \quad (\text{G.1})$$

The kinematic and dynamic equations are linearized separately around a working point. The linearized small signal model forms the 6x6 Jacobin matrix, used in the EKF implementation to calculate the error state.

Furthermore the Jacobin matrix of the measurement model for the error state is derived. The deviation is based on a linearization of Eq. (G.2), where  $\mathbf{B}$  is a measurement vector and  $\underline{\mathbf{C}}(\mathbf{q})$  is a rotation matrix.

$${}^c \mathbf{B} = {}^c \underline{\mathbf{C}}(\mathbf{q})^t \mathbf{B} \quad (\text{G.2})$$

### G.1 Linearization of Kinematic Equation

The kinematic equation for the satellite is

$$\dot{\mathbf{q}}(t) = \frac{1}{2} \underline{\boldsymbol{\Omega}}(\boldsymbol{\omega}(t)) \mathbf{q}(t) \quad (\text{G.3})$$

<sup>1</sup>The quaternions represent the rotation from the ECI to the CRF and omega is the angular velocity of the CRF relative to the ECI, given in the CRF.

By introducing a quaternion  $\mathbf{q}_\omega(t) \equiv \mathbf{i}\omega_1(t) + \mathbf{j}\omega_2(t) + \mathbf{k}\omega_3(t) + 0$  defined in Eq. (G.4)

$$\begin{aligned} \mathbf{q}(t) \otimes \mathbf{q}_\omega(t) &= (\mathbf{i}q_1(t) + \mathbf{j}q_2(t) + \mathbf{k}q_3(t) + q_4(t)) (\mathbf{i}\omega_1(t) + \mathbf{j}\omega_2(t) + \mathbf{k}\omega_3(t) + 0) \quad (\text{G.4}) \\ &= \underbrace{\begin{bmatrix} 0 & \omega_3(t) & -\omega_2(t) & \omega_1(t) \\ -\omega_3(t) & 0 & \omega_1(t) & \omega_2(t) \\ \omega_2(t) & -\omega_1(t) & 0 & \omega_3(t) \\ -\omega_1(t) & -\omega_2(t) & -\omega_3(t) & 0 \end{bmatrix}}_{\underline{\boldsymbol{\Omega}}(\omega(t))} \underbrace{\begin{bmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \\ q_4(t) \end{bmatrix}}_{\mathbf{q}(t)} \end{aligned}$$

the kinematic equation can be rewritten to

$$\dot{\mathbf{q}}(t) = \frac{1}{2} \mathbf{q}(t) \otimes \mathbf{q}_\omega(t) \quad (\text{G.5})$$

The quaternion  $\mathbf{q}(t)$  is first split in its nominal value  $\bar{\mathbf{q}}(t)$  (working point) and its small signal value  $\tilde{\mathbf{q}}(t)$ . This is done in order to find the linearized small signal version of the non-linear satellite equations that defines  $\dot{\tilde{\mathbf{q}}}(t)$ , which is the goal.

$$\mathbf{q}(t) \equiv \bar{\mathbf{q}}(t) \otimes \tilde{\mathbf{q}}(t) \quad (\text{G.6})$$

$$\tilde{\mathbf{q}}(t) \equiv \bar{\mathbf{q}}^*(t) \otimes \mathbf{q}(t) \quad (\text{G.7})$$

Then the time derivative of the small signal value in Eq. (G.7) is

$$\dot{\tilde{\mathbf{q}}}(t) = \dot{\bar{\mathbf{q}}^*}(t) \otimes \mathbf{q}(t) + \bar{\mathbf{q}}^*(t) \otimes \dot{\mathbf{q}}(t) \quad (\text{G.8})$$

Substituting Eq. (G.5) into Eq. (G.8) gives

$$\dot{\tilde{\mathbf{q}}}(t) = \frac{1}{2} [(\bar{\mathbf{q}}(t) \otimes \mathbf{q}_\omega(t))^* \otimes \mathbf{q}(t) + \bar{\mathbf{q}}^*(t) \otimes (\mathbf{q}(t) \otimes \mathbf{q}_\omega(t))] \quad (\text{G.9})$$

Substituting Eq. (G.7) into Eq. (G.9) and utilizing that  $\mathbf{q}_\omega^*(t) = -\mathbf{q}_\omega(t)$  gives

$$\begin{aligned} \dot{\tilde{\mathbf{q}}}(t) &= \frac{1}{2} [(\bar{\mathbf{q}}(t) \otimes \mathbf{q}_\omega(t))^* \otimes \mathbf{q}(t) + \tilde{\mathbf{q}}(t) \otimes \mathbf{q}_\omega(t)] \\ &= \frac{1}{2} [\mathbf{q}_\omega^*(t) \otimes \bar{\mathbf{q}}^*(t) \otimes \mathbf{q}(t) + \tilde{\mathbf{q}}(t) \otimes \mathbf{q}_\omega(t)] \\ &= \frac{1}{2} [-\mathbf{q}_\omega(t) \otimes \tilde{\mathbf{q}}(t) + \tilde{\mathbf{q}}(t) \otimes \mathbf{q}_\omega(t)] \quad (\text{G.10}) \end{aligned}$$

Introducing the definition

$$\mathbf{q}_\omega(t) \equiv \mathbf{q}_{\bar{\omega}}(t) + \mathbf{q}_{\tilde{\omega}}(t) \quad (\text{G.11})$$

gives

$$\begin{aligned} \dot{\mathbf{q}}(t) &= \frac{1}{2} [-\mathbf{q}_{\bar{\omega}}(t) \otimes \tilde{\mathbf{q}}(t) + \tilde{\mathbf{q}}(t) \otimes \mathbf{q}_{\bar{\omega}}(t) + \tilde{\mathbf{q}}(t) \otimes \mathbf{q}_{\tilde{\omega}}(t)] \\ &= -\frac{1}{2} \mathbf{q}_{\bar{\omega}}(t) \otimes \tilde{\mathbf{q}}(t) + \frac{1}{2} \tilde{\mathbf{q}}(t) \otimes \mathbf{q}_{\bar{\omega}}(t) + \frac{1}{2} \tilde{\mathbf{q}}(t) \otimes \mathbf{q}_{\tilde{\omega}}(t) \end{aligned} \quad (\text{G.12})$$

Eq. (G.12) can be rewritten and linearized, which is done in the following. The first part of Eq. (G.12) can be rewritten to

$$\begin{aligned} -\frac{1}{2} \mathbf{q}_{\bar{\omega}}(t) \otimes \tilde{\mathbf{q}}(t) &= -\frac{1}{2} \begin{bmatrix} -\underline{\mathbf{S}}(\tilde{\mathbf{q}}_{1:3}(t)) + \tilde{\mathbf{q}}_4(t) \mathbf{1}_{3 \times 3} & \tilde{\mathbf{q}}_{1:3}(t) \\ -\tilde{\mathbf{q}}_{1:3}^T(t) & \tilde{\mathbf{q}}_4(t) \end{bmatrix} \begin{bmatrix} \bar{\omega}(t) \\ 0 \end{bmatrix} \\ &= -\frac{1}{2} \begin{bmatrix} \underline{\mathbf{S}}(\bar{\omega}(t)) \tilde{\mathbf{q}}_{1:3}(t) + \tilde{\mathbf{q}}_4(t) \mathbf{1}_{3 \times 3} \bar{\omega}(t) \\ -\tilde{\mathbf{q}}_{1:3}^T(t) \bar{\omega}(t) \end{bmatrix} \end{aligned} \quad (\text{G.13})$$

where

$$-\underline{\mathbf{S}}(\tilde{\mathbf{q}}_{1:3}(t)) \equiv \begin{bmatrix} 0 & \tilde{\mathbf{q}}_3(t) & -\tilde{\mathbf{q}}_2(t) \\ -\tilde{\mathbf{q}}_3(t) & 0 & \tilde{\mathbf{q}}_1(t) \\ \tilde{\mathbf{q}}_2(t) & -\tilde{\mathbf{q}}_1(t) & 0 \end{bmatrix} \quad (\text{G.14})$$

and

$$-\underline{\mathbf{S}}(\tilde{\mathbf{q}}_{1:3}(t)) \bar{\omega}(t) = \underline{\mathbf{S}}(\bar{\omega}(t)) \tilde{\mathbf{q}}_{1:3}(t) \quad (\text{G.15})$$

The second part of Eq. (G.12) can be rewritten to

$$\frac{1}{2} \tilde{\mathbf{q}}(t) \otimes \mathbf{q}_{\bar{\omega}}(t) = \frac{1}{2} \begin{bmatrix} -\underline{\mathbf{S}}(\bar{\omega}(t)) & \bar{\omega}(t) \\ -\bar{\omega}^T(t) & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{q}}_{1:3}(t) \\ \tilde{\mathbf{q}}_4(t) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\underline{\mathbf{S}}(\bar{\omega}(t)) \tilde{\mathbf{q}}_{1:3}(t) + \bar{\omega}(t) \tilde{\mathbf{q}}_4(t) \\ -\bar{\omega}^T(t) \tilde{\mathbf{q}}_{1:3}(t) \end{bmatrix} \quad (\text{G.16})$$

By introducing the small rotation approximation

$$\lim_{\text{rotation} \rightarrow 0} \mathbf{q}(t) = \lim_{\theta \rightarrow 0} \begin{bmatrix} \mathbf{q}_{1:3}(t) \\ \mathbf{q}_4(t) \end{bmatrix} = \lim_{\theta \rightarrow 0} \begin{bmatrix} u_{c1} \sin\left(\frac{\theta(t)}{2}\right) \\ u_{c2} \sin\left(\frac{\theta(t)}{2}\right) \\ u_{c3} \sin\left(\frac{\theta(t)}{2}\right) \\ \cos\left(\frac{\theta(t)}{2}\right) \end{bmatrix} \approx \begin{cases} \mathbf{q}_{1:3}(t) \rightarrow 0 \\ \mathbf{q}_4(t) \rightarrow 1 \end{cases} \quad (\text{G.17})$$

it is possible to approximate the third part of Eq. (G.12) with

$$\begin{aligned} \frac{1}{2}\tilde{\mathbf{q}}(t) \otimes \mathbf{q}_{\tilde{\omega}}(t) &= \frac{1}{2} \begin{bmatrix} -\underline{\mathbf{S}}(\tilde{\omega}(t)) & \tilde{\omega}(t) \\ -\tilde{\omega}^T(t) & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{q}}_{1:3}(t) \\ \tilde{\mathbf{q}}_4(t) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\underline{\mathbf{S}}(\tilde{\omega}(t))\tilde{\mathbf{q}}_{1:3}(t) + \tilde{\omega}(t)\tilde{\mathbf{q}}_4(t) \\ -\tilde{\omega}^T(t)\tilde{\mathbf{q}}_{1:3}(t) \end{bmatrix} \\ &\approx \frac{1}{2} \begin{bmatrix} 0 + \begin{bmatrix} \tilde{\omega}_1(t) \\ \tilde{\omega}_2(t) \\ \tilde{\omega}_3(t) \end{bmatrix} \tilde{\mathbf{q}}_4 \\ 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \tilde{\omega}_1(t) \\ \tilde{\omega}_2(t) \\ \tilde{\omega}_3(t) \\ 0 \end{bmatrix} = \frac{1}{2}\mathbf{q}_{\tilde{\omega}}(t) \end{aligned} \quad (\text{G.18})$$

