

CubeSat ADCS

Developing an ADCS for AAUSAT6

Albert Laursen, Jacob Nyvang, Jens Moeslund,
Martin Boysen, Thorbjørn Friis-Christensen

Electronics and IT, EIT-B614, 2023-05

Bachelor Project



Copyright © Aalborg University 2022

This document was generated using L^AT_EX and the aau-report by Jesper Kjær Nielsen, and is available at <https://github.com/jkjaer/aauLatexTemplates>. Figures were generated using python with the third-party libraries matplotlib, schemdraw and scipy.



Electronics and IT
Aalborg Universitet
<http://www.aau.dk>

AALBORG UNIVERSITY

STUDENT REPORT

Title:

ADCS for AAUSAT6

Theme:

Control and Automation

Project Period:

Spring Semester 2023

Project Group:

EIT-614

Participant(s):

Albert Werner Laursen
Jacob Nyvang Jensen
Jens Moeslund Larsen
Martin Lund Boysen
Thorbjørn Friis-Christensen

Supervisor(s):

Rafal Wisniewski

Copies: 1**Page Numbers:** 95**Date of Completion:**

25. Maj 2023

Abstract:

The purpose of this project is to design an Attitude Control System (ACS) for the AAUSAT6 satellite. The satellite is a 1U CubeSat. The ACS is designed to be able to point the satellite's camera payload towards a point of interest on Earth using reaction wheels. The ACS is designed using a linear state space model of the satellite, which is derived from the rotational mechanics, kinematic model and dynamic model of the satellite. The state space control system is designed using LQR and the motor controller is designed using a PI controller. The final control system is simulated using the AAU-SAT Simulation Library in Simulink, and it is concluded that the designed control system passes all the requirements.

Contents

Preface	vi
1 Introduction	1
2 Preanalysis	2
2.1 Orbit description	2
2.2 Reference Frames	4
2.2.1 Earth Centered Inertial	5
2.2.2 Earth Centered Earth Fixed	5
2.2.3 Orbit Reference Frame	6
2.2.4 Satellite Reference Frame	6
2.2.5 Control Reference Frame	7
2.2.6 Target Reference Frame	8
3 Project delimitation	9
4 Technical Analysis	10
4.1 Satellite Control	10
4.1.1 Magnetorquers	10
4.1.2 Thrusters	10
4.1.3 Reaction Wheels	11
4.2 Disturbance torques	11
4.2.1 Solar radiation	11
4.2.2 Aerodynamic drag	13
4.2.3 Magnetic disturbance	14
4.2.4 Gravity gradient	17
4.2.5 Summary of disturbances	18
4.3 Pointing precision	19
4.3.1 Step from Nadir to pointing	21
4.3.2 Reaction wheel control time	25
4.4 Attitude Parametrization	26
4.4.1 Euler angles	26
4.4.2 Euler axis-angle	27
4.4.3 Quaternions	28
4.4.4 Summary	29
5 Specification	31
6 Test Enviroment	32
7 Test specification	34

7.1	Nadir mode test	34
7.1.1	Test procedure	34
7.1.2	Success criteria	34
7.2	Pointing mode test	34
7.2.1	Test procedure	35
7.2.2	Success criteria	35
7.3	Combined test	35
7.3.1	Test procedure	35
7.3.2	Success criteria	35
8	System Development	36
8.1	Satellite Modes	36
8.1.1	Nadir mode	36
8.1.2	Target mode	37
8.2	Rotational Mechanics	40
8.3	Kinematic model	42
8.4	Dynamic Model	44
8.5	Reaction wheels using DC motors	45
8.5.1	Reaction wheel configuration	48
8.5.2	Dynamic model of the the DC motors	52
8.6	Linear State Space Model	54
8.6.1	State Space Model	54
8.6.2	DC Motor Controller	55
8.6.3	State Vector	61
8.6.4	Derived State Model	62
8.6.5	Solving Linear State Space Model	66
8.6.6	Operating Points	67
8.6.7	Control of the Linearised State Space Model	68
8.6.8	LQR	70
8.6.9	Discretization	71
8.6.10	Implementation	73
9	Simulation Implementation	76
9.0.1	LQR controller	77
9.0.2	Tetrahedron Reaction Wheels	78
9.0.3	Satellite System Model	78
9.0.4	Tetrahedron Reaction Wheels	79
9.0.5	Target reference generator	80
9.0.6	Reaction wheel control	81
9.0.7	Animation environment	81
10	Acceptance Test	83
10.1	Nadir mode acceptance test	83
10.1.1	Results	83
10.1.2	Compliance of requirement	84
10.2	Pointing Mode acceptance test	84
10.2.1	Results	84
10.2.2	Compliance of requirement	85
10.3	Durability acceptance test	86
10.3.1	Results	86

10.3.2 Compliance of requirement	87
10.4 Overview of acceptance test results	87
11 Discussion	88
11.1 Steady state error	88
11.2 Momentum dumping	90
11.3 Physical test	90
12 Conclusion	92
Bibliography	93
A Measuringjournals for the requirement tests	96
A.1 Nadir mode Tests	96
A.1.1 Test procedure	96
A.1.2 Success criteria	96
A.1.3 Results	96
A.1.4 Compliance of requirement	97
A.2 Pointing mode Tests	97
A.2.1 Test procedure	97
A.2.2 Success criteria	97
A.2.3 Results	98
A.2.4 Compliance of requirement	99
A.3 Pointing mode Tests	99
A.3.1 Test procedure	100
A.3.2 Success criteria	100
A.3.3 Results	100
A.3.4 Compliance of requirement	100
B Measuringjournals for other tests	102
B.1 Atmospheric density	102

Preface

This thesis is conducted by group 614 as part of the Bachelor of Science in Electronics and IT at Aalborg University. The project is conducted in the spring semester of 2023, and is the final project of the bachelor degree.

The project is supervised by Professor, Ph.D. Rafal Wisniewski.

Reading Guide

The report is structured in chapters, sections, subsections and subsubsections. The chapters, sections and subsections are numbered with Arabic numerals, while the subsubsections will appear without numbering.

Figures, tables, equations and listings are numbered with Arabic numerals, where the first number is the chapter number. For example, Figure 3.2 is the second figure in chapter 3. Appendices are grouped by capital letters, and numbered with Arabic numerals.

The reference method used in this report is the IEEE style. The references are numbered with Arabic numerals, and are referred to in the text with square brackets. They are listed in the bibliography at the end of the report, and are sorted in the order they appear in the report.

Aalborg University, 25. May 2023

Notation

The following mathematical notation is used in this report.

Concept	Notation	Example
Scalar	Italic	s, S
Vector	Italic boldface	v, V
Matrix	Uppercase boldface underlined	$\underline{\mathbf{A}}$
Quaternion	Lowercase boldface	\mathbf{q}
Operating point	Bar	$\bar{\omega}, \bar{\mathbf{q}}$
Small signal	Tilde	$\tilde{\omega}, \tilde{\mathbf{q}}$
Unit vector / quaternion	Hat	$\hat{\mathbf{v}}, \hat{\mathbf{q}}$
Identity matrix	Uppercase boldface I	$\underline{\mathbf{I}}_{3x3}, \underline{\mathbf{I}}_{4x4}$
Skew-symmetric matrix	Uppercase boldface S	$\underline{\mathbf{S}}_3, \underline{\mathbf{S}}_4$
Reference frame	Prescript (y irt. x)	${}_x^y \mathbf{q}, {}_x^y \underline{\mathbf{A}}$

Chapter 1

Introduction

24 years ago the CubeSat standard was made by professor Jordi Puig-Suari and Bob Twiggs from California Polytechnic State University and Stanford University respectively. This standard has since been used widely because it made it easier and cheaper to launch microsatellites [1].

This standard has also been used by Aalborg University which has launched the satellites: AAU CubeSat, AAUSAT-II, AAUSAT3, AAUSAT5, and latest AAUSAT4 in 2016. All the CubeSats have been 1U cubesats meaning they have measured roughly 10 cm X 10 cm X 10 cm.

All the previous CubeSats launched by AAU has used an ADCS to determine the satellites' attitude, detumble and point the satellites antenna towards Earth. The payload for the next satellite, AAUSAT6, is a camera. To point the camera at points of interest, tighter requirements for the control system has to be set.

This project will examine the different ways to make a control system with the stricter requirements.

Chapter 2

Preanalysis

This chapter provides an overview of fundamental concepts related to satellite orbits and reference frames.

Specific types of orbits and reference frames are described, and the mathematical tools used to describe them are introduced. Using these tools, the position and attitude of the satellite can be described.

The core concepts discussed in this chapter create the foundation for the development of the attitude control system.

2.1 Orbit description

An orbit is the curved path one object takes around another object in space, such as stars, planets, moons or satellites. Gravity causes the attraction of such objects, with enough momentum an orbit can be formed without the two objects crashing together. Both objects orbit around the same point, that being their combined center of mass. This means that satellites with an insignificant mass compared to Earth, such as a CubeSat, will approximately orbit around Earth's center of mass.

According to Kepler's first law, two objects that interact gravitationally and are permanently associated will have orbits that are ellipses. On these ellipses there are some key points, the apogee being the point at which the satellite is at its farthest from the object it is orbiting and the perigee being the closest point [2].

There are different types of orbits, the most used ones being Geostationary orbit (GEO), Low Earth orbit (LEO), Medium Earth orbit (MEO), Polar orbit (PO) and Transfer orbits. The Geostationary orbit is the orbit where satellites are at such an altitude and speed that they are constantly over the same spot on Earth. These numbers being an altitude of 35.786 km and a speed of about 3 km/s [3].

The low Earth orbit is an orbit at altitudes up to 1000 km and as low as 160 km. Being so close means that their speed must be much higher than the orbit further away, about 7.8 km/s.

Polar orbits are a type of LEO, defined by a path of the satellite that goes above the poles of Earth [3].

The medium Earth orbit is as the name suggests at higher altitudes than LEO's. These being all the altitudes between LEO and GEO [3].

Transfer orbits are a kind of orbit that is used to transfer satellites from one orbit to another. An example of this is the geostationary transfer orbit, which is used to move satellites to the GEO without having the launch vehicle travel all the way with the satellite. This is done by putting the satellite in a very eccentric elliptical orbit, and releasing the satellite at the perigee and letting it travel to the apogee. When at the apogee the satellite actuates some thrusters keeping the satellite at its current the altitude, that being the GEO [4].

The specifications for AAUSAT6 are yet to be determined, therefore the specifications from AAUSAT4 are used since it is assumed that they will have a very similar orbit. Noteworthy numbers are specified in Table 2.1.

Parameter	Value	Unit
Perigee height	397	km
Apogee height	520	km
Mean orbital speed	7.63	km/s
Orbit period	5652	s
Inclination	98.2	°
Semi-major axis	6829.5	km

Table 2.1: Table showing the specifications for the orbit of AAUSAT4 [5].

The semi-major axis being the diameter of the ellipsis at the point where the satellite is the furthest from earths core. As can be seen in the table, the satellite will be in a LEO and more specifically a PO.

These parameters are visualized, but grossly exaggerated, on Figure 2.1.

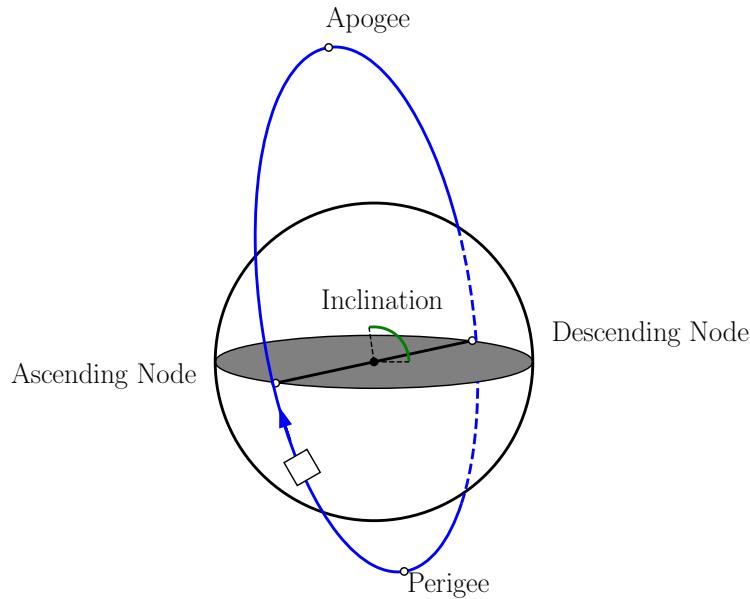


Figure 2.1: Satellite orbit around Earth.

2.2 Reference Frames

When controlling a thing in space it is important to know what the data is in relation to, since in space there is no notion of up or down. In order to specify this the concept of reference frames is used.

The discussed reference frames are outlined in Table 2.2.

Reference frame	Shorthand	Description	Letter
Earth-Centered Inertial	ECI	Main reference frame in this project as it will be the reference frame in which the attitude of the satellite in reference to earth will be defined.	i
Earth-Centered Earth-Fixed	ECEF	Mainly used for calculations regarding the earth's magnetic field.	e
Satellite Reference Frame	SRF	Used to relate the attitude of the satellite to the physical body of the satellite.	s
Control Reference Frame	CRF	Mostly used to simplify calculations for the mathematical modeling of the satellite.	c
Orbit Reference Frame	ORF	Preferred when controlling the satellite in reference to its orbit.	o
Target Reference Frame	TRF	Preferred when controlling the satellite to point at a specific target.	t

Table 2.2: Table of reference frames relevant to this project.

2.2.1 Earth Centered Inertial

This reference frame has its origin point placed in the center of mass of the Earth. The z-axis is placed through the rotational axis of the earth. The x-axis is placed through intersection of earths equatorial plane and its orbital plane around the sun. This means that the x-axis is fixed and does not rotate with the earth, when the vector for this axis points directly towards the sun there is a so-called equinox. This equinox can be used as the principal direction for ECI frames. The y-axis can then be defined as the axis perpendicular to the other two, as shown on Figure 2.2 [6].

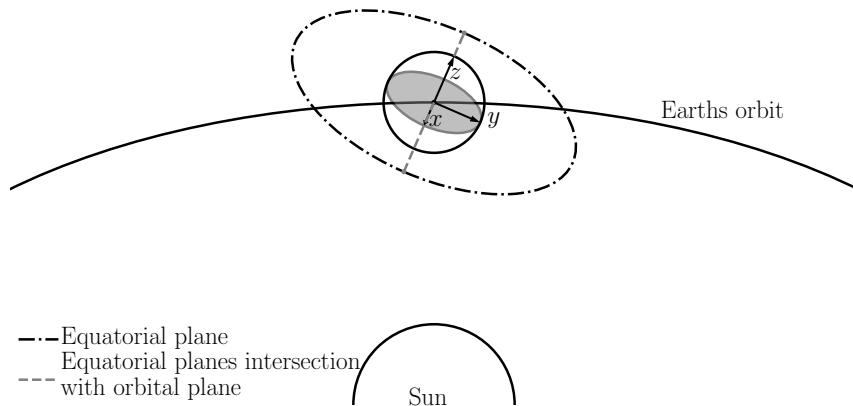


Figure 2.2: ECI reference frame visualized.

This reference frame is useful when calculating the position of a point in space, since the ECI is fixed relative to the stars. This can be useful when calculating the position-vector of a satellite.

2.2.2 Earth Centered Earth Fixed

This reference frame is very similar to the ECI reference frame. It also has its origin point in the center of mass of the Earth, however as the name suggests this reference frame follows Earth and rotates along with it. This means that the z-axis always goes through the North Pole, as the ECI, but the x-axis is always going through one latitude-longitude-coordinate, such as $0^\circ, 0^\circ$, with the y-axis perpendicular to the others [6].

The ECEF is used when calculating the position of a point on Earth, since the ECEF is fixed relative to the Earth, the coordinates of the point does not change.

Notably for both ECI and ECEF, the earth changes obliquity (tilt) and has axial precession (Wobble), this means that the z-axis changes direction relative to fixed positioned stars over time. However, the obliquity only changes 2.4° with 41 000 year cycles, and the precession only does one rotation every 26 000 years, meaning these can be neglected when considering the lifetime of a CubeSat [7].

2.2.3 Orbit Reference Frame

The Orbit Reference frame has its origin point in the center of mass of the satellite. It consists of three axes the first being the roll axis x , the second being pitch y and the last being yaw z . The roll is defined as the direction of the satellites orbit. The yaw axis is pointing towards the center of the Earth. And the pitch being orthogonal to the other two [8]. This can be seen on Figure 2.3.

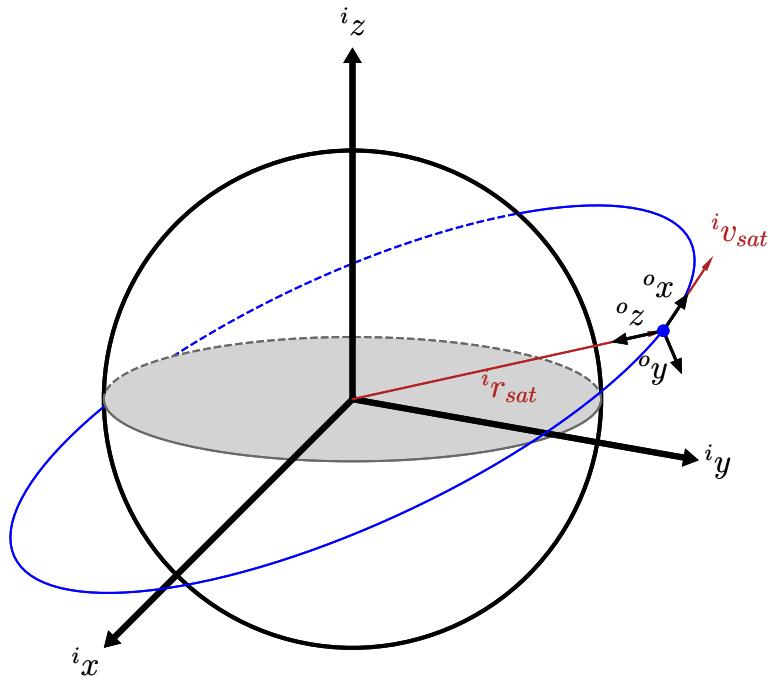


Figure 2.3: Orbit Reference Frame

The orbit reference frame is useful when the satellite has to point its antenna towards the center of the earth, as it orbits. This is also called nadir pointing.

2.2.4 Satellite Reference Frame

The satellite reference frame has its origin point in one of the corners of the frame of the satellite, the axes then goes through the borders of the frame, seen on Figure 2.4. This Reference frame is used to describe the satellites attitude in relation to other reference frames [8].

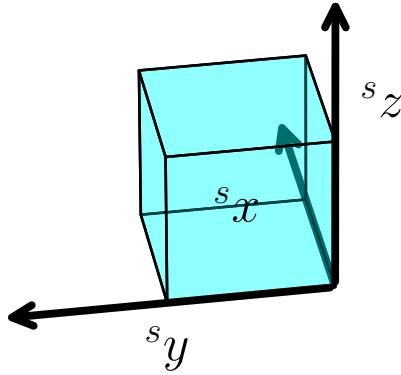


Figure 2.4: Satellite Reference Frame

2.2.5 Control Reference Frame

This reference frame has its origin point placed in the center of mass. The z-axis is defined as pointing in the direction of the major-axis where the Moment of Inertia is the largest. The x-axis is defined as the direction of the minor-axis where the Moment of Inertia is the smallest. The y-axis is then defined as the last axis, being orthogonal to the other two. This can be seen on Figure 2.5.

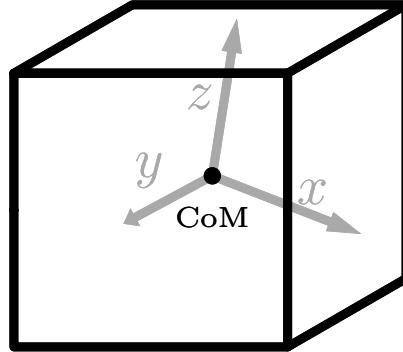


Figure 2.5: Control Reference Frame

As defined above the axes do not necessarily lie parallel with the geometric sides of the satellite. The only instance where this would be the case is if the satellite is a cube with a homogenous mass distribution [9].

This reference frame makes it easier to do calculations, since in this reference frame the Moment of Inertia matrix is diagonal [10].

Another feature is that it is fixed relative to the satellite, meaning it does not change over time.

2.2.6 Target Reference Frame

The target reference frame is used to describe the attitude of the satellite in relation to the target, this is used for tracking. The target is only visible when the satellite is above the horizon of the target, meaning the satellite has to be above the tangent plane of the target. The origin point of the target reference frame is placed in the center of mass of the satellite. The z-axis is pointing from the satellite towards the target on earth. The x-axis lies in the same plane as the velocity vector of the satellite, and orthogonal with both the z-axis and the y-axis.

The y-axis is then defined as the last axis, being orthogonal to the other two. The axis can be seen on Figure 2.6.

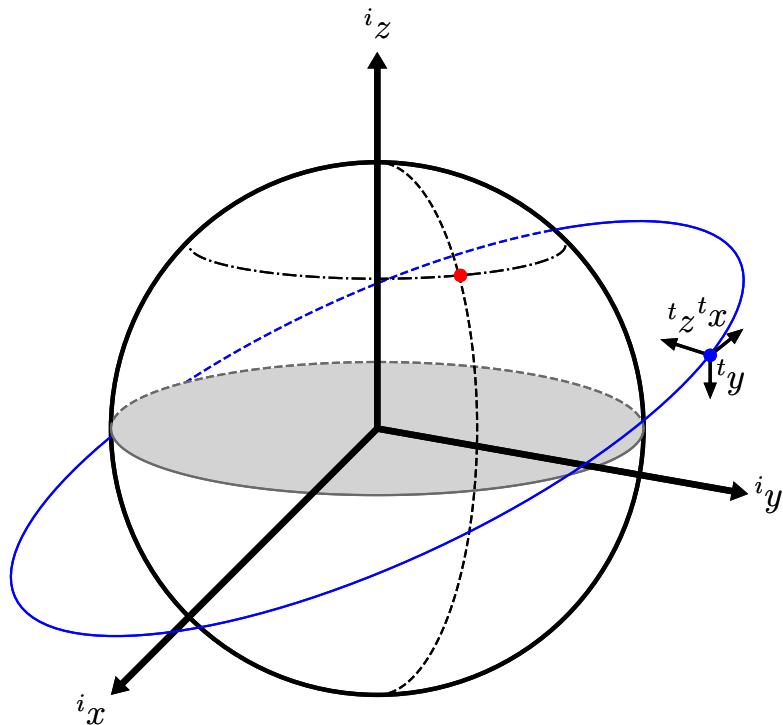


Figure 2.6: Figure of target reference frame.

With this all the required frames have been described. Next the different actuators for the attitude control can be described and the disturbances can be calculated.

Chapter 3

Project delimitation

The mission objective for AAUSAT6 is to capture high-resolution images of Aalborg municipality using a CubeSat with a camera payload, and to inspire interest in space technology among students and the public.

The satellite will be launched into a low Earth orbit, where it will take images of the city of Aalborg and its surrounding areas. The satellite will be controlled and operated by a team of students from Aalborg University, who will receive the images captured by the satellite and process them for public viewing.

Since AAUSAT6 is supposed to have a camera payload it is necessary to design an ADCS which have greater control authority and better precision than the ADCS's used on the previous AAUSAT satellites. The previous AAUSAT satellites have used magnetorquers to control the attitude of the satellite. It is highly likely that these will also be used on AAUSAT6 in tandem with another way of actuation the attitude of the satellite. Since an ADCS using magnetorquers has already been developed and demonstrated in orbit, it has been decided that this part of the ADCS will be out of the scope of this project.

The same is true for the attitude determination system, and thus it has also been decided to focus on the ACS part of the ADCS system. The attitude determination part of the project is therefore assumed to be ideal.

Chapter 4

Technical Analysis

To get an understanding of what requirements have to be set for the system, a technical analysis will be made. The disturbance torques will be analyzed and calculated, and the total disturbance will be determined. The allowable error, settling time and overshoot for the ADCS, will be calculated from the mission specification. Finally, the different attitude parametrizations will be analyzed and compared.

4.1 Satellite Control

To control the satellite, the ADCS must apply a torque to the satellite. This torque can be applied by using magnetorquers, thrusters or reaction wheels. These three methods are described in the following section.

4.1.1 Magnetorquers

Magnetorquers are the most common way of controlling the attitude of small satellites.

When using a magnetorquer to control the attitude of the satellite, the magnetorquer creates a magnetic dipole, by letting current through a coil, which then interacts with the magnetic field of the Earth, to control the attitude. The purpose is to create a controllable magnetic torque which interacts with the magnetic field of the Earth creating a mechanical torque [11].

4.1.2 Thrusters

Thrusters use a propellant to generate a force, which can be used to control the attitude of the satellite. They are often used on larger satellites, but can also be used on smaller satellites [12]. As they require a propellant, they are not suitable for long duration missions, or missions where the satellite must be able to change its attitude many times [13]. For this reason, thrusters are not a viable option for AAUSAT6, and will not be considered further.

4.1.3 Reaction Wheels

Reaction wheels use the conservation of angular momentum to control the attitude of the satellite. Multiple reaction wheels are mounted in a constellation on the satellite, and by accelerating or decelerating the wheels, a torque can be applied [13].

When a reaction wheel is accelerated, a torque is applied to the satellite in the opposite direction of the rotation of the wheel [14]. To apply a continuous torque, such as cancelling disturbance torques, the reaction wheel must be accelerated continuously. This means that the reaction wheel will eventually reach its maximum speed, and will not be able to apply any more torque. For this reason, reaction wheels are often used in tandem with magnetorquers, which can be used to slow down the reaction wheel, and thus allow it to accelerate again [15]. This maneuver is called “momentum dumping” and is considered to be out of scope for this project.

This project will focus on using reaction wheels to control the attitude of the satellite.

4.2 Disturbance torques

When designing a control system it is essential to model the expected disturbances, to ensure that the system is able to handle them. The disturbances evaluated in this section are the gravity gradient, magnetic field, solar radiation, aerodynamic drag and residual magnetic field.

4.2.1 Solar radiation

Since photons carry momentum they can assert a force on the satellite.

This force can cause a torque on the satellite around the center of mass, depending on the radiation intensity, angle of attack and optical properties of the surface of the satellite [16].

The optical properties of the surface of the satellite decides to what degree the surface reflects or absorbs the radiation. The force due to solar radiation can be found using Equation 4.1 [17].

$$f_{sr} = \frac{F_s}{c} A_s (1 + q) \cos(\theta) \quad (4.1)$$

f_{sr}	Force generated due to solar radiation
F_s	Constant solar power per square meter
c	Speed of light
A_s	Surface area of the satellite
q	Reflection factor between 1 or 0.
θ	Angle of incidence

Knowing the force asserted from the solar radiation, the torque from the force can be found

by knowing the distance between the center of pressure and the center of mass. This can be expressed using Equation 4.2.

$$\tau_{sr} = f_{sr} \sin(\varphi) \quad (4.2)$$

τ_{sr}	Resulting torque from solar radiation
f_{sr}	Force generated due to solar radiation
r	Distance between the center of pressure from solar radiation and center of mass
φ	Angle of the force relative to the distance vector.

From Equation 4.1 it can be seen that the force will be largest if all the light is reflected back from where it came and the angle of attack is normal to the satellites surface ie. $\theta = 0$ and $q = 1$. Furthermore, it can be seen that the torque is the largest when the center of mass and the center of pressure is the furthest away from each other. Though in reality it would not be possible to have both a 0-degree angle of attack and a 90-degree angle to the centre of mass, the disturbance is chosen to be calculated for with both of these extremes, to get the worst case torque.

From earlier AAUSAT satellites it is known that usually the center of mass is located close to geometric center of the satellite. Therefore, the worst case scenario will be if the center of pressure is placed at the edge of the satellite. Assuming a 1U satellite the worst case scenario solar radiation can be calculated.

This can be seen in Equation 4.3 [16].

$$f_{sr} = \frac{1362 \text{ W/m}^2}{300 \text{ Mm/s}} 0.01 \text{ m}^2 (1) \cos(0) = 0.9 \text{ nN} \quad (4.3)$$

And the torque τ_{sr} can be calculated as seen in Equation 4.4.

$$\tau_{sr} = 0.9 \text{ nN} 0.05 \text{ m} \sin(90) = 4.5 \text{ nN m} \quad (4.4)$$

Given the nature of the orbit of the satellite, the angle of attack will always be the same, this results in the rotation being around the z-axis.

4.2.2 Aerodynamic drag

As the altitude above Earth rises the density of the atmosphere decreases. In low Earth orbit the density of the atmosphere is vanishingly low, compared to the density at sea level. It is, however, still present and the magnitude of the torque which it can exert on the satellite has to be determined. The aerodynamic drag force can be expressed using Equation 4.5 [18].

$$f_{ad} = \frac{1}{2} \rho C_d A_s v^2 \quad (4.5)$$

f_{ad}	Resulting force from aerodynamic drag
ρ	Force generated due to solar radiation
C_d	Drag coefficient
A_s	Surface area in the direction of travel
v	Velocity of the satellite

The drag coefficient is different for different shapes, and the drag coefficient for a cube is 1.05 [19]. As described earlier the satellite is a 1U CubeSat, and the surface area in the direction of travel is thus 0.01 m^2 . The atmospheric density is dependent on the altitude of the satellite, and is very small in Low Earth Orbit. There is however still an aerodynamic drag present, and the magnitude of the torque which it can generate on the satellite has to be determined.

