



TITLE: Attitude Control system for AAU CubeSat
THEME: Autonomous and reliables systems
PERIOD: 4th of September 2001 to 6th of June 2002
GROUP: IAS-1031

AUTHOR:

Torben Graversen

Michael Kvist Frederiksen

Søren Vejlgaard Vedstesen

SUPERVISOR:

Rafael Wisniewski

NUMBER OF COPIES: 9
NUMBER OF PAGES: 112 pages
HANDED IN: 6th of june 2002

Synopsis

This thesis deals with the analysis and design of the attitude control system for the AAU CubeSat.

The goal is to design both a controller and actuators, which will match the physical and electrical constraint given by the size of the satellite.

First an appropriate actuator configuration, must be found and implemented. This includes interface circuitry, and a micro processor which can handle the control algorithms.

Two controllers must be design, one to detumble the satellite after separation from the launch vehicle, and one which can stabilise the satellite at a give reference.

To accomplish this, a mathematical model of the system is derived. For detumbling a B-dot controller is derived, tested and implemented. For normal operation two controllers will be derived and tested. First a constant gain controller based on periodic optimal control is setup. Then a controller based on LMI is derived. Both controllers are tested for both nadir and inertial pointing stability. The costant gain controller is found to have the best overall performance and is therefore implemented.

Preface

This master thesis documents the work in the specialisation of Intelligent Autonomous Systems at the Department of Control Engineering, Institute of Electronic System, Aalborg University. The thesis was initiated September 4th, 2001 and ended June 6th 2002. This master's work was supervised by Rafael Wisniewski.

The work presented in the master is part of the AAU CubeSat, and is focused on the analysis and design of the attitude control system. The work includes development of attitude hardware and control algorithms.

Throughout the work figures and table are numbered consecutively according to the chapters. References to literature is made in the following way [Wertz, 1978], where [Wertz, 1978] is found in the bibliography. Some general notations used throughout the thesis are listed on page 1.

Aalborg university, June 2002

Torben Graversen

Michael Kvist Frederiksen

Søren Vejlgård Vedstesen

Contents

1	Introduction	3
1.1	Choice of mission	3
1.2	Launch and orbit	4
2	Mission analysis	5
2.1	Mission statement	5
2.2	Life cycle of satellite	5
2.3	Required subsystems	6
2.4	Mission modes	7
2.5	Subsystem interaction	8
3	Control system analysis	9
3.1	Stabilization methods	9
3.2	AAU CubeSat actuator configuration	11
3.3	Pointing accuracy	11
4	Design requirements	13
4.1	Detumbling	13
4.2	Pointing accuracy and slew	13
4.3	Mass budget	14
4.4	Power budget	14
4.5	Magnetorquers	14
4.6	Outline of report	15
5	Modelling of satellite	17
5.1	Coordinate systems	17
5.2	Controllability of a magnetically actuated satellite	18
5.3	Equations of motion	18
5.4	Linearization	21
6	Hardware design	27
6.1	Magnetorquer design	27
6.2	Coil driver	36
6.3	Current sensor	39

7	Controller design	41
7.1	Detumbling Controller	41
7.2	Periodic linear systems	43
7.3	Controllability analysis	48
7.4	Constant gain controller	49
7.5	LMI controller	51
8	Implementation and test	57
8.1	B-dot controller	57
8.2	Constant gain	58
8.3	LMI controller	64
8.4	Test of final controller	66
9	Software design	69
9.1	Interface overview	69
9.2	Mode switching	70
9.3	I ² C communication	70
9.4	Implementation of software	76
9.5	Test of implemented software	80
10	Conclusion	83
	Appendix	87
A	Momentum wheel	87
B	Description of external disturbance torques	91
C	Quaternions	96
D	Hardware schematic diagram	99
E	Printed circuit board layout	101
F	Manufacturing of magnetorquers	104
G	NORAD Lines and the SGP4 orbit Model	107
	Bibliography	112

Notations and constants

Vector and matrices

A, v matrices and vector are written in bold face,
 ${}^c\mathbf{v}, {}^o\mathbf{v}, {}^i\mathbf{v}$, vector \mathbf{v} resolved in control CS, orbit CS or inertial CS respectively,
 v_o^x, v_o^y, v_o^z x, y and z components of vector \mathbf{v}_o .

List of symbols

Ω_{ci} angular velocity of control CS w.r.t. inertial CS,
 Ω_{co} angular velocity of control CS w.r.t. orbit CS,
 Ω_{oi} angular velocity of orbit CS w.r.t. inertial CS,
 ${}^c\mathbf{q}$ attitude quaternion representing rotation of control CS w.r.t orbit CS,
 \mathbf{q}, q_4 vector and scalar part of ${}^c\mathbf{q}$,
 $\mathbf{A}({}^c\mathbf{q})$ rotation matrix from orbit CS to control CS (direction cosine),
 $\delta\Omega$ small pertubation of the angular velocity,
 $\delta\mathbf{q}$ small pertubation of the attitude quaternion,
 \mathbf{I} inertia tensor of the satellite,
 I_x, I_y, I_z moments of inertial about x-, y- and z- principal axis,
 \mathbf{N}_{ctrl} control torque,
 \mathbf{N}_{gg} gravity gradient torque,
 \mathbf{N}_{dist} disturbance torque,
 \mathbf{m} magnetic moment generated by magnetorquers,
 \mathbf{B} magnetic field of the earth,
 ψ, θ, ϕ Yaw, roll and pitch angles respectively,
 \mathbf{E} Identity matrix

Constants

μ Earth gravitational constant = $3.986 \cdot 10^{14} [m^3/s^2]$
 I_x $1.983221309 \cdot 10^{-3} [kgm^2]$
 I_y $2.100895792 \cdot 10^{-3} [kgm^2]$
 I_z $1.882314372 \cdot 10^{-3} [kgm^2]$

chapter 1

Introduction

The CubeSat concept, developed by Stanford University, is making it possible to develop a small satellite and to launch it into space at low cost. The concept specifies that the satellite must be $10\text{ cm} \times 10\text{ cm} \times 10\text{ cm}$ and weighing no more than 1 kg. This makes it possible to launch up to three CubeSats at a time, from a standard deployment mechanism known as a P-POD (Poly Pico-satellite Orbital Deployer). In this way considerations regarding launch and deployment from the launch vehicle, is indeed reduced for the CubeSat developers. Multiple P-PODs can easily be stacked together and many CubeSats can be deployed in a single launch. The cost for launching P-PODs is shared between the CubeSats. The cost is also lowered by launching the P-PODs as secondary payloads in launch vehicles. Secondary payloads are added to launches where a primary payload does not use all the available space and mass on the launch vehicle.

The development of a CubeSat fits ideally to the project based education form used at Aalborg University. This has resulted in a project being launched which involves groups of students from different departments of AAU.

1.1 Choice of mission

The Department of control engineering at AAU has already had experience with developing satellites by involvement in the development of both Ørsted and Rømer spacecraft. These two satellites fall into the category of micro-satellites. The CubeSat is a pico-satellite. A micro satellite has a maximum weight of 100 kg, and a pico-satellite has a maximum weight of 10 kg. The size of a CubeSat gives the advantage of low launch cost, but at the same time it constrains the design. The constraints in the design makes it challenging to fit all the necessary hardware, such as on-board computer, power supply, communication system, attitude control system and payload into the satellite. Depending on the mission the space in the satellite needed for payload varies.

In order to secure a high level of reliability in expensive satellites, it is normal to use components which have been well tested for use in space. However, this being a low cost satellite, some experimentation is done in the choice of components. This is the case with the payload. The satellite is going to photograph earth from a low earth orbit (LEO) at approximately 700 km height. For this a CMOS camera is going to be used. This will take color images in the visible light spectrum. The images will be transmitted to a ground station and made public via the Internet. People using the Internet will have the possibility to request the satellite to take a

photo of a specific geographical location. The scope of the photo coverage will be restricted to Denmark. The purpose of this mission is to increase public interest in space science, technology and natural science in general. The satellite can show that it is possible to monitor or take images of the earth from LEO with a small pico-satellite like the CubeSat.

1.2 Launch and orbit

The launch of the CubeSats is planned by Stanford University in collaboration with the CubeSat launch company. The launch of the first 18 CubeSats is scheduled to be with a Dnepr-rocket from Baikonur Kosmodrom in Kazakhstan in November 2002.

The AAU CubeSat is going to be launched in spring 2003, also from Baikonur Kosmodrom. The orbit is going to be a Sun-Synchronous circular orbit with a local time of 1.30.

chapter 2

Mission analysis

This chapter describes the mission analysis. First the mission statement for the satellite is described. After this the CubeSats life cycle is described and then the subsystems needed to achieve a robust design of the satellite. Then in section 2.4 the modes related to the mission state is described, and finally the interaction between the subsystems is described.

2.1 Mission statement

The mission statement is based on the mission description in chapter 1.

Primary mission: To show capability of designing and building a spacecraft and hereby gaining experience in designing small satellites in particular.

Secondary missions :

- To communicate with the satellite while it is in orbit, in order to receive housekeeping data from the on-board subsystems. Hereby getting knowledge on how well each subsystem works.
- To take images of the earth and transmit them to a ground station. Hereby testing the attitude control and CMOS camera i LEO.
- To allow people outside the university via the Internet to choose a geographical site to be photographed, and later retrieve the photo from the Internet. Hereby increasing public interest in space science, technology and natural science in general.

2.2 Life cycle of satellite

Figure 2.1 shows the CubeSat's life cycle from launch to early mission. After launch and deployment the satellite will most likely be tumbling. This is stopped by using an attitude control mode called detumbling. In situations like the one right after launch, where the satellite has no attitude information available, the satellite uses a basic beacon. The basic beacon is a periodic signal, which is transmitted, so the ground station can detect the satellite and transmit new telecommands.

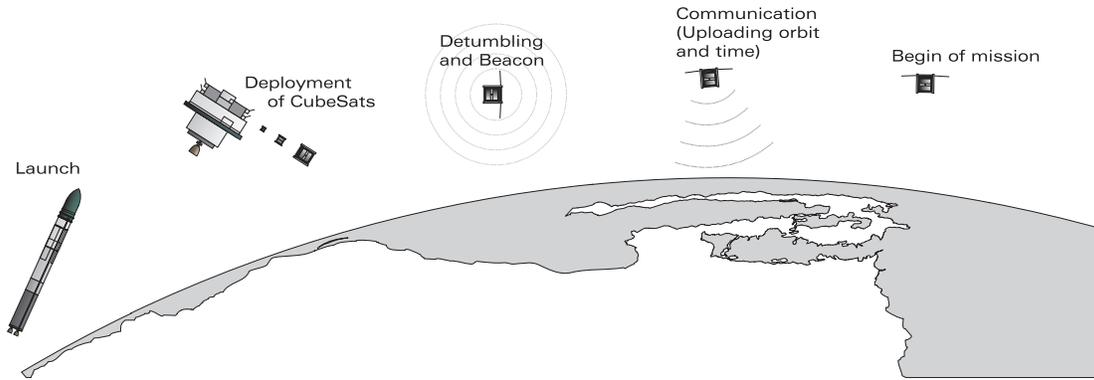


Figure 2.1: The satellite is launched and deployed together with other cube satellites, it detumbles and then receives necessary data from the ground station for the mission to be carried out.

When the satellite has been in mission for period of time, it will deteriorate and stop functioning. Finally the gravitational force from Earth will drag it into the atmosphere, where the satellite will burn.

2.3 Required subsystems

Designing an entire satellite is a complicated task, because of this the satellite is split in to smaller and more manageable subsystems. Each subsystem can then be designed and constructed in parallel by different groups. The subsystems in the AAU CubeSat are; a power supply unit (PSU), a communication system (COM), an onboard computer (OBC), a command and data handling system (CDHS), an attitude determination and control system (ADCS), and the payload, which in this case is the CMOS camera.

The responsibilities of each subsystem is briefly described below. A detailed description of all subsystem is not provided, as the scope of this project is only the ADCS subsystem.

- PSU** The main responsibilities of the PSU is to make sure that there is generated power, and to store the power. In addition it must make sure that there is always power available for minimum functionality of the satellite. The PSU should therefore on command from the CDHS subsystem, switch on and off the other subsystem, or if the power consumption of the individual subsystem is too high.
- COM** The primary responsibility of the COM system is to transmit telemetry to the ground station and to receive telecommands from it. In addition it must be able send out a basic beacon, to enable tracking of the satellite from the ground station.
- OBC** The OBC servers as the platform for the CDHS. It must provide enough memory to store one image taken with the CMOS camera, and control the communication between other subsystems.

CDHS The CDHS runs the operating system on the OBC, and controls all the other subsystems, and keeps track of a flight plan uploaded from the ground station. This flight plan decides when a picture is to be taken.

ADCS The responsibility of this system is to keep the attitude of the satellite within a certain range, of a setpoint decided by the current operating mode.

Payload the CMOS camera must take an image, when it is requested by the CDHS.

2.4 Mission modes

The mission requirements and limited amount of power available makes it desirable to define different operation modes. Some modes are given in the description of the satellite life cycle in section 2.2. Others are related to the amount of power available.

The modes defined for the AAU CubeSat are described below. The modes are internal to the ADCS system, and the description is therefore not general for the entire satellite:

Fail safe mode without actuation, this mode is the boot mode for the ACS system. All systems will stay in this mode until the first communication with the ground station is completed.

Detumbling mode is only used after the satellite has been deployed from the P-POD, and systems has been initialised, initialisation includes upload of orbit parameters and time synchronisation with the ground station.

Power save mode is the normal operation mode, where the pointing accuracy is lowered to save power.

Camera mode is used when an image has to be taken. The satellite is switched into this mode in time to perform the needed slew maneuver to achieve the desired pointing accuracy.

The switching between Power save and Camera mode is also shown in figure 2.2. The dark shaded area is the time where the satellite is in camera mode, and the light shaded area is the time when the satellite is in power save mode.

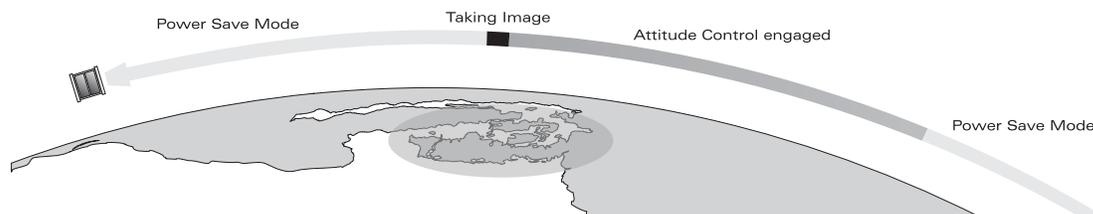


Figure 2.2: Mode switch between camera mode and power save mode.

2.5 Subsystem interaction

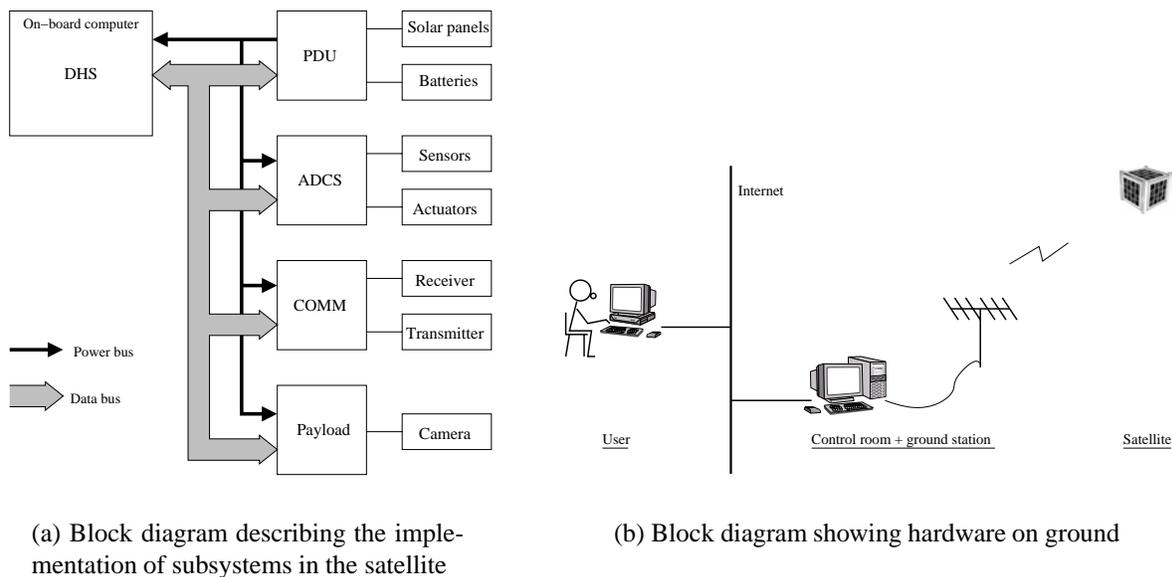
The interaction between subsystems is divided into two kind of interactions, the power bus and a data bus.

The power bus is controlled by the PSU. The PSU controls the power consumption of each subsystem. If a predefined maximum is exceeded, the system is shut down, and rebooted. The CDHS system can indirectly control the power bus, by sending a message to the PSU about which subsystem needs to be rebooted. This message is transmitted via the data bus.

The data bus is controlled by the CDHS/OBC and is implemented as a master/slave system, based on the I²C bus. On the data bus different message between the subsystems can be transmitted. The house keeping data is also collected via the data bus.

How the subsystems are connected to the power and data bus is shown in figure 2.3(a). In figure 2.3(b) the external data communication is shown.

A user connected to the internet, can connect to the control room with the ground station, and thereby get the data communicated between satellite and ground station.



(a) Block diagram describing the implementation of subsystems in the satellite

(b) Block diagram showing hardware on ground

Figure 2.3: Figures show how complete system divided into satellite and ground hardware

chapter 3

Control system analysis

The previous analysis took the whole satellite into account. In the following analysis of the control system, only the ACS system will be considered. First a discussion of passive and active stabilization methods is given. After that the actuator configuration and properties of these are described for the AAU CubeSat. Finally the pointing accuracies in each mode are defined.

3.1 Stabilization methods

There are different methods used in attitude control, some are passive and determined by the physical properties of the satellite, and others are active such as magnetorquers and momentum wheels.

3.1.1 Passive stabilization methods

The passive methods which has been chosen to take into account are: gravity gradient, spin stabilisation and passive magnets. A common feature of all passive stabilization is that the satellite will only be stabilized in two axes.

Gravity gradient The gravity gradient uses the earth gravitational force to stabilize the satellite. The property of this kind of stabilization is the position of the centre of gravity of the satellite.

Spin stabilisation A spin stabilized satellite is passively stable when the satellite is spinning around the axis with the largest momentum of inertia.

Passive magnets Pasive magnet will stabilize the satellite with the earth magnetic field, and the movement of the satellite will follow the magnetic field.

Depending on the position of the centre of gravity, the gravity gradient might not have a stabilizing effect. In the CubeSat requirements from CalPol, it states that the centre of gravity must not be more than 2 cm of centre [Conoly, 2000]. This could be used to distribute the momentum of inertia in a way which would make the gravity gradient a stabilising factor.

The spin stabilization is dependent on the distribution of the momentum of inertial. Ideally only the inertia tensor would have values different from zero, and these values should further more be distinctly different from one another. This would make it easier to control the satellite,

as a torque applied in one axis will have no effect on the others. In the design the spin axis has to be defined, and the mass must be placed mainly on the two other axes. Any other distribution will act as a disturbing effect rather than a stabilizing effect.

Passive magnets will align the satellite to the earth magnetic field, however this is not desirable for most control purposes, since the satellite will only point with sufficient accuracy over the poles. Further more it will make it much harder to use a magnetometer for attitude determination, as the magnetic field from the passive magnets will be much stronger than the earth magnetic field, and thereby possibly saturate the magnetic sensors.

For the CubeSat it is most likely that the use of passive magnets will not be an option. The passive magnets is not an option because a magnetometer will be used as a sensor. The spin stabilizing effect would be preferable to use, as it can provide a passive stability in the axis where the camera is placed. Usually a spin rate of 30 - 90 rpm is used to stabilize a satellite, however in order not to distort the image, the rotational speed would have to be as low as 2 - 3 rpm, and that would have practically no stabilizing effect. The gravity gradient will also be an option with a proper design of the distribution of the mass in the satellite.

3.1.2 Active stabilization methods

The active methods taken into account are magnetorquers and momentum wheels. Active stabilization makes it possible to control the satellite in three axes. Each of the actuators only have an effect in one axis, therefore 3 independent actuators are needed for three axis control. Three types of actuators are commonly used: magnetorquers, momentum wheels and thrusters. Since thrusters are much too heavy to be considered for a CubeSat mission, they will not be discussed further.

Magnetorquers A magnetorquer is a wire coil used to generate an electro magnetic field. This field interacts with the earth magnetic field generating a torque to reorient the satellite. The generated torques are largest when the magnetorquer and the earth magnetic field are parallel, and they have no effect when they are perpendicular.

Momentum wheel Momentum wheel have the nice property that no information about the magnetic field is needed in the control. The stabilizing effect from a momentum wheel is like the passive spin stabilization, however it is possible to control the torque generated by the momentum wheel. It is only possible to store a limited amount of torque in a wheel, which make it necessary also to have either thrusters or magnetorquers for momentum dumping.

Dual Spin Dual Spin stabilization is a derivative of the previously mentioned passive spin stabilization. In dual spin only part the satellite is spinning as opposed to passive spin stabilization, where the entire satellite is spinning. Dual spin can be accomplished by either making part of the satellite spin and keeping the payload platform stationary or by placing a momentum wheel along the axis, that is to be stabilized.

3.2 AAU CubeSat actuator configuration

The choice of actuators for the CubeSat is somewhat special, as the mass and power budgets are very low. With this limitation it would be desirable to use some kind of passive stabilisation, could be achieved if the design is carried out with this in mind.

For the CubeSat three magnetorquers have been chosen as actuators. The possibility of using momentum wheels for 3 axis stabilization, has been considered, but is not an option, because some kind of momentum dumping is needed, and the only way of doing that is by using magnetorquers. This means that no matter what configuration is chosen three magnetorquers are required. The added weight of three momentum wheels would make the system too heavy. Furthermore finding space inside the CubeSat for placing three momentum wheels is very difficult, since they have to be placed orthogonally to one another in order to minimize power consumption and simplify the control.

Therefore using only one momentum wheel mounted in the same axis as the camera has been considered to gain extra pointing accuracy by using dual spin stabilization. The extra weight, that the wheel and the mounting of it adds, seems to be a good compromise between weight and accuracy.

However designing a momentum wheel that would fit inside the CubeSat proved to be a major problem. Initially it was considered using a small electric motor to spin a metal disc. The idea was eventually discarded after a meeting with the structural design team, where it was revealed that there was no space available on the camera axis. A new design was then considered. Here the rotor of the motor was designed as a hoop and placed around the camera. The rotor hoop was designed to have sufficient mass to act as a momentum wheel. The rotor consisted of permanent magnets and would be driven in the same way as a stepping motor. However magnetic shielding would ensure the functionality of the magnetometer, the design was deemed too heavy by the structural design team and it was decided only to use three magnetorquers. The design considerations for the momentum wheel can be found in appendix A.