Now Eq. (G.13), (G.16) and (G.18) can be substituted into Eq. (G.12) giving

$$\begin{aligned} \dot{\mathbf{q}}(t) &\approx -\frac{1}{2} \begin{bmatrix} \underline{\mathbf{S}}(\tilde{\omega}(t))\tilde{\mathbf{q}}_{1:3}(t) + \tilde{\mathbf{q}}_4(t)\mathbf{1}_{3 \times 3}\tilde{\omega}(t) \\ -\tilde{\mathbf{q}}_{1:3}^T(t)\tilde{\omega}(t) \end{bmatrix} + \frac{1}{2} \begin{bmatrix} -\underline{\mathbf{S}}(\tilde{\omega}(t))\tilde{\mathbf{q}}_{1:3}(t) + \tilde{\omega}(t)\tilde{\mathbf{q}}_4(t) \\ -\tilde{\omega}^T(t)\tilde{\mathbf{q}}_{1:3}(t) \end{bmatrix} + \frac{1}{2}\mathbf{q}_{\tilde{\omega}}(t) \\ &\approx \begin{bmatrix} -\underline{\mathbf{S}}(\tilde{\omega}(t))\tilde{\mathbf{q}}_{1:3}(t) \\ 0 \end{bmatrix} + \frac{1}{2}\mathbf{q}_{\tilde{\omega}}(t) \end{aligned} \quad (\text{G.19})$$

## G.2 Linearization of Dynamic Equation

The dynamic equation for the satellite is

$$\dot{\omega}(t) = \underline{\mathbf{I}}_{sat}^{-1}[-\underline{\mathbf{S}}(\omega(t))(\underline{\mathbf{I}}_{sat}\omega(t)) + \mathbf{N}_{dist}(t) + \mathbf{N}_{ctrl}(t)] \quad (\text{G.20})$$

Eq. (G.20) is linearized with a first order Taylor expansion, where  $\mathbf{N}_{dist}(t)$  is removed (assumed to have an expected value of zero).

$$\begin{aligned} \dot{\omega}(t) &\approx \underline{\mathbf{I}}_{sat}^{-1}[-\underline{\mathbf{S}}(\tilde{\omega}(t))(\underline{\mathbf{I}}_{sat}\tilde{\omega}(t)) + \tilde{\mathbf{N}}_{ctrl}(t)] - \underline{\mathbf{I}}_{sat}^{-1} \frac{dt}{d\omega(t)} \left( \underline{\mathbf{S}}(\omega)(\underline{\mathbf{I}}_{sat}\omega(t)) \right) \Big|_{\omega(t)=\tilde{\omega}(t)} \cdot \tilde{\omega}(t) \\ &\quad + \underline{\mathbf{I}}_{sat}^{-1} \frac{dt}{d\mathbf{N}_{ctrl}(t)} \mathbf{N}_{ctrl}(t) \Big|_{\mathbf{N}_{ctrl}(t)=\tilde{\mathbf{N}}_{ctrl}(t)} \cdot \tilde{\mathbf{N}}_{ctrl}(t) \\ &\approx \underline{\mathbf{I}}_{sat}^{-1}[-\underline{\mathbf{S}}(\tilde{\omega}(t))(\underline{\mathbf{I}}_{sat}\tilde{\omega}(t)) + \tilde{\mathbf{N}}_{ctrl}(t)] - \underline{\mathbf{I}}_{sat}^{-1} \underline{\mathbf{S}}(\mathbf{1}_{3 \times 3}) \underline{\mathbf{I}}_{sat} \tilde{\omega}(t) \tilde{\omega}(t) - \underline{\mathbf{I}}_{sat}^{-1} \underline{\mathbf{S}}(\tilde{\omega}) \underline{\mathbf{I}}_{sat} \tilde{\omega}(t) \\ &\quad + \underline{\mathbf{I}}_{sat}^{-1} \tilde{\mathbf{N}}_{ctrl}(t) \\ &\approx \underline{\mathbf{I}}_{sat}^{-1}[-\underline{\mathbf{S}}(\tilde{\omega}(t))(\underline{\mathbf{I}}_{sat}\tilde{\omega}(t)) + \tilde{\mathbf{N}}_{ctrl}(t)] + \underline{\mathbf{I}}_{sat}^{-1} [\underline{\mathbf{S}}(\underline{\mathbf{I}}_{sat}\tilde{\omega}(t)) - \underline{\mathbf{S}}(\tilde{\omega}(t))\underline{\mathbf{I}}_{sat}] \tilde{\omega}(t) + \underline{\mathbf{I}}_{sat}^{-1} \tilde{\mathbf{N}}_{ctrl}(t) \end{aligned} \quad (\text{G.21})$$

By splitting an angular velocity  $\omega(t)$  in its nominal value  $\tilde{\omega}(t)$  (working point) and a small signal value  $\tilde{\omega}(t)$  as shown in Eq. (G.22)

$$\omega(t) = \tilde{\omega}(t) + \tilde{\omega}(t) \quad (\text{G.22})$$

it is possible to derive the small signal dynamic equation from Eq. (G.21)

$$\dot{\tilde{\omega}}(t) \approx \underline{\mathbf{I}}_{sat}^{-1} [\underline{\mathbf{S}}(\underline{\mathbf{I}}_{sat} \tilde{\omega}(t)) - \underline{\mathbf{S}}(\tilde{\omega}(t)) \underline{\mathbf{I}}_{sat}] \tilde{\omega}(t) + \underline{\mathbf{I}}_{sat}^{-1} \tilde{\mathbf{N}}_{ctrl}(t) \quad (\text{G.23})$$

### G.3 Linearized Measurement Model

A linearization of the measurement model about the a priori state estimate is necessary in order to process the measurements in the EKF. The non-linear measurement model is

$${}^c \mathbf{v}_{mag,k} = {}^c \underline{\mathbf{C}}(\mathbf{q}_k) {}^i \mathbf{v}_{mag,k} \quad (\text{G.24})$$

where  ${}^i \mathbf{v}_{mag,k}$  is the geomagnetic field vector in the ECI at time  $k$ , which is given by the onboard IGRF model and  ${}^c \underline{\mathbf{C}}(\mathbf{q}_k)$  is a rotation matrix that rotates the geomagnetic field vector from the ECI to the CRF, which is where the sensor measurement is represented.

According to [3] it is possible to write a rotation matrix  $\underline{\mathbf{C}}(\mathbf{q})$  as

$$\underline{\mathbf{C}}(\mathbf{q}) = (\mathbf{q}_4^2 - \mathbf{q}_{1:3}^T \mathbf{q}_{1:3}) \underline{\mathbf{1}}_{3 \times 3} + 2 \mathbf{q}_{1:3} \mathbf{q}_{1:3}^T - 2 \mathbf{q}_4 \underline{\mathbf{S}}(\mathbf{q}_{1:3}) \quad (\text{G.25})$$

By using the small rotation approximation defined in Eq. (G.17) it is possible to approximate Eq. (G.25) with

$$\begin{aligned} \underline{\mathbf{C}}(\tilde{\mathbf{q}}) &\approx \underline{\mathbf{1}}_{3 \times 3} - 2 \underline{\mathbf{S}}(\tilde{\mathbf{q}}_{1:3}) \\ \underline{\mathbf{C}}(\mathbf{q}) &= \underline{\mathbf{C}}(\tilde{\mathbf{q}}) \underline{\mathbf{C}}(\tilde{\mathbf{q}}) \\ &\approx \underline{\mathbf{C}}(\tilde{\mathbf{q}}) (\underline{\mathbf{1}}_{3 \times 3} - 2 \underline{\mathbf{S}}(\tilde{\mathbf{q}}_{1:3})) \end{aligned} \quad (\text{G.26})$$

By substituting Eq. (G.26) into Eq. (G.24), it is possible to isolate the small signal (error) measurement  ${}^c \tilde{\mathbf{v}}_{mag,k} = {}^c \mathbf{v}_{mag,k} - {}^c \tilde{\mathbf{v}}_{mag,k|k-1}$ , which gives

$$\begin{aligned} {}^c \mathbf{v}_{mag,k} &\approx {}^c \underline{\mathbf{C}}(\tilde{\mathbf{q}}_{k|k-1}) (\underline{\mathbf{1}}_{3 \times 3} - 2 \underline{\mathbf{S}}(\tilde{\mathbf{q}}_{1:3,k})) {}^i \mathbf{v}_{mag,k} \\ {}^c \mathbf{v}_{mag,k} &\approx (\underline{\mathbf{1}}_{3 \times 3} - 2 \underline{\mathbf{S}}(\tilde{\mathbf{q}}_{1:3,k})) {}^c \tilde{\mathbf{v}}_{mag,k|k-1} \\ {}^c \mathbf{v}_{mag,k} - {}^c \tilde{\mathbf{v}}_{mag,k|k-1} &\approx -2 \underline{\mathbf{S}}(\tilde{\mathbf{q}}_{1:3,k}) {}^c \tilde{\mathbf{v}}_{mag,k|k-1} \\ {}^c \mathbf{v}_{mag,k} - {}^c \tilde{\mathbf{v}}_{mag,k|k-1} &\approx 2 \underline{\mathbf{S}}(\tilde{\mathbf{v}}_{mag,k|k-1}) \tilde{\mathbf{q}}_{1:3,k} \end{aligned} \quad (\text{G.27})$$

The vector  ${}^c \tilde{\mathbf{v}}_{mag,k|k-1}$  is the predicted magnetic field measurement in the CRF.

Eq. (G.27) is similar for the sun vector small signal (error) measurement  ${}^c \tilde{\mathbf{v}}_{sun,k}$ .

