



AALBORG UNIVERSITY

STUDENT REPORT

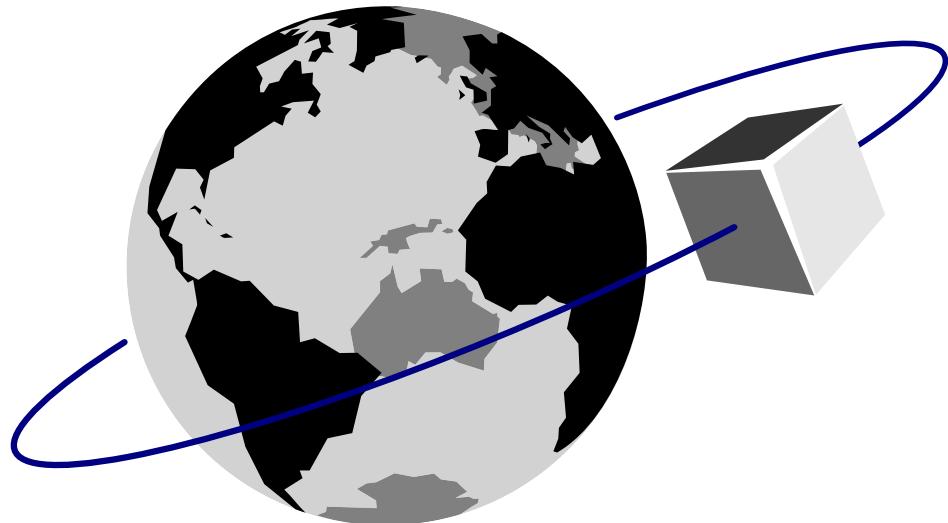
MASTERS THESIS

---

# CubeSat Sliding Mode Attitude Control

- Developing Testbed for Verification of Attitude Control Algorithms

---



*Authors:*

Brian Gasberg Thomsen  
Jens Nielsen

*Supervisor:*

Henrik Schiøler

*Co-supervisors:*

Jesper Abildgaard Larsen &  
Rasmus Holst, GomSpace





**AALBORG UNIVERSITY**  
STUDENT REPORT

**Title:** CubeSat Sliding Mode Attitude Control  
- Developing Testbed for Verification of  
Attitude Control Algorithms

**Project period:** September 2015 - June 2016

**Project group:** 1030

**Group members:**

Brian Gasberg Thomsen  
Jens Nielsen

**Supervisor:**

Henrik Schiøler

**Co-supervisors:**

Jesper Abildgaard Larsen, GomSpace  
Rasmus Holst, GomSpace

**Pages:** 128

**Finished** 2<sup>nd</sup> of June 2016

Department of Electronic Systems  
Fredrik Bajers Vej 7  
9220 Aalborg Ø  
Telephone: 99 40 86 00  
<http://es.aau.dk>

**Abstract:**

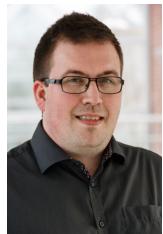
In this thesis an attitude control system has been designed for a CubeSat satellite. The satellite uses 4 reaction wheels in a tetrahedron configuration. Models for the satellite, disturbances and actuators are derived and in term used for simulation and design of the control algorithms. A quaternion based sliding mode control algorithm which compensates for the Coriolis torques from the reaction wheels is designed and simulated. A linear attitude control algorithm is designed and simulated for performance comparison against the sliding mode regulator. An attitude control testbed is developed and built during this thesis and is in term used in the test and verification of the control algorithms. The attitude control testbed is designed as a hardware in the loop component for use together with Matlab/Simulink making three-dimensional attitude control testing available. This is obtained through the use of the Robot Operating System (ROS).

*The contents of this report is freely accessible, however publication (with source references) is only allowed upon agreement with the authors.*



---

## Group Members



Brian Gasberg Thomsen

---



Jens Nielsen

---



---

# Preface

This thesis is written as part of the Master of Science in Engineering - Control and Automation at Aalborg University (AAU), Denmark. The thesis is written during the period September 2015 to June 2016.

The authors would like to thank, Associate Professor Henrik Schiøler, for his supervising during the work of the thesis. In addition the authors would like to thank Jesper Abildgaard Larsen, GomSpace and Rasmus Holst, GomSpace for their supervision and Jesper Dejgaard Pedersen for his help fabricating the mechanical parts of the attitude control testbed designed during this thesis.



---

# Table of Contents

<b>Group Members</b>	<b>i</b>
<b>Preface</b>	<b>iii</b>
<b>Table of Contents</b>	<b>1</b>
<b>Abbreviations</b>	<b>3</b>
<b>List of Figures</b>	<b>6</b>
<b>List of Tables</b>	<b>9</b>
<b>1 Introduction, Mission and Problem Statement</b>	<b>11</b>
1.1 Introduction . . . . .	11
1.2 ALPHASAT Mission . . . . .	12
1.3 Structure of the Thesis . . . . .	12
<b>2 System Overview</b>	<b>15</b>
2.1 ACS Modes . . . . .	15
2.2 Operating the ACS Modes . . . . .	17
2.3 Summary . . . . .	18
<b>3 Requirements</b>	<b>19</b>
3.1 Attitude Control System Functional Requirements . . . . .	19
3.2 ACS Mode Requirements . . . . .	19
3.3 Attitude Control Testbed Requirements . . . . .	22
3.4 Scope of the Thesis . . . . .	23
<b>4 Orbit Description and Reference Frames</b>	<b>25</b>
4.1 Keplerian Orbits . . . . .	25
4.2 Simplified General Perturbation model and The Two Line Element . . . . .	27
4.3 Reference Frames . . . . .	28
<b>5 Spacecraft and Disturbance Modelling</b>	<b>35</b>
5.1 Satellite Kinematic Equation . . . . .	35
5.2 Satellite Dynamic Equation . . . . .	36
5.3 Disturbances . . . . .	37
5.4 Actuator Models . . . . .	43

---

**TABLE OF CONTENTS**

---

5.5 Summary . . . . .	48
<b>6 Simulation Environment</b>	<b>49</b>
6.1 Simulink Implementation . . . . .	49
6.2 The AAUSAT Library . . . . .	50
6.3 Spacecraft Dynamics Model . . . . .	52
6.4 Actuator Model . . . . .	53
6.5 Attitude Estimator Emulator . . . . .	54
6.6 Target Reference Generator . . . . .	54
6.7 Simulation Setup . . . . .	56
<b>7 Attitude Control System</b>	<b>59</b>
7.1 Linear Regulator . . . . .	59
7.2 Sliding Mode Control . . . . .	67
<b>8 Attitude Control Testbed</b>	<b>77</b>
8.1 System Design . . . . .	78
8.2 Attitude Determination . . . . .	78
8.3 Simulation . . . . .	86
8.4 Mechanical Design . . . . .	88
8.5 Hardware Design . . . . .	90
8.6 Software Design . . . . .	94
8.7 Summary . . . . .	98
<b>9 Acceptance Test</b>	<b>99</b>
9.1 Attitude Control System (ACS) Testbed . . . . .	99
9.2 Control algorithm . . . . .	100
<b>10 Closure</b>	<b>103</b>
10.1 Conclusion . . . . .	103
10.2 Future Developments . . . . .	104
<b>Bibliography</b>	<b>105</b>
<b>Appendices</b>	<b>106</b>
<b>A Quaternions and Rotations</b>	<b>107</b>
A.1 Quaternions . . . . .	107
A.2 Quaternion Dynamics . . . . .	109
<b>B Testbed Electronics</b>	<b>111</b>
B.1 Main Board . . . . .	111
B.2 Actuator Board . . . . .	114
<b>C Attachments</b>	<b>119</b>

---

# Abbreviations

AAU . . . . .	Aalborg University
ACS . . . . .	Attitude Control System
ADCS . . . . .	Attitude Determination and Control System
ADS . . . . .	Attitude Determination System
API . . . . .	Application Programming Interface
BLDC . . . . .	Brushless DC Motor
BRF . . . . .	Body Reference Frame
CAD . . . . .	Computer-Aided Design
CoM . . . . .	Centre of Mass
CRF . . . . .	Control Reference Frame
DC . . . . .	DC Motor
ECEF . . . . .	Earth Centred Earth Fixed
ECI . . . . .	Earth Centred Inertial
EKF . . . . .	Extended Kalman Filter
ENU . . . . .	East North Up
ENZ . . . . .	East-North-Zenith
ESD . . . . .	Electro Static Discharge
EVD . . . . .	Eigenvalue Decomposition
FPU . . . . .	Floating Point Unit
FSM . . . . .	Finite State Machine
GC . . . . .	Geometric Centre

GRF . . . . .	Global Reference Frame
HIL . . . . .	Hardware In the Loop
IMU . . . . .	Inertial Measurement Unit
ISS . . . . .	International Space Station
JD . . . . .	Julian Date
KF . . . . .	Kalman Filter
LEO . . . . .	Low Earth Orbit
LEOP . . . . .	Launch and Early Orbit Phase
LQR . . . . .	Linear Quadratic Regulator
MCU . . . . .	Microcontroller Unit
MEKF . . . . .	Multiplicative Extended Kalman Filter
MoI . . . . .	Moment of Inertia
NORAD . . . . .	North American Aerospace Defense Command
ORF . . . . .	Orbit Reference Frame
PCB . . . . .	Printed Circuit Board
PCU . . . . .	Power Conditioning Unit
PI . . . . .	Proportional Integral
PWM . . . . .	Pulse Width Modulation
RAAN . . . . .	Right Ascension of the Ascending Node
RISC . . . . .	Reduced Instruction Set Computing
ROS . . . . .	Robot Operating System
RPM . . . . .	Rotations Per Minute
RTOS . . . . .	Real-Time Operating System
RWRF . . . . .	Reaction Wheel Reference Frame
SGP4 . . . . .	Simplified General Perturbation
SMC . . . . .	Sliding Mode Control
THCS . . . . .	Topocentric Horizon Coordinate System
TLE . . . . .	Two Line Element
TRF . . . . .	Target Reference Frame
UI . . . . .	User Interface

---

# Nomenclature

## Notation

In the following a short description on the notation used in this thesis is given.

### Mathematical:

Matrices:	<u>R</u>
Vectors:	$\bar{r}$
Identity Matrix:	<u>I</u>
Inertia Matrix:	<u>J</u>
Quaternions:	<u>q</u>
Matrix of ones:	<u>1</u>
Matrix of zeros:	<u>0</u>

### Reference frames:

Earth centered inertial reference frame :	$i$
Earth centered Earth fixed reference frame:	$e$
Orbit reference frame	$o$
Target reference frame:	$t$
Body reference frame :	$b$
Control reference frame:	$c$
Global reference frame:	$g$

A vector defined in some reference frame will be denoted  ${}^o\bar{r}$  if for example  $\bar{r}$  is in the the ORF frame. Rotations given by rotation matrices or quaternions are denoted by:  ${}_i^o\mathbf{q}$  or  ${}_i^o\underline{\mathbf{R}}$  for a rotation from the ECI frame to the ORF frame.

---

# List of Figures

2.1	System blockdiagram of the ACS . . . . .	15
2.2	A generel state transition diagram of the ACS . . . . .	17
3.1	Shows the satellite orientation when in Nadir mode where it is seen that the satellite is able to switch the orientation between two payload antennas in $-x$ - and $y$ -axis. . . . .	20
3.2	Shows the target tracking with the change in attitude. . . . .	21
4.1	Geometric interpretation of a keplerian orbit. $F_1$ and $F_2$ is the two foci of the ellipse, where in this case, the Earth is situated in $F_2$ . The ellipse is defined by its semi-minor and semi-major axis denoted $a$ and $b$ in the figure. . . . .	25
4.2	Shows a satellite in orbit where the orbit parameters is put in to perspective.	26
4.3	Shows the Earth Centred Inertial (ECI) reference frame to the left with the ${}^i x$ pointing towards the mean Equinox. To the right the Earth Centred Earth Fixed (ECEF) reference frame is where the ${}^e x$ -axis points to where the fixed point on the Prime Meridian. . . . .	28
4.4	Shows the orientation of the Orbit Reference Frame (ORF) where the ${}^o \bar{z}$ is always pointing Nadir. . . . .	30
4.5	Shows the THCS given by the span of $\{\hat{i}, \hat{y}\}$ with the vector ${}^i \bar{R}_\Delta$ pointing from the target (T) in the direction of the spacecraft (B). . . . .	31
4.6	Shows the Target Reference Frame (TRF) located in the Centre of Mass (CoM) of the satellite body B, where the ${}^t z$ -axis points to the target T and ${}^t x$ -axis points in the direction of flight. . . . .	31
4.7	Shows the CubeSat Body Reference Frame (BRF) which is centred in the CoM. . . . .	33
5.1	The vectors involved in calculating the aerodynamic drag. . . . .	39
5.2	The vectors involved in calculating the gravity gradient torque, where Geometric Centre (GC) is the geometric centre of the satellite. . . . .	40
5.3	Absorption . . . . .	41
5.4	Reflection . . . . .	41
5.5	Diffusion . . . . .	41
5.6	Shows the electric circuit of the armature to the left and the free-body diagram of the rotor to the right. . . . .	44
5.7	Shows direction of each reaction wheel in the tetrahedron configuration. . .	46

6.1	Shows the structure of the simulation environment, where all the system models for spacecraft dynamics and orbit propagation is collected in a single dynamic block. . . . .	49
6.2	Shows the environment blocks of the ALPHASAT, with the spacecraft dynamics, orbit models and the disturbances seen from the spacecraft. . . . .	51
6.3	Shows the Simulink 3D animation environment designed for intuitive illustrating satellite attitude manoeuvres. . . . .	52
6.4	Shows the Simulink block of the spacecraft system model. . . . .	52
6.5	Shows the Simulink block of the spacecraft dynamics. . . . .	53
6.6	Shows the Tetrahedron actuator model. . . . .	53
6.7	Shows the model of a simplified Brushless DC Motor (BLDC). . . . .	54
6.8	Shows the block which should emulate the uncertainties and noise from which is propagated through an estimator. . . . .	54
6.9	Shows the Tetrahedron actuator model. . . . .	55
6.10	Shows the satellite and one specific target $T_i$ where $\alpha_i$ is the inclination of the satellite over the THCS. . . . .	56
7.1	Shows a block diagram of the linear regulator setup where the output of the Linear Quadratic Regulator (LQR) is a desired control torque. . . . .	59
7.2	Shows the step response of the motor Proportional Integral (PI) regulator at 4 Hz sampling rate. . . . .	63
7.3	Shows the satellite attitude error in Euler angles, where the dashed lines represent the maximum error values. . . . .	64
7.4	Shows the satellite attitude ${}^b\mathbf{q}_s$ relative to the ECI, where the dashed lines represent the quaternion attitude reference ${}^b\mathbf{q}_{ref}$ . . . . .	64
7.5	Shows angular velocities on each motor in the Tetrahedron configuration. .	65
7.6	Shows the motor control signal. . . . .	65
7.7	Shows the satellite attitude error in Euler angles, where the dashed lines represent the maximum error values. . . . .	66
7.8	Shows the satellite attitude ${}^b\mathbf{q}_s$ relative to the ECI, where the dashed lines represent the quaternion attitude reference ${}^b\mathbf{q}_{ref}$ . . . . .	66
7.9	Shows angular velocities on each motor in the Tetrahedron configuration. .	66
7.10	Shows the motor control signal. . . . .	67
7.11	Shows how the trajectory is drawn towards the sliding surface, $s_i$ and eventually the equilibrium point. . . . .	68
7.12	Shows the satellite attitude error in Euler angles, where the dashed lines represent the maximum and minimum angle error. . . . .	72
7.13	Shows the satellite attitude ${}^b\mathbf{q}_s$ relative to the ECI, where the dashed lines represent the quaternion attitude reference ${}^b\mathbf{q}_{ref}$ . . . . .	73
7.14	Shows angular velocities on each motor in the Tetrahedron configuration. .	73
7.15	Shows the motor control signal. . . . .	73
7.16	Shows the satellite attitude error in Euler angles. . . . .	74
7.17	Shows the satellite attitude ${}^b\mathbf{q}_s$ relative to the ECI, where the dashed lines represent the quaternion attitude reference ${}^b\mathbf{q}_{ref}$ . . . . .	74
7.18	Shows angular velocities on each motor in the Tetrahedron configuration. .	75
7.19	Shows the motor control signal. . . . .	75
8.1	The testbed with the air-bearing. . . . .	78
8.2	Shows the general flow of the Kalman filter. . . . .	79
8.3	Shows the Extended Kalman Filter (EKF) simulation environment . . . . .	87

8.4	Shows the estimation error of the attitude, where the dashed lines specifies the requirements for the attitude determination. . . . .	87
8.5	Shows that the estimator becomes unstable after $\approx 90$ deg rotation. . . . .	88
8.6	Shows the quaternion attitude estimate behaviour after a quarter of a rotation. . . . .	88
8.7	Shows the CubeSat suspended inside a low mass acrylic ball in the GC, where the colour gradient illustrates the pressurised air flow and $\rho_{in}$ is the pressure inlet. . . . .	89
8.8	Shows the offset in CubeSat CoM which is affected by the gravity $f_g$ . $\tau_g$ is the torque generated by the CoM offset. . . . .	90
8.9	Block diagram of the attitude control testbed electronics hardware. . . . .	90
8.10	Blockdiagram of the main board hardware. . . . .	91
8.11	Blockdiagram of the actuator board hardware. . . . .	92
8.12	Block diagram of the general software of the testbed. . . . .	94
8.13	Block diagram of the software within the main board. . . . .	95
8.14	Actuator board state machine. . . . .	96
8.15	Block diagram of the Matlab/Simulink implementation of the attitude control testbed. . . . .	97
9.1	Attitude error magnitude, $\ \bar{\theta} - \hat{\theta}\ _2$ for a 80 degree step in yaw. . . . .	99
9.2	80 degree step in yaw, with the sliding mode control developed in Section 7.2.100	100
B.1	Main board top. . . . .	111
B.2	Main board bottom. . . . .	111
B.3	Actuator board top. . . . .	114
B.4	Actuator board bottom. . . . .	114
B.5	Actuator board assembled. . . . .	114

---

## List of Tables

2.1	ACS modes . . . . .	16
3.1	Functional requirements for the ACS. . . . .	19
3.2	Requirements for the Detumble mode. . . . .	20
3.3	Requirements for the Nadir mode. . . . .	21
3.4	Requirements for the target tracking mode. . . . .	22
3.5	Requirements for the Attitude Control testbed. . . . .	22
3.6	Requirements for the attitude determination system for the attitude control testbed. . . . .	23
4.1	The first line of the ALPHASAT Two Line Element (TLE) [Markley 14]. . .	27
4.2	The second line of the ALPHASAT TLE [Markley 14]. . . . .	27
9.1	Requirements for the target tracking mode. . . . .	100



---

# Introduction, Mission and Problem Statement

## 1.1 Introduction

The CubeSat standard saw its first light in 1999 as a collaboration between California State Polytechnical University and Stanford University Space Systems Development Laboratory where the CubeSat Design Specification [CalPoly 14] where defined. Since then, the CubeSat standard has gained significant popularity with, among others, University students and developing nations [Woellert 11]. This popularity is, in part, because of the price for launching a CubeSat in orbit which is significantly lower than traditional satellites. The lower price for putting technology in orbit and the technologic development has made picosatellites, as CubeSats, a competitive alternative to larger satellites.

The history of satellites at Aalborg University begins with the first danish satellite, the Oersted Satellite, launced in 1999. The Oersted satellites mission was to collect data in order to build a better map of the geomagnetic field surrounding the Earth. The Oersted satellite used a magnetic approach to attitude control in conjunction with its measurement boom which acted as a stabilising gravity boom [Bak 96].

After the success of the Oersted satellite and the emergence of the CubeSat standard, this was the next step at Aalborg University. This led to the AAU CubeSat, a one unit CubeSat, launched in a sun synchronous Low Earth Orbit (LEO) at an altitude of about 900 km on the first CubeSat launch. The AAU CubeSat had a camera as payload and used, as Oersted, magnetic actuation. The CubeSats that followed AAU CubeSat; AAUSAT2, AAUSAT3, AAUSAT4 and AAUSAT5 all of which where one-unit CubeSats and used magnetic actuation, although AAUSAT2 was also designed for reaction wheels.

With the technologic advancement within the electronic and space industry the possibilities for advanced payloads increases, this also increases the requirements for Attitude Determination and Control Systems (ADCSs) for many payloads. This could for instance be a highspeed communication system which inherently requires an accurate pointing of its antennas. In cases where the suite of payloads requires active pointing or advanced manoeuvres where a higher precision and slew rate are necessary; the magnetic actuation system might not be enough to satisfy the mission parameters of the payload suite.

During the Launch and Early Orbit Phase (LEOP) of the AAUSAT3 and AAUSAT4 missions; a sign error in the detumble algorithm caused the satellite to spin up instead

of spinning down. When developing attitude control algorithms the test phase of the development process is inherently difficult. To support the testphase of a spacecraft attitude control system, a testbed is to be developed. Such a testbed could have been beneficial in the testphase of the AAUSAT3 and 4 missions. For educational purposes an attitude control testbed can also provide a method of working on three-dimensional control problems.

## 1.2 ALPHASAT Mission

During this thesis an ACS will be developed for a hypothetical CubeSat satellie, in orbit, hereafter known as ALPHASAT. The launch provider is given from the International Space Station (ISS). This means that the orbit of ALPHASAT is approximately the same as the orbit of the ISS.

The mission of ALPHASAT will be to track multiple groundstations during its orbit around Earth. This is to serve the payloads aboard ALPHASAT.

The CubeSat is equipped with four reaction wheels for attitude manoeuvring and Magnetic Torquers on each side of the satellite for moment dumping of the reaction wheels and detumbling the satellite after jettison.

However the moment dumping will not be implemented in this project.

### 1.2.1 Problem Statement

One objective of the following project is to develop an attitude control system for a CubeSatm to perfor both active target pointing and Nadir pointing. In order to accomplish this, the problem is broken down into a number subproblems forming the complete problem statement.

*Develop an attitude control system for a CubeSat. The attitude control system must be able to:*

- *Nadir pointing*
- *Active pointing*
- *Reject orbit disturbances.*

Another objective of the following project is to develop an attitude control testbed. The testbed purpose is to aid in the development process of attitude control algorithms. In order to to accomplish this, the problem is broken down into a number subproblems forming the complete problem statement.

*Develop a nonlinear attitude control testbed. The testbed should be able to:*

- *Replicate frictionless rotations in tree dimensions*
- *Determine attitude*
- *Perform control manoeuvres*

In the following section the structure of the thesis is described. This is to provide an overview of the report.

## 1.3 Structure of the Thesis

**Chapter 2 - System Overview:** A general overview of an ACS is given during this chapter. This includes the different modes of operation for an ACS and the transitions between these.

**Chapter 3 - Requirements:** The requirements of both the attitude control algorithms and teststand is given in this chapter, followed by the scope of the thesis.

**Chapter 4 - Orbit Description and Reference Frames:** An orbit in terms of keplerian orbit parameters is described. After this the two-line element and its use for orbit determination is described for use in the simulation environment later in the project. The reference frames used in this project is also defined.

**Chapter 5 - Spacecraft and Disturbance Modelling:** This chapter describes all the models used for simulation and in deriving attitude control algorithms. This includes the kinematic and dynamics of the satellite, disturbance models and actuator models.

**Chapter 6 - Simulation Environment:** Describes the simulation environment used for simulating ALPHASAT. The simulation environment is built in Matlab/Simulink.

**Chapter 7 - Attitude Control System:** Describes the attitude control algorithms. Both a linear and a non-linear approach is considered.

**Chapter 8 - Attitude Control Testbed:** Describes the design of the attitude control testbed. This includes both the mechanical and the electronic hardware, the software and the attitude determination used in the testbed.

**Chapter 9 - Acceptance Test:** Acceptance tests and verification against the requirements listed in Chapter 3.

**Chapter 10 - Closure:** The conclusion of the thesis and future developments.

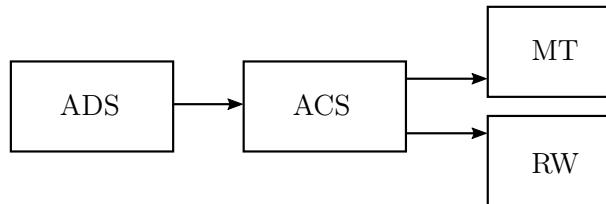


---

# System Overview

In the following chapter there will be given an overview of the ACS functions. The ACS consists of a processing unit and the actuators. The processing unit is a micro-controller which handles both the Attitude Determination System (ADS) and the ACS.

The ACS receives an attitude quaternion (see Appendix A for a description of quaternions), from the ADS. Based on this attitude quaternion, relative to the reference quaternion, the control algorithms in the ACS calculates a control input for a composition of the actuators available. A general block diagram of the ADCS circuit is shown on Figure 2.1.



**Figure 2.1:** System blockdiagram of the ACS.

Where *MT* and *RW* is respectively the magnetic Torquers and the reaction wheels.

The ADS also consists of several sensors used for attitude determination, but this is not in the scope of this thesis and are left out of the block diagram for simplicity.

## 2.1 ACS Modes

The functionality of the ACS are broken down into a set of five modes, used in different phases of the spacecraft's mission. The transitions between these mode will, to some extent, be autonomous. A subset of the ACS mode transitions will be controllable manually from a ground station during a pass or by the use of a flight planner inside the spacecraft.

The modes are listed in Table 2.1.

Mode	Description
Orbit Insertion	Immediately after jettison - Spacecraft is off
Detumble	Attenuate the change in spacecraft attitude
Nadir	Spacecraft should point at the Earth's centre
Target Tracking	Active target tracking
Contingency	A safe mode for when in eclipse or if an emergency arise

**Table 2.1:** ACS modes

Autonomous mode transitions is occurring e.g. when the spacecraft finishes the orbit insertion phase and boot up the attitude control for the fist time.

In the following the different modes is described.

### 2.1.1 Orbit Insertion

The Orbit Insertion mode is active immediately after jettison from the ISS. During the first 45 minutes after jettison the CubeSat is not allowed to generate or transmit any signal [CalPoly 14]. This means that the ACS will not be allowed to use magnetic actuation. The reaction wheels will not be spun up until the spacecraft has a tumble-rate below the threshold defined by the requirements in Chapter 3. The resulting attitude and possible tumbling rate are therefore provided by the launch vehicle, until after 45 minutes.

### 2.1.2 Detumble

When the Orbit insertion phase is finished and the spacecraft powers up, the satellite will tumble with a rate given by the ISS and the accumulated disturbances during the first phase of the spacecraft mission. This means that the spacecraft will need to perform some Detumbling.

Detumbling is performed by use of magnetic Torquers, which acts on the Earth's magnetic field to create a magnetic moment in the opposite direction. This manoeuvre will eventually slow the spacecraft's angular velocity down to a point where a mode transition, to a more advanced attitude control mode is performed.

### 2.1.3 Nadir

The Nadir mode is the mode used when the spacecraft is idle. Idle in the sense that the spacecraft does not have any targets to track, or is collecting data for a payload.

### 2.1.4 Target Tracking

The target tracking mode is the mode used when the spacecraft is performing active pointing, that is, when a specific target on the Earth surface or another target in space needs to be tracked. This can be used with all payloads aboard.

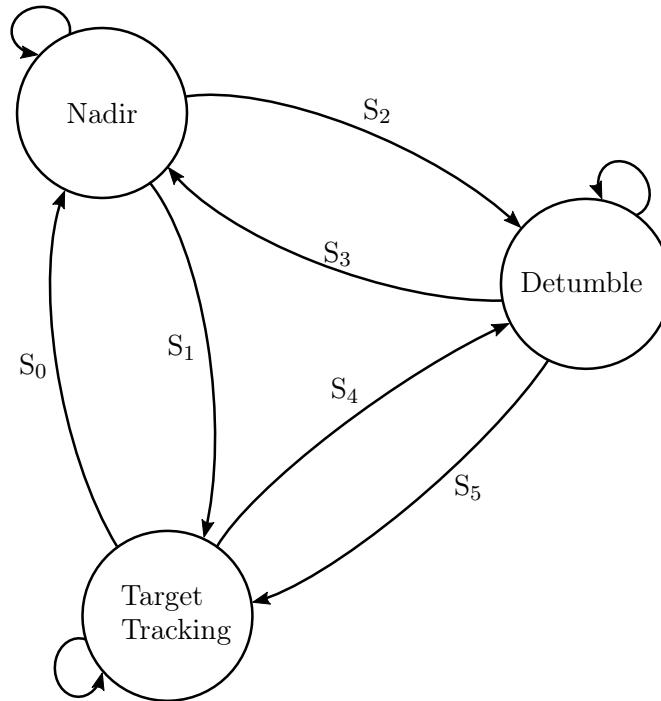
### 2.1.5 Contingency

The contingency mode is a mode used when the spacecraft experience a severe event. This event could be if an actuator partially breaks down. This mode can also be used in the event that the spacecraft ADS degrades significantly, e.g. in eclipse if the ADS relies heavily on solar sensors. For this purpose the Detumble mode are used, this is because

the Detumble controller will be based solely on magnetic actuation. This is because the mode has no moving mechanical parts which can break down, making this mode suitable for use as a contingency.

## 2.2 Operating the ACS Modes

These modes are modelled as a finite state machine where each state represents a mode. A general state transition diagram of the ACS states can be seen in Figure 2.2.



**Figure 2.2:** A generel state transition diagram of the ACS

When the spacecraft enters the LEOP it boots up in the Detumble mode. When the the spacecraft's tumble rate are below an allowed threshold, a state transition occurs, into Nadir mode - from which the spacecraft waits for a command from the ground operator on where to go from here. The state transition conditions from Figure 2.2 are listed below.

**S<sub>0</sub>** is the transition from the active target tracking mode to the Nadir mode. This transition will typically occur when a pass has ended or the object being tracked is not in line of sight any more.

**S<sub>1</sub>** is the transition from Nadir mode to the active target tracking mode. This transition occurs when the spacecraft has been idle and a target is within line of sight.

**S<sub>2</sub>** is the transition between Nadir mode and Detumble mode. This transition can occur if the the spacecraft experiences som event which triggers the contingency mode. This could be breakdown or partial breakdown of the actuators or if the ADS degrades.

**S<sub>3</sub>** is the transition from the Detumble mode to the Nadir mode. This is the transition to occur automatically when the tumble rate of the spacecraft falls below the tumble

threshold defined by the requirements. This is the default transition to happen from the Detumble mode, when the spacecraft has detumbled.

**S<sub>4</sub>** is the transition from the Target tracking mode to the Detumble mode. This transition can occur if the the spacecraft experiences som event which triggers the contingency mode. This could be breakdown or partial breakdown of the actuators or if the ADS degrades.

**S<sub>5</sub>** is the transition from the Detumble mode to the Target tracking mode. This transition is not the default behaviour when the spacecraft has been detumbled but will only happen if the ground operator has expressively commanded it.

All of these state transitions can also be triggered by both a on-board flight planner or by a ground operator.

## 2.3 Summary

In this chapter the general function and modes of behaviour has been described. This includes defining the modes; Orbit Insertion, Detumble, Target Tracking, Nadir and Contingency. These modes are controlled and modelled as finite state machine. In the following chapter the requirements for the modes defined in this will be listed.

---

# Requirements

In the following Chapter the requirements for the ACS are listed. Firstly a set of functional requirements will be listed, and here after these requirements will be broken down into sub-requirements which is necessary in order to comply with the functional requirements.

## 3.1 Attitude Control System Functional Requirements

No.	Title	Description	Parent	Children
R1	Detumbling	The ACS must be able to detumble the spacecraft.	Section 2.1.2	R1.1 & R1.2
R2	Nadir mode	The ACS must be able to perform Nadir pointing.	Section 2.1.3	R2.1, R2.2, R2.3 & R2.4
R3	Target Tracking mode	The ACS must be able to perform active pointing.	Section 2.1.4	R3.1, R3.2, R3.3 & R3.4

**Table 3.1:** Functional requirements for the ACS.

## 3.2 ACS Mode Requirements

### 3.2.1 Orbit Insertion

The spacecraft is not allowed to generate any signals during this phase [CalPoly 14]. There is no requirements for this mode.

### 3.2.2 Detumble

When the spacecraft is in Detumble mode, it is because that, either something triggered the contingency mode, or the spacecraft has just booted up from the orbit insertion phase of the mission. This requires the spacecraft to either keep the angular tumble rate at a minimum; or the spacecraft has a tumble rate that is necessary to handle. When the spacecraft leaves the orbit insertion mode, the detumble rate are given from the attitude provided from the launch vehicle together with accumulated disturbance torques during the first, at least, 45 minutes [CalPoly 14] of the LEOP.

Requirements for the Detumble mode are listed below:

### 3. REQUIREMENTS

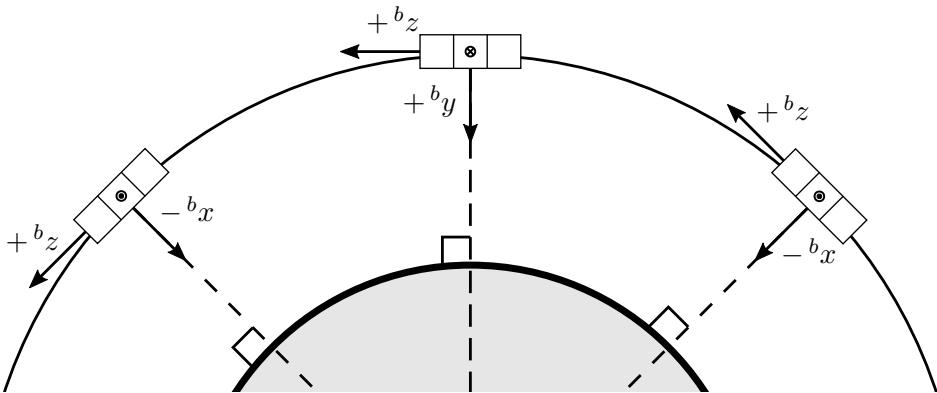
---

No.	Title	Description	Parent
R1.1	Detumble	The ACS must be able to detumble the spacecraft down to two revolutions per orbit.	Req. R1
R1.2	Performance	The ACS must be able to detumble from 10 deg/s to 0.3 deg/s in three orbits.	Req. R1

**Table 3.2:** Requirements for the Detumble mode.

#### 3.2.3 Nadir

In Nadir pointing mode the spacecraft is pointing perpendicular to the Earth surface, in effect pointing at the Earth centre. This requires an angular velocity about the spacecraft CoM which is found from the spacecraft orbital parameters. An attitude illustration of the Nadir can be seen on Figure 3.1.



**Figure 3.1:** Shows the satellite orientation when in Nadir mode where it is seen that the satellite is able to switch the orientation between two payload antennas in  $-x$ - and  $y$ -axis.

In the Nadir mode, one of the payload antennas orientated in the  $-x$ - or  $y$ -axis is defined to point in the direction of Nadir. This requires an angular velocity around the other axis in order to maintain the attitude. Because the orbit of ALPHASAT is circular the angular velocity can be described as a function of the orbital period  $P$  which is defined by the orbital parameters [Wertz 99]. The angular velocity is then given as:

$$\omega_n = \frac{2\pi}{P} \quad (3.1)$$

where

$\omega_n$	is the angular velocity required to perform Nadir pointing	[rad/s]
$P$	is the orbital period of the spacecraft	[s]

(3.2)

The range orbital period  $P$  in the lifetime of ALPHASAT is found to start from 92.56 min. at 400 km down to 90.52 min. at 300 km [Wertz 99]. The largest angular velocity required for Nadir pointing is found to be:

$$\omega_n = \max \left\{ \frac{2\pi}{92.56 \cdot 60}, \frac{2\pi}{90.52 \cdot 60} \right\} = 0.0012 \text{ [rad/s]} \quad (3.3)$$

Requirements for the Nadir pointing mode are listed below:

No.	Title	Description	Parent
R2.1	Accuracy	Must be able to point < 10 deg error.	Req. R2
R2.2	Slew rate	Must be able to change the attitude with a slew rate > 0.0012 rad/s	Req. R2
R2.3	Settling time	< 5 deg maximum motion, 1 min.	Req. R2
R2.4	Range	Must be able to attain Nadir attitude from all initial attitudes	Req. R2

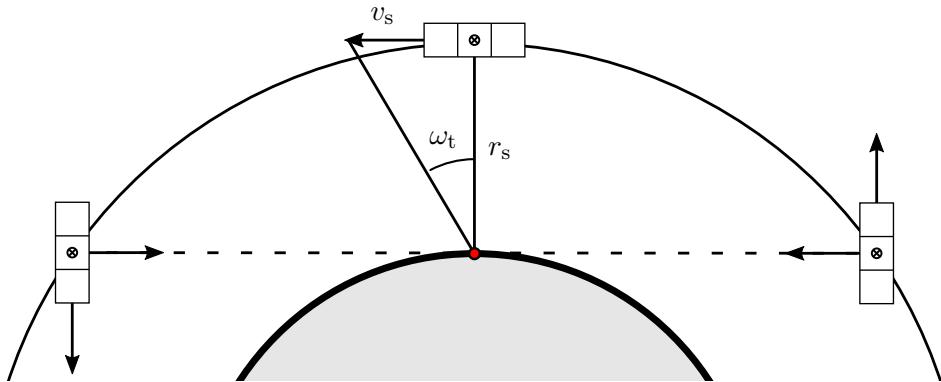
**Table 3.3:** Requirements for the Nadir mode.

### 3.2.4 Contingency

The contingency mode uses the detumble mode, so the requirements are identical.

### 3.2.5 Target Tracking

The target tracking mode is the mode used for active pointing. The mission is specified in Section 1.2 and the fastest slew rate necessary for obtaining pointing happens in the situation where the spacecraft are tracking a specific target on the Earth surface, as opposed to pointing at Nadir. The target tracking situation is illustrated in Figure 3.2.

**Figure 3.2:** Shows the target tracking with the change in attitude.

From Figure 3.2 it is seen that the angular rate of the spacecraft is a composition of the angular rate of pointing at Nadir and the angular rate necessary for tracking the target on the Earth surface. The lesser radius of the rotation sphere causes a higher angular velocity. Therefore a different set of requirements are necessary for this mode compared to the Nadir modes.

The angular velocity necessary is calculated as:

$$\omega_t = \tan^{-1} \left( \frac{v_s}{r_s} \right) \quad (3.4)$$

where

$\omega_t$	is the worst case angular velocity	[rad/s]
$v_s$	is the orbital velocity of the satellite	[m/s]
$r_s$	is the smallest target vector length	[m]

(3.5)

### 3. REQUIREMENTS

---

The largest angular velocity seen from equation 3.4 is where the minimum length of the target vector  $r_s$  is given. The orbit of ALPASAT in its lifetime will experience a minimum distance to the Earth of 300 km at a speed of 7.726 km/s which results in

$$\omega_t = \tan^{-1} \left( \frac{7.726}{300} \right) = 0.0257 \text{ [rad/s]} \quad (3.6)$$

The requirements for the target tracking mode listed below:

No.	Title	Description	Parent
R4.1	Accuracy	Must be able to point < 10 deg.	Req. R4
R4.2	Slew rate	Must be able to change the attitude with a slew rate > 0.0257 rad/s	Req. R4
R4.3	Settling time	< 5 deg max motion, 1 min.	Req. R4

---

**Table 3.4:** Requirements for the target tracking mode.

## 3.3 Attitude Control Testbed Requirements

In the following a set of requirements are listed for the attitude control testbed. First a set of functional requirements followed by requirements for the attitude determination in the testbed.

### 3.3.1 Functional Requirements

The functional requirements for the attitude control testbed is listed below. The requirements are defined in order to develop the attitude control testbed in Chapter 8.

No.	Title	Description	Children
R5.1	Hardware In the Loop (HIL)	Must be a HIL component.	
R5.2	Platform	Must be independent of PC platform.	
R5.3	3D	Must be able to perform three-dimensional attitude manoeuvres.	
R5.4	Centre of mass adjustment	Must be able adjust the centre of mass.	
R5.5	ADS	Must be able to determine the attitude of the testbed hardware.	R5.5.1
R5.6	User Interface (UI)	Must have a UI.	