3.3 Pointing accuracy

The pointing accuracy for the CubeSat will be calculated, based on the fact that the images taken, are restricted to pictures of Denmark.

It has also been chosen that the area, the image is going to cover is approximately a square of 100×100 km, and the maximum deviation from the desired centre of the image to the actual centre of approximately 100 km.

With these restrictions in mind the maximum allowable pointing error for a pass directly over Denmark can be calculated (equation 3.1).

$$error = \arctan\left(\frac{100}{700}\right) = 8.13^\circ \quad (3.1)$$

This pointing accuracy is only valid for a pass directly over Denmark, but such a pass will seldom occur, therefore it has to be calculated how close the satellite needs to pass by Denmark,

to get an usable image. To find this distance it is defined that the width of the image must be at maximum the width of Denmark, measured from the most western point to the most eastern point. This width have been found to approximately 300 km [Kraak, 2002]. The distance can now be calculated, assuming that the camera has the same field of view (FOV). Based on the calculated distance, the pointing accuracy can be calculated, still assuming that the centre of the image must be maximum 100 km off target. This has all been illustrated in figure 3.1, and the calculations are given in equation 3.2 – 3.4.

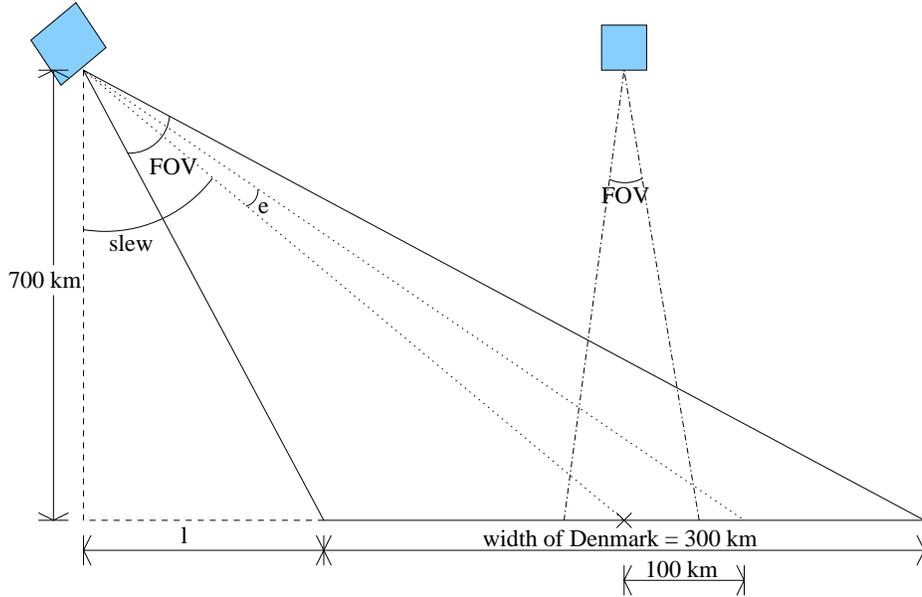


Figure 3.1: Calculation of maximum distance from satellite to centre of picture, and worst case pointing accuracy.

$$\frac{\sqrt{700^2 + l^2}}{\sin\left(\tan^{-1}\left(\frac{700}{l+150}\right)\right)} = \frac{150}{\sin\left(\frac{FOV}{2}\right)} \Rightarrow l = 921.546 \text{ km} \quad (3.2)$$

$$slew = \tan^{-1}\left(\frac{l}{700}\right) + \frac{FOV}{2} = 56.844^\circ \quad (3.3)$$

$$e = \tan^{-1}\left(\frac{l + 150 + 100}{700}\right) - slew = 2.297^\circ \quad (3.4)$$

From the results the maximum slew angle is determined, and the pointing accuracy is found at the maximum distance. These results will be used as design guidelines in the control design. The actual achievable pointing accuracy will be found during the design and test of the controllers. The angular velocity about the camera axis is required to be less than 3 RPM (0.314 rad/s), since higher velocities will distort the image.

chapter 4

Design requirements

In this chapter design requirements will be presented for the ACS system. Mass and power budget will be set up for the entire ADCS system. Finally an outline of the report is given.

4.1 Detumbling

The detumbling controller must be able to slow down the satellite after release from the launch vehicle. It must therefore be able to handle initial velocities of up to 0.1 rad/s on all three axes. The controller should also be able to detumble the satellite within 1-2 orbits. The satellite is considered detumbled when it follows the magnetic field of the earth. The criteria for detumbling can be calculated by assuming that the earth's magnetic field is a dipole and that the satellite is a powerful magnet. When the angular velocity of this magnet corresponds to the rate of change of the earth's magnetic field, the satellite is detumbled. The magnetic field changes direction 2 times with one orbit and the orbit time is approx. 6000 s given the altitude of 700 km. The criteria with a extra margin of 50% become:

$$rotation = \frac{2 \cdot 2\pi}{6000} \cdot 150\% = 0.003 \text{ rad/s} \quad (4.1)$$

4.2 Pointing accuracy and slew

The requirements regarding pointing accuracy and slew, are the ones calculated in section 3.3 and table 4.1. These values will be used as guide lines in the control design. The actual achievable accuracy will be influenced by the mass and power budget, and the space available for the magnetorquers in the structural design. These limitations will be presented in the following section.

	Min	Max
Pointing accuracy	8.13°	2.29°
Slew angle	0°	56.84°

Table 4.1: Pointing accuracy and slew requirements.

4.3 Mass budget

The mass budget is based on information from the structural design team, where the maximum mass is given to 150 g. The mass budget for the entire ACS system is given in table 4.2.

Magnetorquers	70 g
- mounting	5 g
Sensors	5 g
- mounting	10 g
PCB + components	50 g
Wiring	10 g
Total	150 g

Table 4.2: Mass budget for the ACS system.

4.4 Power budget

The power budget is based on information from the power supply design team, where a maximum power consumption is given. The maximum power consumption is given to 450 mW.

The power budget for the entire ACS system is given in table 4.3.

	Min	Max
Magnetorquer	0 mW	375 mW
- loss in driver	5 mW	5 mW
Sensors + interface	0 mW	20 mW
Micro controller	50 mW	50 mW
Total	55 mW	450 mW

Table 4.3: Power budget for the ACS system.

4.5 Magnetorquers

The requirements to the magnetorquers are affected by many factors, the available power, the maximum weight and the position in the structure.

The position of the magnetorquers in the structure gives some physical constraints, which can not be changed. The size of the magnetorquers is shown in figure 4.1.

Based on the size given in figure 4.1, the maximum mass given in table 4.2 and the maximum power consumption given in table 4.3, the design of the magnetorquers must be optimised considering all the constraints.

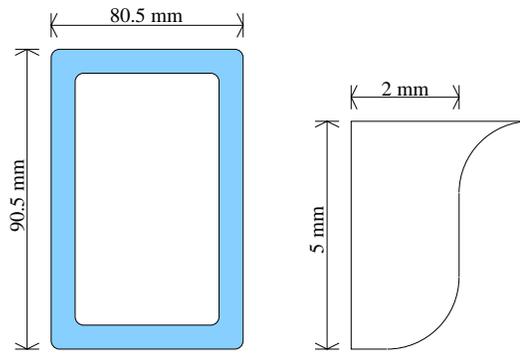


Figure 4.1: Size of magnetorquers.

4.6 Outline of report

To achieve the design requirements, magnetorquers and interface hardware has to be build, software for the OBC and the ACS micro controller must be designed and implemented, and a mathematical model of the CubeSat has to be made for use in the control design.

First the mathematical model will be presented, to give some insight into the dynamics and kinematics of the satellite.

The hardware design will then be prsented. The design is done in cooperation with the ADS group, as the two system are going to be fitted onto the same PCB. The driver circuit for the magnetorquers will be designed and implemented according to the design requirements.

In the controller design a constant gain and a LMI based controller will be tested, to see which is the more suitable for the CubeSat. This will be tested taking pointing accuracy, power consumption and capability to handle actuator saturation into account. Based on the controller analysis, the best one is chosen and implemented.

Last the software developed for the ACS micro controller is described. The software for the OBC will not be described in detail. However the communication between between the PIC and the OBC will be described in detail.

chapter 5

Modelling of satellite

This chapter describes the satellite equation of motion. First the coordinate systems used throughout the project are defined, and then the model is derived. The model will be based on rigid body dynamics and kinematics. The equations will be derived based on quaternion representation, which is described in appendix C. First the equations of motion will be presented, and these will then be linearised for use in the control design. Two models linearized models will be derived one linearised about a Nadir pointing reference, and on about an inertial pointing reference.

5.1 Coordinate systems

The coordinate systems (CS) used in the mathematical model and the control design are shown in figure 5.1, and defined below.

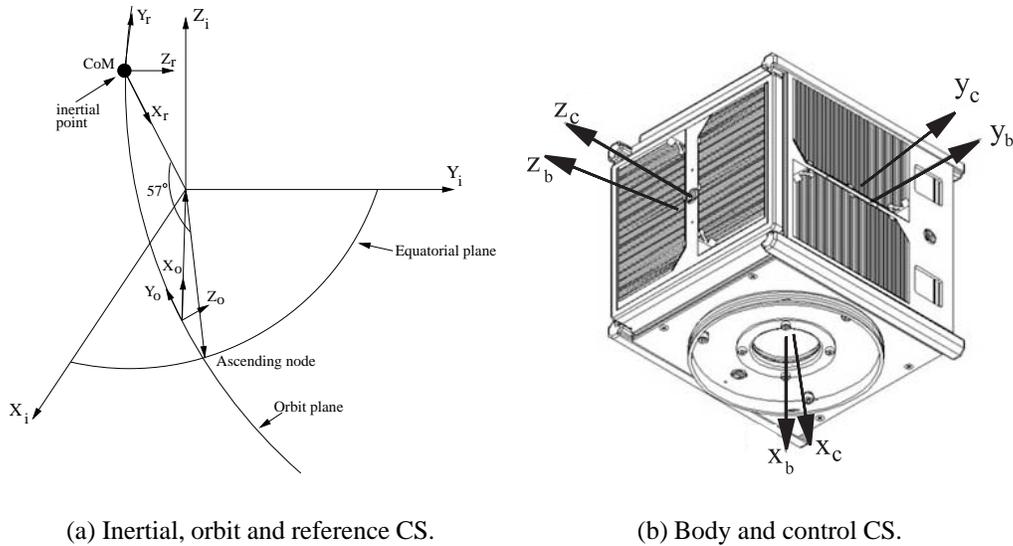
Control CS This CS is a right orthogonal CS, built on the principal axes of inertia, and with the origin placed in the centre of mass (CoM). The axes are defined as shown in figure 5.1(b).

Body CS This CS is a right orthogonal CS, built on the geometric axes, and with origin placed in the CoM. The x axis is defined as the camera axis, and the y and z axis are defined as shown in figure 5.1(b).

Orbit CS This CS is a right orthogonal CS, with the origin defined in the CoM of the satellite. The x axis points towards the centre of the Earth, the z axis is in the orbital plane normal direction, and the y axis coincides with the orbital angular velocity. The orbit CS is shown in figure 5.1(a).

Reference CS This CS is a right orthogonal CS, with the origin in the satellite CoM. The CS is rotated with respect to an inertial point, defined as a rotation about the z axis, so that the x axis points toward Nadir at 57°N . The CS is shown in figure 5.1(a) (Inertial fixed CS).

Inertial CS This CS is an inertial right orthogonal CS, with its origin at the Earth's centre of mass. The z axis is parallel to the Earth rotation axis and points toward the north pole. The x axis is parallel to the line connection the centre of the Earth with Vernal Equinox and points towards Vernal Equinox. The CS is shown in figure 5.1(a).



(a) Inertial, orbit and reference CS.

(b) Body and control CS.

Figure 5.1: Coordinate systems used in the project.

5.2 Controllability of a magnetically actuated satellite

A magnetically actuated satellite has a limitation in control, as it is only possible to control the satellite in two axes at a time. This is due to the torque produced by the interaction of the geomagnetic field and the magnetic field produced by the magnetorquers. The generated torque is always perpendicular to the geomagnetic field. Thus one axis will always be parallel to the geomagnetic field, and thereby not controllable.

The geomagnetic field changes its orientation when the satellite changes its position in orbit. For a Nadir pointing satellite the magnetic field over 16 orbits is shown in figure 5.2(a), and the changes seen from the reference CS are shown in figure 5.2(b).

From figure 5.2 it can be seen that the geomagnetic field changes direction once with a Nadir pointing, and twice for the inertial pointing model. What is also seen in both figures is the daily variation of the geomagnetic field due to the rotation of the Earth.

5.3 Equations of motion

The mathematical model of the satellite is based on the dynamic and kinematic equations of motion. The dynamics relates the torques acting on the satellite's angular velocity in the inertial CS. The kinematic is expressed as the integration of the angular velocity. Throughout the report the attitude is parameterised by four components of a quaternion, but the results are presented as Euler angles (ϕ, θ and ψ).

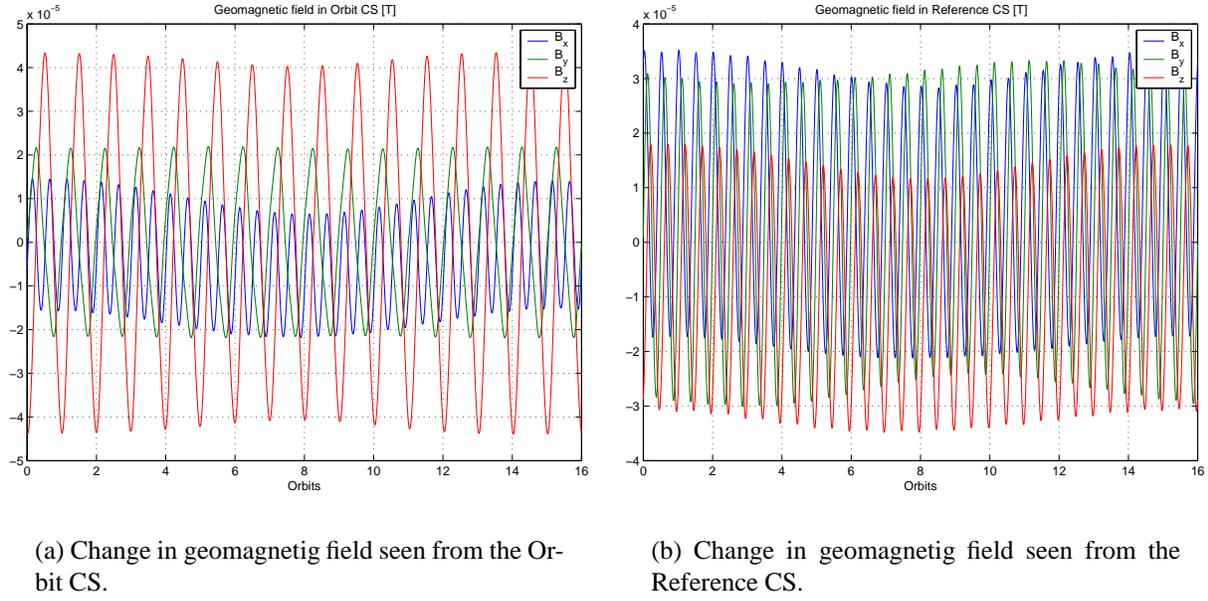


Figure 5.2: Change in geomagnetic field in the orbit.

5.3.1 Dynamics

The dynamics of a rigid satellite in Low earth orbit is [Wertz, 1978]

$$\mathbf{I}^c \dot{\boldsymbol{\Omega}}_{ci}(t) = -{}^c\boldsymbol{\Omega}_{ci}(t) \times \mathbf{I}^c \boldsymbol{\Omega}_{ci}(t) + {}^c\mathbf{N}_{ctrl}(t) + {}^c\mathbf{N}_{gg}(t) + {}^c\mathbf{N}_{dist}(t) \quad (5.1)$$

The cross product ${}^c\boldsymbol{\Omega}_{ci}(t) \times \mathbf{I}^c \boldsymbol{\Omega}_{ci}(t)$ arises due to the fact that the dynamics are described in a rotating CS (control CS), in the inertial CS.

The control torque for one coil is obtained as

$${}^c\mathbf{N}_{ctrl}(t) = {}^c\mathbf{m}(t) \times {}^c\mathbf{B}(t) \quad (5.2)$$

and the magnetic moment as [Serway, 1982]

$$m(t) = n_{coil} i_{coil}(t) A_{coil} \quad (5.3)$$

The coils are placed perpendicular to the x, y and z axis of the body CS, therefore to obtain the control torque the rotation from the body CS to the control CS is necessary, this is given by

$${}^c\mathbf{m}(t) = {}^c_b\mathbf{A}^b \mathbf{m}(t) \quad (5.4)$$

The gravity gradient torque ($\mathbf{N}_{gg}(t)$) is given as (see appendix B)

$$\mathbf{N}_{gg}(t) = \frac{3\mu}{R_{cm}^3} \left({}^c\hat{\mathbf{R}}_{cm} \times \mathbf{I} {}^c\hat{\mathbf{R}}_{cm} \right) \quad (5.5)$$

where

μ Earth gravitational constant

R_{cm} distance from the centre of the Earth to the satellite CoG

${}^c\hat{\mathbf{R}}_{cm}$ is the zenith

The zenith is equivalent to the unit vector ${}^c\mathbf{i}_o$ and the constant $\frac{\mu}{R_{cm}^3} = \omega_0^2$, where ω_0^2 is the orbital rate. This is only true for a circular orbit, but for eccentricities close to zero periodic variations of the orbital rate are small and can be neglected. Given this the gravity gradient torque may be written as in equation 5.6.

$$\mathbf{N}_{gg} = 3\omega_0^2 ({}^c\mathbf{i}_o \times \mathbf{I} {}^c\mathbf{i}_o) \quad (5.6)$$

The disturbance torque ($\mathbf{N}_{dist}(t)$) are presented in appendix B.

5.3.2 Kinematics

The kinematics of the satellite describes the orientation in space, and is obtained by the integration of the angular velocity. The kinematic equations are expressed as separate integrations of the vector and the scalar part of the attitude quaternion [Wertz, 1978]

$$\begin{aligned} \dot{\mathbf{q}} &= \frac{1}{2} {}^c\boldsymbol{\Omega}_{cref} q_4 - \frac{1}{2} {}^c\boldsymbol{\Omega}_{cref} \times \mathbf{q} \\ \dot{q}_4 &= -\frac{1}{2} {}^c\boldsymbol{\Omega}_{cref} \mathbf{q} \end{aligned} \quad (5.7)$$

By extension of the three dimension angular velocity vector (${}^c\boldsymbol{\Omega}_{co}$), to a four dimensional vector $[{}^c\boldsymbol{\Omega}_{co}^T 0]^T$, the kinematics may be expressed as equation 5.8, with $\mathbf{R}(-)$ defined as in appendix C.

$${}^c_{ref}\dot{\mathbf{q}} = \frac{1}{2} \mathbf{R} \left({}^c\tilde{\boldsymbol{\Omega}}_{cref} \right) {}^c_{ref}\mathbf{q} \quad (5.8)$$

5.4 Linearization

The linearization of the dynamic and kinematic equations of motion is done for two different operating points. One is based on an inertial pointing satellite, and the other is based on a Nadir pointing satellite.

The reference for the inertial pointing model is the reference CS, which is fixed in the inertial CS. The Nadir pointing based model has a reference defined as the orbit CS. In both cases the angular velocity of the satellite w.r.t. the reference is zero, and the attitude quaternion is such that the control CS is coincident with the reference.

Both models are considered to be within the neighbourhood of the reference, where the angular velocity of the satellite w.r.t. the reference is zero (${}^c\boldsymbol{\Omega}_{cref} = \mathbf{0}$), and attitude is such that the control CS coincides with the reference (${}^c_{ref}\mathbf{q} = [0 \ 0 \ 0 \ 1]^T$)

5.4.1 Inertial pointing model

The reference CS for the inertial pointing model can be described as 3 successive rotation about the x_i, y_i and z_i axis. These rotations are presented as a quaternion describing the rotation from the inertial CS to the reference CS given by equation 5.9.

$${}^i_r\mathbf{q} = R(\mathbf{q}_z(\psi)) (R(\mathbf{q}_y(\theta)) \mathbf{q}_z(\phi)) \quad (5.9)$$

$$= \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\phi}{2}\right) \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\phi}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\phi}{2}\right) \cos\left(\frac{\psi}{2}\right) - \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\phi}{2}\right) \sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\phi}{2}\right) \sin\left(\frac{\psi}{2}\right) - \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\phi}{2}\right) \cos\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\phi}{2}\right) \cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\phi}{2}\right) \sin\left(\frac{\psi}{2}\right) \end{bmatrix} \quad (5.10)$$

Linearized dynamics

The dynamics are given by equation 5.1. The cross coupling of the angular velocities of the control CS w.r.t. the inertial CS, disappears as the control CS is not initially rotating.

The gravity gradient will be a periodic function of the angle from the ascending node to the position of the satellite in orbit. This is described by the quaternion (${}^o_i\mathbf{q}(t, P)^*$), which is defined as in equation 5.11.

$${}^o_i\mathbf{q}(t, P)^* = \begin{bmatrix} {}^o_i q_1(t, P) \\ {}^o_i q_2(t, P) \\ {}^o_i q_3(t, P) \\ -{}^o_i q_4(t, P) \end{bmatrix} = {}^o_o\mathbf{q}(t, P) = \begin{bmatrix} \sin\left(\frac{i}{2}\right) \cos\left(\frac{\Omega-u(t,P)}{2}\right) \\ \sin\left(\frac{i}{2}\right) \sin\left(\frac{\Omega-u(t,P)}{2}\right) \\ \cos\left(\frac{i}{2}\right) \sin\left(\frac{\Omega-u(t,P)}{2}\right) \\ -\cos\left(\frac{i}{2}\right) \cos\left(\frac{\Omega-u(t,P)}{2}\right) \end{bmatrix} \quad (5.11)$$

where

i inclination

Ω right ascension of the ascending node

$$u(t, P) = \omega_0 \text{mod}(t, P)$$

$$\omega_0 = \sqrt{\frac{\mu}{a^3}}$$

P orbital period

a semimajor axis

The quaternion from equation 5.11 describes the rotation of the orbit CS w.r.t. the inertial CS. For the gravity gradient torque we need the rotation of the orbit CS w.r.t. the control CS, this can be found from equation 5.12.

$${}^c\mathbf{q}(t, P) = R({}^c\mathbf{q}) {}^o\mathbf{q}(t, P) \quad (5.12)$$

where

$${}^c\mathbf{q} = [\delta q_1 \ \delta q_2 \ \delta q_3 \ 1]^T$$

From the quaternion ${}^c\mathbf{q}(t, P)$ the unit vector ${}^c\mathbf{i}_o$ can be setup, from the definition in appendix C.

$${}^c\mathbf{i}_o = \begin{bmatrix} {}^c q_1^2 - {}^c q_2^2 - {}^c q_3^2 + {}^c q_4^2 \\ 2({}^c q_1 {}^c q_2 - {}^c q_3 {}^c q_4) \\ 2({}^c q_1 {}^c q_3 + {}^c q_2 {}^c q_4) \end{bmatrix} \quad (5.13)$$

Inserting the values from equation 5.13 in equation 5.6, and substituting the time varying part with $K_{(1-12)}(t, P)$ the gravity gradient torque may be expressed as

$${}^cN_{gg} = \begin{bmatrix} 3\omega_0 (I_z - I_y) (K_1(t, P)\delta q_1 + K_2(t, P)\delta q_2 + K_3(t, P)\delta q_3 + K_4(t, P)) \\ 3\omega_0 (I_x - I_z) (K_5(t, P)\delta q_1 + K_6(t, P)\delta q_2 + K_7(t, P)\delta q_3 + K_8(t, P)) \\ 3\omega_0 (I_y - I_x) (K_9(t, P)\delta q_1 + K_{10}(t, P)\delta q_2 + K_{11}(t, P)\delta q_3 + K_{12}(t, P)) \end{bmatrix} \quad (5.14)$$

The control torque (${}^c\mathbf{N}_{ctrl}$) is given by equation 5.2. The control torque can be changed to minimise the power consumption. This is due to that any magnetic moment generated parallel to the geomagnetic field has no influence on the satellite motion. This can also be expressed as

$${}^c\mathbf{N}_{ctrl}(t) = ({}^c\mathbf{m}_{||}(t) + {}^c\mathbf{m}_{\perp}(t)) \times {}^c\mathbf{B} = {}^c\mathbf{m}_{\perp}(t) \times {}^c\mathbf{B} \quad (5.15)$$

where ${}^c\mathbf{m}_{||}$ is the component of ${}^c\mathbf{B}$ parallel to ${}^c\mathbf{B}$, and ${}^c\mathbf{m}_{\perp}$ is the perpendicular part.