The table in Appendix B.1 shows the atmospheric density at different altitudes, however at altitudes above 200 km the data points are given with intervals of 50 km. Therefore, in order to obtain an estimation of the aerodynamic drag within a reasonable margin, it is assumed that the force acting on the *bottom* of the satellite is determined at the altitude of 400 km, while the force acting on the *top* of the satellite is determined at the altitude of 450 km.

It is assumed that the surface area of the satellite is 0.005 m^2 at the *bottom* and at the *top*. This assumption is illustrated in Figure 4.1.

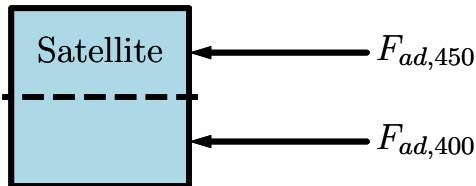


Figure 4.1: The assumption of different atmospheric densities at different altitudes.

$F_{ad,400}$	Aerodynamic drag force at 400 km
$F_{ad,450}$	Aerodynamic drag force at 450 km

The force at an altitude of 400 km can now be calculated using Equation 4.5.

$$F_{ad,400} = \frac{1}{2} 1.585 \text{ ng/m}^3 0.005 \text{ m}^2 1.057628 \text{ m/s}^2 = 568.9 \text{ nN} \quad (4.6)$$

And the force at an altitude of 450 km:

$$F_{ad,450} = \frac{1}{2} 3.725 \text{ ng/m}^3 0.005 \text{ m}^2 1.057628 \text{ m/s}^2 = 242.1 \text{ nN} \quad (4.7)$$

The total force from aero dynamic drag is then calculated as:

$$F_{ad} = F_{ad,400} - F_{ad,450} = 326.8 \text{ nN} \quad (4.8)$$

Similarly to the solar radiation when there is a distance between the center of pressure and the center of mass this force will induce a torque on the satellite. This can be described using Equation 4.9.

$$\tau_{ad} = F_{ad} r \quad (4.9)$$

τ_{ad}	Resulting torque from aerodynamic drag
F_{ad}	Force generated due to aerodynamic drag
r	Distance between the center of pressure and center of mass

The distance between the center of pressure and the center of mass is dependent on the shape of the satellite, and it is assumed that the center of mass is located at the geometric center of the satellite which is also assumed when calculating the solar radiation and therefore the same. This distance r is therefore 0.05 m. The torque from aerodynamic drag can now be calculated using Equation 4.9.

$$\tau_{ad} = 1.556 \mu\text{N} 0.05 \text{ m} = 16.3 \text{ nN m} \quad (4.10)$$

This is assumed to be purely around the y-axis in the SRF as the satellite when controlled is traveling in the x-direction.

4.2.3 Magnetic disturbance

A satellite consist of multiple electrical circuits, which creates a magnetic field. This magnetic field from the satellite interacts with the magnetic field of the Earth, creating some disturbance [11].

When the magnetic moment of the satellite reacts with the magnetic field of the Earth, the resulting mechanical torque is described in Equation 4.11

$$\tau = \mathbf{m} \times \mathbf{B} \quad (4.11)$$

τ	Resulting mechanical torque from the magnetic fields
\mathbf{m}	Magnetic moment of the satellite
\mathbf{B}	Magnetic field of the Earth

As the resulting torque is the cross product of the magnetic torque and the magnetic field of the Earth it is perpendicular to those vectors, which is illustrated in Figure 4.2 [11]

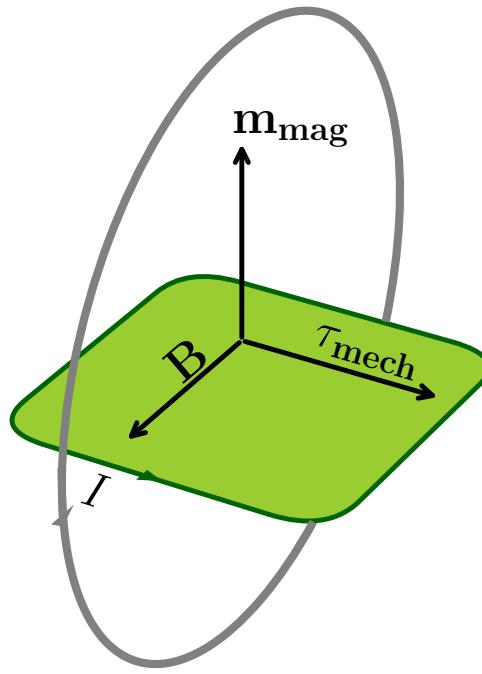


Figure 4.2: Example of the Earths magnetic field \mathbf{B} , interacting with the magnetic dipole moment \mathbf{m}_{mag} , creating the mechanical torque τ_{mech} , acting on the square shaped magnet area, illustrated as the grey plane, with a current I .

The magnetic disturbance for a certain angle is given in Equation 4.12

$$\tau_{magd} = m_s B_{Earth} \sin(\theta) \quad (4.12)$$

τ_{magd}	Torque created because of the magnetic disturbance
m_s	Magnetic moment created by the satellite
B_{Earth}	Magnetic field of the Earth
θ	Angle between \mathbf{m}_s and \mathbf{B}_{Earth}

The magnetic moment created by the satellite, \mathbf{m}_s is difficult to determine, but it has previously been determined to be around $0.01 \text{ A m}^2/\text{kg}$ in a previous master thesis [20]. If the satellite is then assumed to weigh around 1 kg , \mathbf{m}_s is equal to 0.01 A m^2 .

The strength of the Earths magnetic field at a certain altitude can be approximated by Equation 4.13 [21].

$$B(r) \approx B_0 \left(\frac{R_E}{r} \right)^3 \quad (4.13)$$

$B(r)$	Strength of the magnetic field.
B_0	Strength of the magnetic field at the surface of the Earth.
R_E	Radius of the Earth.
r	Distance from the surface of the Earth to the satellite.

If the satellite is considered to be in Low Earth Orbit, the perigee height is 397 km, which is where the magnetic disturbance is the greatest. The strength of the field at the perigee height is therefore calculated to be:

$$B(397 \text{ km}) = 31.2 \mu\text{T} \left(\frac{6371 \text{ km}}{6768 \text{ km}} \right)^3 = 26 \mu\text{T} \quad (4.14)$$

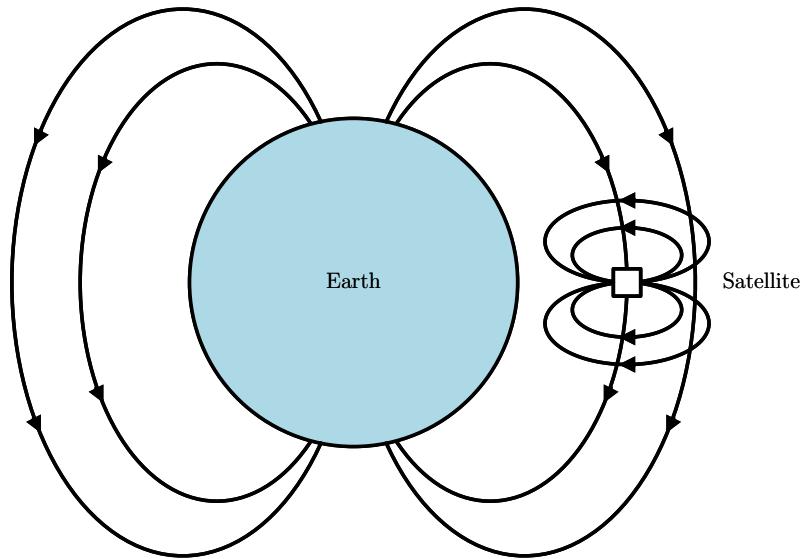


Figure 4.3: Visual representation of the magnetic fields of the Earth and the satellite.

In the worst case scenario, the angle between the magnetic field of the Earth and the magnetic moment of the satellite is 90° .

The disturbance can then be calculated as:

$$\tau_{magd} = 0.01 \text{ A m}^2 28.1 \mu\text{T} = 260 \text{ nN m} \quad (4.15)$$

Using the right hand rule the direction of the torque is around the y -axis, which is the same direction as the torque from the aerodynamic drag.

4.2.4 Gravity gradient

The gravitational force between the earth and the satellite, is what is keeping the satellite in orbit. However, if the mass of the satellite is not uniformly distributed, then the gravity will not pull uniformly on it. This will induce a torque on the satellite, which is given by Equation 4.17 [18].

$$d\mathbf{f}_i = \frac{-\mu s_i dm_i}{|\mathbf{s}_i|^3} \quad (4.16)$$

$d\mathbf{f}_i$	Force acting on each individual mass element dm_i .
μ	Standard gravity parameter GM where G is the gravitational constant and M is the mass of the earth.
m_i	Mass of different elements in the satellite
s_i	Distance between the center of mass of Earth and the mass m_i in the satellite.

With the gravitational force known on all different masses in the satellite, the torque can be computed

$$d\boldsymbol{\tau}_i = \mathbf{r}_i \times d\mathbf{f}_i \quad (4.17)$$

\mathbf{r}_i	Distance from the satellites center of mass to the mass m_i .
\mathbf{f}_i	Gravitational force acting on the mass m_i .

By integrating Equation 4.17, the Gravity gradient can be computed using Equation 4.18 [22].

$$\boldsymbol{\tau}_{GG} = \frac{3\mu}{s^3} (\hat{\mathbf{s}} \times (\underline{\mathbf{J}} \cdot \hat{\mathbf{s}})) \quad (4.18)$$

s	Distance from the satellites center of mass to the earth.
$\underline{\mathbf{J}}$	Inertia matrix of the satellite.
$\hat{\mathbf{s}}$	Zenith in the CRF

Thus the gravity gradient only depends on the inertia matrix of the satellite and the distance from the satellite to the earth.

Since AAUSAT6 is not built yet, the inertia matrix for AAUSAT6 is not known. It is assumed that AAUSAT6 will have the same external size and structure as AAUSAT3. Thus, it is chosen to use the inertia matrix for AAUSAT3 as a good estimate for the inertia matrix for AAUSAT6 [23].

$$\underline{\mathbf{J}}_s = \begin{bmatrix} 0.0017 & 0 & 0 \\ 0 & 0.0022 & 0 \\ 0 & 0 & 0.0022 \end{bmatrix} \text{ kg m}^2 \quad (4.19)$$

s is the vector that starts in the center of the earth and ends at the satellite's center of mass. From this vector the length/distance is denoted with "s" and the position of the vector is denoted \hat{s} . s is based on the orbit of AAUSAT4 which gives an "s" of 6814 km.

Since the vector \hat{s} is the zenith, which is the opposite of the nadir vector, in the CRF, this vector also represents the orientation of the satellite. With these values the maximal disturbance from the gravity gradient can be found using the vectors $\hat{s} = [1, 0, 1]^T / \sqrt{2}$ and $\hat{s} = [1, 1, 0]^T / \sqrt{2}$

Using these values the maximum gravity gradient disturbance can be calculated, with Equation 4.18, to be $\|\tau_{GG}\|_2 = 0.94491 \text{ nN m}$. This is a relatively low disturbance compared to the other disturbances, therefore it is decided to compute the gravity gradient at an orientation where the other disturbances are at their maximum. This is when $\hat{s} = [0, 0, 1]^T / \sqrt{2}$. Inserting this vector into Equation 4.18 the gravity gradient disturbance is found to be $|\tau_{GG}| = 0 \text{ nN m}$.

4.2.5 Summary of disturbances

The disturbances found earlier, as well as the total disturbance can be seen in Table 4.1. The total disturbance is calculated as shown in Equation 4.20.

$$\tau_{total} = \sqrt{\tau_{sr}^2 + (\tau_{ad} + \tau_{magd})^2} \quad (4.20)$$

Since the satellite is assumed to have the same orbit as AAUSAT4 the solar radiation will always apply the torque in around the x-axis in the ORF. This is due to the orbit being sun-synchronous which means that the satellite will always be visible from the sun. The other disturbances are maximized when the ORF and SRF line up. Therefore, the magnetic disturbance and the aerodynamic disturbance will apply in the same direction and can therefore be summed, and they are perpendicular to the solar radiation, therefore the 2-norm is taken. A sketch of the different disturbances can be seen in Figure 4.4.

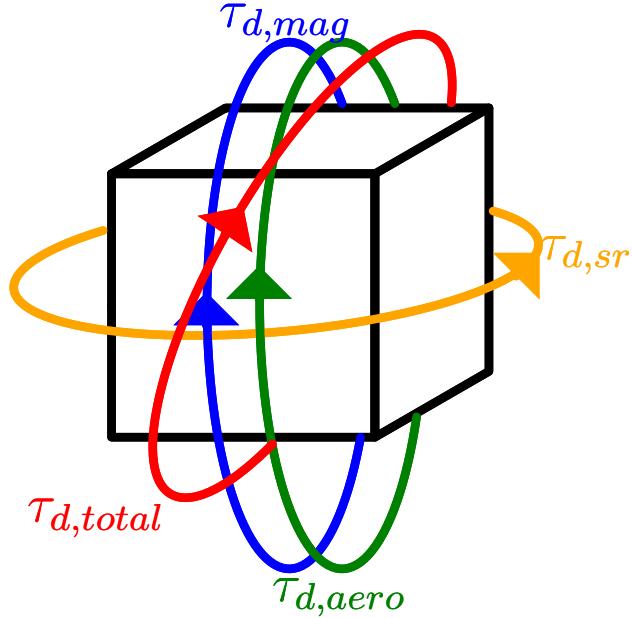


Figure 4.4: Sketch of the different disturbances on the satellite, as well as the total.

Disturbance type	Value
Solar radiation	4.543 nN m
Aerodynamic drag	16.34 nN m
Magnetic disturbance	260.2 nN m
Gravity gradient	0.94 nN m
Total	281.4 nN m

Table 4.1: Table of the different disturbances and the total disturbance.

The table shows that the magnetic disturbance is the largest disturbance, and the gravity gradient is the smallest.

4.3 Pointing precision

The payload of the satellite is a camera, which is used to take pictures of Aalborg Municipality. The specifications of the camera is important for the satellite to be able to take pictures of the municipality. The camera used is a *16 mm Telephoto Lens for RaspberryPi HQ camera* [24].

The specifications of the camera can be seen in Table 4.2

Symbol	Description	Value
px_{size}	Pixel size of the camera	1.55 μm
f_{lens}	Focal length of the lens	16 mm
r	Resolution of the camera	4056 px \times 3040 px

Table 4.2: Camera specifications

The instantaneous field of view of the camera is found by using Equation 4.21 [25]

$$\theta_{IFOV} = 2 \arctan \left(\frac{px_{size}}{2 f_{lens}} \right) = 96.9 \mu\text{rad}/\text{px} \quad (4.21)$$

In order to calculate the area which can be captured of a single pixel is found by using Equation 4.22.

$$A = 2d \tan \left(\frac{\theta_{IFOV}}{2} \right) \quad (4.22)$$

- A Single dimension of the area of Earth covered by a single pixel
- d Distance from the satellite to the Earth

When the satellite is at its lowest altitude, which is 397 km, the dimensions captured by a single pixel is calculated as:

$$A = 2 \cdot 397 \text{ km} \tan \left(\frac{96.9 \mu\text{rad}}{2} \right) = 38.46 \text{ m}/\text{px} \quad (4.23)$$

The dimension of the total area the camera can capture, while its pointing directly down at the earth, can be calculated using Equation 4.24.

$$A_{total} = A n_{px} \quad (4.24)$$

Since the resolution of the camera is 4056 px \times 3040 px, the total area captured by the camera can be calculated as:

$$A_{w,cam} = 4056 \text{ px} \cdot 38.46 \text{ m}/\text{px} = 155.9 \text{ km} \quad (4.25)$$

$$A_{h,cam} = 3040 \text{ px} \cdot 38.46 \text{ m}/\text{px} = 116.9 \text{ km} \quad (4.26)$$

$A_{w,cam}$	Width of the area the camera can see from the satellite.
$A_{h,cam}$	height of the area the camera can see from the satellite.

The area of Aalborg municipality is approximately 60×46 km, the margin of error where the municipality is still captured can be calculated as:

$$A_{w,e} = \frac{A_{w,cam} - A_{w,AA}}{2} = \frac{155.9 \text{ km} - 60 \text{ km}}{2} = 47.96 \text{ km} \quad (4.27)$$

$$A_{h,e} = \frac{A_{h,cam} - A_{h,AA}}{2} = \frac{116.9 \text{ km} - 46 \text{ km}}{2} = 35.46 \text{ km} \quad (4.28)$$

$A_{w,AA}$	Width of Aalborg municipality.
$A_{h,AA}$	Height of Aalborg municipality.
$A_{w,e}$	Allowable error in the width.
$A_{h,e}$	Allowable error in the height.

Knowing the margin of error, the maximum allowable error can be calculated as:

$$\theta_{w,e} = \frac{A_{w,e} \theta_{IFOV}}{A} = \frac{47.96 \text{ km} 96.9 \mu\text{rad}/\text{px}}{38.46 \text{ m}/\text{px}} = 120.8 \text{ mrad} \approx 6.9^\circ \quad (4.29)$$

$$\theta_{h,e} = \frac{A_{h,e} \theta_{IFOV}}{A} = \frac{35.46 \text{ km} 96.9 \mu\text{rad}/\text{px}}{38.46 \text{ m}/\text{px}} = 89.34 \text{ mrad} \approx 5.1^\circ \quad (4.30)$$

(4.31)

$\theta_{w,e}$	Allowable error in the width as an angle from the satellite.
$\theta_{h,e}$	Allowable error in the height as an angle from the satellite.

Therefore when pointing on Aalborg municipality, the maximum deviation is 5.1° in order to capture the total area of the municipality.

4.3.1 Step from Nadir to pointing

Pointing starts when the satellite is above the tangent plane of Aalborg on earths surface, meaning the angle between earths surface and the satellite is 0° . When the satellite is in Nadir mode, the angular velocity of the satellite can be calculated as:

$$\omega_s = \frac{v_s}{r_s} = \frac{7628 \text{ m/s}}{6.768 \text{ Mm}} = 0.0011 \text{ rad/s} \quad (4.32)$$

4.3.1.1 The satellite in reference to Aalborg municipality

In order to find plot the graphs of the satellites angle, distance to target and angular velocity, while it passes over Aalborg municipality, Figure 4.5 is made. The pass here is chosen to be the one where the satellite flies directly over Aalborg, as this will require the highest angular velocity change and is therefore the hardest to control.

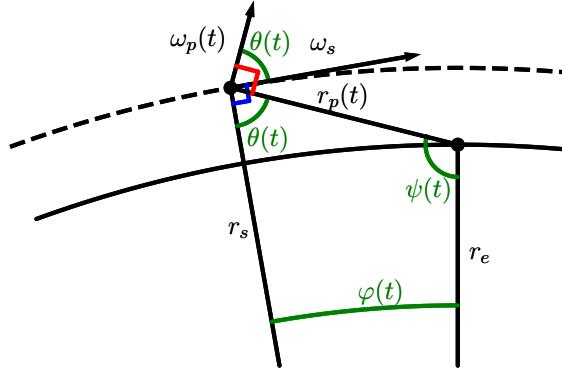


Figure 4.5: The satellite in reference to Aalborg municipality.

Integrating the angular velocity over time gives the angle as given in Equation 4.33.

$$\varphi(t) = \int \omega_s dt = \omega_s t + \varphi_0 \quad (4.33)$$

Where φ_0 is the starting angle, this is decided as the angle when pointing starts i.e. Equation 4.34.

$$\varphi_0 = \cos^{-1} \left(\frac{r_e}{r_s} \right) = 0.3442 \text{ rad} \quad (4.34)$$

With $\varphi(t)$, r_p can be found using the cosine relation as given in Equation 4.35.

$$r_p(t) = \left(r_s^2 + r_e^2 - 2 r_s r_e \cos(\varphi(t)) \right)^{\frac{1}{2}} \quad (4.35)$$

Now using the sine relation $\theta(t)$ can be found as given in Equation 4.36.

$$\theta(t) = \arcsin \left(\frac{\sin(\varphi(t))}{r_p(t)} r_e \right) \quad (4.36)$$

With this, the angular velocity in reference to the pointing point can be found as given in Equation 4.37.

$$\omega_p(t) = \frac{v_s}{r_p(t)} \cos(\theta(t)) \quad (4.37)$$

The manoeuvre is plotted in Figure 4.6.

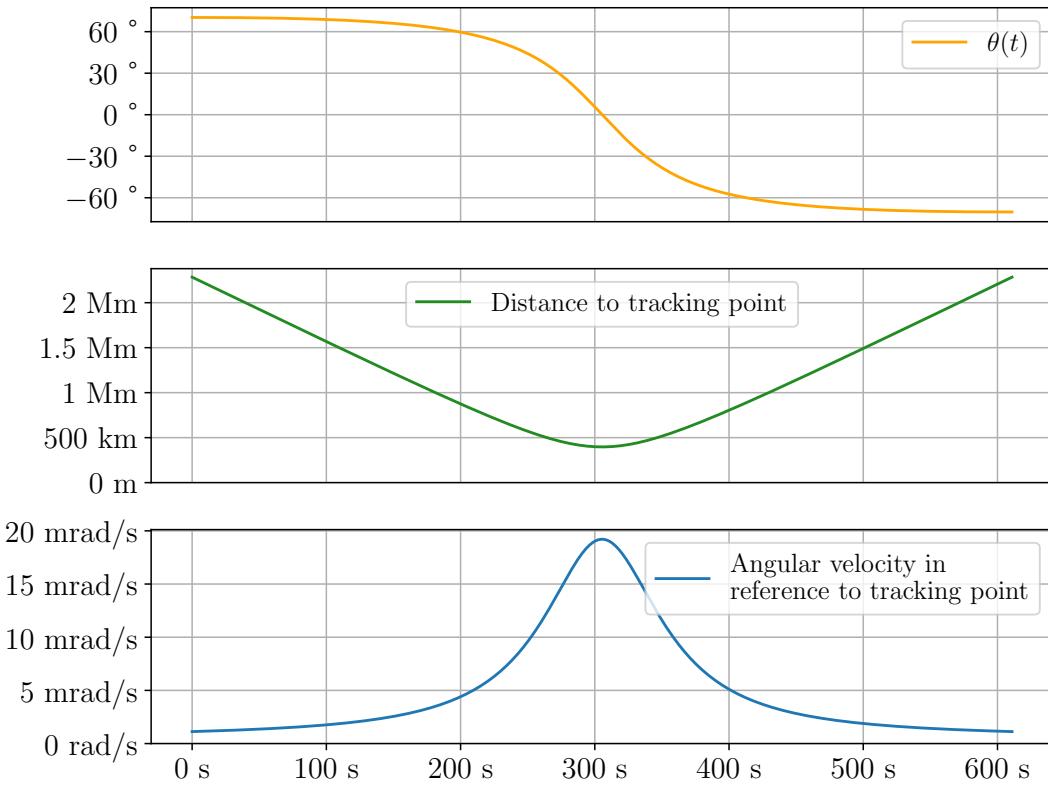


Figure 4.6: Graphs of angle distance and angular velocity in pointing mode.

If the satellite is passing directly above Aalborg the whole pointing period will be 611 seconds.

4.3.1.2 Acceleration

With the angular velocity, the angular acceleration can also be found by Equation 4.38.

$$\alpha(t) = \frac{d\omega(t)}{dt} = \frac{d}{dt} \left[\frac{v_s}{r_p(t)} \cos(\theta(t)) \right] \quad (4.38)$$

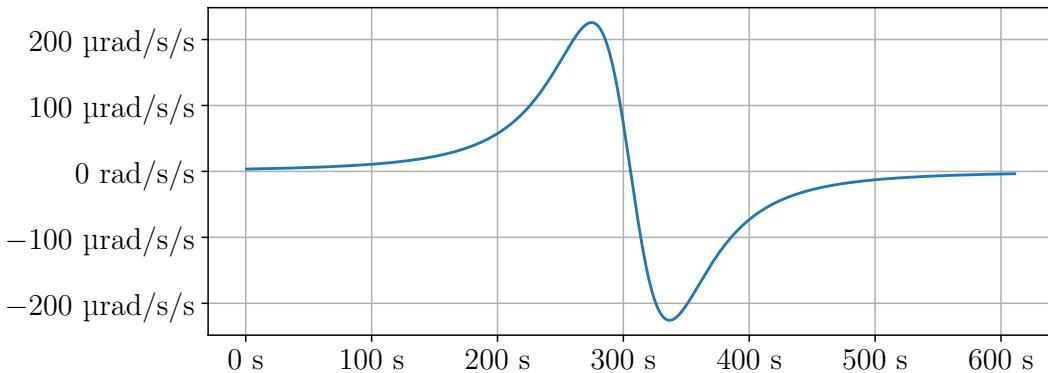


Figure 4.7: Rotational acceleration of the satellite when pointing.

The maximum angular acceleration of the manoeuvre is read in Figure 4.7 to be $225.8 \mu\text{rad}/\text{s}^2$.

4.3.1.3 Settling time

To determine the settling time it is decided that the satellite should have settled by the time the view of Aalborg municipality is 20 % of the total view height-wise. This is because the view of Aalborg Municipality is not interesting if it is not visible or the view is at such an angle that it is very distorted.

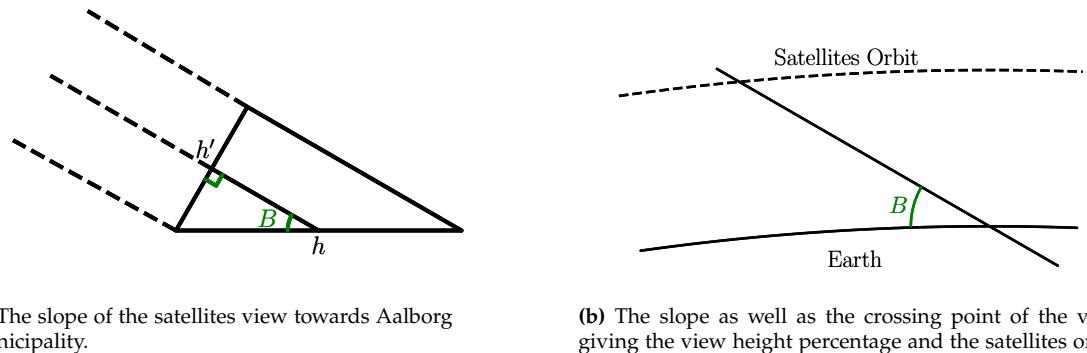


Figure 4.8: The slope of the satellites view towards Aalborg municipality, viewed from a close up and afar.

Where h is the height of the municipality, h' is the height of the viewable area of the municipality, meaning $h' = h \cdot \text{view}\%$, the angle B found using trigonometry:

$$B = \sin^{-1} \left(\frac{h'}{2} \right) = \sin^{-1} \left(\frac{0.2 \cdot 46 \text{ km}}{46 \text{ km}} \right) = 201.4 \text{ mrad} \approx 11.5^\circ \quad (4.39)$$

For ease of calculations, the earth is rotated such that Aalborg municipality is at the top, on the y-axis of a coordinate system with origin at Earth's center. Giving the line that crosses the orbit a slope of $-\tan(B)$ and a crossing of the y axis at the coordinate $(0, r_e)$:

$$y = -\tan(B)x + r_e \quad (4.40)$$

The intersection of the line and the circle is found:

$$\begin{aligned} x^2 + y^2 &= r_s^2 ; y = -\tan(B)x + r_e \\ x^2 + y^2 &= 6768 \text{ km}^2 ; y = -\tan(201.4 \text{ mrad})x + 6371 \text{ km} \\ &\Downarrow \\ X &= -1314 \text{ km} \\ Y &= 6639 \text{ km} \end{aligned}$$

This crossing point gives the satellite's position in the coordinate system, and the distance between the satellite and Aalborg municipality can be found to be 1341 km. This distance can be used with the distance function found in Section 4.3.1.1 to get the settling time. This gives a settling time of 132 s.

4.3.1.4 Overshoot

At the point in the orbit, where pointing mode begins. The distance to the target, Aalborg municipality, is the far side of a right angled triangle.

$$r_{p0} = \sqrt{r_s^2 - r_e^2} = \sqrt{6768 \text{ km}^2 - 6371 \text{ km}^2} = 2284 \text{ km} \quad (4.41)$$

With this, the area per pixel of the camera can be found.

$$d = 2 \cdot 2284 \text{ km} \tan\left(\frac{\theta_{IFOV}}{2}\right) = 129.9 \text{ km/px} \quad (4.42)$$

Multiplying this with the camera resolution of 4056×3040 px gives an area of 943×707 km. The height and width of Aalborg municipality is 60 km by 46 km giving a viewable height of $0.2 \cdot 46 \text{ km} = 9.2 \text{ km}$. This can be seen on Figure 4.9.

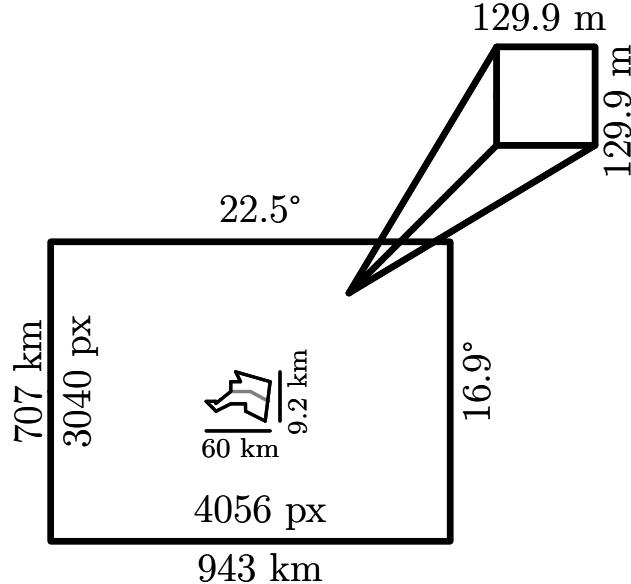


Figure 4.9: The view of Aalborg municipality from the camera, when the satellite starts pointing.