## G.4 Jacobian Matrices

The model Jacobian matrix  $\underline{\mathbf{F}}(t)$  for the error state is found by gathering Eq. (G.19) and (G.23), which gives

$$\begin{bmatrix} \dot{\tilde{\mathbf{q}}}_{1:3}(t) \\ \dot{\tilde{\boldsymbol{\omega}}}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} -\underline{\mathbf{S}}(\tilde{\boldsymbol{\omega}}(t)) & \frac{1}{2}\underline{\mathbf{1}}_{3 \times 3} \\ \underline{\mathbf{0}}_{3 \times 3} & \underline{\mathbf{I}}_{sat}^{-1}[\underline{\mathbf{S}}(\underline{\mathbf{I}}_{sat}\tilde{\boldsymbol{\omega}}(t)) - \underline{\mathbf{S}}(\tilde{\boldsymbol{\omega}}(t))\underline{\mathbf{I}}_{sat}] \end{bmatrix}}_{\underline{\mathbf{F}}(t)} \begin{bmatrix} \tilde{\mathbf{q}}_{1:3}(t) \\ \tilde{\boldsymbol{\omega}}(t) \end{bmatrix} + \begin{bmatrix} \underline{\mathbf{0}}_{3 \times 3} \\ \underline{\mathbf{I}}_{sat}^{-1} \end{bmatrix} \tilde{\mathbf{N}}_{ctrl}(t) \quad (\text{G.28})$$

This corresponds to the result in [4].

The linearized small signal (error) measurement model about the a priori state estimate is found by using the result in Eq. (G.27), which gives

$$\tilde{\mathbf{z}}_k = \underline{\mathbf{H}}_k \begin{bmatrix} \tilde{\mathbf{q}}_{1:3,k} \\ \tilde{\boldsymbol{\omega}}_k \end{bmatrix} = \underbrace{\begin{bmatrix} 2\underline{\mathbf{S}}(c\tilde{\mathbf{v}}_{sun,k|k-1}) & \underline{\mathbf{0}}_{3 \times 3} \\ 2\underline{\mathbf{S}}(c\tilde{\mathbf{v}}_{mag,k|k-1}) & \underline{\mathbf{0}}_{3 \times 3} \\ \underline{\mathbf{0}}_{3 \times 3} & \underline{\mathbf{1}}_{3 \times 3} \end{bmatrix}}_{\underline{\mathbf{H}}_k} \begin{bmatrix} \tilde{\mathbf{q}}_{1:3,k} \\ \tilde{\boldsymbol{\omega}}_k \end{bmatrix} \quad (\text{G.29})$$

where  $\tilde{\mathbf{z}}_k$  is the small signal (error) measurement vector. Again this corresponds to the result in [4].

# Interface Control Document and Budgets

## H.1 Interface Control Document

AAUSAT3 is designed as a distributed system consisting of several subsystems. These subsystems must communicate with each other and the hardware must be compatible. Therefore an Interface Control Document (ICD) is made, which describes the inputs and outputs to and from the ADCS, both software and hardware wise. The ICD is a key element in system engineering.

### H.1.1 Communication

The ADCS must provide a CAN interface to both MCU's and use the Can Space Protocol (CSP) [94] for communication. This communication includes:

- ADCS1 and ADCS2 control (input commands).
- GPS control (input commands).
- TLE updates for the ADCS2 (input from ground station).
- Error messages (output to LOG).
- Warnings (output to LOG).
- State transitions (output to LOG).
- Logged data such as raw sensor values, attitude estimate, residuals, control signal, controller state and time stamp (output to ground station).

The commands that control the ADCS is presented in Table H.1. The AVR MCU both include ADCS1 and the GPS subsystem and the ARM MCU include the ADCS2. Individual CSP addresses are assigned for each of the three subsystems. The ADCS board must also provide the temperature of the board through an I<sup>2</sup>C interface to EPS.

<b>Telecommand</b>	<b>Description</b>
ADCS_ON	This command is sent to EPS, telling it to power up the ADCS.
ADCS_OFF	Turns off the power to ADCS.
ADCS_SET_DEFAULT_1	This command is sent to EPS, telling it which ADCS to use for detumbling per default. 1 means ADCS1 and 2 means ADCS2. Value saved in EEPROM.
ADCS_TLE	Command followed by a new TLE update for the ADCS. Format TBD.
ADCS2_IDLE	Switch to IDLE/SLEEP state if in STANDBY state.
ADCS2_DETUMBLE	Switch to DETUMBLE state if in IDLE, STANDBY or POINTING state.
ADCS2_STANDBY	Switch to STANDBY state if in IDLE, DETUMBLE, TEST or POINTING state.
ADCS2_TEST	Switch to TEST state if in STANDBY state.
ADCS2_POINTING	Switch to POINTING state if in DETUMBLE state. A state switch will only occur if the angular velocity is below a threshold.
ADCS2_POINT_REF1	Sets which reference to track. Replace 1 with the wanted reference identifier.
ADCS2_TEST_SEQ_????_????	TBD. Values saved in EEPROM.
ADCS2_SET_DIAGNOSTICS_0	Sets a guard to 0, meaning that a diagnostics test will not be performed during startup. Normally set to 1, which is on. Value saved in EEPROM.
ADCS2_SET_OMEGA_HYS1_00000	Sets the angular velocity hysteresis determining the change from DETUMBLE to POINTING state. 0000 should be replaced by values between TBD and TBD. Value saved in EEPROM.
ADCS2_SET_OMEGA_HYS2_00000	Sets the angular velocity hysteresis determining the change from POINTING to DETUMBLE state. 0000 should be replaced by values between TBD and TBD. Value saved in EEPROM.
ADCS2_SET_OMEGA_HYS3_00000	Sets the angular velocity hysteresis determining the change from TEST to DETUMBLE state. 0000 should be replaced by values between TBD and TBD. Value saved in EEPROM.

*continued...*

Telecommand	Description
ADCS2_SAMPLE_0001	The ADCS2 samples 0001 seconds of data (replace 0001 with integer value between 0000-6000) and puts it in a buffer. When done it sends the data as telemetry. Data is raw sensor values, attitude estimate, residuals, control signal, controller state and time stamp.
ADCS2_SET_DETUMBLE_00000	Changes the gain of the detumble controller to 00000 (replace 00000 with value between TBD and TBD). Value saved in EEPROM.
ADCS2_SET_ATTITUDE_REF_5_????	Sets reference attitude 5 of the satellite to ?????. Format TBD. Value saved in EEPROM.
ADCS2_SET_ATTITUDE_REF_6_????	Sets reference attitude 6 of the satellite to ?????. Format TBD. Value saved in EEPROM.
ADCS2_USE_MAGNETORQUER_SET_1	This command determines which magnetorquer set to use. 1 means magnetorquers without iron core (default setting) and 2 means magnetorquers with iron core. Value saved in EEPROM.
ADCS2_EN_DRIVER_1	Enables magnetorquer driver 1. Valid values are 1 to 9. This can be used to override decisions made by the supervisory system to disable drivers in case of detected faults. Value saved in EEPROM.
ADCS2_DIS_DRIVER_1	Disables magnetorquer driver 1. Valid values are 1 to 9. Value saved in EEPROM.

*Table H.1: Telecommands for control of the ADCS2. All commands are equivalent for ADCS1.*

Potential values to be stored in EEPROM is presented in Table H.2.

Three sets of identical variables should be kept, in order to make the system more fault tolerant.

### H.1.2 Hardware

All components must as a minimum be graded for -40 to 85 [°C] and be able to withstand vacuum conditions. Outgassing must be held below levels defined by the launch provider, see e.g. standard ECSS-Q-70 [95].

A stack connector supplies the ADCS board with power, CAN interface and I<sup>2</sup>C interface to temperature sensors. This connector is of type M20-6103245 and might be extended in length if more power channels are needed. Table H.3 shows the stack connector pin assignment.

Cast	Variable	Default value	Interval
int	default_adcs	1 (ADCS1)	{1, 2}
int	diagnostics	1 (on)	{0, 1}
float	omega_hys1	TBD	TBD
float	omega_hys2	TBD	TBD
float	omega_hys3	TBD	TBD
TBD	detumble_ctrl_gain	TBD	TBD
TBD	reference_attitude1	NUUK - Greenland	-
TBD	reference_attitude2	Aalborg - Denmark	-
TBD	reference_attitude3	The Earth center	-
TBD	reference_attitude4	The Sun	-
TBD	reference_attitude5	Not used	-
TBD	reference_attitude6	Not used	-
array	test_sequence	-	TBD
int	driver1_enabled	1 (enabled)	{0, 1}
int	driver2_enabled	1 (enabled)	{0, 1}
int	driver3_enabled	1 (enabled)	{0, 1}
int	driver4_enabled	0 (disabled)	{0, 1}
int	driver5_enabled	0 (disabled)	{0, 1}
int	driver6_enabled	0 (disabled)	{0, 1}
int	driver7_enabled	0 (disabled)	{0, 1}
int	driver8_enabled	0 (disabled)	{0, 1}
int	driver9_enabled	0 (disabled)	{0, 1}
int	magnetorquer_set	1	{1, 2}

**Table H.2:** Variables stored in EEPROM to save their value during power off/on cycles.

Pin no.	Name	Function	Comment
1	CANH	CAN differential pair wire	-
2	CANL	CAN differential pair wire	-
3	GND	-	-
4	GND	-	-
5	I2C_SCL	I <sup>2</sup> C clock	-
6	I2C_SDL	I <sup>2</sup> C data	-
7	GND	-	-
8	GND	-	-
9	SS0	3.3 [V] for temperature sensors	-
10	SS1	3.3 [V] for AIS1	-
11	SS2	? [V] for CAM	-
12	SS3	3.3 [V] for AIS2	-
13	SS4	3.0 [V] for GPS	Bat. supply for RTC on GPS
14	SS5	5.0 [V] for GPS	-
15	SS6	5.0 [V] for AIS2	-
16	SS7	5.0 [V] for power injection to LNA	Handled by AIS2 board
17	SS8	3.3 [V] for ADCS	Gyro and sun sensors
18	SS9	5.0 [V] for ADCS	Magnetorquers
19	SS10	3.3 [V] for ADCS	-
20	SS11	3.3 [V] for UHF	-
21	GND	-	-
22	GND	-	-

**Table H.3:** Pin assignment for stack connector.

The ADCS board must provide an interface to the GPS module. The connector type is M40-6205146 and Table H.4 shows the pin assignments.