The control torque only including the perpendicular part of the magnetic moment, may be found by including an extra cross product of the geomagnetic field as in equation 5.16.

$${}^c\mathbf{N}_{ctrl} = ({}^c\mathbf{m} \times {}^c\mathbf{B}) \times {}^c\mathbf{B} \quad (5.16)$$

This expression of the control applies both in the inertial pointing model and the Nadir pointing model.

Linearized kinematics

The kinematics is given by equation 5.7, and the linearized kinematics may be written as

$$\frac{d}{dt} {}^c\mathbf{q} = \frac{1}{2} {}^c\boldsymbol{\Omega}_{ci} - \frac{1}{2} {}^c\boldsymbol{\Omega}_{ci} \times {}^c\mathbf{q} \quad (5.17)$$

where ${}^c\boldsymbol{\Omega}_{ci} = \delta\boldsymbol{\Omega}$ describes the small perturbation of the angular velocities from the reference. And ${}^c\mathbf{q} = R({}^r\mathbf{q}) \delta\mathbf{q}$ describes the small perturbation of the attitude quaternion from the reference.

Then the linearised kinematics may be written as equation 5.18.

$$\frac{d}{dt} {}^c\mathbf{q} = \frac{1}{2} \begin{bmatrix} \delta\omega_x q_{r4} \\ \delta\omega_y q_{r4} \\ \delta\omega_z q_{r4} \end{bmatrix} - \frac{1}{2} \begin{bmatrix} -\delta\omega_y q_{r3} + \delta\omega_z q_{r2} \\ \delta\omega_x q_{r3} - \delta\omega_z q_{r1} \\ -\delta\omega_x q_{r2} + \delta\omega_y q_{r1} \end{bmatrix} \quad (5.18)$$

5.4.2 Nadir pointing model

The reference CS for the Nadir pointing model is the orbit CS. The orbit CS is a rotation frame, as it can be seen from equation 5.19 and 5.20. These equations describe the angular velocity of the satellite w.r.t. the orbit and inertial CS respectively.

$${}^c\boldsymbol{\Omega}_{co} = {}^c\boldsymbol{\Omega}_{ci} - \omega_0 {}^c\mathbf{k}_o \quad (5.19)$$

$${}^c\boldsymbol{\Omega}_{ci} = {}^c\mathbf{A} [0 \ 0 \ \omega_0]^T + \delta\boldsymbol{\Omega} \quad (5.20)$$

Linearized dynamics

The dynamic equation of motion is divided into the cross coupling, the gravity gradient and the control torque.

The linearized cross coupling is given by equation 5.21.

$$-({}^c\boldsymbol{\Omega}_{ci} \times \mathbf{I}^c\boldsymbol{\Omega}_{ci}) = \begin{bmatrix} (I_y - I_z) {}^c\omega_{ciy} {}^c\omega_{ciz} \\ (I_z - I_x) {}^c\omega_{ciz} {}^c\omega_{cix} \\ (I_x - I_y) {}^c\omega_{cix} {}^c\omega_{ciy} \end{bmatrix} \quad (5.21)$$

inserting the values of ${}^c\boldsymbol{\Omega}_{ci}$ from equation 5.20 into equation 5.21, the cross coupling becomes

$$-({}^c\boldsymbol{\Omega}_{ci} \times \mathbf{I}^c\boldsymbol{\Omega}_{ci}) \approx \begin{bmatrix} (I_y - I_x) \omega_0 \delta\omega_y \\ (I_z - I_x) \omega_0 \delta\omega_x \\ 0 \end{bmatrix} \quad (5.22)$$

The linearised gravity gradient is given by

$${}^c\mathbf{N}_{gg} = 3\omega_0^2 \left(\begin{bmatrix} 1 \\ 2\delta q_3 \\ 2\delta q_2 \end{bmatrix} \times \begin{bmatrix} I_x \\ 2I_y\delta q_3 \\ 2I_z\delta q_2 \end{bmatrix} \right) = 6\omega_0 \begin{bmatrix} 0 \\ [I_x - I_z] \delta q_2 \\ [I_y - I_x] \delta q_3 \end{bmatrix} \quad (5.23)$$

Linearized kinematics

From equation 5.7 the kinematics are given. From equation 5.20 and 5.19 the linearization of the kinematics may be written as in equation 5.24.

$$\frac{d}{dt}\delta\mathbf{q} = \frac{1}{2}\delta\boldsymbol{\Omega} - \frac{1}{2}[0 \ 0 \ \omega_0]^T - \delta\mathbf{q} \times [0 \ 0 \ \omega_0]^T = \frac{1}{2} \begin{bmatrix} \delta\omega_x - 2\delta q_1\omega_0 \\ \delta\omega_y + \delta q_2\omega_0 \\ \delta\omega_z - \omega_0 \end{bmatrix} \quad (5.24)$$

5.4.3 Linearized equations of motion

The linearized dynamics and kinematics may be written in matrix form as

$$\frac{d}{dt} \begin{bmatrix} \delta\boldsymbol{\Omega} \\ \delta\mathbf{q} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \delta\boldsymbol{\Omega} \\ \delta\mathbf{q} \end{bmatrix} + \mathbf{B}(t) {}^c\mathbf{m} + \begin{bmatrix} \mathbf{I}^{-1} \\ \mathbf{0} \end{bmatrix} \mathbf{N}_{gg} + \begin{bmatrix} \mathbf{I}^{-1} \\ \mathbf{0} \end{bmatrix} \mathbf{N}_{dist} \quad (5.25)$$

The $\mathbf{B}(t)$ matrix is the same for both models, but the \mathbf{A} matrix is different. The matrix related to the inertial pointing model will have the subscript I, and the matrix related to the Nadir pointing model will have the subscript N.

$$\mathbf{A}_{\mathbf{I}}(t, P) = \begin{bmatrix} 0 & 0 & 0 & 3\omega_0^2\sigma_x K_1(t, P) & 3\omega_0^2\sigma_x K_2(t, P) & 3\omega_0^2\sigma_x K_3(t, P) \\ 0 & 0 & 0 & 3\omega_0^2\sigma_y K_5(t, P) & 3\omega_0^2\sigma_y K_6(t, P) & 3\omega_0^2\sigma_y K_7(t, P) \\ 0 & 0 & 0 & 3\omega_0^2\sigma_z K_9(t, P) & 3\omega_0^2\sigma_z K_{10}(t, P) & 3\omega_0^2\sigma_z K_{11}(t, P) \\ \frac{1}{2}q_{r4} & -\frac{1}{2}q_{r3} & \frac{1}{2}q_{r2} & 0 & 0 & 0 \\ \frac{1}{2}q_{r3} & \frac{1}{2}q_{r4} & -\frac{1}{2}q_{r1} & 0 & 0 & 0 \\ -\frac{1}{2}q_{r2} & \frac{1}{2}q_{r1} & \frac{1}{2}q_{r4} & 0 & 0 & 0 \end{bmatrix} \quad (5.26)$$

$$\mathbf{A}_N = \begin{bmatrix} 0 & \omega_0 \sigma_y & 0 & 0 & 0 & 0 \\ \omega_0 \sigma_x & 0 & 0 & 0 & 6\omega_0 \sigma_y & 0 \\ 0 & 0 & 0 & 0 & 0 & 6\omega_0 \sigma_z \\ \frac{1}{2} & 0 & 0 & -\omega_0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & \omega_0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \end{bmatrix} \quad (5.27)$$

where

$$\sigma_x = \frac{I_y - I_z}{I_x} \quad \sigma_y = \frac{I_z - I_x}{I_y} \quad \sigma_z = \frac{I_x - I_y}{I_z} \quad (5.28)$$

$$\mathbf{B}(t) = \left[\mathbf{I}^{-1} \begin{bmatrix} -B_y^2(t) - B_z^2(t) & B_x(t)B_y(t) & B_x(t)B_z(t) \\ B_x(t)B_y(t) & -B_x^2(t) - B_z^2(t) & B_y(t)B_z(t) \\ B_x(t)B_z(t) & B_y(t)B_z(t) & -B_x^2(t) - B_y^2(t) \end{bmatrix} \right] \quad (5.29)$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

chapter 6

Hardware design

The design of the hardware is based on the design requirements in chapter 4. The limiting factors for the hardware are the power and space available in the CubeSat. The hardware is made in cooperation with the ADS group, because the circuitry is to be mounted on the same print layout, and some of the components is shared like the PIC micro processor.

6.1 Magnetorquer design

The design of the magnetorquers is based on the assumption the all three coil have the same properties. This means that the mass of each coil is equal to one third of the total mass.

The coils will be positioned around the solar panels, on the outside of the satellite as shown in figure 6.1. This position put some constraints on the design. All design parameters are presented in the following.

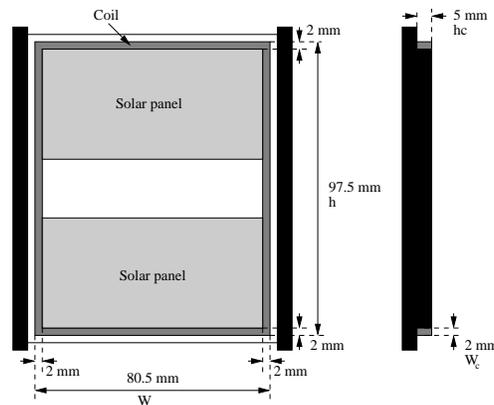


Figure 6.1: Position of coils, on side panel of satellite.

6.1.1 Design parameters

The physical constraints for the coil design such as mass, height, width and temperature are presented in table 6.1.

Parameter	Symbol	Value	Unit
Maximum width	w	80.5	[mm]
Maximum height	h	97.5	[mm]
Maximum cross sectional width	w_c	2	[mm]
Maximum cross sectional heigh	h_c	5	[mm]
Mean area of coil	A	7496	[mm ²]
Cross sectional area	A_{cmax}	10	[mm ²]
Circumference	C	356	[mm]
Total mass of 3 coils incl. insulation	M_{total}	60	g
Mass of one coil	M_c	20	g
Minimum temperature	T_{min}	-100	[°C]
Normal temperature	T_{norm}	15	[°C]
Maximum temperature	T_{max}	100	[°C]

Table 6.1: Physical design dimensions.

Parameter	Symbol	Value (cu)	Value (al)	Unit
Wire material density	ρ_x	8.93e3	2.7e3	[kg/m ³]
Wire material resistivity	σ_x	1.55e-8	2.5e-8	[Ωm]
Temperature coefficient of resistivity	$\alpha_{0(x)}$	3.9e-3	3.9e-3	[K ⁻¹]

Table 6.2: Wire data for cubber and aluminium. The x subscript in the symbol can be al for aluminium or cu for cubber.

Two materials will be considered in the design process. The materials are copper and aluminium, and the data needed in the design process is shown in table 6.2.

From the power budget and the design of the coil driver, the electric design parameters are determined. An overview of the parameters are shown in table 6.3.

Parameter	Symbol	Value	Unit
Maximum power loss in current generator	$P_{gen-100^\circ C}$	3.20	[mW]
Normal power loss in current generator	$P_{gen15^\circ C}$	2.19	[mW]
Minimum power loss in current generator	$P_{gen100^\circ C}$	1.93	[mW]
Maximum power used by one coil	$P_{-100^\circ C}$	214.70	[mW]
Normal power used by one coil	$P_{15^\circ C}$	122.98	[mW]
Minimum power used by one coil	$P_{100^\circ C}$	93.08	[mW]
Minimum power supply voltage	U_{power}	4.9	[V]
Voltage drop in current generator	U_{gen}	0.11	[mV]
Voltage supply to coils	U_{coil}	± 4.79	[V]
Maximum voltage over measurement resistor	$U_{meas-100^\circ C}$	± 4.963	[mV]
Normal voltage over measurement resistor	$U_{meas15^\circ C}$	± 2.8053	[mV]
Minimum voltage over measurement resistor	$U_{meas100^\circ C}$	± 2.1771	[mV]
Measurement resistor	R_{meas}	0.09	[Ω]

Table 6.3: Electric design parameters.

6.1.2 Basic equations

The coil design is based on four equations, one for mass, one for power dissipation, one for the coil resistance and one for the producible magnetic moment.

The mass of one coil can be determined by

$$M_c = nC a_w \rho, \quad (6.1)$$

where

M_c Mass of one coil

n Number of turns

C Circumference

a_w Wire cross sectional wire

ρ Wire material density

The power dissipation in one coil can be determined by

$$P = U_{coil} I = I^2 R, \quad (6.2)$$

where

- P Dissipated power
 U_{coil} Voltages supplied to coil
 I Current in coil
 R Resistance of the coil

The coil resistance is given by

$$R = \frac{nC\sigma(T)}{a_w}, \sigma(T) = \sigma_0(1 + \alpha_0 T), \quad (6.3)$$

where

- n Number of turns
 C Circumference
 $\sigma(T)$ Wire material resistivity with temperature coefficient α_0
 a_w Wire cross sectional area

The magnetic moment for one coil can be determined by

$$m = nIA, \quad (6.4)$$

where

- m Producible magnetic moment
 n Number of turns
 I Current in coil
 A Area of coil

6.1.3 Coil design

Each coil is designed according to the previously mentioned design parameters. The main factors are mass (M_c), the minimum voltage and power requirements. This enables the calculation of the coil resistance, and thereby the wire cross sectional area, which can be translated into a wire cross sectional area of the wires available.

Magnetic moment

From equation 6.1 – 6.4 the relation between mass, power and magnetic moment can be found to

$$M_c = \left(\frac{mC}{A} \right)^2 \frac{\rho\sigma}{P} \quad (6.5)$$

The goal is to optimise the efficiency of the coil, with respect to mass and power consumption. Rewriting equation 6.5 the producible magnetic moment is

$$m = \frac{A}{C} \sqrt{\frac{MP}{\sigma\rho}} \quad (6.6)$$

To maximise the producible magnetic moment (m), the product $\rho\sigma$ has to be minimised. The $\rho\sigma$ product can be minimised by the choice of material. The values can be found in table 6.2. From the table the preferable material is aluminium.

Wire cross sectional area

The minimum wire cross sectional area is dependent on the wire resistance, minimum coil voltage, the power and the number of turns.

The resistance of the coil is given by equation 6.3, and is also related to the minimum coil voltage:

$$R_{max} = \frac{U_{coil,min}^2}{P} \quad (6.7)$$

Equalling equation 6.3 and 6.7 gives

$$\frac{U_{coil,min}^2}{P} = \frac{nC\rho_{max}}{a_{w,min}} \quad (6.8)$$

Substituting the number of turns derived from equation 6.1 into equation 6.8, the minimum wire cross sectional area is

$$a_{w,min} = \frac{1}{U_{coil,min}} \sqrt{\frac{M_c P \sigma_{max}}{\rho}} \quad (6.9)$$

The minimum wire cross sectional area must be compared to the available wire data. For copper the minimum wire diameter is 0.02 mm, and for aluminium the minimum wire diameter is 1.1 mm. Based on this information only copper wire is an option. A list of the interesting data for the copper wires is given in table 6.4.

From the chosen wire material and available wire diameter, the number of turns in each coil can be found

$$n = \frac{M}{C a_{w,avail} \rho} \quad (6.10)$$

Wire diameter [mm]	Area [mm ²]	Insulation [mm]	Diameter incl. insulation [mm]	Filling factor
0.08	0.005027	0.015	0.105	12000
0.085	0.005675	0.016	0.111	11000
0.9	0.006362	0.017	0.117	10000
0.095	0.007088	0.017	0.123	9000
0.10	0.00785	0.018	0.129	8000
0.106	0.00823	0.018	0.137	7100
0.112	0.00985	0.019	0.143	6400
0.118	0.01023	0.02	0.15	5900
0.125	0.01227	0.021	0.159	5200
0.13	0.01327	0.022	0.165	5000
0.132	0.01368	0.022	0.167	4700
0.14	0.01539	0.023	0.176	4200
0.15	0.01767	0.024	0.188	3800
0.16	0.02011	0.024	0.199	3300
0.17	0.0227	0.025	0.211	3000
0.18	0.02545	0.026	0.222	2700
0.19	0.02835	0.026	0.233	2400
0.2	0.03142	0.027	0.23	2300

Table 6.4: Data for available cubber wire.

6.1.4 Design results

The calculation above was carried out, and it was found that aluminium was the more suitable material, but the available wires did not allow that aluminium was used. This can be seen from the design results in table 6.5.

Parameter for one coil	Symbol	Value	Unit
Wire diameter without insulation	d_w	0.13	[mm]
Number of turns	n	415	
Mass of each coil	M	18	[g]
Normal operating current	I_{norm15}	25.0	[mA]
Producible magnetic moment at 15°	m	74.8	[mAm ²]
Wire cross sectional area with isolation	A_w	9.1	[mm ²]

Table 6.5: Design results for magnetorquers.

6.1.5 Magnetorquer properties

In the following the mechanical, electrical and magnetic properties of the designed coils are presented.

Temperature characteristics

The maximum producible magnetic moment in detumbling and normal operating mode is presented in figure 6.2. This shows that the producible magnetic moment decreases with an increase in temperature. In normal operating mode the magnetic moment decreases from 97.18 mAm² to 66.4 mAm².

Total mass

The total mass of the magnetorquers including insulation, and the redesigned wire diameter and number of turns is 54 g.

Electric properties

In table 6.6 the coil resistance, current and voltage is listed for minimum, normal and maximum temperature. The calculation is based on the maximum power dissipation of 214.7 mW for each coil.

Coil self inductance

The self inductance of the coil is important for the design of the coil driver. To make the calculation a bit easier the rectangular shape of the coil is approximated by a square coil with side length d .

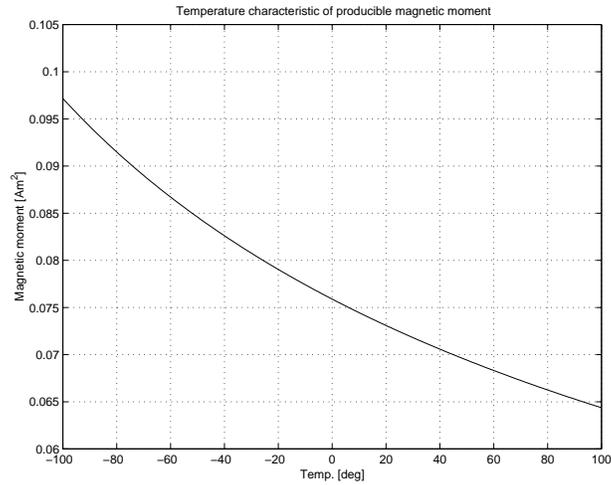


Figure 6.2: Maximum producible magnetic moment in detumbling and normal operation, for the operational temperature range.

Parameter	Symbol	Value	Measured	Unit
Maximum coil resistance	$R_{max100^{\circ}C}$	112.7	–	$[\Omega]$
Normal coil resistance	$R_{norm15^{\circ}C}$	195.6	192.3	$[\Omega]$
Minimum coil resistance	$R_{min-100^{\circ}C}$	256.9	–	$[\Omega]$
Maximum coil current	$I_{max-100^{\circ}C}$	43.3	–	$[mA]$
Normal coil current	$I_{norm15^{\circ}C}$	25.0	25.48	$[mA]$
Minimum coil current	$I_{min100^{\circ}C}$	19.0	–	$[mA]$

Table 6.6: Electrical properties of one coil.

Given this approximation the magnetic flux in the centre of a flat square coil is given by

$$B = \mu_0 \frac{2\sqrt{2}nI}{\pi d}, \quad (6.11)$$

where

μ_0 magnetic permeability for vacuum

n number of turns

I current in coil

d approximated side length

The magnetic flux linking with each turn is

$$\Phi = \int_s B ds, \quad (6.12)$$

where the integral is performed on the cross sectional area (A) of the coil. Assuming a constant magnetic flux density B on the entire surface, gives

$$\Phi = BA \quad (6.13)$$

and the self inductance of the rectangular coil (with the side length $d = \sqrt{hw}$)

$$L = \frac{n\Phi}{I} = \frac{2\sqrt{2}\mu_0 n^2 A}{\pi\sqrt{hw}} = \frac{2\sqrt{2}}{\pi}\mu_0 n^2 \sqrt{hw} \quad (6.14)$$

The self inductance of the designed coils is presented in table 6.7.

Parameter	Symbol	Value	Measured	Unit
Coil self inductance	L	19.69	43.7	[mH]

Table 6.7: Self inductance of coils.

Minimum coil rise time

The time constant of the coils is determined by the coil self inductance, the coil resistance, the measurement resistor and the output resistance of the coil driver. The time constant is given by

$$\tau = \frac{L}{R_{coil} + R_{meas} + R_{driver}} \quad (6.15)$$

The output resistance of the coil driver is assumed negligible, and the coil time constant is then as shown in table 6.8.

Parameter	Symbol	Value	Unit
Maximum coil time constant	$\tau_{max-100^{\circ}C}$	174.52	$[\mu s]$
Normal coil time constant	$\tau_{norm15^{\circ}C}$	100.61	$[\mu s]$
Minimum coil time constant	$\tau_{min100^{\circ}C}$	76.62	$[\mu s]$

Table 6.8: Coil time constant

6.2 Coil driver

The controller on the CubeSat require that the magnetic field generated by the coils has to be both positive and negative. Thereby the current that has to run through the coils has to be both positive and negative. Since there is only a positive 5 V available, the coils will be driven by a H-bridge in order to make both positive and negative currents through the coils. The H-bridge works by either turn on the Q1 and Q3 transistor to generate a positive current or by turning on the Q2 and Q4 transistor to generate a negative current. This is illustrated in figure 6.3. The coil is connected between C_1, C_2 and C_3, C_4 on the H-bridge.