This means the camera can be $\frac{707-9.2}{2} = \pm 348.9 \text{ km}$ off and still capture all of Aalborg municipality. This corresponds to $\frac{348.9 \text{ km}}{129.9 \text{ m}} = 2685 \text{ px}$. This gives a maximum overshoot of $\theta_{IFOV} \cdot 2685 \text{ px} = 260.2 \text{ mrad} \approx 14.9^\circ$

4.3.2 Reaction wheel control time

One of the disadvantages of using reaction wheels is that they will eventually saturate. Since the control systems is intended to use both the magnetorquers and the reaction wheels, it is only required for the reaction wheels to be able to keep the satellite in the desired orientation

for a limited amount of time. This time is arbitrarily chosen to be the time it takes to make 10 orbits, since this is enough time to transfer the control to the reaction wheels and for them to stabilize and then to take pictures during several passes over the target area.

4.4 Attitude Parametrization

There are a lot of different ways to parametrize the attitude of the satellite. It is important to decide how to parametrize the attitude of the satellite since there are advantages and disadvantages for each parametrization.

The attitude of the satellite can be described as a coordinate system representing the satellite rotated in relation to another coordinate system or reference frame. In this case the coordinate system representing the satellite would be the SRF and usually the reference frame in which the coordinate systems is rotated in relation to could for example be the ECI frame.

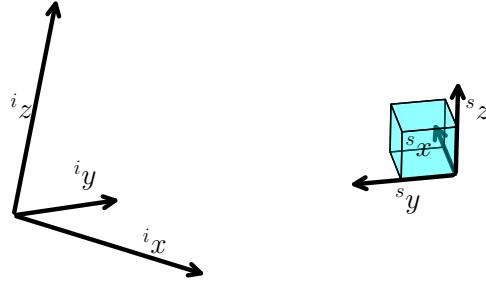


Figure 4.10: Figure of the SRF represented in the ECI frame.

4.4.1 Euler angles

A way to parametrize the attitude of a satellite is to use Euler angles. Euler angles describe the orientation of a rigid body in space by rotating the body about a sequence of three axes. The three angles used to specify the rotations are often referred to as yaw, pitch, and roll [26]. The three axes are illustrated on Figure 4.12.

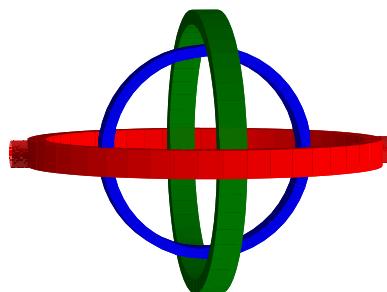


Figure 4.11: Figure of euler angles in a 3D space.

One of the drawbacks of Euler angles is that they suffer from the problem of gimbal lock. Gimbal lock occurs when two of the axes in the sequence become aligned, causing a loss of one degree of freedom in the rotation. This can be seen on Figure 4.12 where the blue circle and the red circle are aligned. This can make it difficult to represent certain types of rotations using Euler angles. For instance in Figure 4.12, if a rotation of the blue circle along its own perimeter is wanted, then this will not be possible without rotating both the red circle and the green circle.

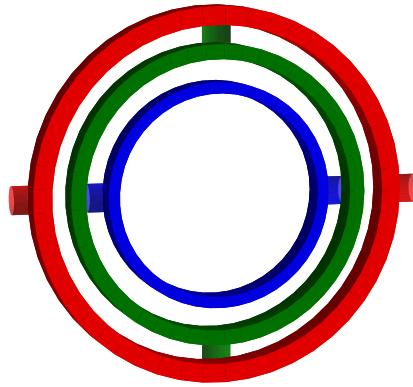


Figure 4.12: Figure of euler angles in gimbal lock.

4.4.2 Euler axis-angle

Another approach is to use the Euler axis and angle representation. The Euler axis and angle representation is a way of specifying a rotation in three-dimensional space by defining an axis of rotation and an angle of rotation. The axis of rotation is a unit vector that specifies the direction of the rotation.

This representation has the advantage of avoiding the problem of gimbal lock [27] and is shown on Figure 4.13.

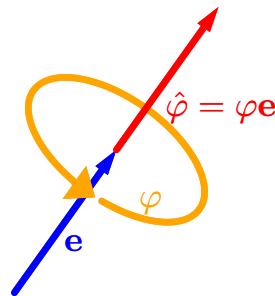


Figure 4.13: Visual representation of the Euler axis and angle representation.

To use the Euler axis and angle representation, you first need to identify the axis of rotation. For any given rotation matrix $\underline{\mathbf{A}}$ there exists an eigen vector $\hat{\mathbf{e}}$ defined as the vector which satisfies the equation $\underline{\mathbf{A}}\hat{\mathbf{e}} = \hat{\mathbf{e}}$. The rotation can then be described by the rotation matrix $\underline{\mathbf{A}}$ given by [22]:

$$\underline{\mathbf{A}}(\varphi, \hat{\mathbf{e}}) = \begin{bmatrix} \cos(\varphi) + e_1^2(1 - \cos(\varphi)) & e_1 e_2(1 - \cos(\varphi)) + e_3 \sin(\varphi) & e_1 e_3(1 - \cos(\varphi)) - e_2 \sin(\varphi) \\ e_1 e_2(1 - \cos(\varphi)) - e_3 \sin(\varphi) & \cos(\varphi) + e_2^2(1 - \cos(\varphi)) & e_2 e_3(1 - \cos(\varphi)) + e_1 \sin(\varphi) \\ e_1 e_3(1 - \cos(\varphi)) + e_2 \sin(\varphi) & e_2 e_3(1 - \cos(\varphi)) + e_1 \sin(\varphi) & \cos(\varphi) - e_3^2(1 - \cos(\varphi)) \end{bmatrix} \quad (4.43)$$

Another advantage of the Euler axis and angle representation is a more intuitive way of representing rotations than other methods [27].

However, one drawback of the Euler axis and angle representation is that it is undefined for $\sin(\varphi) = 0$. Additionally, the representation may be less compact than other methods, such as quaternions [27].

4.4.3 Quaternions

Another way to describe rotation is quaternions, which are a four-dimensional extension of complex numbers. A quaternion \mathbf{q} is given by Equation 4.44:

$$\mathbf{q} = q_4 + q_1 i + q_2 j + q_3 k \quad (4.44)$$

where i , j and k are the imaginary units, satisfying the conditions given by Equations 4.45:

$$\begin{aligned} i^2 &= j^2 = k^2 = -1 \\ ij &= -ji = k \\ jk &= -kj = i \\ ki &= -ik = j \end{aligned} \quad (4.45)$$

The quaternions describing rotations has a 2-norm of 1 and is thus called unit quaternions [28]. Unit quaternions therefore satisfy the condition given by Equation 4.46:

$$\|\hat{\mathbf{q}}\|_2 = q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \quad (4.46)$$

The four parameters can be described by Equation 4.47:

$$\begin{aligned} q_1 &= \hat{e}_1 \sin\left(\frac{\varphi}{2}\right) \\ q_2 &= \hat{e}_2 \sin\left(\frac{\varphi}{2}\right) \\ q_3 &= \hat{e}_3 \sin\left(\frac{\varphi}{2}\right) \\ q_4 &= \cos\left(\frac{\varphi}{2}\right) \end{aligned} \quad (4.47)$$

Where $\hat{\mathbf{e}}$ is the vector around which the rotation happens and φ is the angle of the rotation. The rotation of a three-dimensional vector \mathbf{v} can be done using a 3x3 matrix $\underline{\mathbf{A}}$ by Equation 4.48

$$\mathbf{v}' = \underline{\mathbf{A}}\mathbf{v} \quad (4.48)$$

This can also be done using quaternions, which can be seen in Equation 4.49.

$$\mathbf{v}' = \mathbf{q} \mathbf{v} \mathbf{q}^* \quad (4.49)$$

The three-dimensional vector \mathbf{v} represented by a quaternion has a real part of 0.

One advantage of quaternions is that they do not suffer from gimbal lock. This is because quaternions use a different method for combining rotations that does not rely on a sequential rotation about three axes. Instead, quaternions represent rotations using a single rotation about an axis in four-dimensional space. This allows them to represent any possible orientation of a rigid body without the loss of degrees of freedom that can occur with Euler angles [28].

Quaternions also have the advantage over Euler angles, that computation of two successive rotations is relatively easy. Two successive rotations represented by the quaternions \mathbf{q} and \mathbf{q}' can be computed as the product of the two rotation quaternions [22].

$$\mathbf{q}'' = (\mathbf{q}_1 i + \mathbf{q}_2 j + \mathbf{q}_3 k + \mathbf{q}_4) (\mathbf{q}'_1 i + \mathbf{q}'_2 j + \mathbf{q}'_3 k + \mathbf{q}'_4) \quad (4.50)$$

This multiplication can be represented as a matrix multiplication of the scalar parameters of the quaternions, as seen in Equation 4.52.

$$\begin{bmatrix} \mathbf{q}''_1 \\ \mathbf{q}''_2 \\ \mathbf{q}''_3 \\ \mathbf{q}''_4 \end{bmatrix} = \begin{bmatrix} \mathbf{q}'_4 & \mathbf{q}'_3 & -\mathbf{q}'_2 & \mathbf{q}'_1 \\ -\mathbf{q}'_3 & \mathbf{q}'_4 & \mathbf{q}'_1 & \mathbf{q}'_2 \\ \mathbf{q}'_2 & -\mathbf{q}'_1 & \mathbf{q}'_4 & \mathbf{q}'_3 \\ -\mathbf{q}'_1 & -\mathbf{q}'_2 & -\mathbf{q}'_3 & \mathbf{q}'_4 \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \\ \mathbf{q}_4 \end{bmatrix} \quad (4.51)$$

$$\mathbf{q}'' = \underline{\mathbf{S}}_4(\mathbf{q}') \mathbf{q} \quad (4.52)$$

Where $\underline{\mathbf{S}}_4$ is the skew-symmetric $\mathbb{H} \rightarrow \mathbb{R}^{4 \times 4}$ matrix operator defined in Equation 4.53.

$$\underline{\mathbf{S}}_4(\mathbf{q}) = \begin{bmatrix} \mathbf{q}_4 & \mathbf{q}_3 & -\mathbf{q}_2 & \mathbf{q}_1 \\ -\mathbf{q}_3 & \mathbf{q}_4 & \mathbf{q}_1 & \mathbf{q}_2 \\ \mathbf{q}_2 & -\mathbf{q}_1 & \mathbf{q}_4 & \mathbf{q}_3 \\ -\mathbf{q}_1 & -\mathbf{q}_2 & -\mathbf{q}_3 & \mathbf{q}_4 \end{bmatrix} \quad (4.53)$$

This is convenient since it makes it easy to change reference frame. An example of this could be that the attitude of the CRF is known in the ECI but since it is the whole satellite which is controlled in reference to the earth it is interesting to know the attitude of the SRF in the ECI reference frame.

4.4.4 Summary

In summary, there are several ways to parametrize the attitude of a satellite, each with its own advantages and disadvantages. The choice of parametrization will depend on the specific

requirements of the application and the trade-offs between simplicity, accuracy, and computational efficiency. For this project it has been chosen to use quaternions since it offers a representation without the chance of loosing a degree of freedom. Furthermore, quaternions makes it easy to compute multiple successive rotations.

Chapter 5

Specification

From the technical analysis different requirements for the system has been found. In this chapter the technical requirements will be listed.

Index	Specification	Value	Reason
5.1.1	The system must be able to point at the center of earth with a given tolerance.	$\pm 5.1^\circ$	4.3
5.1.2	The system must be able to point at a specific point on earth with a given tolerance.	$\pm 5.1^\circ$	4.3
5.1.3	When changing from nadir mode to pointing mode the system needs to settle before the target is viewable.	< 132 s	4.3.1.3
5.1.4	When changing from nadir mode to pointing mode, the overshoot must be such that the target is still visible.	< 14.9°	4.3.1.4
5.1.5	The system must be able to control the satellite for the whole pointing period, without the reaction wheels reaching saturation.	≥ 611 s	4.3.1.1
5.1.6	The system must be able to control the satellite for a long period without exceeding the specifications.	10 orbits	4.3.2

Table 5.1: Table showing the technical requirements for the distance control system.

Chapter 6

Test Environment

Two options are available as a test environment for the ADCS system. The first option is to use the AAUSAT Simulation Library to simulate the ADCS system. The second is a physical test environment in the form of a testbed. The testbed is a sphere with the satellite mounted inside it, and the sphere is mounted on a stand that makes an air pillow underneath it, which counteracts gravity and removes friction. With this the attitude of the CubeSat inside the sphere can be controlled by the ADCS system. A picture of the testbed can be seen in Figure 6.1.

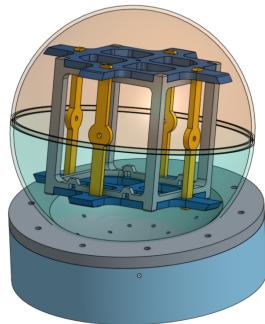


Figure 6.1: Testbed for the ADCS system, made by previous members of AAU satlab.

The advantage of the testbed is that it will also test the hardware of the ADCS system. The disadvantage is that test of ADCS systems are almost impossible on earth. This is because the disturbances on earth are much larger than in space, and the slightest disturbance will make the test unstable.

The other option is to use the AAUSAT Simulation Library to simulate the ADCS system. The AAUSAT Simulation Library is a library for MATLAB/Simulink. It was created by students at Aalborg university[29]. The library is primarily used to create the disturbances and the environment that the satellite is exposed to. Furthermore, the library also includes blocks for:

- Satellite system models
- Actuator model
- Target reference system

- Animation environment

The advantage of using the AAUSAT Simulation Library is that it is much easier to test the ADCS system as the disturbances can be controlled. The disadvantage is that the simulation is not able to completely simulate the final physical system.

It is chosen to use the AAUSAT Simulation Library to simulate the ADCS system. This is because the testbed has been experienced to not be reliable enough to test the ADCS system.

Chapter 7

Test specification

To make sure that the requirements are fulfilled, different test specifications are made. These specifications define how to test for each requirement and are used in the Acceptance test. All tests will be done using the simulation tool Simulink described in Chapter 6. In order to do this multiple libraries will be implemented in the simulation, which all included in the AAUSAT simulation library.

7.1 Nadir mode test

The purpose of the test is to determine whether the prototype can point at the centre of Earth within the given threshold as per Requirement 5.1.1.

7.1.1 Test procedure

The test will be done using Matlab/Simulink, with the reference set to Nadir mode, the simulation will be run for one revolution around earth or approximately 5700 seconds. The reference attitude is therefore changing with $\omega_s = 0.0011 \text{ rad/s}$.

7.1.2 Success criteria

The requirement is fulfilled if the prototype can be in Nadir mode with a precision of $\pm 5.1^\circ$, as described in Requirement 5.1.1.

7.2 Pointing mode test

The purpose of this test is to determine Pointing mode precision, settling time and overshoot as per Requirements 5.1.2, 5.1.3, 5.1.4 and 5.1.5.

7.2.1 Test procedure

The test is performed by simulating the pointing mode procedure, starting at the nadir mode time just before the change to pointing mode. The simulation is run for about 700s and the results are analyzed, until the satellite should be back in nadir mode. After the simulation the results are analyzed to determine whether the requirements are fulfilled.

7.2.2 Success criteria

The requirement is fulfilled if the satellite is able to follow the reference trajectory within the specified tolerance of $\pm 5.1^\circ$. After settling, which should be within the requirement of 132 s, the overshoot is also observed to make sure it is within the requirement of 7.89° .

Lastly it is observed whether the satellite is able to stay in pointing mode for the required time without reaching saturation from the disturbances.

7.3 Combined test

Lastly a durability test will be performed to test the system as a whole.

7.3.1 Test procedure

The simulation is run for a total of 10 orbits, which is about 16 hours as per Requirement 5.1.6.

The following will be measured:

- The attitude reference.
- The angle error.
- The satellite angular velocity.
- The reaction wheel angular velocity.
- The disturbances.

7.3.2 Success criteria

From these measurements the requirements will be deemed fulfilled or failed based on the tolerances given in Chapter 5.

Chapter 8

System Development

With the prerequisites from the technical analysis, the system development can begin. The system development is divided into multiple parts, which describe the chronological development of the system. Throughout the chapter, the mission requirements will be used as to develop a reference generator, the model of the satellite will be developed, and the control system will be designed.

8.1 Satellite Modes

In the context of the ADCS, the satellite has two different modes. Nadir mode for when the satellite is in orbit, and pointing mode for when the satellite is pointing at a target. When shifting into a new mode, the satellite needs to rotate to align the SRF with the new reference frame. To do this the mathematical description of the two reference frames are needed.

Where $\theta(t)$ is the angle of rotation at time t . Given by:

$$\theta(t) = \frac{2\pi t}{t_d} \quad (8.1)$$

$\theta(t)$	Angle between the greenwich meridian and the mean equinox.
t	Time since last alignment of ECI and ECEF.
t_d	Period of a sidereal day.

With this a coordinate in the ECEF can be transformed to ECI, only by knowing the time since the last alignment of the two frames.

8.1.1 Nadir mode

For the satellite to go into nadir mode, the SRF needs to be aligned with the ORF. To do this the axes in the ORF needs to be calculated from the ECI frame.

A visual of the vectors can be found in Section 2.2.3 Figure 2.3. To find ${}^o\mathbf{z}$ the position vector of the satellite from the center of the earth in the ECI frame ${}^i\mathbf{r}_s$ is used. The ${}^o\mathbf{z}$ of the ORF is given as in Equation 8.2.

$${}^o\mathbf{z} = -\frac{{}^i\mathbf{r}_s}{\| {}^i\mathbf{r}_s \|} \quad (8.2)$$

The prescript “o” means that the z axis is in the orbital reference frame. If a prescript is supplied with no reference frame, it is assumed to be in the ECI frame.

It is known that the velocity vector ${}^i\mathbf{v}_s$ for the satellite is pointing in the same direction as the ORF x-axis ${}^o\mathbf{x}$. This is used to find the x-axis of the ORF as in Equation 8.3.

$${}^o\mathbf{x} = \frac{{}^i\mathbf{v}_s}{\| {}^i\mathbf{v}_s \|} \quad (8.3)$$

With two of the axes defined, the last one can be found as in Equation 8.4

$${}^o\mathbf{y} = {}^o\mathbf{z} \times {}^o\mathbf{x} \quad (8.4)$$

Giving a final rotation matrix from ECI to ORF in Equation 8.5.

$${}^i\mathbf{R} = \begin{bmatrix} {}^i\mathbf{v}_{sat} & {}^o\mathbf{z} \times {}^o\mathbf{x} & -\frac{{}^i\mathbf{r}_{sat}}{\| {}^i\mathbf{r}_{sat} \|} \end{bmatrix} \quad (8.5)$$

8.1.2 Target mode

By aligning the SRF with the TRF the satellite goes into pointing mode. An illustration of the TRF can be seen in Figure 8.1.

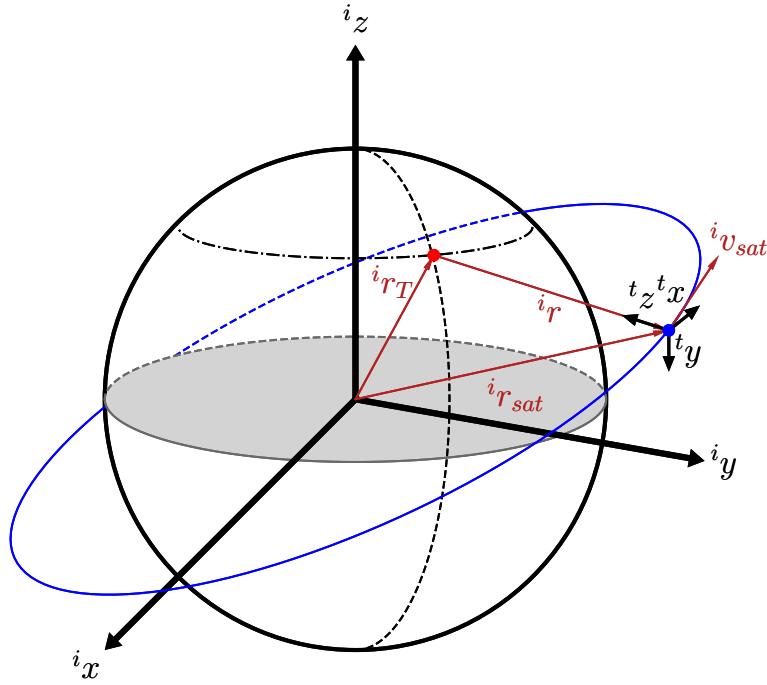


Figure 8.1: Figure of target reference frame.

In order to construct the TRF it is necessary to know the vector from the satellite to the target in the ECI frame. This is done by finding the vector from the center of the earth to the target in the ECEF frame, and then transforming it to the ECI frame. To do this the rotation matrix from ECEF to ECI is needed.

The rotation between ECI and ECEF is done by rotating the ECI frame around the z-axis, which is the same for both ECI and ECEF shown on Figure 8.2.

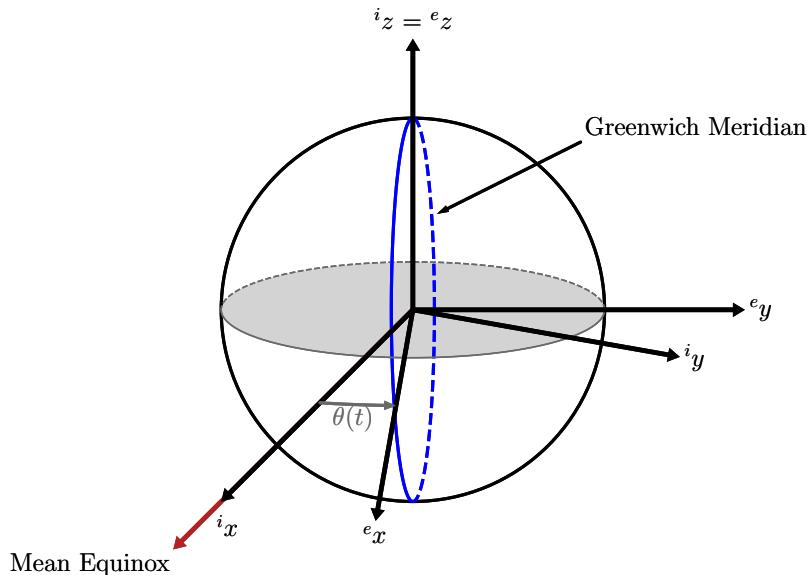


Figure 8.2: Rotation between ECEF and ECI visualized.

This is done with a period of a sidereal day denoted by t_d . A sidereal day being the time it takes the earth to rotate 360° around its own axis, this is about 86164.1 seconds. The rotation matrix for this transformation is shown in Equation 8.6.

$${}^e_i \underline{\mathbf{R}} = \begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) & 0 \\ \sin(\theta(t)) & \cos(\theta(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8.6)$$

here $\theta(t)$ is the angle of rotation at time t . Given by:

$$\theta(t) = \frac{2\pi t}{t_d} \quad (8.7)$$

$\theta(t)$	Angle between the greenwich meridian and the mean equinox.
t	Time since last alignment of ECI and ECEF.
t_d	Period of a sidereal day.

With this a coordinate in the ECEF can be transformed to ECI, only by knowing the time since the last alignment of the two frames.

With this a coordinate in the ECEF can be transformed to ECI, only by knowing the time since the last alignment of the two frames.

Now utilizing the rotation matrix in Equation 8.6 the vector from the center of the earth to the target,in the ECI frame, can be calculated, which is done in Equation 8.8.

$${}^i \underline{\mathbf{r}}_T = {}^e_i \underline{\mathbf{R}} {}^e \underline{\mathbf{r}}_T \quad (8.8)$$

The vector from the center of the earth to the satellite in the ECI frame ${}^i \underline{\mathbf{r}}_s$ is known. The vector from the satellite to the target can be found by subtracting the two vectors, as seen in Equation 8.9.

$${}^i \underline{\mathbf{r}} = {}^i \underline{\mathbf{r}}_s - {}^i \underline{\mathbf{r}}_T \quad (8.9)$$

With vector ${}^i \underline{\mathbf{r}}_s$ it is possible to find the z axis of the TRF, ${}^t \underline{\mathbf{z}}$. This is done by normalizing the vector, as seen in Equation 8.10.

$${}^t \underline{\mathbf{z}} = - \frac{{}^i \underline{\mathbf{r}}_s}{\| {}^i \underline{\mathbf{r}}_s \|} \quad (8.10)$$

To find the vector ${}^t \underline{\mathbf{y}}$ the velocity vector for the satellite in the ECI frame ${}^i \underline{\mathbf{v}}_s$ can be used. Since $\frac{{}^i \underline{\mathbf{v}}_s}{\| {}^i \underline{\mathbf{v}}_s \|} \neq {}^t \underline{\mathbf{x}}$, however ${}^t \underline{\mathbf{x}}$ and ${}^i \underline{\mathbf{v}}_s$ lie in the same plane. Since ${}^t \underline{\mathbf{y}}$ is defined as the orthogonal of ${}^t \underline{\mathbf{z}}$ and ${}^t \underline{\mathbf{x}}$, it can be found by taking the cross product of ${}^t \underline{\mathbf{z}}$ and $\frac{{}^i \underline{\mathbf{v}}_s}{\| {}^i \underline{\mathbf{v}}_s \|}$ described in Equation 8.11.

$${}^t\mathbf{y} = {}^t\mathbf{z} \times \frac{{}^i\mathbf{v}_s}{\|{}^i\mathbf{v}_s\|} \quad (8.11)$$

At last ${}^t\mathbf{x}$ can be calculated using the other two axis as can be seen in Equation 8.12.

$${}^t\mathbf{x} = {}^t\mathbf{y} \times {}^t\mathbf{z} \quad (8.12)$$

Giving a final rotation matrix from ECI to TRF in Equation 8.13.

$${}^t\mathbf{R} = \begin{bmatrix} {}^t\mathbf{y} \times {}^t\mathbf{z} & {}^t\mathbf{z} \times \frac{{}^i\mathbf{v}_s}{\|{}^i\mathbf{v}_s\|} & -\frac{{}^i\mathbf{r}_s}{\|{}^i\mathbf{r}_s\|} \end{bmatrix} \quad (8.13)$$

Since both the TRF and the ORF are based on coordinates in the ECI frame, they are time dependent. This means that they need to be updated continuously as the satellite moves around the Earth.

8.2 Rotational Mechanics

To model the satellite it is important to have knowledge about the relevant quantities regarding rotational mechanics. Rotational mechanics and translational mechanics are two closely related branches of mechanics. Many of the concepts and principles that apply to one also apply to the other. Rotational mechanics is the branch of mechanics that deals with the motion of objects that rotate around a fixed axis. The rotational counterpart to force is torque and the counterpart to momentum is angular momentum. Angular momentum can be described using Equation 8.14.

$$\mathbf{L} = \mathbf{J} \boldsymbol{\omega} \quad (8.14)$$

\mathbf{L}	Angular momentum.
\mathbf{J}	Inertia.
$\boldsymbol{\omega}$	Angular velocity.

Angular momentum is related to translational momentum through the principle of conservation of angular momentum. This principle states that the total angular momentum of a system of objects is conserved as long as there are no external torques acting on the system. This means that if one object in the system gains angular momentum, another object must lose an equal amount of angular momentum. This will be a very important property which will be utilized when using reaction wheels.

Like mass in translational mechanics, moment of inertia determines how much torque is required to produce a given amount of rotational acceleration.

In rotational mechanics the concept of torque is used, which is the rotational equivalent of force. The torque relates to the angular momentum as the time derivative of angular momen-

tum. This can also be seen in Equation 8.15.

$$\tau = \frac{dL}{dt} = J \frac{d\omega}{dt} \quad (8.15)$$

τ | Torque.

Bringing this into a 3-dimensional space will also make the variables 3 three-dimensional vectors and the moment of inertia will become a real symmetric three-dimensional matrix. Thus, the matrix of inertia has three real eigenvalues and three orthogonal eigenvectors, which will satisfy the relation 8.16.

$$\underline{J}\hat{\underline{p}}_i = \lambda_i \hat{\underline{p}}_i \quad \text{for } i = 1, 2, 3 \quad (8.16)$$

\underline{J}	Inertia matrix.
$\hat{\underline{p}}_i$	i 'th eigenvector from the inertia matrix.
λ_i	i 'th eigenvalue from the inertia matrix.

Since the inertia matrix fulfills the relation, it can be concluded that every inertia matrix has an eigenvalue decomposition as in Equation 8.17.

$$\underline{J} = \underline{P} \underline{J}_D \underline{P}^T \quad (8.17)$$

\underline{J}_D	Diagonalized inertia matrix.
\underline{P}	Orthogonal matrix consisting of the eigenvectors of the inertia matrix.

The eigenvalues in the inertia matrix correspond to the principal moments of inertia around the eigenvectors, which corresponds to the principal axes of the CRF.