Pin no.	Name	Function	Comment
1	VCC5V_GPS	5 [V] power supply	-
2	VANT_IN	Antenna power	Connected to 3.3 [V]
8	MULTI_FN_IO	Multi function I/O	Connected to AVR
11	U2TXDX	UART interface 2	Connected to AVR
13	U2RXDX	UART interface 2	Connected to AVR
15	TIC	Clock output (sec)	Not connected
19	U1TXDX	UART interface 1	Connected to AVR
21	U1RXDX	UART interface 1	Connected to AVR
25	P_ON_RESET	Power on reset	Connected to 3.3 [V]
30	GPIO7	General purpose I/O	Connected to 3.3 [V]
51	VCC3V_GPS_VBAT	3.0 [V] ext. battery supply	Not mandatory
31-32, 50, 52	GND	-	-
3-7, 9-10, 12, 14, 16-18, 20, 22-24, 26-29, 33-49	NC	-	Not connected

*Table H.4: Pin assignment for GPS connector.*

The ADCS board also provides interface connector for magnetorquers (see Table H.5), sun sensors (see Table H.6) and JTAG daisy chain (see Table H.7). These connectors are type S13B-ZR-SM4A-TF.

Pin no.	Name	Function	Comment
1	NC	-	Not Connected
2	Mag1_1_B	Coil X, first half	-
3	Mag1_1_A	Coil X, first half	-
4	Mag2_1_B	Coil X, second half	-
5	Mag2_1_A	Coil X, second half	-
6	Mag1_2_B	Coil Y, first half	-
7	Mag1_2_A	Coil Y, first half	-
8	Mag2_2_B	Coil Y, second half	-
9	Mag2_2_A	Coil Y, second half	-
10	Mag1_3_B	Coil Z, first half	-
11	Mag1_3_A	Coil Z, first half	-
12	Mag2_3_B	Coil Z, second half	-
13	Mag2_3_A	Coil Z, second half	-

*Table H.5: Pin assignment for magnetorquer connector.*

Connector placements and mounting hole sizes and placements are documented in the PCB layout placed on the appended CD.

Pin no.	Name	Function	Comment
1	SUN_EN1	Sun sensor 1 enable	-
2	SUN_EN2	Sun sensor 2 enable	-
3	SUN_EN3	Sun sensor 3 enable	-
4	SUN_EN4	Sun sensor 4 enable	-
5	SUN_EN5	Sun sensor 5 enable	-
6	SUN_EN6	Sun sensor 6 enable	-
7	SPI1_MOSI	SPI master out slave in	-
8	SPI1_MISO	SPI master in slave out	-
9	SPI1_SPCK	SPI clock	-
10	TWD	I <sup>2</sup> C data	-
11	TWCK	I <sup>2</sup> C clock	-
12	GND	-	-
13	VCC3V3_ADACS_sens	3.3 [V] power supply for sun sensors	-

*Table H.6: Pin assignment for sun sensor connector. Both SPI and I<sup>2</sup>C interface is provided.*

Pin no.	Name	Function	Comment
1	TDI_MOSI (bottom connector)	SPI target data in	
	TDO_MISO (top connector)	SPI target data out	-
2	TDO_MISO_return	SPI target data out	Last board returns TDO to JTAG device
3	TCK_SCK	SPI clock	-
4	TMS_RST	SPI reset	-
5	GND	-	-
6	VCC	3.3 [V] sense	Only one board provides this
7	UART_TX	UART interface	Only connected to ARM
8	UART_RX	UART interface	Only connected to ARM
9	-	-	Subsystem specific
10	-	-	Subsystem specific
11	-	-	Subsystem specific
12	Reset	Resets ARM	Subsystem specific
13	Reset2	Resets AVR	Subsystem specific

*Table H.7: Pin assignment for JTAG daisy chain connector.*

## H.2 Budgets

### H.2.1 Power

The estimated power consumption for the ADACS is presented in Table H.8.

The values are not based on any measurements, but on datasheets and estimates. Average detumbling power consumption of 81 [mW] and average pointing power consumption of 460 [mW] serve as performance measures when evaluating the controllers.

The states not addressed in Table H.8 have the power consumptions shown in Table H.9.

Approximately 100 [mW] of power is saved under worst case conditions, if the magnetorquers with iron core are used instead of magnetorquers without.

State and part	Worst case [mW]	Average [mW]	On time [%]	Total [mW]
ADCS1 Detumble total:	183	<b>81</b>	90	81
- Magnetorquers	135	33	90	29.7
- Magnetometer	15	15	90	13.5
- AVR + CAN transceiver	42	42	90	37.8
ADCS2 Pointing total:	530	<b>460</b>	10	46
- Magnetorquers	135	65	10	6.5
- Magnetometer	15	15	10	1.5
- Sun sensors	100	100	10	10
- Gyroscopes	40	40	10	4
- ARM + CAN transceiver	240	240	10	24
Total power consumed by ADCS	-	-	100	127

**Table H.8:** Estimated power consumption for ADCS. Worst case is with maximum actuation on all magnetorquer coils without core. “On time” is the estimated percentage of time the state will be used, and the total is calculated from this value and average consumption.

State	Worst case [mW]	Average [mW]	Comment
ADCS1 Off	0	0	
ADCS2 Off	0	0	
ADCS1 Idle	30	24	CAN active.
ADCS2 Idle	30	24	CAN active.
ADCS1 Standby	57	57	Magnetometer active.
ADCS2 Standby	395	395	Attitude estimation active.
ADCS1 Test	192	122	Magnetometer and actuator active.
ADCS2 Test	530	460	Attitude estimation and actuator active.
ADCS1 Diagnostics	192	122	Magnetometer and actuator active.
ADCS2 Diagnostics	530	460	Sensor sampling and actuator active.
ADCS2 Detumble	390	288	Magnetometer and actuator active.

**Table H.9:** Estimated power consumption for other states in ADCS.

## H.2.2 Size

It has been accepted by the system engineering group that the ADCS has the following sizes (within limits available in the Cubesat):

- ADCS board dimensions (height includes the coils with iron core and the Phoenix GPS module):
  - Length: 87 [mm]
  - Width: 87 [mm]
  - Height: 25 [mm]
- Magnetorquer dimensions (placed perpendicular to each other on three side panels):
  - Length: 75 [mm]
  - Width: 75 [mm]
  - Cross sectional area: 2.5x4 [mm<sup>2</sup>]

- Maximum sun sensor board dimensions<sup>1</sup> (one board will be placed on each of the six side panels):
  - Length: 20 [mm]
  - Width: 20 [mm]
  - Height: 4 [mm]

Furthermore the ADCS board must adhere to the space board design layout made for AAUSAT3, to fit into the stack and for modularity.

### H.2.3 Mass

The mass budget for the ADCS is 0.2 [kg] (determined by system engineering group).

The ADCS has been calculated/estimated to weigh:

- Magnetorquer coils with iron core: 0.0186 [kg]
- Magnetorquer coils without core: 0.0534 [kg]
- ADCS prototype board: 0.032 [kg]
- Sun sensor boards: 0.090 [kg]
- Total weight: 0.194 [kg]

The Total weight is within the budget, but it is only a first estimate of the weight of the flight ready ADCS.

### H.2.4 Data Storage

In order to be able to save at least one orbit of data from the ADCS1, approximately 1.3 [MB] storage space must be available at a sampling frequency of 10 [Hz]. Saved data is raw magnetometer measurements, control signal and controller state.

In order to be able to save at least one orbit of data from the ADCS2, approximately 1 [MB] storage space must be available at a sampling frequency of 1 [Hz]. Saved data is raw sensor values, attitude estimate, residuals, control signal, controller state and time stamp.

Additional space for software images must be available.

### H.2.5 Economy

An estimated cost for one ADCS prototype board is presented in Table H.10. Prizes are from component vendor Digikey Corp. and Elprint APS.

---

<sup>1</sup>Subject to changes.

<b>Part</b>	<b>Amount</b>	<b>Total cost (DKK)</b>
ADCS PCB manufacturing (5 is min. amount)	5	2057
HMC6343 (magnetometer)	2	1200
HMC5843 (magnetometer)	2	192
SI9988 (driver)	9	360
ISZ-1215 (gyro)	1	33
IDG-1215 (gyro)	1	45
AT90CAN128 (AVR)	1	84
AT91SAM7A3 (ARM)	1	68
M25P128 (Flash memory)	2	138
Miscellaneous components	-	500
Total cost	-	4677

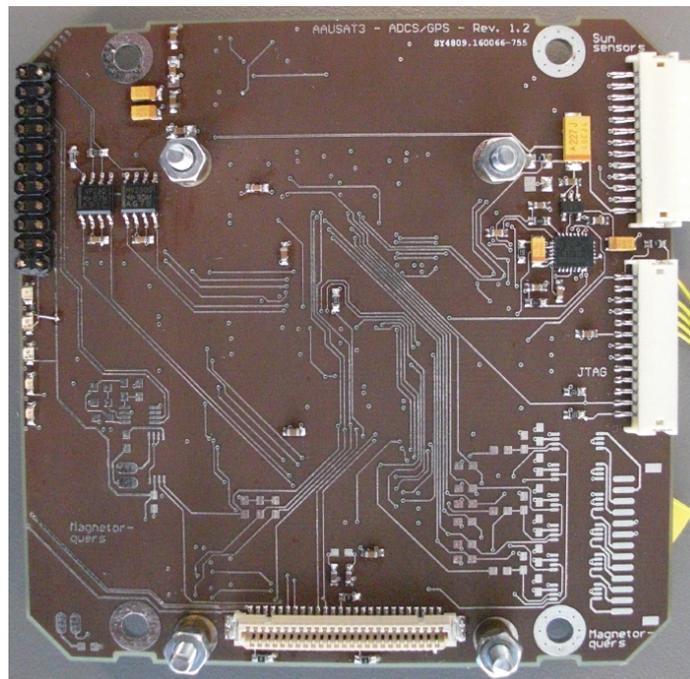
*Table H.10: Estimated ADCS prototype cost.*