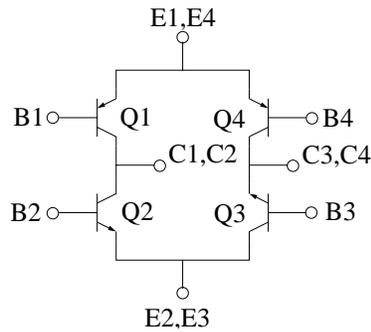


Figure 6.3: H-Bridge

6.2.1 Basis resistor

The H-bridge chosen consists of two NPN transistors and two PNP transistor. The transistors can not be driven directly by the PIC, because they will use too much current, therefore a resistor is needed between the transistors and the PIC. The transistor must be put into saturation in order to save power lost in the H-bridge due to resistance in the transistors. To save power in the H-bridge equation 6.16 must be fulfilled.

$$I_b > I_{b_{min}} = \frac{I_c}{HFE} \quad (6.16)$$

Where I_b is the current applied to the transistors, $I_{b_{min}}$ is the minimum current required, I_c is the current running through the coils and HFE is the transistor gain. Because the HFE and I_c

change with the temperature, equation 6.16 has to be fulfilled at the different temperatures the coils and transistor will experience.

The HFE has been found in the H-bridge datasheet and is showed in table 6.9, [Zetex, 2002a].

Temperature	PNP transistor	NPN transistor
$100^{\circ}C$	660	1250
$15^{\circ}C$	490	1050
$-100^{\circ}C$	265	590

Table 6.9: Table for HFE

The current has been calculated in section 6.1 and the values are shown in table 6.10

$I_{c100^{\circ}C}$	19 mA
$I_{c15^{\circ}C}$	25 mA
$I_{c-100^{\circ}C}$	43.3 mA

Table 6.10: Coil currents at different temperatures.

The values in table 6.9 and in table 6.10 are substituted into equation 6.16 and thereby finding the minimum value for the basis current I_b showed in table 6.11. Since the HFE is different for the NPN and PNP transistor both transistors have to be taken into account.

	PNP Transistor	NPN Transistor
$I_{b100^{\circ}C}$	$26.53\ \mu A$	$14.00\ \mu A$
$I_{b15^{\circ}C}$	$49.94\ \mu A$	$21.91\ \mu A$
$I_{b-100^{\circ}C}$	$150.60\ \mu A$	$67.65\ \mu A$

Table 6.11: Calculated minimum values of $I_{b_{min}}$

The basis current for the PNP transistor has to be much higher than for the NPN transistor, because of the much lower HFE in the PNP transistor. To make sure that the transistor will be in saturation mode all the time, the worst case value is found at $-100^{\circ}C$. Therefore the value chosen for the resistors needed for the PNP transistors is $28\text{ k}\Omega$ and for the NPN transistors is $63\text{ k}\Omega$. These values are found by usage of Ohms law $R_b = \frac{4.3\text{ V}}{I_{b_{min}}}$.

Even though the transistors are in saturation mode there will be a small power loss in the transistors. The power loss in the H-bridge is found by $P_{gen} = V_{ce}I_c$ where V_{ce} is the voltage across the collector and emitter. V_{ce} can be found in the datasheet for the H-bridge. The power loss is shown in table 6.12.

6.2.2 Schottky diodes

Diodes are placed at C1,C2 and C3,C4 on the H-bridge to either V_{ce} or Gnd, in order to remove peak voltage from the coils. The peak voltage will pass through the H-bridge thereby damaging

Power lost	PNP transistor	NPN transistor
$P_{gen100^{\circ}C}$	1.05 mW	0.88 mW
$P_{gen15^{\circ}C}$	1.15 mW	1.04 mW
$P_{gen-100^{\circ}C}$	1.60 mW	1.60 mW

Table 6.12: Power lost in H-bridge.

it if the diodes were not placed on the H-bridge. The h-bridge will be destroyed if the spark voltage is applied for longer periods of time. Figure 6.4 shows how the diodes are placed in the circuitry.

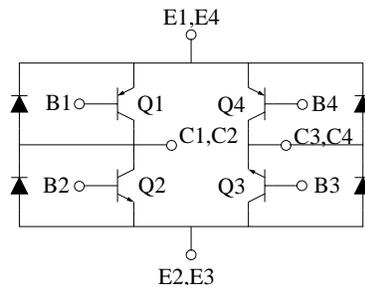


Figure 6.4: H-bridge with applied schottky diodes.

The effect of the diodes have been measured and is showed on figure 6.5. It can be seen that the voltage starts oscillating when the diodes cut off the current. The reason for this is that the inductor build up voltage since it is not fully discharged. The current that passes through the schottky diodes will charge the capacitors on the 5 V supply. The peak voltage reach without diodes with 90% duty cycle is 7.56 V and -5.69 V but only 4.19 V and -3.56 V with diodes. It is therefore concluded that the diodes works as expected.

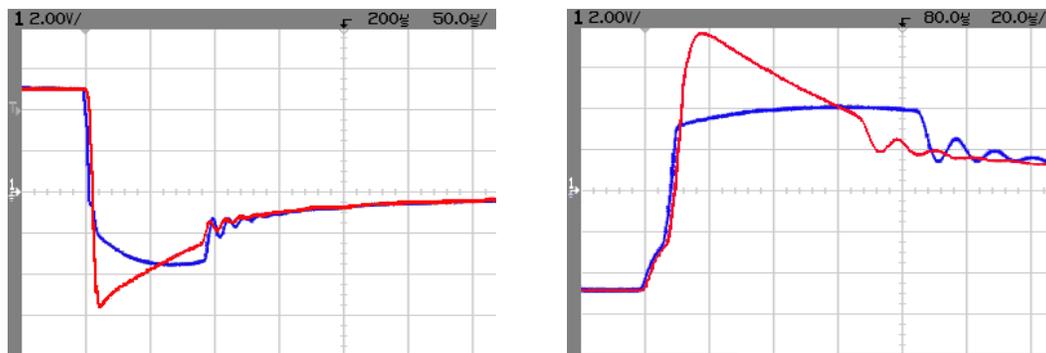
(a) Negative peak voltage ($50 \frac{\mu s}{div}$)(b) Positive peak voltage ($20 \frac{\mu s}{div}$)

Figure 6.5: Figures show the effect of the schottky diodes. Blue line is with diodes

6.3 Current sensor

The controllability of the satellite depends very much on the magnetic field generated by the coils. The magnetic field is linearly dependent on the current passing through the coils and the current is linearly dependent on the temperature of the coils. It is therefore necessary to either measure the temperature or the current. The temperature of the coils is not accessible in the CubeSat but the current is. Therefore it has been chosen to measure the current passing through the coils. The current is measured by applying a small resistors in series with the coils. The resistance of the measuring resistor is chosen to be at least 1000 times smaller than the minimum resistance of the coils. This is approximately 0.1Ω . The resistor found is 0.09Ω . The voltage across the measuring resistor varies between $\pm 0.780 \text{ mV}$ at 100°C and $\pm 3.592 \text{ mV}$ at -100°C where the coils are applied with a positive or a negative current. The ADC in the PIC requires a voltage between 0 and 5 V therefore the signal from the measuring resistor has to be amplified and offset by 2.5 V. The gain needed to amplify the voltage across the measuring resistor can be calculated with equation 6.17.

$$G = \frac{\frac{V}{2}}{V_{resmax}} = \frac{2.5 \text{ V}}{3.592 \text{ mV}} = 696 \quad (6.17)$$

The amplifier chosen is a micro power single supply rail to rail operational amplifier (LMC6462AIM). Since the input voltage is both positive and negative the amplifier circuitry is implemented as a differential amplifier see figure 6.6. The voltage across measuring resistor is applied between V_1 and V_2 .

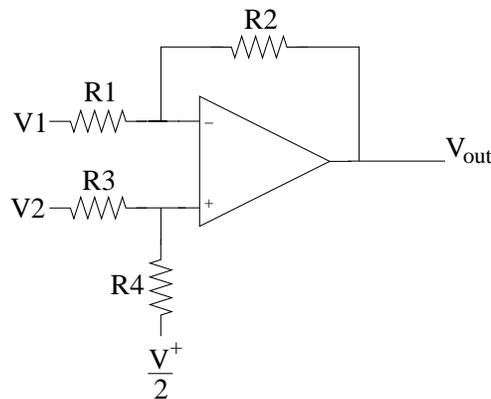


Figure 6.6: Current sensor amplifier.

The amplifier output for the circuit in figure 6.6 can be calculated with equation 6.18.

$$V_{out} = \left(\frac{R_1 + R_2}{R_3 + R_4} \right) \frac{R_4}{R_1} - \frac{R_2}{R_1} + \left(\frac{R_1 + R_2}{R_3 + R_4} \right) \frac{R_3}{R_1} \frac{V}{2} \quad (6.18)$$

Equation 6.18 can be simplified to 6.19 because $R_1 = R_3$ and $R_2 = R_4$.

$$V_{out} = \frac{R_2}{R_1}(V_2 - V_1) + \frac{V}{2} \quad (6.19)$$

The gain of the circuit in figure 6.6 is $G = \frac{R_2}{R_1}$, setting $R_2 = 100 \text{ k}\Omega$ and using the gain calculated in equation 6.17, we obtain $R_1 = 143 \text{ }\Omega$.

The reference voltage needed by the coil sensor amplifier and other circuitry on the board are constructed by using a 2.5 V reference zener diode with a resistor (R_{ref}) in series. To minimize noise on the reference voltage a $1 \text{ }\mu\text{F}$ capacitor (C_{ref}) are put in parallel with the reference diode this is illustrated in figure 6.7. The value of the resistor are calculated by usage of Ohms law. The voltage across the resistor is 2.5 V and the minimum recommended current through the reference diode is $20 \text{ }\mu\text{A}$ therefor the resistor value becomes $R_{ref} = \frac{2.5 \text{ V}}{20 \text{ }\mu\text{A}} = 125 \text{ k}\Omega$. The advantage of this circuit is that the voltage across the resistor do not vary with the current used at V_{refout} if a too big resistor have been chosen. The disadvantage is that there will be a small voltage difference between V_{zener} and V_{refout} because of the input offset of the operational amplifier. The input voltage offset is according to the data sheet [Simiconductors, 2002] only 1.5 mV this is small comparing to the $\pm 2.5 \text{ mV}$ (1%) tolerance of the zener diode in the data sheet [Zetex, 2002b].

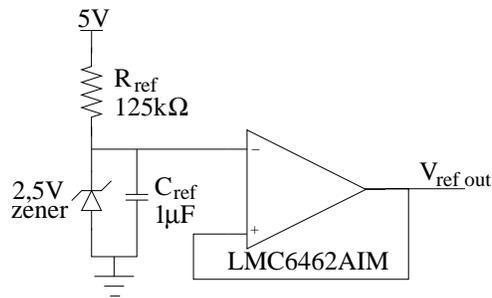


Figure 6.7: Reference amplifier

The full diagram of the ACS and the ADS can be found in appendix D.

chapter 7

Controller design

This chapter includes the design of the controllers for the CubeSat. First the rate detumbling controller is designed, and then the periodic linear systems derived in chapter 5 will be analysed for reachability, controllability and stability. Based on the results of this analysis a controller will be designed, and theory related to analysis and control of periodic linear system will be presented before they are used.

From the analysis of the open loop system, different types of controllers are considered. The main objective in the choice of a suitable controller will be that the online computational burden must be low, as the micro controller is a limiting factor.

7.1 Detumbling Controller

The purpose of the detumbling controller is to detumble the satellite after release from the launch vehicle and to ensure that the satellite remains controllable. In situations where control of the satellite has been lost, the controller must be able to regain control. This is done by reducing the kinetic energy of the satellite, thereby reducing the angular velocities to a level, where a more accurate controller can take over.

7.1.1 Design of B-dot controller

A B-dot controller has been chosen for detumbling of the satellite. This type of controller is particularly well suited for this job, since it only requires the derivate of the magnetic field as input. Hence it is capable of functioning even when all other orbital information is missing.

The aim of the B-dot controller is to reduce the kinetic energy of the satellite. Figure 7.1 shows a block diagram of the B-dot controller.



Figure 7.1: Block diagram of B-dot Controller

The kinetic energy decreases if the dot product of the angular velocity and the torque is negative.

$${}^c\boldsymbol{\Omega}_{ci}^T {}^c\mathbf{N}_{ctrl} < 0 \quad (7.1)$$

The control torque is

$${}^c\mathbf{N}_{ctrl} = {}^c\mathbf{m} \times {}^c\mathbf{B} \quad (7.2)$$

Inserting this into equation 7.1 yields

$${}^c\boldsymbol{\Omega}_{ci}^T ({}^c\mathbf{m} \times {}^c\mathbf{B}) < 0 \Leftrightarrow -{}^c\boldsymbol{\Omega}_{ci}^T ({}^c\mathbf{B} \times {}^c\mathbf{m}) < 0 \quad (7.3)$$

Since $\mathbf{a}^T (\mathbf{b} \times \mathbf{c}) = \mathbf{c}^T (\mathbf{a} \times \mathbf{b})$ the above can be rewritten as

$${}^c\mathbf{m}^T ({}^c\boldsymbol{\Omega}_{ci} \times {}^c\mathbf{B}) \quad (7.4)$$

The above inequity (equation 7.4) can be solved by

$${}^c\mathbf{m} = C ({}^c\boldsymbol{\Omega}_{ci} \times {}^c\mathbf{B}) \quad (7.5)$$

where C is a scalable gain. The choice of C will decide how fast the kinetic energy of the system converges to zero. Equation 7.5 holds as a solution to 7.4 as long as ${}^c\boldsymbol{\Omega}_{ci}$ and ${}^c\mathbf{B}$ are not alligned, in that case the result is zero. Since changes in the magnetic field (${}^c\mathbf{B} \approx {}^c\boldsymbol{\Omega}_{ci} \times {}^i\mathbf{B}$) are mainly due to rotation of the satellite, a simple control law can be obtained

$${}^c\mathbf{m} = C {}^c\dot{\mathbf{B}} \quad (7.6)$$

However since ${}^c\dot{\mathbf{B}}$ is difficult to measure an estimate ${}^c\hat{\dot{\mathbf{B}}}$ will be used in practice.

$${}^c\mathbf{m} = C {}^c\hat{\dot{\mathbf{B}}} \quad (7.7)$$

7.1.2 Filter implementation

In order to estimate the magnetic field a first order state variable filter is implemented. A block diagram of the filter is shown in figure 7.2.

The transfer function of the filter can be derived directly from the diagram. In the Laplace domain it is given by

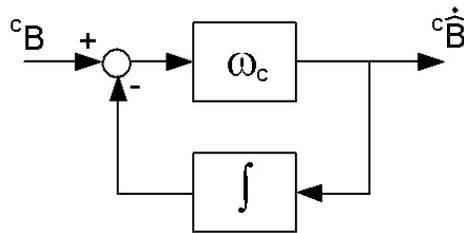


Figure 7.2: Block diagram of state variable filter

$${}^c\dot{\mathbf{B}}(s) = \omega_c \left({}^c\mathbf{B} - \frac{1}{s} {}^c\dot{\mathbf{B}}(s) \right) \quad (7.8)$$

which yields the the transfer function

$$\frac{{}^c\dot{\mathbf{B}}(s)}{{}^c\mathbf{B}(s)} = \frac{s\omega_c}{s + \omega_c} \quad (7.9)$$

In equations 7.8 and 7.9 ω_c is a selectable cutoff frequency. The cutoff frequency should be chosen 5 – 10 times larger than the highest rate of change of ${}^c\mathbf{B}$. The value of ω_c was determined by running a Simulink simulation of the magnetic field shown in figure 7.3.

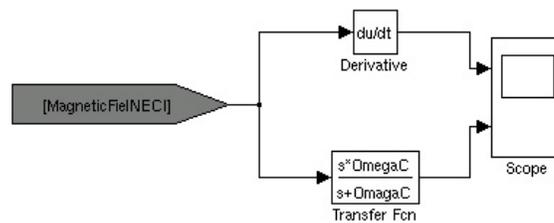


Figure 7.3: Simulink simulation of magnetic field.

By comparing the two output scopes and manually tuning the value of ω_c , the best value was determined to be 0.7.

7.2 Periodic linear systems

The nature of a magnetic actuated satellite is that it is only controllable in two degrees of freedom at any given position in orbit. The variation of the geomagnetic field over one orbit can be used in the design of the controller. This though requires some background in methods for time varying

linear systems. The type of time varying linear systems of interest are the periodic linear systems, because of the periodic nature of the geomagnetic field.

In the following sections theory of periodic linear systems will be presented. Starting out with Floquet theory, methods for analysing periodic linear systems for stability, reachability and controllability will be formulated. The objective of the analysis is to derive a gain control law, which will stabilise the satellite in the orbit.

In the derivation of the control law we assume the system to be a T-periodic system as the one in equation 7.10.

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \quad (7.10)$$

$$\mathbf{y} = \mathbf{C}(t)\mathbf{x}(t) \quad (7.11)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$ are state and input vectors respectively, $\mathbf{A}(t)$, $\mathbf{B}(t)$ are periodic matrices of period T

$$\mathbf{A}(t+T) = \mathbf{A}(t), \mathbf{B}(t+T) = \mathbf{B}(t) \quad (7.12)$$

7.2.1 Floquet theory

In this section the results from the Floquet theory on linear differential equations with periodic coefficients are presented. This is mainly based on [Mohler, 1991].

The stability of a periodic system is to be considered over time, therefore the state transition matrix calculated for one period, plays a significant role in the stability analysis. The placement of the eigenvalues of the transition matrix in the open unit disk, determines the stability of the system.

The system in equation 7.13 with zero input vector has the transition matrix denoted by $\Phi_A(t, t_0)$, where t_0 is the initial time, and t is the final time.

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) \quad (7.13)$$

The following holds

$$\Phi(t+T, t_0) = \Phi_A(t, t_0)\mathbf{C} \quad (7.14)$$

where \mathbf{C} is a constant matrix.

To check the equality, it is enough to calculate the time derivative of $\Phi(t+T, t_0)$

$$\Phi(t+T, t_0) = \Phi(t, t_0)\mathbf{C} = \mathbf{A}(t)\Phi_A(t, t_0) = \mathbf{A}(t+T)\Phi_A(t+T, t_0) \quad (7.15)$$

since

$$\dot{\Phi}_A = \mathbf{A}(t)\Phi_A \quad (7.16)$$

There exist a constant matrix \mathbf{R} such that

$$\mathbf{C} \equiv e^{\mathbf{R}T} \quad (7.17)$$

From the equality $\Phi_A(t_0, t_0) = \mathbf{I}$ and equation 7.14 it follows that

$$\Phi_A(t_0 + T, t_0) = e^{\mathbf{R}T} \quad (7.18)$$

Thus the state transition matrix consist of a periodically modulated exponential matrix function. The matrix $\Phi_A(t_0 + T, t_0)$ is named the monodromy matrix at t_0 and is denoted by $\Psi_A(t_0)$. It is concluded from equation 7.18 that the eigenvalues of the monodromy matrix are independent of t_0 .

The system 7.13 is asymptotically stable if, and only if the eigenvalues of matrix \mathbf{R} have all negative real parts. The eigenvalues of \mathbf{R} are defined as characteristic exponents of $\mathbf{A}(t)$ and are denoted by ρ . That is:

$$\det[\mathbf{I}\rho - \mathbf{R}] = 0 \quad (7.19)$$

The eigenvalues of the monodromy matrix are called characteristic multipliers of $\mathbf{A}(t)$ and are denoted by λ . The system is stable if all λ are located on the open unit disc. That is:

$$\det[\mathbf{I}\lambda - e^{\mathbf{R}T}] = 0 \quad (7.20)$$

which implies:

$$|\lambda_i| < 1 \quad (7.21)$$

7.2.2 Criteria for stability, reachability and controllability

Now the criteria to determine whether the system is stable, reachable and controllable is set up.

Stability

From the previous section we know that the criteria for stability is determined from the eigenvalues of the monodromy matrix, which are supposed to be within the open unit disk. The equivalent method is to check the real parts of the eigenvalues of the matrix \mathbf{R} , which shall have negative real parts.

Reachability

A state x_1 is said to be reachable if there exist an input that transfers the state $x(t)$ from the zero state to x_1 in finite time.

If a system is found to be reachable, the reachability refers to the ability of the system to reach x_1 from the origin in some finite time; it specifies neither the time it takes to achieve this, nor the trajectory to be followed.

The reachability of the pair $(\mathbf{A}(t), \mathbf{B}(t))$ can be examined by calculating the controllability gramian matrix $(\mathbf{G}_c(t_0 + T, t_0))$ given by equation 7.22.

$$\mathbf{G}_c(T, 0) \equiv \int_0^T \Phi(T, s) \mathbf{B}(s) \mathbf{B}^*(s) \Phi^*(T, s) ds \quad (7.22)$$

If the controllability gramian is not singular, then the statement

$$\mathbf{B}(t)^* \Phi_A^*(t, t_0) \mathbf{x} = 0, \text{ for all } t \in [t_0, t_0 + T] \quad (7.23)$$

is false for any nonzero vector \mathbf{x} , and the pair $(\mathbf{A}(t), \mathbf{B}(t))$ is reachable [Boulet, 2002].

Reachability is a property of a periodic system within one period. The system is reachable if the controllability gramian \mathbf{G}_c is not singular.

Controllability

A state x_0 is said to be controllable if there exists an input that transfers the state from x_0 to the zero state in some finite time.

Similar to reachability, controllability does not specify the time to reach the zero state, nor the state trajectory to be followed.

To determine whether the system is controllable or not, we first introduce the controllability operator (eq. 7.24), which is the mapping of the admissible input signals $u \in \mathcal{U}_{ad}$, where \mathcal{U}_{ad} is the set of admissible inputs as defined in equation 7.25 [Boulet, 2002]

$$\mathcal{L} : \mathcal{U}_{ad} \rightarrow \mathbb{R}^n, \quad \mathcal{L}u(t) = \int_{t_0}^{t_1} \Phi(t_1, \tau) \mathbf{B}(\tau) u(\tau) d\tau \quad (7.24)$$

$$\mathcal{U}_{ad} \equiv \mathcal{L}_2^m[t_0, t_1] \quad (7.25)$$

With this given the controllability problem is concerned with finding whether any terminal state $x_1 = x(t_1)$ can be attained by a suitable choice of the input, and with determining the inputs which can attain the terminal state.

This problem can be solved by finding the solution to the controllability equation (eq. 7.26), in which the input produces a terminal state $z \in \mathbb{R}^n$ from the zero initial state, given that a solution can be found.

$$\mathcal{L}u(t) = z \quad (7.26)$$

More generally, the input u produces a change in the terminal state $z = x_1 - \Phi(t_1, t_0)x_0$ from what it would be as a free trajectory starting at (x_0, t_0) .