Throughout the rest of the report, when the inertia matrix appears in an equation, it will be assumed to be in the CRF, which means the matrix will be on the form seen in Equation 8.18.

$$\underline{J} = \begin{bmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3 \end{bmatrix} \quad (8.18)$$

As described in Section 4.2.4 a previous master thesis project have determined the inertia matrix for AAUSAT3 within the CRF to be as in Equation 8.19.

$${}^c\underline{J}_s = \begin{bmatrix} 0.0017 & 0 & 0 \\ 0 & 0.0022 & 0 \\ 0 & 0 & 0.0022 \end{bmatrix} \text{ kg m}^2 \quad (8.19)$$

The rotation from CRF to SRF is given by Equation 8.20.

$${}^c_s \mathbf{q} = \begin{bmatrix} 0.0207 & 0.7147 & 0.0073 & 0.6991 \end{bmatrix}^T \quad (8.20)$$

Since AAUSAT6 is not built yet, the inertia matrix for AAUSAT6 is not known. It is assumed that AAUSAT6 will have the same external size and structure as AAUSAT3. Thus, it is chosen to use the inertia matrix for AAUSAT3 as a good estimate for the inertia matrix for AAUSAT6.

8.3 Kinematic model

The kinematic model is a model of the motion of the satellite without taking into account the forces acting upon it. The goal of the kinematic model is to find an expression of the time derivative of the attitude of the satellite.

As described in Section 4.4.3 the attitude of the satellite is described using quaternions.

If ${}^c \mathbf{q}(t)$ is a quaternion, that describes the attitude in the CRF. Then ${}^c \mathbf{p}(\Delta t)$ is a quaternion that describes the rotation to ${}^c \mathbf{q}(t + \Delta t)$ in the CRF. From this ${}^c \mathbf{q}(t + \Delta t)$ can be expressed as in equation 8.21.

$${}^c \mathbf{q}(t + \Delta t) = {}^c \mathbf{p}(\Delta t) {}^c \mathbf{q}(t) \quad (8.21)$$

Using Euler symmetric parameters, the quaternion ${}^c \mathbf{p}$ can be expressed as in Equation 8.22.

$${}^c \mathbf{p}(\Delta t) = \begin{bmatrix} \hat{\mathbf{e}} \sin\left(\frac{\Delta\theta}{2}\right) \\ \cos\left(\frac{\Delta\theta}{2}\right) \end{bmatrix} \quad (8.22)$$

$\Delta\theta$	Angle of rotation
$\hat{\mathbf{e}}$	Vector which the rotation happens about

Since ${}^c \mathbf{p}$ needs to be multiplied with ${}^c \mathbf{q}$, it is necessary to convert ${}^c \mathbf{p}$ to a matrix. This is done using the matrix notation for the quaternion multiplication described in Section 4.4.3. Therefore, Equation 8.21 can be rewritten as in Equation 8.23.

$${}^c \mathbf{q}(t + \Delta t) = \left[\cos\left(\frac{\Delta\theta}{2}\right) \mathbf{I}_{4 \times 4} + \sin\left(\frac{\Delta\theta}{2}\right) \underline{\mathbf{S}}_4(\hat{\mathbf{e}}) \right] {}^c \mathbf{q}(t) \quad (8.23)$$

$\mathbf{I}_{4 \times 4}$	4×4 identity matrix
$\underline{\mathbf{S}}_4$	Skew-symmetric $\mathbb{R}^3 \rightarrow \mathbb{R}^{4 \times 4}$ matrix operator

$\underline{\mathbf{S}}_4$ is defined as in Equation 8.24.

$$\underline{\mathbf{S}}_4(\boldsymbol{e}) = \begin{bmatrix} 0 & e_3 & -e_2 & e_1 \\ -e_3 & 0 & e_1 & e_2 \\ e_2 & -e_1 & 0 & e_3 \\ -e_1 & -e_2 & -e_3 & 0 \end{bmatrix} \quad (8.24)$$

Since $\Delta\theta$ is equal to the satellite moving with angular velocity ω around the vector $\hat{\mathbf{e}}$ for a time of Δt , $\Delta\theta$ can be written as $\omega\Delta t$.

By assuming that Δt is very small, the small angle approximation can be used. These are the first order Taylor polynomial of cos and sin around 0, given as Equation 8.25.

$$\cos\left(\frac{\Delta\theta}{2}\right) \approx 1 \quad \sin\left(\frac{\Delta\theta}{2}\right) \approx \frac{1}{2}\omega\Delta t \quad (8.25)$$

With this Equation 8.23 can be rewritten as in Equation 8.26.

$${}^c\mathbf{q}(t + \Delta t) \approx \left[\mathbf{I}_{4 \times 4} + \frac{1}{2}\underline{\mathbf{S}}_4(\boldsymbol{\omega})\Delta t \right] {}^c\mathbf{q}(t) \quad (8.26)$$

This can be simplified to Equation 8.27.

$${}^c\mathbf{q}(t + \Delta t) \approx {}^c\mathbf{q}(t) + \frac{1}{2}\underline{\mathbf{S}}_4(\boldsymbol{\omega})\Delta t {}^c\mathbf{q}(t) \quad (8.27)$$

The time derivative of ${}^c\mathbf{q}$ can now be found by taking the time derivative of Equation 8.26. This results in Equation 8.28.

$${}^c\dot{\mathbf{q}}(t) = \frac{d {}^c\mathbf{q}(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{{}^c\mathbf{q}(t + \Delta t) - {}^c\mathbf{q}(t)}{\Delta t} = \frac{1}{2}\underline{\mathbf{S}}_4(\boldsymbol{\omega}) {}^c\mathbf{q}(t) \quad (8.28)$$

Thus Equation 8.28 gives an expression for the time derivative of the attitude of the satellite in the ECI frame.

8.4 Dynamic Model

In this section a dynamic model for the satellite will be found. A dynamic model is a model which takes into account the forces acting upon it. The goal of the dynamic model is to find an expression of the time derivative of the angular velocity of the satellite, leaving out the translational motion. The dynamic equations can be described using Euler's second law of motion. Which states that the rate of change in angular momentum, about a point that is fixed in an inertial reference frame, is equal to the torque applied to the system. From Newton's second law of motion, the time derivative of the angular momentum of the satellite or torque can be found using Equation 8.29.

$${}^i\tau(t) = \frac{d}{dt} {}^iL(t) = {}^i\dot{L}(t) \quad (8.29)$$

$\tau(t)$	Torque acting on the satellite.
$L(t)$	Angular momentum of the satellite.

Since the satellite is controlled in the CRF, the angular momentum and torque must be transformed into the CRF. The ${}^cL(t)$ matrix can be found by multiplying with the rotation matrix ${}^i\underline{R}(t)$, which can be written as Equation 8.30.

$${}^cL(t) = {}^c_i\underline{R}(t) {}^iL(t) \quad (8.30)$$

$\underline{R}(t)$	Rotation matrix.
$L(t)$	Angular momentum of the satellite.

To get the time derivative of the angular momentum in CRF, the time derivatives of both matrices are needed, as they are both time dependent, therefore the product rule applies. This gives the following equation.

$${}^c\dot{L}(t) = {}^c_i\underline{\dot{R}}(t) {}^iL(t) + {}^c_i\underline{R}(t) {}^i\dot{L}(t) \quad (8.31)$$

The time derivative of the rotation matrix \underline{R} can be found by using Equation 8.32 [30].

$${}^c_i\underline{\dot{R}}(t) = -{}^c\omega_s(t) \times {}^c_i\underline{R}(t) \quad (8.32)$$

$\omega_s(t)$	Angle velocity of the satellite.
---------------	----------------------------------

Substituting Equation 8.32 into Equation 8.31 gives the following equation.

$${}^c\dot{L}(t) = -{}^c\omega_s(t) \times {}^c_i\underline{R}(t) {}^iL(t) + {}^c_i\underline{R}(t) {}^i\dot{L}(t) \quad (8.33)$$

Since ${}^i\mathbf{L}(t)$ rotated by ${}^i\underline{\mathbf{R}}(t)$ is equal to ${}^c\mathbf{L}(t)$, Equation 8.33 can be rewritten as Equation 8.34.

$$\begin{aligned} {}^c\dot{\mathbf{L}}(t) &= -{}^c\boldsymbol{\omega}_s(t) \times {}^c\mathbf{L}(t) + {}^c_i\underline{\mathbf{R}}(t) {}^i\boldsymbol{\tau}(t) \\ &= -{}^c\boldsymbol{\omega}_s(t) \times {}^c\mathbf{L}(t) + {}^c\boldsymbol{\tau}(t) \end{aligned} \quad (8.34)$$

Angular momentum of a rigid body can be written as

$${}^c\mathbf{L}(t) = {}^c\underline{\mathbf{J}}_s {}^c\boldsymbol{\omega}_s(t) \quad (8.35)$$

Substituting this into Equation 8.34 gives the following equation.

$$\begin{aligned} {}^c\underline{\mathbf{J}}_s {}^c\dot{\boldsymbol{\omega}}(t) &= -{}^c\boldsymbol{\omega}_s(t) \times {}^c\underline{\mathbf{J}}_s {}^c\boldsymbol{\omega}_s(t) + {}^c\boldsymbol{\tau}(t) \\ {}^c\dot{\boldsymbol{\omega}}(t) &= {}^c\underline{\mathbf{J}}_s^{-1} \left[-{}^c\boldsymbol{\omega}_s(t) \times {}^c\underline{\mathbf{J}}_s {}^c\boldsymbol{\omega}_s(t) + {}^c\boldsymbol{\tau}(t) \right] \end{aligned} \quad (8.36)$$

The torque ${}^c\boldsymbol{\tau}$, which is the total torque acting on the satellite, can be split into the disturbance torque ${}^c\boldsymbol{\tau}_d$ from Section 4.2 and the torque from the reaction wheel ${}^c\boldsymbol{\tau}_w$. This gives the following equation.

$${}^c\dot{\boldsymbol{\omega}}(t) = {}^c\underline{\mathbf{J}}_s^{-1} \left[-\underline{\mathbf{S}}_3({}^c\boldsymbol{\omega}_s(t)) {}^c\underline{\mathbf{J}}_s {}^c\boldsymbol{\omega}_s(t) + {}^c\boldsymbol{\tau}_d(t) + {}^c\boldsymbol{\tau}_w(t) \right] \quad (8.37)$$

Where $\underline{\mathbf{S}}_3$ is the skew-symmetric $\mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$ matrix operator given by Equation 8.38.

$$\underline{\mathbf{S}}_3(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (8.38)$$

When the rotation happens about a non-principal axis, the cross product of the angular velocity and the angular momentum is non-zero. This will result in nutation. This means that the angular momentum of the satellite will be non-zero even if no external torque is applied to the satellite.

It can be seen that Equation 8.37 is nonlinear due to the cross product of the angular velocity and the angular momentum. To be able to use this equation to control the satellite, it will have to be linearized.

8.5 Reaction wheels using DC motors

It is chosen to use DC motors to spin the reaction wheels. The DC motors are chosen because they are easy to control and because they are relatively easy to model. The DC motors used are the same as the ones used on the testbed. Because it is assumed that the ones used on

AAUSAT6 will be similar to these.

During the design of the reaction wheels, it is assumed that the DC motors have encoders. Which is easily implemented in simulation.

The DC motor parameters can be seen in Table 8.1.

Parameter	Symbol	Value	Unit
Armature resistance	R_a	4.44	Ω
Armature inductance	L_a	0.12e - 3	H
Motor constant / generator constant	K_e, K_t	1.81e - 3	N m/A
Motor friction	b	63.9e - 9	N ms
Combined moment of inertia	J_w	0.336e - 6	kg m ²
Mechanical time constant without reaction wheels	τ_{mech}	32.10e - 3	s

Table 8.1: DC motor parameters [31].

To model the DC motor and reaction wheel, a free body diagram is made. The free body diagram is shown in Figure 8.3.

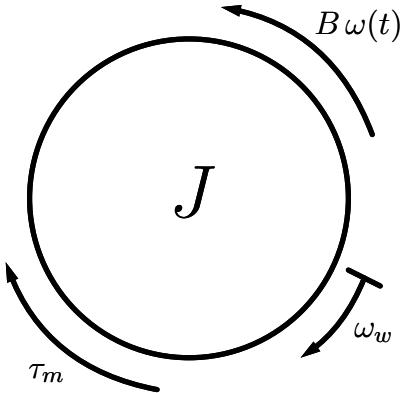


Figure 8.3: Free body diagram of the reaction wheel

From this free body diagram, Equation 8.39 can be derived.

$$J\dot{\omega}_w = \tau_m - B\omega_w \quad (8.39)$$

J	Moment of inertia of the reaction wheel and the motor
$\dot{\omega}_w$	Angular acceleration of the reaction wheel
τ_m	Torque from the motor
B	Damping coefficient of the reaction wheel and motor

The motor torque is given by Equation 8.40.

$$\tau_m = K_t i_a \quad (8.40)$$

K_t	Generator constant
i_a	Armature current

To model the correlation between the armature current the electrical model of the motor is used. The electrical model of the motor is shown in Figure 8.4.

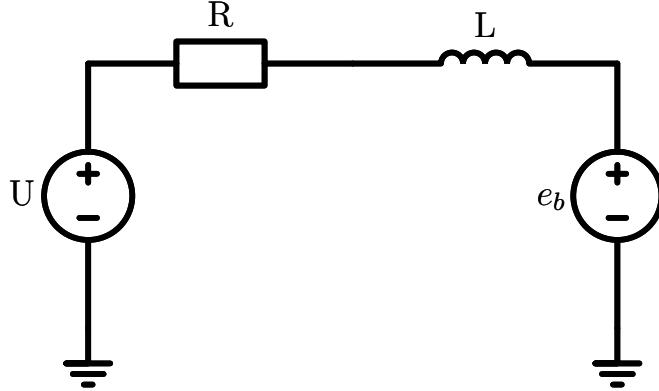


Figure 8.4: Electrical model of the DC motor

Using Kirchhoff's voltage law, Equation 8.41 can be derived.

$$V_a = R_a i_a + L_a \frac{di_a}{dt} + e_b \quad (8.41)$$

V_a	Armature voltage
R_a	Armature resistance
L_a	Armature inductance
e_b	Back EMF

The back EMF is given by $e_b = K_e \omega_w$. From the data sheet of the DC motor [31], the mechanical time constant is 32.1 ms, which will be higher when the reaction wheel is mounted.

The electrical time constant is given by $L_a / R_a = 0.0027$ ms. Since the mechanical time constant without the reaction wheels is already much larger than the electrical time constant, the motor can be considered as a first order system.

Assuming that the motor is a first order system, Equation 8.41 can be rewritten as Equa-

tion 8.42.

$$\begin{aligned} V_a &= R_a i_a + K_e \omega_w \\ i_a &= \frac{V_a - K_e \omega_w}{R_a} \end{aligned} \quad (8.42)$$

Combining Equation 8.42 and Equation 8.39 gives Equation 8.44.

$$J\dot{\omega}_w = K_t \frac{V_a - K_e \omega_w}{R_a} - B\omega_w \quad (8.43)$$

$$= \frac{K_t}{R_a} V_a - \left(\frac{K_t K_e}{R_a} + B \right) \omega_w \quad (8.44)$$

This equation is used to model one DC motor. To model how the DC motors influence the satellite it is necessary to know the configuration of the reaction wheels.

8.5.1 Reaction wheel configuration

There are several ways to configure the reaction wheels. Placing one reaction wheel in the direction of each of the principal axis of the CRF is the simplest configuration. This configuration is shown in Figure 8.5.

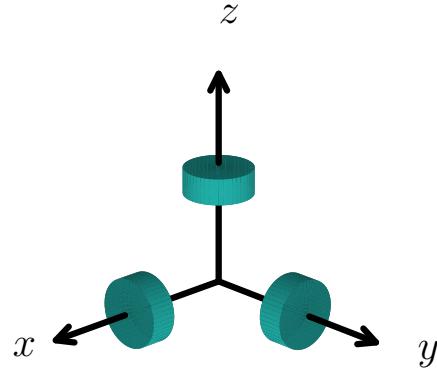


Figure 8.5: Reaction wheel configuration 1

The reaction wheel configuration has the advantage that it is simple to implement. The disadvantage is that the DC motors are non-linear near the zero speed. To get a linear response, the DC motors must work at a bias point. A sketch of the non-linearity can be seen on Figure 8.6.

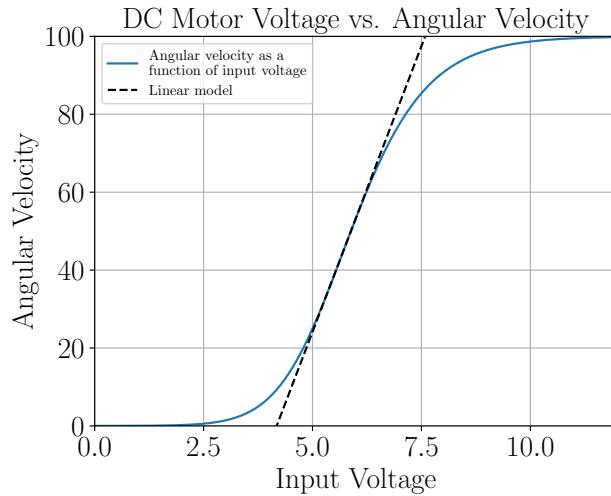


Figure 8.6: Speed vs. applied voltage for a DC motor

This means that the DC motors must work around a speed that is non-zero. In this configuration it is not possible to spin up the DC motors to get them to the bias point without spinning the satellite.

Another configuration is to place 4 reaction wheels in a tetrahedron configuration. This configuration is shown in Figure 8.7.

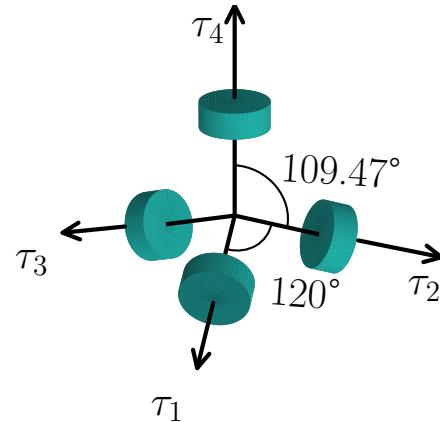


Figure 8.7: Reaction wheel configuration 2

The advantage of this configuration is that it is possible to spin up the DC motors to get them to the bias point. The disadvantage is that it is more complicated to implement.

Because of the ability to get the DC motors to a bias point it is chosen to use the tetrahedron configuration. The torques from the reaction wheels expressed in the CRF can be written as Equation 8.47.

$${}^c\tau_{w,x} = \tau_1 \cos(\beta) - \tau_2 \cos(\beta) \cos(\alpha) - \tau_3 \cos(\beta) \cos(\alpha) \quad (8.45)$$

$${}^c\tau_{w,y} = \tau_2 \cos(\beta) \cos\left(\frac{\alpha}{2}\right) - \tau_3 \cos(\beta) \cos\left(\frac{\alpha}{2}\right) \quad (8.46)$$

$${}^c\tau_{w,z} = -\tau_1 \sin(\beta) - \tau_2 \sin(\beta) - \tau_3 \sin(\beta) + \tau_4 \quad (8.47)$$

Where α is half the angle between τ_1 , τ_2 and τ_3 meaning $\alpha = 120^\circ/2 = 60^\circ$ and β is the angle between the same reaction wheels and the xy plane, giving $\beta = 109.47^\circ - 90^\circ = 19.47^\circ$

This can be rewritten in vector form as Equation 8.48.

$$\begin{bmatrix} {}^c\tau_{w,x} \\ {}^c\tau_{w,y} \\ {}^c\tau_{w,z} \end{bmatrix} = \underline{\mathbf{P}} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} \quad (8.48)$$

Where $\underline{\mathbf{P}}$ is the 3×4 distribution matrix given by Equation 8.49 [32].

$$\underline{\mathbf{P}} = \begin{bmatrix} \cos(\beta) & -\cos(\beta) \cos(\alpha) & -\cos(\beta) \cos(\alpha) & 0 \\ 0 & \cos(\beta) \cos\left(\frac{\alpha}{2}\right) & -\cos(\beta) \cos\left(\frac{\alpha}{2}\right) & 0 \\ -\sin(\beta) & -\sin(\beta) & -\sin(\beta) & 1 \end{bmatrix} \quad (8.49)$$

Since the angles β and α are known to be $\beta = 19.47^\circ$ and $\alpha = 60^\circ$, from Figure 8.7, the distribution matrix is given by Equation 8.50.

$$\underline{\mathbf{P}} = \begin{bmatrix} 0.94 & -0.47 & -0.47 & 0 \\ 0 & 0.82 & -0.82 & 0 \\ -0.33 & -0.33 & -0.33 & 1 \end{bmatrix} \quad (8.50)$$

From $\underline{\mathbf{P}}$ it is evident that if $\tau_1 = \tau_2 = \tau_3 = \tau_4$ then the torques from the reaction wheels will cancel each other out since the rows sum to zero. This means that it is possible to spin up the reaction wheels to their bias point without spinning the satellite.

Since the motor is known it is possible to calculate the necessary bias point. The bias point needs to be a compromise since the motors are both non-linear at zero speed and at high speed. At zero speed the non-linearity is due to the static friction. At high speed the non-linearity is due to saturation.

To find the appropriate bias point it is necessary to find the maximum speed the motor will reach during the maneuver. The speed of the reaction wheels can be described using Equation 8.51.

$$\omega_w = \int_{t_0}^{t_1} \alpha_w dt \quad (8.51)$$

t_0	Time when the maneuver starts
t_1	Time when the maneuver ends
α_w	Angular acceleration of the reaction wheel

The acceleration of the reaction wheel is given by Equation 8.52.

$$\alpha_w = \frac{\tau_s}{J_w} \quad (8.52)$$

τ_s	Required torque on the satellite to do the maneuver
J_w	Moment of inertia of the reaction wheel

Since the torque produced by the reaction wheels and the torque acting on the satellite are equal. The required torque on the satellite can be described using Equation 8.53.

$$\tau_w = \alpha_s J_s \quad (8.53)$$

α_s	Angular acceleration of the satellite
J_s	Moment of inertia of the satellite

The required angular acceleration of the satellite can be found as the time derivative of the rotational speed required to do the maneuver.

To find the requirements for the reaction wheels it is necessary to rewrite Equation 8.48 as Equation 8.54. Since $\underline{\mathbf{P}}$ is not square, there is infinite solutions to the equation. The Moore–Penrose-inverse of $\underline{\mathbf{P}}$ is used to find the solution with the smallest euclidian norm.

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \underline{\mathbf{P}}^+ \begin{bmatrix} {}^c\tau_{w,x} \\ {}^c\tau_{w,y} \\ {}^c\tau_{w,z} \end{bmatrix} \quad (8.54)$$

The pseudo inverse is given by Equation 8.55.

$$\underline{\mathbf{P}}^+ = \underline{\mathbf{P}}^T (\underline{\mathbf{P}} \underline{\mathbf{P}}^T)^{-1} = \begin{bmatrix} 0.71 & 0 & -0.25 \\ -0.35 & 0.61 & -0.25 \\ -0.35 & 0.61 & -0.25 \\ 0 & 0 & 0.75 \end{bmatrix} \quad (8.55)$$

It is not yet known which of the axes the satellite will rotate around. Because the cameras position is yet to be determined. Therefore, the worst case scenario is used, meaning the rotation around a singular axis with the largest single motor torque. It can be seen in Equation 8.55 that the largest single motor torque is τ_4 in the case of a rotation around the z -axis. Therefore, this case is graphed in Figure 8.8 with the required torque, required acceleration and rotational speed of the reaction wheels is shown, when performing the pointing manoeuvre.

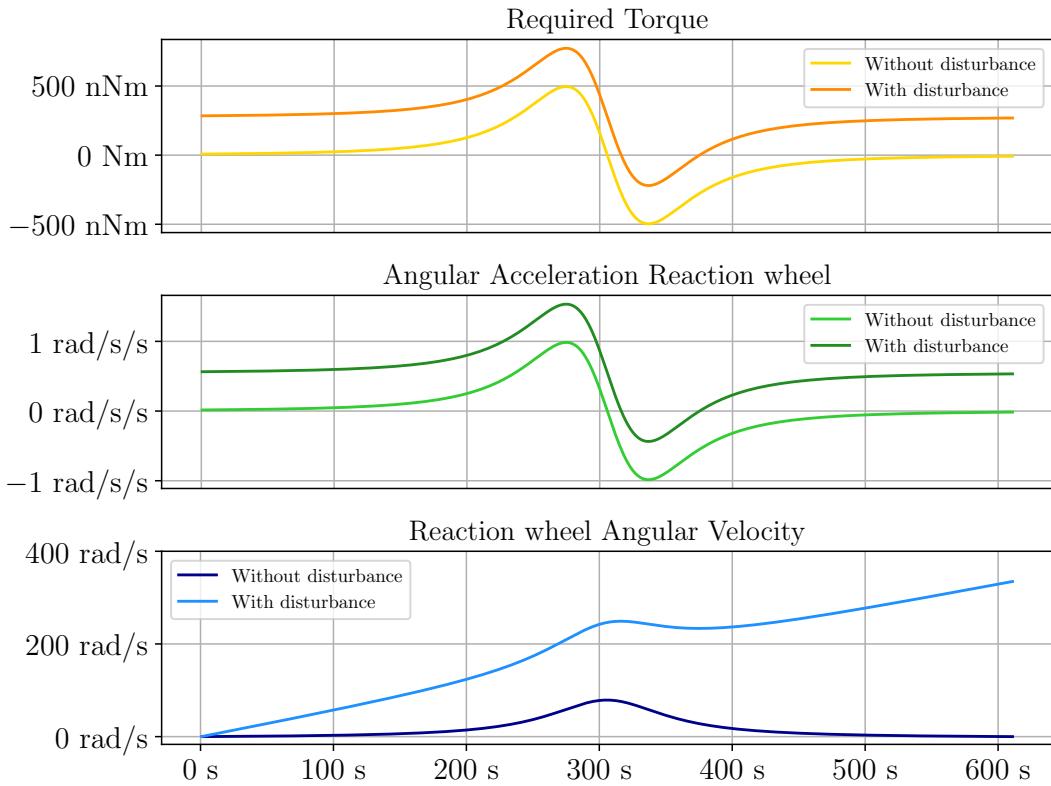


Figure 8.8: Required torque, required acceleration and rotational speed of the reaction wheels.

From this graph it is determined that the bias angular velocity of the reaction wheels should be around 400 rad/s but to be safe it is chosen to be double that, which gives 800 rad/s = 3821 rpm.

8.5.2 Dynamic model of the the DC motors

The dynamic equations of the reaction wheels in the satellite can be described the same way as the dynamic model for the satellite.

$$\begin{aligned} {}^i\boldsymbol{\tau}_{ext} &= \frac{d}{dt} {}^i\mathbf{L}_w(t) \\ &= {}^i\dot{\mathbf{L}}_w(t) \end{aligned} \quad (8.56)$$

This change in angular momentum of a single reaction wheel can be described using a rotation matrix from ECI to a single reaction wheel reference frame (RWRF) given as ${}^w_i\mathbf{R}$. As the rotation matrix changes over time.

$${}^w \dot{\underline{L}}_w(t) = \frac{d}{dt} \left({}^w_i \underline{\mathbf{R}}(t) {}^i \underline{L}_w(t) \right) \quad (8.57)$$

$$= {}^w_i \dot{\underline{\mathbf{R}}}(t) {}^i \underline{L}_w(t) + {}^w_i \underline{\mathbf{R}}(t) {}^i \dot{\underline{L}}_w(t) \quad (8.58)$$

Where the derivative of the rotation matrix can be rewritten as the skew symmetric matrix of the angular velocity of the reaction wheel multiplied by the rotation matrix as in Equation 8.59.

$${}^w_i \dot{\underline{\mathbf{R}}}(t) = -\underline{\mathbf{S}}_3({}^w \boldsymbol{\omega}(t)) {}^w_i \underline{\mathbf{R}}(t) \quad (8.59)$$

Inserting this into Equation 8.58 gives Equation 8.60.

$$\begin{aligned} {}^w \dot{\underline{L}}_w(t) &= -\underline{\mathbf{S}}_3({}^w \boldsymbol{\omega}(t)) {}^w_i \underline{\mathbf{R}}(t) {}^i \underline{L}_w(t) + {}^w_i \underline{\mathbf{R}}(t) {}^i \dot{\underline{L}}_w(t) \\ &= -\underline{\mathbf{S}}_3({}^w \boldsymbol{\omega}(t)) {}^w \underline{L}_w(t) + {}^w \boldsymbol{\tau}_{ext}(t) \end{aligned} \quad (8.60)$$

Where ${}^w \boldsymbol{\tau}_{ext}$ is the torque generated by the reaction wheels. And the other part of the equation is the Coriolis force [33].

Where ${}^w \boldsymbol{\omega}$ is the sum of the angular velocities of the reaction wheel and the satellite, see Equation 8.61.

$${}^w \boldsymbol{\omega}(t) = {}^w \boldsymbol{\omega}_s(t) + {}^w \boldsymbol{\omega}_w(t) \quad (8.61)$$

Where ${}^w \boldsymbol{\omega}_s(t) = [{}^w \boldsymbol{\omega}_{sx}(t) \ {}^w \boldsymbol{\omega}_{sy}(t) \ {}^w \boldsymbol{\omega}_{sz}(t)]^T$ and ${}^w \boldsymbol{\omega}_w(t) = [0 \ 0 \ {}^w \boldsymbol{\omega}_{wz}(t)]^T$.