---

**Table 3.5:** Requirements for the Attitude Control testbed.

### 3.3.2 Attitude Determination Requirements

No.	Title	Description	Parent
R5.5.1	Accuracy	Must be able to determine the attitude within < 5 deg.	Req. R5.5

**Table 3.6:** Requirements for the attitude determination system for the attitude control testbed.

## 3.4 Scope of the Thesis

There are two parts to this thesis; the ACS and the attitude control testbed. On the ACS side this thesis will be limited to developing control algorithms for performing active target tracking as the requirements for this is the most demanding compared to Nadir tracking. Hence this will result in control algorithms which can be used for both Nadir mode and target tracking. The detumble controller algorithm will not be designed during this thesis; the requirements for the Detumble mode will be used to determine what state of the spacecraft from which the target tracking mode must be able to regain control of the spacecraft attitude.



---

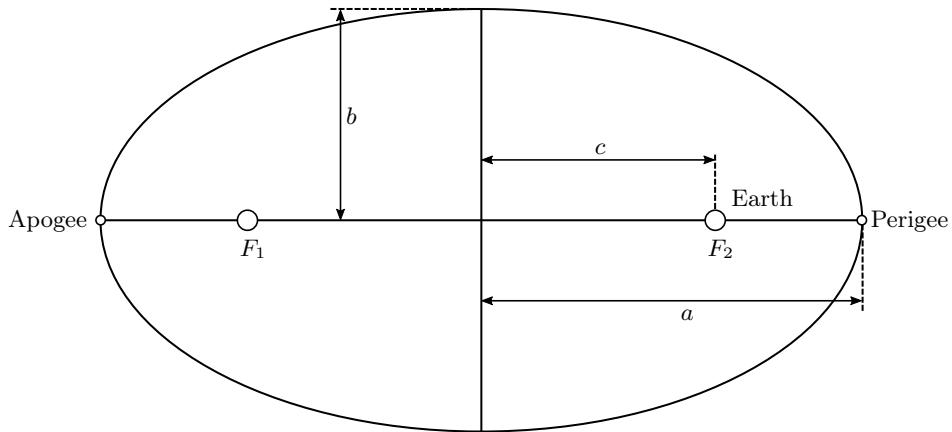
# Orbit Description and Reference Frames

In the following chapter the properties of an orbit is described. The Simplified General Perturbation (SGP4) model for calculating orbit parameters is described. A number of reference frames are presented, both orbital reference frames but also a reference frame used in the attitude determination in the attitude control testbed is presented.

## 4.1 Keplerian Orbits

Sources: [Markley 14] [Sidi 97] [Wertz 95] [Wie 98]

According to Keplers laws of planetary motion; an orbit about a planet can be described by an ellipse. The ellipse has two focal points, where the planet, of which the spacecraft is in orbit about, must lie in one. An illustration of an elliptical orbit is shown in Figure 4.1.



**Figure 4.1:** Geometric interpretation of a keplerian orbit.  $F_1$  and  $F_2$  is the two foci of the ellipse, where in this case, the Earth is situated in  $F_2$ . The ellipse is defined by its semi-minor and semi-major axis denoted  $a$  and  $b$  in the figure.

An ellipse can be defined by the ratio between the distance from the centre of the ellipse to the focal point where the Earth is located,  $c$ , and the semi-major axis of the ellipse; this ration is denoted by the eccentricity  $e$

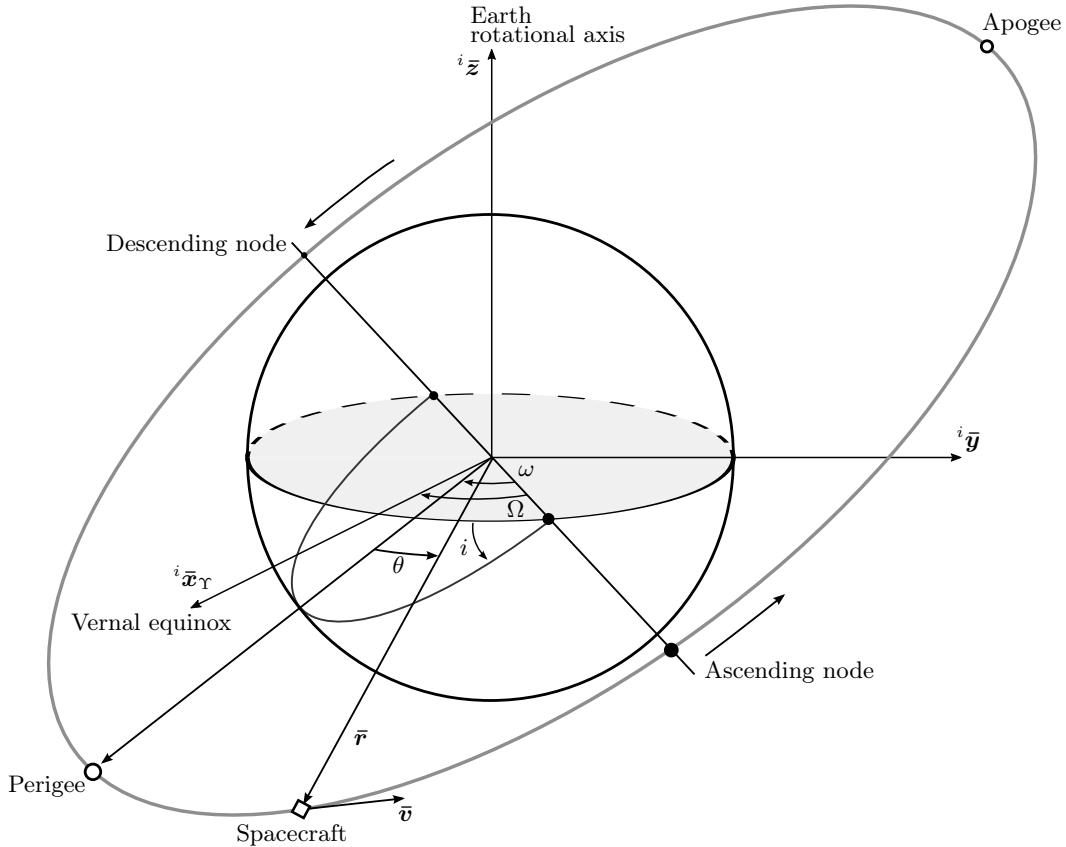
$$e = \frac{c}{a} = \frac{\sqrt{a^2 - b^2}}{a} = \sqrt{1 - \frac{b^2}{a^2}} \quad (4.1)$$

It is clear from Equation (4.1) that for a non-elliptic circle having  $a = c$ , the eccentricity is 0. For an elliptic orbit the eccentricity is  $0 < e < 1$ .

The elliptic orbit shown in Figure 4.1, can be expanded and used to describe the orbit, with a number of orbit parameters, illustrated in Figure 4.2. The orbit parameters are:

- $i$ , the inclination
- $\Omega$ , the Right Ascension of the Ascending Node (RAAN)
- $\omega$ , argument of perigee

Where the inclination  $i$  is the angle between the orbital plane and the equatorial plane. The intersecting line between the orbital plane and the equatorial plane is called the *line of nodes*, and the point where the spacecraft intersects with this travelling in north direction, is called the ascending node; the opposite node where the spacecraft is travelling south is called the descending node. The angle between the vernal equinox,  ${}^i\bar{x}_Y$ , and the line of nodes is called the RAAN,  $\Omega$ .



**Figure 4.2:** Shows a satellite in orbit where the orbit parameters is put in to perspective.

The Apogee and Perigee denotes respectively the point in orbit with the highest and lowest orbital altitude. The argument of perigee,  $\omega$ , is the angle between the line of nodes and the Perigee of an orbit. In Figure 4.2 the vectors  $\bar{r}$  and  $\bar{v}$  is in term the position and velocity vectors of the orbiting spacecraft.

## 4.2 Simplified General Perturbation model and The Two Line Element

Sources: [Markley 14] [Hoots 80]

The Simplified General Perturbation models is a set of mathematical models used to obtain a velocity and a positional vectors of an element orbiting Earth. The models are: SGP, SGP4, SDP4, SGP8 and SDP8, the SDP models are an extension of the SGP models, where the SDP models are used for deep-space satellites and the SGP models are used for near-Earth orbits.

the North American Aerospace Defense Command (NORAD) tracks all artificial satellites orbiting about the Earth and maintains a database of all these. The database of all these orbiting elements, or satellites, contains what is called a TLE, which is used to describe the orbit of the satellite in question. The model used in calculation of the TLE is the SGP4 model, and this the one used during this thesis, for simulation.

An example of a TLE can be seen below. The ALPHASAT TLE is used as an example.

ALPHASAT

```
1 40949U 98067HA 16131.17243197 .00049328 00000-0 32059-3 0 9990
2 40949 51.6335 230.6137 0003739 51.3487 308.7846 15.75443623 34062
```

The first line is the name of the satellite. The two other lines is described in table 4.1 and 4.2.

Field	Columns	Content	ALPHASAT
1	01-01	Line Number	1
2	03-07	Satellite Number	40949
3	08-08	Classification (U = Unclassified)	U
4	10-11	International Designator (Last two digits of launch year)	98
5	12-14	International Designator (Launch number of the year)	067
6	15-17	International Designator (Piece of the launch)	HA
7	19-20	Epoch Year (Last two digits of year)	16
8	21-32	Epoch (Day of the year and fractional portion of the day)	131.1724319
9	34-43	First Time Derivative of the Mean Motion divided by two	.00049328
10	45-52	Second Time Derivative of Mean Motion divided by six	00000-0
11	54-61	Drag term (decimal point assumed)	32059-3
12	63-63	The number 0 (Originally Ephemeris type)	0
13	65-68	Element number	999
14	69-69	Checksum (Modulo 10)	0

**Table 4.1:** The first line of the ALPHASAT TLE [Markley 14].

Field	Columns	Content	ALPHASAT
1	01-01	Line number	2
2	03-07	Satellite number	40949
3	09-16	Inclination (Deg)	51.6335
4	18-25	Right Ascension (Deg)	230.6137
5	27-33	Eccentricity (decimal point assumed)	0003739
6	35-42	Argument of Perigee (Deg)	51.3487
7	44-51	Mean Anomaly (Deg)	308.7846
8	53-63	Mean Motion (Revs per Day)	15.75443623
9	64-68	Revolution number at epoch (Revs)	3406
10	69-69	Checksum (Modulo 10)	2

**Table 4.2:** The second line of the ALPHASAT TLE [Markley 14].

The SGP4 model, just described, is used in the simulations of ALPHASAT. This is for generating a positional and velocity which can be used together with the disturbance models.

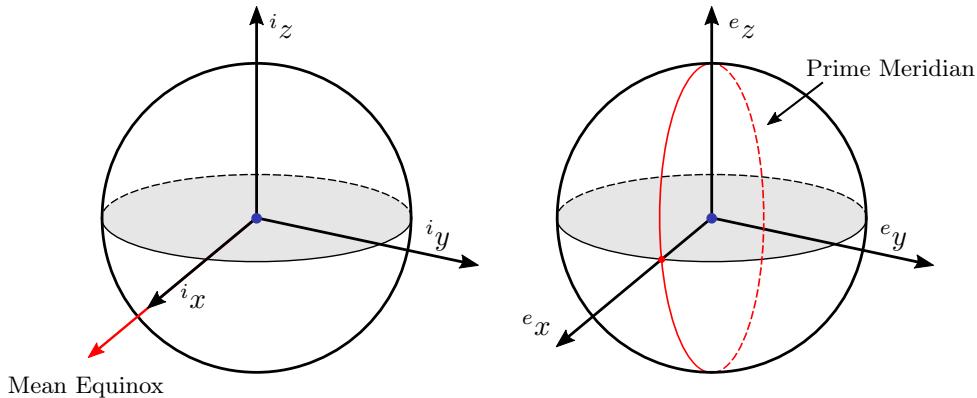
## 4.3 Reference Frames

In the following sections various of notations and definitions of coordinate changes will be introduced in order to simplify the kinematic and dynamic equations of the satellite. The section includes the basic descriptions of the necessary reference frames.

The choice of reference frames used to describe the attitude of the given satellite depends on the mission during its lifetime. For an orbit around the Earth, an inertial coordinate frame is fixed in the centre of the Earth. It is possible for the spacecraft to switch between inertial coordinate frames e.g. voyaging towards another planet in the solar system. In this project the mission of the spacecraft is defined in an Earth orbit.

### 4.3.1 Earth Centred Inertial (ECI) and Earth Centred Earth Fixed (ECEF)

The ECI reference frame is defined to have its origin fixed in the Earth's CoM. The orientation of the ECI is defined according to the J2000 ECI reference frame. In this frame the  $i_z$ -axis is pointing in the rotational axis of the Earth, the  $i_x$ -axis points in the direction of the mean equinox and the  $i_y$ -axis completes the right handed Cartesian coordinate system. The  $(i_x, i_y)$  plane spans the equatorial plane of the Earth, as shown to the left in Figure 4.3.



**Figure 4.3:** Shows the ECI reference frame to the left with the  $i_x$  pointing towards the mean Equinox. To the right the ECEF reference frame is where the  $e_x$ -axis points to where the fixed point on the Prime Meridian.

The ECEF reference frame is used as a fixed coordinate system which follows the rotation of the Earth's. As in the ECI the  $e_z$ -axis points in the direction of the Earth rotation axis. The  $e_x$ -axis points in the direction of the prime meridian perpendicular to the  $e_z$ -axis. The  $e_y$ -axis completes the right handed Cartesian coordinate system and the  $(e_x, e_y)$ -plane forms the equatorial plane, as shown to the right in Figure 4.3.

### Rotation Between ECI and ECEF

The rotation from ECI to ECEF is given by the rotation of the Earth. The  ${}^e z$ -axis is rotating around the  ${}^i z$  with a period time of a sidereal day denoted by  $t_d$ . This rotation is described by a direction cosine matrix with a rotation around the  $z$ -axis.

$${}^i \underline{\mathbf{R}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

where  ${}^i \underline{\mathbf{R}}$  is the rotation matrix from ECI to ECEF and  $\theta(t)$  is the angle between the ECI and ECEF and is given as:

$$\theta(t) = \frac{2\pi \cdot t}{t_d} \quad (4.3)$$

Where  $t$  is the time since last alignment with the ECI frame. The direction cosine can be represented in form of a quaternion as following:

$$\begin{aligned} {}^i q_4 &= \pm \frac{1}{2} \sqrt{1 + R_{11} + R_{22} + R_{33}} = \sqrt{\frac{1 + \cos(\theta)}{2}} \\ {}^i q_1 &= \frac{R_{23} - R_{32}}{4q_4} = 0 \\ {}^i q_2 &= \frac{R_{31} - R_{13}}{4q_4} = 0 \\ {}^i q_3 &= \frac{R_{12} - R_{21}}{4q_4} = \frac{-2 \sin(\theta)}{4q_4} \end{aligned} \quad (4.4)$$

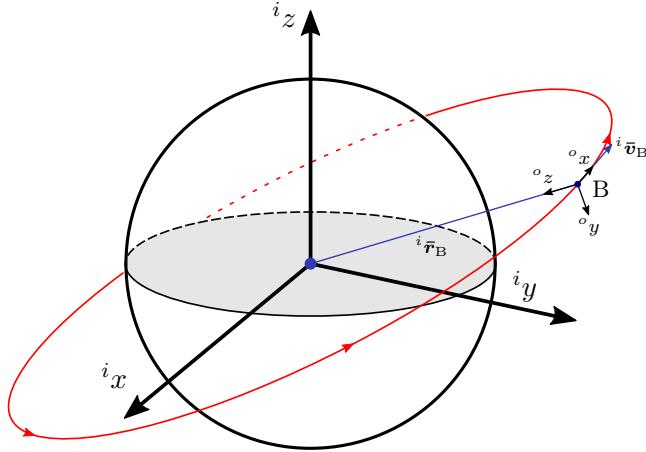
it should be noted that when  ${}^i q_4$  is small calculations becomes inaccurate as described in Appendix A.1.1.

### 4.3.2 Orbit Reference Frame (ORF)

The ORF is used when the satellite is said to be pointing Nadir where the origin is located in the satellite CoM. The  ${}^o z$ -axis of the ORF is always pointing in the direction of the Earth's CoM (Nadir pointing). The  ${}^o x$ -axis points in the direction of the velocity vector of the spacecraft. The  ${}^o y$ -axis of the ORF is then normal to the  $({}^o x, {}^o z)$ -plane completing the right handed Cartesian coordinate system.

### Rotation Between ECI and ORF

In order to find the rotation from ECI to ORF is derived from the position vector and velocity vector of the spacecraft given in the ECI frame as shown in Figure 4.4.



**Figure 4.4:** Shows the orientation of the ORF where the  ${}^o\bar{z}$  is always pointing Nadir.

The  ${}^o\bar{z}$  of the ORF frame is defined to point in the direction of Nadir

$${}^o\bar{z} = -\frac{{}^i\bar{r}_B}{\|{}^i\bar{r}_B\|} \quad (4.5)$$

$${}^o\bar{x} = \frac{{}^i\bar{v}_B}{\|{}^i\bar{v}_B\|} \quad (4.6)$$

Where  ${}^i\bar{v}_B$  is the velocity vector of the spacecraft given in ECI. Then  ${}^o\bar{y}$  is given as the crossproduct of  ${}^o\bar{z}$  and  ${}^o\bar{x}$ .

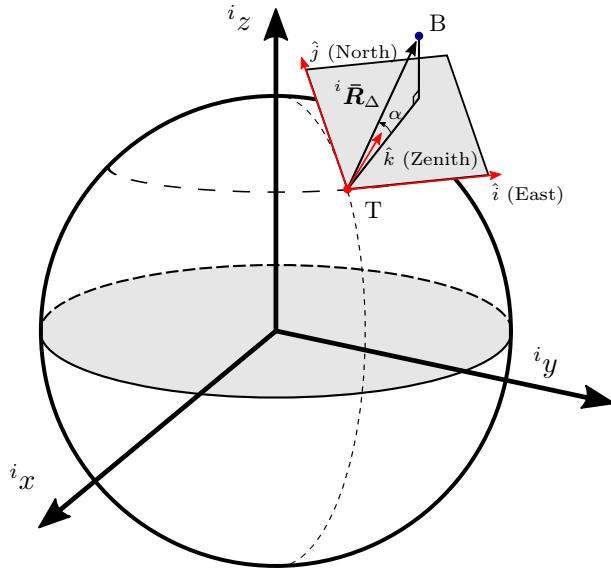
$${}^o\bar{y} = {}^o\bar{z} \times {}^o\bar{x} \quad (4.7)$$

The rotation matrix from ECI to ORF can then be written as:

$${}^o\underline{\mathbf{R}} = [{}^o\bar{x} \quad {}^o\bar{y} \quad {}^o\bar{z}] \quad (4.8)$$

### 4.3.3 Target Reference Frame (TRF)

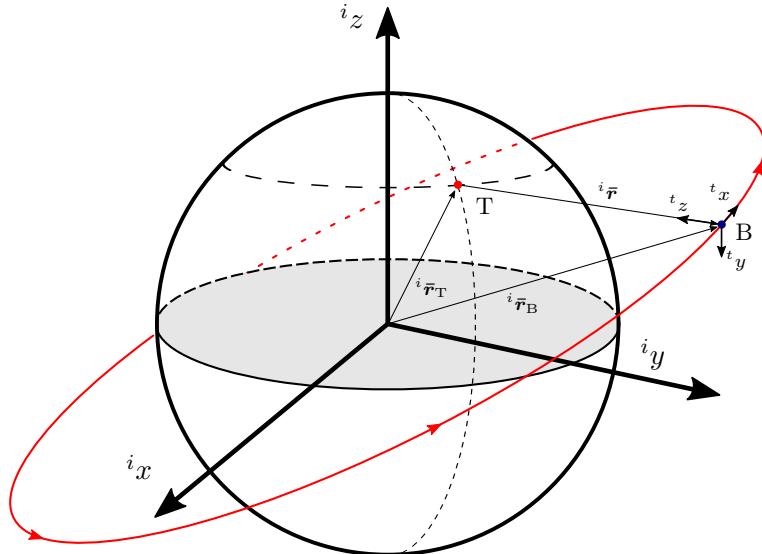
The purpose of the TRF is to track an interest point on the surface of the Earth. The point is only visible from the spacecraft when it is over the local horizon plane which is given as the tangent plane to the specific point. This plane is also called the Topocentric Horizon Coordinate System (THCS) and its coordinates are given as the East-North-Zenith (ENZ) with the  $x$ -axis pointing East,  $y$ -axis pointing North and with the  $z$ -axis normal to the plane (Zenith) as shown in Figure 4.5.



**Figure 4.5:** Shows the THCS given by the span of  $\{\hat{i}, \hat{j}\}$  with the vector  ${}^i\bar{R}_\Delta$  pointing from the target (T) in the direction of the spacecraft (B).

The origin of the TRF is defined where the position vector  ${}^i\bar{R}_\Delta$  intersects with the spacecraft's CoM. The  ${}^t z$ -axis of the TRF is the negative unit vector of the  ${}^i\bar{R}_\Delta$  position vector denoted by  $-{}^i\bar{r}_\Delta$ . The calculation of  ${}^i\bar{r}_\Delta$  is described in Section 6.6.

The  ${}^t y$ -axis is given as a unit vector perpendicular to the plane spanned by the velocity vector  ${}^o\bar{x}$  and the  ${}^t\bar{z}$ . The  ${}^t x$ -axis is then given as the cross product between  ${}^t z$  and  ${}^t y$  to complete the right handed Cartesian coordinate system as shown in Figure 4.6.



**Figure 4.6:** Shows the TRF located in the CoM of the satellite body B, where the  ${}^t z$ -axis points to the target T and  ${}^t x$ -axis points in the direction of flight.

#### 4.3.4 Rotation Between ECI and TRF

The rotation between ECI and TRF is derived from the position vector of the satellite and the target given in the ECI frame.

$${}^t\bar{z} = -\frac{{}^i\bar{r}_B - {}^i\bar{r}_T}{\|{}^i\bar{r}_B - {}^i\bar{r}_T\|} \quad (4.9)$$

where:

${}^i\bar{r}_B$	is the satellite position vector.	[m]
${}^i\bar{r}_T$	is the target position vector.	[m]

(4.10)

The  ${}^t\bar{y}$ -axis is then derived from the velocity vector of the satellite given in the ORF. The  ${}^t\bar{y}$ -axis of the TRF is defined as being perpendicular to the plane spanned by the velocity vector  ${}^i\bar{v}_B$  and the  ${}^t\bar{z}$ .

$${}^t\bar{y} = {}^t\bar{z} \times \frac{{}^i\bar{v}_B}{\|{}^i\bar{v}_B\|} \quad (4.11)$$

where

${}^i\bar{v}_B$	is the satellite velocity vector given in the ECI frame.	[m/s]
-----------------	--	-------

(4.12)

The  ${}^t\bar{x}$  of the TRF is then the normal to the  $({}^t\bar{y}, {}^t\bar{z})$ -plane forming a right handed Cartesian coordinate system.

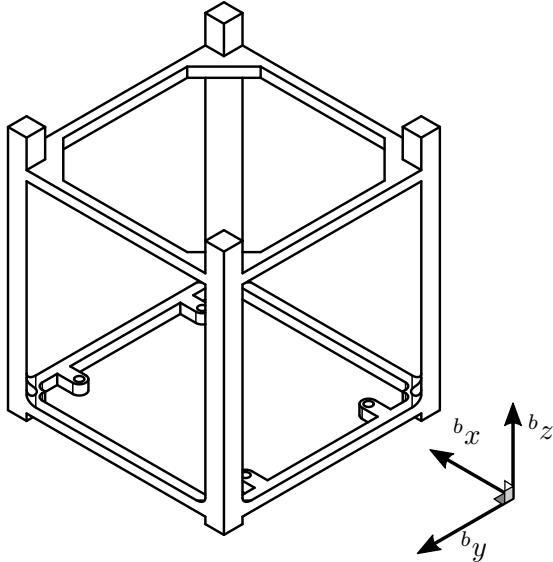
$${}^t\bar{x} = {}^t\bar{y} \times {}^t\bar{z} \quad (4.13)$$

#### 4.3.5 Body Reference Frame (BRF)

The satellite BRF is defined to have its origin in CoM of the satellite. The orientation of the coordinate system is defined in relation to the mission of the satellite e.g. a mission payload containing a camera would have one of the axis pointing in the direction of the camera. In general the BRF is defined such that the coordinate system is easily recognised by looking at the spacecraft.

The attitude error of a spacecraft in orbit could for example be the angular difference between the TRF and the BRF which is the rotation needed for pointing e.g. a camera at a specific point on the Earth's surface.

In this project the BRF is defined as the coordinate system shown in Figure 4.7.



**Figure 4.7:** Shows the CubeSat BRF which is centred in the CoM.

### Rotation Between ECI and BRF

The attitude of the satellite is given by the rotation relative to the ECI. The orientation of the spacecraft is found through an ADS.

#### 4.3.6 Control Reference Frame (CRF)

The spacecraft dynamics originates from the Control Reference Frame (CRF). The CRF is used to describe the satellites mass distribution and has its origin in the CoM of the spacecraft. All the external torques from the actuators and disturbances is described in this reference frame. The frame originates from the satellite's inertia matrix where the axes are defined by the principal axes of the inertia matrix.

The  $^c z$ -axis of the CRF points in the direction of the major-axis where the Moment of Inertia (MoI) is largest. The  $^c x$ -axis is then defined as pointing in the direction of the minor-axis where the MoI is smallest. The  $^c y$ -axis is the perpendicular to the  $(^c x, ^c z)$ -plane so it completes the right handed Cartesian coordinate system.

### Rotation Between BRF and CRF

It is possible that the CRF and the BRF have zero rotation between each other. This would mean that the satellite's mass was distributed symmetrically over the body of the satellite, but this is not always the case. Often the CoM of the satellite is moved away from the geometric centre and the principal axes of the inertia matrix is rotated because of uneven mass distribution.

This rotation caused by the uneven distributed mass is found by taking the Eigenvalue Decomposition (EVD) of the inertia matrix and thereby finding its corresponding eigenvectors and eigenvalues as shown in equation 4.14

$$\underline{\mathbf{J}} = \underline{\mathbf{R}} \underline{\mathbf{J}}' \underline{\mathbf{R}}^{-1} \quad (4.14)$$

Where

<u>J</u>	is the satellite body inertia matrix.	[m]
<u>J'</u>	is the satellite principal moments of inertia.	[m]
<u>R</u>	is the transformation matrix, with respective eigen vectors.	[m]

(4.15)

#### 4.3.7 Global Reference Frame (GRF)

The Global Reference Frame (GRF) is a reference frame used by the attitude control testbeds attitude determination system. The purpose of the GRF is to ease the mathematical derivations of the attitude estimator. The GRF is defined as an East North Up (ENU) reference frame where the  $x$ -axis,  ${}^g\bar{x}$ , is pointing East, the  $y$ -axis,  ${}^g\bar{y}$ , is pointing North and the  $z$ -axis,  ${}^g\bar{z}$  is pointing up; all defined in the ECEF.

#### 4.3.8 Summary

Through this chapter a keplerian orbit has been described together with the orbital perturbation model, SGP4. After this a number of reference frames where defined. These reference frames will be used, through the rest of the report, in order to simplify the mathematics. In the following chapter the models used in the design of the attitude control algorithms, and in the simulations, are described.

---

# Spacecraft and Disturbance Modelling

In this chapter the models for the satellite are identified. These models will be used to analyse the motion of the satellite, this includes both the kinematic and the dynamic equations. The disturbances are identified and modelled in order to take these into account during the design of the control strategies. In order to be able to exert the exact necessary torque on the satellite body, the chosen actuators, as described in Section 1.2, are modelled.

## 5.1 Satellite Kinematic Equation

Sources: [Sidi 97] [Wertz 95] [Markley 14]

The kinematics of a satellite are defined as the motion relative to time. Quaternions are used as attitude representation and this means that the kinematics are the rate of change in attitude - the derivative of the attitude quaternion. The quaternion algebra is described in Appendix A.

The attitude quaternion  $\mathbf{q}(t + \Delta t)$  at the time  $t + \Delta t$  can be split up into:

$$\mathbf{q}(t + \Delta t) = \mathbf{q}(t) \otimes \mathbf{q}(\Delta t) \quad (5.1)$$

Dividing the quaternion into a real and a pure quaternion part

$$\mathbf{q}(\Delta t) = \cos\left(\frac{\Delta\Phi}{2}\right) + \bar{\mathbf{u}} \sin\left(\frac{\Delta\Phi}{2}\right) \quad (5.2)$$

Where the sine and cosine parts can be replace by the small angle approximation:

$$\begin{aligned} \cos\left(\frac{\Delta\Phi}{2}\right) &\approx 1 \\ \sin\left(\frac{\Delta\Phi}{2}\right) &\approx \frac{1}{2}\omega\Delta t \end{aligned} \quad (5.3)$$

Equation (5.1) and equation (5.3) can then be combined into:

$$\mathbf{q}(t + \Delta t) \approx \left( \mathbf{I} + \frac{1}{2}\omega\Delta t \begin{bmatrix} 0 & -u_3 & u_2 & u_1 \\ u_3 & 0 & -u_1 & u_2 \\ -u_2 & u_1 & 0 & u_3 \\ -u_1 & -u_2 & -u_3 & 0 \end{bmatrix} \right) \mathbf{q}(t) \quad (5.4)$$

This can be simplified to:

$$\mathbf{q}(t + \Delta t) \approx \frac{1}{2} \Delta t \cdot \underline{\Omega}(\bar{\omega}) \mathbf{q}(t) + \mathbf{q}(t) \quad (5.5)$$

Where  $\underline{\Omega}(\bar{\omega})$  is the four-dimensional skewsymmetric crossproduct operator

$$\underline{\Omega}(\bar{\omega}) = \begin{bmatrix} 0 & -\omega_3(t) & \omega_2(t) & \omega_1(t) \\ \omega_3(t) & 0 & -\omega_1(t) & \omega_2(t) \\ -\omega_2(t) & \omega_1(t) & 0 & \omega_3(t) \\ -\omega_1(t) & -\omega_2(t) & -\omega_3(t) & 0 \end{bmatrix} \quad (5.6)$$

Equation (5.5) is rearranged to

$$\frac{\mathbf{q}(t + \Delta t) - \mathbf{q}(t)}{\Delta t} = \frac{1}{2} \underline{\Omega}(\bar{\omega}) \mathbf{q}(t) \quad (5.7)$$

If a limit is imposed the kinematic equation appears

$$\dot{\mathbf{q}}(t) = \lim_{\Delta t \rightarrow 0} \frac{\mathbf{q}(t + \Delta t) - \mathbf{q}(t)}{\Delta t} = \frac{1}{2} \underline{\Omega}(\bar{\omega}) \mathbf{q}(t) \quad (5.8)$$

The kinematic equation of the satellite is now derived. This kinematic equation is used to describe of the satellites motion relative to time. In the following section, the satellites change in angular momtm relative to the sum of all external torques exerted onto the satellite body is derived.

## 5.2 Satellite Dynamic Equation

Sources: [Sidi 97] [Wertz 95] [Markley 14]

The dynamics of a satellite describes the rotational motion of the satellite relative to the torques exerted, whether the torques stems from actuator torques or disturbance torques. The rotational dynamics are derived using Euler's rigid body dynamics.

The angular momentum can be written as:

$${}^i \bar{\mathbf{L}}(t) = \underline{\mathbf{J}}_s {}^i \bar{\omega}(t) \quad (5.9)$$

Where  $\bar{\omega}(t) = [\omega_1 \ \omega_2 \ \omega_3]^T$  and  $\underline{\mathbf{J}}_s$  is the inertia matrix of the satellite.

According to Eulers second law the applied torque can be derived as the time derivative of the angular momentum [Serway 14]:

$$\begin{aligned} {}^i \bar{\tau}_{\text{ext}}(t) &= \frac{d}{dt} {}^i \bar{\mathbf{L}}(t) \\ &= {}^i \dot{\bar{\mathbf{L}}}(t) \end{aligned} \quad (5.10)$$

Where the torque described in the control reference frame can be described as:

$$\begin{aligned} {}^c \bar{\tau}_{\text{ext}}(t) &= \frac{d}{dt} \left( {}^c \underline{\mathbf{R}} {}^i \bar{\mathbf{L}}(t) \right) \\ &= {}^c \dot{\underline{\mathbf{R}}} {}^i \bar{\mathbf{L}}(t) + {}^c \underline{\mathbf{R}} {}^i \dot{\bar{\mathbf{L}}}(t) \end{aligned} \quad (5.11)$$

Where the derivative of a rotation matrix can be written as [Markley 14]:

$${}^c \dot{\underline{\mathbf{R}}} = -{}^c \bar{\omega}(t) \times {}^c \underline{\mathbf{R}} \quad (5.12)$$

Inserted in equation (5.11) this gives:

$$\begin{aligned} {}^c\bar{\tau}_{\text{ext}}(t) &= -{}^c\bar{\omega}(t) \times {}^c_i\underline{\mathbf{R}}^i \bar{\mathbf{L}}(t) + {}^c_i\underline{\mathbf{R}}^i \dot{\bar{\mathbf{L}}}(t) \\ &= -{}^c\bar{\omega}(t) \times {}^c\bar{\mathbf{L}}(t) + {}^c\dot{\bar{\mathbf{L}}}(t) \end{aligned} \quad (5.13)$$

Using equation (5.9) and isolating the angular acceleration, results in the satellite dynamic equation.

$$\begin{aligned} {}^c\bar{\tau}_{\text{ext}}(t) &= -{}^c\bar{\omega}(t) \times \underline{\mathbf{J}}_s {}^c\bar{\omega}(t) + \underline{\mathbf{J}}_s {}^c\dot{\bar{\omega}}(t) \\ \underline{\mathbf{J}}_s {}^c\dot{\bar{\omega}}(t) &= -{}^c\bar{\omega}(t) \times \underline{\mathbf{J}}_s {}^c\bar{\omega}(t) + {}^c\bar{\tau}_{\text{ext}}(t) \\ {}^c\dot{\bar{\omega}}(t) &= \underline{\mathbf{J}}_s^{-1} [-{}^c\bar{\omega}(t) \times \underline{\mathbf{J}}_s {}^c\bar{\omega}(t) + {}^c\bar{\tau}_{\text{ext}}(t)] \end{aligned} \quad (5.14)$$

Equation (5.14) can be simplified to:

$${}^c\dot{\bar{\omega}}(t) = \underline{\mathbf{J}}_s^{-1} [-\underline{\mathbf{S}}({}^c\bar{\omega}) \underline{\mathbf{J}}_s {}^c\bar{\omega}(t) + {}^c\bar{\tau}_{\text{ext}}(t)] \quad (5.15)$$

where  ${}^c\bar{\tau}_{\text{ext}}$  is the sum of all external torques; both the actuator torques and the disturbances.  $\underline{\mathbf{S}}({}^c\bar{\omega})$  is the three-dimensional skew-symmetric crossproduct operator, defined as:

$$\underline{\mathbf{S}}(\bar{\omega}) \triangleq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (5.16)$$

In the previous sections both the kinematics and dynamics of the satellite body, has been derived. These can be used in a deterministic approach. The next step is to identify the forces and torques the satellite will be subject to in a space environment. These are described in the following sections.

## 5.3 Disturbances

Space is a harsh environment featuring multiple disturbances. The major disturbances taken into account in a LEO are the residual magnetic moment, aerodynamic drag and the radiation pressure [Wertz 95]. In the following sections these three disturbances will be described.

### 5.3.1 Residual Magnetic Torque

Sources: [Wertz 95] [NASA 69b]

A satellite in orbit, travels in the geomagnetic field of the Earth. This leads to three main magnetic disturbance torques. These are the effects of the interaction between the satellite and geomagnetic field.

- Eddy currents
- Hysteresis
- Spacecraft magnetic moments

*Eddy Current* induces a small magnetic field in the metal. When a ferromagnetic material is exposed to magnetism then an electric field is induced in the material. This electric field leads to small currents running in the ferromagnetic material these currents are called eddy currents.

*Hysteresis* is the phenomenon of ferromagnetic materials on a molecular level gets magnetised by another magnet. When exposed to a magnetic field, the molecules in the material will align with the field and thereby rendering its own magnetic field.

*Spacecraft Magnetic Moments*; of the three disturbance mentioned here, this is usually the largest one. Spacecraft generated magnetic moment are caused by for example the wiring inside the satellite, the Printed Circuit Board (PCB) layout and everywhere else, where a current loop is present.

Eddy Currents and Hysteresis are negligible, compared to the Spacecraft Magnetic Moment and in the following only the latter will be considered.

The spacecraft magnetic moment can be calculated as:

$$\bar{\tau}_{\text{mag}} = \bar{m} \times \bar{B} \quad (5.17)$$

Where:

$\bar{m}$	is the residual dipole moment of the satellite	$[\text{Am}^2]$
$\bar{B}$	is the Earth geocentric magnetic flux density	$\left[ \frac{\text{Wb}}{\text{m}^2} \right]$

The magnetic residual torque  $\bar{\tau}_{\text{mag}}$  is inherently difficult to calculate because of the uncertainty in residual dipoles direction and size which depends on the specific satellite. In [Wertz 95], the Firesat example is suggested to have a magnetic moment  $\bar{m}$  of  $1 \text{ Am}^2$  for a 110W satellite. A single unit CubeSat is expected to have an average power of 1W which is approximately 100 times less than the Firesat. This would give a qualified guesstimate of  $0.01 \text{ Am}^2$  for the satellites residual dipole moment  $m$ . The worst case magnetic residual torque  $\tau_m$  can then be approximated by

$$\tau_m = m \cdot B \quad (5.18)$$

where  $B$  can be approximated for a polar orbit by

$$B = \frac{2 \cdot M}{R^3} \quad (5.19)$$

where  $M$  is the magnetic moment of the Earth given as  $7.96 \cdot 10^{15} \text{ tesla} \cdot \text{m}^3$  and  $R$  is the radius from the centre of the Earth to the satellite where  $R = (6378.1 + 300) \cdot 10^3$  [Wertz 95]. The worst case magnetic residual torque is then found by

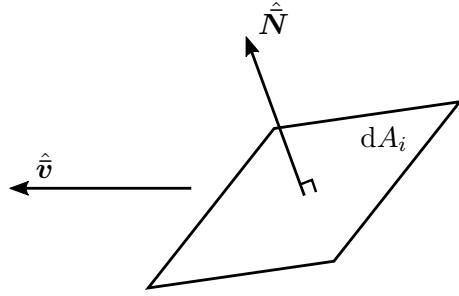
$$\tau_m = 0.01 \cdot \frac{2 \cdot 7.96 \cdot 10^{15}}{(6978100)^3} = 4.6852 \cdot 10^{-7} \quad [\text{Nm}] \quad (5.20)$$

Which is later found to be the largest disturbance of those selected.

### 5.3.2 Aerodynamic Drag

Sources: [Wertz 95] [NASA 71]

In the upper atmosphere at orbital heights for a LEO, the gas particles that hit the satellite are reflected. This creates a force on the spacecraft opposite to the direction of travel.



**Figure 5.1:** The vectors involved in calculating the aerodynamic drag.

The force created are dependent on the size of the incident area, the atmospheric density, the travel velocity and the drag coefficient of the incidence surface.

The  $i$ 'eth aerodynamic force  $d\bar{\mathbf{F}}_{D,i}$  on the  $i$ 'eth infinitesimal small surface element  $dA_i$ , can be calculated as:

$$d\bar{\mathbf{F}}_{D,i} = -\frac{1}{2}C_D\rho V^2 \left[ \hat{\mathbf{N}}^\top \hat{\mathbf{v}} \right] \hat{\mathbf{v}} dA_i \quad (5.21)$$

where:

$C_D$	is the drag coefficient	[.]
$\rho$	is the atmospheric density	$\left[ \frac{\text{kg}}{\text{m}^3} \right]$
$V$	is the orbital velocity	$\left[ \frac{\text{m}}{\text{s}} \right]$
$A$	is the incidence area	$[\text{m}^2]$
$\hat{\mathbf{N}}$	is unit vector normal to the incident area	[.]
$\hat{\mathbf{v}}$	is a unit vector in the velocity direction	[.]