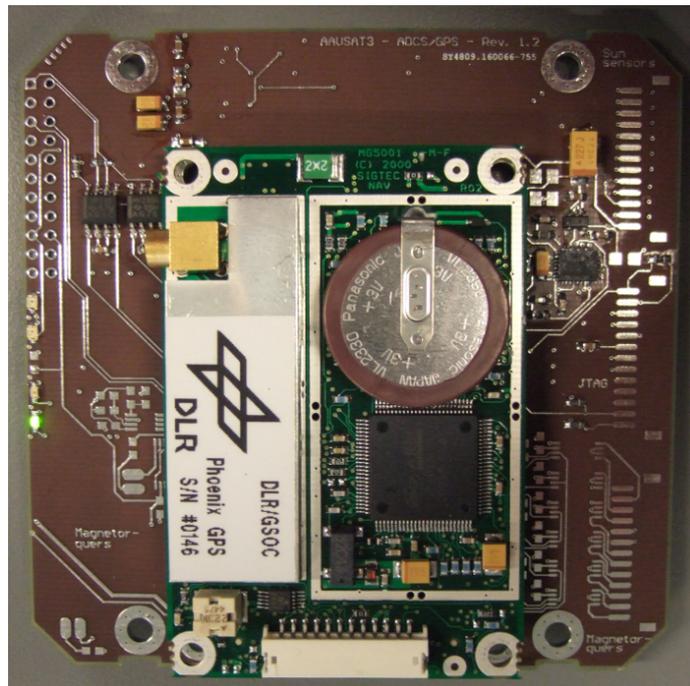
# Appendix I

## ADCS Hardware Schematics

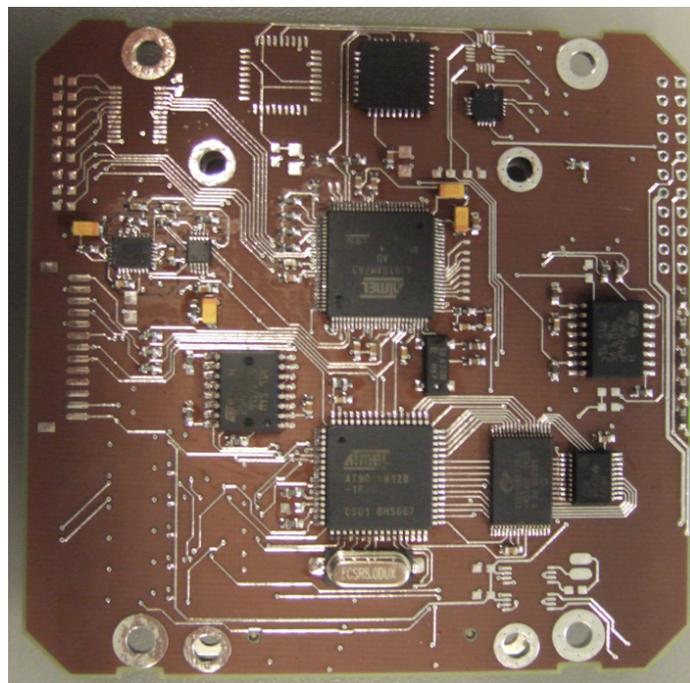
This appendix contains pictures and schematics of the prototype ADCS PCB. The PCB schematics and PCB layout are drawn with Altium Design Winter 09 and the documents are placed on the appended CD.



**Figure I.1:** Top view of ADCS PCB (not completely soldered yet).

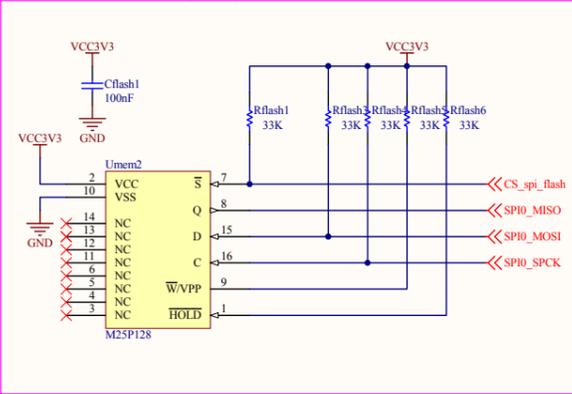


*Figure I.2: Top view of ADCS PCB (not completely soldered yet) and GPS module (green PCB).*

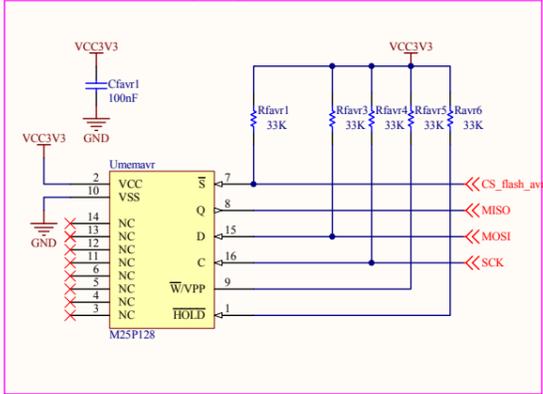


*Figure I.3: Bottom view of ADCS PCB (not completely soldered yet).*

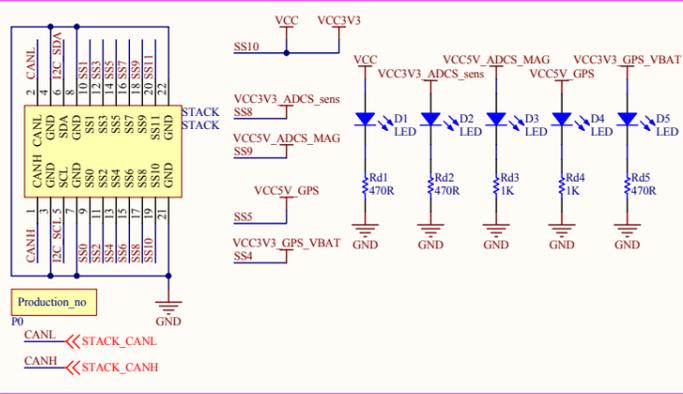
16 MB SPI FLASH (NOR)



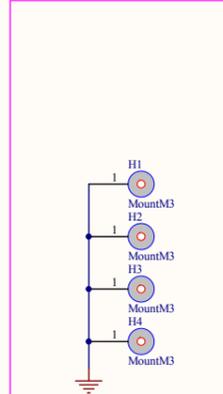
16 MB SPI FLASH (NOR)