Translating this CRF using the rotation matrix ${}^c_w \underline{\mathbf{R}}$ gives Equation 8.62. Since this rotation matrix does not change over time, the skew matrix is not required here.

$${}^c \dot{\underline{L}}_w(t) = -{}^c_w \underline{\mathbf{R}} {}^w \dot{\underline{L}}_w(t) \quad (8.62)$$

The sign change is due to Newtons third law of motion, which states that for every action there is an equal and opposite reaction, for a rigid body. This also applies for rotational bodies.

Since there are four reaction wheels Equation 8.62 can be rewritten as Equation 8.63. It should be noted that there is a unique rotation matrix for each reaction wheel.

$${}^c \boldsymbol{\tau}_w(t) = -\sum_{i=1}^4 {}^c_w \underline{\mathbf{R}}_i {}^w \dot{\underline{L}}_{w,i}(t) \quad (8.63)$$

This model shows that the reaction wheel model can be transformed using the P-matrix. This is because the rotation matrices for each reaction wheel is constant.

This dynamic model can be added to the model found in Section 8.4.

8.6 Linear State Space Model

To design a controller for the satellite, a model of the satellite is needed. The model of the satellite is derived from the rotational mechanics, kinematic model and dynamic model of the satellite. A linear state space model is chosen to describe the system, since it is a multi-input multi-output system.

8.6.1 State Space Model

When modelling a system with multiple inputs and outputs, traditional control theory is not sufficient. Since the system has multiple inputs and outputs, it is decided to use a state space model to describe the system.

State space models are a way of describing a dynamic system by determining a set of state variables, which describe the current “state” of the system, along with a set of input and output variables. The conditions for the input and output variables are that they must be measurable, and that they must be related to the state variables.

How the these variables are related is described by a set of first order differential equations, which are called the state equations. [34].

The state equations are described in Equation 8.64 and 8.65.

$$\dot{\underline{x}}(t) = \underline{A}\underline{x}(t) + \underline{B}\underline{u}(t) \quad (8.64)$$

$$\underline{y}(t) = \underline{C}\underline{x}(t) + \underline{D}\underline{u}(t) \quad (8.65)$$

\underline{x}	The state vector
\underline{u}	The input vector
\underline{y}	The output vector
\underline{A}	The state matrix
\underline{B}	The input matrix
\underline{C}	The output matrix
\underline{D}	The feedforward matrix

The block diagram of a linear state space system can be seen in Figure 8.9.

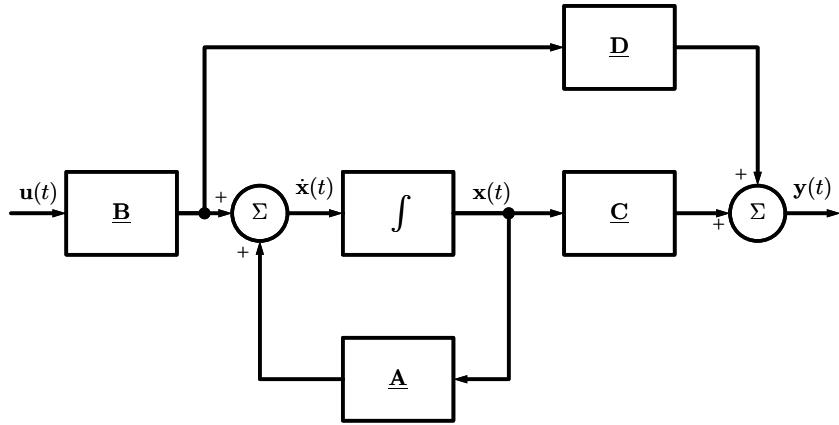


Figure 8.9: Block diagram of the system

The block diagram with the motor loop as input to the state space model can be seen in Figure 8.10.

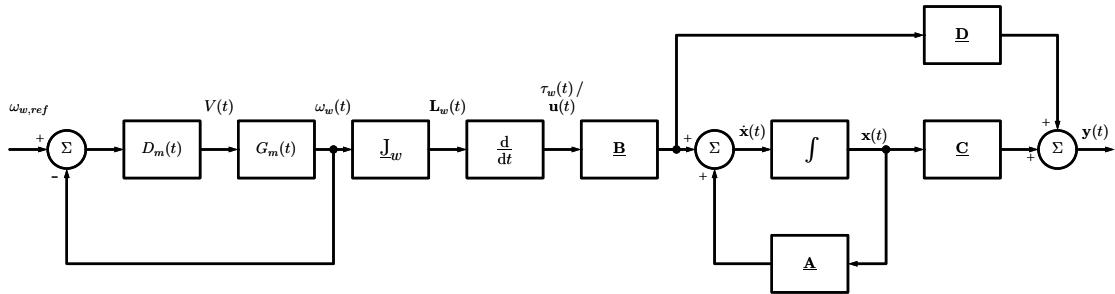


Figure 8.10: Block diagram of the system with motor loop.

It is possible to avoid including the motor dynamics in the state space model by choosing gains for the satellite attitude controller, such that the poles are 10-15 times lower than the bandwidth of the motor controller [35]. This means, that the motor controller will be 10-15 times faster than the satellite attitude controller and therefore the torque of the motor loop will, from the outer loops point of view, seem “instantaneous”. With this a controller can be designed for the motor loop.

8.6.2 DC Motor Controller

As shown above, the motor will be controlled in a cascaded control loop as seen in Figure 8.10.

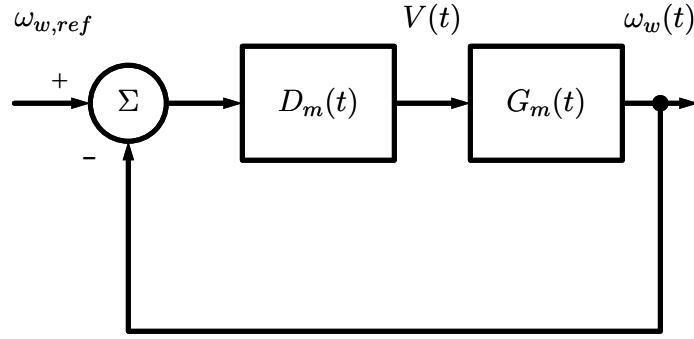


Figure 8.11: Motorloop

Furthermore, the motor will be controlled using a PI controller. The PI controller is chosen because it is a simple controller that is easy to implement and tune, while eliminating potentially significant steady state error.

8.6.2.1 DC-Motor transfer function

Rewriting the motor model from Equation 8.44 gives the following transfer function.

$$G_m(s) = \frac{\omega_w}{V_a} = \frac{K_t}{R_a B + K_e K_t + J_w R_a s} \quad (8.66)$$

Inserting the values from Table 8.1 gives the plant model in Equation 8.67.

$$G_m(s) = \frac{508.5}{0.4191 s + 1} \quad (8.67)$$

8.6.2.2 Controller

A PI controller consist of a proportional and an integral part. With a transfer function given as Equation 8.68.

$$C_m(s) = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s} \quad (8.68)$$

Combining this with the plant model in Equation 8.67 gives the closed loop transfer function in Equation 8.69.

$$T(s) = \frac{C_m(s)G_m(s)}{1 + C_m(s) + G_m(s)} = \frac{\frac{508.5(K_p s + K_i)}{(0.4191 s + 1)s}}{1 + \frac{508.5(K_p s + K_i)}{(0.4191 s + 1)s}} = \frac{\frac{508.5(K_p s + K_i)}{0.4191}}{s^2 + \frac{508.5 K_p + 1}{0.4191} s + \frac{508.5 K_i}{0.4191}} \quad (8.69)$$

It can be seen that the closed loop system is second order with a zero. While the system is not on the canonical form, shown in Equation 8.70, of a second order system, it is treated as such in the following analysis. Therefore, the initial design will not reflect the final design, but it will be a good starting point for the tuning of the controller.

$$C_{canon}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (8.70)$$

The smart thing about using the canonical form is that the poles of the system can easily be placed, using the requirements of the system. Because there is a direct correlation between the poles and the behaviour of the system. These correlations are shown in Equations 8.71, 8.72 and 8.73 [36].

$$T_r \approx \frac{1.8}{\omega_n} \quad (8.71)$$

$$T_s(x) = -\frac{\ln(x)}{\zeta\omega_n} \Rightarrow T_s(0.01) = \frac{4.6}{\zeta\omega_n} \quad (8.72)$$

$$M_p = e^{-\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}} \quad (8.73)$$

T_r	Rise time of the motor control system
T_s	Settling time of the motor control system
M_p	Overshoot
x	The desired settling margin

Using the requirements from the system and making them “tighter” by 15 times, gives the following requirements.

$$T_s < t_s/15 < 8.8 \quad (8.74)$$

$$M_p < PO/15 < 0.0141 \quad (8.75)$$

t_s	Settling time of the system
PO	Overshoot of the system

The maximum overshoot when changing from nadir mode to pointing mode is 14.9° as per

requirement 5.1.4, and the step size can be calculated as:

$$\theta_{step} = \arcsin\left(\frac{r_e}{r_s}\right) = \arcsin\left(\frac{6371}{6768}\right) = 1.22 \text{ rad} \approx 70.3^\circ \quad (8.76)$$

The percent overshoot PO is therefore calculated as:

$$PO = \frac{14.9}{70.3} \cdot 100\% = 21.2\% \quad (8.77)$$

The system does not have any requirements for the rise time, so it is not included in the design.

Combining Equation 8.69 and Equation 8.70 gives the following equations, where the difference of the two, being the zero, is ignored.

$$2\zeta\omega_n = 2\sigma = \frac{508.5K_p + 1}{0.4191} \quad (8.78)$$

$$\omega_n^2 = \frac{508.5K_i}{0.4191} \quad (8.79)$$

With these equations its possible to find the dampening coefficient ζ from Equation 8.73.

$$\zeta > -\ln(M_p) \sqrt{\frac{1}{\pi^2 + \ln^2(M_p)}} = 0.8048 \quad (8.80)$$

$$T_s = \frac{4.6}{\sigma} < 8.8 \quad (8.81)$$

\Downarrow

$$\sigma > \frac{4.6}{8.8} = 0.5227 \quad (8.82)$$

The pole is chosen to be at -4.6, meaning σ is 4.6. This is more than required and making the pole any larger than -4.6, can make the system unstable, since it might make the motors reach their saturation voltage.

There is no need for a complex pole, since the system does not have a rise time requirement.

Inserting σ into Equation 8.78 gives Equation 8.83.

$$\begin{aligned} 2 \cdot \sigma &= \frac{508.5K_p + 1}{0.4191} \\ &\Downarrow \\ K_p &= \frac{9.2 \cdot 0.4191 - 1}{508.5} = 0.005619 \end{aligned} \quad (8.83)$$

Furthermore, with σ its possible to find ω_n .

$$\begin{aligned}\sigma &= \zeta\omega_n = 4.6 \\ \Downarrow \\ \omega_n &= \frac{4.6}{\zeta} = \frac{4.6}{0.8048} = 5.716\end{aligned}\tag{8.84}$$

Inserting ω_n into Equation 8.79 gives Equation 8.85.

$$\begin{aligned}\omega_n^2 &= \frac{508.5 K_i}{0.4191} \\ \Downarrow \\ K_i &= \frac{5.716^2 \cdot 0.4191}{508.5} = 0.02694\end{aligned}\tag{8.85}$$

Now the K_p and K_i values have been determined and the controller can be simulated. A simulated response can be seen in Figure 8.12.

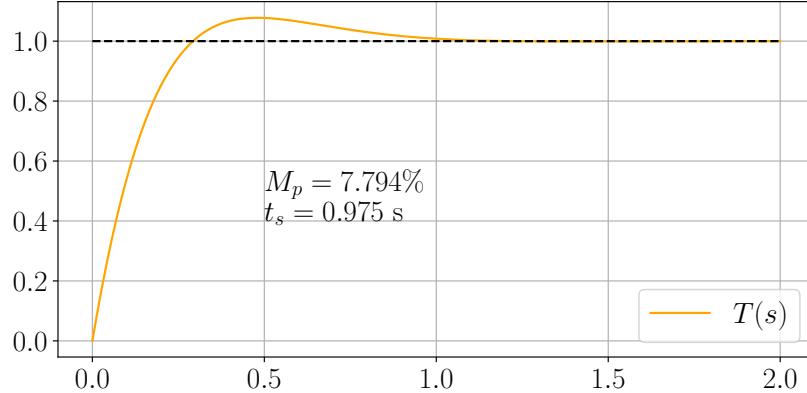


Figure 8.12: Simulated response of the controller before adjustments.

As seen on the figure the overshoot does not meet the requirement of less than 1.41%. To solve this the K_i parameter can be decreased, on Figure 8.13 the K_i parameter has been decreased by a factor of 2.

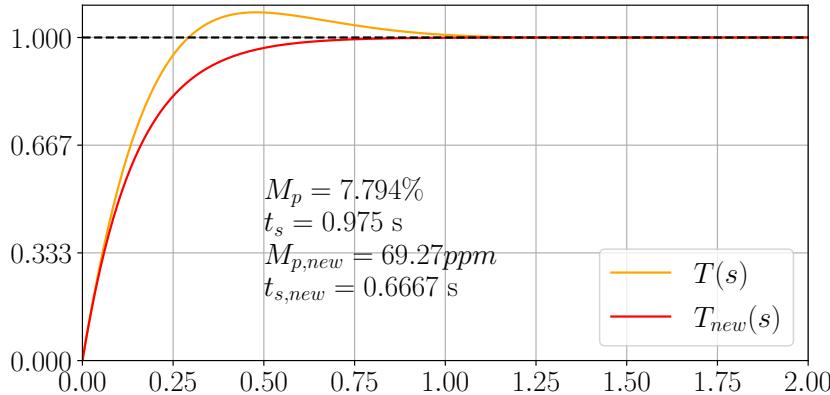


Figure 8.13: Simulated response of the controller after adjustments.

This system is well within the specifications and the settling time has also been reduced. Therefore, no further tuning is required. Giving a final transfer function for the controller as in Equation 8.86.

$$C_m(s) = \frac{0.005619 \cdot s + 0.01347}{s} = 0.005619 + \frac{0.01347}{s} \quad (8.86)$$

8.6.2.3 Controller sampling time

As it can be seen on Figure 8.13 the settling time of the reaction wheels are $t_s = 0.667$ s, which is the shortest settling time of the system. It is therefore chosen that the sampling time of the outerloop should be $T_s = 1$ s. Since the inner loop needs to be sampling at least 10 times faster than the outer loop, as mentioned in 8.6.1 the sampling time of the inner loop is chosen to be $\frac{1}{10}$ s.

8.6.2.4 Discretization

Since the controller is to be implemented on a microcontroller, it must be discretized. This is done using zero order hold (ZOH). ZOH is done with the method seen in Equation 8.87 [37].

$$H(z) = (1 - z^{-1}) \mathcal{Z} \left(\mathcal{L}^{-1} \left\{ \frac{H(s)}{s} \right\} \right) \quad (8.87)$$

$H(s)$	Transfer function in the s-domain
$H(z)$	Transfer function in the z-domain

The discretized transfer function is shown in Equation 8.88.

$$\begin{aligned}
 C_m(z) &= (1 - z^{-1}) \mathcal{Z} \left(\mathcal{L}^{-1} \left\{ \frac{0.005619 + \frac{0.01347}{s}}{s} \right\} \right) \\
 &= (1 - z^{-1}) \mathcal{Z} (0.005619 + 0.01347t)_{t=nT_s} \\
 &= (1 - z^{-1}) \mathcal{Z} (0.005619 + 0.01347 n T_s) \\
 &= (1 - z^{-1}) \mathcal{Z} (0.005619 + 0.001347 n) \\
 &= (1 - z^{-1}) \left(\frac{0.005619z}{z-1} + \frac{0.001347z}{(z-1)^2} \right) \\
 &= \frac{0.005619z - 0.004272}{z-1}
 \end{aligned} \tag{8.88}$$

With this the discrete controller can be implemented on a microcontroller. With the inner-loop controller ready to be implemented the State Space model can be designed.

8.6.3 State Vector

The state vector holds the states of the system. The state vector is defined as:

$$\boldsymbol{x} = \begin{bmatrix} {}^c \mathbf{q}_s \\ {}^c \boldsymbol{\omega}_s \\ {}^c \mathbf{L}_s \end{bmatrix} \tag{8.89}$$

${}^c \mathbf{q}_s$	Attitude of the satellite
${}^c \boldsymbol{\omega}_s$	Angular velocity of the satellite
${}^c \mathbf{L}_s$	Angular momentum of the satellite

The angular momentum of the satellite, ${}^c \mathbf{L}_s$, can be split up into the angular momentum contribution to the satellite from the disturbance torques, ${}^c \mathbf{L}_d$, and the angular momentum of the reaction wheels, ${}^c \mathbf{L}_w$.

$${}^c \mathbf{L}_s = {}^c \mathbf{L}_d + {}^c \mathbf{L}_w \tag{8.90}$$

The angular momentum of the disturbance torques, ${}^c \mathbf{L}_d$, is assumed to be constant, and can therefore be excluded from the state model. The angular momentum of the satellite can therefore be approximated as:

$${}^c \mathbf{L}_s \approx {}^c \mathbf{L}_w \tag{8.91}$$

As described in Section 8.6.1, the motor will be controlled in a cascaded control loop as seen in Figure 8.10. Assuming the inner loop is at least 10 times faster than the outer loop, the motor can be assumed to be at a steady state [35]. This allows the motor model to be removed from the state space model.

The state vector can therefore be reduced to:

$$\mathbf{x} = \begin{bmatrix} {}^c\dot{\mathbf{q}}_s \\ {}^c\dot{\boldsymbol{\omega}}_s \end{bmatrix} \quad (8.92)$$

Taking Figure 8.10 into account, the output $\mathbf{y}(t)$ represents the observable states of the system. However, the contents of the state vector $\mathbf{x}(t)$, are directly observable. This results in the output matrix $\underline{\mathbf{C}}$ being the identity matrix, and the output vector $\mathbf{y}(t)$ being equal to the state vector $\mathbf{x}(t)$.

In addition, the feedforward matrix $\underline{\mathbf{D}}$ is zero, as there is no direct feedthrough from the input $\mathbf{u}(t)$ to the output $\mathbf{y}(t)$.

The second term in the state space model Equation 8.65, can therefore be removed, represented in the block diagram in Figure 8.14.

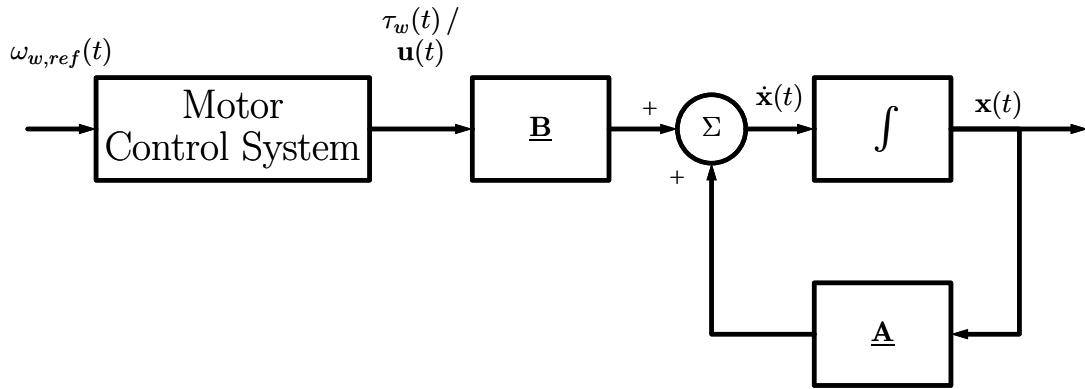


Figure 8.14: Block diagram of the system with motor loop.

8.6.4 Derived State Model

To determine $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ in the state space model, Equation 8.64 states that the derivative of the state vector must be known.

$$\dot{\mathbf{x}} = \begin{bmatrix} {}^c\dot{\mathbf{q}}_s \\ {}^c\dot{\boldsymbol{\omega}}_s \end{bmatrix} \quad (8.93)$$

From the kinematic equations of the satellite in Section 8.3, the derivative of the satellite attitude is given by:

$${}^c\dot{\mathbf{q}}_s = \frac{1}{2}\underline{\mathbf{S}}_4({}^c\boldsymbol{\omega}_s){}^c\mathbf{q}_s \quad (8.94)$$

And from the dynamic equations of the satellite in Section 8.4, the derivative of the satellite angular velocity is given by:

$${}^c\dot{\boldsymbol{\omega}}_s = \underline{\mathbf{J}}_s^{-1} \left[-\underline{\mathbf{S}}_3({}^c\boldsymbol{\omega}_s)\underline{\mathbf{J}}_s {}^c\boldsymbol{\omega}_s + \boldsymbol{\tau}_w + \boldsymbol{\tau}_d \right] \quad (8.95)$$

The derivative of the state vector can then be written as:

$$\begin{bmatrix} {}^c\dot{\mathbf{q}}_s \\ {}^c\dot{\boldsymbol{\omega}}_s \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\underline{\mathbf{S}}_4({}^c\boldsymbol{\omega}_s){}^c\mathbf{q}_s \\ \underline{\mathbf{J}}_s^{-1} \left[-\underline{\mathbf{S}}_3({}^c\boldsymbol{\omega}_s)\underline{\mathbf{J}}_s {}^c\boldsymbol{\omega}_s + \boldsymbol{\tau}_w + \boldsymbol{\tau}_d \right] \end{bmatrix} \quad (8.96)$$

As can be seen in Equation 8.96 the derivative of the state-vector is non-linear. In order to get a Linear state space representation of this system, the state vector must be linearised.

8.6.4.1 Linearised kinematic model

The kinematic model is found in Section 8.3 to be:

$${}^c\dot{\mathbf{q}}_s = \frac{1}{2}\underline{\mathbf{S}}_4({}^c\boldsymbol{\omega}_s){}^c\mathbf{q}_s \quad (8.97)$$

The kinematic model is linearised using Taylor approximation in the operating point ${}^c\bar{\boldsymbol{\omega}}$ for the angular velocity, and ${}^c\bar{\mathbf{q}}_s$ for the attitude. The angular velocity ${}^c\boldsymbol{\omega}_s$ is then rewritten as ${}^c\bar{\boldsymbol{\omega}} + {}^c\tilde{\boldsymbol{\omega}}_s$, and the attitude ${}^c\mathbf{q}_s$ is rewritten as ${}^c\bar{\mathbf{q}}_s + {}^c\tilde{\mathbf{q}}_s$. The kinematic model is then rewritten as:

$${}^c\dot{\mathbf{q}}_s ({}^c\tilde{\boldsymbol{\omega}}_s, {}^c\tilde{\mathbf{q}}_s) = \frac{1}{2}\underline{\mathbf{S}}_4({}^c\bar{\boldsymbol{\omega}}_s + {}^c\tilde{\boldsymbol{\omega}}_s)({}^c\bar{\mathbf{q}}_s + {}^c\tilde{\mathbf{q}}_s) \quad (8.98)$$

$$(8.99)$$

${}^c\bar{\boldsymbol{\omega}}_s$	Operating point of satellite angular velocity
${}^c\tilde{\boldsymbol{\omega}}_s$	Deviation from operating point of satellite angular velocity
${}^c\bar{\mathbf{q}}_s$	Operating point of satellite attitude
${}^c\tilde{\mathbf{q}}_s$	Deviation from operating point of satellite attitude

The first order Taylor approximation of ${}^c\dot{\mathbf{q}}_s ({}^c\boldsymbol{\omega}_s, {}^c\mathbf{q}_s)$ is then found as:

$${}^c\dot{\mathbf{q}}_s ({}^c\tilde{\boldsymbol{\omega}}_s, {}^c\tilde{\mathbf{q}}_s) \approx \underline{\mathbf{S}}_4 {}^c(\bar{\boldsymbol{\omega}}_s) {}^c\bar{\mathbf{q}}_s + \frac{\partial {}^c\dot{\mathbf{q}}_s ({}^c\bar{\boldsymbol{\omega}}_s, {}^c\bar{\mathbf{q}}_s)}{\partial {}^c\bar{\boldsymbol{\omega}}_s} {}^c\tilde{\boldsymbol{\omega}}_s + \frac{\partial {}^c\dot{\mathbf{q}}_s ({}^c\bar{\boldsymbol{\omega}}_s, {}^c\bar{\mathbf{q}}_s)}{\partial {}^c\bar{\mathbf{q}}_s} {}^c\tilde{\mathbf{q}}_s \quad (8.100)$$

The partial derivatives of a function $f(\mathbf{x}, \mathbf{y})$ with respect to vectors \mathbf{x} and \mathbf{y} are equal to the Jacobian of the function ($\nabla \mathbf{f}$):

$$\frac{\partial f(\mathbf{x}, \mathbf{y})}{\partial \mathbf{x}} = \nabla \mathbf{f}(\mathbf{x}) \quad \frac{\partial f(\mathbf{x}, \mathbf{y})}{\partial \mathbf{y}} = \nabla \mathbf{f}(\mathbf{y}) \quad (8.101)$$

Equation 8.101 can be combined in a single term:

$$\frac{\partial f(\mathbf{x}, \mathbf{y})}{\partial \mathbf{x}}_a + \frac{\partial f(\mathbf{x}, \mathbf{y})}{\partial \mathbf{y}}_b = \nabla \mathbf{f}(\mathbf{x}, \mathbf{y}) \begin{bmatrix} a \\ b \end{bmatrix} \quad (8.102)$$

Applying Equation 8.101 to Equation 8.100 yields:

$${}^c \dot{\mathbf{q}}_s ({}^c \tilde{\boldsymbol{\omega}}_s, {}^c \tilde{\mathbf{q}}_s) \approx \underline{\mathbf{S}}_4 ({}^c \tilde{\boldsymbol{\omega}}_s) {}^c \tilde{\mathbf{q}}_s + \nabla {}^c \dot{\mathbf{q}} ({}^c \tilde{\boldsymbol{\omega}}_s, {}^c \tilde{\mathbf{q}}_s) \begin{bmatrix} {}^c \tilde{\boldsymbol{\omega}}_s \\ {}^c \tilde{\mathbf{q}}_s \end{bmatrix} \quad (8.103)$$

The Jacobian of the kinematic model is then found as:

$$\nabla {}^c \dot{\mathbf{q}}_s ({}^c \tilde{\boldsymbol{\omega}}_s, {}^c \tilde{\mathbf{q}}_s) = \frac{1}{2} \begin{bmatrix} {}^c \bar{\mathbf{q}}_{s4} & {}^c \bar{\mathbf{q}}_{s3} & -{}^c \bar{\mathbf{q}}_{s2} & 0 & -{}^c \bar{\omega}_{s3} & {}^c \bar{\omega}_{s2} & {}^c \bar{\omega}_{s1} \\ -{}^c \bar{\mathbf{q}}_{s3} & {}^c \bar{\mathbf{q}}_{s4} & {}^c \bar{\mathbf{q}}_{s1} & {}^c \bar{\omega}_{s3} & 0 & -{}^c \bar{\omega}_{s1} & {}^c \bar{\omega}_{s2} \\ {}^c \bar{\mathbf{q}}_{s2} & -{}^c \bar{\mathbf{q}}_{s1} & {}^c \bar{\mathbf{q}}_{s4} & -{}^c \bar{\omega}_{s2} & {}^c \bar{\omega}_{s1} & 0 & {}^c \bar{\omega}_{s3} \\ -{}^c \bar{\mathbf{q}}_{s1} & -{}^c \bar{\mathbf{q}}_{s2} & -{}^c \bar{\mathbf{q}}_{s3} & -{}^c \bar{\omega}_{s1} & -{}^c \bar{\omega}_{s2} & -{}^c \bar{\omega}_{s3} & 0 \end{bmatrix} \quad (8.104)$$

It is recalled that the attitude can as stated by Equation 8.25 be approximated with the small angle approximation:

$$\Delta {}^c \tilde{\mathbf{q}}_s \approx \begin{bmatrix} \Delta {}^c \tilde{\mathbf{q}}_s \\ 1 \end{bmatrix} \quad (8.105)$$

where ${}^c \tilde{\mathbf{q}}_s$ is the imaginary part of the quaternion ${}^c \tilde{\mathbf{q}}_s$. The fact that ${}^c \tilde{\mathbf{q}}_{s4} \approx 1$ as shown in Equation 8.25, results in the derivative ${}^c \dot{\tilde{\mathbf{q}}}_{s4} \approx 0$ such that:

$${}^c \dot{\tilde{\mathbf{q}}}_s \approx \begin{bmatrix} {}^c \dot{\tilde{\mathbf{q}}}_s \\ 0 \end{bmatrix} \quad (8.106)$$

This results in the fourth row of the linearisation being zero, such that the terms can be expressed as Equation 8.107 and Equation 8.108 respectively.

$$\underline{\mathbf{S}}_4({}^c\bar{\boldsymbol{\omega}}_s) {}^c\bar{\mathbf{q}}_s \approx \underline{\mathbf{S}}_3({}^c\bar{\boldsymbol{\omega}}_s) {}^c\bar{\mathbf{q}}_s \quad (8.107)$$

$$\begin{aligned} \nabla {}^c\dot{\mathbf{q}}_s({}^c\tilde{\boldsymbol{\omega}}_s, {}^c\tilde{\mathbf{q}}_s) &\approx \nabla {}^c\dot{\mathbf{q}}_s({}^c\tilde{\boldsymbol{\omega}}_s, {}^c\tilde{\mathbf{q}}_s) \\ &= \frac{1}{2} \begin{bmatrix} 1 & {}^c\bar{q}_{s3} & -{}^c\bar{q}_{s2} & 0 & -{}^c\bar{\omega}_{s3} & {}^c\bar{\omega}_{s2} \\ -{}^c\bar{q}_{s3} & 1 & {}^c\bar{q}_{s1} & {}^c\bar{\omega}_{s3} & 0 & -{}^c\bar{\omega}_{s1} \\ {}^c\bar{q}_{s2} & -{}^c\bar{q}_{s1} & 1 & -{}^c\bar{\omega}_{s2} & {}^c\bar{\omega}_{s1} & 0 \end{bmatrix} \end{aligned} \quad (8.108)$$