The force  $d\bar{\mathbf{F}}_D$  leads to a torque about the CoM, which can be calculated as:

$$\bar{\tau}_D = \int \bar{\mathbf{r}}_{D,i} \times d\bar{\mathbf{F}}_{D,i} \quad (5.22)$$

Where  $\bar{\mathbf{r}}_D$  is a vector from the satellite CoM to the  $i$ 'eth surfaces element  $dA_i$ .

From equation (5.21) and equation (5.22) it can be seen that the disturbance torque caused by the aerodynamic drag is dependent on the surface area in the direction of travel. For ALPHASAT this entails that the disturbance caused by drag will be attitude dependent, because of the spacecraft's uneven shape. Furthermore the aerodynamic drag is dependent on the orbital velocity  $V$  squared. Both the orbital velocity and the atmospheric density  $\rho$  depends on the orbital altitude. This means that the altitude and the attitude is the two most important parameters for determine the aerodynamic drag for ALPHASAT.

### 5.3.3 Gravity-Gradient Torque

Sources: [NASA 69a] [Wertz 95]

In a non-symmetric orbit, the satellite will experience gravitational torques due to the Earth's gravitational field. The gravity-gradient torque is caused by the varying of the Earth's gravitational field in a non-symmetric orbit, resulting in variations in the gravitational force exerted on the spacecraft. In a symmetric orbit there would be no

variations in the gravitational field, that is, if the Earth were to be modelled with an evenly distributed spherical mass.

The Earth and the satellite are modelled as to point masses affecting each other. When dividing the satellite up into infinitesimal mass elements the force acting on these can be written as:

$$d\bar{\mathbf{F}}_i = \frac{-\mu \bar{\mathbf{R}}_{E,i} dm_i}{\|\bar{\mathbf{R}}_{E,i}\|_2^2} \quad (5.23)$$

Where:

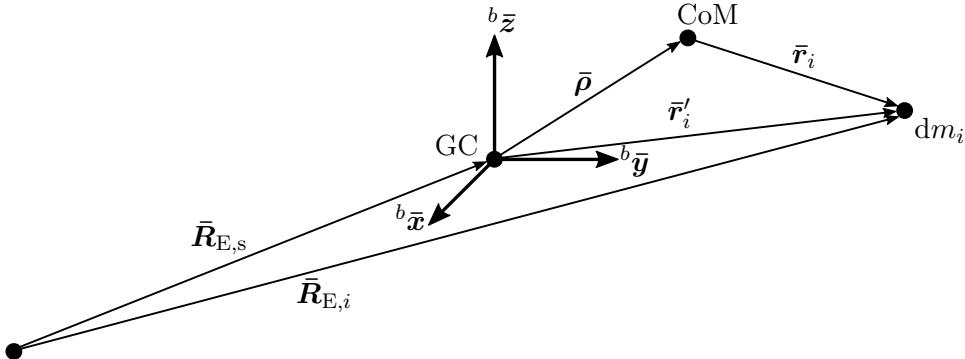
$d\bar{\mathbf{F}}_i$	is the force on the $i$ 'eth mass element	[N]
$\mu$	is the Earth's gravitational constant	[·]
$\bar{\mathbf{R}}_{E,i}$	is the vector from the Earth's CoM to the $i$ 'eth mass elements CoM	[·]
$dm_i$	is the $i$ 'eth mass element	[·]

The torque contribution from the infinitesimal mass element about the satellites CoM, is then found as:

$$d\bar{\tau}_i = \bar{\mathbf{r}}_i \times d\bar{\mathbf{F}}_i = (\bar{\rho} + \bar{\mathbf{r}}'_i) \times d\bar{\mathbf{F}}_i \quad (5.24)$$

Where:

$d\bar{\tau}_i$	is the torque from the $i$ 'eth mass element	[Nm]
$\bar{\mathbf{r}}_i$	is the vector from CoM to the mass element	[·]
$\bar{\rho}$	is a vector from the geometric centre to CoM the satellite	[·]
$\bar{\mathbf{r}}'_i$	is a vector from the geometric centre to $dm_i$	[·]



**Figure 5.2:** The vectors involved in calculating the gravity gradient torque, where GC is the geometric centre of the satellite.

The resulting torque affecting the entire satellite can then be written as:

$$\bar{\tau}_{GG} = \int \bar{\mathbf{r}}_i \times d\bar{\mathbf{F}}_i = \int (\bar{\rho} + \bar{\mathbf{r}}'_i) \times \frac{-\mu \bar{\mathbf{R}}_{E,i}}{\|\bar{\mathbf{R}}_{E,i}\|_2^3} dm_i \quad (5.25)$$

If the satellites CoM and geometric centre is assumed identical equation (5.25) can be simplified to:

$$\bar{\tau}_{GG} = \frac{3\mu}{\|\bar{\mathbf{R}}_{E,s}\|_2^3} \left[ \hat{\bar{\mathbf{R}}}_{E,s} \times (\underline{\mathbf{J}}_s \cdot \hat{\bar{\mathbf{R}}}_{E,s}) \right] \quad (5.26)$$

Where:

- |                            |   |     |
|----------------------------|---|-----|
| $\underline{\mathbf{J}}_s$ | is the moment of inertial tensor of the satellite       | [·] |
| $\hat{\mathbf{R}}_{E,s}$   | is the vector from the Earth's CoM to the satellite CoM | [·] |

From equation (5.26) it can be seen that it produces a vector which the gravity gradient torque is about, and a scaling of that vector which depends on the distance to the Earth's centre.

If a spacecraft is in an elliptical orbit then the gravity gradient would vary over the coarse of the orbit; caused by the varying of the distance to the centre of the Earth. The gravity gradient torque also depends on the mass distribution of the spacecraft as the cross product is between  $\hat{\mathbf{R}}_{E,s}$  and  $\underline{\mathbf{J}}_s \hat{\mathbf{R}}_{E,s}$  where the latter is a perturbation caused by the spacecraft's moment of inertia tensor. If the spacecraft had a perfect evenly distributed moment of inertia then there would be no torque disturbance from the gravity gradient.

### 5.3.4 Radiation Pressure

Sources: [NASA 69c] [Wertz 95] [Cappellari 76]

A satellite in a LEO, is exposed to electromagnetic radiation pressure. This sources of this electromagnetic radiation pressure are: solar illumination, Earth and Moon Albedo and radiation directly from the Earth and its atmosphere. The radiation pressure from the Albedo and directly from the Earth are however negligible compared to the radiation pressure radiated from the Sun. The radiation pressure hits the satellite as a force and causes a torque about the spacecraft's CoM.

The radiation influx on a spacecraft amounts to three different phenomena, namely difusely reflection, specularly reflection and absorption.

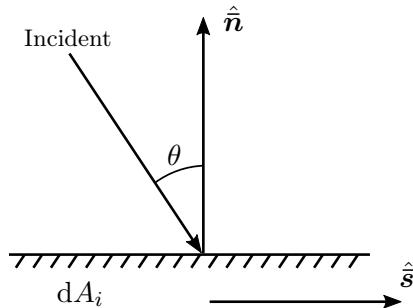


Figure 5.3: Absorption

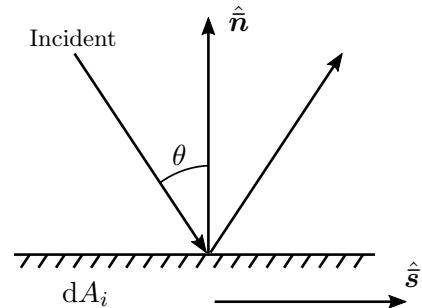


Figure 5.4: Reflection

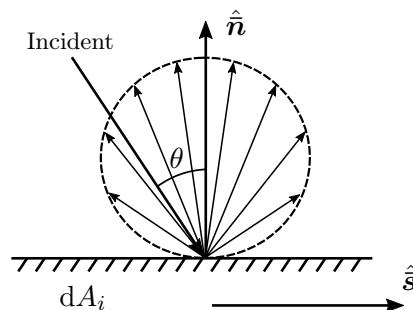


Figure 5.5: Diffusion

An illustration of the three phenomena are shown in Figures 5.3, 5.4 and 5.5

When the spacecraft body is hit by the radiation influx, the force generated can be calculated by dividing the spacecraft's incident area up into infinitesimal small surface areas for each type of radiation reaction, and then by integrating over the incident surface area the resulting force can be found.

If the incident surface areas emissivity constant and the reflection distribution is known then by knowing the amount of radiation influx it is possible to calculate the disturbance force analytically. However the emissivity constant and reflection constant are not easily known and therefore the radiation force are often calculated as either reflected, absorbed completely, or a combination of those. The reflection can be both difused or specularly reflection.

The  $i$ 'eth force  $d\bar{\mathbf{F}}_{a,i}$  caused by absorbed radiation pressure can be calculated for the  $i$ 'eth infinitesimal surface element  $dA_i$  can be calculated as:

$$d\bar{\mathbf{F}}_{a,i} = \frac{I}{c} [c_a (-\cos(\theta)\hat{\mathbf{n}} + \sin(\theta)\hat{\mathbf{s}})] \cos(\theta) dA_i \quad (5.27)$$

Where:

$I$	is the energy per second through a surface element	[N/m <sup>2</sup> ]
$c$	is the speed of light	[m/s]
$c_a$	is the absorption coefficient of the incident surface	[.]
$\theta$	is the incidence angle relative to the surface normal	[rad]
$\hat{\mathbf{n}}$	is a unit normal to the incidence area	[.]
$\hat{\mathbf{s}}$	is a unit vector parallel to the incidence area	[.]

In equation (5.27) it can be seen that the force generated from the absorbed radiation pressure is proportional to the energy per second through a surface element from the radiation and inverse proportional to the speed of light. The disturbance force is furthermore proportional to the incidence surface area and thereby the disturbance results in a mass flow rate. The absorption coefficient  $c_a$  is a constant pertaining to the surface parameters of the spacecraft, and  $c_a \in [0; 1]$ , for completely absorbed radiation  $c_a = 1$  and for completely reflected radiation  $c_a = 0$ .

The  $i$ 'eth force  $d\bar{\mathbf{F}}_{rs,i}$  caused by specularly reflected radiation pressure can be calculated for the  $i$ 'eth infinitesimal surface element  $dA_i$  can be calculated as:

$$d\bar{\mathbf{F}}_{rs,i} = \frac{I}{c} [-(1 + c_{rs}) \cos(\theta)\hat{\mathbf{n}} + (1 - c_{rs}) \sin(\theta)\hat{\mathbf{s}}] \cos(\theta) dA_i \quad (5.28)$$

Where:

$c_{rs}$	is the specularly reflection coefficient of the incident surface	[.]
----------	--	-----

As for absorbed radiation pressure, the specularly reflected radiation pressure results in a mass flow rate. The specularly reflection coefficient  $c_{rs}$  is a constant pertaining to the surface parameters of the spacecraft, and  $c_{rs} \in [0; 1]$ , for completely specularly reflected radiation  $c_{rs} = 1$  and for completely absorbed radiation  $c_{rs} = 0$ .

The  $i$ 'eth force  $d\bar{\mathbf{F}}_{rd,i}$  caused by diffusely reflected radiation pressure can be calculated for the  $i$ 'eth infinitessimal surface element  $dA_i$  can be calculated as:

$$d\bar{\mathbf{F}}_{rd,i} = \frac{I}{c} \left[ -\left( \cos(\theta) + \frac{2}{3}c_{rd} \right) \hat{\mathbf{n}} + \sin(\theta)\hat{\mathbf{s}} \right] \cos(\theta) dA_i \quad (5.29)$$

Where:

$c_{rd}$  is the diffusely reflected coefficient of the incident surface [.]

As for both the absorbed radiation pressure and the specularly reflected radiation pressure, the difusely reflected radiation pressure results in a mass flow rate. The diffusely reflection coefficient  $c_{rd}$  is a constant pertaining to the surface parameters of the spacecraft, and  $c_{rd} \in [0; 1]$ , for completely diffusely reflected radiation  $c_{rd} = 1$  and for no diffusely reflected radiation pressure  $c_{rd} = 0$ .

The absorption coefficient, specularly reflection coefficient and the diffusely reflection coefficient must comply with:

$$c_a + c_{rs} + c_{rd} \in [0; 1] \quad (5.30)$$

The torque about CoM stemming from radiation pressure can then be found as:

$$\bar{\tau}_r = \sum_{i=0}^N [\bar{r}_{rs,i} \times d\bar{F}_{rs,i} + \bar{r}_{a,i} \times d\bar{F}_{a,i} + \bar{r}_{rd,i} \times d\bar{F}_{rd,i}] \quad (5.31)$$

Where:

$\bar{r}_{rs,i}$ ,  $\bar{r}_{rd,i}$  &  $\bar{r}_{a,i}$  is vectors from CoM to the  $i$ 'eth surface element [.] for each of the three types of radiation pressure

From Equations 5.27, 5.28 and 5.29 it can be seen that the magnitude of radiation pressure depends primarily on the attitude of the spacecraft and the surface material impacted. The attitude of the spacecraft is the parameter, which determines the size of the surface area impacted by the radiation, and the surface material determines if the radiation pressure is caused by difusely reflection, specularly reflection or absorption.

## 5.4 Actuator Models

As shown in section 5.3 the satellite experience different types of disturbances when in orbit. These disturbance torques will quickly reorientate the spacecraft unless it generates an equally large torque to counter the disturbances. This is a job for the ADCS, which in this thesis is equipped with reaction wheels to acquire disturbance rejection and active attitude control by generating corrective torques on the satellite body. The model of the reaction wheels will be described through out this section.

### 5.4.1 Reaction wheel

Source: [Sidi 97] [Franklin 10]

Euler's moment equations states that, any external torques acting on the rigid body translates to the angular momentum of the spacecraft. Active attitude control is accomplished by translating mechanical momentum to the principal axis of the rigid body of the spacecraft, by use of reaction wheels.

A three wheeled configuration, where each reaction wheel is placed parallel to a principal axis of the body, simplifies the control problem e.g. by having direct translation of moment between reaction wheel momentum and the spacecraft principal axes. This actuator configuration has no redundancy and looses controllability if an actuator is damaged. This reason justifies the use of a fourth reaction wheel where the configuration of these will be described in the following sections.

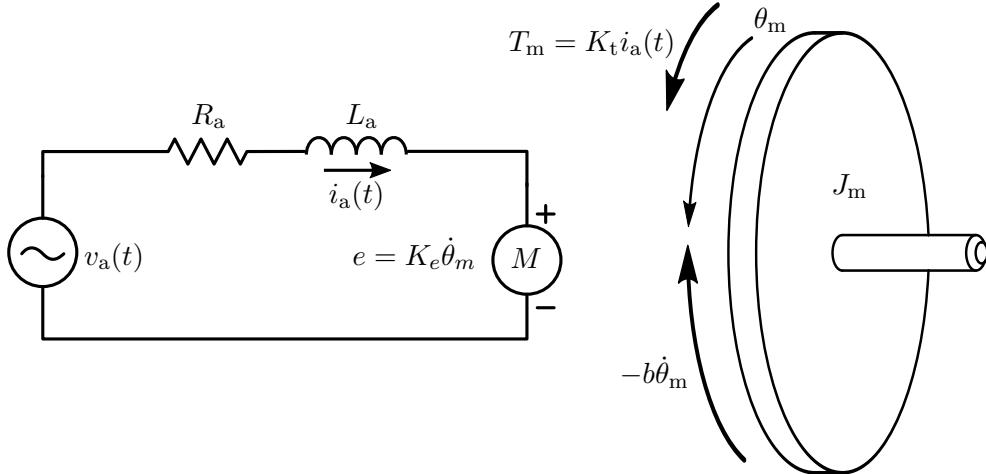
An ACS with four actuators can be described as an over-actuated control system for which presents the problem of allocating the control signal over all four actuators. This topic will be described in the Control Allocation section. First a dynamic model of the motors generating the torque is described.

### Dynamic Model

Source: [Xia 12]

The reaction wheels momentum is generated by four BLDCs. Compared to a traditional DC Motor (DC), the BLDC windings is energised dependant on the rotor position and controlled by a three-phased signal. For each phase the winding is conducted the electromagnetic torque and back-EMF is the same as for a DC motor. This means that the same models for a DC motor can be used when modelling a BLDC motor.

Each motor is modelled as a dynamic system with an electrical and mechanical part as shown in Figure 5.6.



**Figure 5.6:** Shows the electric circuit of the armature to the left and the free-body diagram of the rotor to the right.

The free-body diagram in Figure 5.6 shows two torques affecting the rotor  $T_m$  and the friction  $-b\dot{\theta}_m$  acting in the opposite direction of the rotation. The rotational equivalent of Newton's law of motion states that the sum of all external moments about the CoM of a rotating body is given as

$$\sum \tau_{\text{ext}} = J\dot{\omega}$$

Where  $J$  is the one-dimensional inertia of the rotating body in the rotation axis and  $\dot{\omega}$  is the angular acceleration. The equation of angular motion for the mechanical part of the motor is given in equation 5.32.

$$J_m \ddot{\theta}_m = K_t i_a - b\dot{\theta}_m \quad (5.32)$$

where:

$K_t$	is the motor torque constant	$\left[ \frac{\text{Nm}}{\text{A}} \right]$
$i_a$	is the armature current	[A]
$b$	is the motor friction coefficient	$\left[ \frac{\text{Nm} \cdot \text{s}}{\text{rad}} \right]$

The electrical part of the motor is given as shown in Figure 5.6 is given as a RL circuit and modelled as in equation

$$L_a \frac{d}{dt} i_a(t) + R_a i_a(t) = v_a(t) - K_e \omega_m \quad (5.33)$$

where:

$L_a$	is the armature self inductance	[H]
$R_a$	is the armature electrical resistance	[\Omega]
$v_a$	is the armature voltage	[V]
$K_e$	is the motor friction coefficient	$\left[ \frac{V \cdot s}{rad} \right]$

where for a BLDC the current running through the windings is controlled by the three-phases generated by the motor driver. Each phase is separated by a 120 [deg] which means that there will effectively only be running current in two of the windings meaning that the motor coefficients should be measured from phase to phase.

The dynamics of the electrical circuit induction relative to the dynamics of the mechanical motion is often said to have negligible effect and is therefore neglected. The combination of equation of motion from equation 5.32 and electrical dynamics from equation 5.33 is then given as equation 5.34.

$$J_m \dot{\omega}_m = - \left( \frac{K_t K_e}{R_a} + b \right) \omega_m + \frac{K_t}{R_a} v_a(t) \quad (5.34)$$

As seen from equation 5.34,  $J_m \dot{\omega}_m = \tau_m$  which is the sum of all torques on the motor shaft. Then it is known from Newton's third law of motion that for every action there is an equally large reaction and in the case of a rotating body the action exerted on the motor shaft is the motor torque  $\tau_m$  which has an equally large reaction  $-\tau_m$  acting on the mount on which the motor is mounted in the satellite rig.

The reaction torque  $-\tau_m$  is directly translated to the satellite body if the motor is pointing in the direction of the principal axis of the body. To suppress disturbance torques and reorientate the satellite, the reaction torque generated by the motors must be large enough. This is accomplished by increasing the inertia on the motor shaft. As seen from equation 5.34, the torque is directly dependent on the inertia  $J_m$  and the angular acceleration  $\dot{\omega}_m$ .

$$\tau_{rw} = -(J_w + J_m) \dot{\omega}_m \quad (5.35)$$

where  $\tau_{rw}$  is the reaction torque acting on the satellite body and  $J_w$  is the inertia of the reaction wheel. Since  $J_w \gg J_m$  the shaft inertia  $J_m$  is neglected. The dimensioning of the reaction wheels is also dependent on the number of actuators used and their configuration.

### Reaction Wheel Configuration

As described in the introduction, the use of four reaction wheels increases the reliability of the ACS by having redundancy if one motor fails. The configuration of these reaction wheels should be orientated such that the moment generated by each reaction wheel has a torque components in at least two of the three principal axis of the satellite body.

The ALPHASAT attitude control is equipped with four reaction wheels in a Tetrahedron configuration. This is due to the fact that the Tetrahedron configuration has

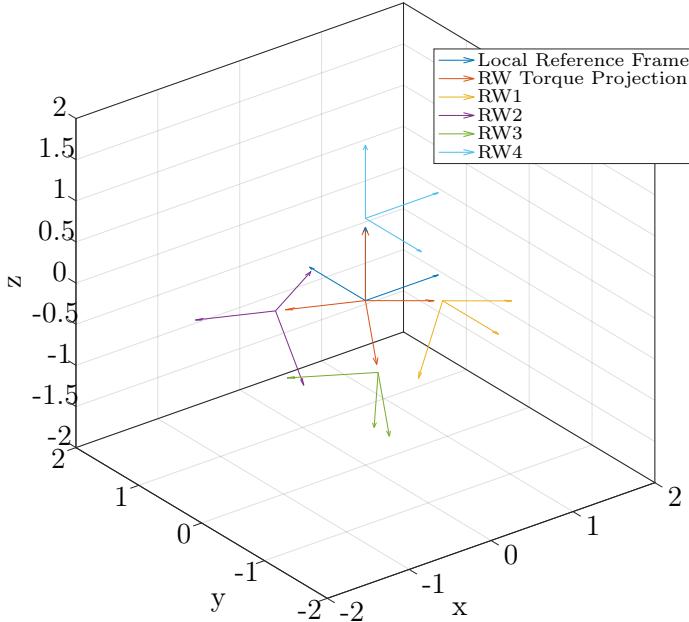
redundancy and the capability of zero angular momentum transition. The configuration is also capable of performing twice the maximum torque of a reaction wheel on a single axis.

All the reaction wheels in the tetrahedron configuration is orientated such that all angular momentums cancel out each other, when running with the same biased angular velocity. The distribution of the angular moments of the reaction wheels  $\tau_i, i \in [1; 4]$  is projected onto the tree-dimensional control space by the distribution matrix  $\underline{\mathbf{P}}_w$  as shown in equation 5.36.

$$\begin{bmatrix} \tau_{cx} \\ \tau_{cy} \\ \tau_{cz} \end{bmatrix} = \underline{\mathbf{P}}_w \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} \quad (5.36)$$

where  $\underline{\mathbf{P}}_w$  is the Tetrahedron configurations distribution matrix which is defined to have a row sum of zero, which means that all torques generated by the reaction wheels will chancel out each other when running at same biased angular velocity and is given by:

$$\underline{\mathbf{P}}_w = \begin{bmatrix} \cos \beta & -\cos \beta \cos \alpha & -\cos \beta \cos \alpha & 0 \\ 0 & \cos \beta \cos \frac{\alpha}{2} & -\cos \beta \cos \frac{\alpha}{2} & 0 \\ -\sin \beta & -\sin \beta & -\sin \beta & 1 \end{bmatrix} \quad (5.37)$$



**Figure 5.7:** Shows direction of each reaction wheel in the tetrahedron configuration.

The  $\bar{\tau}_c$  vector is calculated from the control laws. Since the there are four reaction the system is said to be over-actuated which means that there exists infinite solutions to the torque component  $\tau_i$ . This is seen from the distribution matrix  $\underline{\mathbf{P}}_w$  which is an under-determined system.

## Control Allocation

In order to find a solution to an over-actuated system given as.

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \underline{\mathbf{P}}_{\text{w}}^+ \begin{bmatrix} \tau_{\text{cx}} \\ \tau_{\text{cy}} \\ \tau_{\text{cz}} \end{bmatrix} \quad (5.38)$$

where  $\underline{\mathbf{P}}_{\text{w}}^+$  is a right Moore-Penrose pseudo-inverse matrix. The right pseudo-inverse of a under-determined system can be found as:

$$\underline{\mathbf{A}}^+ = \underline{\mathbf{A}}^\top (\underline{\mathbf{A}} \underline{\mathbf{A}}^\top)^{-1} \quad (5.39)$$

where  $\underline{\mathbf{A}}^+$  is the minimum norm solution to the over-actuated system, which means that the control torque will be distributed evenly amongst all four reaction wheels. Then the linear-map from the control torque  $\bar{\boldsymbol{\tau}}_c$  given in the CRF to a desired reaction wheel torque is given as.

$${}^w\bar{\boldsymbol{\tau}} = \underline{\mathbf{P}}_{\text{w}}^+ {}^c\bar{\boldsymbol{\tau}}_c \quad (5.40)$$

Which means that from now on, when ever a control value has been calculated, there exists a linear combination that maps the tree-dimensional control torque to the four-dimensional reaction wheel torque by the use of the Moore-Penrose pseudo-inverse matrix.

### 5.4.2 Dynamic Model of Reaction Wheel Configuration

The dynamic equations of reaction wheels mounted in the satellite frame is described by Euler's second law which states that a change in angular momentum  ${}^i\dot{\bar{\mathbf{L}}}(t)$  in an inertial reference frame is equal to the sum of all external forces  $\bar{\boldsymbol{\tau}}_{\text{ext}}$  acting on a rigid body, given as:

$$\begin{aligned} {}^i\dot{\bar{\boldsymbol{\tau}}}_{\text{ext}} &= \frac{d}{dt} {}^i\bar{\mathbf{L}}(t) \\ &= {}^i\dot{\bar{\mathbf{L}}}(t) \end{aligned} \quad (5.41)$$

Where as the change in angular momentum of a single reaction wheel is given as the rotation of the angular momentum between ECI and one Reaction Wheel Reference Frame (RWRF) and is given by multiplication of the rotation matrix  ${}^w_i\bar{\mathbf{R}}$

$$\begin{aligned} {}^w\dot{\bar{\boldsymbol{\tau}}}_{\text{ext}}(t) &= \frac{d}{dt} \left( {}^w_i\bar{\mathbf{R}} {}^i\bar{\mathbf{L}}(t) \right) \\ &= {}^w_i\dot{\bar{\mathbf{R}}} {}^i\bar{\mathbf{L}}(t) + {}^w_i\bar{\mathbf{R}} {}^i\dot{\bar{\mathbf{L}}}(t) \end{aligned} \quad (5.42)$$

where the derivative of the rotation matrix is expressed as the product between the skew-symmetric cross product operator defined in equation 5.16 and the rotation matrix from ECI to RWRF.

$${}^w_i\dot{\bar{\mathbf{R}}} = -\underline{\mathbf{S}}({}^w\bar{\boldsymbol{\omega}}) {}^w_i\bar{\mathbf{R}} \quad (5.43)$$

Using the rotational derivative expressed by equation 5.43 in equation 5.42 the following is seen.

$$\begin{aligned} {}^w\dot{\bar{\boldsymbol{\tau}}}_{\text{ext}}(t) &= -\underline{\mathbf{S}}({}^w\bar{\boldsymbol{\omega}}(t)) {}^w_i\bar{\mathbf{R}} {}^i\bar{\mathbf{L}}(t) + {}^w_i\bar{\mathbf{R}} {}^i\dot{\bar{\mathbf{L}}}(t) \\ &= -\underline{\mathbf{S}}({}^w\bar{\boldsymbol{\omega}}(t)) {}^w\bar{\mathbf{L}}(t) + {}^w\dot{\bar{\mathbf{L}}}(t) \end{aligned} \quad (5.44)$$

Further from this it can be seen that the change of angular momentum of a reaction wheel mounted in the satellite body is equal to the sum of the Coriolis force and the external forces.

$${}^w \dot{\bar{\mathbf{L}}}(t) = -\underline{\mathbf{S}}({}^w \bar{\boldsymbol{\omega}}(t)) {}^w \bar{\mathbf{L}}(t) + {}^w \bar{\tau}_{\text{ext}}(t) \quad (5.45)$$

where  ${}^w \bar{\tau}_{\text{ext}}(t)$  is the torque generated by the BLDC motor and  ${}^w \bar{\boldsymbol{\omega}}$  is the sum of the rotational velocity of the satellite in RWRF and the reaction wheel given as.

$${}^w \bar{\boldsymbol{\omega}}(t) = {}^w \bar{\boldsymbol{\omega}}_s(t) + {}^w \bar{\boldsymbol{\omega}}_w(t) \quad (5.46)$$

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

The change in angular momentum  ${}^w \dot{\bar{\mathbf{L}}}(t)$  is now described for one wheel in the RWRF. The torque from the reaction wheel is described in RWRF and is translated to the CRF where the MoI of the satellite is defined. Therefore the change in angular momentum in the CRF caused by the reaction wheel is described as:

$${}^c \dot{\bar{\mathbf{L}}}_w(t) = -{}^c \underline{\mathbf{R}} {}^w \dot{\bar{\mathbf{L}}}_w(t) \quad (5.47)$$

The sign change of the torque exerted on the CRF is explained by Newtons third law of motion, that states, for every action acting on a rigid body there exists an equally large opposite reaction, which also applies for rotational bodies. This implies that the change in angular momentum on the reaction wheel has an opposite effect on the satellite CRF. The rotational dynamics described in equation 5.45 is only accompanying one reaction wheel. The total change in angular momentum is then given as the sum of all moments generated by the reaction wheels. It should be noted that for every reaction wheel, a unique rotation matrix exists and is denoted by  $i \in [1; 4]$

$${}^c \dot{\bar{\mathbf{L}}}_{\text{th}}(t) = -\sum_{i=1}^4 {}^c \dot{\bar{\mathbf{L}}}_{w,i}(t) \quad (5.48)$$

The resulting change in angular momentum  ${}^c \dot{\bar{\mathbf{L}}}_{\text{th}}(t)$  is then added to the external torques in the dynamic model of the satellite given in Section 5.2.

## 5.5 Summary

Now the models for both the satellite kinematics, dynamics and the actuator model but also the disturbance torques, a spacecraft can expect to meet in a space environment, has been presented. The satellite kinematics and dynamics will be used in simulations of the satellite in order to describe the spacecraft's behaviour both in terms of time and exerted torques and forces. The disturbance models are used to model the space environment during the simulations, and the actuator models are used for both simulating the spacecraft behaviour but also in the design of the attitude control algorithms. In the following chapter the simulation environment is described.

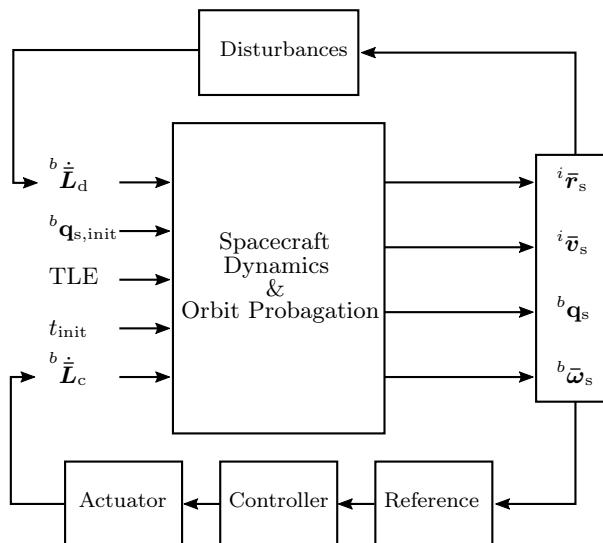
# Simulation Environment

Through the previous chapters, knowledge of the satellite model and its orientation in orbit, how it is propagated over time and disturbances is gained through analysis. In order to simulate the control algorithms, a simulation environment is built. The environment enables the possibilities to simulate weightlessness, vacuum and the disturbances in a space environment.

Through this Chapter, the elements of the simulation environment is described. First an outline of the simulation environment is described and then a short description of the simulation blocks used.

## 6.1 Simulink Implementation

The purpose of the simulation environment is for verification and performance test of both the linear and non-linear regulator designed in the Sections 7.1 and 7.2. The scope of this project is not to design attitude determination, but to investigate regulator design for CubeSats. The basic structure of the simulation environment is illustrated in Figure 6.1.



**Figure 6.1:** Shows the structure of the simulation environment, where the all the system models for spacecraft dynamics and orbit propagation is collected in a single dynamic block.

The system block illustrated in Figure 6.1, includes the spacecraft dynamics given by the models obtained in Sections 5.1 and 5.2, which is a function of external torques and initial conditions, where:

- ${}^b\dot{\tilde{L}}_c$  is the control torque from the actuators
- ${}^b\dot{\tilde{L}}_d$  is the disturbance torques
- ${}^b\mathbf{q}_{s,\text{init}}$  is the initial attitude of the satellite

The satellite orbit propagation model is based on the SGP4 model which estimates the position vector  ${}^i\bar{\mathbf{r}}_s$  and velocity vector  ${}^i\bar{\mathbf{v}}_s$  of a spacecraft where:

- TLE is the Two Line Element, which contains the orbit parameters
- $t_{\text{init}}$  is the initial simulation time, given in Julian Date (JD)

The outputs of the dynamic system block is then used to calculate the expected disturbances which depends on the orbital elements and the current attitude of the spacecraft. The output is further used in the regulator to calculate the desired control signal and following a given attitude reference.

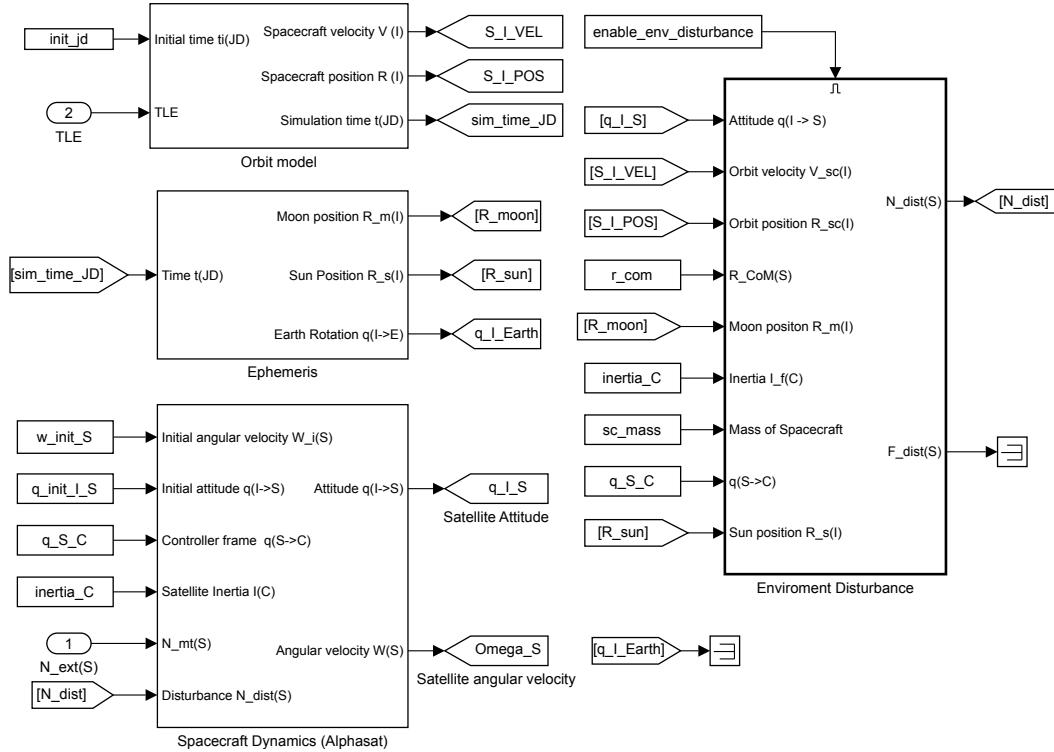
## 6.2 The AAUSAT Library

The environment is built in Matlab/Simulink, where the AAUSAT library is defined. The simulation environment build for the ALPHASAT is based on a few selected building blocks used for AAUSAT3 by [Jensen 10].

The AAUSAT library contains reusable building blocks for simulating LEO and the corresponding environmental disturbances found in space. The library has been developed over the past decade by former student groups through the campaigns of AAUSAT-II and AAUSAT3. The following building blocks from the library have been used:

- Disturbance models
- Ephemeris models
- SGP4 model

The disturbance models is built on the same principles as described in Section 5.3. The complete model of the satellite dynamics, orbit propagation and external torques is shown in Figure 6.2.



**Figure 6.2:** Shows the environment blocks of the ALPHASAT, with the spacecraft dynamics, orbit models and the disturbances seen from the spacecraft.

This thesis has contributed to this library by developing:

- Satellite system models
- Tetrahedron actuator model
- Attitude Estimator Emulator
- Target reference system
- Animation environment

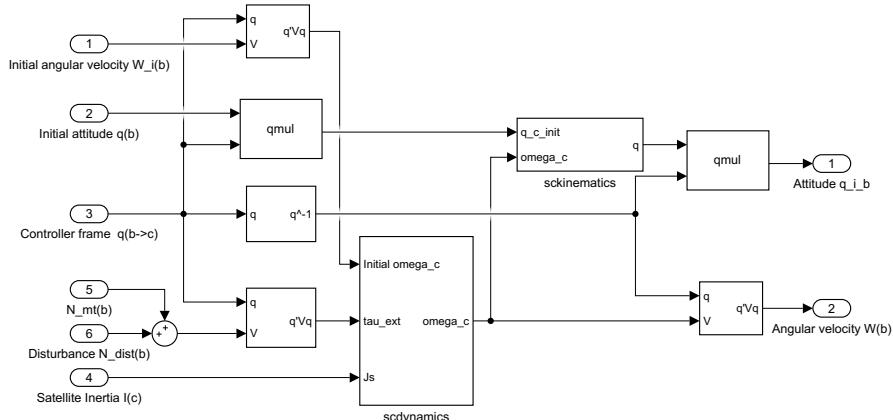
The target reference system is designed to simulate the case where the satellite is tracking multiple ground stations during orbit. Furthermore, an animation environment has been developed, in order to intuitively illustrate attitude manoeuvres during orbit simulations. A Preview of the 3D animation environment is shown in Figure 6.3.



**Figure 6.3:** Shows the Simulink 3D animation environment designed for intuitive illustrating satellite attitude manoeuvres.

### 6.3 Spacecraft Dynamics Model

The satellite model block is built of a kinematic and dynamic block. The input attitude, angular velocity and external torques is given in BRF. These are then rotated to the satellite CRF where the inertia matrix is diagonal. This rotation simplifies the equations. External torques and the initial angular velocity is then fed to the “scdynamics” block where the satellite dynamic is defined as shown in Figure 6.4.



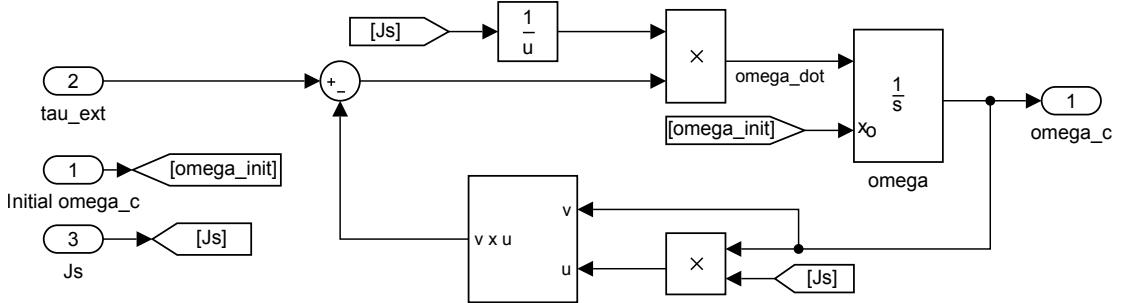
**Figure 6.4:** Shows to Simulink block of the spacecraft system model.

The “scskinematics” block uses the initial attitude quaternion to extrapolate the change in attitude given an estimated angular velocity from the satellite dynamics block. The “scskinematics” block is based on the kinematic equation 6.1.

$${}^c\dot{\mathbf{q}}_s = \frac{1}{2}\underline{\Omega}({}^c\bar{\boldsymbol{\omega}}_s){}^c\mathbf{q}_s \quad (6.1)$$

The output quaternion of the kinematic block is then rotated back to the satellite BRF.

The “scdynamics” is shown in Figure 6.5 where the sum of all external torques acting on the satellite CRF is integrated to find the resulting angular velocity. The angular velocity is then used to find the corresponding Coriolis torque acting on the satellite CRF.