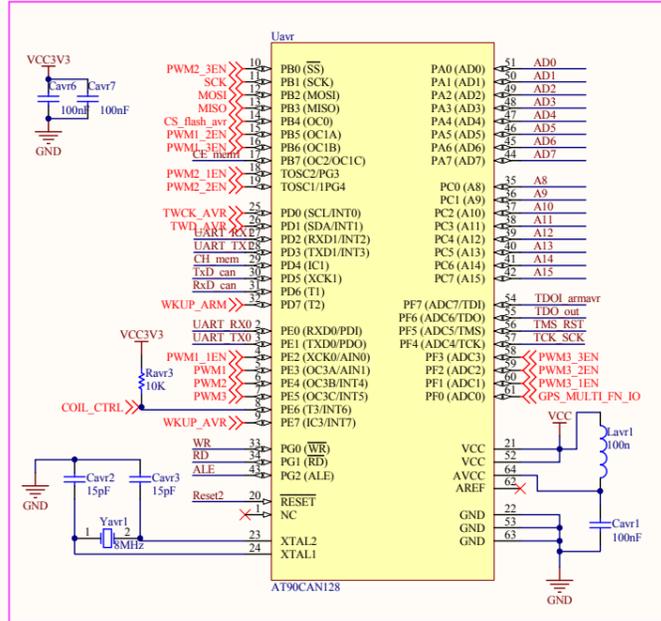
Stack connector



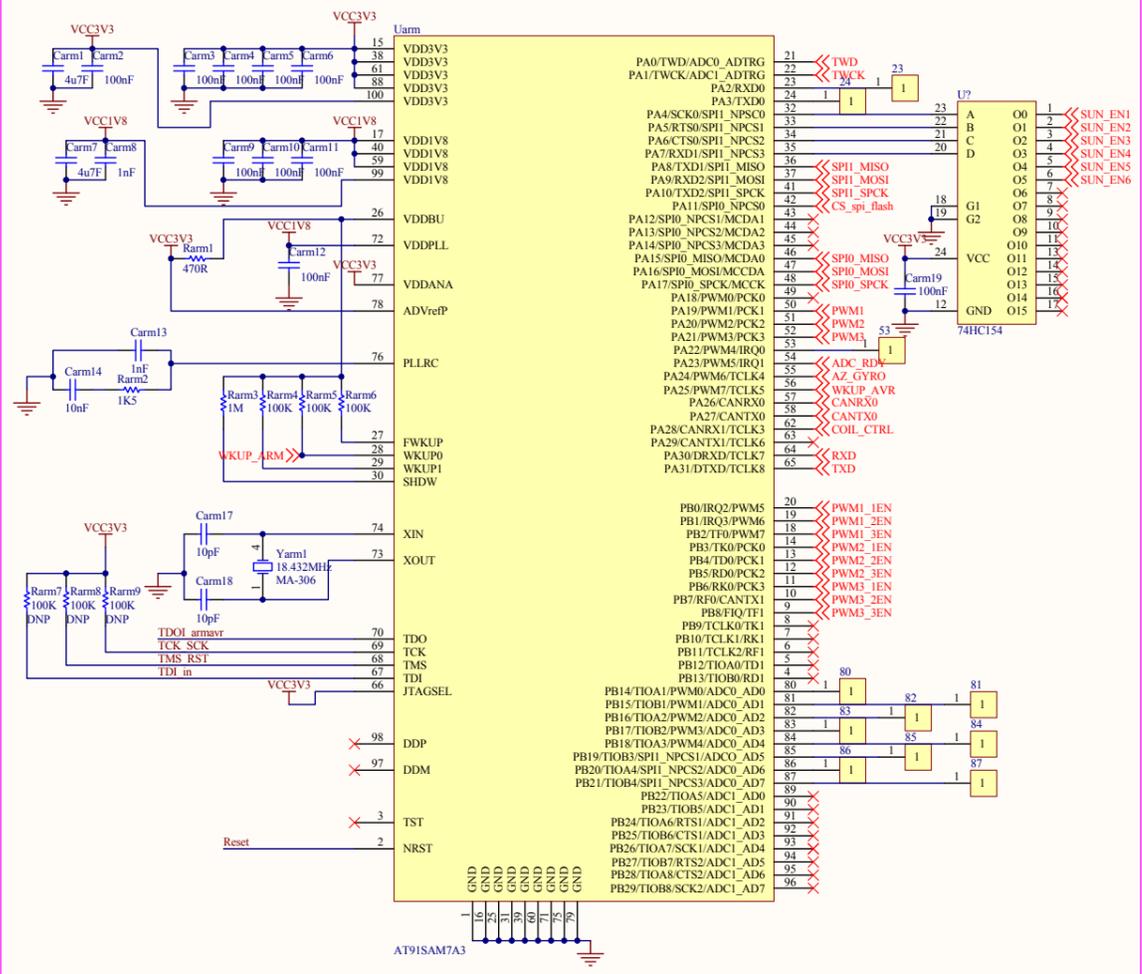
Mount Stack



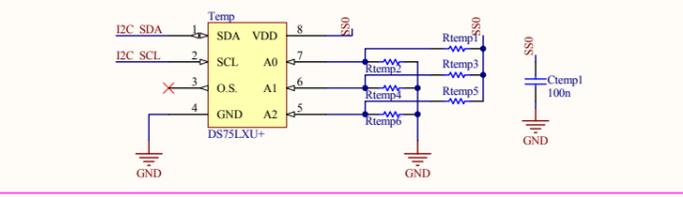
AVR8 + clock



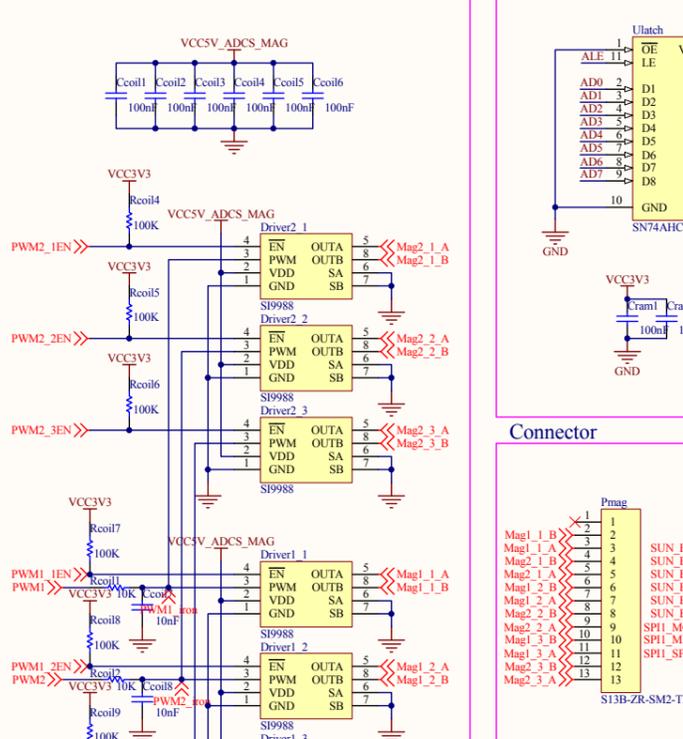
ARM7 + stuff



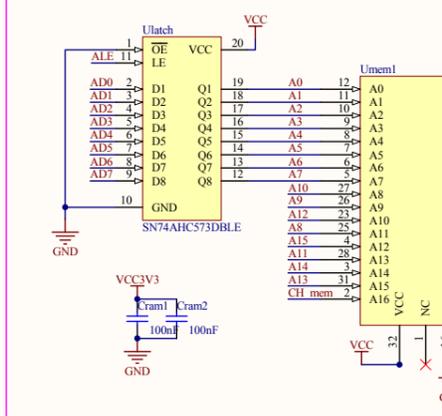
Temperature sensor



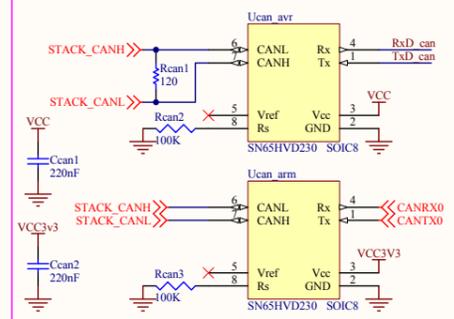
Magnetorquer drivers (redundant)



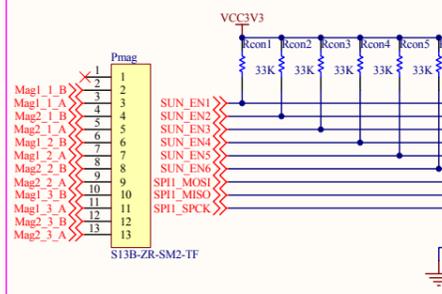
External ram



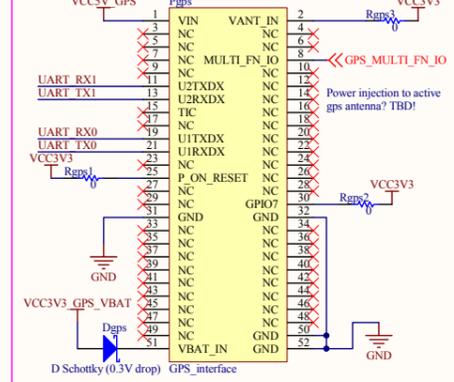
CAN transceivers



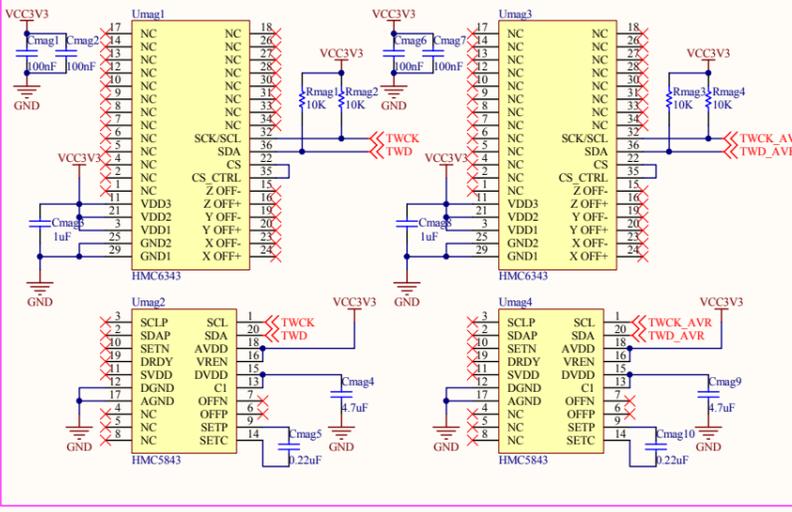
Connector



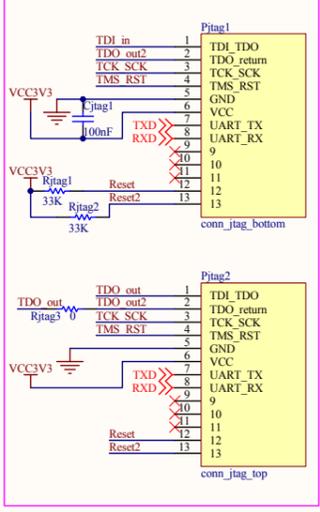
GPS connector



Magnetometers



JTAG



Gyros

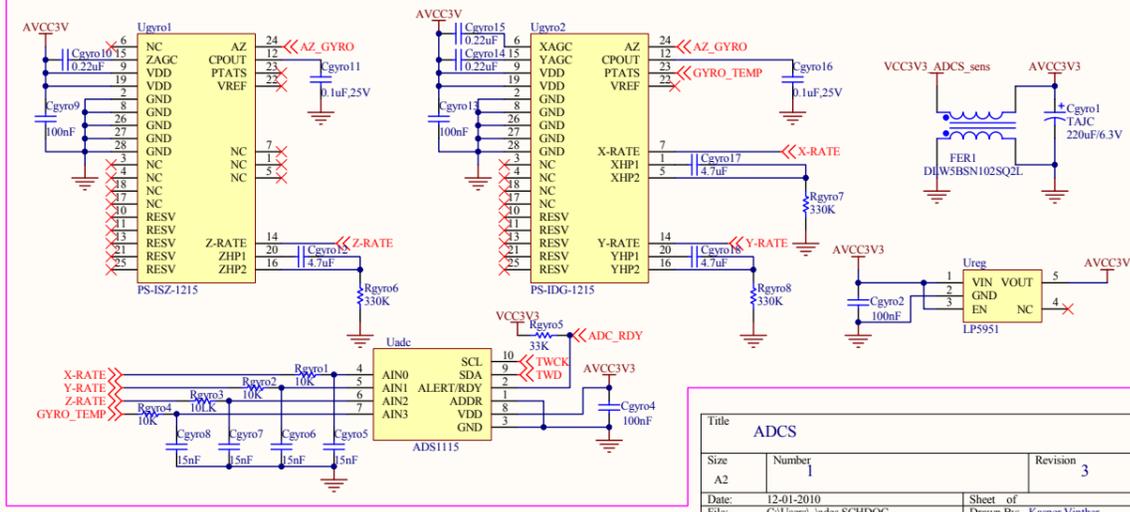


Table with columns: Title (ADCS), Number (1), Revision (3), Date (12-01-2010), Sheet of (1), File (C:\Users\kasper\Documents\ADCS\SCHDOC), and Drawn By (Kasper Vinther).



# Appendix J

## CD Content

The content of the top folders on the appended CD is explained in the following:

- **datasheets:** This folder contains datasheets for the components used on the ADCS hardware prototype.
- **design:** Files used for design of magnetorquers, calculation of worst case disturbance torques and satellite inertia are placed in this folder.
- **lib:** This folder contains the AAUSAT3 Matlab Simulink library and all associated components.
- **misc:** This folder contains a mex compile file and figure printing files.
- **models:** This folder contains various Simulink models used during development.
- **test:** This folder contains Simulink models and Matlab-files used during test of various modules of the ADCS.
- **toolboxes:** Yalmip and SeDuMi are used as solvers and are therefore provided.
- **igrf2005.d:** IGRF coefficients valid from 2005-2010.
- **readMe.txt:** This is an installation guide which also explains how to use the satellite simulation environment.
- **startup.m:** This file must be run every time the satellite simulation environment is started up.
- **thesis\_10gr1035.pdf:** The thesis in pdf format.



# Bibliography

- [1] J. R. Wertz, *Spacecraft Attitude Determination and Control*. Kluwer Academic Publishers, 1st ed., 1994.
- [2] Group 05gr833, "Attitude Determination System for AAUSAT-II," tech. rep., Department of Control Engineering - Aalborg University, 2005.
- [3] B. Wie, *Space Vehicle Dynamics and Control*. American Institute of Aeronautics and Astronautics, Inc., 1st ed., 1998.
- [4] T. Bak, *Spacecraft Attitude Determination - A Magnetometer Approach*. PhD thesis, Department of Control Engineering - Aalborg University, 1999. 2nd ed., pp. 9-69.
- [5] C. Arbinger and W. Enderle, "Spacecraft attitude determination using a combination of gps attitude sensor and star sensor measurements," *ION GPS 2000*, pp. 2634–2642, September 2000.
- [6] Honeywell Inc., "Hmc6343 - 3-axis compass with algorithms." <http://www.ssec.honeywell.com/magnetic/datasheets/HMC6343.pdf>.
- [7] SILONEX Inc., "Slcd-61n8 - solderable planar photodiode." <http://www1.silonex.com/datasheets/specs/images/pdf/104118.pdf>.
- [8] InvenSense Inc., "Idg-1215 - dual-axis gyroscope." <http://invensense.com/mems/gyro/documents/PS-IDG-1215B-00-02.pdf>.
- [9] InvenSense Inc., "Isz-1215 - single-axis z-gyro." <http://invensense.com/mems/gyro/documents/PS-ISZ-1215B-00-03.pdf>.
- [10] Sintex, "Sintered neodymium magnets (ndfeb)." <http://www.sintex.dk/>, 2009.
- [11] Danish Meteorological Institute, "The Ørsted satellite." URL: <http://dmiweb.dmi.dk/fsweb/projects/oersted/mission/mission.html> [cited 11 April 2010].
- [12] Aalborg University, "Aau cubesat - aalborg universitys studentsatellite." URL: <http://www.cubesat.auc.dk/> [cited 11 April 2010].