The final linearised kinematic model is then found as Equation 8.109.

$$\begin{aligned} {}^c\dot{\mathbf{q}}_s({}^c\tilde{\boldsymbol{\omega}}_s, {}^c\tilde{\mathbf{q}}_s) &\approx \frac{1}{2}\underline{\mathbf{S}}_4({}^c\bar{\boldsymbol{\omega}}_s) {}^c\bar{\mathbf{q}}_s + \nabla {}^c\dot{\mathbf{q}}_s({}^c\tilde{\boldsymbol{\omega}}_s, {}^c\tilde{\mathbf{q}}_s) \begin{bmatrix} {}^c\tilde{\boldsymbol{\omega}}_s \\ {}^c\tilde{\mathbf{q}}_s \end{bmatrix} \\ &\approx \frac{1}{2}\underline{\mathbf{S}}_3({}^c\bar{\boldsymbol{\omega}}_s) {}^c\bar{\mathbf{q}}_s + \nabla {}^c\dot{\mathbf{q}}_s({}^c\tilde{\boldsymbol{\omega}}_s, {}^c\tilde{\mathbf{q}}_s) \begin{bmatrix} {}^c\tilde{\boldsymbol{\omega}}_s \\ {}^c\tilde{\mathbf{q}}_s \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} {}^c\bar{\omega}_{s1} + {}^c\tilde{\omega}_{s1} + {}^c\bar{\omega}_{s2} {}^c\bar{q}_{s3} - {}^c\bar{\omega}_{s3} {}^c\bar{q}_{s2} + {}^c\tilde{\omega}_{s2} {}^c\bar{q}_{s3} - {}^c\tilde{\omega}_{s3} {}^c\bar{q}_{s2} + {}^c\bar{\omega}_{s2} {}^c\bar{q}_{s3} - {}^c\bar{\omega}_{s3} {}^c\bar{q}_{s2} \\ {}^c\bar{\omega}_{s2} + {}^c\tilde{\omega}_{s2} - {}^c\bar{\omega}_{s1} {}^c\bar{q}_{s3} + {}^c\bar{\omega}_{s3} {}^c\bar{q}_{s1} - {}^c\tilde{\omega}_{s1} {}^c\bar{q}_{s3} + {}^c\tilde{\omega}_{s3} {}^c\bar{q}_{s1} - {}^c\bar{\omega}_{s1} {}^c\bar{q}_{s3} + {}^c\bar{\omega}_{s3} {}^c\bar{q}_{s1} \\ {}^c\bar{\omega}_{s3} + {}^c\tilde{\omega}_{s3} + {}^c\bar{\omega}_{s1} {}^c\bar{q}_{s2} - {}^c\bar{\omega}_{s2} {}^c\bar{q}_{s1} + {}^c\tilde{\omega}_{s1} {}^c\bar{q}_{s2} - {}^c\tilde{\omega}_{s2} {}^c\bar{q}_{s1} + {}^c\bar{\omega}_{s1} {}^c\bar{q}_{s2} - {}^c\bar{\omega}_{s2} {}^c\bar{q}_{s1} \end{bmatrix} \end{aligned} \quad (8.109)$$

8.6.4.2 Linearised dynamic model

The dynamic model is found in Section 8.4 to be:

$${}^c\dot{\boldsymbol{\omega}}_s = \underline{\mathbf{J}}_s^{-1} \left[-\underline{\mathbf{S}}_3({}^c\boldsymbol{\omega}_s) \underline{\mathbf{J}}_s {}^c\boldsymbol{\omega}_s + {}^c\boldsymbol{\tau}_w + {}^c\boldsymbol{\tau}_d \right] \quad (8.110)$$

The disturbance torque ${}^c\boldsymbol{\tau}_d$ is assumed to be zero, and is excluded from the linearised model. The dynamic model used in the linearisation is then reduced to:

$${}^c\dot{\boldsymbol{\omega}}_s = \underline{\mathbf{J}}_s^{-1} \left[-\underline{\mathbf{S}}_3({}^c\boldsymbol{\omega}_s) \underline{\mathbf{J}}_s {}^c\boldsymbol{\omega}_s + {}^c\boldsymbol{\tau}_w \right] \quad (8.111)$$

The same procedure as for the kinematic model is used to linearise the dynamic model.

The angular velocity and reaction wheel torques are split into operating points and perturbations:

$${}^c\boldsymbol{\omega}_s = {}^c\bar{\boldsymbol{\omega}}_s + {}^c\tilde{\boldsymbol{\omega}}_s \quad {}^c\boldsymbol{\tau}_w = {}^c\bar{\boldsymbol{\tau}}_w + {}^c\tilde{\boldsymbol{\tau}}_w \quad (8.112)$$

The dynamic model is then rewritten as:

$${}^c\dot{\omega}_s({}^c\tilde{\omega}_w, {}^c\tilde{\tau}_w) = \underline{J}_s^{-1} \left[-\underline{S}_3({}^c\bar{\omega}_s + {}^c\tilde{\omega}_s) \underline{J}_s({}^c\bar{\omega}_s + {}^c\tilde{\omega}_s) + {}^c\bar{\tau}_w + {}^c\tilde{\tau}_w \right] \quad (8.113)$$

Using the same procedure as for the kinematic model, the linearised dynamic model is found to be:

$$\begin{aligned} {}^c\dot{\omega}_s({}^c\tilde{\omega}_s, {}^c\tilde{\tau}_w) &\approx {}^c\dot{\omega}_s({}^c\bar{\omega}_s, {}^c\bar{\tau}_w) + \nabla {}^c\dot{\omega}_s({}^c\bar{\omega}_s, {}^c\bar{\tau}_w) \begin{bmatrix} {}^c\tilde{\omega}_s \\ {}^c\tilde{\tau}_w \end{bmatrix} \\ &= \begin{bmatrix} \underline{J}_{s1}^{-1} \left({}^c\tilde{\tau}_{w1} - {}^c\bar{\omega}_{s2} {}^c\tilde{\tau}_{w3} + {}^c\bar{\omega}_{s3} {}^c\tilde{\tau}_{w2} + {}^c\tilde{\omega}_{s3} \left[{}^c\bar{\tau}_{w2} + \underline{J}_{s2} {}^c\bar{\omega}_{s2} - \underline{J}_{s3} {}^c\bar{\omega}_{s2} \right] - {}^c\tilde{\omega}_{s2} \left[{}^c\bar{\tau}_{w3} - \underline{J}_{s2} {}^c\bar{\omega}_{s3} + \underline{J}_{s3} {}^c\bar{\omega}_{s3} \right] \right) \\ \underline{J}_{s2}^{-1} \left({}^c\tilde{\tau}_{w2} + {}^c\bar{\omega}_{s1} {}^c\tilde{\tau}_{w3} - {}^c\bar{\omega}_{s3} {}^c\tilde{\tau}_{w1} - {}^c\tilde{\omega}_{s3} \left[{}^c\bar{\tau}_{w1} + \underline{J}_{s1} {}^c\bar{\omega}_{s1} - \underline{J}_{s3} {}^c\bar{\omega}_{s1} \right] + {}^c\tilde{\omega}_{s1} \left[{}^c\bar{\tau}_{w3} - \underline{J}_{s1} {}^c\bar{\omega}_{s3} + \underline{J}_{s3} {}^c\bar{\omega}_{s3} \right] \right) \\ \underline{J}_{s3}^{-1} \left({}^c\tilde{\tau}_{w3} - {}^c\bar{\omega}_{s1} {}^c\tilde{\tau}_{w2} + {}^c\bar{\omega}_{s2} {}^c\tilde{\tau}_{w1} + {}^c\tilde{\omega}_{s2} \left[{}^c\bar{\tau}_{w1} + \underline{J}_{s1} {}^c\bar{\omega}_{s1} - \underline{J}_{s2} {}^c\bar{\omega}_{s1} \right] - {}^c\tilde{\omega}_{s1} \left[{}^c\bar{\tau}_{w2} - \underline{J}_{s1} {}^c\bar{\omega}_{s2} + \underline{J}_{s2} {}^c\bar{\omega}_{s2} \right] \right) \end{bmatrix} \end{aligned} \quad (8.114)$$

8.6.5 Solving Linear State Space Model

The expression for the derivative of the state vector \dot{x} is examined in order to solve the linear state space model. The expression for \dot{x} is given in Section 8.6.1 as:

$$\dot{x} = \underline{A}x + \underline{B}u \quad (8.115)$$

The linearised expression for \dot{x} is given in Section 8.6.1 as:

$$\dot{x} = \begin{bmatrix} {}^c\dot{q}_s \\ {}^c\dot{\omega}_s \end{bmatrix} \approx \begin{bmatrix} \frac{1}{2}\underline{S}_3({}^c\bar{\omega}_s) {}^c\bar{q}_s + \nabla {}^c\dot{q}({}^c\bar{\omega}_s, {}^c\bar{q}_s) \begin{bmatrix} {}^c\tilde{\omega}_s \\ {}^c\tilde{q}_s \end{bmatrix} \\ {}^c\dot{\omega}_s({}^c\bar{\omega}_s, {}^c\bar{\tau}_w) + \nabla {}^c\dot{\omega}_s({}^c\bar{\omega}_s, {}^c\bar{\tau}_w) \begin{bmatrix} {}^c\tilde{\omega}_s \\ {}^c\tilde{\tau}_w \end{bmatrix} \end{bmatrix} \quad (8.116)$$

The expressions of \underline{A} and \underline{B} can be found by computing the Jacobian matrices of \dot{x} with respect to x and u respectively.

The Jacobian matrix of \dot{x} with respect to x is given as:

$$\underline{A} = \nabla \dot{x}(x) = \nabla \dot{x} \left(\begin{bmatrix} {}^c\tilde{\omega}_s \\ {}^c\tilde{q}_s \end{bmatrix} \right) \quad (8.117)$$

The Jacobian matrix of \dot{x} with respect to u is given as:

$$\underline{B} = \nabla \dot{x}(u) = \nabla \dot{x}({}^c\tilde{\tau}_w) \quad (8.118)$$

Computing the expressions for $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ yields:

$$\underline{\mathbf{A}} = \begin{bmatrix} 0 & -\frac{^c\bar{\omega}_{s3}}{2} & \frac{^c\bar{\omega}_{s2}}{2} & \frac{1}{2} & \frac{^c\bar{q}_3}{2} & -\frac{^c\bar{q}_2}{2} \\ \frac{^c\bar{\omega}_{s3}}{2} & 0 & -\frac{^c\bar{\omega}_{s1}}{2} & -\frac{^c\bar{q}_3}{2} & \frac{1}{2} & \frac{^c\bar{q}_1}{2} \\ -\frac{^c\bar{\omega}_{s2}}{2} & \frac{^c\bar{\omega}_{s1}}{2} & 0 & \frac{^c\bar{q}_2}{2} & -\frac{^c\bar{q}_1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{^c\bar{\omega}_{s3}(J_{s2} - J_{s3})}{J_{s1}} & \frac{^c\bar{\omega}_{s2}(J_{s2} - J_{s3})}{J_{s1}} \\ 0 & 0 & 0 & -\frac{^c\bar{\omega}_{s3}(J_{s1} - J_{s3})}{J_{s2}} & 0 & -\frac{^c\bar{\omega}_{s1}(J_{s1} - J_{s3})}{J_{s2}} \\ 0 & 0 & 0 & \frac{^c\bar{\omega}_{s2}(J_{s1} - J_{s2})}{J_{s3}} & \frac{^c\bar{\omega}_{s1}(J_{s1} - J_{s2})}{J_{s3}} & 0 \end{bmatrix} \quad (8.119)$$

$$\underline{\mathbf{B}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{J_{s1}} & 0 & 0 \\ 0 & \frac{1}{J_{s2}} & 0 \\ 0 & 0 & \frac{1}{J_{s3}} \end{bmatrix} \quad (8.120)$$

8.6.6 Operating Points

The operating points of where the system is linearised is now inserted into the state matrix $\underline{\mathbf{A}}$ and the input matrix $\underline{\mathbf{B}}$

The operating point for the kinematic model is, $\theta = 0$ which results in the operating point for the kinematic model being described as Equation 8.121.

$$\bar{\mathbf{q}} = \hat{\mathbf{q}} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T \quad (8.121)$$

When in nadir mode, the angular velocity for the satellite in SRF, is given as:

$${}^s\boldsymbol{\omega}_s = \begin{bmatrix} 0 & -1.127e-3 & 0 \end{bmatrix} \quad (8.122)$$

The angular velocity for the satellite in the CRF, which is the reference frame used in the kinematic and dynamic model, is calculated using the rotation quaternion ${}^s\mathbf{q}$ and results in the following angular velocity:

$${}^c\bar{\boldsymbol{\omega}}_s = \begin{bmatrix} 4.485e-5 & -1.127e-3 & -2.086e-5 \end{bmatrix} \quad (8.123)$$

The inertia matrix $\underline{\mathbf{J}}_s$ from AAUSAT4 described in section 4.2.4 is also inserted.

This results in the matrices for the state space model being given as Equation 8.124 and Equa-

tion 8.125.

$$\underline{\mathbf{A}} = \begin{bmatrix} 0 & 1.04e-5 & -0.56e-3 & 0.5 & 0 & 0 \\ -1.04e-5 & 0 & -2.24e-5 & 0 & 0.5 & 0 \\ 0.56e-3 & 2.24e-5 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & -4.74e-6 & 0 & 1.02e-5 \\ 0 & 0 & 0 & 2.56e-4 & -1.02e-5 & 0 \end{bmatrix} \quad (8.124)$$

$$\underline{\mathbf{B}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 588.24 & 0 & 0 \\ 0 & 454.55 & 0 \\ 0 & 0 & 454.55 \end{bmatrix} \quad (8.125)$$

Using the numerical values for the $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ matrices, the system is now ready to be controlled.

8.6.7 Control of the Linearised State Space Model

To control the system, a feedback matrix $\underline{\mathbf{K}}$ is added to the system, such that state error feedback is obtained. The controlled system is shown in Figure 8.15.

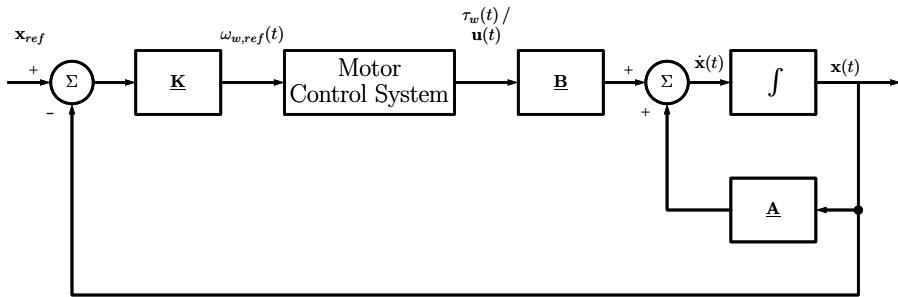


Figure 8.15: Block diagram of the controlled system.

The feedback matrix $\underline{\mathbf{K}}$ essentially moves the poles of the system to the desired locations.

8.6.7.1 Controllability and pole placement

After the system has been linearised, it is important to check if the system is controllable. The controllability matrix is given by Equation 8.126.

$$\underline{\mathbf{P}} = [\underline{\mathbf{B}} \ \underline{\mathbf{A}}^1 \underline{\mathbf{B}} \ \underline{\mathbf{A}}^2 \underline{\mathbf{B}} \ \cdots \ \underline{\mathbf{A}}^{n-1} \underline{\mathbf{B}}] \quad (8.126)$$

n	Number of states in the system
-----	--------------------------------

The system is controllable if the rank of the controllability matrix is equal to the number of states in the system [38]. The rank of the controllability matrix is 6, which is equal to the number of states in the system, and the system is therefore controllable.

After determining the controllability of the system, the eigenvalues has to be placed to fulfill the requirements of the system. This can be done by varying the feedback matrix \underline{K} . By applying the feedback matrix \underline{K} the eigenvalues of the system are the eigenvalues of the matrix $\underline{A} - \underline{B}\underline{K}$. There are several different ways to determine values for \underline{K} .

One way to do this is by placing the eigenvalues for a second order system. Then placing the remaining eigenvalues, using the general rule of thumb, that the eigenvalues should be placed at least 10 times further to the left than the dominant eigenvalues in the complex plane [38].

In an under damped second order system, the eigenvalues are given by [38]:

$$\lambda_{1,2} = -\zeta \omega_n \pm j \omega_d \quad (8.127)$$

ζ	Damping ratio
ω_n	Fixed undamped natural frequency
ω_d	Damped natural frequency

The damped natural frequency is given by:

$$\omega_d = \omega_n \sqrt{1 - \zeta^2} \quad (8.128)$$

The percent overshoot is given by:

$$PO = 100 e^{\frac{-\zeta \pi}{\sqrt{1-\zeta^2}}} \quad (8.129)$$

From the requirements the maximum overshoot and the settling time is known. By using these values the dominant eigenvalues can be placed, and the remaining eigenvalues can be placed using the rule of thumb.

When the poles are calculated, a feedback matrix \underline{K} can be calculated. This can be done by using the MATLAB function `place`.

Since the system is of higher order the second order approximation will not necessarily give the desired result. Furthermore it can be difficult to determine what different pole placements will do, for higher order systems. Instead of looking at different pole placements, and trying to get the desired result, the LQR method can be used. This method uses a cost function where the cost of having error and the cost of using the actuators is taken into account. This means that the values of feedback matrix \underline{K} can be found by defining the cost of error and actuator effort [39].

8.6.8 LQR

As mentioned the values of the feedback matrix $\underline{\mathbf{K}}$ can be found by first minimizing the cost function. The cost function is given by Equation 8.130 [39].

$$J(\mathbf{x}, \mathbf{u}) = \int_0^\infty (\mathbf{x}^T \underline{\mathbf{Q}} \mathbf{x} + \mathbf{u}^T \underline{\mathbf{R}} \mathbf{u}) dt \quad (8.130)$$

$\underline{\mathbf{Q}}$	Positive definite diagonal matrix, weighting state error
$\underline{\mathbf{R}}$	Positive diagonal matrix, weighting control effort

The control law minimizing the cost function is given by Equation 8.131.

$$\mathbf{u} = -\underline{\mathbf{K}} \mathbf{x} \quad (8.131)$$

where $\underline{\mathbf{K}}$ is given by Equation 8.132.

$$\underline{\mathbf{K}} = \underline{\mathbf{R}}^{-1} \underline{\mathbf{B}}^T \underline{\mathbf{S}} \quad (8.132)$$

and $\underline{\mathbf{S}}$ is the solution to Equation 8.133 [39].

$$\underline{\mathbf{A}}^T \underline{\mathbf{S}} + \underline{\mathbf{S}} \underline{\mathbf{A}} - \underline{\mathbf{S}} \underline{\mathbf{B}} \underline{\mathbf{R}}^{-1} \underline{\mathbf{B}}^T \underline{\mathbf{S}} + \underline{\mathbf{Q}} = 0 \quad (8.133)$$

The LQR method essentially consists of defining the weighting matrices $\underline{\mathbf{Q}}$ and $\underline{\mathbf{R}}$, using knowledge of the system, and then solving the Riccati equation to find the gain matrix $\underline{\mathbf{K}}$.

Finding the weighting matrices $\underline{\mathbf{Q}}$ and $\underline{\mathbf{R}}$ is done by tuning. Initially, the weighting matrices are chosen to be diagonal matrices with the diagonal elements being the reciprocal of the square of the maximum desired state values and control effort values [40].

$$Q_{ii} = \frac{1}{x_{i,\max}^2} \quad R_{ii} = \frac{1}{u_{i,\max}^2} \quad (8.134)$$

Initial values for x_{\max} and u_{\max} are chosen based on the system requirements.

The maximum deviation from the desired attitude is determined in Section 4.3 to be 5° , which corresponds to $5 \frac{\pi}{180} = 87.27$ mrad. The maximum angular velocity is determined in Section 4.3 to be 19.21 mrad/s.

The maximum control effort is determined in Section 8.5 to be 773.3 nN m.

$$\underline{x}_{\max} = \begin{bmatrix} q_{\max} \\ \omega_{\max} \end{bmatrix} = \begin{bmatrix} 87.27e - 3 \\ 87.27e - 3 \\ 87.27e - 3 \\ 19.21e - 3 \\ 19.21e - 3 \\ 19.21e - 3 \end{bmatrix} \quad (8.135)$$

$$\underline{u}_{\max} = \tau_{\max} = \begin{bmatrix} 773.3e - 9 \\ 773.3e - 9 \\ 773.3e - 9 \end{bmatrix} \quad (8.136)$$

The initial weighting matrices are therefore calculated to be:

$$\underline{\mathbf{Q}} = \begin{bmatrix} 131.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 131.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 131.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2710 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2710 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2710 \end{bmatrix} \quad (8.137)$$

$$\underline{\mathbf{R}} = \begin{bmatrix} 1.672e12 & 0 & 0 \\ 0 & 1.672e12 & 0 \\ 0 & 0 & 1.672e12 \end{bmatrix} \quad (8.138)$$

Using the initial weighting matrices, a feedback controller can be implemented in the simulation, and the performance of the controller can be evaluated.

To be able to simulate the controller, the system needs to be discretized.

8.6.9 Discretization

When implementing the controller on a microcontroller, the controller needs to be discretized to run on a discrete system.

The discretization is done using the bilinear transform, which requires laplace transform of the continuous system. The laplace transform of the continuous system is given by Equation 8.139.

$$s\underline{\mathbf{X}}(s) = \underline{\mathbf{A}}\underline{\mathbf{X}}(s) + \underline{\mathbf{B}}\underline{\mathbf{U}}(s) \quad (8.139)$$

$\underline{\mathbf{X}}(s)$	Laplace transform of the state vector
$\underline{\mathbf{U}}(S)$	Laplace transform of the input vector

The bilinear transform is given by Equation 8.140.

$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (8.140)$$

$T \mid$ Sampling period

The discrete system becomes as shown in Equation 8.141.

$$\underline{x}[n+1] = \underline{\mathbf{A}}_d \underline{x}[n] + \underline{\mathbf{B}}_d \underline{u}[n] \quad (8.141)$$

Where $\underline{x}[n]$ and $\underline{u}[n]$ are the state and control vectors at time step n respectively

$\underline{x}[n]$	State vector at time step
$\underline{u}[n]$	State input at time step

and $\underline{\mathbf{A}}_d$ and $\underline{\mathbf{B}}_d$ are given by Equation 8.142 and 8.143 respectively [41].

$$\underline{\mathbf{A}}_d = \left(\underline{\mathbf{I}} - \frac{\underline{\mathbf{A}} T_s}{2} \right)^{-1} \left(\underline{\mathbf{I}} + \frac{\underline{\mathbf{A}} T_s}{2} \right) \quad (8.142)$$

$$\underline{\mathbf{B}}_d = \left(\underline{\mathbf{I}} - \frac{\underline{\mathbf{A}} T_s}{2} \right)^{-1} \underline{\mathbf{B}} T_s \quad (8.143)$$

The discretization results in the discrete LQR cost function given by Equation 8.144.

$$J(\underline{x}[k], \underline{u}[k]) = \sum_{k=0}^{\infty} (\underline{x}[k]^T \underline{\mathbf{Q}} \underline{x}[k] + \underline{u}[k]^T \underline{\mathbf{R}} \underline{u}[k]) \quad (8.144)$$

Where the optimal control law minimizing the cost function is given by Equation 8.145.

$$\underline{u}[k] = -\underline{\mathbf{K}}_d \underline{x}[k] \quad (8.145)$$

where $\underline{\mathbf{K}}_d$ is the discrete LQR gain matrix given by Equation 8.146.

$$\underline{\mathbf{K}}_d = \left(\underline{\mathbf{R}} + \underline{\mathbf{B}}_d^T \underline{\mathbf{S}}_d \underline{\mathbf{B}}_d \right)^{-1} \underline{\mathbf{B}}_d^T \underline{\mathbf{S}}_d \underline{\mathbf{A}}_d \quad (8.146)$$

and $\underline{\mathbf{S}}_d$ is the solution to the discrete Equation 8.147.

$$\underline{\mathbf{S}}_d = \underline{\mathbf{A}}_d^T \underline{\mathbf{S}}_d \underline{\mathbf{A}}_d - \underline{\mathbf{A}}_d^T \underline{\mathbf{S}}_d \underline{\mathbf{B}}_d \left(\underline{\mathbf{R}} + \underline{\mathbf{B}}_d^T \underline{\mathbf{S}}_d \underline{\mathbf{B}}_d \right)^{-1} \underline{\mathbf{B}}_d^T \underline{\mathbf{S}}_d \underline{\mathbf{A}}_d + \underline{\mathbf{Q}} \quad (8.147)$$

Using this method, the discrete LQR gain matrix can be calculated numerically using software such as MATLAB.

8.6.10 Implementation

The continuous system is constructed in MATLAB using the control toolbox.

```
>> C_sys = ss(A,B)
```

```
C_sys =
```

```
A =
      x1          x2          x3          x4          x5          x6
x1    0  1.043e-05 -0.000563    0.5          0          0
x2 -1.043e-05        0 -2.242e-05    0          0.5          0
x3  0.000563  2.242e-05        0          0          0      0.5
x4    0          0          0          0          0          0
x5    0          0          0 -4.741e-06    0  1.019e-05
x6    0          0          0  0.0002559 -1.019e-05        0
```

```
B =
```

```
      u1          u2          u3
x1    0          0          0
x2    0          0          0
x3    0          0          0
x4  588.2        0          0
x5    0  454.5        0
x6    0          0  454.5
```

The system is discretized using the `c2d` function in MATLAB. The sampling period is set to 1 second as determined in 8.6.2.3.

```

>> D_sys = c2d(C_sys, 1, 'tustin')

D_sys =
A =
    x1      x2      x3      x4      x5      x6
x1      1  1.042e-05 -0.000563   0.5  2.606e-06 -0.0001407
x2 -1.044e-05           1 -2.242e-05 -3.794e-06   0.5 -3.057e-06
x3  0.000563  2.243e-05           1  0.0002047  3.059e-06   0.5
x4      0      0      0      1      0      0
x5      0      0      0 -4.739e-06   1  1.019e-05
x6      0      0      0  0.0002559 -1.019e-05   1
B =
    u1      u2      u3
x1  147.1  0.0005924 -0.03199
x2 -0.001116  113.6 -0.0006948
x3  0.06021  0.0006951  113.6
x4  588.2     0     0
x5 -0.001394  454.5  0.002317
x6  0.07526 -0.002317  454.5

```

Through tuning, the **Q** and **R** matrices are found to be:

```

>> Q = diag(ones(1,6)*(pi/180)^-2)

Q =
1.0e+03 x
  3.2828      0      0      0      0      0
      0  3.2828      0      0      0      0
      0      0  3.2828      0      0      0
      0      0      0  3.2828      0      0
      0      0      0      0  3.2828      0
      0      0      0      0      0  3.2828

```

```
>> R = eye(3)*(7.73e-5)^(-2)
```

```

R =
1.0e+08 x
  1.6736      0      0
      0  1.6736      0
      0      0  1.6736

```

Finally, the discrete LQR gain matrix is calculated using the `dlqr` function in MATLAB.

```
>> [K,S,e] = dlqr(D_sys.A, D_sys.B, Q, R)

K =
1.0e-03 *
0.2970    0.0000   -0.0003    0.7259   -0.0000    0.0002
-0.0000    0.3061   -0.0000   -0.0000    0.8348   -0.0000
 0.0004    0.0000    0.3061    0.0004    0.0000    0.8348
```

Using these numerical values, the discrete LQR controller can be implemented in the simulation.

Chapter 9

Simulation Implementation

In order to evaluate the performance of the system, a simulation environment is implemented using the blocks described in Chapter 6.

The final implementation can be seen in Figure 9.1.