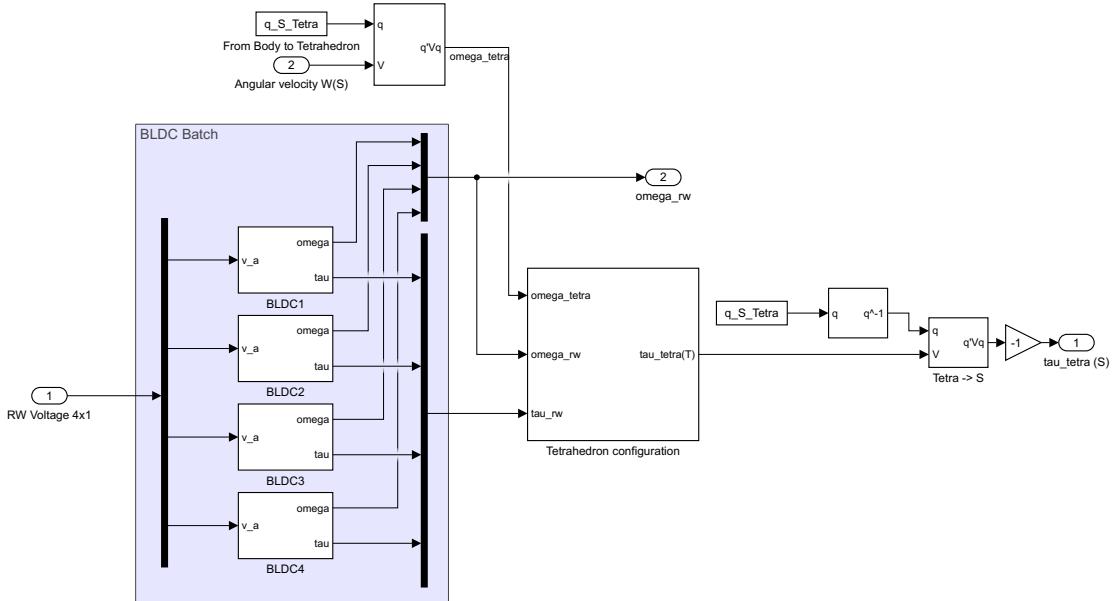
**Figure 6.5:** Shows the Simulink block of the spacecraft dynamics.

The “scdynamics” block is based on the satellite dynamics equation 6.2.

$${}^c\dot{\boldsymbol{\omega}}_s = \underline{\mathbf{J}}_s^{-1} [-\underline{\mathbf{S}}({}^c\bar{\boldsymbol{\omega}}_s) \underline{\mathbf{J}}_s {}^c\bar{\boldsymbol{\omega}}_s + {}^c\bar{\tau}_{ext}] \quad (6.2)$$

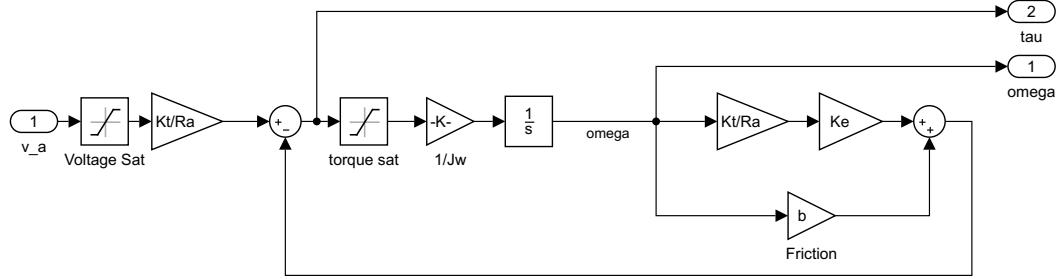
## 6.4 Actuator Model

The actuators used in this simulation environment is four BLDCs in a tetrahedron configuration mounted with reaction wheels. Each motor is modelled independently and is described in section 5.4.1. The motors are fed with a voltage input, which controls the speed of the motor. The change in angular momentum of the reaction wheel will then act as a torque on the body. Each motor torque is then projected and summed together in the “Tetrahedron configuration” block and then rotated back to the satellite BRF as shown in Figure 6.6.



**Figure 6.6:** Shows the Tetrahedron actuator model.

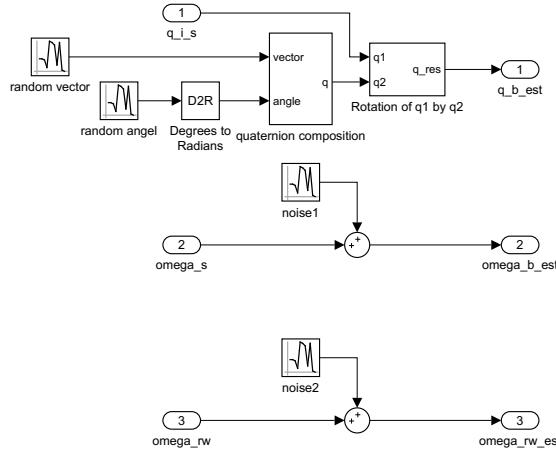
The dynamic model of the BLDC is based on the simplified first-order model described by [Xia 12]. The block includes saturation terms as upper and lower bounds on the armature voltage and maximum rated torque one motor can provide.



**Figure 6.7:** Shows the model of a simplified BLDC.

## 6.5 Attitude Estimator Emulator

In order to simulate the feedback from an attitude estimator, with its given uncertainties on the estimated states, an attitude estimator emulator has been build as shown in Figure 6.8.



**Figure 6.8:** Shows the block which should emulate the uncertainties and noise from which is propagated through an estimator.

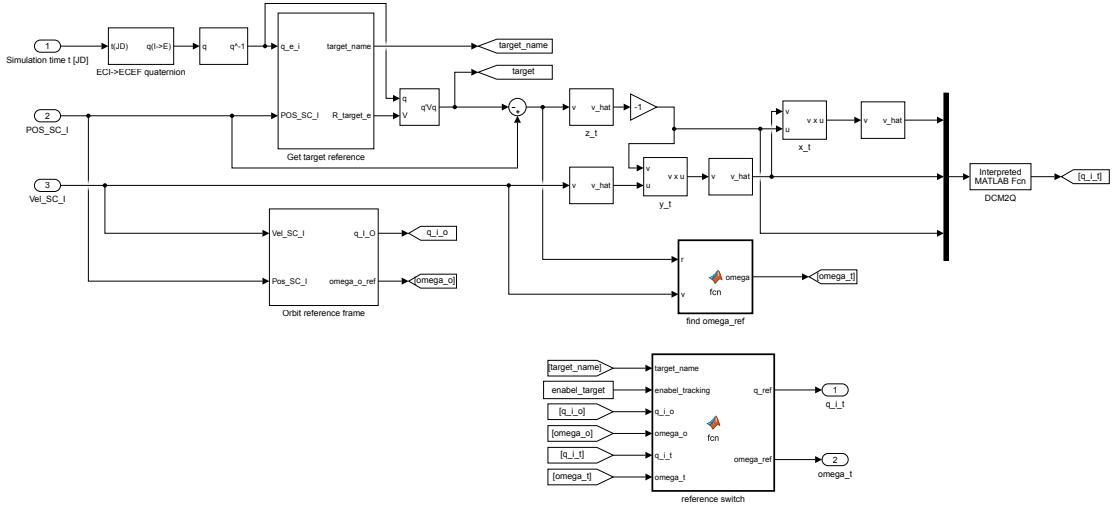
The function of the emulator is to add noise to the feedback from the deterministic model so that the feedback imitates the specifications of an attitude estimator, resulting in the property to test the robustness of the regulator design.

The size of the noise term summed on the attitude  ${}^b\mathbf{q}_s$  and the angular velocity  ${}^b\bar{\omega}_s$  of the spacecraft follows the attitude estimator requirements specified by [Jensen 10]. This results in an estimated attitude within  $\pm 5$  [deg] and a sample rate of 1 Hz.

## 6.6 Target Reference Generator

Source: [Curtis 05]

The reference generator is designed in order to simulate a logic component on the spacecraft, which should choose a target reference from a list of interest points. This block uses an estimate of the rotation of the ECEF and the satellites positions vector  ${}^i\bar{r}_s$  to calculate which interest points on the Earth's surface is in line of sight and then choose the one closest to the spacecraft.



**Figure 6.9:** Shows the Tetrahedron actuator model.

The block ‘‘Get target Reference’’ generates a set of position vectors  ${}^i\bar{\mathbf{R}}_{t,i}$  where  $i = [0, \dots, n]$  and  $n$  is the number of interest points in the ‘‘target\_list’’ shown below:

```
%> Define targets in the format {name, [lat,lon,alt]}
2 targets = {'Aalborg HQ', [57.013835, 9.987509, 15];
    'Thule Air Base', [76.5322, -68.0731, 500];
    'Paderborn', [51.5334, 8.8725, 10]
    'Vienna', [48.206251, 16.373879, 542];
    'Lisboa', [38.718298, -9.142485, 200];
    'Sao Paulo', [-2.000000, -44.600000, 10];
    };
8
save('target_list.mat','targets');
```

From the spherical coordinates given by the latitude and longitude, a unit vector pointing in the direction of the target can be found as shown in equation 6.3. This vector is fixed in the ECEF.

$${}^e\bar{\mathbf{r}}_{t,i} = \begin{bmatrix} \cos(\theta_{\text{lat},i}) \cos(\theta_{\text{lon},i}) \\ \cos(\theta_{\text{lat},i}) \sin(\theta_{\text{lon},i}) \\ \sin(\theta_{\text{lat},i}) \end{bmatrix} \quad (6.3)$$

The position vector is then given by the unit vector multiplied with scalar sum of the Earth’s radius and the altitude of the target from sea level. The position vector is given by equation 6.4.

$${}^e\bar{\mathbf{R}}_{t,i} = (R_{\text{alt},i} + R_e) {}^e\bar{\mathbf{r}}_{t,i} \quad (6.4)$$

The position vector is then rotated into the ECI by the inverse rotation found in Section 4.3.1. The reason for this rotation is because the Spacecraft position vector is given in the ECI.

$${}^i\bar{\mathbf{R}}_{t,i} = {}_e^i\mathbf{q}^* {}^e\bar{\mathbf{R}}_{t,i} {}_e^i\mathbf{q} \quad (6.5)$$

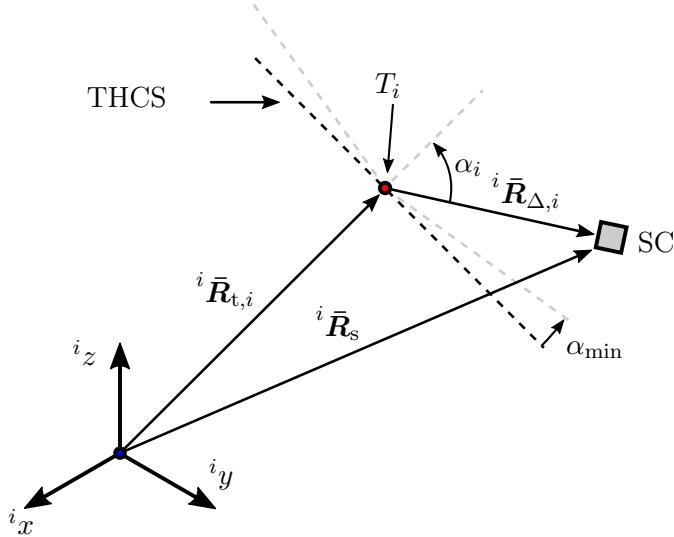
Now that the target position vector is represented in ECI, it is possible to take the difference between the two vectors in order to find a vector pointing from the spacecraft to the target as shown in equation 6.6.

$${}^i\bar{\mathbf{R}}_{\Delta,i} = {}^i\bar{\mathbf{R}}_s - {}^i\bar{\mathbf{R}}_{t,i} \quad (6.6)$$

When the satellite is in line of sight, it is said to be over the THCS which is explained in section 4.3.3. This is true if the angle between the two position vectors  $|\alpha_i| < 90$  [deg].

$$\alpha_i = \cos^{-1} \left( \frac{^i\bar{\mathbf{R}}_{\Delta,i}^\top \cdot ^i\bar{\mathbf{R}}_{t,i}}{\| ^i\bar{\mathbf{R}}_{\Delta,i} \| \| ^i\bar{\mathbf{R}}_{t,i} \|} \right) \quad (6.7)$$

Depending on the surroundings of the target or groundstation, a minimum elevation can be set, which is then subtracted from the  $\pm 90$  [deg] threshold. This means that the “Get target reference” block only triggers on targets which is below  $|90 - \alpha_{\min}|$  [deg]. This is also illustrated in Figure 6.10.



**Figure 6.10:** Shows the satellite and one specific target  $T_i$  where  $\alpha_i$  is the inclination of the satellite over the THCS.

As indicated by the notation, this calculation is performed for each interest point in the list which means that more targets could potentially be in line of sight. In that case, the algorithm chooses the target closest to the satellite.

When a target has been selected, the difference vector  $^i\bar{\mathbf{R}}_{\Delta,i}$  pointing from the spacecraft to the target is used to find the TRF explained in 4.3.3. In cases where no targets is in line of sight, the target reference generator will use the Nadir reference frame.

## 6.7 Simulation Setup

The simulation environment for ALPHASAT, as described, enables spacecraft simulation in a theoretical space environment with a selection of worst case disturbances a satellite will endure in space. Through this Section, the setup used for evaluating the controllers developed in Chapter 7 is described.

As mentioned in the Introduction and Mission Section 1.1, the hypothetical satellite ALPHASAT is launched into orbit from the ISS by an arbitrary deployer. This type of launch will give the satellite approximately the same orbital parameters as the ISS, but will change over time due to the difference in mass and surface area compared to the ISS.

When launched from the deployer, the satellite may experience tumbling, sometimes with a greater effect, but this is expected. This can be due to skewness in the inhibit

switches in the bottom of the CubeSat which is the only contact area between the CubeSat and the deployer. The expected tumble rate that the CubeSat can experience is approximately 10 deg/s. A De-tumble algorithm will then dampen the initial spin of the satellite down to below  $\pm 0.3$  deg/s [Jensen 10]. The conditions after the detumbling has occurred is defined as the initial conditions for attitude control algorithms tested with the simulation.

### Simulation Specifications

In the following section, a list of the simulation conditions is specified. First the simulation time which is used to determine the satellites position in the given orbit.

---

```
t_sim = '2015-10-20T14:35Z'
```

---

Second the following TLE is used for simulating the ALPHASAT orbit given an ISS launch.

---

```
ALPHASAT
1 40949U 98067HA 16131.17243197 .00049328 00000-0 32059-3 0 9990
2 40949 51.6335 230.6137 0003739 51.3487 308.7846 15.75443623 34062
```

---

Given the initial simulation time and the TLE, the CubeSat is at an altitude of approximately 379 Km and flying over the Atlantic ocean in direction of the Germany where a ground station is positioned.

#### 6.7.1 Simulation Scenarios

The ALPHASAT will be simulated in two scenarios where the each controller is verified. The initial values is the same for both controllers in each simulation for comparison. The scenarios are:

- Nadir Pointing

*In this scenario, the satellite is initialised, as mentioned, in the state after it has been detumbled. The attitude is therefore initialised in an arbitrary direction. ALPHASAT should then begin to point Nadir and align with the ORF.*

- Ground Tracking

*In this scenario, the satellite is all ready pointing Nadir. A few seconds into the simulation ALPHASAT would be able to detect a target coming into line of sight. ALPHASAT should then break from pointing at Nadir to align with the given TRF. ALPHASAT should then keep pointing at the target until it is out of sight and then return to pointing Nadir.*

With these two scenarios defined it is now possible to simulate the control algorithm design. In the following chapter two control algorithms are designed; a linear and a non-linear, both of these will be simulated and the results will be presented in the respective sections.



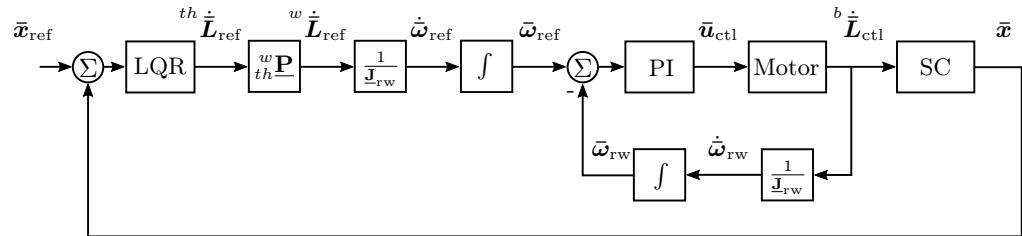
# Attitude Control System

In Chapter 5, a non-linear model of the satellite is derived, in which the satellite kinematics and dynamics is described. These models describe how the motion of the satellite is propagated when excited by external torques, e.g. actuators and disturbances.

In the following Sections, multiple control laws are designed, an LQR for comparison followed by design of a non-linear regulator; Sliding Mode Control (SMC). Each control law use quaternion attitude feedback compared with a reference attitude which results in a quaternion error, which will be calculated.

## 7.1 Linear Regulator

A linear regulator is designed, which is used for performance comparison with a non-linear regulator (SMC, Section 7.2). Through the design process it has been chosen to split the regulator into a LQR for attitude control and a PI regulator for each motor with an angular velocity as a reference. This setup is illustrated in Figure 7.1.



**Figure 7.1:** Shows a block diagram of the linear regulator setup where the output of the LQR is a desired control torque.

The LQR outputs a desired control torque which is then projected on to each motor direction and translated into a change in angular velocity reference for the PI regulator. The PI regulator ensures that the motors follows an angular velocity reference which lies within the bandwidth of the motors dynamics.

The inner loop, which regulates the motor speeds, is sampling two times faster than the fastest motor time constant, which is  $\approx 4$  [Hz], while the outer loop is sampling with 1 [Hz] which is the sampling rate given by the ADS [Jensen 10].

The motor dynamics is not included in the LQR design but is instead taken into account by ensuring that the closed-loop attitude control is 10-15 times lower than the bandwidth of the motor dynamics [Blanke 10].

### 7.1.1 Linear System Model

The non-linear system equation ?? is linearised in the point of operation where the satellite BRF is aligned with the ORF, which means the satellite is pointing Nadir and the angular velocities is infinitesimal small.

The states  $\bar{\mathbf{x}}$  measured at the output of the “SC” block in Figure 7.1 is given as:

$$\bar{\mathbf{x}} = \begin{bmatrix} {}^c\mathbf{q}_s \\ {}^c\bar{\boldsymbol{\omega}}_s \end{bmatrix} \quad (7.1)$$

where  ${}^c\mathbf{q}_s$  is the quaternion attitude representation and  ${}^c\bar{\boldsymbol{\omega}}_s$  is the satellite angular velocity. The states is given in CRF.

The system model is linearised with a first order Taylor approximation, where the general form is

$$f(x) \approx f(\check{x}) + \frac{\partial f(x)}{\partial x} \Big|_{x=\check{x}} \cdot \Delta \tilde{x} \quad (7.2)$$

$$x = \check{x} + \Delta \tilde{x} \quad (7.3)$$

where

- $x$  is the function variable
- $\check{x}$  is the point of linearisation
- $\Delta \tilde{x}$  is small deviation form the linearisation

The angular velocity of the satellite is then approximated to be equal to:

$${}^c\bar{\boldsymbol{\omega}}_s \approx {}^b\underline{\mathbf{R}} \begin{bmatrix} 0 \\ {}^b\check{\boldsymbol{\omega}}_o \\ 0 \end{bmatrix} + \Delta {}^c\tilde{\boldsymbol{\omega}}_s \quad (7.4)$$

where  $\Delta {}^c\tilde{\boldsymbol{\omega}}_s$  is small perturbations in the angular velocity and  ${}^b\check{\boldsymbol{\omega}}_{ref}$  is the point of linearisation, which is the angular velocity of the BRF relative to ECI when aligned with the ORF.

The attitude  ${}^c\mathbf{q}_s$  is linearised in the orientation of the ORF as shown 7.5.

$${}^c\mathbf{q}_s \approx {}^c\check{\mathbf{q}}_o \otimes \Delta {}^c\tilde{\mathbf{q}}_s \quad (7.5)$$

Where  ${}^c\check{\mathbf{q}}_o$  is the orbit attitude and linearisation point, and  $\Delta {}^c\tilde{\mathbf{q}}_s$  is the deviation from the point of operation. Furthermore it is seen when left multiplying with the conjugated rotation, the error quaternion appears.

$$\begin{aligned} {}^c\check{\mathbf{q}}_o^* \otimes {}^c\mathbf{q}_s &\approx {}^c\check{\mathbf{q}}_o^* \otimes {}^c\check{\mathbf{q}}_o \otimes \Delta {}^c\tilde{\mathbf{q}}_s \\ {}^c\mathbf{q}_e &\approx \Delta {}^c\tilde{\mathbf{q}}_s \end{aligned} \quad (7.6)$$

From this expression it is seen that the point of linearisation is indeed defined where the quaternion error  ${}^c\mathbf{q}_e = [0 \ 0 \ 0 \ 1]^\top$ .

### 7.1.2 Linear Kinematics

From equation 7.6 the kinematic equation can be written as small changes in the attitude for each time interval

$$\Delta {}^c \dot{\tilde{\mathbf{q}}}_s = \frac{1}{2} \underline{\Omega}({}^c \bar{\omega}_s) \Delta {}^c \tilde{\mathbf{q}}_s \quad (7.7)$$

The attitude can as stated by equation 5.3 be approximated with the small angle approximation.

$$\Delta {}^c \tilde{\mathbf{q}}_s \approx \begin{bmatrix} \Delta {}^c \tilde{\mathbf{q}}_s \\ 1 \end{bmatrix} \quad (7.8)$$

From this it is seen that  $\Delta \tilde{q}_{s4} \approx 1$  which makes  $\Delta \dot{\tilde{q}}_{s4} \approx 0$ . The kinematic equation is then reduced to the complex part of the quaternion derivative.

The reduced kinematic equation is then linearised in the point of operation.

$$\begin{aligned} {}^c \dot{\tilde{\mathbf{q}}}_s ({}^c \bar{\omega}_s, {}^c \bar{\mathbf{q}}_s) &\approx {}^c \dot{\tilde{\mathbf{q}}}_s ({}^c \check{\bar{\omega}}_o, {}^c \check{\bar{\mathbf{q}}}_o) + \left. \frac{\partial {}^c \dot{\tilde{\mathbf{q}}}_s ({}^c \bar{\omega}_s, {}^c \bar{\mathbf{q}}_s)}{\partial {}^c \bar{\mathbf{q}}_s} \right|_{{}^c \check{\bar{\omega}}_o, {}^c \check{\bar{\mathbf{q}}}_o} \cdot \Delta {}^c \tilde{\mathbf{q}}_s \\ &+ \left. \frac{\partial {}^c \dot{\tilde{\mathbf{q}}}_s ({}^c \bar{\omega}_s, {}^c \bar{\mathbf{q}}_s)}{\partial {}^c \bar{\omega}_s} \right|_{{}^c \check{\bar{\omega}}_o, {}^c \check{\bar{\mathbf{q}}}_o} \cdot \Delta {}^c \tilde{\bar{\omega}}_s \end{aligned} \quad (7.9)$$

where the Jacobian is given as

$$\left. \frac{\partial {}^c \dot{\tilde{\mathbf{q}}}_s ({}^c \bar{\omega}_s, {}^c \bar{\mathbf{q}}_s)}{\partial {}^c \bar{\mathbf{q}}_s} \right|_{{}^c \check{\bar{\omega}}_o, {}^c \check{\bar{\mathbf{q}}}_o} = \frac{1}{2} \underline{\mathbf{S}}({}^c \check{\bar{\omega}}_o) \quad (7.10)$$

$$\left. \frac{\partial {}^c \dot{\tilde{\mathbf{q}}}_s ({}^c \bar{\omega}_s, {}^c \bar{\mathbf{q}}_s)}{\partial {}^c \bar{\omega}_s} \right|_{{}^c \check{\bar{\omega}}_o, {}^c \check{\bar{\mathbf{q}}}_o} = \frac{1}{2} \underline{\mathbf{I}}_{3 \times 3} \quad (7.11)$$

where  $\Delta \tilde{q}_{s4} = 1$ .

### 7.1.3 Linearised Dynamic Model

The satellite dynamics is given as:

$${}^c \dot{\bar{\omega}}_s = \underline{\mathbf{J}}_s^{-1} \left[ -\underline{\mathbf{S}}({}^c \bar{\omega}_s) {}^c \bar{\mathbf{L}}_s - \underline{\mathbf{S}}({}^c \bar{\omega}_s) {}^c \bar{\mathbf{L}}_{th} + {}^c \dot{\bar{\mathbf{L}}}_{th} + {}^c \dot{\bar{\mathbf{L}}}_d \right] \quad (7.12)$$

The disturbance torque  ${}^c \dot{\bar{\mathbf{L}}}_d$  is assumed to have an expected value of zero and is therefore removed in the linearised equations [Jensen 10]. Furthermore, the Coriolis term from the resulting angular momentum of the actuators  $-\underline{\mathbf{S}}({}^c \bar{\omega}_s) {}^c \bar{\mathbf{L}}_{th} \approx 0$  and is neglected in the linearisation. This is due to the internal cancellation of momentum in the Tetrahedron configuration when operating at the biased angular velocities.

Left is the Coriolis term from the satellites angular momentum where the first order Taylor approximation is used.

$$-\underline{\mathbf{J}}_s^{-1} \underline{\mathbf{S}}({}^c \bar{\omega}_s) {}^c \bar{\mathbf{L}}_s \approx -\underline{\mathbf{J}}_s^{-1} \underline{\mathbf{S}}({}^c \check{\bar{\omega}}_o) \underline{\mathbf{J}}_s {}^c \check{\bar{\omega}}_o - \underline{\mathbf{J}}_s^{-1} \frac{\partial (\underline{\mathbf{S}}({}^c \check{\bar{\omega}}_o) \underline{\mathbf{J}}_s {}^c \check{\bar{\omega}}_o)}{\partial {}^c \check{\bar{\omega}}_o} \cdot \Delta {}^c \tilde{\bar{\omega}}_s \quad (7.13)$$

The Jacobian is then found by using the product rule when partially deriving crossproducts.

$$\underline{\mathbf{A}}_{{}^c \bar{\omega}_s} = -\underline{\mathbf{J}}_s^{-1} \frac{\partial (\underline{\mathbf{S}}({}^c \check{\bar{\omega}}_o) \underline{\mathbf{J}}_s {}^c \check{\bar{\omega}}_o)}{\partial {}^c \check{\bar{\omega}}_o} = \underline{\mathbf{J}}_s^{-1} [\underline{\mathbf{S}}(\underline{\mathbf{J}}_s {}^c \check{\bar{\omega}}_o) - \underline{\mathbf{S}}({}^c \check{\bar{\omega}}_o) \underline{\mathbf{J}}_s] \quad (7.14)$$

### 7.1.4 State-Space Model

For now the kinematic and dynamic equations has been linearised in a point of operation by a first order Taylor approximations. The derived Jacobians is now used to construct a state-space system of the satellite. Where the states  $\bar{\mathbf{x}}$  and inputs  $\bar{\mathbf{u}}$  is given as

$$\bar{\mathbf{x}} = \begin{bmatrix} {}^c\bar{\mathbf{q}}_e \\ {}^c\tilde{\boldsymbol{\omega}}_s \end{bmatrix}, \quad \bar{\mathbf{u}} = {}^c\dot{\bar{\mathbf{L}}}_{th} \quad (7.15)$$

Recall from equation 7.6 that  ${}^c\bar{\mathbf{q}}_e \approx \Delta {}^c\tilde{\mathbf{q}}_s$

$$\dot{\bar{\mathbf{x}}} = \begin{bmatrix} \frac{1}{2}\underline{\mathbf{S}}({}^c\tilde{\boldsymbol{\omega}}_o) & \frac{1}{2}\underline{\mathbf{I}}_{3 \times 3} \\ \underline{\mathbf{0}}_{3 \times 3} & \underline{\mathbf{A}}_{{}^c\tilde{\boldsymbol{\omega}}_s} \end{bmatrix} \bar{\mathbf{x}} + \begin{bmatrix} \underline{\mathbf{0}}_{3 \times 3} \\ -\underline{\mathbf{J}}_s^{-1} \end{bmatrix} \bar{\mathbf{u}} \quad (7.16)$$

### 7.1.5 Design of LQR

Source: [Sørensen 10]

At this point, a continuous state-space model of the satellite has been constructed in Equation (7.16). In the live implementation, the states would be measured in discrete time intervals, therefore a discrete state-space model is used with a discrete notation.

$$\bar{\mathbf{x}}_{k+1} = \underline{\Phi}\bar{\mathbf{x}}_k + \underline{\Gamma}\bar{\mathbf{u}}_k \quad (7.17)$$

The aim of the regulator is optimal state feedback in the sense that it finds an input signal  $\bar{\mathbf{u}}_k$  which brings the system to a zero state where  $\bar{\mathbf{x}} = \bar{\mathbf{0}}$  in an optimal solution to draw the system states towards the reference point with as little control effort as possible. This control problem is formulated by assigning a cost to each state and input. This is done by minimizing the deterministic cost function

$$J(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k) = \frac{1}{2} \sum_{k=0}^{\infty} (\bar{\mathbf{x}}_k^\top \underline{\mathbf{Q}} \bar{\mathbf{x}}_k + \bar{\mathbf{u}}_k^\top \underline{\mathbf{R}} \bar{\mathbf{u}}_k) \quad (7.18)$$

Where  $\underline{\mathbf{Q}} = \underline{\mathbf{Q}}^\top \geq 0$  and  $\underline{\mathbf{R}} = \underline{\mathbf{R}}^\top > 0$  are weighting matrices which punishes respectively the deviation of the state error and the control signal. These are used as design parameters and found by simulating the system.

The optimal control input to time  $k$  can then be found by

$$\bar{\mathbf{u}}_k = -\underline{\mathbf{L}}_k \bar{\mathbf{x}}_k \quad (7.19)$$

where the control gain  $\underline{\mathbf{L}}_k$  is given as

$$\underline{\mathbf{L}}_k = [\underline{\mathbf{R}} + \underline{\Gamma}^\top \underline{\mathbf{S}}_{k+1} \underline{\Gamma}]^{-1} \underline{\Gamma}^\top \underline{\mathbf{S}}_{k+1} \underline{\Phi} \quad (7.20)$$

and

$$\underline{\mathbf{S}}_k = \underline{\mathbf{Q}} + \underline{\Phi}^\top \underline{\mathbf{S}}_{k+1} [\underline{\Phi} - \underline{\Gamma} \underline{\mathbf{L}}_k] \quad (7.21)$$

The optimal control gain is found by recursive calculations of  $\underline{\mathbf{L}}_k$  and  $\underline{\mathbf{S}}_k$  for a given time horizon  $N$ . In this thesis an infinite time horizon LQR,  $N \rightarrow \infty$ , is designed where the feedback gains are constant. In this case  $\underline{\mathbf{L}}_0$  is found when the condition that the recursive function  $\underline{\mathbf{S}}_k$  converge where  $\underline{\mathbf{S}}_k \approx \underline{\mathbf{S}}_{k+1}$  which leads to an Algebraic Riccati Equation.

### 7.1.6 Implementation and Tuning

Now the general structure of the linear control system is set up. The design parameters for the attitude LQR is chosen such that the closed-loop dynamics obey the motor dynamics and meets the ACS requirements formulated in Section 3.2.

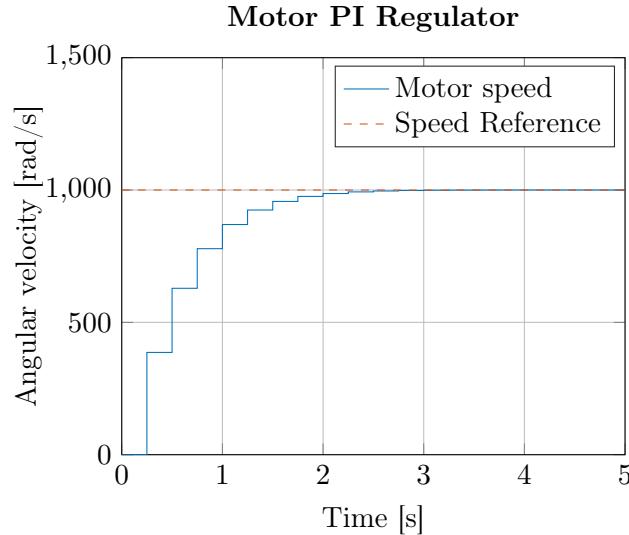
The motor PI regulator is tuned such that it is able to follow the given motor speed reference, with zero steady-state error and a rise time close to the motor open-loop response. The PI is given as:

$$C(s) = K_p + \frac{K_i}{s} \quad (7.22)$$

where the design parameters is found through tuning and is given as:

$$K_p = 0.0018 \quad , \quad K_i = 0.004 \quad (7.23)$$

It is clear that the motor speed control system is a Type 1 system and is able to follow a velocity reference, but in order to follow the reference given by the LQR, which in general is a velocity ramp, the reference bandwidth should be lower than the motor bandwidth.



**Figure 7.2:** Shows the step response of the motor PI regulator at 4 Hz sampling rate.

The LQR design parameters is often found through iterative tuning in simulations, but a good initial guess is suggested by Bryson's rule [Bryson 69]. In order to obtain acceptable values for the states  $\bar{x}$  and input  $\bar{u}$ , is to choose diagonal matrices for  $\underline{\mathbf{Q}}$  and  $\underline{\mathbf{R}}$ , where

$$Q_{ii} = \frac{1}{\text{Maximum state value } [x_i^2]}$$

$$R_{ii} = \frac{1}{\text{Maximum input value } [u_i^2]}$$

Thereafter  $\underline{\mathbf{Q}}$  and  $\underline{\mathbf{R}}$  are tuned relatively in order to obtain an acceptable weights between performance and control effort.

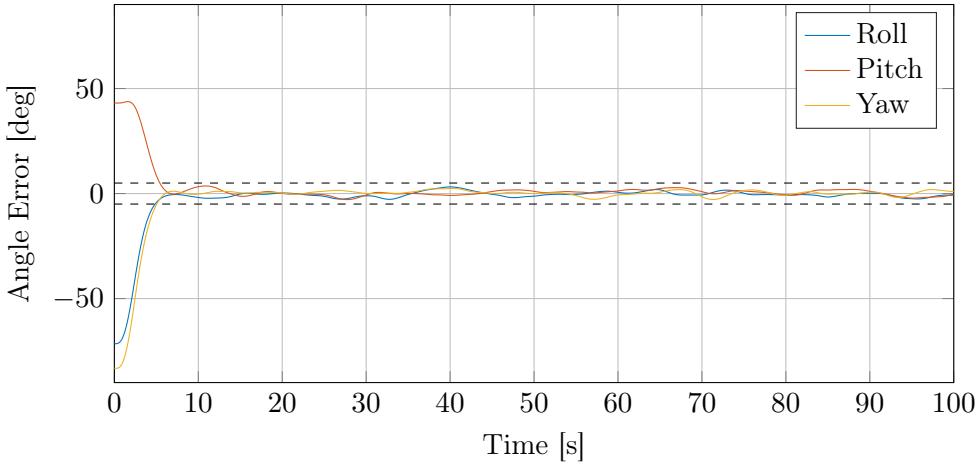
Now the linear attitude control has been developed and it is tested through the simulation environment.

### 7.1.7 Results

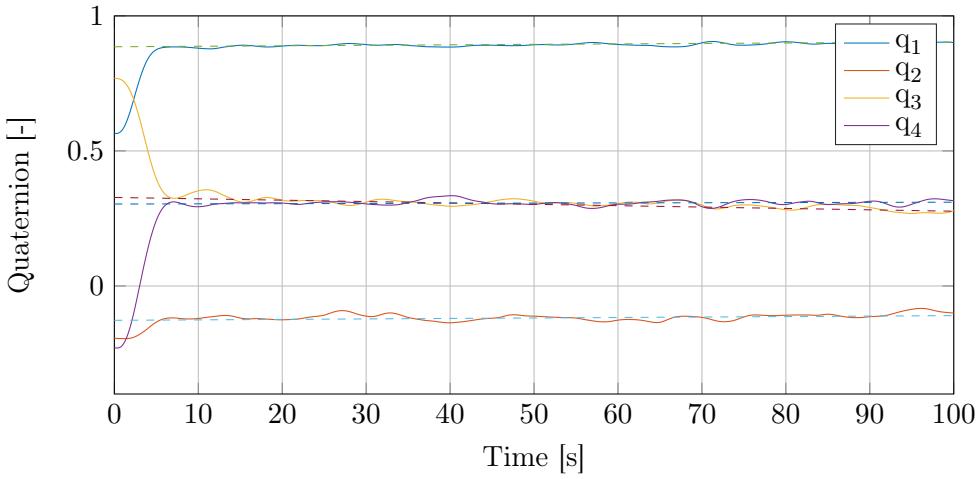
The linear regulator is tested through two scenarios. First Nadir pointing and then target tracking.

#### Nadir Pointing

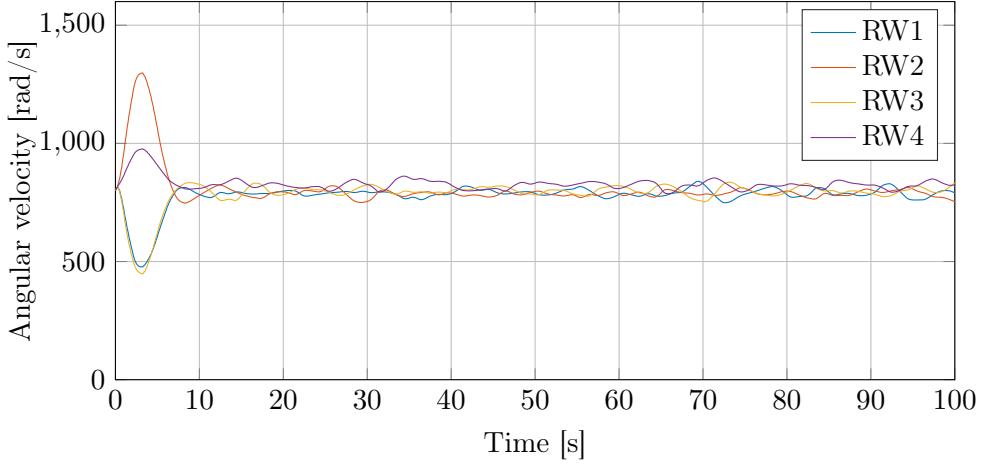
The first simulation shows the case where the satellite attitude control is initialised just after the de-tumbling has been performed. In this situation, the satellite is initialised with an absolute 90 deg attitude error and an initial spin of  $\approx 0.3$  deg/s [Jensen 10] in any axis. The satellite attitude control should then dampen the spin and correct the attitude to point Nadir.



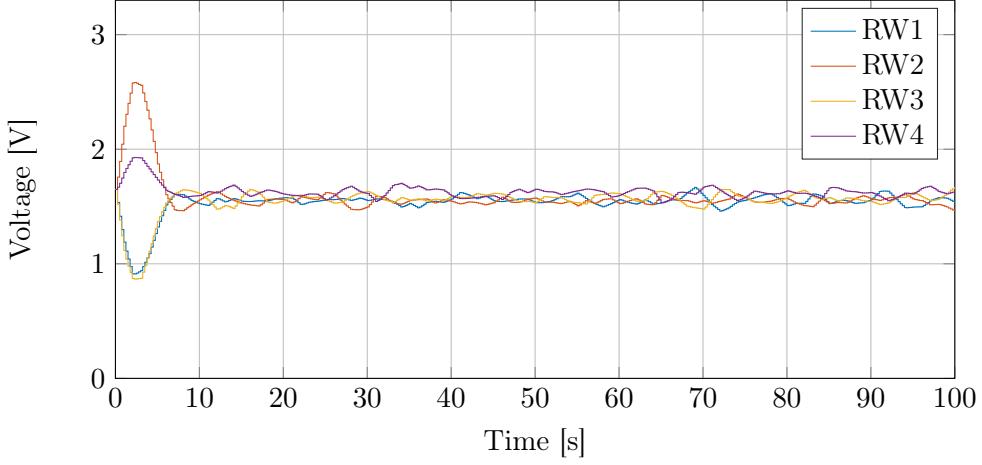
**Figure 7.3:** Shows the satellite attitude error in Euler angles, where the dashed lines represent the maximum error values.