## BIBLIOGRAPHY

---

- [13] AAUSAT-II team, "Aausat-ii." <http://aausatii.space.aau.dk/>, 2009.
- [14] ESA, "Esa - sseti express." URL: [http://www.esa.int/esaMI/sseti\\_express/SEMDUENFGLE\\_0.html](http://www.esa.int/esaMI/sseti_express/SEMDUENFGLE_0.html) [cited 11 April 2010].
- [15] Danish Maritime Safety Administration, "Mission and vision." URL: [http://frv.dk/EN/ABOUT\\_US/PROFILE/Pages/Mission\\_vision.aspx](http://frv.dk/EN/ABOUT_US/PROFILE/Pages/Mission_vision.aspx) [cited 11 April 2010].
- [16] Exactearth, "Global vessel monitoring with exactais. precise, secure, reliable." URL: <http://www.exactearth.com/> [cited 11 April 2010].
- [17] France Développement Conseil, "Fdc news." URL: <http://www.fdc.fr/en/news.php> [cited 11 April 2010].
- [18] LuxSpace Sarl., "Pasta mara." URL: <http://www.pasta-mare.eu/> [cited 11 April 2010].
- [19] ESA, "Esa space engineering." URL: [http://www.esa.int/SPECIALS/Space\\_Engineering/SEMWN5W0EZF\\_0.html](http://www.esa.int/SPECIALS/Space_Engineering/SEMWN5W0EZF_0.html) [cited 11 April 2010].
- [20] Orbcomm, "What is satellite ais." URL: <http://www.orbcomm.com/ais/ais.htm> [cited 11 April 2010].
- [21] Spacequest, "Technical exchange on ais via satellite." URL: <http://www.spacequest.com/Articles/SpaceQuest-TEXAS-III-Presentation.pdf> [cited 11 April 2010].
- [22] U.S. Naval Research Laboratory, "Gladis: Global ais and data-x international satellite constellation." URL: <http://code8100.nrl.navy.mil/programs/images/GLADIS-INTL-web-pub.pdf> [cited 11 April 2010].
- [23] Jotron, "Tron ais-sart - approved by bsh." URL: <http://www.jotron.com/Default.asp?Cat=705> [cited 11 April 2010].
- [24] AAUSAT3 team, "Aausat3." <http://www.aausat3.space.aau.dk>, 2010.
- [25] Swedish Space Corporation, "Bexus." URL: <http://www.ssc.se/?id=7114> [cited 11 April 2010].
- [26] J. Puig-Suari, C. Turner, and R. J. Twiggs, "CubeSat: The Development and Launch Support Infrastructure for Eighteen Different Satellite Customers on One Launch," *In Proceedings of the 15th Annual AIAA/USU Conference on Small Satellites*, 2001. Logan, UT.
- [27] NORAD, "About norad." <http://www.norad.mil/about/index.html>, 2009.
- [28] J. R. Wertz and W. J. Larson, *Spacecraft Mission Analysis and Design*. Microcosm Press and Kluwer Academic Publishers, 3rd ed., 2005.

- [29] Aalborg University, “Studieordning for uddannelserne: Civilingeniør i informatik, civilingeniør i kommunikationsnetværk, civilingeniør i proceskontrol, civilingeniør i signalbehandling, civilingeniør i telekommunikation 1.-4. semester.” [http://www.esn.aau.dk/fileadmin/esn/Studieordning/Cand\\_S0\\_ed\\_aalborg\\_sep08.pdf](http://www.esn.aau.dk/fileadmin/esn/Studieordning/Cand_S0_ed_aalborg_sep08.pdf), 2008.
- [30] AAUSAT3 team, “Aausat3 documentation page.” <https://calvin.space.aau.dk/aausat3/>, 2010.
- [31] L. Alminde, “Semi-active attitude control and off-line attitude determination for the sseti-express student micro-satellite,” tech. rep., Department of Control Engineering - Aalborg University, 2005.
- [32] T. Kelso, “Frequently asked questions: Two-line element set format.” <http://celestrak.com/columns/v04n03/>, 2006.
- [33] F. R. Hoots and R. L. Roehrich, “Models for propagation of norad element sets.” SPACETRACK REPORT NO. 3, 1988.
- [34] T. Kelso, “Norad two-line element sets current data.” <http://celestrak.com/NORAD/elements/>, 2009.
- [35] T. Kelso, “Tracking the sun and the moon.” <http://celestrak.com/columns/v03n03/>, 2006.
- [36] R. A. Serway, *Physics for Scientists and Engineers with Modern Physics*. Thomson Brooks/Cole, 6th ed., 2004.
- [37] Group 06gr1032, “Robust disturbance rejecting attitude control,” tech. rep., Department of Control Engineering - Aalborg University, 2006.
- [38] J. B. Kuipers, *Quaternions and Rotation Sequences*. Princeton University Press, 1st ed., 2002.
- [39] P. C. Hughes, *Spacecraft Attitude Dynamics*. Dover Publications, Inc., 1st ed., 2004.
- [40] J. J. Butler, B. C. Johnson, J. P. Rice, E. L. Shirley, and R. A. Barnes, “Sources of Differences in On-Orbital Total Solar Irradiance Measurements and Description of a Proposed Laboratory Intercomparison,” *Journal of Research of the National Institute of Standards and Technology*, vol. 113, pp. 187–203, July-August 2008.
- [41] A. Foletti and P. Kaewkerd, “Adcs report,” tech. rep., École Polytechnique Fédérale de Lausanne, 2006.
- [42] DTUsat team, “Dtusat.” <http://www.dtusat.dtu.dk/>, 2009.
- [43] J. H. Hales and M. Pedersen, “Two-Axis MOEMS Sun Sensor for Pico Satellites,” *16th Annual AIAA/USU Conference on Small Satellites*, 2002.
- [44] T. Graversen, M. K. Frederiksen, and S. V. Vedstesen, “Attitude control system for aau cubesat,” tech. rep., Department of Control Engineering - Aalborg University, 2002.

## BIBLIOGRAPHY

---

- [45] Group 05gr834, "Attitude control system for aausat-ii," tech. rep., Department of Control Engineering - Aalborg University, 2005.
- [46] P. Moonjelly and K. Vemulapalli, "Hybrid Switched System Approach to Nano-Satellite Attitude Control Using Magnetorquers Alone," *Unknown*, 2008. Stanford, CA.
- [47] Vishay Siliconex, "Si9988 - buffered h-bridge." <http://www.vishay.com/docs/71326/71326.pdf>.
- [48] MIT, "The hostile space environment." <http://cyclo.mit.edu/~bmonreal/environment.html>, 2002.
- [49] The Ørsted Science Data Centre, "The Ørsted satellite." <http://web.dmi.dk/projects/oersted/SDC/oersted.html>, 2010.
- [50] DLR, "Phoenix - spaceborne gps receiver." [http://www.weblab.dlr.de/rbrt/GpsNav/Phoenix/Phoenix\\_DataSheet\\_v11.pdf](http://www.weblab.dlr.de/rbrt/GpsNav/Phoenix/Phoenix_DataSheet_v11.pdf), 2007.
- [51] Group 04gr830a, "Attitude control system for aausat-ii," tech. rep., Department of Control Engineering - Aalborg University, 2004.
- [52] Group 04gr830b, "Attitude determination system for aausat-ii," tech. rep., Department of Control Engineering - Aalborg University, 2004.
- [53] D. D. V. Bhanderi, "Earth albedo." <http://bhanderi.dk/research/albedo/>, 2008.
- [54] Analytical Graphics, Inc., "Orbit propagators for satellites." [https://www.stk.com/resources/help/stk613/helpSystem/stk/vehSat\\_orbitProp\\_choose.htm](https://www.stk.com/resources/help/stk613/helpSystem/stk/vehSat_orbitProp_choose.htm), 2009.
- [55] Space alliance, "Orbit propagation." <http://www.spacealliance.ro/articles/view.aspx?id=200903060423>, 2009.
- [56] Group 07gr833, "Three-axis satellite attitude determination using sun and earth albedo vectors," tech. rep., Department of Control Engineering - Aalborg University, 2007.
- [57] R. McPeters, "Total ozone mapping spectrometer." <http://jwocky.gsfc.nasa.gov/eptoms/ep.html>, 2009.
- [58] National Geophysical Data Center, "Geomagnetism." <http://www.ngdc.noaa.gov/geomag/>, 2009.
- [59] IAGA, "10th generation igrf - released 2005." <http://www.ngdc.noaa.gov/IAGA/vmod/home.html>, 2008.
- [60] Kimmo Korhonen, "The igrf-applet." <http://www.geo.fmi.fi/MAGN/igrf/applet.html>, 2000.