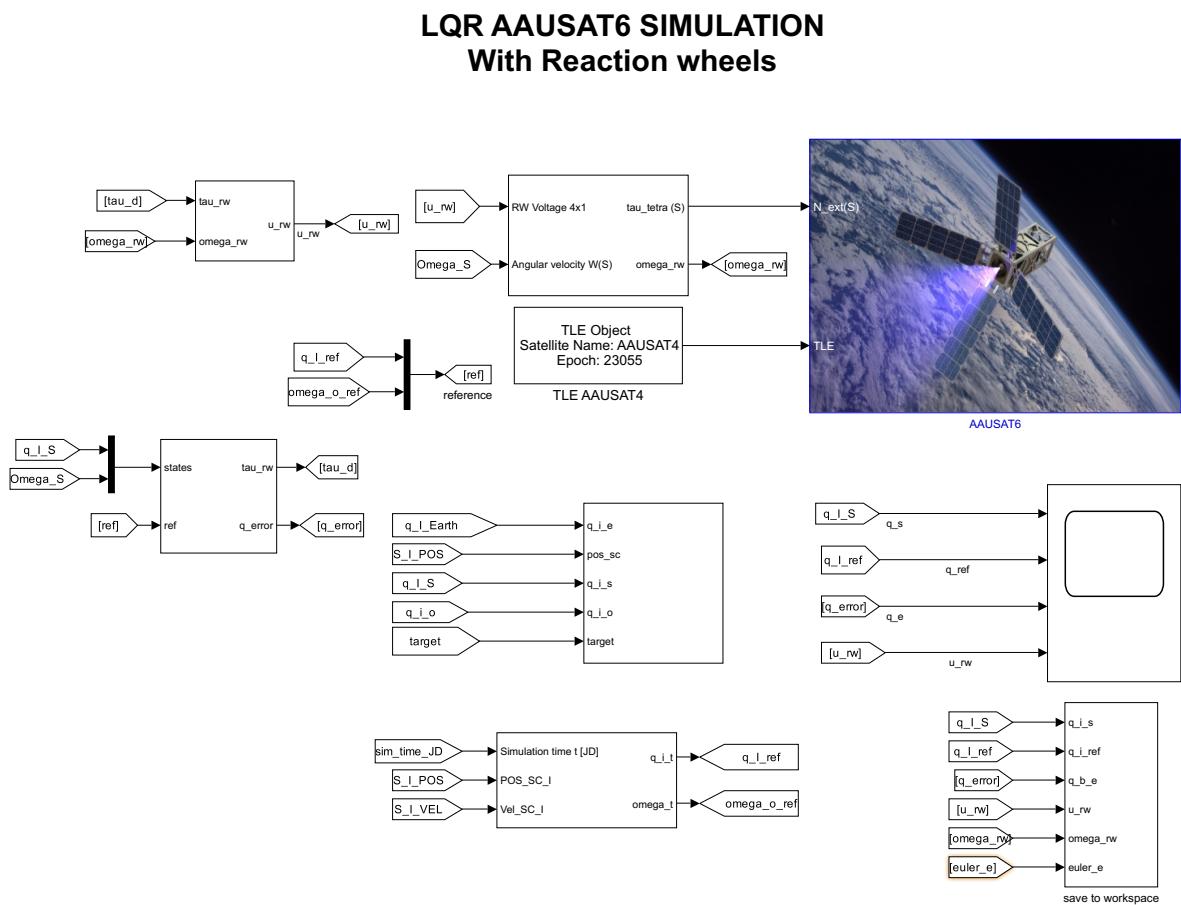


Figure 9.1: The final implementation of the simulation environment.

The following sections will describe the different blocks in the simulation environment.

9.0.1 LQR controller

The LQR controller block is used to apply the feedback matrix $\underline{\mathbf{K}}$. It is a modified version of an existing block from the AAUSAT simulation library. The input to the block is the current state of the satellite, and the target reference. The output is the desired torque to be applied to the reaction wheels, and the quaternion error. The block contains a Matlab function which first takes the references and the target quaternions and rotates them to the CRF and then the error is calculated. The error is then used to calculate the desired torque to be applied to the reaction wheels. The Matlab function can be seen in Listing 1.

```

function [d_tau_tetra,q_error] = fcn(states,ref,K,q_s_c,q_b_tetra)

q_i_s = states(1:4); % The current quaternion of the satellite in the ECI frame
omega = states(5:7); % The current angular velocity of the satellite
q_i_ref = ref(1:4); %The quaternion reference in the ECI frame
omega_ref = ref(5:7); %The angular velocity reference.

q_c_s = qmult(q_i_s,q_s_c); %The rotation from the satellite frame to the
                           %control frame
q_c_ref = qmult(q_i_ref,q_s_c); %The rotation of the reference from the ECI
                           %frame to the control frame
omega_c = qRot(omega,q_s_c);%Rotating the current angular velocity to be in the
                           %control frame
omega_c_ref = qRot(omega_ref,q_s_c);%Rotating the reference angular velocity to
                           %be in the control frame

q_c_tetra = qmult(qinv(q_s_c),q_b_tetra);    %The rotation from the control frame
                                               %to the tetrahedron frame

q_s_e = qmult(q_c_ref,qconj(q_c_s)); %The attitude error in the control frame
x = [q_s_e(1:3); omega_c];
r = [[0 0 0]'; omega_c_ref];

d_tau_c = K*(r - x); %The desired torque in the control frame

d_tau_tetra = qRot(d_tau_c,q_c_tetra); %The desired torque in the
                                         %tetrahedron frame
q_error = q_s_e;

```

Listing 1: The LQR Matlab function

It can be seen that the quaternion error is calculated separately since the attitude error is calculated as a quaternion multiplication unlike how the angular velocity error is calculated, which is a subtraction.

9.0.2 Tetrahedron Reaction Wheels

To implement the reaction wheels in a tetrahedron configuration, a block from the AAUSAT simulation library is used. The tetrahedron reaction wheels block consists of a model of the reaction wheels in a tetrahedron configuration. The input to the block is the voltage applied to the reaction wheels. The output is the angular velocity of the reaction wheels, and the resulting torque from the reaction wheels. Inside the block, the 4 BLDC motors are modelled, as a simplified first order system. The torque from the reaction wheels is then projected and summed in the “Tetrahedron configuration” block, and then rotated into the body frame of the satellite.

The tetrahedron reaction wheels block is shown in Figure 9.4.

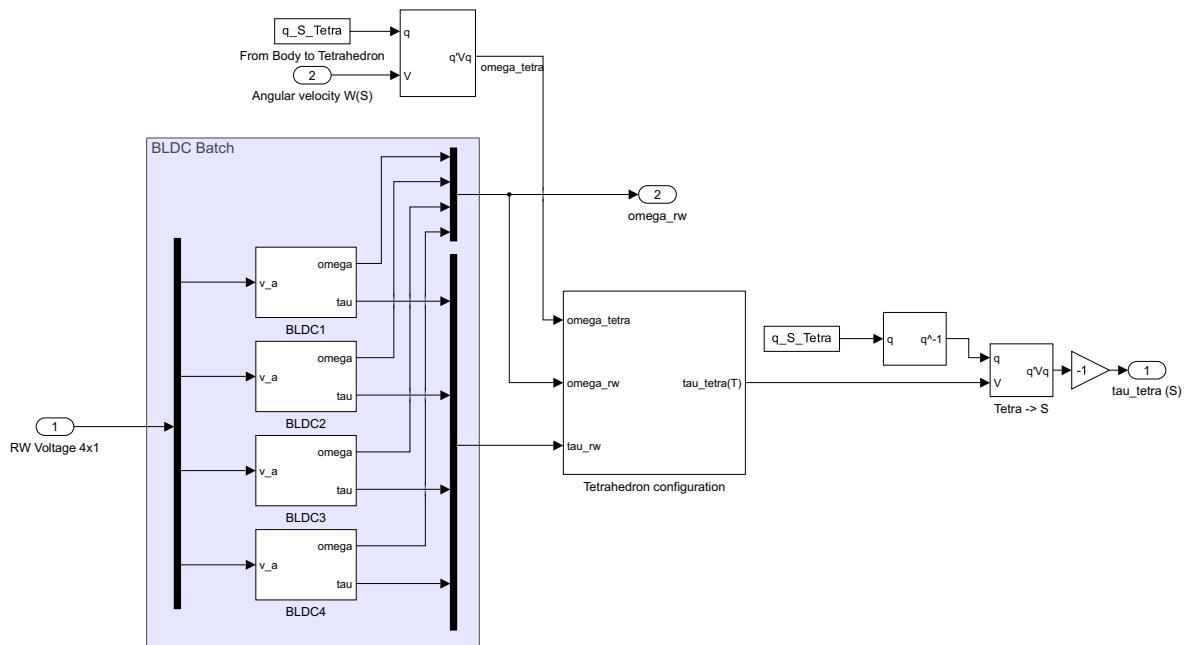


Figure 9.2: Tetrahedron reaction wheels block

With the simulation environment set up as described above, the simulations from the test specification can be performed.

9.0.3 Satellite System Model

The satellite system model is used to simulate the satellite, and contains the nonlinear kinematic and dynamic equations of the satellite described in Section 8.4 and Section 8.3.

Furthermore, the model also contains the equations for the disturbances acting on the satellite and a model of the solar system with the sun and the planets (Epemeris model). ephemeris model The satellite system model is shown in Figure 9.3.

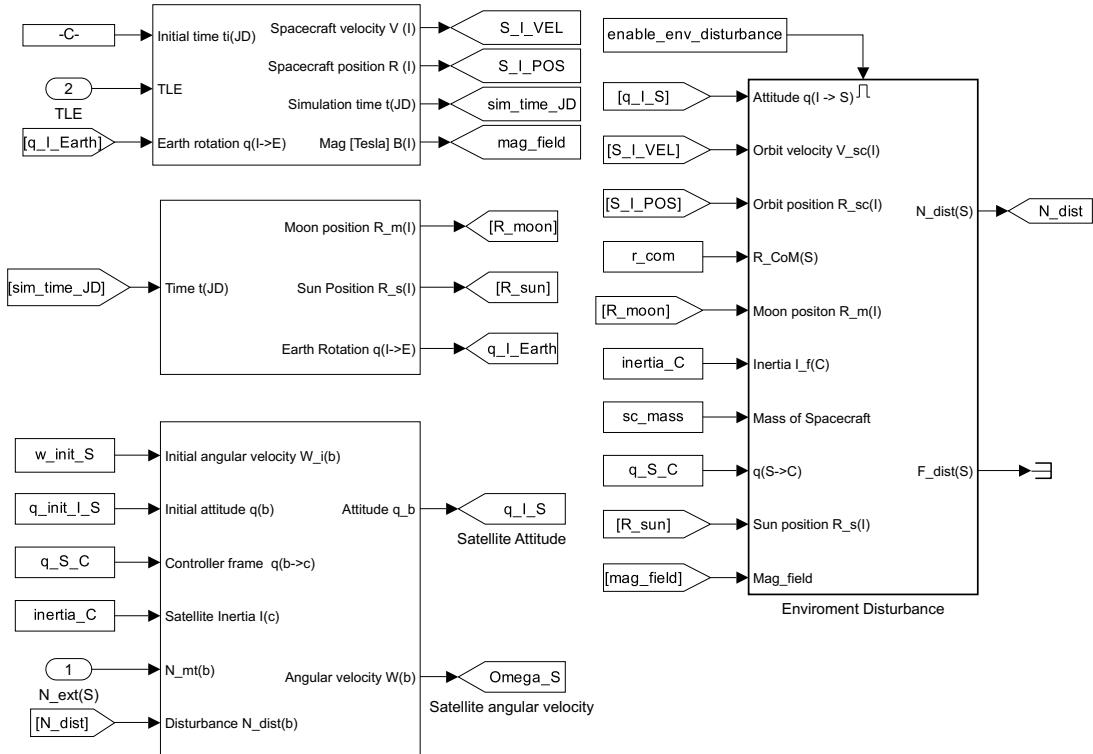


Figure 9.3: Satellite system model

The input to the block is a TLE data set. TLE is an abbreviation for Two Line Element, and is a data set containing the orbital elements of a satellite, such as inclination and eccentricity. The TLE data set is used to calculate the position and velocity vectors of the satellite, and predict future position- and velocity vectors.

This is done using SGP4, which is a model for predicting the position and velocity of a satellite, based on the TLE data set. SGP4 is an abbreviation for Simplified General Perturbations 4, and is an open source model developed by NASA. It includes models for the impact of the gravitational forces from the moon and the sun. Furthermore, it also includes perturbations such as Earth's shape, gravitational differences along Earth, atmospheric drag and solar radiation pressure on the satellite. The simulation is based on the TLE of AAUSAT4, on the 24th of February 2023, as the calculations in Chapter 4 are based on the parameters of this orbit.

9.0.4 Tetrahedron Reaction Wheels

The tetrahedron reaction wheels block consists of a model of the tetrahedron reaction wheels. The input to the block is the voltage applied to the reaction wheels. The output is the angular velocity of the reaction wheels, and the resulting torque from the reaction wheels. Inside the block, the 4 different brushless DC (BLDC) motors are modelled, as a simplified first order system. The torque from the reaction wheels is then projected and summed in the "Tetrahedron configuration" block, and then rotated into the body frame of the satellite.

The tetrahedron reaction wheels block is shown in Figure 9.4.

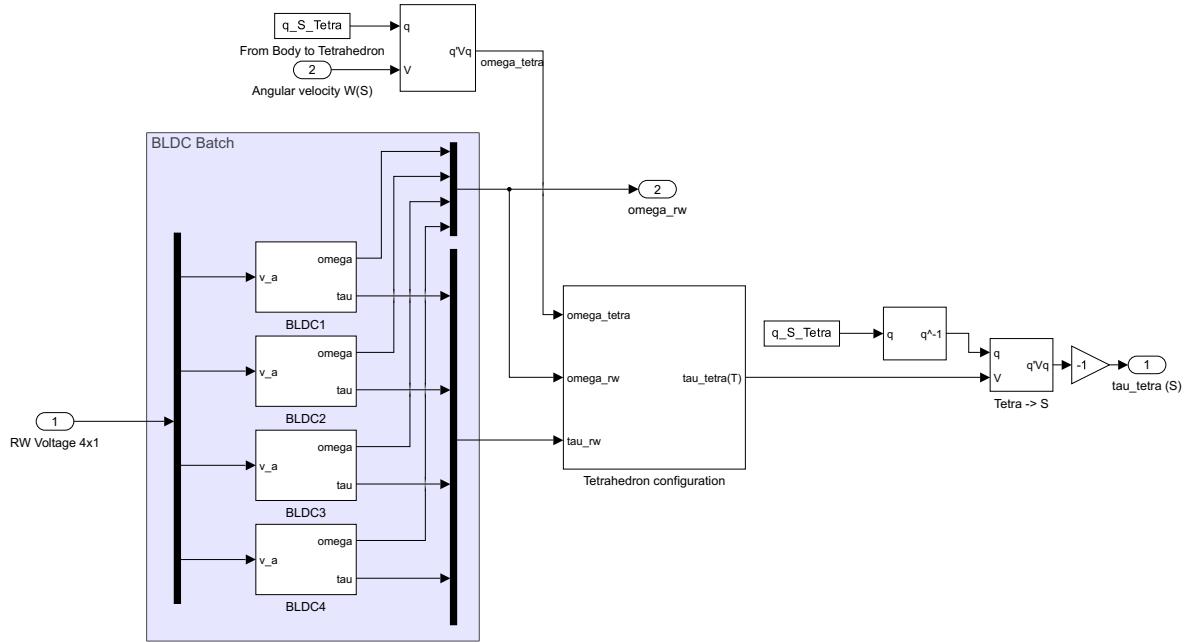


Figure 9.4: Tetrahedron reaction wheels block

9.0.5 Target reference generator

The target reference generator block is used to generate a target reference for the satellite to point at. This is done using the method described in Section 8.1.2. The target reference is generated based on the position of the satellite, and a list of targets and their positions given in the ECEF frame. Using an estimate of the rotation of the ECEF frame, and the position vector of the satellite, the targets which are in sight can be calculated. The output of this block is the target reference quaternion and the target reference angular velocity.

The target reference generator block can be seen on Figure 9.5.

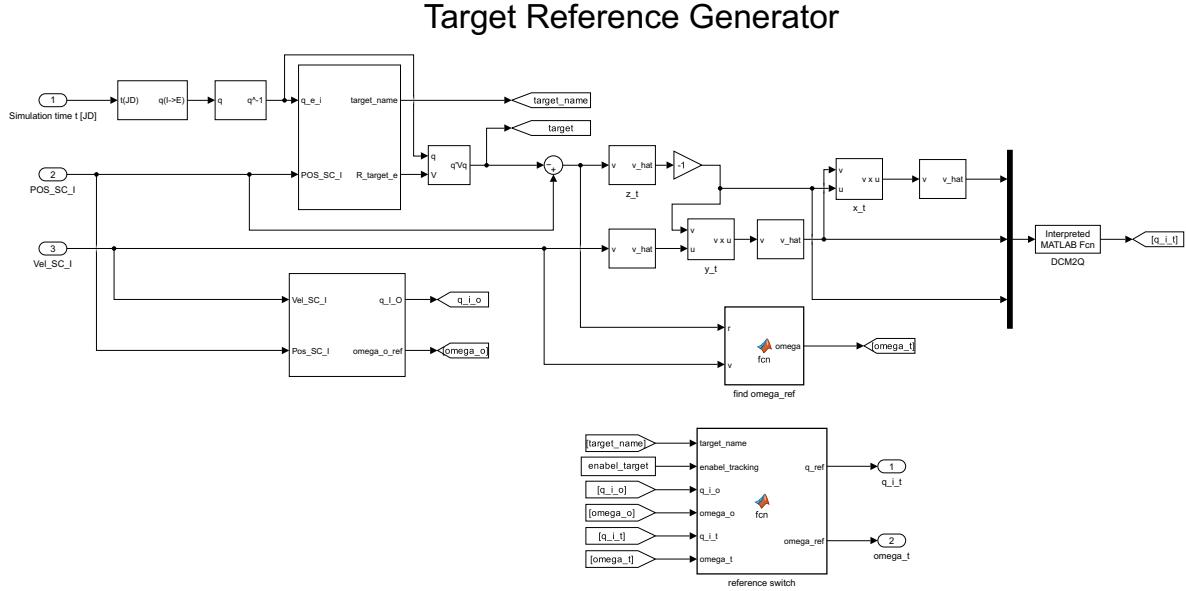


Figure 9.5: Target reference generator block

9.0.6 Reaction wheel control

The reaction wheel control block includes the PI controller for the 4 BLDC motors with the reaction wheels attached. The input for this block is the desired torque, and the current angular velocity of the BLDC motors. The output from this block is the applied voltage on the BLDC motors.

The reaction wheel control block can be seen on Figure 9.6.

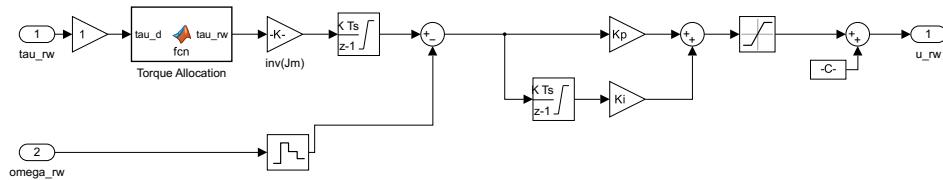


Figure 9.6: Reaction wheel control block

9.0.7 Animation environment

The animation environment block is used to animate the satellite in a 3D environment. It uses the Simulink 3D Animation Viewer to animate the satellite. The input to the block is the position vector and attitude of the satellite in the ECI frame, the rotation quaternion from ECEF to ECI, the rotation quaternion from ECI to ORF, and the target coordinates in the ECI frame. The output is the animation of the satellite in the 3D environment. An example of the output can be seen on Figure 9.7.

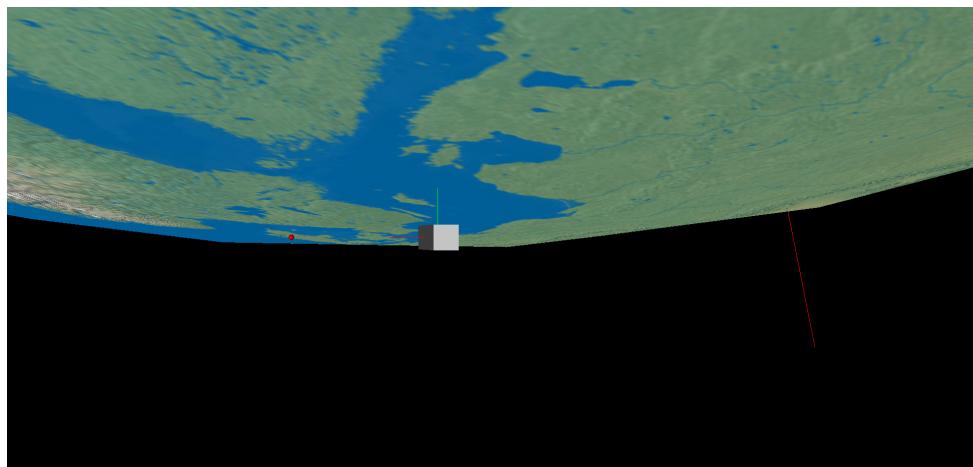


Figure 9.7: Animation environment block

Chapter 10

Acceptance Test

This chapter describes the acceptance test of the system. The tests are performed using the test specification described in Chapter 7.

All the measuring journals can be found in Appendix A.

10.1 Nadir mode acceptance test

This acceptance tests show the systems ability to point correctly in Nadir mode, according to Requirement 5.1.1.

10.1.1 Results

The results of the test can be seen in Figure 10.1.

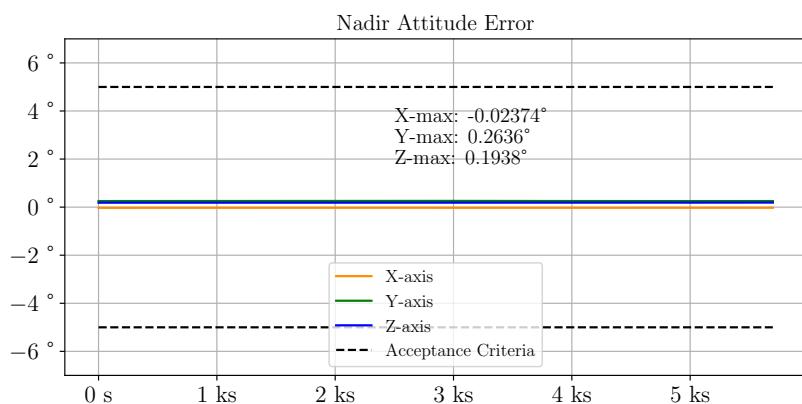


Figure 10.1: Graph showing the pointing accuracy of the system.

10.1.2 Compliance of requirement

As seen the system complies with the requirement of having a pointing accuracy within 5.1 degrees.

Thus, the system complies with the requirements.

10.2 Pointing Mode acceptance test

This acceptance tests show the systems ability to point correctly in pointing mode, according to Requirement 5.1.2. Furthermore, it shows whether the systems step response is within the Requirements 5.1.3, 5.1.4 and 5.1.5.

10.2.1 Results

The results of the test can be seen in Figures 10.2, 10.3 and 10.3.

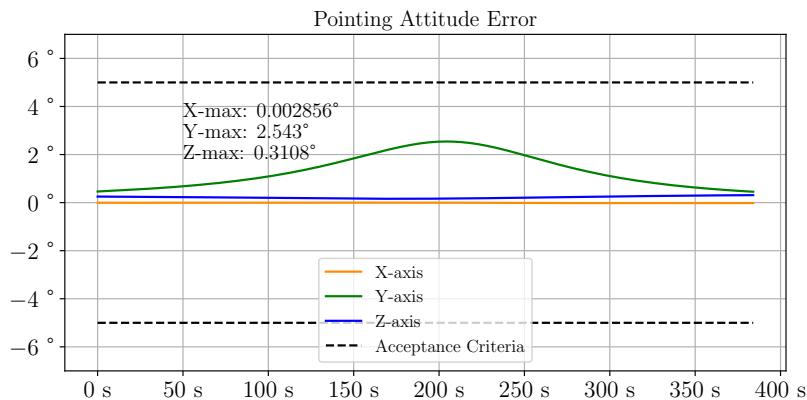


Figure 10.2: Graph showing the pointing accuracy of the system after settling.

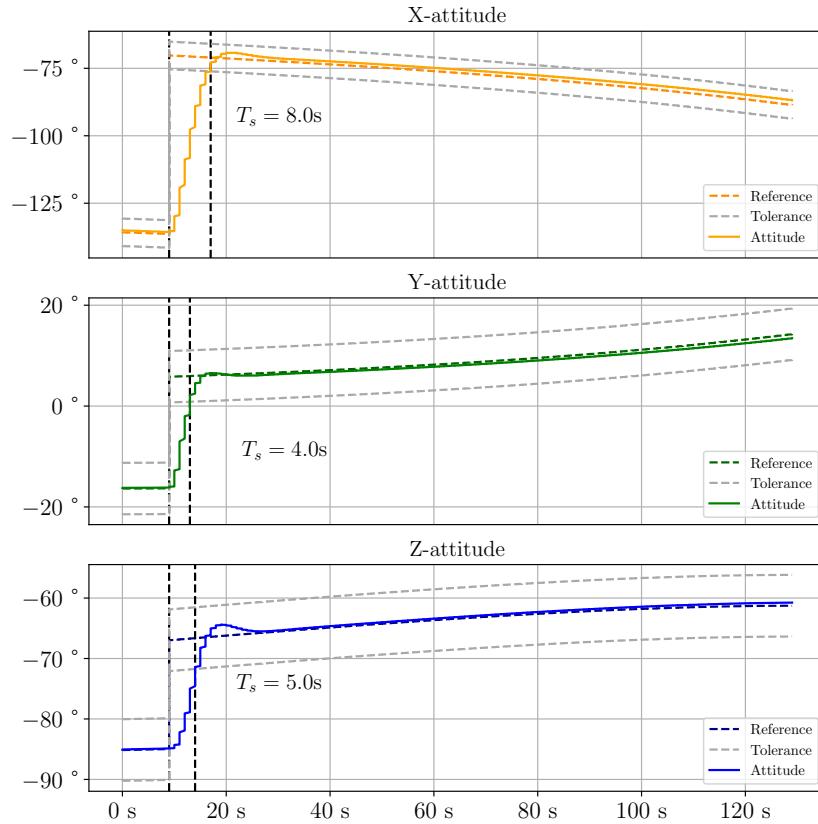


Figure 10.3: Graph showing the step response of the system.

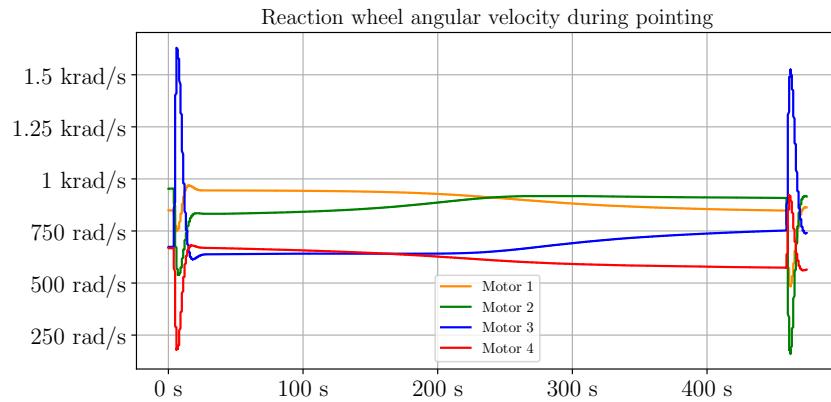


Figure 10.4: Graph showing the angular velocity of each motor during pointing.

10.2.2 Compliance of requirement

As seen when in pointing mode, the systems accuracy is within 5.1 degrees. And the settling time is well within the 132 seconds requirement. Furthermore, the step responses overshoot is within the 5.1 degrees accuracy requirement, meaning it is also within the 14.9 degrees required by the specification of the manoeuvre. The maximum angular velocity of the reaction

wheels is:

$$\begin{aligned}\omega_{1,max} &= 968.8 \text{ rad/s} \\ \omega_{2,max} &= 935.5 \text{ rad/s} \\ \omega_{3,max} &= 1630 \text{ rad/s} \\ \omega_{4,max} &= 922.8 \text{ rad/s}\end{aligned}\tag{10.1}$$

The minimum angular velocity of the reaction wheels is:

$$\begin{aligned}\omega_{1,min} &= 484.2 \text{ rad/s} \\ \omega_{2,min} &= 160.6 \text{ rad/s} \\ \omega_{3,min} &= 613.3 \text{ rad/s} \\ \omega_{4,min} &= 179.5 \text{ rad/s}\end{aligned}\tag{10.2}$$

It can be seen that the reaction wheels does not reach saturation during the pointing period.

Thus, the system complies with the requirements.

10.3 Durability acceptance test

This acceptance tests show the systems ability to control the satellite for 10 orbits without exceeding the specifications described above as per Requirement 5.1.6.

10.3.1 Results

The results of the test can be seen in Figure 10.5.

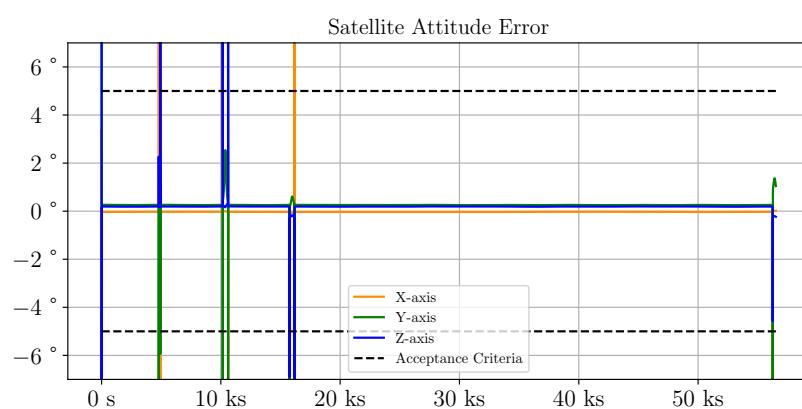


Figure 10.5: Graph showing the accuracy of the system over 10 orbits around Earth.

10.3.2 Compliance of requirement

While not immediately obvious from the graph, the system complies with the requirement of being able to control the satellite for 10 orbits around Earth, without losing control. The spikes in error are caused by the change from Nadir to pointing mode, and are therefore not a problem. As a matter of fact Figure 10.2 shows pointing mode between the second pair of spikes seen on 10.5 around ten thousand seconds.