**Figure 7.4:** Shows the satellite attitude  ${}^b\mathbf{q}_s$  relative to the ECI, where the dashed lines represent the quaternion attitude reference  ${}^b\mathbf{q}_{ref}$ .



**Figure 7.5:** Shows angular velocities on each motor in the Tetrahedron configuration.

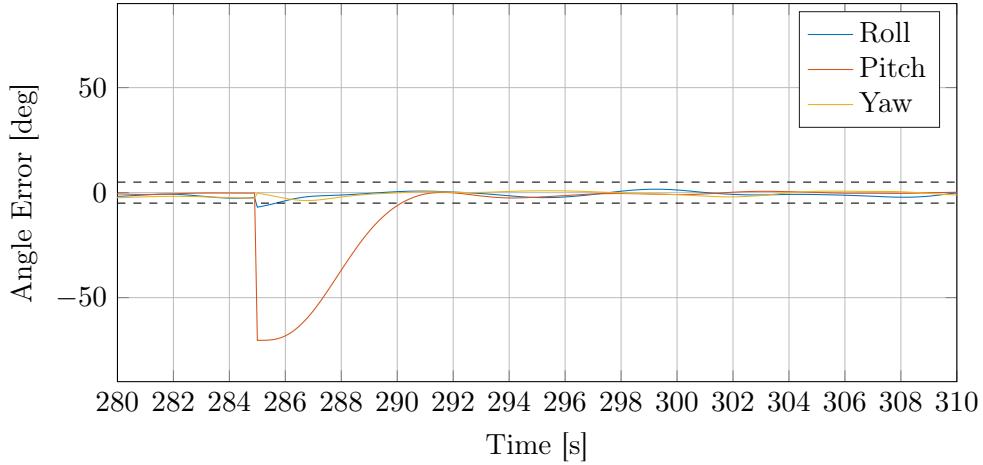


**Figure 7.6:** Shows the motor control signal.

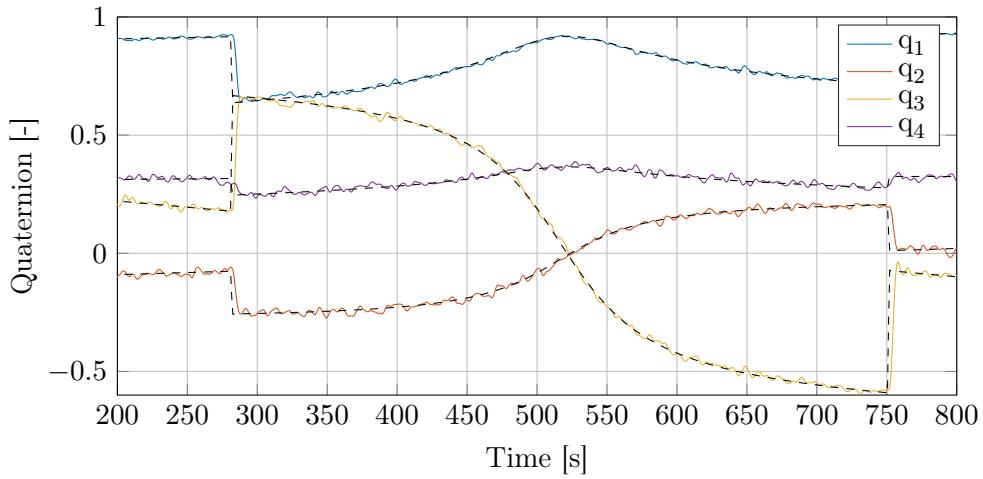
From Figure 7.3 it is seen that the attitude controller manages to slow the initial velocity down and rotate the satellite to align with the ORF and thereby point Nadir. The attitude is well within the boundaries of the attitude error requirements as it is seen in the given time period.

### Target Tracking

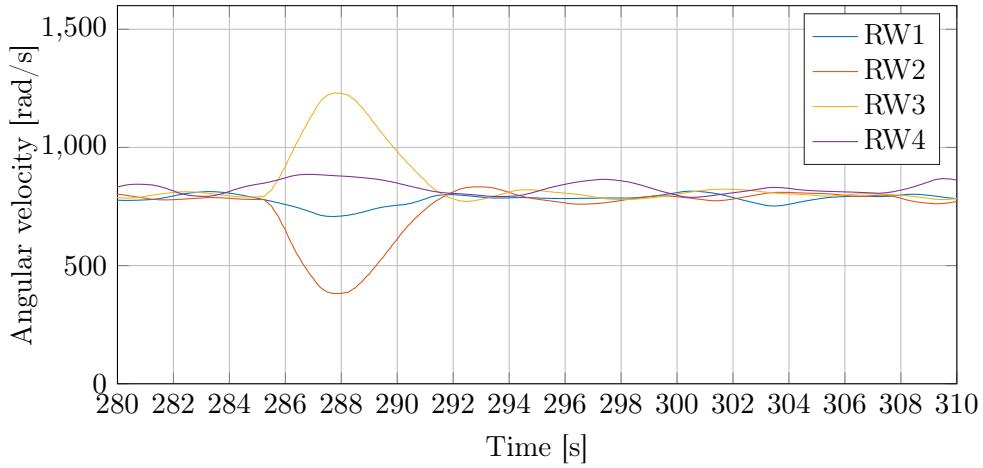
In the next scenario the satellite is initialised to point Nadir and an expected point of interest is coming into the satellites line of sight. The satellite should then break from the ORF to begin pointing at the specific TRF, when in sight. During the pass there will be an increase in the angular velocity reference from the TRF as the satellite flies over the interest point, which it should follow.



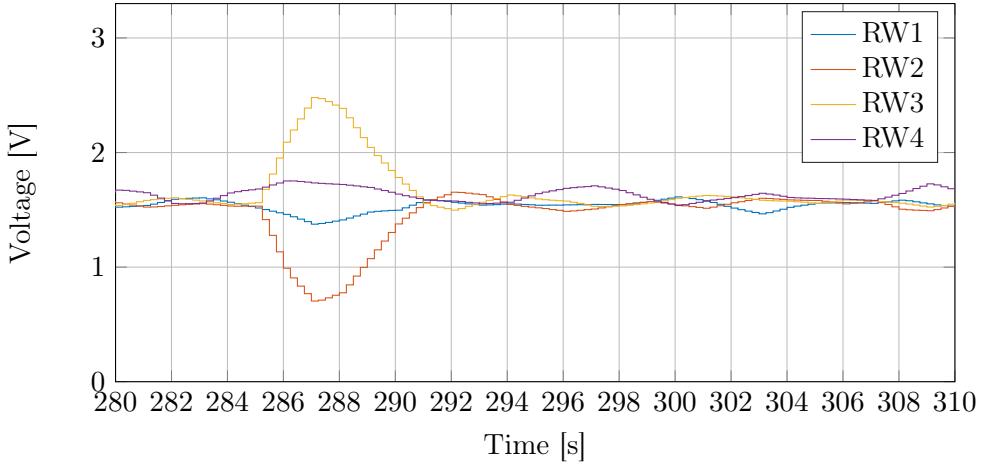
**Figure 7.7:** Shows the satellite attitude error in Euler angles, where the dashed lines represent the maximum error values.



**Figure 7.8:** Shows the satellite attitude  ${}^b\mathbf{q}_s$  relative to the ECI, where the dashed lines represent the quaternion attitude reference  ${}^b\mathbf{q}_{ref}$ .



**Figure 7.9:** Shows angular velocities on each motor in the Tetrahedron configuration.



**Figure 7.10:** Shows the motor control signal.

From Figure 7.7 it is clear to see when the target is insight. The satellite starts with an error in pitch of  $\approx -75$  deg which it then compensates for with a settling time of  $\approx 5$  sec, which is well under the required.

### 7.1.8 Summary

Through the simulations it can be seen that an attitude LQR in combination with a motor speed PI controller is working well within the requirements. Even after making sure that the LQR closed-loop bandwidth is lower than the motor dynamics, the regulator still performs.

## 7.2 Sliding Mode Control

Sources: [Khalil 02][Utkin 09]

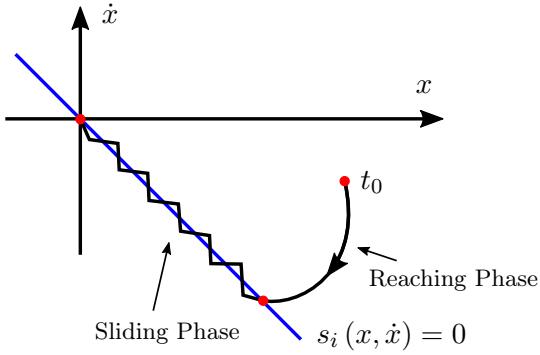
In this section a short introduction to sliding mode control will be given and then further used to design an attitude controller for the ALPHASAT.

SMC can essentially be described as a relay system where control signal drives the system state towards an equilibrium and if it overshoots then the control signal is turned the other way around. In other terms a “BANG BANG” regulator. This control strategy can eventually end up in a state of high frequency switching which is called sliding mode. The low frequency motion which is inherited by the sliding mode is the first order dynamics on the sliding surface.

A SMC can be split up into two phases; the reaching phase and the sliding phase. The trajectories starting with an initial condition where  $|\bar{s}_i(t_0)| > 0$  for  $i = 1, \dots, m$  is drawn towards the sliding surface and is called the reaching phase. The sliding phase is reached in the intersection of  $m$  manifolds where

$$s_i = 0 \quad \forall \quad i = 1, \dots, m \quad (7.24)$$

The motion on the sliding manifolds represents the desired stable dynamics of the system, for which all the trajectories beginning away from the manifold, is driven to by a control law; this is seen in Figure 7.11.



**Figure 7.11:** Shows how the trajectory is drawn towards the sliding surface,  $s_i$  and eventually the equilibrium point.

The control is given as a discontinuous function of the system states,  $u_i$ , which is described as a sign function where the control is given as

$$u_i(x, t) = \text{sign}(s_i) = \begin{cases} u_i^+(x, t) & \text{if } s_i(x) > 0 \\ u_i^-(x, t) & \text{if } s_i(x) < 0 \end{cases}, \quad i = 1, \dots, m \quad (7.25)$$

To reduce the high frequency “chattering” a saturation function can be defined, which essentially is a continuous approximation of a discontinuous control. This saturation function is shown by

$$u_i(x, t) = \text{sat}(s_i) = \begin{cases} \text{sign}(s_i) & \text{if } \|s_i\| > \epsilon \\ \frac{s_i}{\epsilon} & \text{if } \|s_i\| \leq \epsilon \end{cases} \quad (7.26)$$

where  $\epsilon$  defines a saturated area around the sliding surface and dampens the high frequency switching.

### 7.2.1 Design of the Sliding Manifold

First a coordinate transformation of the dynamic and kinematic models of the satellite are performed. This results in an equilibrium point in the origin. Then a sliding manifold is designed on which the order of the system is reduced. In order drive the system to the sliding manifold and contain it there for all future time, a control law is designed.

### 7.2.2 Quaternion Error Calculation

Let  ${}^b\mathbf{q}_e$  denote the relative error between the desired attitude and the current attitude of the satellite, given in the body reference frame. This quaternion error is found through quaternion multiplication as shown:

$${}^b\mathbf{q}_e = {}^b\mathbf{q}_{\text{ref}} \otimes {}^b\mathbf{q}_s^{-1} \quad (7.27)$$

The relative angular velocity error is denoted by  ${}^b\bar{\omega}_e$  and is given as

$${}^b\bar{\omega}_e = {}^b\bar{\omega}_{\text{ref}} - {}^b\underline{\mathbf{R}} {}^c\bar{\omega}_s \quad (7.28)$$

With these change of variables the equilibrium point of the system is now moved to the origin where  ${}^b\mathbf{q}_e = [0 \ 0 \ 0 \ 1]^\top$  and  ${}^b\dot{\mathbf{q}}_e = [0 \ 0 \ 0 \ 0]^\top$ , which is where the angular velocity

of the satellite  ${}^b\bar{\omega}_e = [0 \ 0 \ 0]^\top$ . The system model described in section ?? can then be rewritten to the given form.

$$\begin{bmatrix} {}^b\dot{\bar{q}}_e \\ {}^b\dot{q}_{e4} \\ {}^b\dot{\bar{\omega}}_e \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{T}}({}^b\mathbf{q}_e) {}^b\bar{\omega}_e \\ -\frac{1}{2} {}^b\bar{q}_e^\top {}^b\bar{\omega}_e \\ {}^b\dot{\bar{\omega}}_{\text{ref}} - {}^b\underline{\mathbf{R}} \underline{\mathbf{J}}_s^{-1} \left[ -\underline{\mathbf{S}}({}^c\bar{\omega}_s) {}^c\bar{\mathbf{L}}_s - \underline{\mathbf{S}}({}^c\bar{\omega}_s) {}^c\bar{\mathbf{L}}_{\text{th}} - {}^c\dot{\bar{\mathbf{L}}}_{\text{th}} + {}^c\dot{\bar{\mathbf{L}}}_d \right] \end{bmatrix} \quad (7.29)$$

where  $\underline{\mathbf{T}}(\mathbf{q})$  is a quaternion transformation function which consists of a skew-symmetric cross-product operator from the quaternion derivative with the real values on the diagonal as shown in equation 7.30.

$$\underline{\mathbf{T}}(\mathbf{q}) = -\frac{1}{2} [\underline{\mathbf{S}}(\bar{q}) - q_4 \mathbf{I}_{3 \times 3}] \quad (7.30)$$

The quaternion dynamics  ${}^b\ddot{\bar{q}}_e$  consists of the satellite error dynamics which is shown in equation 7.31.

$$\begin{aligned} {}^b\dot{\bar{\omega}}_e &= {}^b\dot{\bar{\omega}}_{\text{ref}} - {}^b\underline{\mathbf{R}} \underline{\mathbf{J}}_s^{-1} \left[ -\underline{\mathbf{S}}({}^c\bar{\omega}_s) {}^c\bar{\mathbf{L}}_s({}^c\bar{\omega}_s, t) \right. \\ &\quad \left. + {}^c\dot{\bar{\mathbf{L}}}_{\text{th}}({}^b\bar{\omega}_s, \bar{\omega}_m, \bar{u}, t) + {}^c\dot{\bar{\mathbf{L}}}_d(t) \right] \end{aligned} \quad (7.31)$$

Where the external moment component  ${}^c\dot{\bar{\mathbf{L}}}_{\text{th}}({}^b\bar{\omega}_s, \bar{\omega}_m, \bar{u}, t)$  originates from the resulting moments generated by the Tetrahedron reaction wheel configuration and is given as.

$$\begin{bmatrix} {}^c\dot{\bar{\mathbf{L}}}_{\text{th}} \\ {}^{th}\dot{\bar{\mathbf{L}}}_c \\ {}^{th}\dot{\bar{\mathbf{L}}}_{\text{cor}} \end{bmatrix} = \begin{bmatrix} -{}^{th}\underline{\mathbf{R}} \left( {}^{th}\dot{\bar{\mathbf{L}}}_{\text{cor}}({}^b\bar{\omega}_s, \bar{\omega}_m, t) + {}^{th}\dot{\bar{\mathbf{L}}}_c(\bar{\omega}_m, \bar{u}, t) \right) \\ {}^{th}\underline{\mathbf{P}} (-\underline{\Lambda}_m \bar{\omega}_m + \underline{\Gamma}_m \bar{u}_m) \\ \sum_{i=1}^4 -{}^{th}\underline{\mathbf{P}} \underline{\mathbf{S}}({}^{w_i}\bar{\omega}_s + {}^{w_i}\bar{\omega}_{m_i}) \underline{\mathbf{J}}_{w_i} ({}^{w_i}\bar{\omega}_s + {}^{w_i}\bar{\omega}_{m_i}) \end{bmatrix} \quad (7.32)$$

Where  $\bar{u}_m$  is the control input vector where the entries represents the voltage across each motor, controlling the motor torque  ${}^{th}\dot{\bar{\mathbf{L}}}_c$ .

The control objective is then to drive the system states from an initial state  $[{}^b\mathbf{q}_e^\top(0), {}^b\dot{\mathbf{q}}_e^\top(0)]^\top$  to the desired attitude and angular velocity  $[{}^b\mathbf{q}_{\text{ref}}^\top(t), {}^b\dot{\mathbf{q}}_{\text{ref}}^\top(t)]^\top$  within a finite time  $t$ .

### 7.2.3 Sliding Manifold

The sliding manifold is designed as shown in equation 7.33.

$$\bar{s} = {}^b\dot{\bar{q}}_e + \underline{\mathbf{K}} {}^b\bar{q}_e = \bar{0} \quad (7.33)$$

where:

- ${}^b\dot{\bar{q}}_e$  is the quaternion error kinematics
- ${}^b\bar{q}_e$  is the quaternion error
- $\underline{\mathbf{K}}$  is a real, positive definite, diagonal matrix

All motion on the sliding surface is then governed by the first order dynamics written as

$${}^b\dot{\bar{q}}_e = -\underline{\mathbf{K}} {}^b\bar{q}_e \quad (7.34)$$

Further it is shown that the motion on the manifold in equation 7.33 is proven to be stable by the Lyapunov stability criteria.

Stability on the sliding surface is investigated by a chosen quadratic Lyapunov function:

$$V_s(b\bar{q}_e) = \frac{1}{2} b\bar{q}_e^\top b\bar{q}_e > 0 \quad \text{in } D - \{\mathbf{0}\} , \quad \left\{ b\bar{q}_e \in D : \|b\bar{q}_e\| \leq 1 \right\} \quad (7.35)$$

where  $D \subset \mathbb{R}^3$ ,  $V_s(b\bar{q}_e)$  is positive definite and  $V_s(\mathbf{0}) = 0$  is the origin of the Lyapunov function. The derivative of the Lyapunov function is then

$$\begin{aligned} \dot{V}_s(b\bar{q}_e) &= \frac{1}{2} b\dot{\bar{q}}_e^\top b\bar{q}_e + \frac{1}{2} b\bar{q}_e^\top b\dot{\bar{q}}_e \\ &= b\bar{q}_e^\top b\dot{\bar{q}}_e \leq 0 \end{aligned} \quad (7.36)$$

The origin is proven to be stable if the inequality  $\dot{V} \leq 0$  holds for the entire domain  $D$ . By inserting the dynamics of the sliding surface equation 7.34 into equation 7.36 yields

$$\dot{V}_s(b\bar{q}_e) = -b\bar{q}_e^\top \underline{\mathbf{K}} b\bar{q}_e < 0 \quad \text{in } D - \{\mathbf{0}\} \quad (7.37)$$

which is negative definite which means that all motion on the sliding surface is asymptotically stable.

#### 7.2.4 The Reaching Phase

For now a sliding surface is designed, upon which, all motion is bounded by a reduced order dynamics. Next step is to design a control law which drives all the trajectories towards the sliding manifold.

In order to investigate if all the trajectories starting off the sliding surface where  $|\bar{s}(0)| > \mathbf{0}$  goes towards  $\bar{s}(t) = 0$  when  $t \rightarrow \infty$ , a Lyapunov function is constructed.

The Lyapunov function represents a quadratic function of the sliding surface which is positive semi-definite.

$$V(\bar{s}) = \frac{1}{2} \bar{s}^\top \bar{s} > 0 \quad \text{in } D - \{\mathbf{0}\} , \quad V(0) = 0; \quad (7.38)$$

which indeed is positive definite and only zero at the origin. In order to prove stability the derivative of the Lyapunov function is investigated.

$$\dot{V}(\bar{s}) = \frac{1}{2} \bar{s}^\top \dot{\bar{s}} + \frac{1}{2} \dot{\bar{s}}^\top \bar{s} = \bar{s}^\top \dot{\bar{s}} \quad (7.39)$$

where  $\dot{\bar{s}}$  is given as

$$\dot{\bar{s}} = b\ddot{\bar{q}}_e + \underline{\mathbf{K}} b\dot{\bar{q}}_e \quad (7.40)$$

by inserting the derivative of the sliding surface  $\dot{\bar{s}}$  into equation 7.39.

$$\begin{aligned} \dot{V}(\bar{s}) &= \bar{s}^\top \dot{\bar{s}} = \bar{s}^\top \left[ b\ddot{\bar{q}}_e + \underline{\mathbf{K}} b\dot{\bar{q}}_e \right] \\ &= \bar{s}^\top \left[ \underline{\mathbf{T}}(b\mathbf{q}_e)\dot{\bar{\omega}}_e - \frac{1}{4} b\bar{q}_e \|\bar{\omega}_e\|_2^2 + \underline{\mathbf{K}} b\dot{\bar{q}}_e \right] \end{aligned} \quad (7.41)$$

By inserting the error dynamics  $\dot{\bar{\omega}}_e$  from equation 7.31 into 7.41 the control  $\bar{u}_m$  appears.

$$\begin{aligned} \dot{V}(\bar{s}) &= \bar{s}^\top \left[ \underline{\mathbf{T}}(b\mathbf{q}_e)\dot{\bar{\omega}}_{ref} - \underline{\mathbf{T}}(b\mathbf{q}_e) \underline{\mathbf{R}}_s^{-1} \left( -\underline{\mathbf{S}}(c\bar{\omega}_s) {}^c\bar{\mathbf{L}}_s - {}^{th}\underline{\mathbf{R}} \dot{{}^{th}\bar{\mathbf{L}}}_{cor} \right. \right. \\ &\quad \left. \left. + {}^c\underline{\mathbf{R}} {}^{th}\underline{\mathbf{P}} \underline{\Lambda}_m \bar{\omega}_m - {}^{th}\underline{\mathbf{R}} {}^{th}\underline{\mathbf{P}} \underline{\Gamma}_m \bar{u}_m + {}^c\dot{\bar{\mathbf{L}}}_d(t) \right) \right. \\ &\quad \left. - \frac{1}{4} b\bar{q}_e \|b\bar{\omega}_e\|_2^2 + \underline{\mathbf{K}} b\dot{\bar{q}}_e \right] \end{aligned} \quad (7.42)$$

Equation 7.42 is simplified by rewriting the affine system to the regular form given by:

$$\begin{aligned}\dot{\bar{x}}_1 &= \bar{f}_1(\bar{x}_1, \bar{x}_2, t) \\ \dot{\bar{x}}_2 &= \bar{f}_2(\bar{x}_1, \bar{x}_2, t) + \underline{\mathbf{B}}_2(x_1, x_2) \bar{u}\end{aligned}\quad (7.43)$$

where the first term  $\bar{f}$  dose not dependent on the control input and second term  $\underline{\mathbf{B}}_2$  is dependant on the control  $\bar{u}$ . The new isolated form is given as:

$${}^b\ddot{\bar{q}}_e = \bar{f}_2 \left( {}^b\mathbf{q}_e, {}^b\bar{\omega}_s, {}^b\bar{\omega}_e, \bar{\omega}_m, t \right) + \underline{\mathbf{B}}_2 \left( {}^b\mathbf{q}_e, t \right) \bar{u}_m \quad (7.44)$$

where  $\bar{f}_b$  represents the affine complex part of the quaternion acceleration which is independent on the control and is given as

$$\begin{aligned}\bar{f}_2 \left( {}^b\mathbf{q}_e, {}^b\bar{\omega}_s, {}^b\bar{\omega}_e, \bar{\omega}_m, t \right) &= \underline{\mathbf{T}}({}^b\mathbf{q}_e) \left[ \dot{\bar{\omega}}_{ref} - {}^c\underline{\mathbf{R}} \underline{\mathbf{J}}_s^{-1} \left( - \underline{\mathbf{S}}({}^c\bar{\omega}_s) {}^c\bar{\mathbf{L}}_s \right. \right. \\ &\quad \left. \left. - {}^{th}\underline{\mathbf{R}} {}^{th}\dot{\bar{\mathbf{L}}}_{cor} + {}^{th}\underline{\mathbf{R}} {}^{th}\underline{\mathbf{P}} \underline{\mathbf{\Lambda}}_m \bar{\omega}_m \right) \right] - \frac{1}{4} {}^b\bar{q}_e \left\| {}^b\bar{\omega}_e \right\|_2^2\end{aligned}\quad (7.45)$$

The matrix  $\underline{\mathbf{B}}_2$  is a  $3 \times 4$  matrix which is inherited from the over-actuated system with four reaction wheels. The moments generated by each of the four reaction wheels is projected from  $\mathbb{R}^4 \rightarrow \mathbb{R}^3$  by the projection matrix  ${}^{th}\underline{\mathbf{P}}$ . The matrix  $\underline{\mathbf{B}}_2$  is given as.

$$\underline{\mathbf{B}}_2 \left( {}^b\mathbf{q}_e, t \right) = \underline{\mathbf{T}}({}^b\mathbf{q}_e) {}^b\underline{\mathbf{R}} \underline{\mathbf{J}}_s^{-1} {}^{th}\underline{\mathbf{R}} {}^{th}\underline{\mathbf{P}} \underline{\mathbf{\Gamma}}_m \quad (7.46)$$

As described in Section 5.4.1 the projection matrix  ${}^{th}\underline{\mathbf{P}}$  is a  $3 \times 4$  matrix which is non-invertible but has a pseudoinverse which can be described as a minimum-norm solution to distribute the desired moment evenly amongst the reaction wheels. The pseudoinverse is then written as

$$\underline{\mathbf{B}}_2^+ = \underline{\mathbf{\Gamma}}_m^{-1} {}^{th}\underline{\mathbf{P}}^+ {}^{th}\underline{\mathbf{R}}^{-1} \underline{\mathbf{J}}_s {}^b\underline{\mathbf{R}}^{-1} \underline{\mathbf{T}}({}^b\mathbf{q}_e)^\top \quad (7.47)$$

where it is known that the quaternion derivative operator  $\underline{\mathbf{T}}({}^b\mathbf{q}_e)$  is unitary which means that  $\underline{\mathbf{T}}^* = \underline{\mathbf{T}}^{-1}$ . By replacing the new form into the derivative of the Lyapunov function the equation is simplified to

$$\dot{V}(\bar{s}) = \bar{s}^\top \left[ \bar{f}_2 + \underline{\mathbf{B}}_2 \bar{u}_m - \underline{\mathbf{T}}({}^b\mathbf{q}_e) {}^b\underline{\mathbf{R}} \underline{\mathbf{J}}_s^{-1} {}^c\dot{\bar{\mathbf{L}}}_d(t) + \underline{\mathbf{K}} {}^b\dot{\bar{q}}_e \right] \quad (7.48)$$

In order to stabilise the system and bring  $\bar{s} = 0$  which follows that  $\dot{\bar{s}} = 0$ , the following control is chosen.

$$\bar{u}_m = -\underline{\mathbf{B}}_2^+ \left( \bar{f}_2 + \underline{\mathbf{K}} {}^b\dot{\bar{q}}_e + g_0 \frac{\bar{s}}{\|\bar{s}\|} \right) \quad (7.49)$$

where  $g_0$  is a positive scalar and  $\bar{s}/\|\bar{s}\|$  is the switching term which ensures that all the trajectories is drawn towards the sliding surface. By insertion the control given in equation 7.49 into equation 7.48 results in the following

$$\begin{aligned}\dot{V}(\bar{s}) &= -\bar{s}^\top \left[ \underline{\mathbf{T}}({}^b\mathbf{q}_e) {}^b\underline{\mathbf{R}} \underline{\mathbf{J}}_s^{-1} {}^c\dot{\bar{\mathbf{L}}}_d(t) + g_0 \frac{\bar{s}}{\|\bar{s}\|} \right] \\ &\leq -g_0 \|\bar{s}\| + \|\bar{s}\| \frac{1}{2} \frac{D^+}{M^-} \leq 0\end{aligned}\quad (7.50)$$

$\dot{V}(\bar{s})$  is bounded by the upper bound of the disturbance term in equation 7.48 which is denoted by  $D^+$ . The upper bound of  $J_s^{-1}$  is given as

$$J_s^{-1} \leq \frac{1}{M^-} \quad (7.51)$$

where  $M^- > 0$  and represents the smallest MoI of the satellite. The knowledge of the worst case disturbance, a lower bound for the switching gain  $g_0$  is derived.

$$g_0 \geq \frac{1}{2} \frac{D^+}{M^-} + \sigma_0 , \quad \sigma_0 > 0 \quad (7.52)$$

if  $g_0$  is inserted into 7.50 the following inequality appears.

$$\dot{V}(\bar{s}) \leq -\sigma_0 \|\bar{s}\| \quad (7.53)$$

where this inequality ensures that all trajectories starting off the sliding manifold where  $\bar{s} = \bar{\mathbf{0}}$  will reach it within a finite time depending on the size of  $\sigma_0$ .

In this particular case, the size of the switching gain  $g_0$  is only dependant on the magnitude of the disturbance and  $\sigma_0$ . This is only valid if the dynamic model is without uncertainties. In models with large uncertainties the control would again consist of a continuous part dependant on the nominal model of the satellite and a switching part, which magnitude would depend on the upper-bounds of the uncertainty from the nominal model. These uncertainties could result in large switching gains which would introduce “chattering” when crossing the sliding surface as illustrated in Figure 7.11. The chattering is reduced by applying a saturation function as described in equation 7.26.

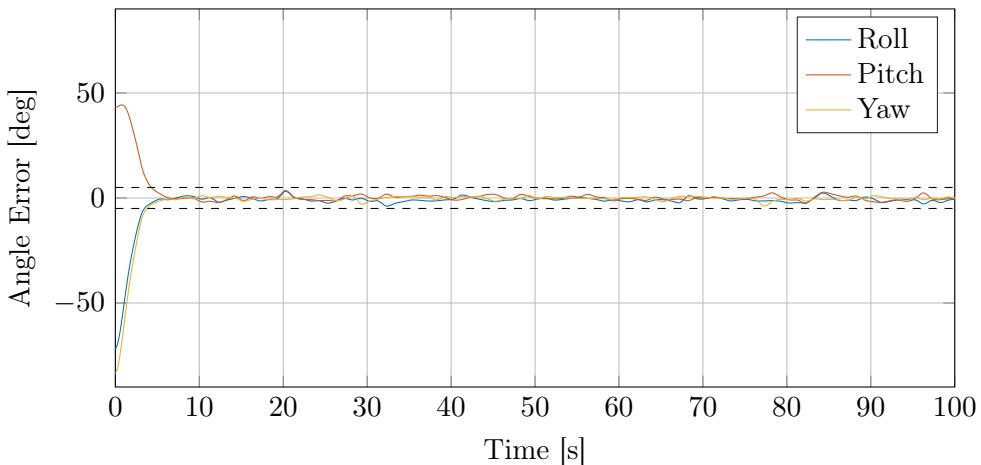
Now that the sliding mode attitude controller is designed it should be verified through the simulation environment.

### 7.2.5 Results

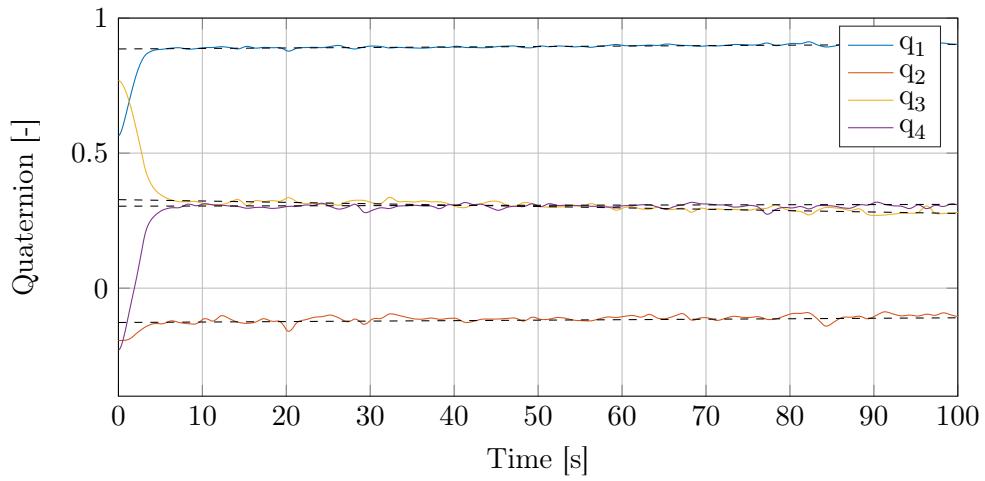
The SMC is in general verified with the same simulation configuration as the linear regulator, in order to perform a genuine performance comparison.

#### Nadir Pointing

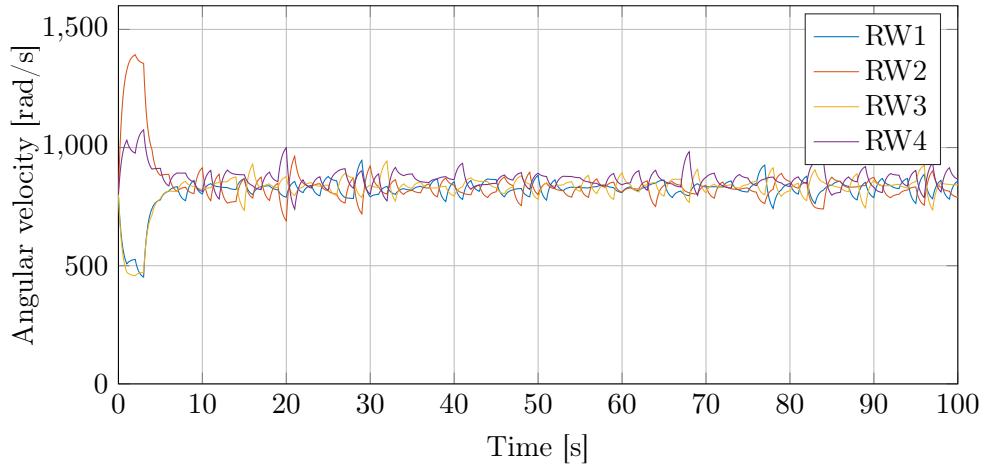
The satellite is first initialised in an arbitrary orientation with an initial spin, same as the linear attitude controller simulation. The controller should then bring align the satellite attitude with the ORF in order to point Nadir.



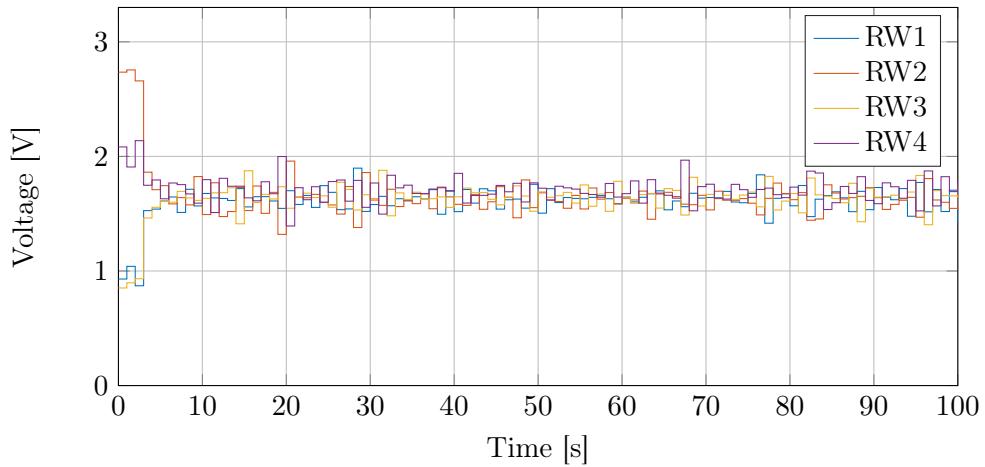
**Figure 7.12:** Shows the satellite attitude error in Euler angles, where the dashed lines represent the maximum and minimum angle error.



**Figure 7.13:** Shows the satellite attitude  ${}^b\mathbf{q}_s$  relative to the ECI, where the dashed lines represent the quaternion attitude reference  ${}^b\mathbf{q}_{\text{ref}}$ .



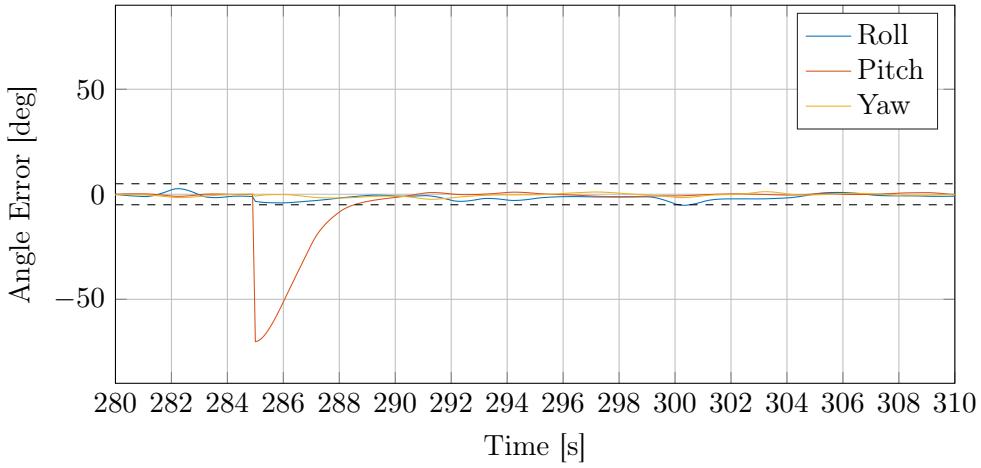
**Figure 7.14:** Shows angular velocities on each motor in the Tetrahedron configuration.



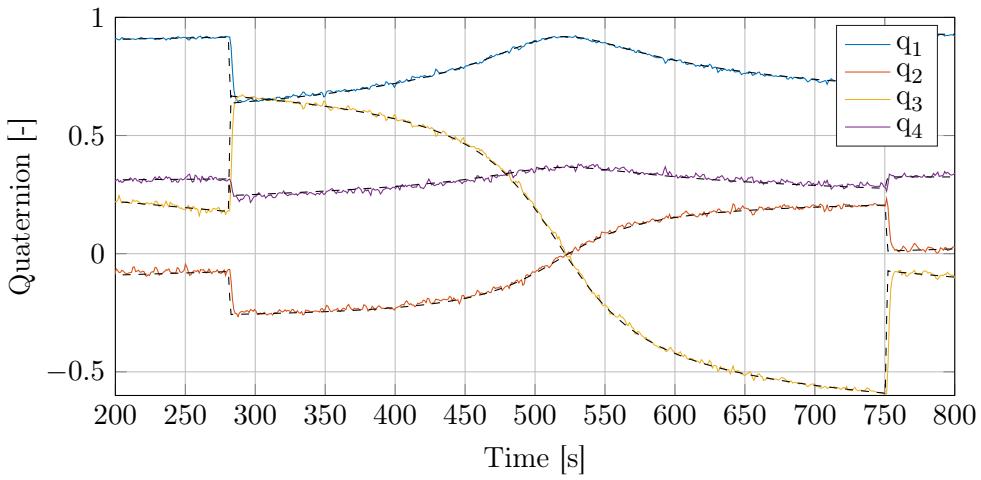
**Figure 7.15:** Shows the motor control signal.

From Figure 7.12 it is seen that the angular error in Roll, Pitch and Yaw is compensated for within  $\approx 4$  sec. which is well within the requirements.