To solve the controllability problem, the system is transferred into a discrete time system, as the solution in continuous time has infinite dimension. In discrete time the sampling of input and state trajectories, the problem becomes finite-dimensional.

The discrete time controllability operator is given by equation 7.27.

$$\mathcal{L} : \mathcal{U}_{ad} \rightarrow \mathbb{R}^n, \quad \mathcal{L}u(i) = \sum_{i=r}^{k-1} \Phi(k, i+1)\mathbf{B}(i)u(i) \quad (7.27)$$

where the discrete time state transition matrix is defined as equation 7.28.

$$\Phi(k, r) = \prod_{i=r}^{k-1} \mathbf{A}(i) \quad (7.28)$$

Then the discrete time controllability equation may be written in matrix form as equation 7.29.

$$\mathcal{L}u \equiv \underbrace{[(k, r+1)\mathbf{B}(r) \ (k, r+2)\mathbf{B}(r+1) \ \cdots \ \Phi(k, k)\mathbf{B}(k-1)]}_{\equiv L} \underbrace{\begin{bmatrix} u(r) \\ u(r+1) \\ \vdots \\ u(k-1) \end{bmatrix}}_u = z \quad (7.29)$$

The system is controllable if z belongs to the vector space given by \mathcal{L} , and this implies that there exist many solutions to the controllability equation [Boulet, 2002].

It can be verified that $z \in \mathcal{R}\{\mathcal{L}\}$, by finding $\mathcal{R}\{\mathcal{L}\}$, which is the space of attainable states. The dimension of $\mathcal{R}\{\mathcal{L}\}$, is equal to the rank of L (eq. 7.29). It is also of interest to determine whether the pseudo inverse solution (eq. 7.30) to the controllability equation exist.

$$u_{opt} = \mathcal{L}^* (\mathcal{L}\mathcal{L}^*)^{-1} z \quad (7.30)$$

If the pseudo inverse solution $(\mathcal{L}\mathcal{L}^*)^{-1}$ exist and the controllability operator (L) has full rank, the system is said to be completely controllable.

The controllability operator can also be used to find the discrete time version of the controllability gramian, as it is defined as $M(r, k)$ in equation 7.31.

$$\mathcal{L}\mathcal{L}^* = \underbrace{\sum_{i=r}^{k-1} \Phi(k, i+1)\mathbf{B}(i)\mathbf{B}(i)^*\Phi(k, i+1)^*}_{M(r, k)} z \quad (7.31)$$

7.3 Controllability analysis

In this section the two models developed in chapter 5 is analysed for stability, reachability and controllability based on the theory just presented.

First a general method for the analysis is given, and then the results from the analysis are given.

7.3.1 Analysis method

The method used for the analysis is given below.

1. First the continuous time system is modelled and converted to discrete time, with a sampling time of one second.
2. Then the discrete time state transition matrix (eq. 7.28) is calculated.
3. The last iteration of the transition matrix is equal to the monodromy matrix, and the eigenvalues are found.
4. The controllability operator is calculated and the rank and pseudo inverse solution is found.
5. Finally the controllability gramian is calculated.

7.3.2 Results and evaluation

The results from the simulations for both the inertial and Nadir pointing model are presented in table 7.1.

	Inertial pointing model	Nadir pointing model
$\lambda\Psi_A$	$\begin{bmatrix} 4.925 \\ 1.153 \pm i1.559 \\ 0.305 \pm i0.434 \\ 0.191 \end{bmatrix}$	$\begin{bmatrix} 534.924 \\ 180.574 \\ 2.11 \\ 1.401 \\ -0.912 \pm i0.408 \end{bmatrix}$
$M(k, r)^{-1}$	exist	exist
$cond(M(k, r))$	15423249.884	132166107462.888
$rank(\mathcal{L})$	6	6
$(\mathcal{L}\mathcal{L}^*)^{-1}$	exist	exist
$cond((\mathcal{L}\mathcal{L}^*)^{-1})$	94548270.009	131872486261.284

Table 7.1: Results from systems analysis.

The results in table 7.1 implies that none of the systems are asymptotically stable, which is what was expected. In both cases the inverse of the controllability exist, but the condition

number shows that the matrix is close to singular. The same thing is seen with the pseudo inverse controllability operator.

From these results the system should be reachable and controllable. Thus the large condition number for both the controllability gramian and pseudo inverse controllability operator implies that these matrices are close to being singular. From this observation two design methods for the control law has been chosen. One is a constant gain controller, based on a time invariant approximation of the periodic linear systems derived in chapter 5 will be used. It is assumed that it will be hard to design a stabilizing controller using this approach. This assumption is based on the reachability and controllability analysis, where the matrices where found close to singular.

The other approach chosen, is the linear matrix inequality (LMI) representation of the control problem. This approach gives either a time varying or constant control law. The advantage of using the LMI approach is that the manual search for the weighing matrices may not result in the solution, which is the global minimum of the performance function. With LMI the solution will always be the global minimum.

In the following the constant gain controller will be investigated, and after that the LMI approach is presented.

7.4 Constant gain controller

The constant gain controller requires very little on-line computation. It is therefore very attractive for use in small satellites, where computational power is often limited. The controller developed in this chapter will be based on a linear quadratic regulator.

7.4.1 Linear Quadratic Controller

Consider the T-periodic system:

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \quad (7.32)$$

where matrices \mathbf{A} and \mathbf{B} are T-periodic.

The aim of this section is to find a control law that will actuate the above system with a constant gain such that:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}_c(t)\mathbf{x}(t) \\ \mathbf{A}_c(t) &= \mathbf{A}(t) - \mathbf{B}(t)\mathbf{K} \end{aligned} \quad (7.33)$$

In order to calculate a constant gain the equivalent time invariant system must be found:

$$\dot{\mathbf{x}} = \tilde{\mathbf{A}}\mathbf{x}(t) + \tilde{\mathbf{B}}\mathbf{u}(t) \quad (7.34)$$

where

$$\tilde{\mathbf{A}} = \frac{1}{T} \int_{t_0}^{t_0+T} \mathbf{A}(t) dt \text{ and } \tilde{\mathbf{B}} = \frac{1}{T} \int_{t_0}^{t_0+T} \mathbf{B}(t) dt \quad (7.35)$$

Using the time invariant system the control gain \mathbf{K} , can be calculated using the steady state solution to the time invariant Riccati equation:

$$-\dot{\mathbf{P}} = 0 = \mathbf{P}\tilde{\mathbf{A}} + \tilde{\mathbf{A}}^T\mathbf{P} - \mathbf{P}\tilde{\mathbf{B}}\mathbf{R}^{-1}\tilde{\mathbf{B}}^T\mathbf{P} + \mathbf{Q} \quad (7.36)$$

where matrices \mathbf{Q} and \mathbf{R} are diagonal weighing matrices. The control gain matrix \mathbf{K} is equal to:

$$\mathbf{K} = -\mathbf{R}^{-1}\tilde{\mathbf{B}}^T\mathbf{P} \quad (7.37)$$

where \mathbf{P} is the steady state solution to equation 7.36.

And thus the control torque is:

$${}^c\tilde{\mathbf{m}} = -\mathbf{R}^{-1}\tilde{\mathbf{B}}^T\mathbf{P}\mathbf{x}(t) \quad (7.38)$$

Once the the control gain \mathbf{K} has been calculated, it is necessary to check the stability of the system in 7.33 by calculating its characteristic multipliers as described in section 7.2. If the characteristic multipliers are all on the open unit disc, then the system is stable.

7.4.2 Choosing \mathbf{R} and \mathbf{Q}

The stability check in the previous is necessary because stability the time invariant system 7.34 the time varying system 7.32 is not equivalent. This is because the time invariant system only is equivalent to the time varying system to the first order approximation. The sensitivities of the two systems are not equal either, where the time invariant system is only controllable when viewed over a full period, the time invariant system is fully controllable at any given time.

When examining the equations 7.36 and 7.33 it is seen that the values of \mathbf{R} and \mathbf{Q} will greatly influence the stability of the system. Eventhough some values of \mathbf{R} and \mathbf{Q} will stabilize the time invariant system, they may not necessarily stabilize the time varying system.

There exits no exact formulas for calculating the values of \mathbf{R} and \mathbf{Q} , however there exits some rules of thumb. If the designer has sufficient knowledge of system and is capable of determining the maximum values of the individual states and inputs, the matrices can be specified as follows:

$$\mathbf{R}(i, i) = \frac{1}{u_{i,max}^2} \text{ and } \mathbf{Q}(i, i) = \frac{1}{x_{i,max}^2} \quad (7.39)$$

In most cases this is not so, and the values have to be determined other wise. Since it is the numerical relationship between the matrices that determines stability and not specific values, one matrix can remain constant, while the other is tuned for stability. For simplicity the \mathbf{R} matrix usually equals the identity matrix \mathbf{R} , while the \mathbf{Q} matrix is used to tune the system.

7.5 LMI controller

In this section the LMI approach to the state feedback problem is describe. Presentation of the control algorithms will be mainly based on [Wisniewski and Stoustrup, 2001], whereas the more general properties of LMI's will be based on [Scherer and Weiland, 1999].

The first problem to be addressed is to represent the control synthesis problem as a LMI formulation. This is done for the generalised H_2 optimal control problem. When an algorithm to solve the H_2 optimal control problem has been set up, the implementation in MATLAB is described as an introduction to the LMI tool-box.

7.5.1 Control synthesis using LMI

The control synthesis problem addressed in this section, is the generalised H_2 optimal control problem. The application of H_2 control synthesis on periodic systems is valid because the separation principle is valid for a periodic system [Wisniewski and Stoustrup, 2001].

Equation 7.40 specifies the system used for the H_2 control synthesis. The system (s) is considered to have full state information, which makes $\mathbf{C}_2 = \mathbf{E}$, and $\mathbf{D}_{21} = \mathbf{0}$.

$$\begin{aligned}\mathbf{x}(t+1) &= \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}_1(t)\mathbf{w}(t) + \mathbf{B}_2(t)\mathbf{u}(t) \\ \mathbf{z}(t) &= \mathbf{C}_1(t)\mathbf{x}(t) + \mathbf{D}_{12}(t)\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}_2(t)\mathbf{x}(t) + \mathbf{D}_{21}(t)\mathbf{u}(t)\end{aligned}\tag{7.40}$$

Now the periodic state feedback $\mathbf{u}(t) = \mathbf{K}(t)\mathbf{x}(t)$, $\mathbf{K}(t+P) = \mathbf{K}(t)$. The objective of the control design is to compute the control gain $\mathbf{K}(t)$ for which the close loop system (s_c) is given as in equation 7.41.

$$\begin{aligned}\mathbf{x}(t+1) &= \mathbf{A}_c(t)\mathbf{x}(t) + \mathbf{B}_1(t)\mathbf{w}(t) \\ \mathbf{z}(t) &= \mathbf{C}_c(t)\mathbf{x}(t)\end{aligned}\tag{7.41}$$

where $\mathbf{A}_c(t) = \mathbf{A}(t) + \mathbf{B}_2(t)\mathbf{K}(t)$, $\mathbf{C}_c(t) = \mathbf{C}_1(t) + \mathbf{D}_{12}(t)\mathbf{K}(t)$ which satisfies

$$\|s_c\|_2 < \gamma\tag{7.42}$$

The H_2 norm for the periodic system s may be written as [Wisniewski and Stoustrup, 2001]

$$\|s\|_2 = \left(\frac{1}{N} \text{tr} \sum_{t=0}^{N-1} \mathbf{B}_2(t)^T \tilde{\mathbf{Q}}(t) \mathbf{B}_2(t) \right)^{\frac{1}{2}}\tag{7.43}$$

where $\tilde{\mathbf{Q}}$ is the steady state solution of the algebraic Lyapunov equation 7.44 is solved for each $t \in \{0, \dots, N-1\}$.

$$\tilde{\mathbf{Q}}(t) = \Phi(t+N, t)^T \tilde{\mathbf{Q}}(t) \Phi(t+N, t) + \sum_{j=t}^{t+N-1} \Phi(t+N, j+t)^T \mathbf{C}(j)^T \mathbf{C}(j) \Phi(t+N, j+1) \quad (7.44)$$

From equation 7.43 the generalised h_2 norm is equivalent to

$$\text{tr} \left(\sum_{t=0}^{N-1} \mathbf{B}_2(t) \mathbf{Q}^{-1}(t) \mathbf{B}_2(t) \right) < N\gamma^2 \quad (7.45)$$

where \mathbf{Q}^{-1} is N-periodic and satisfies the inequality

$$\mathbf{Q}^{-1}(t) - \mathbf{A}_c(t)^T \mathbf{Q}^{-1}(t) \mathbf{A}_c(t) - \mathbf{C}_c(t)^T \mathbf{C}_c(t) > 0 \quad (7.46)$$

Defining $\mathbf{Z}(t)$ as the solution to the following LMI

$$\mathbf{B}_1(t)^T \mathbf{Q}^{-1}(t) \mathbf{B}_1(t) < \mathbf{Z}(t) \quad (7.47)$$

then equation 7.45 is equivalent to

$$\text{tr} \left(\sum_{t=0}^{N-1} \mathbf{Z}(t) \right) < N\gamma^2 \quad (7.48)$$

Then applying the Schur complement [Scherer and Weiland, 1999] on equation 7.47, we get the LMI given by equation 7.49.

$$\begin{bmatrix} \mathbf{Q}(t) & \mathbf{B}_1(t) \\ \mathbf{B}_1(t)^T & \mathbf{Z}(t) \end{bmatrix} > 0 \quad (7.49)$$

Next we use the Schur complement twice on equation 7.46. This results in the following LMI.

$$\begin{bmatrix} -\mathbf{Q}(t) & \mathbf{A}_c(t) & \mathbf{0} \\ \mathbf{A}_c(t)^T & -\mathbf{Q}^{-1}(t-1) & \mathbf{C}_c(t)^T \\ \mathbf{0} & \mathbf{C}_c(t) & -\mathbf{E} \end{bmatrix} < 0 \quad (7.50)$$

Equation 7.50 can be grouped into two terms, one dependent on $\mathbf{K}(t)$, and one dependent on $\mathbf{Q}(t)$. This may be represented as in equation 7.51.

$$\begin{bmatrix} -\mathbf{Q}(t) & \mathbf{A}(t) & \mathbf{0} \\ \mathbf{A}(t)^T & -\mathbf{Q}^{-1}(t-1) & \mathbf{C}(t)^T \\ \mathbf{0} & \mathbf{C}(t) & -\mathbf{E} \end{bmatrix} + [\mathbf{B}_2(t) \ \mathbf{0} \ \mathbf{D}_{12}(t)^T]^T \mathbf{K}(t) [\mathbf{0} \ \mathbf{E} \ \mathbf{0}] \quad (7.51)$$

$$+ [\mathbf{0} \ \mathbf{E} \ \mathbf{0}]^T \mathbf{K}(t)^T [\mathbf{B}_2(t) \ \mathbf{0} \ \mathbf{D}_{12}(t)^T] < 0$$

To find the solution of the LMI in equation 7.51 the projection lemma [Scherer and Weiland, 1999] is introduced.

Given a LMI with arbitrary matrices Ψ_a , Ψ_b and a symmetric \mathbf{P} , the LMI

$$\Psi_a^T \mathbf{X} \Psi_b + \Psi_b^T \mathbf{X} \Psi_a + \mathbf{P} < 0 \quad (7.52)$$

is solvable if and only if

$$\mathbf{W}_a^T \mathbf{P} \mathbf{W}_a < 0 \text{ and } \mathbf{W}_b^T \mathbf{P} \mathbf{W}_b < 0 \quad (7.53)$$

where \mathbf{W}_a , \mathbf{W}_b are any matrices with columns forming bases for the null spaces of Ψ_a and Ψ_b .

Equation 7.51 has the same structure as equation 7.52 from the projection lemma. This can be used to determine if equation 7.51 is solvable or not. The LMI in equation 7.51 is solvable if and only if

$$\mathbf{W}_a(t)^T \begin{bmatrix} -\mathbf{Q}(t) & \mathbf{A}(t) & \mathbf{0} \\ \mathbf{A}(t)^T & -\mathbf{Q}^{-1}(t-1) & \mathbf{C}(t)^T \\ \mathbf{0} & \mathbf{C}(t) & -\mathbf{E} \end{bmatrix} \mathbf{W}_a(t) < 0 \quad (7.54)$$

and

$$\mathbf{W}_b(t)^T \begin{bmatrix} -\mathbf{Q}(t) & \mathbf{A}(t) & \mathbf{0} \\ \mathbf{A}(t)^T & -\mathbf{Q}^{-1}(t-1) & \mathbf{C}(t)^T \\ \mathbf{0} & \mathbf{C}(t) & -\mathbf{E} \end{bmatrix} \mathbf{W}_b(t) < 0 \quad (7.55)$$

where

$$\mathbf{W}_a(t) = \begin{bmatrix} \mathbf{W}_1(t) & \mathbf{0} \\ \mathbf{0} & \mathbf{E} \\ \mathbf{W}_2(t) & \mathbf{0} \end{bmatrix} \text{ and } \mathbf{W}_b(t) = \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{E} \end{bmatrix} \quad (7.56)$$

where

$$\text{im} \begin{bmatrix} \mathbf{W}_1(t) \\ \mathbf{W}_2(t) \end{bmatrix} = \ker [\mathbf{B}_2(t)^T \ \mathbf{D}_{12}(t)^T] \quad (7.57)$$

The LMI (7.55) is always fulfilled, whereas (7.54) is equivalent to equation 7.58.

$$\begin{bmatrix} -\mathbf{W}_1(t)^T \mathbf{q}(t) \mathbf{W}_1(t) - \mathbf{W}_2(t)^T \mathbf{W}_2(t) & \mathbf{W}_1(t)^T \mathbf{A}(t) + \mathbf{W}_2(t)^T \mathbf{C}_1(t) \\ \mathbf{A}(t)^T \mathbf{W}_1(t) + \mathbf{C}_1(t)^T \mathbf{W}_2(t) & -\mathbf{Q}^{-1}(t-1) \end{bmatrix} < 0 \quad (7.58)$$

Applying the Schur complement on equation 7.58. the LMI in equation 7.59 is the result.

$$\begin{aligned} & \mathbf{W}_1(t)^T \mathbf{A}(t) + \mathbf{W}_2(t)^T \mathbf{C}_1(t) \mathbf{Q}(t-1) \mathbf{A}(t)^T \mathbf{W}_1(t) + \mathbf{C}(t)^T \mathbf{W}_2(t) \\ & - \mathbf{W}_1(t)^T \mathbf{Q}(t) \mathbf{W}_1(t) - \mathbf{W}_2(t)^T \mathbf{W}_2(t) < 0 \end{aligned} \quad (7.59)$$

To find the controller ($\mathbf{K}(t)$), the periodic matrix $\mathbf{Q}(t)$ must first be found. This is done by solving the LMI's 7.59, 7.49 and 7.48, and then the LMI 7.51. This is also shown in the algorithm below.

1. Find a symmetric matrix $\mathbf{Q}(t)$ and a matrix $\mathbf{Z}(t)$ for $t = 0, 1, \dots, N-1$, satisfying the LMI's 7.59, 7.49 and 7.48.
2. Find a matrix $\mathbf{K}(t)$ for $t = 0, 1, \dots, N-1$ satisfying the LMI 7.51.

7.5.2 Implementation

The implementation is based on the system s in equation 7.40. The system is considered to have full state information ($\mathbf{C}_2 = \mathbf{E}$ and $\mathbf{D}_{21} = \mathbf{0}$). The matrices $\mathbf{A}(t)$ and $\mathbf{B}_2(t)$ are the linear system matrices found in chapter 5. Finally the matrices $\mathbf{B}_2(t)$, $\mathbf{C}_1(t)$ and $\mathbf{D}_{12}(t)$ are chosen. The three matrices to be chosen, may be seen like the weight matrices in optimal control, and they are chosen as shown in equation 7.60.

$$\mathbf{B}_1 = \begin{bmatrix} \mathbf{E} \\ \mathbf{0} \end{bmatrix}, \mathbf{C}_1 = \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{E} \end{bmatrix}, \mathbf{D}_{12} = \begin{bmatrix} \mathbf{E} \\ \mathbf{E} \end{bmatrix} \quad (7.60)$$

Now the control synthesis for the system is set up based on the control synthesis algorithm. The first step of the algorithm is solved for the entire period at once, using the property of LMI's that they can be expressed as a system of LMI [Scherer and Weiland, 1999], which is written as shown in equation 7.61.

$$F(x) := \begin{pmatrix} F_1(x) & 0 & \dots & 0 \\ 0 & F_2(x) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & F_k(x) \end{pmatrix} > 0 \quad (7.61)$$

The system of LMI's is set up using functions from the LMI toolbox in MATLAB. The LMI-lab, which is a part of the LMI toolbox can handle any system of LMI's of the form

$$N^T L(X_1, \dots, X_k) N < M^T R(X_1, \dots, X_k) M \quad (7.62)$$

where X_i are the unknown matrix variables, in this case \mathbf{Q} , \mathbf{Z} and \mathbf{K} , and \mathbf{M} and \mathbf{N} are matrices with the same dimensions as \mathbf{L} and \mathbf{R} .

In MATLAB the system is defined and verified using five different commands briefly describe below [Gahninet et al., 1995].

- `stelmis` used to initialise the LMI system.
- `lmivar` used to specify the variable (\mathbf{L} and \mathbf{R}) and their structure.
- `lmiterm` used to specify the matrices (\mathbf{M} and \mathbf{N}).
- `newlmi` used to specify the next entry (i) in the LMI system.
- `getlmis` used to make the internal representation of the LMI system, from the data entered with `lmivar` and `lmiterm`.
- `lmiedit` graphical user interface to enter or verify an entered LMI system.

Once the system is entered the LMI toolbox provides a solver for each of the three generic problems related to LMI's.

One (`feasp`) is used to check the feasibility of the LMI system. If the system is not feasible there exist no solutions to the LMI. When a LMI system has been found feasible, the optimisation problem can be solved using `mincx`. The solution to the optimization problem gives the optimal sequence which minimises the performance function. The last function (`gevp`) is used to solve the generalised eigenvalue problem, which minimises the largest eigenvalue of the system.

chapter 8

Implementation and test

In this chapter the efforts made to implement and test the controllers are described, and the results are presented. First the designed B-dot controller will be described, and simulation results will be presented for both a continuous and discrete time controller. After that the efforts made to manually tune the weight matrices for the constant gain controller will be described, and after that the LMI approach to the control design will be described. The final choice of a controller will be tested and verified with the nonlinear model, including the disturbance torques.

8.1 B-dot controller

The detumbling controller described in section 7.1, will be implemented and tested. First the continuous time implementation is tested, and then a discrete time implementation is derived and tested, for use in the micro controller.

The implementation of the filter used to estimate the derivative of the geomagnetic field, is made according to the transfer function 8.1, with a cut off frequency (ω_c) equal to 0.7, as it was found in section 7.1.

$$tf = \frac{s\omega_c}{s + \omega_c} \quad (8.1)$$

The gain used for the detumbling controller was found through a number of simulations, and determined to -11000, which resulted in the fastest energy dissipation of the system.