Additionally, it is worth mentioning that the orbit of the satellite shifts so that the target is not visible for multiple orbits after around 16 thousand seconds, until the end where a new pointing mode begins, however the simulation ends whilst this is going on.

10.4 Overview of acceptance test results

Requirement	Result	Reason
Requirement 5.1.1	Passed	Described in Section 10.1, the system complies with the requirement of having a pointing accuracy within 5.1 degrees.
Requirement 5.1.2	Passed	Described in Section 10.2, the systems accuracy is within 5.1 degrees when in pointing.
Requirement 5.1.3	Passed	Described in Section 10.2, the systems settle time is well within 132 seconds.
Requirement 5.1.4	Passed	Described in Section 10.2, the overshoot is within the 14.9 degrees requirement.
Requirement 5.1.5	Passed	Described in Section 10.3, the system is able to control the satellite for the entire pointing period.
Requirement 5.1.6	Passed	Described in Section 10.3, the system is able to control the satellite for 10 orbits

Table 10.1: Acceptance test results.

Chapter 11

Discussion

In this chapter the results from the Acceptance test in Chapter 10, and any unexpected behaviour will be discussed. First the unexplained steady state error will be discussed. Then the momentum dumping will be discussed. Lastly ideas and alternatives for a physical test will be discussed.

11.1 Steady state error

From the acceptance tests it is seen in Figure 10.1 that the largest steady state error during nadir mode is 0.26° . Examining the acceptance test results for pointing mode in Figure 10.2 it is seen that the error grows to 2.54° and then shrinks during the second half of the manoeuvre.

Applying traditional control theory, this behaviour resembles a type 0 system. In relation to state space control, this seems likely, since the feedback matrix, \underline{K} , only moves the poles of the system.

Traditionally, the type of system can be changed by adding a pole at the origin, making the system a type 1 system. This would result in a steady state error of zero for a step input, and a constant steady state error for a ramp input.

In this case, Figure 10.1 would show no steady state error, and the error of Figure 10.2 would be reduced, since it resembles a parabola input.

Practically, this would be implemented by adding an integrator to the system, which could be realised by adding a variable to the state vector.

This variable would be integrated by the system, and the feedback matrix, \underline{K} , would be extended to include this variable.

Another reason for the steady state error could be that linear state space models will always regulate x to zero. Since what should be regulated to zero is the error, the state vector should be changed to include the error. This can be done by defining a new state vector which is given as Equation 11.1.

$$e = x - x_{ref} \quad (11.1)$$

x	Original state vector
x_{ref}	Reference state vector

This would result in a new state space model, which would be given as Equation 11.2.

$$\dot{e} = \underline{\mathbf{A}}x + \underline{\mathbf{B}}u \quad (11.2)$$

Since the state vector is defined as $e = x - x_{ref}$, Equation 11.2 can be rewritten as Equation 11.3.

$$\dot{e} = \underline{\mathbf{A}}(e + x_{ref}) + \underline{\mathbf{B}}u \quad (11.3)$$

The input vector u can be split up into two parts. This can be written as Equation 11.4.

$$u = u_k - u_{ref} \quad (11.4)$$

u	Original input vector
u_k	Control input vector
u_{ref}	Reference input vector

Equation 11.3 can then be rewritten as Equation 11.5.

$$\dot{e} = \underline{\mathbf{A}}e + \underline{\mathbf{A}}x_{ref} + \underline{\mathbf{B}}u_{ref} + \underline{\mathbf{B}}u_k \quad (11.5)$$

It is seen that to get Equation 11.5 to be on the form of a standard state space model, the $\underline{\mathbf{A}}x_{ref} + \underline{\mathbf{B}}u_{ref} = 0$

From this Equation 11.6 can be written.

$$-\underline{\mathbf{A}}x_{ref} = \underline{\mathbf{B}}u_{ref} \quad (11.6)$$

To get a solution for u_{ref} , an inverse of $\underline{\mathbf{B}}$ is needed. This can be done using the Moore-Penrose pseudo inverse. This will result in Equation 11.7.

$$u_{ref} = -\underline{\mathbf{B}}^{-1}\underline{\mathbf{A}}x_{ref} \quad (11.7)$$

This will then be implemented as can be seen in Figure 11.1.

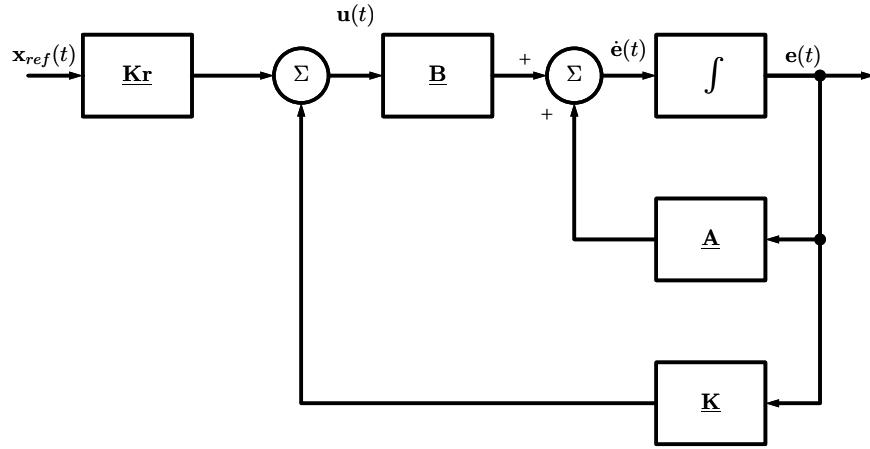


Figure 11.1: State space model with error

Where \underline{K} is the original K matrix and \underline{K}_r is given as $(-\underline{B}^{-1} \underline{A})$.

11.2 Momentum dumping

The non-symmetric nature of the disturbances in the system, will cause the reaction wheels to spin up or down over time. To avoid the reaction wheels ending up in the non-linear part, the momentum needs to be dumped. This can be done using the magnetorquers, since they are able to dump momentum without saturating.

By applying a torque with the magnetorquers which is equal but opposite to the torque from the reaction wheels, while they are spinning up or down to get to the original operating point, the satellite attitude should not be affected by this momentum dumping.

This can be implemented in the state space model by adding a torque from the magnetorquers, τ_m , to the input vector, u , and adding the corresponding rows to the \underline{B} matrix. The angular momentum of the reaction wheels, L_w , would also need to be added to the state vector, x . This was removed from the state vector in section 8.6.3 which made the system controllable.

However, adding magnetorquers to the system would make the system controllable again, because the system would be able to control the angular momentum of the reaction wheels without changing the angular momentum of the satellite.

11.3 Physical test

The system has been tested in simulation, but has not been tested physically.

This is primarily because of the reasons described in chapter 6

If the system were to be tested physically, there would be a few ways to realise this. One way would be to alter and finetune the described testbed, to minimize the disturbances caused by

the current setup. Another way would be to test individual axes one by one, by hanging the satellite from a long thin wire, and then applying a torque to the satellite. This would be a simple test, but it would not be able to test the system as a whole. A third way would be to test the system by placing it in water, whilst making sure that the currents of both water and air are as close to none as possible.

All these tests environments do however still suffer from the fact that the disturbances on earth are so small, that it is impossible eliminate them completely nor would it be possible to model them accurately enough to compensate for them.

This is also why commercial satellites' ADCS's are not physically tested, but only tested in simulation [42].

In the end, the best way to test the system would be to launch it into space, and test it there. But before this is done, it would probably be a good idea to test the system as good as possibly on earth, at least enough to know that the basics of the control system work as intended.

Chapter 12

Conclusion

The purpose of this project was to create a control system, such that a CubeSat with a camera payload, could point at a point of interest, in this project Aalborg Municipality, using reaction wheels.

Through the pre-analysis the orbit of the satellite was determined, and the different reference frames used in the project were described.

In the technical analysis, topics such as: disturbance torques, the pointing precision, the change from nadir to pointing mode were described, as well as the parametrization of the attitude. This section about parametrization included descriptions of Euler angles, Euler axis-angle and quaternions. It was also determined, that the attitude should be parameterized with quaternions, as they do not suffer from singularities.

From the technical analysis, the requirements for the system was determined.

In the design phase, the kinematic and dynamic models of the system were described.

From this a linear state space model of the system were defined, and LQR was used in order to calculate a feedback gain matrix, which places the closed loop poles at the desired locations and thus fulfilling the system requirements.

A DC motor model was also derived, and a PI controller was designed for the control loop.

The linear state space model and the DC motor model were then combined into a final control system, which was simulated in Simulink with the AAUSAT Simulation Library, and it can be concluded that the designed control system passes all the technical requirements.

Bibliography

- [1] Elizabeth Howell. *CubeSats: Tiny Payloads, Huge Benefits for Space Research*. en. June 2018. URL: <https://www.space.com/34324-cubesats.html> (visited on 05/23/2022).
- [2] NASA. *Orbits and Kepler's Laws*. June 2008. URL: <https://solarsystem.nasa.gov/resources/310/orbits-and-keplers-laws> (visited on 05/15/2023).
- [3] Holli Riebeek. *Catalog of Earth Satellite Orbits*. en. Sept. 2009. URL: <https://earthobservatory.nasa.gov/features/OrbitsCatalog> (visited on 05/15/2023).
- [4] ESA. *Types of orbits*. en. Mar. 2020. URL: https://www.esa.int/Enabling_Support/Space_Transportation/Types_of_orbits (visited on 05/15/2023).
- [5] SATCAT. URL: <https://celestrak.org/satcat/table-satcat.php?NAME=AAUSAT&PAYLOAD=1&MAX=500>.
- [6] Roger B. Bate; Donald D. Mueller; Jerry E. White. *Fundamentals of Astrodynamics*. Dover Publications, INC., 1971.
- [7] By Alan Buis Laboratory NASA's Jet Propulsion. *Milankovitch (Orbital) Cycles and Their Role in Earth's Climate*. URL: <https://climate.nasa.gov/news/2948/milankovitch-orbital-cycles-and-their-role-in-earths-climate> (visited on 05/22/2023).
- [8] Shane Ross. *Satellite Reference Frames, Viewing Geometry, and Mission Analysis*. June 2021. URL: <https://www.youtube.com/watch?v=tNny-tt7-0I>.
- [9] *Solid Cube Problem*. URL: https://campus.mst.edu/physics/courses/409/Problem-Solutions/HW#10/HW10_prob4_SolidCube.pdf.
- [10] J. Peraire; S. Widnall. *Rigid Body Dynamics: The Inertia Tensor*. Tech. rep. MIT, 2008.
- [11] Niccoló Bellini. "Magnetic Actuators for Nanosatellite Attitude Control". MA thesis. University of Bologna, 2014.
- [12] Robert A. Braeunig. "Rocket Propulsion". In: *Rocket & Space Technology* (2012).
- [13] Matt Nehrens; Matt Sorgenfrei. "A Comparison of Thruster Implementation Strategies for a Deep Space Nanosatellite". In: *American Institute of Aeronautics and Astronautics* (2014).
- [14] NASA. "Reaction/Momentum Wheel". In: *NASA Spinoff* (1997).
- [15] Jean-Francois Tregouelt; Denis Arzeiler; Dimitri Peaucelle; Christelle Pittet; Luca Zaccarian. "Reaction wheels desaturation using magnetorquers and static input allocation". In: *HAL open science* (2018).
- [16] Michael Tolstoj. "Analysis of disturbance torques on satellites in low-earth orbit based upon grace". MA thesis. RWTH Aachen University, 2017.

- [17] Scott R. Starin; John Eterno. "Attitude Determination and Control Systems". In: *NASA* (2011).
- [18] Paweł Zagórski. "Modeling disturbances influencing an Earth-orbiting satellite". In: *AGH University of Science and Technology* (2012).
- [19] *Drag Coefficient Overview Equation What is Drag in Physics?* 2022. URL: <https://study.com/academy/lesson/drag-coefficient-overview-equation.html>.
- [20] Brian Gasberg Thomsen; Jens Nielsen. "CubeSat Sliding Mode Attitude Control". MA thesis. Aalborg University, 2016.
- [21] *How strong is Earth's magnetic field in space?* July 2016. URL: <https://physics.stackexchange.com/questions/267158/how-strong-is-earths-magnetic-field-in-space>.
- [22] James R. Wertz. *Spacecraft attitude determination and control*. eng. Repr. Astrophysics and space science library ; vol 73. Dordrecht: Kluwer Academic Publ., 1994. ISBN: 9027709599.
- [23] Kasper Fuglsang Jensen; Kasper Vinther. "Attitude Determination and Control System for AAUSAT3". MA thesis. Aalborg University, 2010.
- [24] Raspberry Pi. *Raspberry Pi High Quality Camera*. Tech. rep. Raspberry Pi Ltd., 2023.
- [25] Nathan Bergey. *Cubesat Camera Field of View*. Mar. 2016.
- [26] Eric W. Weisstein. "Euler Angles". In: *Mathworld – A Wolfram Web Resource* (). URL: <http://mathworld.wolfram.com/EulerAngles.html>.
- [27] *Maths - Axis Angle*. URL: <https://www.euclideanspace.com/maths/geometry/rotations/axisAngle/index.htm>.
- [28] Grant Sanderson; Ben Eater. *Visualizing quaternionsquaternions*. URL: <https://eater.net/quaternions>.
- [29] Gasberg Brian. "Space Simulator AAUSAT3". In: (2018). URL: https://github.com/aausat/space_simulator.
- [30] Shiyu Zhao. *Time Derivative of Rotation Matrices*. Tech. rep. Department of Automatic Control and Systems Engineering. University of Sheffield, UK, 2016.
- [31] EC 9.2 flat Ø10 mm, brushless, 0.5 Watt, sensorless. URL: <https://www.maxongroup.com/maxon/view/product/371119>.
- [32] S. Yaacob T. Vil. Cherd R. Daud. "Optimal control of a tetrahedral configured reaction wheels for Quaternion rotation based on rigid satellite model". In: (2019). URL: <https://iopscience.iop.org/article/10.1088/1757-899X/670/1/012033>.
- [33] Anders Persson. "How do we understand the Coriolis Force". In: *European Centre for Medium-Range Weather Forecasts* (1998).
- [34] Derek Rowell. *State-Space Representation of LTI Systems*. Tech. rep. MIT, 2002.
- [35] Mogens Blanke ; Martin Birkelund Larsen. *Satellite Dynamics and Control in a Quaternion Formulation (2nd edition)*. Tech. rep. DTU, 2010.
- [36] *PI Control of a DC Motor*. URL: https://ctms.engin.umich.edu/CTMS/index.php?aux=Activities_DCmotorB.
- [37] Xu Chen. *Discretization of Continuous-time Transfer-function Systems*. Tech. rep. University of Washington, 2020.
- [38] Robert L. Williams; Douglas A. Lawrence. *Linear state-space control systems*. John Wiley & Sons, Inc., 2007.

- [39] Christopher Lum. *Introduction to Linear Quadratic Regulator (LQR) Control*. 2019. URL: <https://www.youtube.com/watch?v=wEevt2a4SKI&t=3s>.
- [40] Arthur E. Bryson Jr. ; Yu-Chi Ho. *Applied Optimal Control: Optimization, Estimation and Control 1st Edition*. CRC Press, 1975.
- [41] *Discretization of Continuous-time Systems*. URL: [http://www.adjutojunior.com.br/control_processos/slides_chapter8_DISCRETIZATION_OF_CONTINUOS_SYSTEMS.pdf](http://www.adjutojunior.com.br/control_processos/slides_chapter8_DISCRETIZATION_OF_CONTINUOUS_SYSTEMS.pdf).
- [42] Eric A. Basset. "Test and Verification of a CubeSat Attitude Determination and Control System in Variable Magnetic Fields". MA thesis. Air Force Institute of Technology, 2016.

Appendix A

Measuringjournals for the requirement tests

These are measuring journals for the tests, that are done to test the requirements in Chapter 7.

A.1 Nadir mode Tests

This measuring journal documents the Nadir mode test, which is described in section 7.1.

A.1.1 Test procedure

The test will be done using Matlab/Simulink, with the reference set to Nadir mode, the simulation will be run for one revolution around earth or approximately 5700 seconds. The reference attitude is therefore changing with $\omega_s = 0.0011 \text{ rad/s}$.

A.1.2 Success criteria

The requirement is fulfilled if the prototype can be in Nadir mode with a precision of $\pm 5.1^\circ$, as described in Requirement 5.1.1.

A.1.3 Results

The results of the test can be seen in Figure A.1.

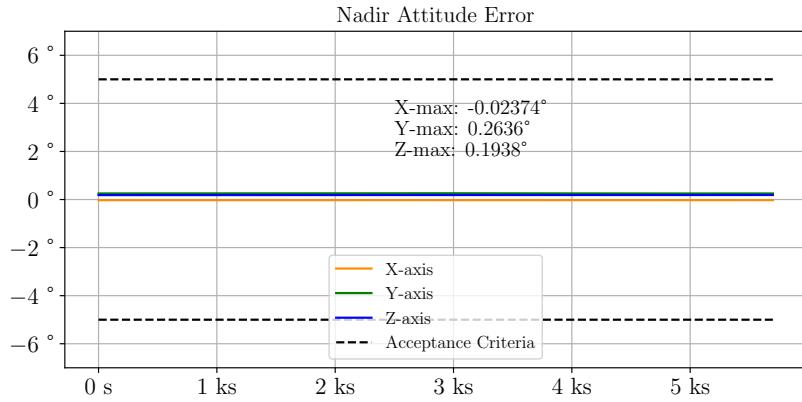


Figure A.1: Graph showing the pointing accuracy of the system.

A.1.4 Compliance of requirement

As seen the system complies with the requirement of having a pointing accuracy within 5.1 degrees.

Thus, the system complies with the requirements.

A.2 Pointing mode Tests

This measuring journal documents the Pointing mode test, which is described in section 7.2.

A.2.1 Test procedure

The test is performed by simulating the pointing mode procedure, starting at the nadir mode time just before the change to pointing mode. The simulation is run for about 700 s and the results are analyzed, until the satellite should be back in nadir mode. After the simulation the results are analyzed to determine whether the requirements are fulfilled.

A.2.2 Success criteria

The requirement is fulfilled if the satellite is able to follow the reference trajectory within the specified tolerance of $\pm 5.1^\circ$. After settling, which should be within the requirement of 132 s, the overshoot is also observed to make sure it is within the requirement of 7.89° .

Lastly it is observed whether the satellite is able to stay in pointing mode for the required time without reaching saturation from the disturbances.

A.2.3 Results

The results of the test can be seen in Figures A.2, A.3 and A.3.

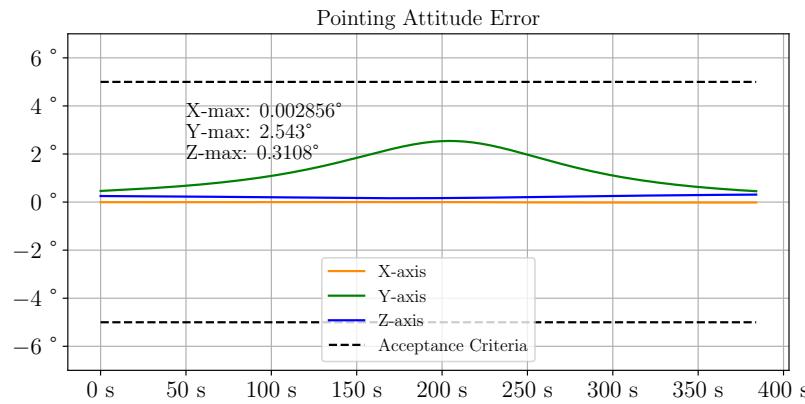


Figure A.2: Graph showing the pointing accuracy of the system after settling.

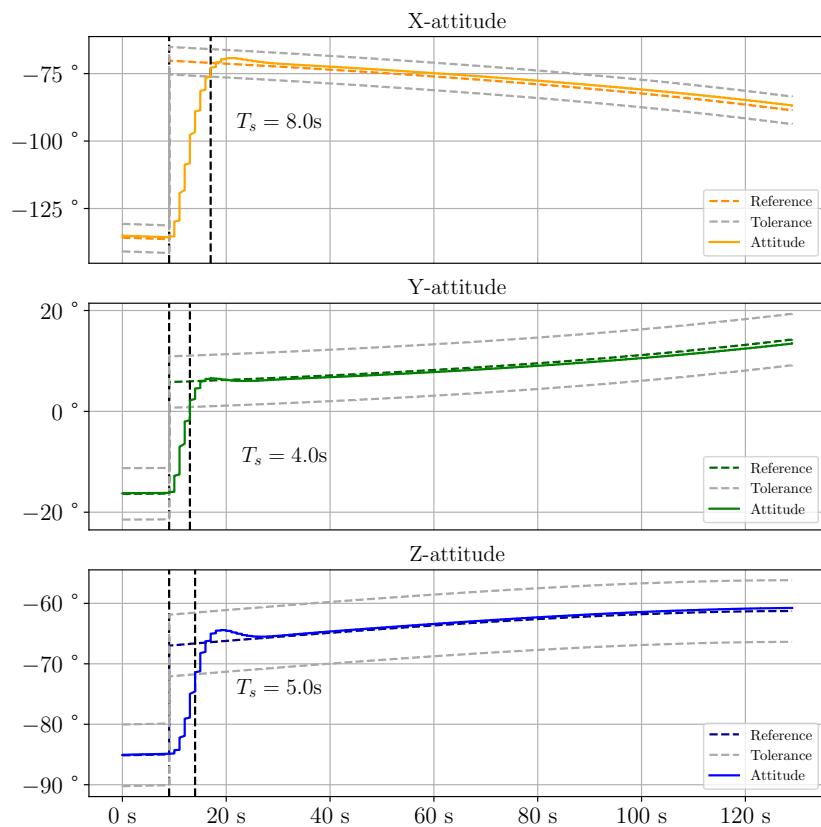


Figure A.3: Graph showing the step response of the system.

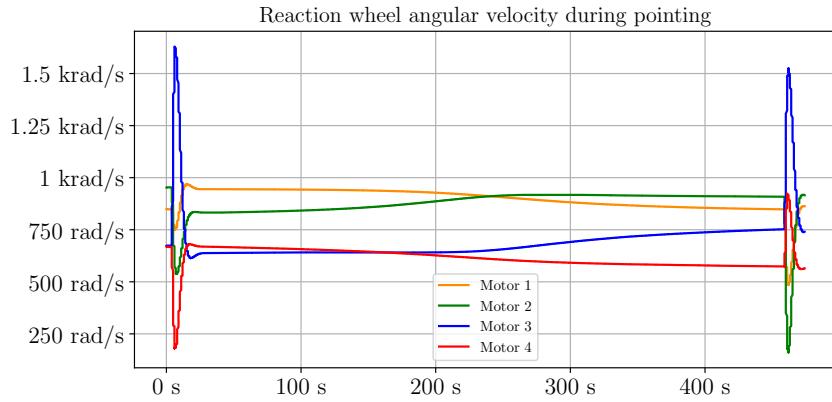


Figure A.4: Graph showing the angular velocity of each motor during pointing.

A.2.4 Compliance of requirement

As seen when in pointing mode, the systems accuracy is within 5.1 degrees. And the settling time is well within the 132 seconds requirement. Furthermore, the step responses overshoot is within the 5.1 degrees accuracy requirement, meaning it is also within the 14.9 degrees required by the specification of the manoeuvre. The maximum angular velocity of the reaction wheels is:

$$\begin{aligned}\omega_{1,max} &= 968.8 \text{ rad/s} \\ \omega_{2,max} &= 935.5 \text{ rad/s} \\ \omega_{3,max} &= 1630 \text{ rad/s} \\ \omega_{4,max} &= 922.8 \text{ rad/s}\end{aligned}\tag{A.1}$$

The minimum angular velocity of the reaction wheels is:

$$\begin{aligned}\omega_{1,min} &= 484.2 \text{ rad/s} \\ \omega_{2,max} &= 160.6 \text{ rad/s} \\ \omega_{3,max} &= 613.3 \text{ rad/s} \\ \omega_{4,max} &= 179.5 \text{ rad/s}\end{aligned}\tag{A.2}$$

It can be seen that the reaction wheels does not reach saturation during the pointing period.

Thus, the system complies with the requirements.

A.3 Pointing mode Tests

This measuring journal documents the Pointing mode test, which is described in section 7.2.

A.3.1 Test procedure

The simulation is run for a total of 10 orbits, which is about 16 hours as per Requirement 5.1.6.

The following will be measured:

- The attitude reference.
- The angle error.
- The satellite angular velocity.
- The reaction wheel angular velocity.
- The disturbances.

A.3.2 Success criteria

From these measurements the requirements will be deemed fulfilled or failed based on the tolerances given in Chapter 5.

A.3.3 Results

The results of the test can be seen in Figure A.5.

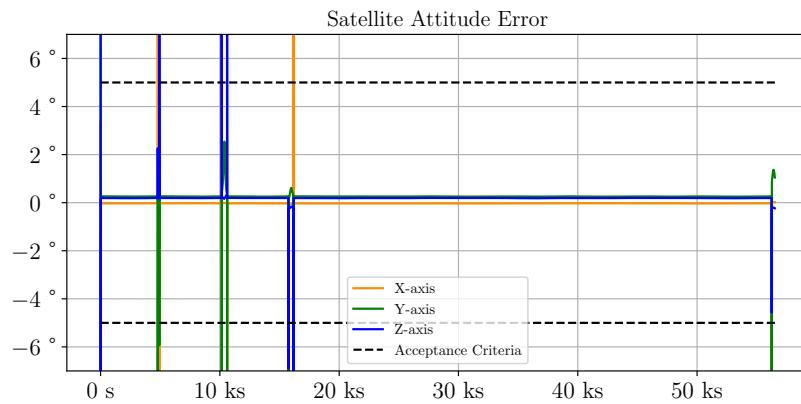


Figure A.5: Graph showing the accuracy of the system over 10 orbits around Earth.

A.3.4 Compliance of requirement

While not immediately obvious from the graph, the system complies with the requirement of being able to control the satellite for 10 orbits around Earth, without losing control. The spikes in error are caused by the change from Nadir to pointing mode, and are therefore not a problem. As a matter of fact Figure A.2 shows pointing mode between the second pair of spikes seen on A.5 around ten thousand seconds.

Additionally, it is worth mentioning that the orbit of the satellite shifts so that the target is not visible for multiple orbits after around 16 thousand seconds, until the end where a new pointing mode begins, however the simulation ends whilst this is going on.

Appendix B

Measuringjournals for other tests

These are measuring journals for other tests used in the project, that are not done to test the requirements.

B.1 Atmospheric density

The atmospheric density as a function of altitude is shown in B.1.

ALTITUDE (KM)	MEAN KINETIC TEMPERATURE (°K)	DENSITY (kg/m ³)			SCALE HEIGHT (KM)
		MINIMUM	MEAN	MAXIMUM	
0	288.2		$1.225 \times 10^{+0}$		8.44
25	221.7		3.899×10^{-2}		6.49
30	230.7		1.774×10^{-2}		6.75
35	241.5		8.279×10^{-3}		7.07
40	255.3		3.972×10^{-3}		7.47
45	267.7		1.995×10^{-3}		7.83
50	271.6		1.057×10^{-3}		7.95
55	263.9		5.821×10^{-4}		7.73
60	249.3		3.206×10^{-4}		7.29
65	232.7		1.718×10^{-4}		6.81
70	216.2		8.770×10^{-5}		6.33
75	205.0		4.178×10^{-5}		6.00
80	195.0		1.905×10^{-5}		5.70
85	185.1		8.337×10^{-6}		5.41
90	183.8		3.396×10^{-6}		5.38
95	190.3		1.343×10^{-6}		5.74
100	203.5	3.0×10^{-7}	5.297×10^{-7}	7.4×10^{-7}	6.15
110	265.5	6.0×10^{-8}	9.661×10^{-8}	3.0×10^{-7}	8.06
120	334.5	1.0×10^{-8}	2.438×10^{-8}	6.0×10^{-8}	11.6
130	445.4	4.5×10^{-9}	8.484×10^{-9}	1.6×10^{-8}	16.1
140	549.0	2.0×10^{-9}	3.845×10^{-9}	6.0×10^{-9}	20.6
150	635.2	1.2×10^{-9}	2.070×10^{-9}	3.5×10^{-9}	24.6
160	703.1	6.5×10^{-10}	1.244×10^{-9}	2.0×10^{-9}	26.3
180	781.2	2.4×10^{-10}	5.464×10^{-10}	9.0×10^{-10}	33.2
200	859.3	1.0×10^{-10}	2.789×10^{-10}	3.2×10^{-10}	38.5
250	940.2	4.0×10^{-11}	7.248×10^{-11}	1.6×10^{-10}	46.9
300	972.8	1.6×10^{-11}	2.418×10^{-11}	8.8×10^{-11}	52.5
350	986.5	2.0×10^{-12}	9.158×10^{-12}	6.0×10^{-11}	56.4
400	992.6	3.7×10^{-13}	3.725×10^{-12}	5.0×10^{-11}	59.4
450	995.7	9.0×10^{-14}	1.585×10^{-12}	3.8×10^{-11}	62.2
500	997.3	1.3×10^{-14}	6.967×10^{-13}	3.0×10^{-11}	65.8
600	1000.0		1.454×10^{-13}		79
700	1000.0		3.614×10^{-14}		109
800	1000.0		1.170×10^{-14}		164
900	1000.0		5.245×10^{-15}		225
1000	1000.0		3.019×10^{-15}		268

Figure B.1: Atmospheric density as a function of altitude [22].