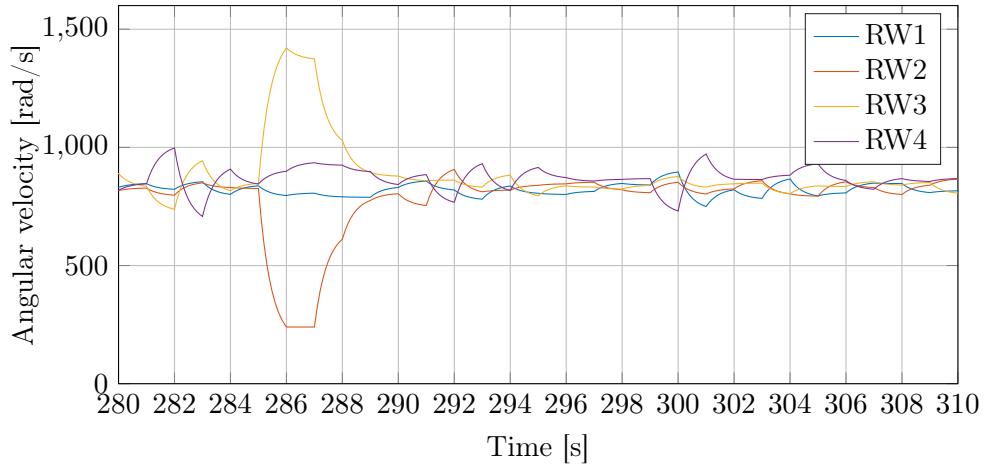
### Target Tracking



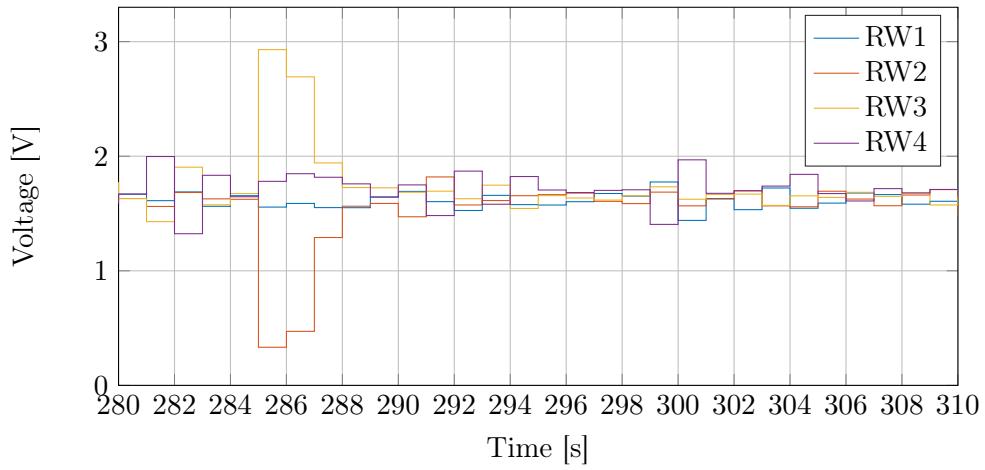
**Figure 7.16:** Shows the satellite attitude error in Euler angels.



**Figure 7.17:** Shows the satellite attitude  ${}^b\mathbf{q}_s$  relative to the ECI, where the dashed lines represent the quaternion attitude reference  ${}^b\mathbf{q}_{ref}$ .



**Figure 7.18:** Shows angular velocities on each motor in the Tetrahedron configuration.



**Figure 7.19:** Shows the motor control signal.

### 7.2.6 Summary

The sliding mode control algorithm is a quaternion based sliding mode, which takes the coriolis torques from the reaction wheels into account. Through the simulations it can be seen that the sliding mode attitude control algorithm performs well within the requirements.



---

## Attitude Control Testbed

Attitude Determination and Control Systems for CubeSat satellites are inherently difficult to test and validate. This is because of both the relatively small actuation torques available and the environment present at the Earth's surface level, compared to the environment at LEO altitudes. When designing attitude control algorithms for a spacecraft as small as a CubeSat, the physical size of the spacecraft impose innate constraints on the dimensioning of the actuators and thereby inherently also on the actuator torques available. The small actuator torques available, hence presents a challenge when the control algorithms is to be validated in a live implementation.

To accommodate for these challenges, a testbed is designed and built in order to permit testing capabilities of the attitude control algorithms designed for ALPHASAT and other future projects at AAU.

A testbed can be a valuable tool, both in the design processes of the next student built satellites at AAU but also for use in education and semester projects.

AAU Student Space has, in the time of writing, launched five student built CubeSat satellites. The satellites, AAUSAT3 and 4 both used magnetic actuation for attitude control. During LEOPs of both satellites, the detumbling algorithm where initiated which eventually caused the satellites to accelerate instead of decelerate. A wrong sign in the detumbling algorithm caused this unfortunate error.

The test phase of both satellites might have benefited by having an attitude control testbed. This could have provided the ability to perform, not only, active pointing testing, but also the ability to perform basic sanity checks of the algorithms. These are two examples of cases without severe consequences, as this sign where simply changed when the fault where registered. However if this sign was not easily changed this could have led to the loss of the satellites.

Another example, is the rotations performed within an attitude control algorithm. The ability to perform sanity checks of these can be beneficial as an addition to simulations during the design and test phase, of a satellite mission.

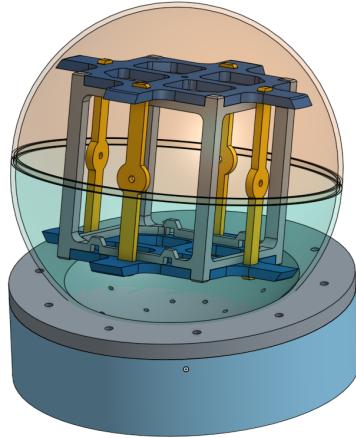
The attitude control testbed designed throughout, the following chapter, aims to provide a method of testing the general properties and ability to perform sanity checks on control algorithms. In addition to this, an attitude control testbed provides an educational tool for implementing three-dimensional control algorithms.

The testbed does however not provide the ability to test and validate the exact controllers designed for a spacecraft, this would require a mechanical design allowing an actual satellite to be inserted into the testbed. A typical satellite consists of rather

expensive hardware, which may be highly sensitive to Electro Static Discharge (ESD), furthermore; in order to counteract the effect, the Earth's gravitational pull has on the satellites CoM, the testbed should have the ability to adjust the CoM. Instead, the testbed is designed as a piece of HIL equipment. This approach results in a testbed where the exact controller, with coefficients designed for space, not necessarily performs well for the testbed, but then the testbed serves as a method of validating algorithms instead of controller coefficients.

## 8.1 System Design

The physical part of the testbed system consists of a foot with an air bearing built into the spherical cap holding an acrylic sphere. The sphere is containing the electronics e.g. the actuators, sensors, computer and a wireless communication interface. When the testbed is in use the sphere with electronics will float on a thin film of pressurised air coming from the foot of the testbed. A Computer-Aided Design (CAD) model can be seen on Figure 8.1.



**Figure 8.1:** The testbed with the air-bearing.

Apart from the physical part, the testbed consists of a software part. The testbed is designed to be used in conjunction with an external software suite. This could be MATLAB/Simulink or another scripting tool or software capable of running the attitude estimator and control algorithms. Initially the software will be designed to use MATLAB/Simulink with the testbed as a HIL component. This will make it possible to watch changes in the attitude while plotting and running simulations of control algorithms.

## 8.2 Attitude Determination

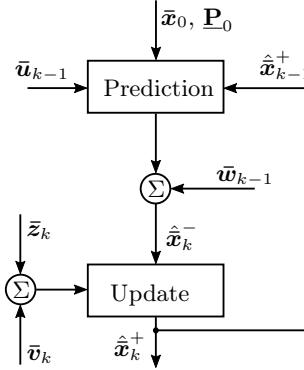
The attitude determination system is serving as a feedback and thereby closing the control loop round the controller and the physical system. This can be performed via a number of methods. Both deterministic and stochastic methods exists however for the testbed, a stochastic methods will be used. A widespread method of performing stochastic estimation or sensor fusion is the Kalman Filter (KF) [Grewal 08]. In the following - the linear KF is summarised and compared with the EKF. After this an EKF is developed and simulated for use as an attitude determination system for the testbed.

### 8.2.1 The Kalman Filter

Sources: [Grewal 08] [Wertz 95]

The KF is a two-step recursive stochastic filter. The KF uses knowledge of the current state and sensors measurements to estimate the true state. The fusion of multiple sensors and the current state results in a better estimate than using a single sensor to measure. This is done through a prediction step and an update step.

The general flow of the Kalman filter algorithm can be observed in Figure 8.2.



**Figure 8.2:** Shows the general flow of the Kalman filter.

The “prediction step” uses the previous control input  $\bar{u}_{k-1}$  and the previous a posteriori state estimate  $\hat{x}_{k-1}^+$ , note that an a posteriori estimate is denoted with a (+) and an estimate is denoted with a (^). In the first run of the filter, it is initialised with an initial state vector  $\bar{x}_0$  and an initial state error covariance matrix  $P_0$  instead of previous estimates. The prediction produces an a priori state estimate  $\hat{x}_k^-$ , note that an a priori estimate is denoted with a (-).

In the “update step” the a priori state estimate is then updated using sensor measurements  $\bar{z}_k$ . The result is a current a posteriori state vector estimate  $\hat{x}_k^+$

### 8.2.2 The Extended Kalman Filter

An Extended Kalman filter is a non-linear approach to Kalman filtering. In the traditional Kalman filter the model is linearised in a point of operation. In the non-linear version, that is the EKF; a non-linear model is used for prediction and the system is linearised about the point of the state space where the system currently resides. This linearisation is performed using the Jacobian matrix for the state transition matrix and the output matrix.

The EKF uses an affine model as shown in Equation (8.1).

$$\begin{aligned}\bar{x}_k &= f(\bar{x}_{k-1}, \bar{u}_{k-1}) + \bar{w}_{k-1} \\ \bar{z}_k &= h(\bar{x}_k) + \bar{v}_k\end{aligned}\tag{8.1}$$

Where:

$\bar{\mathbf{x}}$	is the system state vector	[.]
$\bar{\mathbf{u}}$	is the control input	[.]
$\bar{\mathbf{w}}$	is the process noise	[.]
$\bar{\mathbf{z}}$	is the measurement vector	[.]
$\bar{\mathbf{v}}$	is the measurement noise	[.]

In the prediction step this model is used to propagate the previous a posteriori state estimate  $\hat{\mathbf{x}}_{k-1}^+$  and the previous actuator input  $\bar{\mathbf{u}}_{k-1}$ , resulting in an a priori state estimate  $\hat{\mathbf{x}}_k^-$ .

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}^+, \bar{\mathbf{u}}_{k-1}) + \bar{\mathbf{w}}_{k-1} \quad (8.2)$$

Where:

$\hat{\mathbf{x}}_k^-$	is the a priori state estimate	[.]
$\hat{\mathbf{x}}_{k-1}^+$	is the previous a posteriori state estimate	[.]

The process noise  $\bar{\mathbf{w}}_{k-1}$  is assumed white Gaussian noise with mean 0 and covariance matrix  $\underline{\mathbf{Q}}$ .

A state transition matrix  $\underline{\Phi}_k$  is then calculated by the Jacobian matrix. The  $\underline{\Phi}_k$  matrix and the process noise covariance matrix  $\underline{\mathbf{Q}}$  is used to propagate the previous a posteriori state vector error covariance matrix.

$$\begin{aligned} \underline{\Phi}_k &= \frac{\partial f(\bar{\mathbf{x}}, \bar{\mathbf{u}}_{k-1})}{\partial \bar{\mathbf{x}}} \Big|_{\bar{\mathbf{x}}=\hat{\mathbf{x}}_k^-} \\ \underline{\mathbf{P}}_k^- &= \underline{\Phi}_k \underline{\mathbf{P}}_{k-1}^+ \underline{\Phi}_k^\top + \underline{\mathbf{Q}} \end{aligned} \quad (8.3)$$

Where:

$\underline{\mathbf{P}}_k^-$	is the a priori state error covariance matrix	[.]
$\underline{\mathbf{P}}_{k-1}^+$	is the previous a posteriori state error covariance matrix	[.]

In the update step; the a priori state estimate is updated via a sensor measurement. The a priori state vector estimate is mapped into an a priori measurement vector estimate  $\hat{\mathbf{z}}_k^-$  using a sensor model.

$$\hat{\mathbf{z}}_k^- = h(\hat{\mathbf{x}}_k^-) + \bar{\mathbf{v}}_k \quad (8.4)$$

Where:

$\hat{\mathbf{z}}_k^-$	is the a priori measurement vector estimate	[.]
$\bar{\mathbf{v}}_k$	is the measurement noise	[.]

The measurement noise is assumed white Gaussian noise with mean 0 and covariance matrix  $\underline{\mathbf{R}}$ .

The sensor model is linearised about the a priori state vector estimate using a Jacobian and a Kalman gain matrix is calculated using the a priori state error covariance matrix, the measurement matrix and the measurement noise covariance matrix.

$$\begin{aligned}\underline{\mathbf{H}}_k &= \frac{\partial h(\bar{\mathbf{x}})}{\partial \bar{\mathbf{x}}} \Big|_{\bar{\mathbf{x}}=\hat{\mathbf{x}}^-_k} \\ \underline{\mathbf{K}}_k &= \underline{\mathbf{P}}_k^- \underline{\mathbf{H}}_k^\top \left( \underline{\mathbf{H}}_k \underline{\mathbf{P}}_k^- \underline{\mathbf{H}}_k^\top + \underline{\mathbf{R}} \right)^{-1}\end{aligned}\quad (8.5)$$

A sensor measurement is performed and the Kalman gain is used to update the a priori state vector estimate with the measurement prediction error resulting in an a posteriori state vector estimate

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \underline{\mathbf{K}}_k \left( \bar{z}_k - \hat{z}_k^- \right) \quad (8.6)$$

Where:

$$\bar{z}_k \quad \text{is the sensor measurements} \quad [\cdot]$$

The last step is to update the a priori state vector covariance matrix into a a posteriori state vector covariance matrix. This is done by utilising the Kalman gain matrix and the measurement matrix.

$$\underline{\mathbf{P}}_k^+ = (\mathbf{I} - \underline{\mathbf{K}}_k \underline{\mathbf{H}}_k) \underline{\mathbf{P}}_k^- \quad (8.7)$$

A comparison between the Kalman filter and the Extended Kalman filter algorithms are shown below. The main difference between the traditional Kalman filter and the EKF is that an extended Kalman filter uses a non-linear prediction model.

---

### Prediction:

KF	EKF
$\hat{\mathbf{x}}_k^- = \underline{\Phi} \hat{\mathbf{x}}_{k-1}^+ + \underline{\Gamma} \bar{\mathbf{u}}_{k-1} + \bar{\mathbf{w}}_{k-1}$	$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}^+, \bar{\mathbf{u}}_{k-1}) + \bar{\mathbf{w}}_{k-1}$
$\hat{\mathbf{z}}_k^- = \underline{\mathbf{H}} \hat{\mathbf{x}}_k^- + \bar{\mathbf{v}}_k$	$\hat{\mathbf{z}}_k^- = h(\hat{\mathbf{x}}_k^-) + \bar{\mathbf{v}}_k$
$\underline{\mathbf{P}}_k^- = \underline{\Phi} \underline{\mathbf{P}}_{k-1}^+ \underline{\Phi}^\top + \underline{\mathbf{Q}}_{k-1}$	$\underline{\Phi}_k = \frac{\partial f(\bar{\mathbf{x}}, \bar{\mathbf{u}}_{k-1})}{\partial \bar{\mathbf{x}}} \Big _{\bar{\mathbf{x}}=\hat{\mathbf{x}}_k^-} \quad (8.8)$
	$\underline{\mathbf{H}}_k = \frac{\partial h(\bar{\mathbf{x}})}{\partial \bar{\mathbf{x}}} \Big _{\bar{\mathbf{x}}=\hat{\mathbf{x}}_k^-}$
	$\underline{\mathbf{P}}_k^- = \underline{\Phi}_k \underline{\mathbf{P}}_{k-1}^+ \underline{\Phi}_k^\top + \underline{\mathbf{Q}}_{k-1}$

### Update:

$\underline{\mathbf{K}}_k = \underline{\mathbf{P}}_k^- \underline{\mathbf{H}}_k^\top \left( \underline{\mathbf{H}}_k \underline{\mathbf{P}}_k^- \underline{\mathbf{H}}_k^\top + \underline{\mathbf{R}}_k \right)^{-1}$	$\underline{\mathbf{K}}_k = \underline{\mathbf{P}}_k^- \underline{\mathbf{H}}_k^\top \left( \underline{\mathbf{H}}_k \underline{\mathbf{P}}_k^- \underline{\mathbf{H}}_k^\top + \underline{\mathbf{R}}_k \right)^{-1}$
$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \underline{\mathbf{K}}_k \left( z_k - \hat{z}_k^- \right)$	
$\underline{\mathbf{P}}_k^+ = (\mathbf{I} - \underline{\mathbf{K}}_k \underline{\mathbf{H}}_k) \underline{\mathbf{P}}_k^-$	

---

The method of using non-linear equations where possible and then linearise the system at the current state where necessary, does however have a constraint. The constraint is that it is only possible for *sufficiently smooth* systems [Grewal 08].

If a system with some irregular dynamics and kinematics, is in a point in its state space where the rate of change in the function is to great compared to the rate that the EKF is running; then the estimate cannot be trusted, because the system can differ to much from its linearisation point. Other methods of estimation must then be considered.

This is however not the case for a satellite and hence not the attitude control testbed either.

### 8.2.3 Quaternions and the Extended Kalman Filter

Sources: [Markley 14] [Bak 99] [Markley 03] [Humphreys 02] [Stuelpnagel 64]

A parameter set of four, is the smallest set for which an attitude in a three-dimensional space can be represented singularity free [Stuelpnagel 64]. The quaternion is such parameter set; the only constraint on a quaternion as an attitude representation is a unit norm constraint. A quaternion is not closed under addition and scalar multiplication, which implies that the quaternion does not span a vector space. The quaternion is a sphere in a four-dimensional space. When observing the update equations in Equation (8.9) it becomes apparent that a consequence of not spanning a vector space is that the resulting quaternion can not, with certainty, be a quaternion of unit length, and hence not the an orthogonality preserving rotation.

Furthermore according to [Bak 99] and [Markley 14]; using a four-dimensional attitude representation for something which is inherently three-dimensional, can lead to a singular state error covariance matrix.

One way to address these challenges is to modify the Kalman filter, such that the quaternion part is updated multiplicatively and thereby adhering to the quaternion algebra. This results in a Multiplicative Extended Kalman Filter (MEKF). The idea of the MEKF is to use a global attitude representation and perturb it with a local “error” representation. This means using an error state vector inside the MEKF. Where the quaternion part of the state vector contains only the vector part of the quaternion; thereby making the covariance propagation possible without singularity problems.

The prediction step is:

$$\begin{aligned}\hat{\bar{x}}_k^- &= f(\hat{\bar{x}}_{k-1}^+, \bar{u}_{k-1}) + \bar{w}_{k-1} \\ \hat{\bar{z}}_k^- &= h(\hat{\bar{x}}_k^-) + \bar{v}_k\end{aligned}\tag{8.9}$$

Next step is to calculate the  $\delta\hat{\bar{x}}_k^+$  which is the vector containing the perturbations for the global state.

$$\delta\hat{\bar{x}}_k^+ = \underline{\mathbf{K}}_k \left[ \bar{z}_k - \hat{\bar{z}}_k^- \right]\tag{8.10}$$

Where:

$$\delta\hat{\bar{x}}_k^+ = \begin{bmatrix} \delta\hat{\bar{q}}_k^+ \\ \delta\hat{\bar{\xi}}_k^+ \end{bmatrix}\tag{8.11}$$

Where  $\hat{\bar{\xi}}_k^-$  collects all the states which can be updated the traditional linear way. Note that the quaternion part ( $\hat{\bar{q}}_k^-$ ) only consists of the imaginary vector part of the quaternion.

The quaternion part of  $\delta\hat{\bar{x}}_k^+$  consists of only the vector part of the quaternion. In order to enforce the quaternion norm, and thereby ensuring orthogonality, the real part of the  $\delta\hat{\bar{q}}_k^+$  is calculated as:

$$\delta\hat{q}_{4,k}^+ = \sqrt{1 - \left\| \delta\hat{\bar{q}}_k^+ \right\|_2^2} \quad (8.12)$$

The  $\delta\hat{\bar{q}}_k^+$  is augmented with this and the attitude quaternion updated:

$$\hat{\mathbf{q}}_k^+ = \begin{bmatrix} \delta\hat{\bar{q}}_k^+ \\ \delta\hat{q}_{4,k}^+ \end{bmatrix} \otimes \hat{\mathbf{q}}_k^- \quad (8.13)$$

The  $\bar{\xi}$  part of the state vector is updated, as with traditional EKFs.

$$\hat{\xi}_k^+ = \hat{\xi}_k^- + \delta\hat{\xi}_k^+ \quad (8.14)$$

With this method the covariance singularity problem is addressed and the quaternion norm is enforced, making this algorithm a good choice for attitude determination.

A summary of the complete algorithm is shown in equations (8.15) and (8.16).

#### Prediction Step:

$$\begin{aligned} \hat{\bar{x}}_k^- &= f(\hat{\bar{x}}_{k-1}^+, \bar{u}_{k-1}) + \bar{w}_{k-1} \\ \hat{\bar{z}}_k^- &= h(\hat{\bar{x}}_k^-) + \bar{v}_k \\ \underline{\Phi}_k &= \frac{\partial f(\bar{x}, u_{k-1})}{\partial \bar{x}} \Big|_{\bar{x}=\hat{\bar{x}}_k^-} \\ \underline{\mathbf{H}}_k &= \frac{\partial h(\bar{x})}{\partial \bar{x}} \Big|_{\bar{x}=\hat{\bar{x}}_k^-} \\ \underline{\mathbf{P}}_k^- &= \underline{\Phi}_k \underline{\mathbf{P}}_{k-1}^+ \underline{\Phi}_k^\top + \underline{\mathbf{Q}}_{k-1} \end{aligned} \quad (8.15)$$

#### Update Step:

$$\begin{aligned} \underline{\mathbf{K}}_k &= \underline{\mathbf{P}}_k^- \underline{\mathbf{H}}_k^\top \left( \underline{\mathbf{H}}_k \underline{\mathbf{P}}_k^- \underline{\mathbf{H}}_k^\top + \underline{\mathbf{R}}_k \right)^{-1} \\ \delta\hat{\bar{x}}_k^+ &= \underline{\mathbf{K}}_k \left[ \bar{z}_k - \hat{\bar{z}}_k^- \right] \\ \hat{\mathbf{q}}_k^+ &= \begin{bmatrix} \delta\hat{\bar{q}}_k^+ \\ \sqrt{1 - \left\| \delta\hat{\bar{q}}_k^+ \right\|_2^2} \end{bmatrix} \otimes \hat{\mathbf{q}}_k^- \\ \hat{\xi}_k^+ &= \hat{\xi}_{k-1}^+ + \delta\hat{\xi}_k^+ \\ \hat{\bar{x}}_k^+ &= \begin{bmatrix} \hat{\mathbf{q}}_k^+ \\ \hat{\xi}_k^+ \end{bmatrix} \\ \underline{\mathbf{P}}_k^+ &= (\underline{\mathbf{I}} - \underline{\mathbf{K}}_k \underline{\mathbf{H}}_k) \underline{\mathbf{P}}_k^- \end{aligned} \quad (8.16)$$

This concludes the description of this slightly modified EKF, the MEKF. The MEKF is a Kalman filter which takes the quaternion in the state vector into account in

the update step. Taking the quaternion of the state vector into account has the advantage of adhering to the quaternion norm constraint in the update step, thereby minimising the numerical rounding errors of performing a brute-force normalisation after an additive update step; furthermore it has the advantage of avoiding the risk of the state error covariance matrix becoming ill-conditioned or singular described by [Bak 99] and [Markley 14].

#### 8.2.4 Implementation and Tuning

The EKF described in the previous section is implemented in Matlab. This is done to facilitate the HIL approach of the testbed and have both the control and attitude determination algorithms running in Matlab/Simulink.

The states that is to be estimated in the EKF is:

$$\bar{\mathbf{x}} = \begin{bmatrix} {}^b\mathbf{q}_s \\ {}^b\bar{\boldsymbol{\omega}}_s \\ \bar{\boldsymbol{\omega}}_{rw} \end{bmatrix} \quad (8.17)$$

The attitude quaternion  ${}^b\mathbf{q}_s$  describes the testbed hardware's attitude relative to the GRF. For a spacecraft in orbit this would be the satellites attitude relative to the ECI instead. Likewise  ${}^b\bar{\boldsymbol{\omega}}_s$  is the angular velocity of the testbed hardware relative to GRF. The  $\bar{\boldsymbol{\omega}}_{rw}$  is the angular velocities of the four reaction wheels.

The models used for estimating these 11 states are developed in Chapter 5 and is summarised in equation (8.18).

$$\begin{bmatrix} {}^c\dot{\mathbf{q}}_s \\ {}^c\dot{\boldsymbol{\omega}}_s \\ \dot{\boldsymbol{\omega}}_{rw} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\underline{\boldsymbol{\Omega}}(\bar{\boldsymbol{\omega}})\mathbf{q} \\ \underline{\mathbf{J}}_s^{-1} \left[ -\underline{\mathbf{S}}({}^c\bar{\boldsymbol{\omega}}_s) {}^c\bar{\mathbf{L}}_s - \underline{\mathbf{S}}({}^c\bar{\boldsymbol{\omega}}_s) {}^c\dot{\bar{\mathbf{L}}}_{th} + {}^c\dot{\bar{\mathbf{L}}}_{th} \right] \\ \underline{\mathbf{J}}_{rw}^{-1} [{}^h\mathbf{P}({}^w\bar{\boldsymbol{\Lambda}}_m \bar{\boldsymbol{\omega}}_m + \underline{\boldsymbol{\Gamma}}_m \bar{\mathbf{u}}_m)] \end{bmatrix} \quad (8.18)$$

These model are used to propagate the previous a posteriori state estimation to an a priori state estimation. This is done using a forward Euler numerical integration method.

The testbed hardware is equipped with sensors for performing attitude determination. These sensors are: accelerometer, gyroscope and magnetometer, furthermore the reaction wheels is equipped with encoders which enables measurements of the angular velocities of these. For designing output equations for the MEKF, and for simulation purposes, the sensor measurements are modelled and presented in the following.

The accelerometer measurements is modelled as a gravity vector in GRF rotated by the attitude quaternion.

$${}^b\bar{\mathbf{a}} = \underline{\mathbf{K}}_a \left[ {}_g^b\mathbf{q}^* {}^g\bar{\mathbf{g}} {}_g^b\mathbf{q} + {}^b\bar{\mathbf{a}}_{bd} \right] + \bar{\mathbf{v}}_a \quad (8.19)$$

Where:

$\underline{\mathbf{K}}_a$	is a scale factor matrix	[.]
${}_g^b\mathbf{q}$	is the rotation quaternion from GRF to BRF	[.]
${}^g\bar{\mathbf{g}}$	is the gravity	[m/s <sup>2</sup> ]
$\bar{\mathbf{v}}_a$	is the accelerometer noise	[.]
${}^b\bar{\mathbf{a}}_{bd}$	is the linear acceleration of the body	[m/s <sup>2</sup> ]

The Gyroscope measurements is modelled as the estimated angular velocity of the body, taken from the state vector, added with gyroscope noise.

$${}^b\bar{\boldsymbol{\omega}} = \underline{\mathbf{K}}_g {}^b\bar{\boldsymbol{\omega}}_s + \bar{\boldsymbol{v}}_g \quad (8.20)$$

Where:

${}^b\bar{\boldsymbol{\omega}}$	is the angular velocity measured	[rad/s]
$\underline{\mathbf{K}}_g$	is a scale factor matrix	[.]
$\bar{\boldsymbol{v}}_g$	is the gyroscope noise	[.]

The magnetometer measurements is modelled similar to the accelerometer. A magnetic field vector pointing north in GRF is rotated by the attitude quaternion.

$${}^b\bar{\mathbf{h}} = \underline{\mathbf{K}}_m \left[ {}_g^b\mathbf{q}^* {}^g\bar{\mathbf{h}} {}_g^b\mathbf{q} \right] + \bar{\boldsymbol{v}}_m \quad (8.21)$$

Where:

${}^g\bar{\mathbf{h}}$	is the magnetic field vector pointing north in GRF	[.]
$\underline{\mathbf{K}}_m$	is a scale factor matrix	[.]
${}_g^b\mathbf{q}$	is the rotation quaternion from GRF to BRF	[.]
$\bar{\boldsymbol{v}}_m$	is the magnetometer noise	[.]

The last of the sensors is the reaction wheel encoders. The reaction wheels angular velocity is in the state vector of the MEKF, hence the sensor model is:

$$\bar{\boldsymbol{\omega}}_{rw} = \underline{\mathbf{I}}_{4 \times 4} \bar{\boldsymbol{\omega}}_{rw} + \bar{\boldsymbol{v}}_{rw} \quad (8.22)$$

Where:

$\bar{\boldsymbol{\omega}}_{rw}$	is the angular velocities of the reaction wheels	[rad/s]
$\bar{\boldsymbol{v}}_{rw}$	is the encoder noise	[rad/s]

The scale factor matrices  $\underline{\mathbf{K}}_a$ ,  $\underline{\mathbf{K}}_g$  and  $\underline{\mathbf{K}}_m$  are positive definite matrices which is used to scale the modelled vector into something comparable to sensor readings. The sensor noises  $\bar{\boldsymbol{v}}_a$ ,  $\bar{\boldsymbol{v}}_g$ ,  $\bar{\boldsymbol{v}}_m$  and  $\bar{\boldsymbol{v}}_{rw}$  is assumed uncorrelated white Gaussian noise with mean zero and respectively covariance matrices  $\underline{\mathbf{R}}_a$ ,  $\underline{\mathbf{R}}_g$ ,  $\underline{\mathbf{R}}_m$  and  $\underline{\mathbf{R}}_{rw}$ .

Initial covariance matrices  $\underline{\mathbf{R}}_a$ ,  $\underline{\mathbf{R}}_g$ ,  $\underline{\mathbf{R}}_m$  and  $\underline{\mathbf{R}}_{rw}$  is found from a sample of sensor data logged from the actual sensors on the built testbed.

The covariance matrices  $\underline{\mathbf{Q}}$ ,  $\underline{\mathbf{R}}$  and  $\underline{\mathbf{P}}_0$  is however considered tuning matrices for the MEKF. This means that the covariance matrix found from sensor analysis is only considered an initial guess. The MEKF is tuned through an iterative process, first through simulations and since, on the actual testbed to obtain the best possible performance for the attitude determination.

The state error covariance matrix to time  $k = 0$ ,  $\underline{\mathbf{P}}_0$ , is defined to:

$$\underline{\mathbf{P}}_0 = \underline{\mathbf{I}}_{10 \times 10} \quad (8.23)$$

This relative large covariance is set because the filter has no knowledge of the initial state vector.

The process noise error covariance matrix  $\underline{\mathbf{Q}}$  and the measurement noise covariance matrix are defined to:

$$\underline{\mathbf{Q}} = \text{diag}(\underline{\mathbf{1}}_{1 \times 6} \cdot 10^{-4} \quad \underline{\mathbf{1}}_{1 \times 4} \cdot 10^{-3})$$

$$\underline{\mathbf{R}} = \text{diag}(\underline{\mathbf{1}}_{1 \times 3} \cdot 10^{-1} \quad \underline{\mathbf{1}}_{1 \times 3} \cdot 3.4 \cdot 10^{-6} \quad \underline{\mathbf{1}}_{1 \times 3} \cdot 2.23 \cdot 10^{-3} \quad \underline{\mathbf{1}}_{1 \times 4} \cdot 10^{-1})$$

Where  $\text{diag}()$  is a diagonal matrix with the arguments on its main diagonal.

### The Jacobian Matrices

Two Jacobian matrices are used in the MEKF. The first one is the  $\underline{\Phi}_k$  matrix calculated from the system equations. The  $\underline{\Phi}_k$  matrix is calculated as:

$$\underline{\Phi}_k = \underline{\mathbf{I}}_{10 \times 10} + \begin{bmatrix} \frac{\partial^c \mathbf{q}_s}{\partial^c \mathbf{q}_s} & \frac{\partial^c \mathbf{q}_s}{\partial^c \bar{\omega}_s} & \underline{\mathbf{0}}_{3 \times 4} \\ \underline{\mathbf{0}}_{3 \times 3} & \frac{\partial^c \bar{\omega}_s}{\partial^c \bar{\omega}_s} & \frac{\partial^c \bar{\omega}_s}{\partial \bar{\omega}_{rw}} \\ \underline{\mathbf{0}}_{4 \times 3} & \underline{\mathbf{0}}_{4 \times 3} & \frac{\partial \bar{\omega}_{rw}}{\partial \bar{\omega}_{rw}} \end{bmatrix} \cdot T_s \quad (8.24)$$

Where  $T_s$  is the sample time of the MEKF.

The  $\underline{\mathbf{H}}_k$  matrix is calculated as:

$$\underline{\mathbf{H}}_k = \begin{bmatrix} \frac{\partial^b \bar{\mathbf{a}}}{\partial^c \mathbf{q}_s} & \underline{\mathbf{0}}_{3 \times 3} & \underline{\mathbf{0}}_{3 \times 4} \\ \underline{\mathbf{0}}_{3 \times 3} & \underline{\mathbf{I}}_{3 \times 3} & \underline{\mathbf{0}}_{3 \times 4} \\ \frac{\partial^b \bar{\mathbf{h}}}{\partial^c \mathbf{q}_s} & \underline{\mathbf{0}}_{3 \times 3} & \underline{\mathbf{0}}_{3 \times 4} \\ \underline{\mathbf{0}}_{4 \times 3} & \underline{\mathbf{0}}_{4 \times 3} & \underline{\mathbf{I}}_{4 \times 4} \end{bmatrix} \quad (8.25)$$

### 8.3 Simulation

The MEKF presented in the previous Section, is simulated in order to validate its performance. For this a simulation environment is constructed in Simulink. The Simulation environment includes the spacecraft system model and sensor models. The output of the deterministic system models is fed to the sensor models which then outputs a modelled sensor signal used as measurements for the MEKF.

The simulation of the satellite model is initialised with an initial spin of 0.3 deg/s and an attitude of  $\mathbf{b}_{s,\text{init}} = [0 \ 0 \ 0.0872 \ 0.9962]^\top$  which corresponds to the angular velocity a CubeSat would have, after it has been detumbled and a quaternion rotation of 10 deg.

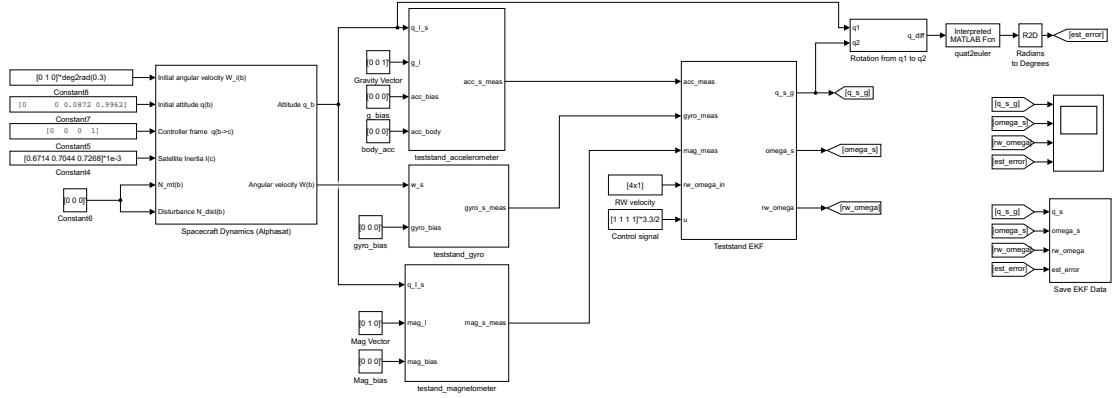
The initial states of the MEKF is defined as:

$$\bar{\mathbf{x}}_0 = [\mathbf{q}_{s,\text{init}} \quad \bar{\omega}_{s,\text{init}} \quad \bar{\omega}_{rw,\text{init}}]^\top \quad (8.26)$$

where

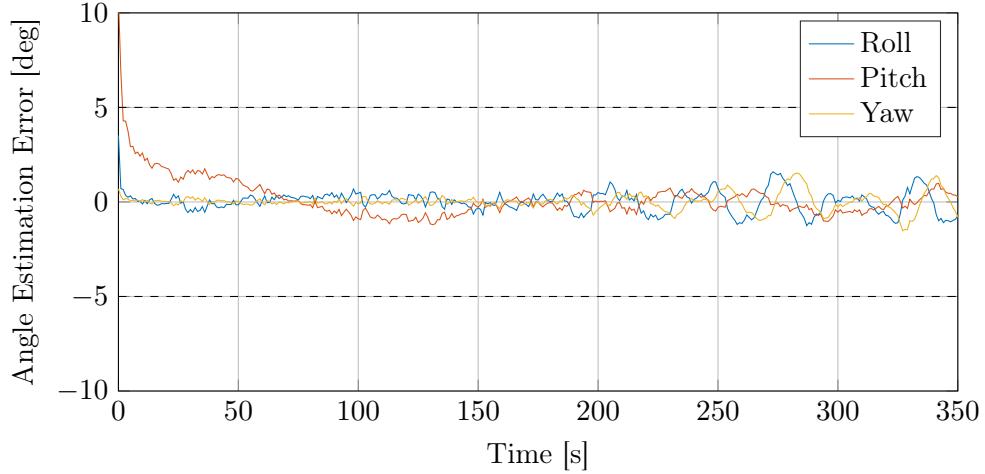
$$\begin{aligned} \mathbf{q}_{s,\text{init}} &= [0 \ 0 \ 0 \ 1] \\ \bar{\omega}_{s,\text{init}} &= [0 \ 0.0052 \ 0] \\ \bar{\omega}_{rw,\text{init}} &= [800 \ 800 \ 800 \ 800] \end{aligned} \quad (8.27)$$

which will give an initial estimation error of 10 deg. The simulation environment is shown in Figure 8.3.

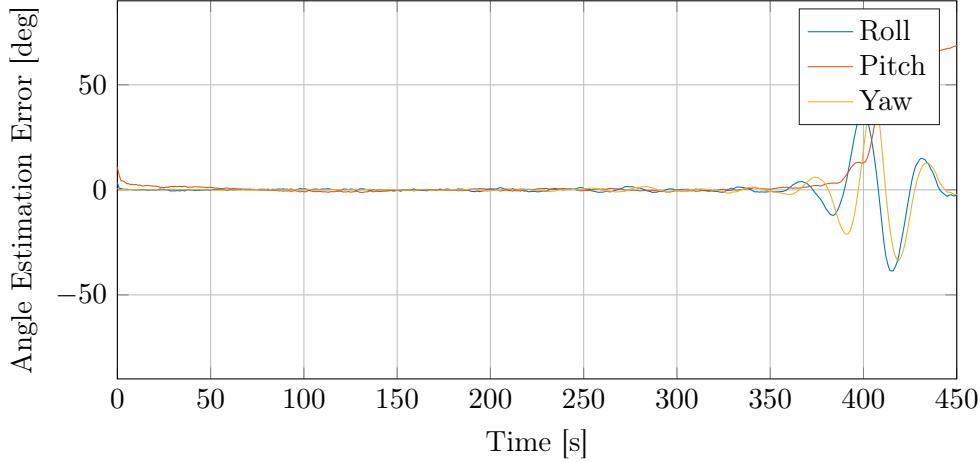
**Figure 8.3:** Shows the EKF simulation environment

The initial spin of the system model results in a change in attitude, which the EKF through the measurements should be able to estimate.

The first plot shown in Figure 8.4 shows that the estimation error converges from 10 deg to  $\approx 0$  which is expected. In the first quarter of a full rotation of the satellite, the EKF has an estimation error well within the requirements.

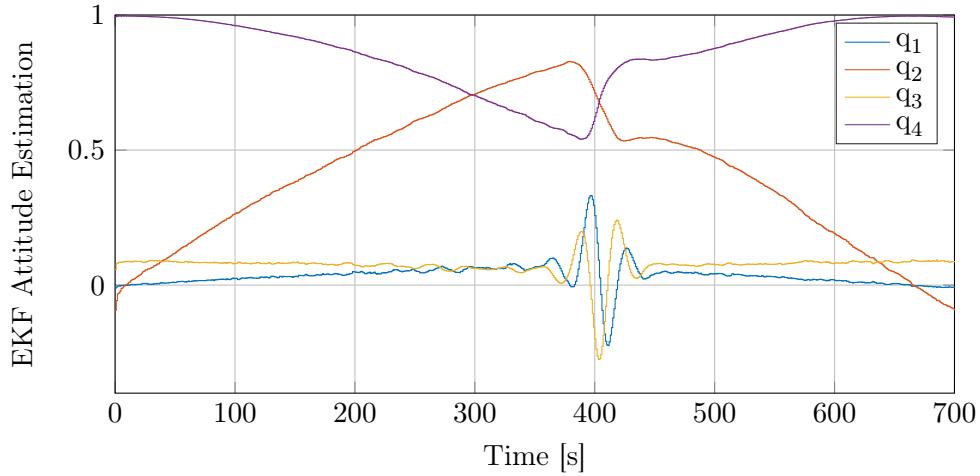
**Figure 8.4:** Shows the estimation error of the attitude, where the dashed lines specifies the requirements for the attitude determination.