The filter and controller is now converted into discrete time, for the implementation in the micro controller. To fit the output of the magnetometer, the input to the filter, was scaled to fit Gauss instead of Tesla. This resulted in a gain, that was one tenth of a thousand times smaller, corresponding to the scaling factor from Tesla to Gauss.

To fit the sampling of the AD-converters, quantisers was added to the input of the geomagnetic field.

Simulation of one orbit, was carried out with an initial angular velocity of 0.1 *rad/s* on each axis. The results from the simulations of the continuous and discrete time implementation of the detumbling controller are shown in figure 8.1.

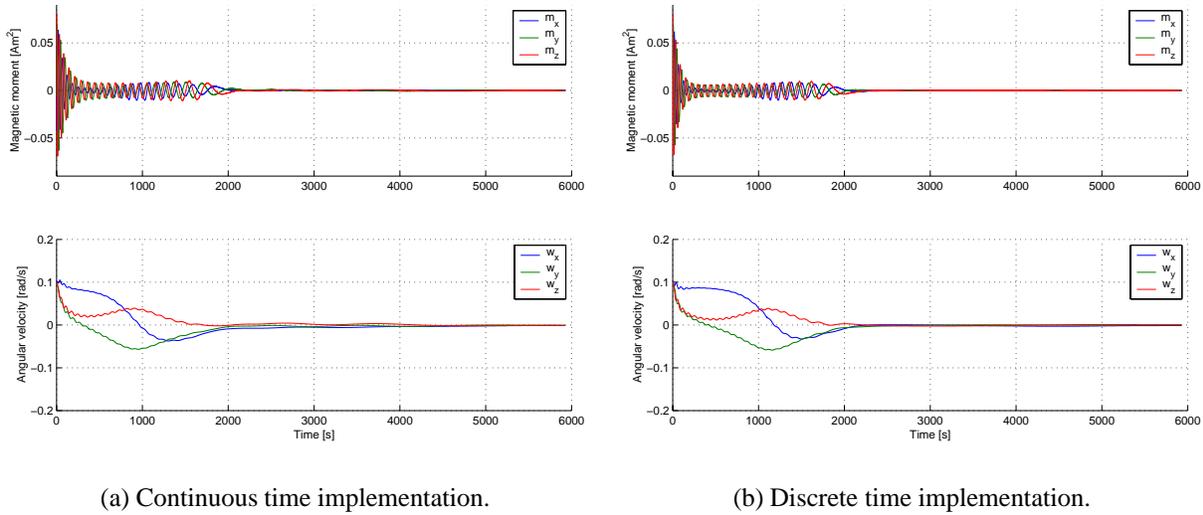


Figure 8.1: Test of continuous and discrete time implementation of detumbling controller.

8.1.1 Summary

It has been verified that the detumbling controller reduces the angular velocities within the specified time, which is within 1 – 2 orbits.

8.2 Constant gain

The design of a constant gain controller described in section 7.4 has been implemented both for the inertial and Nadir pointing model. The implementation was done using the procedure given below.

1. Setup / Update \mathbf{Q} matrix.
2. Calculate gain matrix using the `lqr` function in MATLAB.
3. Calculate the monodromy matrix for the periodic closed loop system.
4. Verify eigenvalues of the monodromy matrix

8.2.1 Choosing the weights

From section 7.4 guide lines for choice of the weight matrices were given. The first method mentioned where the weights are found from maximum inputs and states gave no stabilising results. Whereas the method where the \mathbf{R} matrix is held constant to the identity matrix and the values in the \mathbf{Q} matrix is changed gave better results.

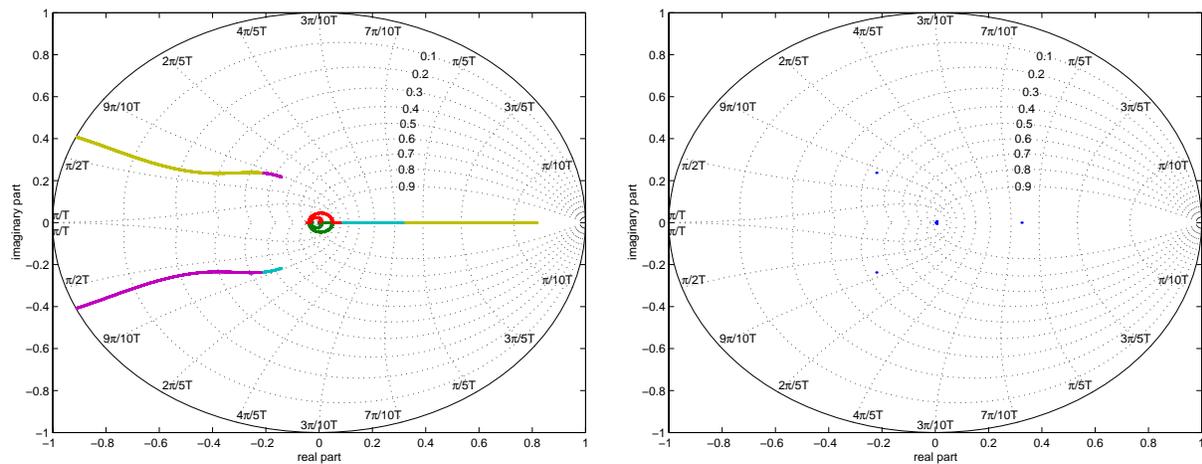
The simple solution with weights only in the diagonal was not found suitable. Different ways of scaling the \mathbf{Q} matrix were tried in order to find a solution which would stabilise the satellite in the orbit. The best solution found is shown in equation 8.2.

$$\mathbf{Q} = \begin{bmatrix} qk_1\mathbf{E} & \mathbf{0} \\ \mathbf{0} & qk_2\mathbf{E} \end{bmatrix}^T \begin{bmatrix} 0.5\mathbf{E} & 0.5\mathbf{E} \\ 0.5\mathbf{E} & 0.5\mathbf{E} \end{bmatrix} \begin{bmatrix} qk_1\mathbf{E} & \mathbf{0} \\ \mathbf{0} & qk_2\mathbf{E} \end{bmatrix} \quad (8.2)$$

where $k_1 = 0.001$, $k_2 = 0.00001$ and q is the scaling factor used to find the optimal solution.

8.2.2 Constant gain for nadir pointing model.

The constant gain controller for the Nadir pointing model, has been found with the use of the procedure and weight matrix from above. The results from the calculation of the closed loop monodromy matrix is shown in figure 8.2(a), and the optimal value of the scaling factor q , was found by searching for the minimum value of the largest normed eigenvalue in the monodromy matrix. This search is shown in figure 8.3. Where the value of q corresponding to the optimal eigenvalue was found to be 381500. The eigenvalues for the optimal values of q is shown in figure 8.2(b).



(a) Plot of eigenvalues of the closed loop monodromy matrix, for $q = 1000 - 500000$

(b) Plot of eigenvalues with best values of $q = 381500$.

Figure 8.2: Plot of eigenvalues of monodromy matrix for the Nadir pointing model.

The calculated gain matrix is tested with the nonlinear model, including the disturbance torques. The test is carried out for an initial offset of 30° on each axis, and an initial angular velocity of $3 \cdot 10^{-3} [rad/s]$. It is then verified that the satellite converges to the reference. The results are shown in figure 8.4(a).

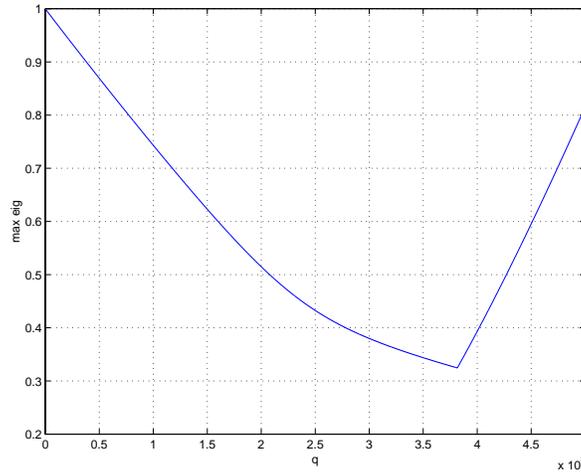


Figure 8.3: Search for best choice of scaling factor q for the nadir pointing model.

The simulation shows that the angular velocities decrease to values of approximately $1 \cdot 10^{-4}$ [rad/s], which is acceptable. When it comes to the quaternions q_1 , q_2 and q_4 converges to the reference, within an error of $\pm 2^\circ$. Whereas q_3 converges towards 0.1, which corresponds to an error of 15° on the z-axis.

This offset can be compensated by including integral action on the quaternions. The integral action needed to compensate the offset, has been found empirically. From figure 8.4(b) it can be seen that the inclusion of the integral action has the desired effect. The integral action is applied by adding a integral state (\mathbf{q}_i) to the state vector. The system with the extended state vector become:

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \mathbf{A} & \mathbf{0}_{6 \times 3} \\ \tilde{\mathbf{C}} & \mathbf{0}_{3 \times 3} \end{bmatrix} \mathbf{x}_e + \begin{bmatrix} \mathbf{B} \\ \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (8.3)$$

$$y = \mathbf{C} \mathbf{x}_e \quad (8.4)$$

where $\mathbf{x}_e = \begin{bmatrix} \boldsymbol{\Omega} \\ \mathbf{q} \\ \mathbf{q}_i \end{bmatrix}$ is the extended state vector, $\tilde{\mathbf{C}} = [\mathbf{0}_{3 \times 3} \quad \mathbf{E}_{3 \times 3}]$ and $\mathbf{C} = \mathbf{E}_{9 \times 9}$

The system with integral action has been tested with the closed loop monodromy matrix. From figure 8.5 it can be verified that the system is periodically stable.

8.2.3 Constant gain for inertial pointing model

The constant gain controller for the inertial pointing model, was found using the same method as for the Nadir pointing model. The eigenvalues of the closed loop monodromy matrix are shown

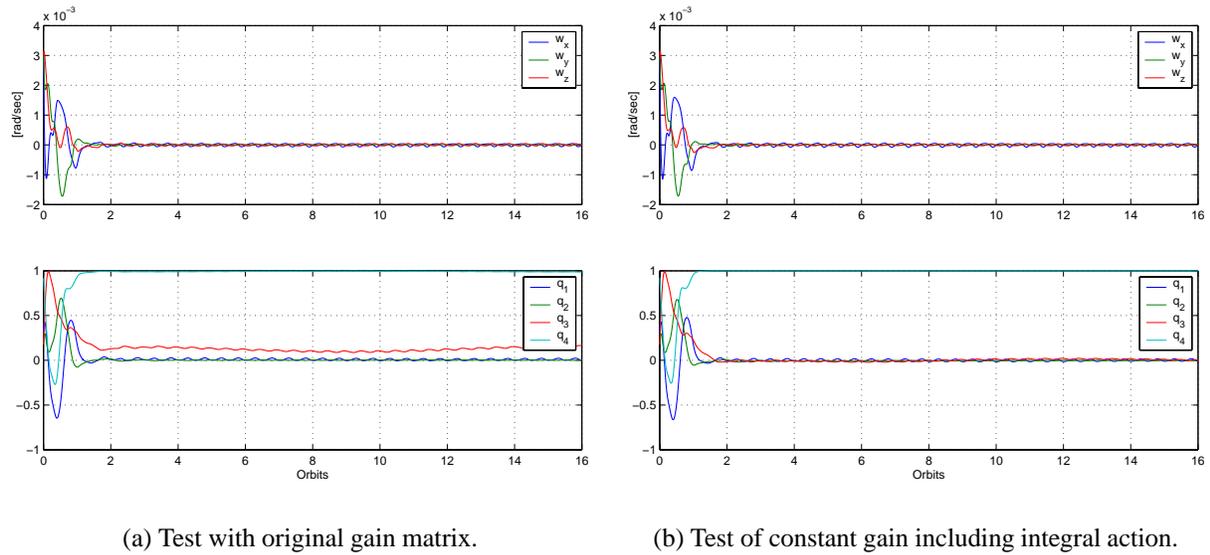


Figure 8.4: Test of constant gain controller with nonlinear model.

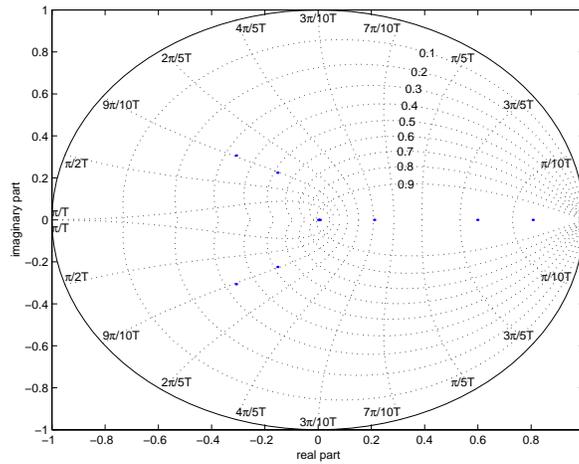
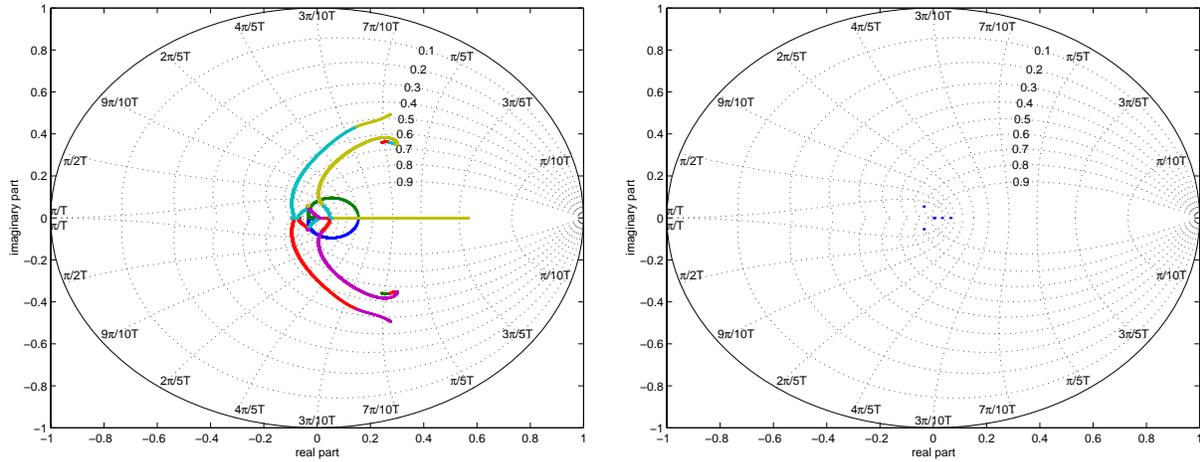


Figure 8.5: Plot of eigenvalues with integral state



(a) Plot of eigenvalues of the closed loop monodromy matrix, for $q = 1000 - 2000000$

(b) Plot of eigenvalues with best values of $q = 837500$.

Figure 8.6: Plot of eigenvalues of monodromy matrix for the inertial pointing model.

in figure 8.6(a), with the optimal eigenvalues in figure 8.6(b), where it can be seen that the system is stable.

From figure 8.7 it is shown that the optimal value of the scaling factor q is found to be 837500.

The gain matrix is tested using the initial offset, as used for the nadir pointing model. The results are shown in figure 8.8.

Simulations using the nonlinear model, and including the disturbance torques show that the controller is stable, but does not converge towards the reference. This non convergence is due to the lack of the cross coupling in the inertial pointing model. When the cross coupling is missing the system loses controllability.

8.2.4 Summary

The constant gain controllers have been implemented and tested, and the results have proved that the system can be controlled using a constant gain controller.

The inclusion of integral action with the nadir pointing model, have proved to help the system converge to the reference.

From the search of an optimal scaling factor q for the inertial pointing model, it is shown that the controllers implemented, may not be optimal. This indicates that the solution to the riccati equation which has been found, could be a solution which would minimize the performance function to a local minimum and not a global minimum.

To verify if there exist a more optimal solution, the control law will be implemented, by solving the LMI problem stated in section 7.5, and the results will be presented in the following.

With the results from the simulation of the constant gain inertial pointing model, it is chosen that the Nadir pointing model will be used in the implementation of the LMI based controller.

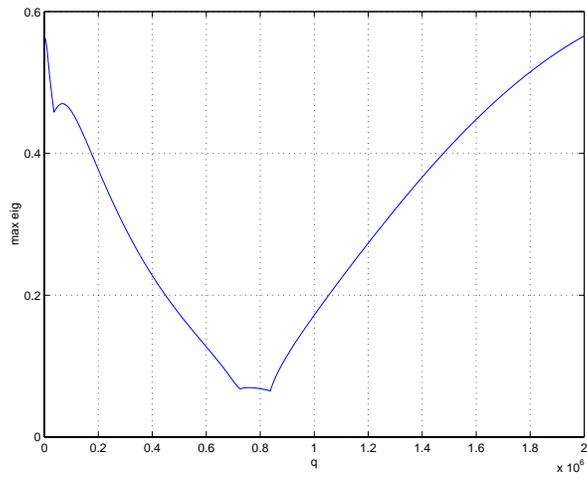


Figure 8.7: Search for best choice of scaling factor q for the inertial pointing model.

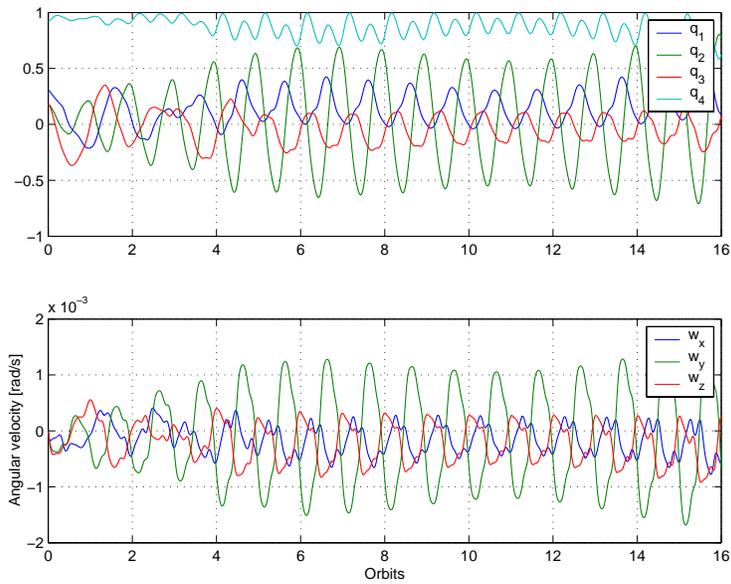


Figure 8.8: Test of constant gain controller with nonlinear model (Inertial pointing).

8.3 LMI controller

The LMI based controller has been set up using only the Nadir pointing model. First the periodic linear system is setup to fit the representation presented in section 7.5. As it was described, the system has full state information, so only the matrices \mathbf{B}_1 , \mathbf{C}_1 and \mathbf{D}_{12} have to be found. They are chosen to have the form as scaled identity matrices as described, so only the scaling of each matrix have to be found. The scaling factors have to be selected so the solution to the first step in the algorithm yields a periodic \mathbf{Q} matrix. More solutions may exist, and the best is found from a test with constant gain matrix. This matrix is found by solving the LMI problems, and tested with the nonlinear model to see which choice of scaling factor gives the best result.

Another thing that needs to be taken care of is the number of sample, an orbit is split into. If too many samples (more than 100) are chosen the algorithms in the LMI tool box becomes unstable, and the calculation towards the end of the period becomes incorrect.

8.3.1 Implementation of algorithm

The algorithm is implemented to fit both the time varying controller, and a constant gain controller. The only controller tested and implemented with the nonlinear model, is the constant gain controller. As it will be impossible to fit the periodic controller in the memory of the micro controller.

The first step in the algorithm, where the \mathbf{Q} matrix is calculated is setup as one big LMI system, with each step in orbit represented as a LMI. The result from the feasibility problem, will be used to find the optimal set of decision variables. The result from the optimization problem is checked, to verify if the values in \mathbf{Q} matrix are periodic. If this is the case step two in the algorithm is carried out.

Step two in the algorithm is implemented in two different ways. One where \mathbf{K} is calculated for each step, which gives the periodic gain matrix. The other way gives the constant gain matrix, in this case \mathbf{K} is kept constant, and the LMI is setup as a LMI system, with only one LMI variable.

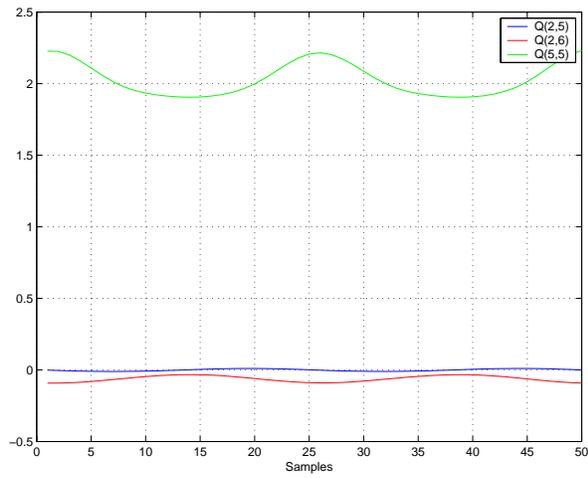
8.3.2 Results

The implementation describe above, with the \mathbf{B}_1 , \mathbf{C}_1 and \mathbf{D}_{12} having the following values:

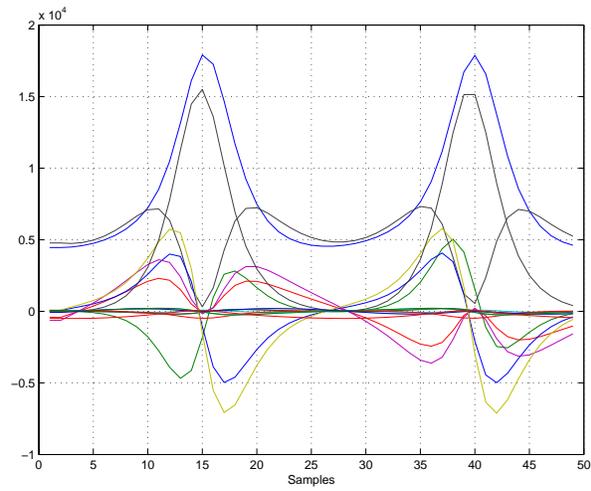
$$\mathbf{B}_1 = \begin{bmatrix} 0.001\mathbf{E} \\ \mathbf{0} \end{bmatrix} \text{ and } \mathbf{C}_1 = \begin{bmatrix} 0.001\mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{E} \end{bmatrix} \text{ and } \mathbf{D}_{12} = \begin{bmatrix} 0.001\mathbf{E} \\ 0.001\mathbf{E} \end{bmatrix} \quad (8.5)$$

With this system setup and 50 samples per orbit, the values of the \mathbf{Q} matrix gave a periodic result, examples of the values in the \mathbf{Q} matrix is given in figure 8.9(a). Based on the \mathbf{Q} matrix, the periodic gain was calculated, and the result is shown in figure 8.9(b). From the plot of the periodic gain, it can be verified that the gain vary periodically as expected.