- 
- [61] J. L. Crassidis, F. L. Markley, and Y. Cheng, "A Survey of Nonlinear Attitude Estimation Methods," *Journal of Guidance, Control and Dynamics*, vol. 30, no. 1, 2007.
- [62] M. S. Grewal and A. P. Andrews, *Kalman Filtering - Theory and Practice Using MATLAB*. John Wiley and Sons, Inc., 3rd ed., 2008.
- [63] J. L. Crassidis and F. L. Markley, "Unscented Filtering for Spacecraft Attitude Estimation," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 26, pp. 536–542, August 2003.
- [64] Y.-J. Cheon, "Unscented Filtering in a Unit Quaternion Space for Spacecraft Attitude Estimation," *International Conference on Control, Automation and Systems*, June 2005.
- [65] S. Pourtakdoust and H. Ghanbarpour, "An Adaptive Unscented Kalman Filter for Quaternion-based Orientation estimation in Low-cost AHRS," *Aircraft Engineering and Aerospace Technology: An international Journal*, vol. 79, no. 5, pp. 485–493, 2007.
- [66] E. Kraft, "A Quaternion-based Unscented Kalman Filter for Orientation Tracking," *Proceedings of the 6th International Conference on Information Fusion*, pp. 47–54, July 2003. Cairns, Queensland.
- [67] M. C. VanDyke, J. L. Schwartz, and C. D. Hall, "Unscented Kalman Filtering for Spacecraft Attitude State and Parameter Estimation," *Proceedings of the 14th AAS/AIAA Space Flight Mechanics Conference*, vol. 115, February 2004. Maui, Hawaii.
- [68] S. Haykin, *Kalman Filtering and Neural Networks*. John Wiley and Sons, Inc., 1st ed., 2001. pp. 221-267.
- [69] D. Simon, *Optimal State Estimation - Kalman,  $H_\infty$  and Nonlinear Approaches*. John Wiley and Sons, Inc., 1st ed., 2006.
- [70] G. Wahba, "Problem 65-1: A Least Squares Estimate of Satellite Attitude," *Siam Review*, vol. 7, no. 3, p. 409, 1965.
- [71] F. L. Markley and D. Mortari, "Quaternion Attitude Estimation Using Vector Observations," *The Journal of Astronautical Sciences*, vol. 48, pp. 359–380, April-September 2000.
- [72] M. D. Shuster, "The Generalized Wahba Problem," *The Journal of Astronautical Sciences*, vol. 54, pp. 245–259, April-June 2006.
- [73] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge University Press, 2nd ed., 1995.
- [74] J. J. LaViola, "A Comparison of Unscented and Extended Kalman Filtering for Estimating Quaternion Motion," *Proceedings of the American Control Conference*, pp. 2435–2440, June 2003. Denver, CO.

## BIBLIOGRAPHY

---

- [75] The CubeSat Program, Cal Poly SLO, "Cubesat design specification rev.12." URL: [http://www.cubesat.org/images/developers/cds\\_rev12.pdf](http://www.cubesat.org/images/developers/cds_rev12.pdf) [cited 7 Marts 2010].
- [76] R. Wisniewski, *Spacecraft Attitude Control Using Only Electromagnetic Actuation*. PhD thesis, Department of Control Engineering - Aalborg University, 1996.
- [77] J. Jung, N. Kuzuya, and J. Alvarez, "The Design of the OPAL Attitude Control System," *Proceedings of the 10th Annual AIAA/USU Conference on Small Satellites*, 1996. Stanford, CA.
- [78] T. W. Flatley, W. Morgenstern, A. Reth, and F. Bauer, "A B-Dot Acquisition Controller for the RADARSAT Spacecraft," *Flight Mechanics Symposium*, pp. 79–89, 1997. Stanford, CA.
- [79] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 3rd ed., 2002.
- [80] P. Menold, R. Pearson, and F. Allgöwer, "Online Outlier Detection and Removal," *Proceedings of the 7th Mediterranean Conference on Control and Automation (MED99)*, pp. 1110–1133, June 1999. Haifa, Israel.
- [81] E. Silani and M. Lovera, "Magnetic Spacecraft Attitude Control: A Survey and Some New Results," *Control Engineering Practice*, vol. 13, pp. 357–371, December 2003. Milano, Italy.
- [82] A. Varga and M. Lovera, "Optimal Discrete-time Magnetic Attitude Control of Satellites," *IFAC World Congress*, 2005. Prague, Czech Republic.
- [83] E. Silani and M. Lovera, "Bias Momentum Satellite Magnetic Attitude Control Based On Genetic Algorithms," *UKACC*, 2006. Harbin, China.
- [84] M. Psiaki, "Spacecraft Attitude Control Using Magnetic Actuators," *AIAA Journal of Guidance, Control and Dynamics*, vol. 24, no. 2, pp. 386–394, 2001. New York, USA.
- [85] M. Loveraa and A. Astolfi, "Spacecraft Attitude Control Using Magnetic Actuators," *Automatica*, vol. 40, no. 8, pp. 1405–1414, 2004. Milano, Italy.
- [86] . Hegrenæs, J. T. Gravdahl, and P. Tøndel, "Spacecraft Attitude Control using Explicit Model Predictive Control," *Automatica*, vol. 41, no. 12, pp. 2107–2114, 2005. Trondheim, Norway.
- [87] Rolf Findeisen and Frank Allgöwer, "An Introduction to Nonlinear Model Predictive Control," tech. rep., Institute for Systems Theory in Engineering - University of Stuttgart, 2002.
- [88] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control - Theory and Design*. Nob Hill Publishing, 1st ed., 2009.
- [89] J. M. Maciejowski, *Predictive Control with Constraints*. Pearson Prentice Hall, 1st ed., 2002.

- [90] J. Löfberg, “Yalmip wiki.” <http://control.ee.ethz.ch/~joloef/wiki/pmwiki.php?n=Main.What>, 2009.
- [91] I. The MathWorks, “The mathworks - accelerating the place of engineering and science.” <http://www.mathworks.com/>, 2010.
- [92] D. C. Lay, *Linear Algebra and Its Applications*. Addison Wesley, 3rd update ed., 2006.
- [93] J. D. P. G. F. Franklin and A. Emami-naeini, *Feedback Control of Dynamic Systems*. Pearson Prentice Hall, 5th ed., 2006.
- [94] Group 08gr724, “Software framework for reconfigurable distributed system on ausat3,” tech. rep., Department of Control Engineering - Aalborg University, 2008.
- [95] ESA, “Outgassing database.” [http://esmat.esa.int/Services/outgassing\\_data/outgassing\\_data.html](http://esmat.esa.int/Services/outgassing_data/outgassing_data.html).

# Index

- Actuators, 41
  - Gravity boom, 42
  - Magnetorquer, 42, 48
  - Momentum wheels, 41
  - Permanent magnet, 42, 43
  - Spin stabilization, 41
  - Thrusters, 41
- Angular momentum, 143, 144
- Attitude estimation, 79
  - Deterministic point by point solution, 79
  - Extended Kalman Filter, 81, 82, 161
  - Kalman filter, 80
  - Recursive stochastic filters, 79
  - Sampling method, 81
  - SVD-method, 81, 90
  - Unscented Kalman Filter, 81, 84, 92
  - Wahba's problem, 81
- Attitude parameterizations, 26
  - Axis-azimuth, 26
  - Direction cosine matrix, 26
  - Euler angles, 26
  - Rodriguez, 26
- Automatic Identification System (AIS), 2, 8
- B-dot control, 104
- B-dot filter, 106
- Budgets, 176
  - Data storage, 178
  - Mass, 178
  - Power, 176
  - Size, 177
- Center of Mass (CoM), 137
- Cholesky factorization, 85, 93
- Connector
  - GPS, 175
  - JTAG daisy chain, 176
  - Magnetorquer, 175
  - Stack, 174
  - Sun sensor, 176
- Control law, 104
- Covariance matrix
  - Error, 83, 85
  - Measurement noise, 83, 87
  - Process noise, 83, 86
- Disturbances, 28, 145
  - Atmospheric drag, 30, 148, 159
  - Gravitational field, 28, 147, 159
  - Magnetic residual, 33, 149, 159
  - Solar radiation, 31, 148, 159
- Dynamic equations, 143, 168
- Economy, 178
- Equation of motion, 27
- Equatorial plane, 19, 22
- Error quaternion, 90
- Field of View (FoV), 152
- Geometric center of Mass (GoM), 137
- Inclination, 19
- Inertia, 137, 144
- Interface Control Document (ICD), 171
- International Geomagnetic Reference Field (IGRF), 71
- Jacobian matrix, 161, 165
- Julian Date, 19
- Keplerian orbits, 17

- Kinematic equations, 141, 165
- Krasovskii-LaSalle invariance theorem, 106
- Lyapunov function, 105
- Magnetic actuation, 104, 113, 121
- Matlab, 61
- Mechanical properties, 140
- Mechanical structure, 138
- Mission, 2
- Model predictive control, 115
  - Constraints, 117
  - Control horizon, 115
  - Cost function, 116
  - Input cost matrix, 121
  - input cost matrix, 116
  - Nonlinear, 124
  - Nonlinear optimization problem, 125
  - Objective function, 116
  - Prediction horizon, 115
  - Re-linearization, 118
  - Stability analysis, 117
  - State cost matrix, 116, 121
  - Terminal constraint, 117
  - Terminal cost, 117
  - Terminal cost matrix, 117, 121
- Models, 61
  - Earth Albedo, 67, 153
  - Eclipse, 66
  - Ephemeris, 65, 151
  - Gyroscope, 74
  - Magnetic field, 70, 154
  - Magnetometer, 72
  - Magnetorquer, 76
  - Permanent magnet, 75
  - Sun sensor, 73
- NORAD, 19
- Orbit propagators, 19, 63, 151
  - SDP4, 19, 64
  - SDP8, 21, 64
  - SGP4, 19, 64, 151
  - SGP8, 21, 64
- Orbital elements, 17
- Orbital perturbation, 21
- Orbital plane, 19
- Orbitron, 151
- Outlier detection, 109
- Perigee, 18
- Printed Circuit Board (PCB), 183
- Prototype hardware, 54, 181
- Quadratic programming problem, 120
- Quaternions, 25, 133, 141
  - Error, 136
  - Inverse, 135
  - Product, 134
- Receding horizon control, 115
- Reference frames, 22
  - Controller (CRF), 24
  - Earth centered Earth fixed (ECEF), 24
  - Earth centered inertial (ECI), 22
  - Orbit (ORF), 24
  - Satellite body (SBRF), 25
- Requirements, 11
  - Functional, 12
  - Performance, 13
- Rotation matrix, 25
- Runge Kutta (RK), 93
- Sensors, 35
  - GPS antenna array, 37
  - Gyroscopes, 36
  - Horizon sensor, 37
  - Magnetometer, 37
  - Star tracker, 37
  - Sun sensor, 37
- Sigma points, 81, 84
- Simulation environment, 61
- Simulink, 61
- Space environment, 54
- Special orthogonal group, 25
- Spheres of Influence (SoI), 21
- Stability analysis, 104
- State transition diagram, 59
- Subsystems, 8
  - ADCS1, 9, 55, 59, 171
  - ADCS2, 9, 55, 59, 171
  - AIS1, 8

## INDEX

---

AIS2, 8  
CAM, 9  
EPS, 8, 172  
FP, 8  
FPGA, 9  
GPS, 9, 172  
Ground Station/MCC, 8  
LOG, 8, 171  
UHF, 8

Telecommands, 173  
Torque, 143, 144  
Total Ozone Mapping Spectrometer (TOMS),  
69  
Total Solar Irradiance (TSI), 31  
Truth model, 61  
Two Line Element, 19

Unscented TransForm (UTF), 84

Vernal equinox, 19, 22, 24