But after approximately 380 sec. which is a rotation  $> 90$  deg, the estimation error grows rapidly. This is an unexpected result and deserves a further investigation. This behaviour is shown in Figure 8.5.



**Figure 8.5:** Shows that the estimator becomes unstable after  $\approx 90$  deg rotation.

By looking at the quaternion attitude estimate 8.6 it is seen that for unknown reasons,  $q_2$  and  $q_4$  changes directions. The expected result would have been have followed the characteristics of a sine and cosine.



**Figure 8.6:** Shows the quaternion attitude estimate behaviour after a quarter of a rotation.

Through further investigation the error has been narrowed down to the Update step in the MEKF. It has been concluded that the estimator is still usable in an area of  $\pm 90$  deg from the initial attitude, which means that it is still possible to use it in the testbed for attitude control algorithms.

## 8.4 Mechanical Design

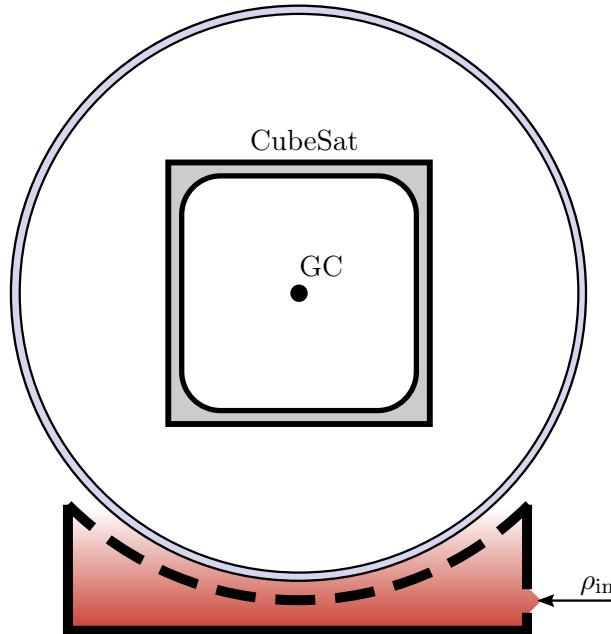
In order to test a satellite attitude control, a testbed is designed, which allows the satellite to rotate three-dimensional. First a short introduction to the problems to be solved and then the testbed design is described.

A satellite in orbit has the freedom to rotate around all three axes in weightlessness. This property is of course hard to replicate on the surface of the Earth where mass is under the constant influence of the gravitational pull. The force from the gravitation acting on a mass can be countered by applying an equally large opposite force, but there

lies a challenge in distributing the force evenly on the mass and allowing it to rotate freely. The motion created when a force is applied would also add a resisting force, friction, working in the opposite direction of the motion.

To create a friction reduced platform for attitude control the force acting on the satellite body must be evenly distributed and adding minimal friction. These challenges is solved by applying the same principles used in air-barrings.

Air-barrings uses a thin film of pressurised air to create low friction interface between surfaces. This requires that the satellite has a evenly smooth surface and the force created by the pressurised air is pointing in the direction of the satellite CoM. This is accomplished by installing the CubeSat in a low mass acrylic ball as shown in Figure 8.7.

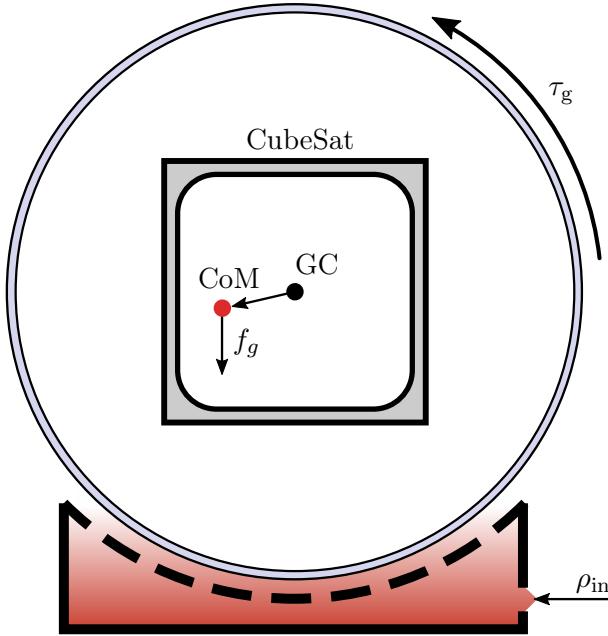


**Figure 8.7:** Shows the CubeSat suspended inside a low mass acrylic ball in the GC, where the colour gradient illustrates the pressurised air flow and  $\rho_{in}$  is the pressure inlet.

The layer of pressurised air between the ball and the bearing creates an outward going force and essentially levitating the ball. This effect enables the ball, with the CubeSat suspended inside, to rotate freely in all three-dimensions.

In order to perform successful test and verification of three-axis control algorithms on the testbed, the CubeSat should experience zero angular acceleration. This means that the only torque added to the rotating mass should be from the spacecraft's own actuators.

This problem arises especially if the CubeSats CoM is not in the GC of the ball of which the force from the pressurised air acting on the ball is perpendicular to. These forces are illustrated in Figure 8.8.



**Figure 8.8:** Shows the offset in CubeSat CoM which is affected by the gravity  $f_g$ .  $\tau_g$  is the torque generated by the CoM offset.

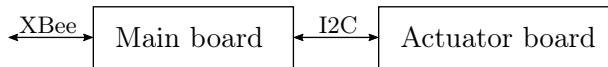
The external torque from the CoM offset will result in an unwanted oscillating angular acceleration. This is accompanied for by the use of distributable masses in the principal axes of the ball, this makes CoM adjustments possible. It should be mentioned that even though the CoM is adjustable it is still hard to make it perfect.

Now that the mechanical part of the testbed is designed it is possible to perform free movement in three-axis.

## 8.5 Hardware Design

In the following section the electronics of the attitude control testbed is described. The electronics of the testbed is divided up into two PCBs, a main board and an actuator board. This is done in order to make the system design as modular as possible - giving the possibility to change actuators, or try an alternative configuration of the reaction wheels. A block diagram of the electronics is shown in Figure 8.9.

The general approach of both the main and actuator boards are, that they work independent of each other and the only things common is the communication interface and the battery supply. The communication between the two boards is over an I2C serial connection. A block diagram of the attitude control testbed electronics is shown on Figure 8.9.

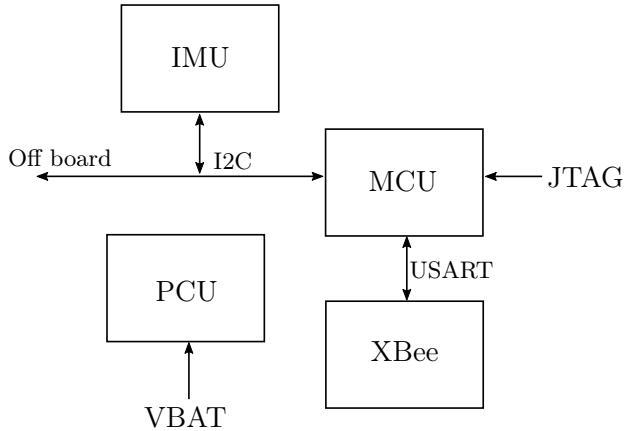


**Figure 8.9:** Block diagram of the attitude control testbed electronics hardware.

### 8.5.1 Main Board

The main board of the testbed is the *master*, in the sense that it is this subsystem which controls the sample time of, and order of execution of the complete system. This is

done in software. The main board features a power conditioning system with under-voltage protection, Microcontroller Unit (MCU) an Inertial Measurement Unit (IMU), consisting of accelerometer, gyroscope and a magnetometer, the communication with Robot Operating System (ROS) is also performed here, this is done using an XBee module mounted on the main board. A block diagram of the main board subsystem is shown on Figure 8.10.



**Figure 8.10:** Blockdiagram of the main board hardware.

An important point in the attitude control testbed is that the control and determination algorithms is running off-board, in Matlab/Simulink and therefore the computational complexity of these are not imposing on the choice of MCU, a Floating Point Unit (FPU) is therefore not necessary in the MCU. The main criteria for choosing a MCU are listed in below.

- Low power consumption
- Easy to implement
- Cost
- I2C interface
- USART Interface

Based on these criterias the Atmega2561 MCU from Atmel is chosen as main computer for the main board. The Atmega2561 MCU has an AVR8, 8-bit Reduced Instruction Set Computing (RISC) Harvard architecture, making it a simpler alternative than for example a 32-bit ARM MCU. Furthermore the Atmega2561 have enough RAM and flash memory to leave room for future ideas. The AVR8 architecture is a well known and well tested architecture, as both AAUSAT3, AAUSAT4 and AAUSAT5 flew with this architecture in mission critical subsystems. This is not a critical system in the sense that the systems on a satellite, but being well known make it easy to implement.

The Power Conditioning Unit (PCU) on the main board consists of a step-down switch-mode power supply, delivering 3.3 volts and up to 1.5 Ampere to all the modules of the main board. The switch-mode approach is chosen for its high efficiency compared to a linear power supply approach. The under-voltage protection in the PCU is because that the attitude control testbed is powered by a lithium-polymer battery. A lithium-polymer battery can be damaged if its cells are drained below a certain level which necessitates

under-voltage protection in order to protect the battery from being damaged. The power supply simply closes down when the battery level goes below a threshold and turns on again when charged back up to a higher threshold, introducing a hysteresis.

For attitude determination three sensors are used; a magnetometer, an accelerometer and a gyroscope. The criterias for choosing these are listed below.

- Accuracy
- Low power consumption
- Easy to implement
- Cost

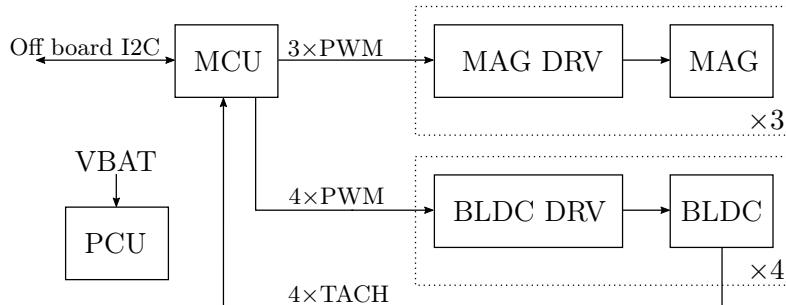
Based on these criterias the MPU9250 IMU from Invensense is chosen. The MPU9250 has all three sensors built into

the same package and the total current consumption when all three axes of all three sensors are in use is at 3.5 mA [Inv 14].

Pictures and schematics of the main board can be seen in Section B.1.

### 8.5.2 Actuator Board

The actuator board contains the drivers for both magnetic Torquers and the reaction wheels, furthermore the reaction wheels are mounted on this PCB. The actuator board has two functions seen from the main board; the main board can control the actuators via the I2C serial interface and the main board can request the instantaneous angular velocity of the reaction wheels. A block diagram of the hardware on the actuator board is shown in Figure 8.11.



**Figure 8.11:** Blockdiagram of the actuator board hardware.

A central piece on the actuator board is an MCU, which serves as the point of contact for the main board and is the controlling part for both the magnetic Torquers and the reaction wheels. An MCU for the actuator board does not need to be able to perform complex mathematical operations due to the fact that the MCU only serves as a low level controller. This controller needs to be able to control four reaction wheels and three magnetic Torquers individually, meaning that a total of seven Pulse Width Modulation (PWM) channels are necessary. The tachometer signals from the BLDCs are in the form of pulses, where each BLDC generates three pulses per revolution, which results in the need for four external interrupt inputs. The main criterias for choosing a MCU are listed in below.

- Seven PWM channels
- I2C interface
- Four external interrupt inputs
- Low power consumption
- Easy to implement
- Cost

Based on these criterias the Atmega64 MCU from Atmel is chosen. The Atmega64 is based on the same AVR8 architecture as with the MCU chosen for the main board. This match between architectures eases the implementation due to the fact that the low level software drivers can be used for both MCUs. Additionally the Atmega64 MCU has enough RAM and Flash memory to be future proof if someone in the future wanted to experiment with running local controllers aboard the actuator board.

The PCU on the actuator board is designed similar to the PCU of the main board. It is designed as a step-down switch-mode regulator with under-voltage protection for the lithium-polymer battery. The PCU of the actuator board delivers both 3.3 and 5 volts and up to 1.5 Ampere per power channel.

The magnetic Torquers are driven by drivers on the actuator board. The SI9988 full H-bridge drivers are chosen for this. The SI9988 drivers are chosen because they are well known and well tested as they were flown on both the AAUSAT3, AAUSAT4 and AAUSAT5 missions.

The drivers for the BLDC motors are chosen based on the requirements stated in the datasheet for the Maxon BLDC [Max 15]. Furthermore in order to simplify the interface to the reaction wheels a BLDC driver with a single PWM and a direction input where desired. The main criterias for choosing a BLDC driver are listed in below.

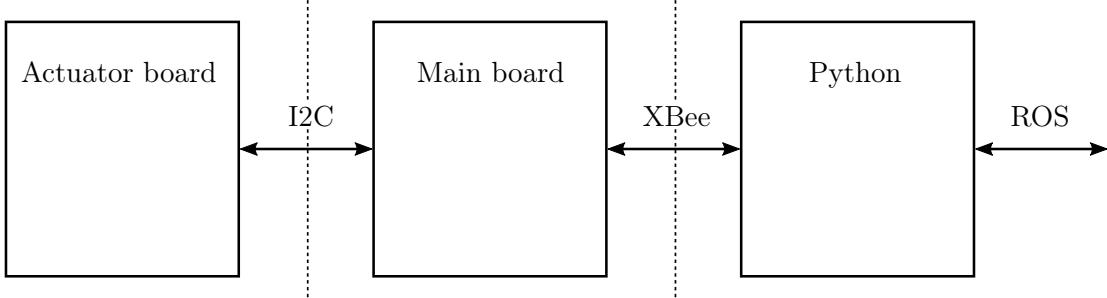
- A single PWM input
- A single direction pin
- Easy to implement
- Cost

Based on these main criterias the DRV10963 from Texax Industries is chosen. The DRV10963 BLDC motor driver has a simple interface with just one PWM input and one direction pin where high/low decides which direction the motor turns, this direction can be changed during operations. However for the reaction wheels the direction pin is fixed at a potential resulting in a counter clockwise rotational direction. The reaction wheel runs at a bias Rotations Per Minute (RPM) equalling half of their maximum revolution speed, and hence no direction change is needed after initialisation.

Pictures and schematics of the actuator board can be seen in Section B.2.

## 8.6 Software Design

The attitude control testbed software is comprised of three parts: the software on the main board, actuator board and the computer side interface to the embedded system. The software in the embedded systems; main and actuator boards are written in the C programming language, and the computer side software is written in the Python language. In the following, these three parts will be described. A block diagram of the main software parts can be observed in Figure 8.12.



**Figure 8.12:** Block diagram of the general software of the testbed.

The main software parts shown in Figure 8.12, spans over three different “computers”, illustrated by the dashed lines; on the actuator and main board, the software is running on an embedded processor and the Python part is running on a PC, which may also be running the control and estimation algorithms. For this implementation, this is the case. The general flow of the attitude control testbed is; that the main board sends a sample of sensor measurements and reaction wheels angular velocity to the embedded system interface and hence Matlab/Simulink, this is then processed and a set of control inputs for the reaction wheels is generated and sent back to the main board where this is applied. Matlab/Simulink does not do anything between the arrival of a new set of sensor and angular velocity measurements.

### 8.6.1 Embedded System Interface

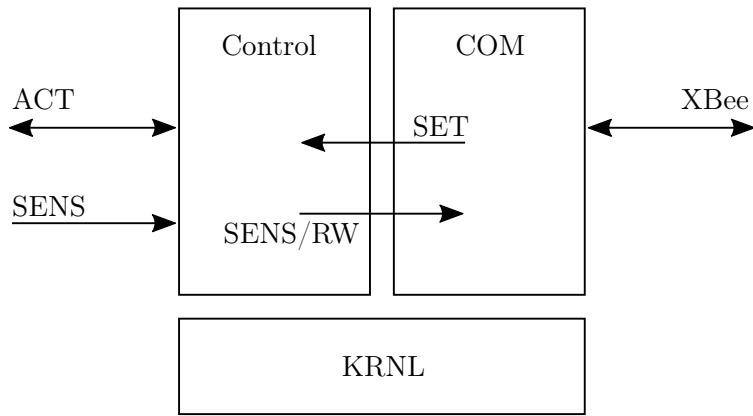
The software on the PC side of the attitude control testbed is in charge of the communication with the embedded electronics. It then serves as an interface between, in this case, Matlab/Simulink on one side and the embedded system. This interface is implemented as a ROS node written in the Python language. ROS is a meta-operating system for robots; meta-operating system in the sense that it is a layer which is installed on top of another operating system. A ROS network is built of a number of different ROS nodes, written in C++ or Python. A ROS node is a process that, when it initialises it, announces itself to the ROS master and are given an identity on the network. ROS nodes communicate through a publish/subscribe message passing system, over a topic. A topic is the ROS terminology for a communication channel, all nodes in the network can publish or subscribe to a topic. A ROS network can span over several physical computers, making ROS a highly modular and flexible approach to controlling robots, or as in this case giving a bridge to an embedded system. This capability to span over several physical computers, can give access to the attitude control testbed without connecting to the physical system, by dedicating a computer to the testbed.

The approach of using ROS as a communication interface enables, Matlab/Simulink or any other software suites which is able to connect to a ROS network, to use the testbed. This implements the idea of a general testbed mentioned in Section 8.1. The

publish/subscribe communication pattern provided by ROS, simplifies the Application Programming Interface (API) to the testbed. This API results in, from the controller/estimators point of view - connection to the sensors and actuators of the hardware.

### 8.6.2 Main Board

The main board is the subsystem which controls the sampling time and the timing of the different jobs in the overall system. The general flow of the overall system is obtained by the main board. This is achieved by making this board, the master in the sense that it is this subsystem, which initiates communication with, both the PC side software and the actuator board. In order to ensure a deterministic sample time of the system, a Real-Time Operating System (RTOS) is implemented. For this the RTOS KRNL [Nielsen 16] is used. A block diagram of the software with the threads is shown in Figure 8.13.



**Figure 8.13:** Block diagram of the software within the main board.

In the main board software there are two threads of equal priority; one for communication with the PC, denoted *COM* in Figure 8.13, side software and one for reading the IMU and controlling the actuator board, denoted *Control* in Figure 8.13. For threads of equal priority KRNL uses a pre-emptive prioritised round robin scheduling algorithm, ensuring a deterministic sampling time for the attitude control testbed [Nielsen 16]. The communication between these two threads is performed through two globally defined structs, one for relaying set point from the communication thread to the control thread, denoted *SET* in Figure 8.13 and one for the sensor reading combined with the angular velocities of the reaction wheels, denoted *SENS/RW* in Figure 8.13. Both these structs are protected by binary semaphores in order to prevent race conditions. The scheduler runs at 100 Hz; where each thread checks for new data. The communication thread polls for new messages from the PC, when new control values are received, they are relayed to the control thread, and when there are new sensor measurements they are relayed to the PC. The control task performs sensor readings with a 10 Hz sample rate, sensors readings contains both IMU and reaction wheels angular velocity, and relay them to the communication thread. When the control thread receives new reaction wheels set points they are relayed to the actuator board.

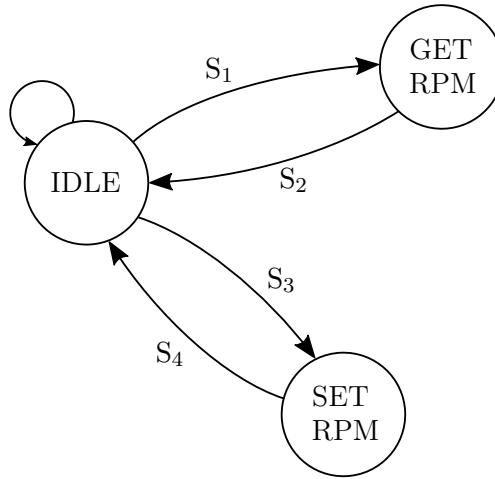
The control thread is therefore the responsible for the testbed running at a sample rate of 10 Hz.

### 8.6.3 Actuator Board

As previously mentioned the actuator board functions as a slave to the main board. The software of the actuator board is in charge of three things:

- Handling the tachometer pulses from the reaction wheels
- Controlling the reaction wheels
- Controlling the magnetic Torquers
- Communication with the main board

This is accomplished by implementing the software of the actuator board as a Finite State Machine (FSM). The states are: **IDLE**, **GET\_RPM** and **SET\_RPM**. A state transition diagram of the FSM is shown in Figure 8.14.



**Figure 8.14:** Actuator board state machine.

The state transition conditions S<sub>1</sub> to S<sub>4</sub> are:

- S<sub>1</sub>: When the actuator board receives an order to fetch the angular velocities of the reaction wheels.
- S<sub>2</sub>: When the angular velocities has been read and send back to the main board.
- S<sub>3</sub>: When the actuator board receives an order adjust the control values of the reaction wheels.
- S<sub>4</sub>: When the reaction wheels control values has been adjusted.

When the actuator board is powered up, it while initially go into the **IDLE** state. When a command from the main board is received the actuator board will perform a state transition based on the state transition conditions. In the state **GET\_RPM**, the software returns values, to the main broad, representing the current angular velocity of each of the reaction wheels, before transitioning back to the **IDLE** state, waiting for new commands. When a command for setting the angular velocities of the reaction wheels, along with set points, are received; the FSM transitions to the **SET\_RPM** state, where the angular

velocities is set to the value received from the main board. The FSM transitions back to the IDLE state when this is performed successfully.

Implementing the software as an I2C slave lets the actuator board act as a slave to the main board. The tachometer pulses are handled with external interrupts - a tachometer pulse happens three times per revolution of the reaction wheel and with four reaction wheels this generates as many tachometer pulses. A tachometer interrupt triggers an associated interrupt service routine which increases a counter variable. This is in turn used when the main board requests the angular velocity of the reaction wheels, this is happening in the GET\_RPM state.

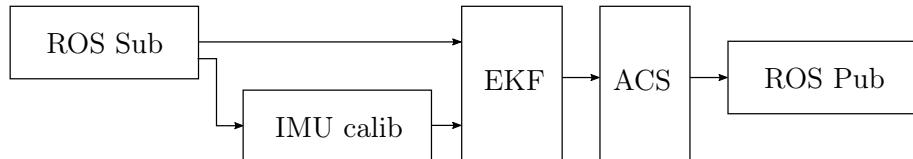
Control of the reaction wheels and the magnetic Torquers are done by generating a PWM for each. When a command for changing these is received, then new set points are received at the same time, and the value is written to a control register for the associated PWM output.

The I2C communication with the main board is performed using an interrupt driven I2C interface.

#### 8.6.4 Matlab/Simulink Implementation

The purpose of the implementation in Matlab/Simulink is to provide an environment for which, attitude control algorithms can easily be implemented and changed during the research and development processes. Due to the nature of the system where the testbed is, from a software point of view, interfaced to, as a ROS node; this could be in for instance the Python or Julia languages, but it is chosen to implement in Matlab/Simulink.

The Matlab/Simulink implementation for the attitude control testbed is based upon the Robotics System Toolbox from MathWorks. Furthermore the Simulink implementation is created as an extension of the AAUSAT Simulink library, along with other contributions described in Chapter 6. A general block diagram of the implementation is shown in Figure 8.15.



**Figure 8.15:** Block diagram of the Matlab/Simulink implementation of the attitude control testbed.

Where the blocks *ROS Pub* and *ROS Sub* respectively represents where the actuator set points is published to the ROS network and where the sensors and angular velocities of the reaction wheels is subscribed to from the ROS network.

From the ROS network the sensor values from magnetometer, accelerometer and gyroscope is received together with tachometers readings through the associated ROS topic. The tachometer readings is sent to the Kalman filter described in Sections 8.2.3 and 8.2.4 and the sensor measurements arrives in their raw form and is sent through their respective calibration blocks.

Both the EKF and the ACS is being enabled by the sensor topic; this is in order to enforce the sample rate defined in the main board RTOS. Consequently reaction

wheels set point values are published by the rate which the sensors values arrive to Matlab/Simulink.

## 8.7 Summary

In this chapter the design of the attitude control testbed is described. Both mechanical and electronic hardware was described along with the software implementation. The software in the PC running the control and attitude determination algorithms and the embedded software implementation in the embedded micro-controllers in the testbed is described.

# Acceptance Test

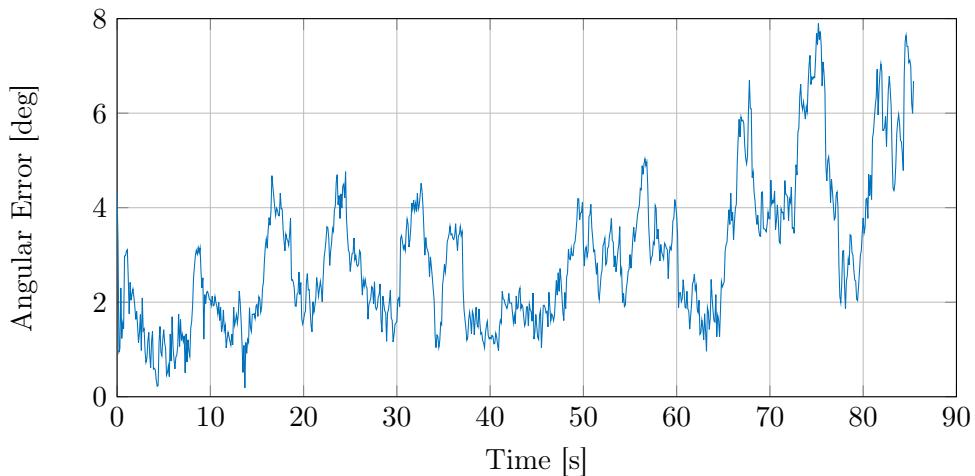
During the previous chapters, the design of attitude control algorithms and the design of the attitude control testbed has been described. In the following chapter these designs will be tested against the requirements listed in chapter 3.

The first thing to verify against the requirements is the attitude control testbed, since this is the one which will be used to test the requirements of the attitude control algorithm.

## 9.1 ACS Testbed

Two sets of requirements were defined; functional requirements and requirements for the Attitude determination. The functional requirements are not tested but verified through the design.

In the simulations of the attitude determination in Section 8.3 it was observed that the attitude determination performs within the requirements during attitudes with a roll, pitch or yaw below 90 degrees. A test with the estimator running with inputs from the sensors within the testbed is performed. This is shown in Figure 9.1.



**Figure 9.1:** Attitude error magnitude,  $\|\bar{\theta} - \hat{\theta}\|_2$  for a 80 degree step in yaw.

In Figure 9.1 the estimation error is shown. The estimation error is found by using a true vector with Euler angles,  $\bar{\theta}$ , calculating the estimated Euler angle vector,  $\hat{\theta}$ ,

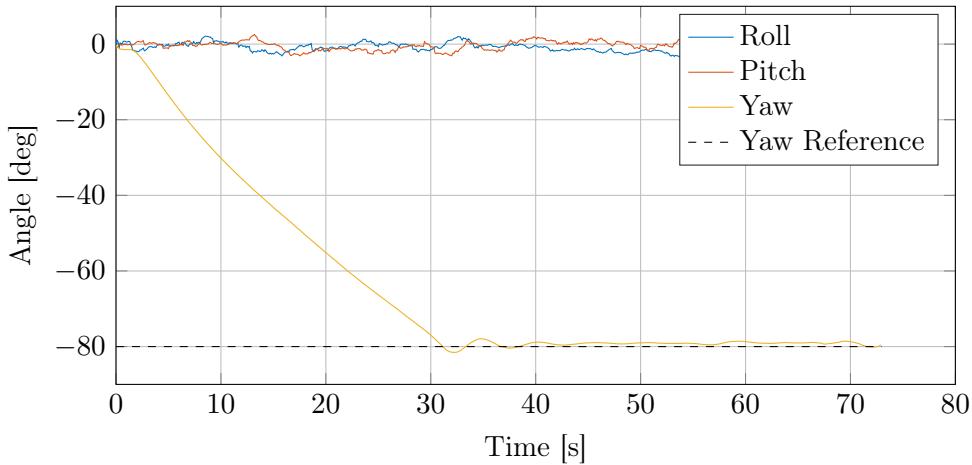
from the estimated attitude quaternion and calculating the two-norm of the difference between these,  $\|\bar{\theta} - \hat{\theta}\|_2$ . It should be noted that the *true* value is calculated from raw sensor measurements because an actual *true* was not available, this can be expected to induces some noise on the  $\bar{\theta}$  vector, and hence also the estimation error. With this in mind there is room for improvements in the tuning of the attitude estimator. The accuracy requirement for the attitude estimator states that the estimator should be able to estimate within an error of 5 deg, this performance is not completely reached but sufficient for attitude control algorithm verification.

## 9.2 Control algorithm

In the following the test and verification of the sliding mode attitude control algorithm are described.

During the simulations of the sliding mode control algorithm in Section 7.2, it was shown that the sliding mode control algorithm performs well within the requirements listed in Section 3. This will be tested in the following section using the attitude control testbed developed and built during this thesis.

In Figure 9.2 an 80 degree step in yaw is shown.



**Figure 9.2:** 80 degree step in yaw, with the sliding mode control developed in Section 7.2.

The Attitude control requirements from Section 3.2.5 is listed in Table 9.1.

No.	Title	Description
R4.1	Accuracy	Must be able to point < 10 deg.
R4.2	Slew rate	Must be able to change the attitude with a slew rate > 0.0257 rad/s
R4.3	Settling time	< 5 deg max. motion, 1 min.

**Table 9.1:** Requirements for the target tracking mode.

**R4.1 Accuracy:** The attitude control is required to have an accuracy of below 10 deg; in Figure 9.2 is can be seen that the attitude control is well within this requirement.

**R4.2 Slew rate:** The attitude control is required to be able to deliver a slew rate of > 0.0257 rad/s. From Figure 9.2 is can be seen that the control algorithm can

rotate the testbed hardware 80 degrees in just over 30 seconds, giving a slew rate of 0.045 rad/s: This is well within the requirement.

**R4.3 Settling time:** The attitude control is required to be within 5 degrees of reference in less than a minute, From Figure 9.2 it can be seen that the attitude control is within this requirement in 29.2 seconds.

This concludes the acceptance tests of both the attitude control testbed and the sliding mode attitude control algorithm. The attitude determination of the testbed is still missing some work. The determination performance is sufficient, within a roll, pitch and yaw angle, below 90 degrees, but above this, the MEKF behaves erratically. A solution has yet to be found. With a solution, and some tuning, the filter is expected to perform within the requirements.

The sliding mode control algorithm performs well and with a large margin within the requirements.



---

# Closure

Through the scope of this thesis, different aspects of satellite attitude control has been investigated, analysed, designed and then tested. This chapter collects the conclusions drawn on each area from the development of the attitude control to ideas for future improvements.

## 10.1 Conclusion

Through this thesis the following conclusions has been drawn.

### 10.1.1 Quaternion Sliding Mode Control

The SMC designed through this thesis benefits from using a pure quaternion sliding surface. Moreover, the SMC has taken the largest environment disturbances into account in order to ensure stability. Through the simulations it was seen that the SMC was able to perform target tracking. The SMC in comparison to the linear regulator was able to change the attitude more rapid but this could have something to do with the fact that the linear regulator was scaled down to ensure stability of the inner and outer loop.

Summing up; both the linear and the nonlinear control algorithms performs well within the requirements listed for Nadir and Target tracking, and they are both able to reject orbit disturbances, shown through simulations and for the SMC also shown in the attitude control testbed.

Moreover, it has been investigated that the Coriolis moments generated by each reaction wheel in the Tetrahedron configuration when rotation, in an inertial reference frame is sufficiently small enough to be neglected in this thesis. But for a spacecraft mission with very high attitude accuracy requirements it might be beneficial to compensate for the coriolis moments.

### 10.1.2 Design of Multiplicative Extended Kalman Filter

An MEKF has been designed and simulated. The simulation showed that the non-linear estimator was able to perform within the requirements defined in chapter 3 in an area of  $\pm 90$  from the initial attitude. It was concluded that the estimator was still able to be used in order to verify the SMC in the testbed.

### 10.1.3 Attitude Control Testbed

Through this thesis an attitude control testbed was designed. The testbed is able to perform friction reduced motion in all three axes. Moreover, a actuator and main board for telemetry, control feedback and wireless communication with a user interface has been developed. This enables the use of HIL for verification of control algorithms or future experiments.

Even though that the testbed has the capabilities of centre of mass adjustment, perfectly mass centering was not accomplished. This results in an initial acceleration of the ball on the air baring which is to large for the actuator board to counter. The conclusion to this problem was to only perform attitude control in the yaw axis.

### 10.1.4 Sliding Mode Implementation

The SMC was tested and verified by the use of the testbed. Even though the test only was performed in one axis the results was still sufficient enough to conclude that on the performance on the SMC. The test showed that the controller was well within the requirements.

## 10.2 Future Developments

During the tests of the attitude control testbed, it was observed that the centre of mass of the testbed was rather difficult to adjust, by the manual adjustment system developed, during this thesis. It is possible to adjust the mass centre to be close to the geometric centre but it was not close enough to perform three-dimensional attitude control by the use of reaction wheels. The reaction wheels saturate trying to overcome the gravity and falls down, just to try again. We propose that in order to perform three-dimensional attitude control on the testbed, an automatic adjustment system shoud be developed and implemented.

---

# Bibliography

- [Bak 96] Thomas Bak, Rafał Wiśniewski & Mogens Blanke. *Autonomous Attitude Determination and Control System for the Ørsted Satellite*. IEEE, Aerospace Application Conference, 1996.
- [Bak 99] Thomas Bak. *Spacecraft Attitude Determination - a Magnetometer Approach*. PhD thesis, Aalborg University Denmark, 1999.
- [Blanke 10] Mogens Blanke & Martin Birkelund Larsen. *Lecture note in Satellite Dynamics and Control in a Quaternion Formulation*, September 2010.
- [Bryson 69] Bryson, A. E. & Y. C. Ho. Applied optimal control. Waltham, MA: Blaisdell, 1969.
- [CalPoly 14] CalPoly. *CubeSat Design Specification Rev. 13*. Rapport technique, California Polytechnic State University, February 2014.
- [Cappellari 76] J. O. Cappellari, C. E. Velez & A. J. Fuchs. Mathematical theory of the goddard trajectory determination system. NASA - Goddard Space Flight Center, Greenbelt, Maryland, April 1976.
- [Curtis 05] Howard D. Curtis. Orbital mechanics for engineering students. Elsevier Butterworth-Heinemann, 1st edition, 2005.
- [Franklin 10] Gene F. Franklin, J. David Powell & Abbas Emami-Naeini. Feedback control of dynamic systems. Pearson, 6th edition, 2010.
- [Grewal 08] Mohinder S. Grewal & Angus P. Andrews. Kalman filtering - theory and practice using matlab. John Wiley & Sons, Inc., 3rd edition, 2008.
- [Hoots 80] Felix R. Hoots & Ronald L. Roehrich. *Models for Propagation of NORAD Element Sets*. Spacetrack Report, no. 3, December 1980.
- [Humphreys 02] Todd E. Humphreys. *Attitude Determination for Small Satellites with Modest Pointing Constraints*. Utah State University, 2002.
- [Inv 14] Invensense. *MPU9250 Product Specification rev. 1.0*, 01 2014.
- [Jensen 10] Kasper Fuglsang Jensen & Kasper Vinther. Attitude determination and control system for aausat3. Master's thesis, Aalborg Universitet, 6 2010.

- [Khalil 02] Hassan K. Khalil. Nonlinear systems. Prentice Hall, 3rd edition, 2002.
- [Kuipers 02] Jack B. Kuipers. Quaternions and rotation sequences. Princeton University Press, 2002.
- [Markley 03] F.Landis Markley. *Multiplicative vs. Additive Filtering for Spacecraft Attitude Determination*. Rapport technique, NASA Goddard Space Flight Center, December 2003.
- [Markley 14] F. Landis Markley & John L. Crassidis. Fundamentals of spacecraft attitude determination and control. Springer, 1st edition, 2014.
- [Max 15] Maxon Motor. *Maxon flat motors*, 04 2015.
- [NASA 69a] NASA. Spacecraft gravitational torques. NASA, May 1969.
- [NASA 69b] NASA. Spacecraft magnetic torques. NASA, Marts 1969.
- [NASA 69c] NASA. Spacecraft radiation torques. NASA, October 1969.
- [NASA 71] NASA. Spacecraft aerodynamic torques. NASA Jet Propulsion Laboratory, January 1971.
- [Nielsen 16] Jens Frederik Dalsgaard Nielsen. *KRNL*. <https://github.com/jdn-aau/krnl>, 2016. V.2016052.
- [Serway 14] Raymond A. Serway & John W. Jewett. Physics for scientists and engineers with modern physics. Cengage Learning, 9th edition, 2014.
- [Sidi 97] Marcel J. Sidi. Spacecraft dynamics & control. Cambridge University Press, 1st edition, 1997.
- [Stuelpnagel 64] John Stuelpnagel. *On the Parametrization of the Three-Dimensional Rotation Group*. SIAM Review, vol. 6, no. 4, October 1964.
- [Sørensen 10] Ole Sørensen. *Lecture note in Optimal Control*, February 2010.
- [Utkin 09] Vadim Utkin, Jürgen Guldner & Jingxin Shi. Sliding mode control in electro-mechanical systems. Automation and Control Engineering Series. Taylor & Francis Group, 2nd edition, 2009.
- [Wertz 95] James R. Wertz. Spacecraft attitude determination and control. Kluwer Academic Publishers, 1995.
- [Wertz 99] James R. Wertz & Wiley J. Larson. Space mission analysis and design. Space Technology Library. Springer Netherlands, 1999.
- [Wie 98] Bong Wie. Space vehicle dynamics and control. AIAA Education Series. American Institute of Aeronautics and Astronautics, Inc., 1st edition, 1998.
- [Woellert 11] Kirk Woellert, Pascale Ehrenfreund, Antonio J. Ricco & Henry Hertzfeld. *Cubesats: Cost-effective science and technology platforms for emerging and developing nations*. Advances in Space Research, 2011.
- [Xia 12] Chang-Liang Xia. Permanent magnet brushless dc motor drivers and controls. John Wiley & Sons Singapore Pte. Ltd., 2012.

# Quaternions and Rotations

## A.1 Quaternions

Sources: [Wertz 95] [Markley 14] [Kuipers 02] [Stuelpnagel 64]

In this thesis, rotations are performed by the use of quaternions. Quaternions provide a convenient method of handling three-dimensional rotations. In the following section, the basics of quaternions are described followed by a description of the use of quaternions for rotations.

A quaternion is a hypercomplex number consisting of a real part and three imaginary parts. This can be written as:

$$\mathbf{q} = q_4 + q_1 \bar{i} + q_2 \bar{j} + q_3 \bar{k} \quad (\text{A.1})$$

Where the imaginary numbers  $\bar{i}$ ,  $\bar{j}$  and  $\bar{k}$  can be interpreted as vectors, spanning an orthonormal basis in  $\mathbb{R}^3$ . The quaternion can be written as:

$$\mathbf{q} = q_4 + \bar{\mathbf{q}} = q_4 + \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} q_4 \\ \bar{\mathbf{q}} \end{bmatrix} \quad (\text{A.2})$$

and

$$\bar{i}^2 = \bar{j}^2 = \bar{k}^2 = \bar{i}\bar{j}\bar{k} = -1 \quad (\text{A.3})$$

Where the imaginary parts form a noncommutative ring with

$$\begin{aligned} \bar{i}\bar{j} &= -\bar{j}\bar{i} = \bar{k} \\ \bar{j}\bar{k} &= -\bar{k}\bar{j} = \bar{i} \\ \bar{k}\bar{i} &= -\bar{i}\bar{k} = \bar{j} \end{aligned} \quad (\text{A.4})$$

The vector part of a quaternion is called the *pure* quaternion.