The test of the constant gain with the nonlinear model, with the same initial condition that was used in the previous test, has shown, that the system behaves as expected. It converges to the reference within two orbits, the results from the simulation can be seen in figure 8.10.

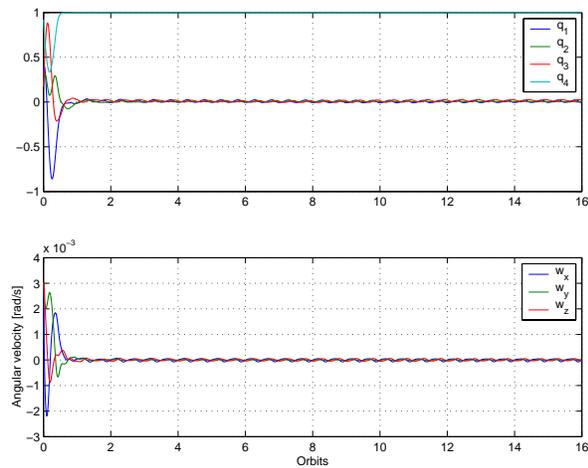


(a) Plot of $Q(2,5)$, $Q(2,6)$ and $Q(5,5)$.

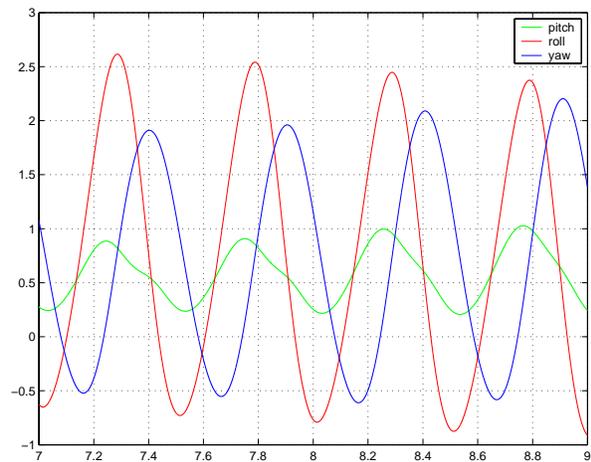


(b) Plot of periodic gain.

Figure 8.9: LMI based controller.



(a)



(b)

Figure 8.10: Results from simulation of LMI based constant gain controller.

8.3.3 Summary

Running the LMI design algorithm with different scaling factors on the \mathbf{B}_1 , \mathbf{C}_1 and \mathbf{D}_{12} matrices, and different numbers of samples, have shown that large numbers of samples makes the algorithms in the tool box unstable, and low numbers of samples fails to give a periodic solution.

The setup of the scaling factor also proved to influence the feasibility of the solutions to the LMI's significantly, and it was not possible to find any other setup of scaling factor, that would give a stabilising solution.

The results from the simulation with the non linear model have shown that the designed constant gain controller converges to the reference. Simulations with gain matrices calculated with other numbers of samples, show that a high number of samples yields the same tendency as the constant gain controller designed in the previous section. Low numbers of sample (30–40) have shown that the stability of the system increases with the number of samples.

8.4 Test of final controller

The results from the tests already performed, the difference between the constant gain controller with integral action, and the LMI based constant gain controller, was not very big. Both controller was able to converge to the reference, the LMI controller with a small offset though.

The final test will shown which controller is the more suitable choice for the CubeSat. The test will show the two controllers ability to track a reference different from the $[0001]^T$ quaternion.

The choice of the controller chosen for implementation will be based on how close the reference is tracked, if there is any offset from the reference and how fast the reference is achieved.

The test is performed over 32 orbits, with the same initial condition which have been used for the previous simulations. When the system has stabilised, the reference is changed, according to a slew of 20° roll. Then the reference is changed back to the origin, and finally a slew of -20° is performed. The results are shown in figure 8.11 and 8.12.

From this test the constant gain controller with integral action is chosen, as the one to be implemented. This controller has the advantage that it is able to track a reference without an offset. The LMI based controller has an offset error of 1-2 degrees, which is the problem the first tests showed. The LMI based controller is also slower than the constant gain controller, and it tends to make a large overshoot. The constant however has a problem, when it returns from a slew, it loses control of the yaw axis, and it takes about 8 orbits, before it is back at the reference. This is a minor problem, because error in the yaw axis, which is the camera axis is not a problem.

8.4.1 Summary

The final test show that the constant gain controller with integral action, is the suitable choice of controller for the CubeSat. It converges from the initial disturbed state to the reference within 2 orbits, and stays at the reference with an error less than 2° in pitch and roll, and a slightly bigger error in the yaw. The same size of error is seen when a slew is performed.

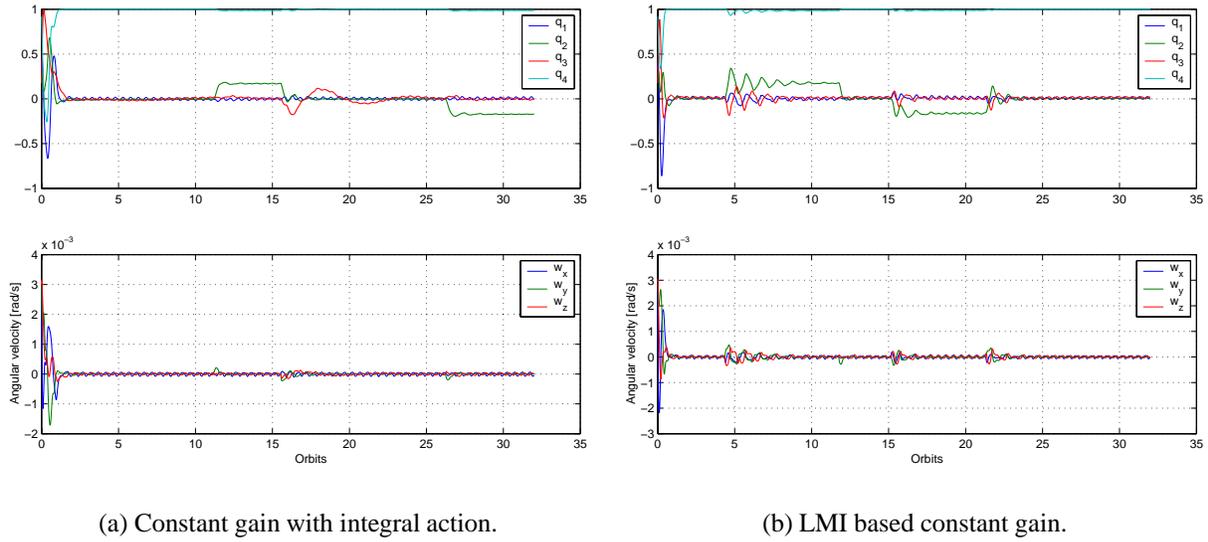


Figure 8.11: Plot of attitude quaternions and angular velocities.

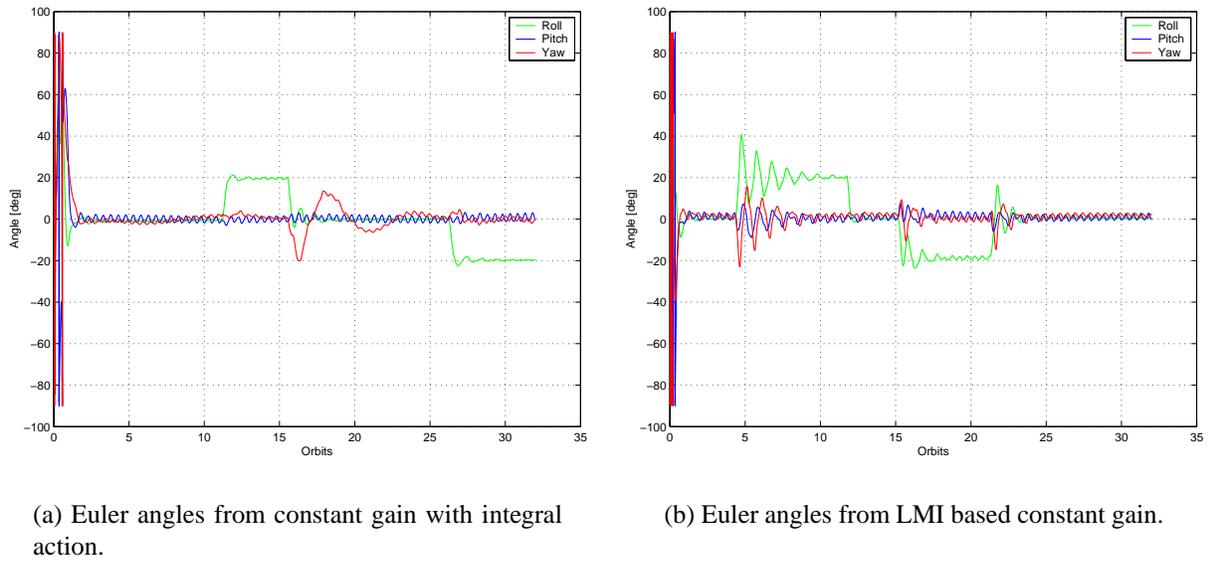


Figure 8.12: Plot of euler angles from final test.

The implementation of this controller on the micro processor is described in chapter 9.

chapter 9

Software design

In this chapter the software for both the micro processor and the on-board computer will be designed, and the implementation is described and partly tested. In the analysis and design all software parts will be considered. The implementation is made in cooperation with the ADS subsystem. The ADS group will be responsible for the software on the on-board computer and the ACS group will design the software for the PIC processor.

9.1 Interface overview

In the following the interface between the Command and Data Handling System (CDHS) and the ADCS subsystem will be described. This interface includes the supervisor (SPV), ADCS software on the on-board computer, and the ADCS software on the PIC processor.

The interaction between the CDHS and the ACS subsystem, is communicated via the I²C bus. The communication is described in section 9.3. The ADS software running on the on-board computer is implemented as a thread (ADC-task). The interface to the supervisor (SPV-task), running as the main thread on the on-board computer, is a message system.

In figure 9.1 the interface between the SPV and the ADCS system is shown.

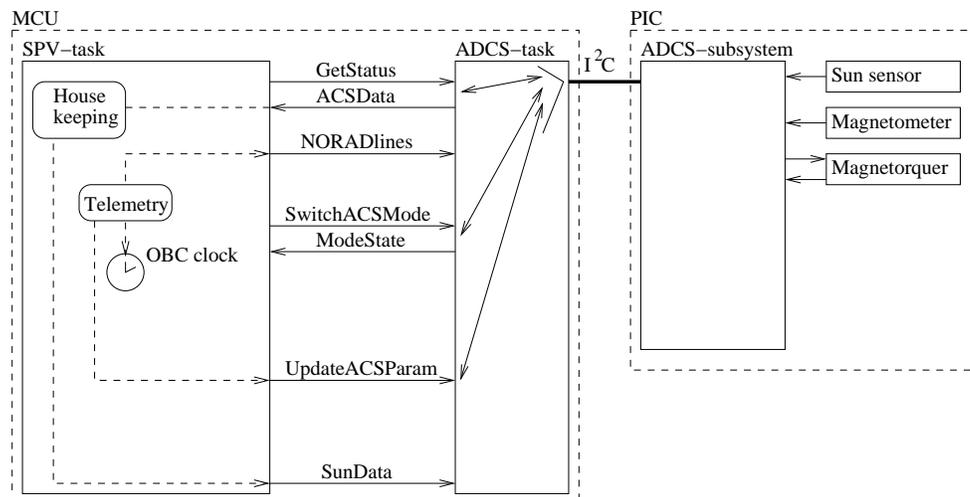


Figure 9.1: Interface between CDH and ACS system.

Due to the limited amount of memory available in the PIC micro processor (256 bytes), the attitude determination algorithms have been placed on the on-board computer. This leaves the sampling of sensors, control algorithms, current feedback form magnetorquers, mode switching and control of housekeeping to the ACS subsystem.

9.2 Mode switching

The mission modes described in section 2.4, represent the normal mission modes. In the software design three more modes OFF, External control and Initialization are considered. The modes are:

OFF	Power is off, or a reset has been issued.
Initialisation	All initialisation and test of the hardware is done.
Fail safe	The magnetorquers are deactivated and the sensors are sampled so housekeeping can be collected.
Detumbling	The controller in this mode will minimize the rotation of the satellite.
Power Save	The controller will point one corner toward the sun thereby optimizing the power delivered by the solar panels.
Camera mode	This controller will point the camera toward the reference.
External control	This mode makes it possible to control the magnetorquers externally.

All modes in the ACS subsystem are shown as a state diagram in figure 9.2, where the modes and transitions in the figure are described in table 9.1. The normal operation cycle, when the boot sequence has finished, is shifting between Power Save and Camera mode.

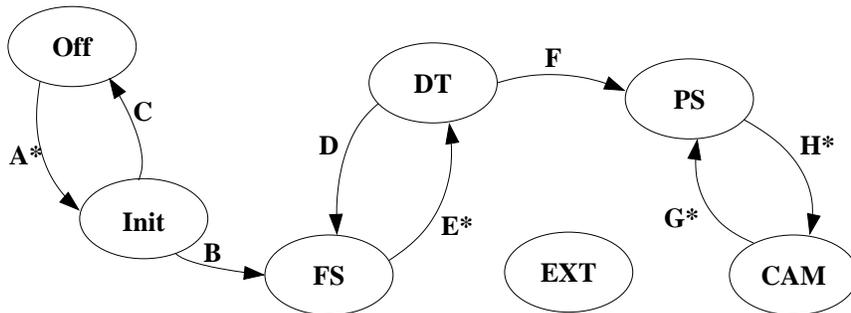


Figure 9.2: State transition diagram for the ACS subsystem.

9.3 I²C communication

The I²C bus is used to transfer housekeeping, system data and the data needed by the power save and camera controller. The data is given in table 9.2, where the module column is related

Mode	Name	Transition	Trigger
OFF	Power off/reset ^(Note 1)	A*	Power on
init	initialisation	B	Initialisation complete
FS	Fail safe	C	Initialisation failed
DT	Detumbling	D	Detumbling time out
PS	Power save	E*	Start detumbling
CAM	Camera	F	Detumbling completed successfully
EXT	External control	G*	Image taken
		H*	Initiate camera control
* externally triggered (CDHS ^(Note 2) /PSU ^(Note 3))			
Note 1: On failure any state may goto OFF (soft-reset)			
Note 2: CDHS may force ACS into any state			
Note 3: PSU may force ACS into OFF, with a HW-reset or power-off			

Table 9.1: List of modes, and state transitions.

to the I²C protocol described later. The module named housekeeping will contain data from the different sensors on the ADCS-subsystem and an error vector.

Update mode changes the mode in the PIC software. The algONOFF vector can be either updated or read.

The number given in the position column is related to the position in the I²C data array defined in the micro processor software (PIC-SW) and the name is related to the content of the variable. Some bytes are split up in two 4 bit variables to minimize memory consumption.

9.3.1 I²C protocol

The I²C protocol defined for the CubeSat has the structure as shown in figure 9.3 and 9.4, defining respectively a master write cycle and a master read cycle. The Address is the 8-bit address of the subsystem, with the least significant bit as the R/\bar{W} bit, the ACS address is 0x80.

The Chksum is calculated as the sum of all data transmitted including the module number. The Chksum is the given as the least significant byte of the sum, and the Read... and Write... represent reading and writing the data related to the chosen module.

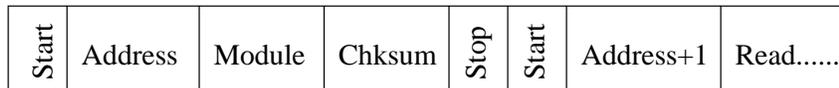


Figure 9.3: Master write cycle.

9.3.2 Error vector

The error vector is 16 bit where each bit indicate an individual hardware or software error see table 9.4. A logical “1” indicate an error. Bits 14 and 15 are reserved for further updates.

Module	Pos.	Name	Module	Pos.	Name		
3	0	0	0 (cont.)	23	Current Z	Mag Z	
		1		24	Temp 1	Sun 1	
		2		25	Temp 2	Sun 2	
		3		26	Temp 3	Sun 3	
		4		27	Temp 4	Sun 4	
		5		28	Temp 5	Sun 5	
		6		29	Temp 6	Sun 6	
		7		2	30	Q1_LSB	
		8			31	Q1_MSB	
		9			32	Q2_LSB	
		10			33	Q2_MSB	
		11			34	Q3_LSB	
		12			35	Q3_MSB	
		13			36	Q4_LSB	
		14			37	Q4_MSB	
		15			38	WX_LSB	
		16			39	WX_MSB	
		17			40	WY_LSB	
		18			41	WY_MSB	
		19			42	WZ_LSB	
	20	43	WZ_LSB				
	21	Current X	Mag X	1/4	44	algON/OFF	
22	Current Y	Mag Y					

Table 9.2: Data communicated via the I²C bus.

Module number	Module type
0	Housekeeping
1	Update algONOFF
2	Update reference
3	Update mode
4	Read algONOFF

Table 9.3: I2C module number

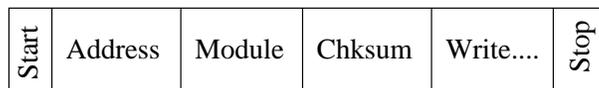


Figure 9.4: Master read cycle.

bit number	Error type
0	Coil X error
1	Coil Y error
2	Coil Z error
3	Magnetometer X error
4	Magnetometer Y error
5	Magnetometer Z error
6	Multiplexor 1 error
7	Multiplexor 2 error
8	Magnetometer Set/Reset error
9	Sun sensor error
10	ADC error
11	Reference 1.22 V error
12	Reference 2.5 V error
13	I ² C error
14	Not used
15	Not used

Table 9.4: Error vector

9.3.3 AlgONOFF vector

The algONOFF is an 8 bit vector table 9.5 where bits 5-7 that are reserved of updates. The coils can be deactivated in this vector, a coil will be deactivated if it is short circuited. The same is the case with the magnetometer Set/Reset circuit. It is also possible to change the algorithm that calculate magnetic field vector. This vector can be changed by either the PIC software routine or by the CDHS. In this vector “ON” and “Mean” (bit 3) is a logical “1”.

bit number	Error type
0	Coil X (ON/OFF)
1	Coil Y (ON/OFF)
2	Coil Z (ON/OFF)
3	Magnetometer algorithm (Mean/Difference)
4	Magnetometer Set/Reset (ON/OFF)
5	Not used
6	Not used
7	Not used

Table 9.5: AlgONOFF vector

9.3.4 PIC software architecture

The PIC software is split up into two interrupt routines and a main loop. The I²C communication is implemented as an interrupt routine. This interrupt routine will be executed each time the On Board Computer (OBC) attempts to communicate with the PIC. The other interrupt routine is handling actuation of the coils and sampling of the sensors at a given duty cycle. This interrupt routine is started by the internal clock at a frequency of 1220 Hz. The main loop controls which of the following modes are executed.

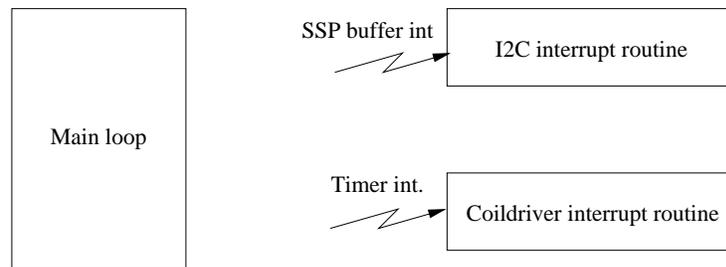


Figure 9.5: Program structure

9.3.5 Initialization mode

This mode starts with an initialization of the magnetometer, after this the I²C and coildriver interrupt routines are initialized. If the initialization is completed successfully different tests of the hardware are performed. If a hardware error is found, an error bit is set in the error vector see 9.3.2. First the magnetometer Set/Reset circuit is tested, this test is done by sampling the magnetometer 16 times and then taking the mean of the result. If the result is zero an error has been detected and the Set/Reset circuit is deactivated. In case of a fault the magnetic field will be obtained by sampling the magnetometer without the Set/Reset circuit. This will result in a less precise measurement.

Together with the magnetometer test the offsets on the current amplifiers are measured. The offsets will be used in coildriver tests in order to calculate the real power consumption of the coils. The coils and the coildriver will be tested for short circuits and disconnections. If a coil is short circuited the respective coil will be deactivated. When testing the coils the magnetometer is also tested. This is possible because the magnetometer is sensitive enough to measure the magnetic field generated by the coils. It is thereby possible to verify that both the magnetometer and the coils are working. The multiplexor and the 1.22 and 2.5V reference are tested together. This is done by sampling channel 7 and 8 on the multiplexor. The reference voltage will be determined faulty if the sampled voltage is $\pm 1\%$ from the desired reference.

If the initialization was successful the mode will change to failsafe mode otherwise the PIC will make a reboot.

9.3.6 Failsafe mode

In this mode the actuation is not allowed. Therefore the dutycycle of the coils is set to zero each time this mode is executed. Since housekeeping is still collected the sensors must still be sampled. The sensors are sampled with a period of one second.

9.3.7 Detumbling mode

The detumbling mode consist of a B-dot controller and a housekeeping sampling routine. The purpose of the B-dot controller is to minimize the angular velocities of the satellite. This controller will be used when the satellite is released form the P-POD or if the ACS-subsystem has been shutdown. The B-dot controller is needed to stabilize the satellite so the power save or camera controller can take over.

9.3.8 Camera control mode

This mode contains a controller and a housekeeping samplings routine. The controller is used to point the camera toward Denmark. The flight plan in the CDHS will decide when the camera mode is activated. When the picture is taken the CHDS changes the mode to power save mode. The pointing requirements for this controller are very high, because total error must be smaller than 8.3° in order to point the camera toward Denmark according to chapter 3.3. The flight plan also contains a pointing reference. The pointing reference is used to calculate the reference error. This reference error is calculated in the ADS thread and then send to the controller in the PIC processor.

9.3.9 Power save mode

Power save mode consist of a controller and a housekeeping sampling routine. The purpose of this controller is to maximize the power delivered by the solar panels. According to the power supply rapport [Bucelea et al., 2001] the maximum input power is obtained when the satellite is pointing a corner toward the sun. The power safe mode uses the same controller as in camera mode, the only difference is that the controller will stop actuating if the error is smaller than 15 degrees thereby saving power. The pointing error is calculated by the ADS thread and then send to the controller in the PIC processor.

9.3.10 External control mode

External control mode make it possible to control the coils form the CHDS. Thereby it possible to control the satellite form the CHDS. This mode will only be used if the implemented controller is faulty or if a better controller is to be tested. In this mode the sensors will still be sampled.

9.4 Implementation of software

The software has been implemented in ANSI C on a PIC micro processor (PIC16C774). The compiler that has been used is made by Custom Computer Services, INC [CCS, 2002]. The version of the compiler is 3.091. The software has been implemented as efficiently as possible in order to save memory and program ROM.