Multiplication between two quaternions are performed as

$$\begin{aligned}
\mathbf{q} \otimes \mathbf{p} &= (q_4 + \bar{\mathbf{q}})(p_4 + \bar{\mathbf{p}}) \\
&= q_4 p_4 - \bar{\mathbf{q}}^\top \bar{\mathbf{p}} + q_4 \bar{\mathbf{p}} + p_4 \bar{\mathbf{q}} + \bar{\mathbf{q}} \times \bar{\mathbf{p}} \\
&= \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} \\
&= \begin{bmatrix} p_4 & p_3 & -p_2 & p_1 \\ -p_3 & p_4 & p_1 & p_2 \\ p_2 & -p_1 & p_4 & p_3 \\ -p_1 & -p_2 & -p_3 & p_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}
\end{aligned} \tag{A.5}$$

Quaternion conjugate, norm and inverse are given as:

$$\begin{aligned}
\mathbf{q}^* &= q_4 - \bar{\mathbf{q}} = q_4 - \bar{i}q_1 - \bar{j}q_2 - \bar{k}q_3 \\
|\mathbf{q}| &= \sqrt{\mathbf{q}^* \otimes \mathbf{q}} = \sqrt{(q_4 - \bar{\mathbf{q}}) \otimes (q_4 + \bar{\mathbf{q}})} \\
\mathbf{q}^{-1} &= \frac{\mathbf{q}^*}{|\mathbf{q}|^2}
\end{aligned} \tag{A.6}$$

### A.1.1 Rotation by Quaternions

When using a quaternion for performing rotations, a constraint is imposed on the norm of a quaternion.

$$|\mathbf{q}| = 1 \tag{A.7}$$

A unit norm quaternion representing a rotation in  $\mathbb{R}^3$  about a vector  $\bar{\mathbf{u}}$ , can be written as

$$\mathbf{q} = \cos\left(\frac{\theta}{2}\right) + \bar{\mathbf{u}} \sin\left(\frac{\theta}{2}\right), \tag{A.8}$$

a vector  $\bar{\mathbf{v}}$  can then be rotated  $\theta$  degrees, in positive direction of rotation, about  $\bar{\mathbf{u}}$  by

$$\bar{\mathbf{v}}_r = \mathbf{q} \otimes \begin{bmatrix} \bar{\mathbf{v}} \\ 0 \end{bmatrix} \otimes \mathbf{q}^* \tag{A.9}$$

where  $\bar{\mathbf{v}}_r$  is the rotated vector  $\bar{\mathbf{v}}$ .

If instead a vector is to be rotated from one reference frame,  $x$ , to another,  $y$ , this can be done by

$${}^y\bar{\mathbf{v}} = \mathbf{q}^* \otimes \begin{bmatrix} {}^x\bar{\mathbf{v}} \\ 0 \end{bmatrix} \otimes \mathbf{q} \tag{A.10}$$

Succesive rotations is simply performed as:

$$\begin{aligned}
{}^y\bar{\mathbf{v}} &= \mathbf{q}^* \otimes \begin{bmatrix} {}^x\bar{\mathbf{v}} \\ 0 \end{bmatrix} \otimes \mathbf{q} \\
\mathbf{p}^* \otimes \begin{bmatrix} {}^y\bar{\mathbf{v}} \\ 0 \end{bmatrix} \otimes \mathbf{p} &= \mathbf{p}^* \otimes \left( \mathbf{q}^* \otimes \begin{bmatrix} {}^x\bar{\mathbf{v}} \\ 0 \end{bmatrix} \otimes \mathbf{q} \right) \otimes \mathbf{p} \\
&= (\mathbf{q} \otimes \mathbf{p})^* \otimes \begin{bmatrix} {}^x\bar{\mathbf{v}} \\ 0 \end{bmatrix} \otimes (\mathbf{q} \otimes \mathbf{p})
\end{aligned} \tag{A.11}$$

Notice the sequence of rotations. First by  $\mathbf{q}$  then by  $\mathbf{p}$ .

A rotation matrix can be constructed from a quaternion as

$$\underline{\mathbf{R}}(\mathbf{q}) = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1 q_2 + q_3 q_4) & 2(q_1 q_3 - q_2 q_4) \\ 2(q_1 q_2 - q_3 q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2 q_3 + q_1 q_4) \\ 2(q_1 q_3 + q_2 q_4) & 2(q_2 q_3 - q_1 q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \quad (\text{A.12})$$

Where  $\underline{\mathbf{R}}(\mathbf{q})$  is the rotation matrix constructed form  $\mathbf{q}$ .

From a Rotation matrix  $\underline{\mathbf{R}}$ , the quaternion representing the same rotation can be found as

$$\begin{aligned} q_4 &= \pm \frac{1}{2} \sqrt{(1 + R_{11} + R_{22} + R_{33})} \\ q_1 &= \frac{1}{4q_4} (R_{23} - R_{32}) \\ q_2 &= \frac{1}{4q_4} (R_{31} - R_{13}) \\ q_3 &= \frac{1}{4q_4} (R_{12} - R_{21}) \end{aligned} \quad (\text{A.13})$$

In cases where  $q_4$  is very small, the calculations of the rest of the quaternion becomes inaccurate. To solve this, an other solution to find the quaternion is used in terms of  $R_{ij}$  in order to find a larger divisor [Sidi 97].

## A.2 Quaternion Dynamics

For now the an angular acceleration of the satellite  ${}^c\dot{\boldsymbol{\omega}}$  has been derived. The quaternion representation of this acceleration is investigated by the second derivative of the quaternion.

As described in the satellite kinematics Section 5.1, the first derivative of the quaternion is given as

$$\dot{\mathbf{q}} = \frac{1}{2} \underline{\Omega}(\bar{\boldsymbol{\omega}}) \mathbf{q} \quad (\text{A.14})$$

alternative representation of the first derivative shows

$$\begin{bmatrix} \dot{\bar{\mathbf{q}}} \\ \dot{q}_4 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} \bar{\mathbf{q}} \times \bar{\boldsymbol{\omega}} + \frac{1}{2} q_4 \bar{\boldsymbol{\omega}} \\ -\frac{1}{2} \bar{\mathbf{q}}^\top \bar{\boldsymbol{\omega}} \end{bmatrix} \quad (\text{A.15})$$

The second derivative of the quaternion is then given as

$$\ddot{\bar{\mathbf{q}}} = -\frac{1}{2} \dot{\bar{\mathbf{q}}} \times \bar{\boldsymbol{\omega}} - \frac{1}{2} \bar{\mathbf{q}} \times \dot{\bar{\boldsymbol{\omega}}} + \frac{1}{2} q_4 \bar{\boldsymbol{\omega}} + \frac{1}{2} q_4 \dot{\bar{\boldsymbol{\omega}}} \quad (\text{A.16})$$

if the quaternion kinematics in equation A.15 is used in equation A.16

$$\begin{aligned} \ddot{\bar{\mathbf{q}}} &= -\frac{1}{2} \left( -\frac{1}{2} \bar{\mathbf{q}} \times \bar{\boldsymbol{\omega}} + \frac{1}{2} q_4 \bar{\boldsymbol{\omega}} \right) \times \bar{\boldsymbol{\omega}} \\ &\quad - \frac{1}{2} \bar{\mathbf{q}} \times \dot{\bar{\boldsymbol{\omega}}} + \frac{1}{2} \left( -\frac{1}{2} \bar{\mathbf{q}}^\top \bar{\boldsymbol{\omega}} \right) \bar{\boldsymbol{\omega}} + \frac{1}{2} q_4 \dot{\bar{\boldsymbol{\omega}}} \\ &= -\frac{1}{4} \bar{\boldsymbol{\omega}} \times (\bar{\mathbf{q}} \times \bar{\boldsymbol{\omega}}) - \frac{1}{2} \bar{\mathbf{q}} \times \dot{\bar{\boldsymbol{\omega}}} - \frac{1}{4} \bar{\mathbf{q}}^\top \bar{\boldsymbol{\omega}} \bar{\boldsymbol{\omega}} + \frac{1}{2} q_4 \dot{\bar{\boldsymbol{\omega}}} \end{aligned} \quad (\text{A.17})$$

since there is a vector triple product the following relation holds

$$a \times (b \times c) = b(a^\top c) - c(a^\top b) \quad (\text{A.18})$$

By using this relation in equation A.17 the following is shown

$$\begin{aligned}\ddot{\bar{q}} &= -\frac{1}{4}\bar{q}(\bar{\omega}^\top \bar{\omega}) + \frac{1}{4}\bar{\omega}(\bar{\omega}^\top \bar{q}) - \frac{1}{2}\bar{q} \times \dot{\bar{\omega}} \\ &\quad - \frac{1}{4}(\bar{\omega}^\top \bar{q})\bar{\omega} + \frac{1}{2}q_4\dot{\bar{\omega}} \\ &= -\frac{1}{2}\bar{q} \times \dot{\bar{\omega}} + \frac{1}{2}q_4\dot{\bar{\omega}} - \frac{1}{4}\bar{q}\|\bar{\omega}\|_2^2 \\ &= -\frac{1}{2}[\underline{\mathbf{S}}(\bar{q}) - q_4\underline{\mathbf{I}}_{3\times 3}]\dot{\bar{\omega}} - \frac{1}{4}\bar{q}\|\bar{\omega}\|_2^2\end{aligned}\tag{A.19}$$

$$\underline{\mathbf{T}}(\mathbf{q}) = -\frac{1}{2}[\underline{\mathbf{S}}(\bar{q}) - q_4\underline{\mathbf{I}}_{3\times 3}]\dot{\bar{\omega}}\tag{A.20}$$

$$\begin{aligned}\ddot{q}_4 &= -\frac{1}{2}\dot{\bar{q}}^\top \bar{\omega} - \frac{1}{2}\bar{q}^\top \dot{\bar{\omega}} \\ &= -\frac{1}{2}\left(-\frac{1}{2}\bar{q} \times \bar{\omega} + \frac{1}{2}q_4\bar{\omega}\right)^\top \bar{\omega} - \frac{1}{2}\bar{q}^\top \dot{\bar{\omega}}\end{aligned}\tag{A.21}$$

In order to simplify this expression further the properties for a scalar triple product is used, for which following holds

$$a^\top (b \times c) = (a \times b)^\top c\tag{A.22}$$

applying the property of the scalar triple product to equation A.21 shows

$$\begin{aligned}\ddot{q}_4 &= \frac{1}{4}\bar{q}^\top (\bar{\omega} \times \bar{\omega}) - \frac{1}{4}q_4\bar{\omega}^\top \bar{\omega} - \frac{1}{2}\bar{q}^\top \dot{\bar{\omega}} \\ &= -\frac{1}{2}\bar{q}^\top \dot{\bar{\omega}} - \frac{1}{4}q_4\|\bar{\omega}\|_2^2\end{aligned}\tag{A.23}$$

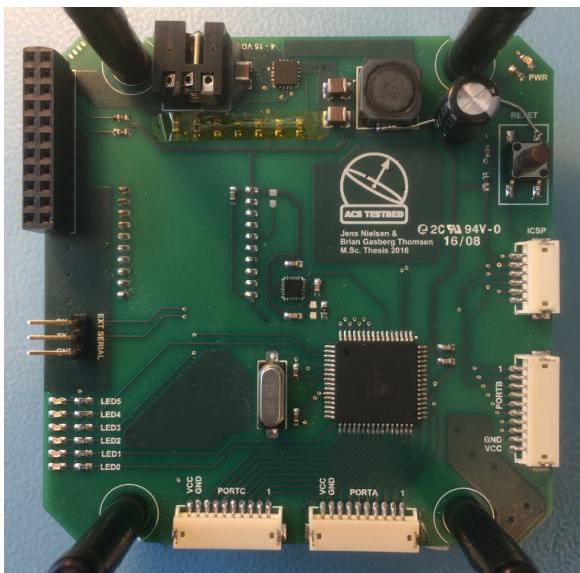
The quaternion acceleration is now represented as a function of the dynamic equation and the current orientation

$$\ddot{\mathbf{q}} = \begin{bmatrix} \ddot{\bar{q}} \\ \ddot{q}_4 \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{T}}(\mathbf{q})\dot{\bar{\omega}} - \frac{1}{4}\bar{q}\|\bar{\omega}\|_2^2 \\ -\frac{1}{2}\bar{q}^\top \dot{\bar{\omega}} - \frac{1}{4}q_4\|\bar{\omega}\|_2^2 \end{bmatrix}\tag{A.24}$$

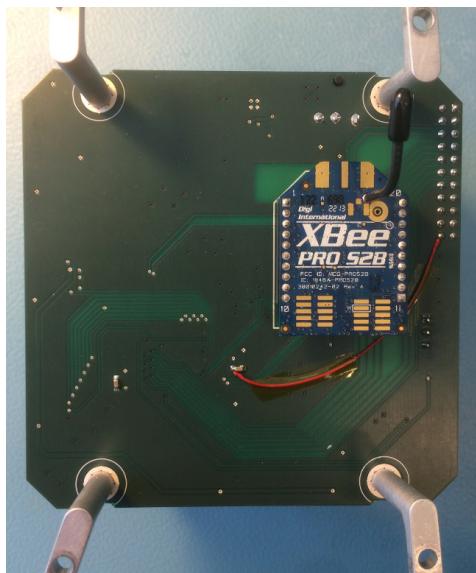
# Testbed Electronics

## B.1 Main Board

In the following, pictures and schematics of the main bord is shown.

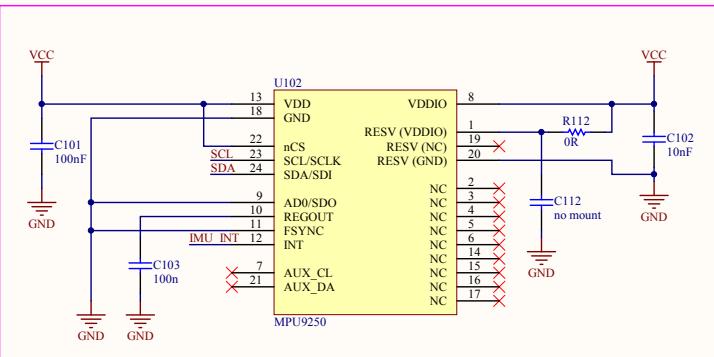
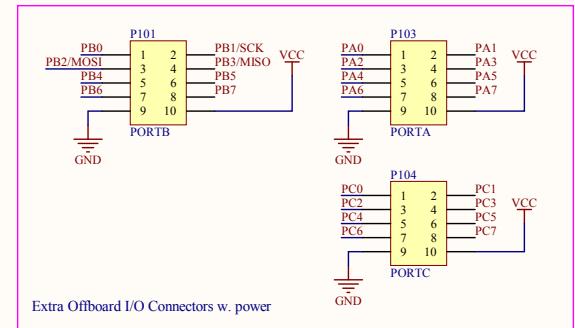
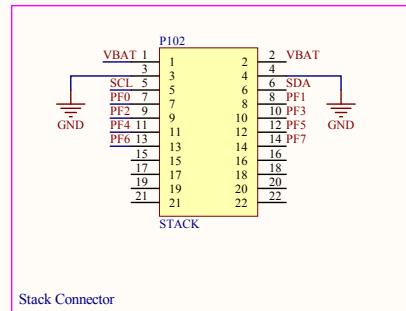


**Figure B.1:** Main board top.

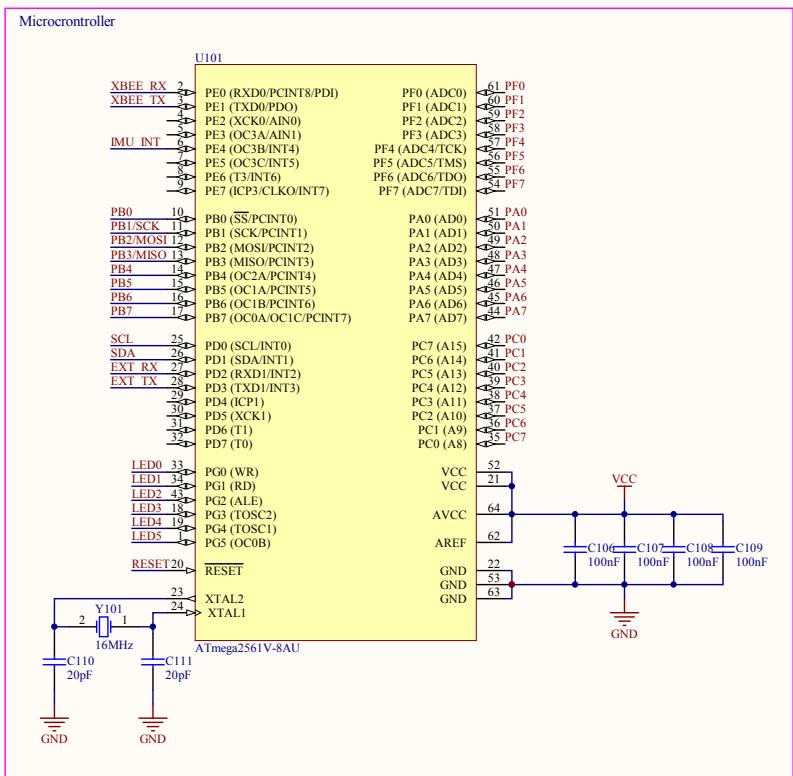
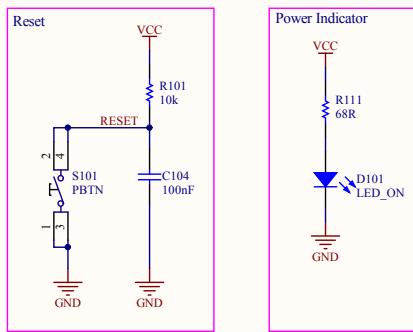


**Figure B.2:** Main board bottom.

A



B



A

A

B

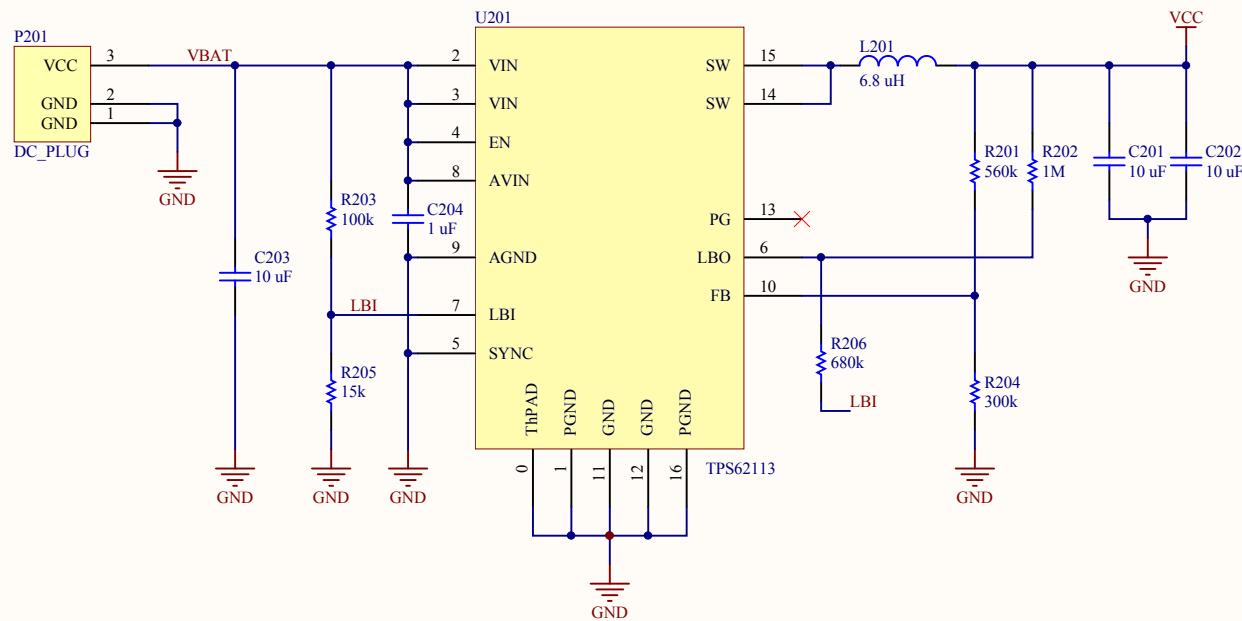
B

C

C

D

D



$$V_{cc} = 3.3 \text{ Volt}$$

$$\text{Undervoltage protection cut out (VtripL)} = 9.5 \text{ Volt}$$

$$\text{Undervoltage protection cut in (VtripH)} = 10 \text{ Volt}$$

Governing equations:

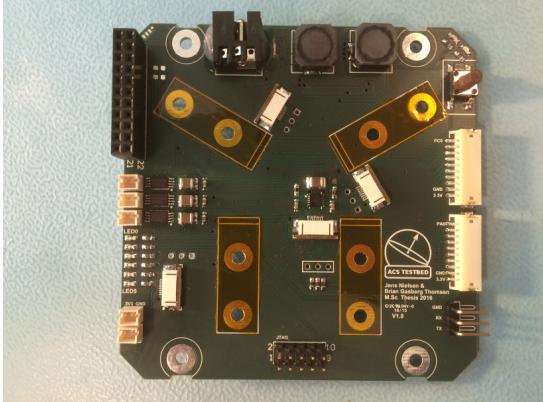
$$(VtripL - 1.256)/R203 + (Vout - 1.256)/(R202 + R206) = 1.256/R205$$

$$1.281/R205 + 1.281/R206 = (VtripH - 1.281)/R203$$

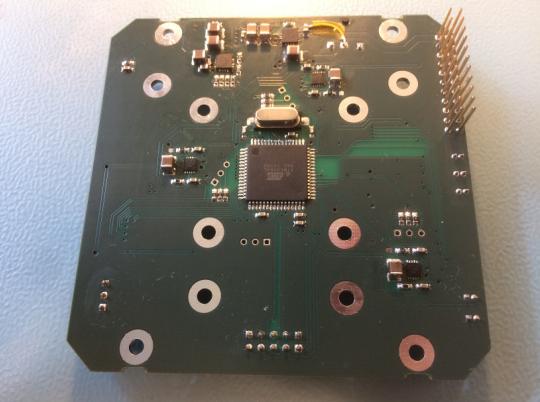
Title		
Size	Number	Revision
A4	Power Regulation Unit	
Date: 09-04-2016	Sheet 2 of 2	
File: \\pru.SchDoc	Drawn By:	

## B.2 Actuator Board

In the following, pictures and schematics of the actuator bord is shown.



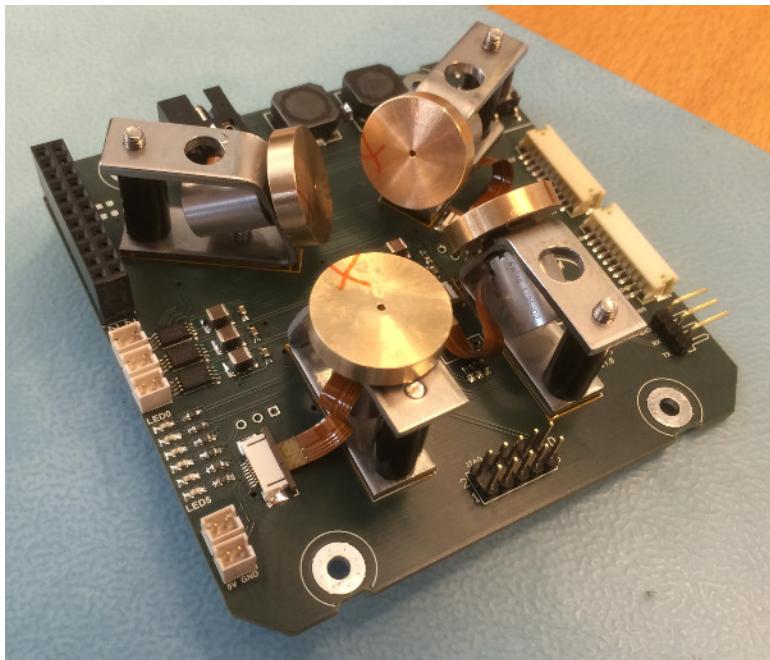
**Figure B.3:** Actuator board top.



**Figure B.4:** Actuator board bottom.

In Figure B.3 the top side of the actuator board is shown without the reaction wheels mounted. The four pieces of kapton tape attached to the top side is to protect the soldermask top from being damaged by the reaction wheel mounts when they are screwed on to the PCB. The stack connector is in the top left side of the PCB; below this is the connections for the magnetic torquers, six LEDs for informational purpose and to auxillary power supply outputs - one 5 volt and one 3.3 volt. At the bottom middle is the JTAG programming connector for the MCU. On the right side of the PCB - from the top; reset button, two connectors for two unused port of the MCU and a external serial port. At the top of the PCB is battery connection point and the coils used in the PCU.

In Figure B.4 the bottom side of the actuator board PCB is shown. At the top right, the stack connector can be seen. At the top the two switch-mode regulators of the PCU is located. At the middle of the PCB the MCU is located.

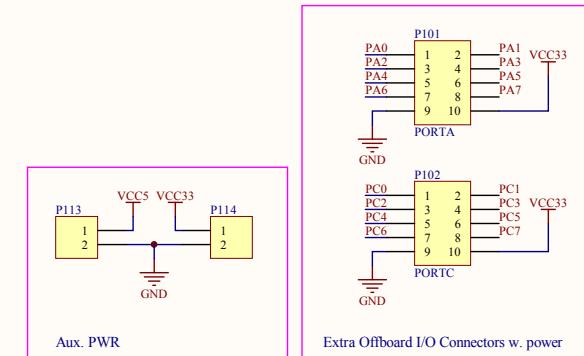
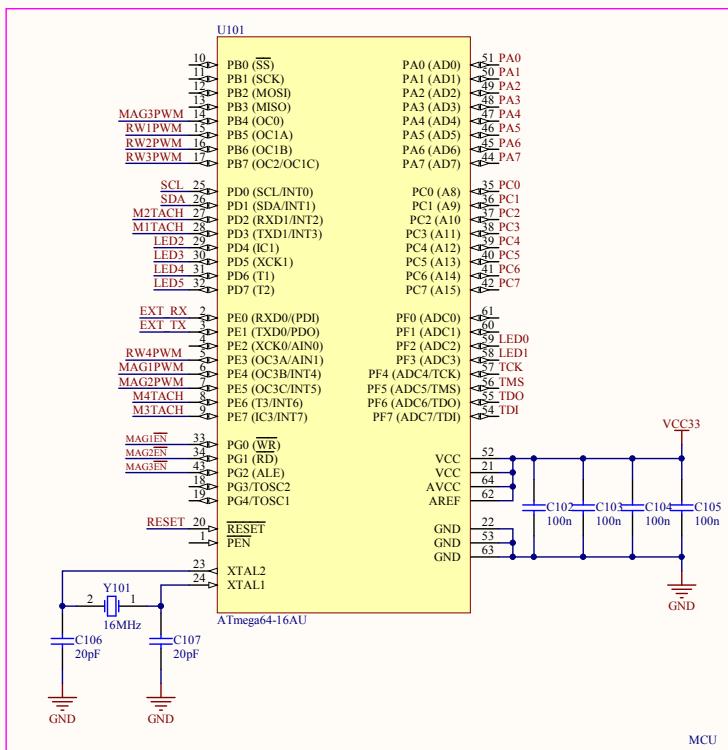
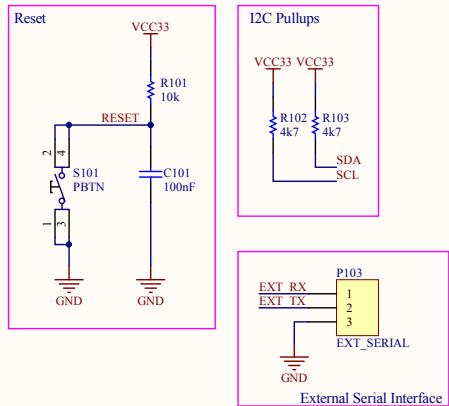


**Figure B.5:** Actuator board assembled.

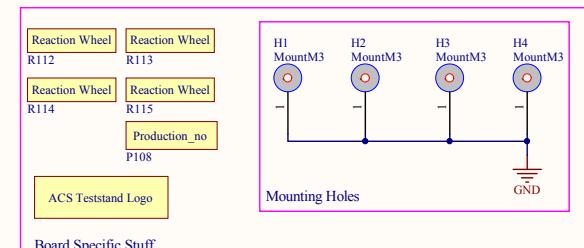
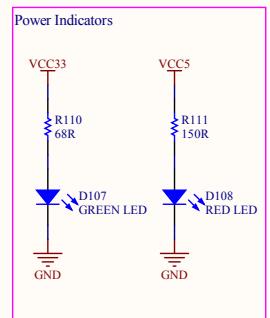
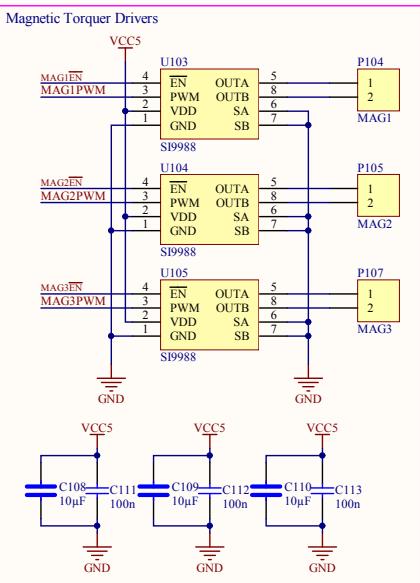
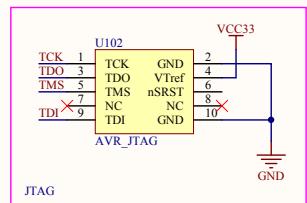
The final assembly of the actuator board is shown in Figure B.5, ready for integration with the main board.

The schematics is shown in the next three pages, the schematics and the PCB layout are designet in Altium Designer.

A



B



Title Actuator Board		
Size A3	Number MCU	Revision
Date: 24-04-2016		Sheet 1 of 3
File: \\actuator board.SchDoc		Drawn By:

C

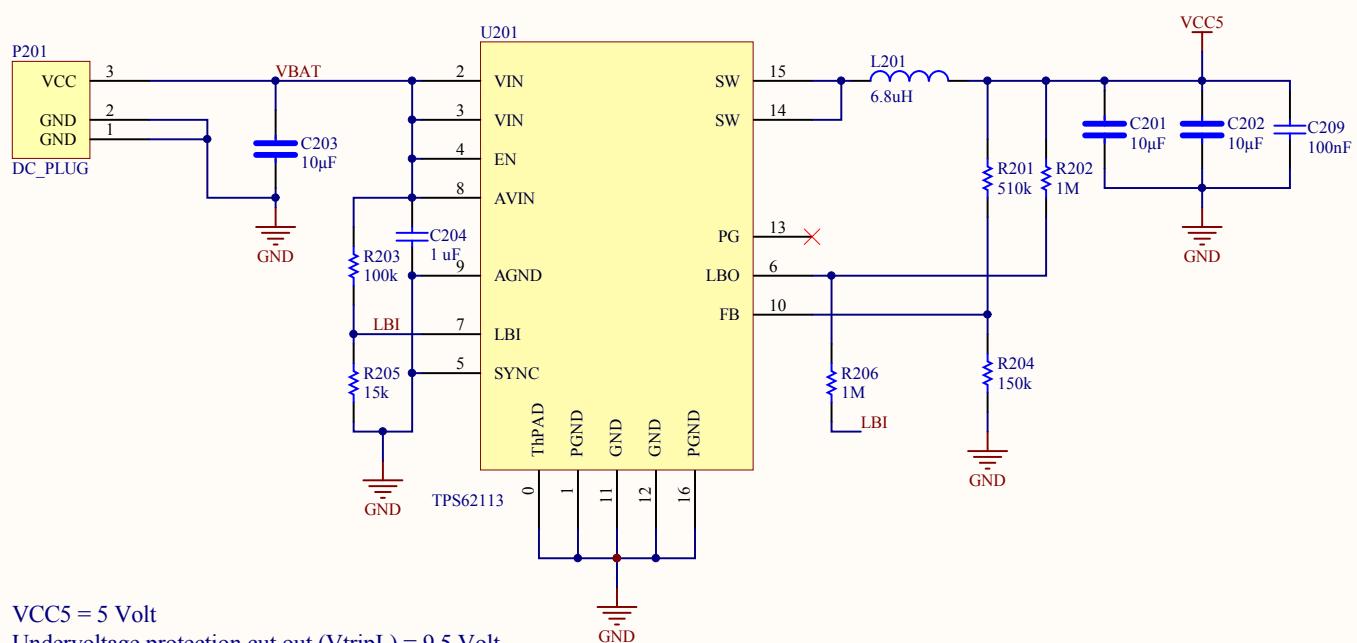
4

5

7

8

A



VCC5 = 5 Volt

Undervoltage protection cut out (VtripL) = 9.5 Volt

Undervoltage protection cut in (VtripH) = 10 Volt

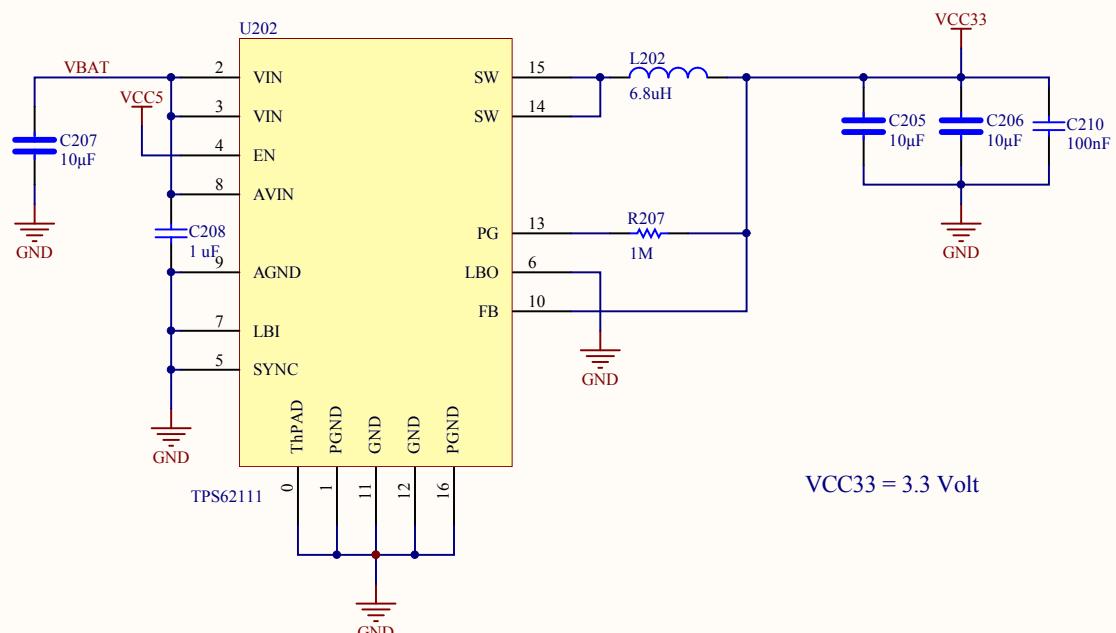
A

B

B

C

C



VCC33 = 3.3 Volt

D

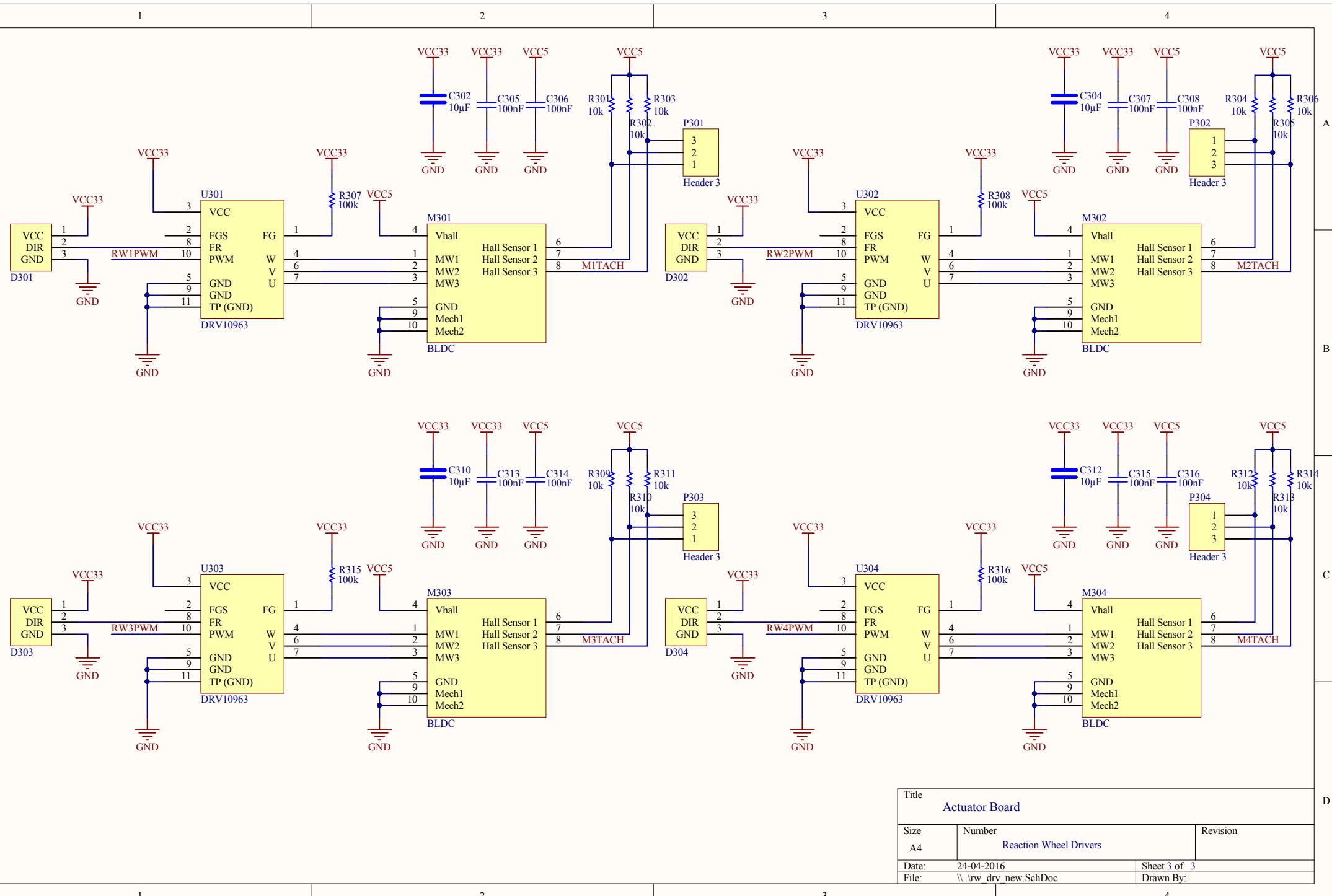
D

Title Actuator Board		
-------------------------	--	--

Size A4	Number Power Regulation Unit	Revision
------------	---------------------------------	----------

Date: 22-05-2016	Sheet 2 of 3
------------------	--------------

File: \pru.SchDoc	Drawn By:
-------------------	-----------



---

## Attachments

A snapshot of the attitude control testbed hardware and software is attached in the zip-file accompanying this thesis. The zip-file includes:

- `acs_testbed`
- `space_simulator`
- links to the mechanical hardware design.

In the following a short description the three parts of the attachment is given.

**acs\_testbed:** The `acs_testbed` folder holds a snapshot of the repository containing:  
the electronic hardware diagrams with PCB layouts and the software written for  
the embedded microcontrollers for both the `main_board` and the `actuator_board`.

**space\_simulator:** The `space_simulator` folder holds the complete space simulator  
used in this thesis. A `readme` file in the root of the attachment file explains how  
to get this up and running.

**mechanical drawings:** The mechanical design is performed by using Onshape, which  
is an online 3D CAD software. The links leads to the 3D CAD models of the  
attitude control testbed.