9.4.1 Implementation of the coildriver interrupt routine

The coildriver interrupt routine is interrupted by the internal clock. This internal clock is initialized at clock frequency of 1220 Hz. When entering the interrupt routine a counter is increased and the counter is reset when it reach 1220. The dutycycle is controlled by three global variables ($dyty_x, dyty_y, dyty_z$) and three global sign bits ($dyty_sign_x, dyty_sign_y, dyty_sign_z$). The dutycycle on the coils is generated by turning on the coils when the counter is equal zero and then turn the individual coils off when the counter is equal or greater the value that correspond to the dutycycle see figure 9.6. When sampling the data the coils are turned off to avoid corruption of magnetic field measurement. To avoid that the data is read by the I²C interrupt routine before it is fully updated the I²C interrupt routine is deactivated through out the sampling procedure. Otherwise the data that are read can be corrupted.

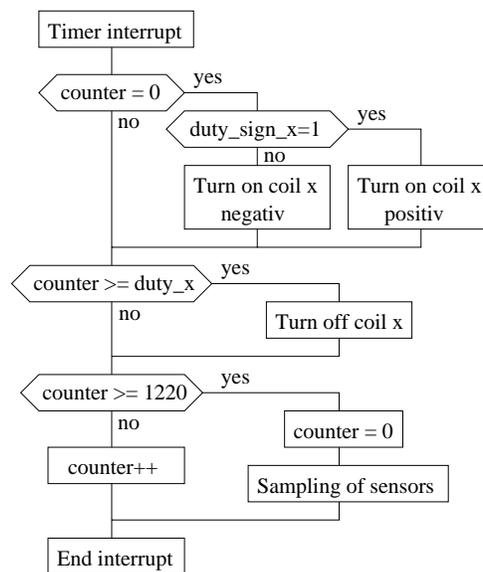


Figure 9.6: Coildriver interrupt service routine (only one coil illustrated)

9.4.2 B-dot implementation

Detumbling mode consist of a simple controller which have been optimized for speed and program ROM usage. The detumbling controller consist of a gain and a filter part see section 7.1 for

the derivation of the B-dot control. Figure 9.7 illustrate the implemented controller in simulink and in the PIC software.

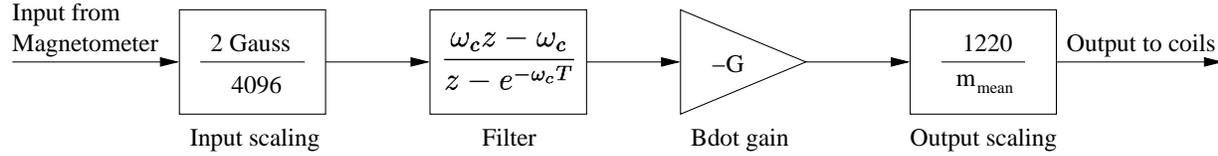


Figure 9.7: Implemented B-dot filter

The filter in equation 7.9 transformed to discrete time becomes:

$$\frac{c\dot{\mathbf{B}}(z)}{c\mathbf{B}(z)} = \frac{\omega_c z - \omega_c}{z - e^{-\omega_c T}} \quad (9.1)$$

where,

\mathbf{T} is the sampling time.

ω_c is the cutoff frequency.

The controller is implemented with a sampling frequency of 1 Hz. The implemented B-dot controller with input scaling, controller gain, and output scaling becomes:

$$u[k] = \frac{1220}{m_{max}(t)} G \omega_c \frac{4096}{2} (I[k] - I[k - 1]) + e^{\omega_c T} u[k - 1] \quad (9.2)$$

where, $m_{max}(t) = nI_{max}(t)A$

The implemented version of equation 9.2 uses 44% of the program ROM. Since there is not that much program ROM available for just the B-dot controller, equation 9.2 must be simplified. $m_{max}(t)$ will be approximated by a constant thereby simplifying the controller. $m_{max}(t)$ is approximated with a constant by taking the mean value of the maximum current in the coils through a full orbit. Since a thermal model of the CubeSat does not exist the mean coil current has been chosen at a temperature of $15^\circ C$.

Substituting the two constants K1 and K2 in equation 9.2 it becomes:

$$u[k] = K1 (I[k] - I[k - 1]) + K2 u[k - 1] \quad (9.3)$$

where,

$$K1 = \frac{1220}{m_{mean}} G \omega_c \frac{4096}{2}$$

$$K2 = e^{\omega_c T}$$

In chapter 8.1 the control gain (G) has been found to be $1.1 [Am^2/Gauss]$ with the cutoff frequency $\omega_c = 0.6 [rad/s]$ in the filter. With these values the constants become $K1 = 4.8653$ and $K2 = 0.5488$. This simplification minimized the implemented program ROM usage to only

24%. Because the PIC processor does not support fixed or floating point multiplication the B-dot controller can be simplified even further by changing the constants to $K_1 = 4$ and $K_2 = 0.5$. This simplification will change the control gain and the cut off frequency to $G = 0.775 [Am^2/Gauss]$ and $\omega_c = 0.693 [rad/s]$. With these constants simulations indicate that the total detumbled time is approximately 41 minutes slower than the optimal controller. But it still fulfill the design requirements in 4.2 of detumbling within 2 orbits (197 minutes) and rotations under $0.003 [rad/s]$. The B-dot controller used only 80 minutes. The final controller uses only 4% of the program ROM.

9.4.3 Camera controller implementation

The camera controller consist of a input and output scaling part, and a controller part. The input to the controller consist of a velocity vector Ω , a quaternion vector \mathbf{q} (without the real part) and a magnetic field vector \mathbf{B} (see figure 9.8). The velocity vector and the quaternion vector are calculated by the ADS-thread in the CHDS. The velocity vector consist of three 16 bit numbers and has a range between -0.1 to $0.1 [rad/s]$. This vector is scaled by the constant $K_1 = \frac{0.2}{2^{16}} = 305.2 \cdot 10^{-9} [rad/s]$. The quaternion vector consist of three 16 bit number and has a range between -1 to 1 . This vector is scaled by the constant $K_2 = \frac{2}{2^{16}} = 30.52 \cdot 10^{-6}$. The magnetic field is measured with the magnetometer and is scaled by the constant K_3 which have the value $K_3 = \frac{2 \cdot 10^{-4}}{2^{12}} = 488.3 \cdot 10^{-9} [T]$. The controller consist of at gain and a integrator part. The gain (\mathbf{K}_4) and integrator (\mathbf{K}_5) matrices are calculated in chapter 8.2. The gain part of the controller is calculated by multiplying the matrix \mathbf{K}_4 with the state vector \mathbf{X} . The integrator is calculated by multiplying the matrix \mathbf{K}_5 and the quaternion vector \mathbf{q} , and then integrate the result. The result form the gain controller (\mathbf{m}_{gain}) and the integrator controller (\mathbf{m}_{int}) is added which gives m_{total} . The crossproduct of the magnetic field and \mathbf{m}_{total} gives the output of the controller. The output to the coils is a value between -1220 and 1220 where 1220 is maximum power to the coils. The maximum power to the coils changes with respect to the current in the coils therefore the output of the controller is scaled by $K_6(t)$, where $K_6(t) = \frac{1220}{m_{max}(t)} = \frac{1220}{nI_{max}(t)A} [Am^2]$.

The controller shown in figure 9.8 consist of many multiplications, this requires a lot of program ROM, which is why a simplification of the controller would be preferable. The input scaling and the control gain can be combine by multiplying the controller gain with a diagonal scaling matrix where K_1 is the first three diagonal elements and K_2 is the last three elements. The same can be done with the integrator gain, here K_2 and \mathbf{K}_5 is multiplied together. The controller can be simplified even further by combining the output scaling gain (K_6) and magnetic field input scaling gain (K_3) with the controller gains (\mathbf{K}_4 and \mathbf{K}_5). This is possible because the crossproduct does not change the length of the vector. The simplified controller can be seen in figure 9.9. Where $\mathbf{G}_1 = K_6 K_3 \mathbf{K}_4 \begin{bmatrix} K_1 \mathbf{E} & \mathbf{0} \\ \mathbf{0} & K_2 \mathbf{E} \end{bmatrix}$ and $\mathbf{G}_2 = K_6 K_3 \mathbf{K}_5 K_2$

9.4.4 Implementation of external controller

The CDHS can control the coils by first sending a reference update and then change the mode to external control. The reference update contains the dutycycle of the coils in the position of

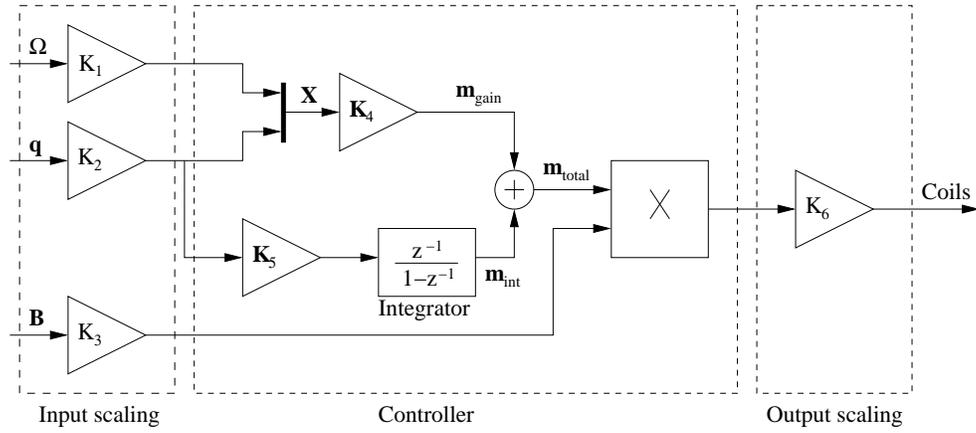


Figure 9.8: Implemented gain controller with integrator

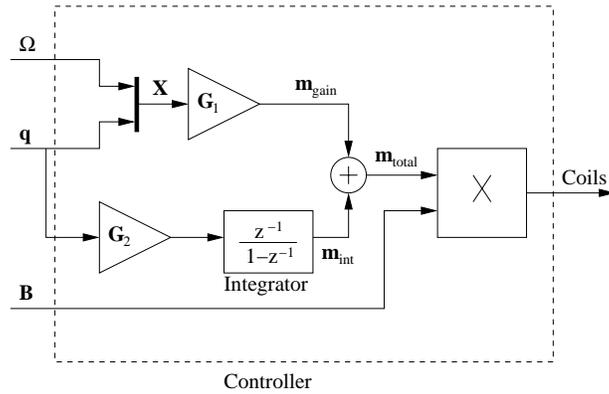


Figure 9.9: Simplified gain controller with integrator

quaternions see table 9.6. The mode must be changed to external control each time a new duty cycle is available in the I²C data.

Byte number	Name
30	Coil _X _LSB
31	Coil _X _MSB
32	Coil _Y _LSB
33	Coil _Y _MSB
34	Coil _Z _LSB
35	Coil _Z _MSB

Table 9.6: I²C data coil reference

9.4.5 I²C interrupt routine implementation

The I²C interrupt routine is woken up each time the CDHS sends or request one byte. This makes the I²C interrupt routine very complex and therefor only the functionality of the interrupt routine will be presented.

When the CDHS request the housekeeping, the housekeeping data and checksum will be transmitted after this a time synchronization between the ADCS-task and the ADCS-subsystem is done. The time synchronization is minimizing the delay between the time when the data sampled and the time when the data is read. The time difference is synchronized, by testing if the counter in the coildriver interrupt routine is greater than 100 in case of this the counter will be subtracted by 100. Time synchronization is implemented in order to minimize the phase difference between the input and output signal of the controller. The maximum time delay will be only 100ms. On figure 9.10 implemented modules can be seen. The CCS compiler used to compile the source code was found to have a error in the “i2cread” command. The error was that the “i2cread” command did not generate the acknowledge when it was suppose to. After a lot of test the error was corrected by applying a few extra lines of assembler code into the C-code.

9.5 Test of implemented software

The implemented software will be burned into the ROM in the PIC processor. Thereby the software can not be changed in space and must work the first time. This makes it very important to test the different sub programs. The following test are planned but some are not yet performed.

9.5.1 Test of sensor interface

The sensor interface will be tested by checking data in the housekeeping vector if it correspond to the signal applied to the sensors. The temperatures sensors are tested by applying three different known temperatures to all the sensors. The temperature used in this case are -44, 0 and 100 °C.

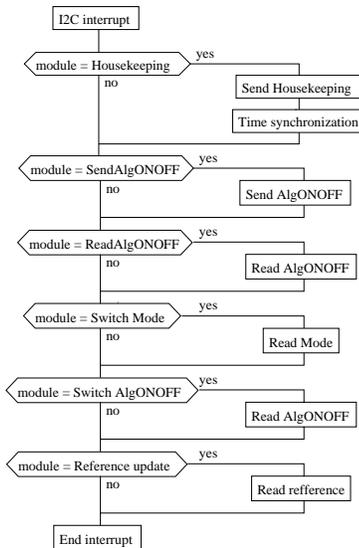


Figure 9.10: I2C interrupt service routine

The sun sensors are tested by applying two known light intensity to all the sensors. In this test total darkness and a light that correspond to sun light in space were used. The magnetometer was tested by rotating the sensor on a horizontal non magnetic platform. The magnetic field vector will form a circle when plotting all the sampled data. These tests were performed by the ADS group [Krogh and Schreder, 2002].

9.5.2 Test of the implemented controllers

The implemented controllers are tested if they work like the ones in simulink. The test is split up into two test one for the B-dot controller and one for the camera controller. The difference between the two test is that the state vector is not used when testing the B-dot controller. The test is performed by first run a simulation in simulink where the input and output vectors of the respective controller are saved into two files. The simulation will be performed through out two orbits with a time step of 1 second, where the data will be saved for all the time steps, this is illustrated in figure 9.11.

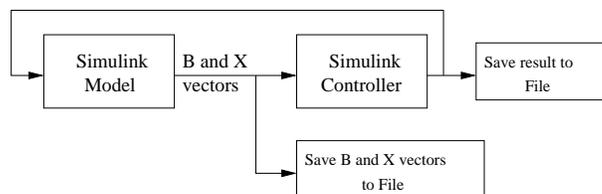


Figure 9.11: Test of the camera controller part I

The file containing the input to the controller will be loaded by a small DOS and then sent to the PIC processor via the I²C bus, where the input is applied to the controller. The result from

the controller is sent back via the I²C bus and then saved to a file. This process is repeated until all the input vectors has been transmitted, this is illustrated in figure 9.12. The output files from the implemented controller and the simulink controller is compared to check whether there is a significant difference between the result in the files. The B-dot and camera controllers have not yet been tested on the PIC.

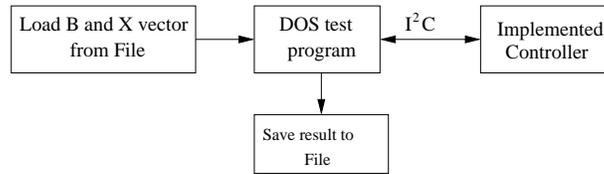


Figure 9.12: Test of the camera controller part II

9.5.3 Test of mode changing

This test is a test of all the implemented software and the I²C communication. The test is a burn in test where the PIC software will be tested for a longer period of time and will indicate whether the I²C communication will work for longer period of time without making any communication errors. The test will indicate to some degree if the implemented software contains any errors. This test will be performed with a small DOS program that will change the software mode like if it were the CDHS in space. This test is not made yet but test programs indicate that the mode can be changed.

The external control test is tested with a small dos program that changes the duty cycle of the coils. The test was performed by applying a duty cycle with the dos program to the PIC via the I²C bus. The duty cycle is checked with a oscilloscope if the correct duty cycle is applied to the coils. The test was performed and the software functioned as expected.

chapter 10

Conclusion

The design of the attitude control system for the AAU CubeSat, was made as a part of the CubeSat project at AAU. The scope of this thesis was the attitude control system, which was to be analysed and designed. First the system was analysed, and different actuator configurations was considered. Then specifications for the entire attitude determination and control system was defined, before the development of the attitude hardware, software and control algorithms was started.

Requirements

The requirements for the ADCS system was set up based on the requirements from other subsystems, such as the power supply unit and the structural design. This put a lot of constraints on the design considerations.

The mission for the satellite was a great factor in definition of the requirements for the control algorithms. The constraints is mainly due to the fact that the satellite is supposed to take images of Denmark. It was calculated the a pointing accuracy of approximately 8° was necessary, for a pass directly over Denmark. Whereas a pass in a distance of 900 km would require a pointing accuracy of approximately 2° . A pass 900 km away from Denmark would require a slew, roll, of 56° . These numbers were used as guide lines in the development of the control algorithms.

Hardware design

There were a lot of constraints in the design of the hardware. A limited amount of power was available, and the space for actuators, was also a problem that needed a lot of attention. It was considered that the optimal choice of actuator for the CubeSat would be three magnetorquers and a momentum wheel. The use of a momentum wheel was discarded, because the design would be too complicated, with the space available. The final choice of actuators was three magnetorquers.

Interface circuitry had to be design, to meet the power requirements, and be able to communicate with the on board computer. Magnetorquers were designed and constructed to fulfill the requirements. The magnetorquers which was designed, turned out to be too power consuming. This error has not been addressed, because it is not considered a major problem, as the actuator are oversized, and will newer work at their maximum.

A lot of effort was put into the design and prototyping of the entire ADCS hardware. This included all from design and testing, to the development of the printed circuit board for the engineering model. Finding and placing the components needed turned out to be a huge task.

All hardware has been tested on the prototype, and it is working. The engineer model, is manufactured, but not tested.

Control design

The control design considered for the CubeSat is a constant gain approximation of a periodic system. The design of a controller of this kind, required derivation of a nonlinear model, which has also been linearised about two set points. The linearised model chosen for implementation is a Nadir pointing model, as the inertial pointing model, was found to have a lack of controllability.

The online computational burden played a significant role in the choice of controllers considered to be investigated. The simpler choice is an approach, where the periodic system is approximated by the mean of a period. This kind of system was analysed using Floquet theory, and a stabilising controller was found.

While testing this controller with the nonlinear model it was found that it did not converge to the reference, but had an offset in the pitch axis. This problem was overcome by including integral action in the controller. Analysis using Floquet theory of the system with integral action, proved that the systems was stable.

One other approach to designing a controller was implemented. The method was based on the linear matrix inequality representation of the H_2 optimal control synthesis. Results from the synthesis, proved that the system could be stabilised with a constant approximation of the periodic gain, from the LMI analysis. Tests with the nonlinear model, showed that the system converged towards the reference with a small offset on the pitch axis. There was also some trouble when a reference different from the origin was given to the system.

Based on the results from the simulation of the controllers, with the nonlinear model. The constant gain controller including integral action was chosen for implementation in the attitude software.

Software design

The design of the software turned out to be a complicated task, as the micro processor found feasible for the system, had a very limited amount of program RAM and ROM.

Not only the control algorithms was to be programmed, but also interface functions to sample the sensors, communicate with the on board computer and handling different mission modes, had to be implemented.

All software has been written, but only part of it has been completely tested. The sampling of sensor, including actuating the magnetorquers, have been implemented and tested, with an extra feature, which enables control of the magnetorquers from the on board computer, in case the control algorithm is not working properly. The functions for handling communication with the on board computer has been implemented and tested, with a test interface, but not with the command and data handling system. Handling of different mission modes have also been implemented and

tested. The main task that still need some work is, testing and final implementation of the control algorithms.

Summary

The goal of the project, was to analyse and design the control system for the AAU CubeSat, and fulfill the requirements defined by other subsystems, and the mission.

The design of the control system has almost been completed. Hardware is designed and working, control algorithms are designed and working, only software needs the final implementation of the control algorithms and testing with the other subsystems, which have not yet been possible.

Simulations of the control algorithms with nonlinear model of the satellite, have proved, that the required pointing accuracy is achievable with a pass directly over Denmark.

appendix A

Momentum wheel

The limited space available inside the CubeSat makes it a complicated task to design a momentum wheel. Most of the space along the X-axis is taken up by the camera and the lens system, and the placement of other equipment makes it impossible to place a conventional electric motor along this axis.

This has placed some unusual requirements on the design of the motor. It has been chosen to design the motor as a brushless permanent magnet motor. It has to be brushless, since mechanical brushes would shorten the life span significantly. The rotor of the motor will consist of permanent magnets in order to achieve the highest momentum, by constructing it of the heaviest material. A sketch drawing of the motor is shown in Fig A.1, and it works like a stepping motor.

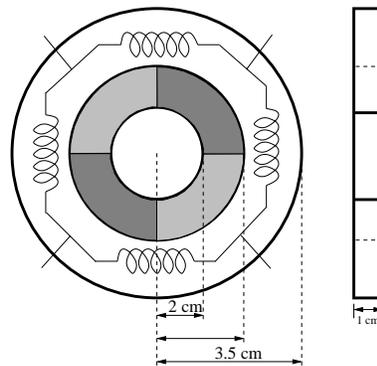


Figure A.1: Sketch of motor

The motor must be able to contain the generated magnetic field in order not to disturb sensitive electronics inside the satellite, especially the magnetometer. The design criteria for the motor are shown in Table A:

Voltage	5 V
Nominal Power	50 mW
Maximum Power	200 mW
Stored Torque	2.66 mNms
Mass	45 g

Table A.1: Design criteria.

The power and mass constraints have been given by the power supply and structural design teams respectively. The stored torque have been calculated from the following formulas [Larson and Wertz, 1992].

First the disturbance torque, T_D , is defined. According to appendix B the gravity gradient torque T_G , is by far the most significant disturbance, hence:

$$T_D = T_G = \frac{3\mu}{2R^3} | I_Z - I_X | \sin(2\theta) \quad (\text{A.1})$$

where

μ is the earths gravity constant.

R is the orbit radius [m].

θ is the maximum allowable motion of the Z-axis from local vertical [rad].

I_Z and I_X are moments of inertia about the Z and X axes [Nm].

When sizing a momentum wheel one approach is to estimating the angular momentum, H , is to integrate the worst case disturbance, T_D , over a full orbit. When the disturbance is gravity gradient the maximum disturbance accumulates over a quarter of an orbit. Therefore the following rule of thumb can be applied:

$$H = \frac{T_D(P/4)}{\theta_a} \quad (\text{A.2})$$

where

P is the orbit period [s].

θ_a is the maximum allowable motion [rad].

Inserting values into equations A.1 and A.2 yields:

$$T_D = \frac{3 \cdot 3,986 \cdot 10^{14}}{2 \cdot (700 \cdot 10^3)^3} | 1.8 \cdot 10^{-3} - 1,9 \cdot 10^{-3} | \sin(2 \cdot 0.035) = 3.048 \cdot 10^{-9} [Nm] \quad (\text{A.3})$$

where

θ is chosen to $2^\circ = 2 \cdot \frac{\pi}{180} = 0.035$ rad.

$$H = \frac{3.048 \cdot 10^{-9} \cdot (5927/4)}{0.0017} = 2.66 \cdot 10^{-3} [Nm \cdot s] \quad (\text{A.4})$$

where θ_a is chosen to $0.1^\circ = 0.0017$ [rad].

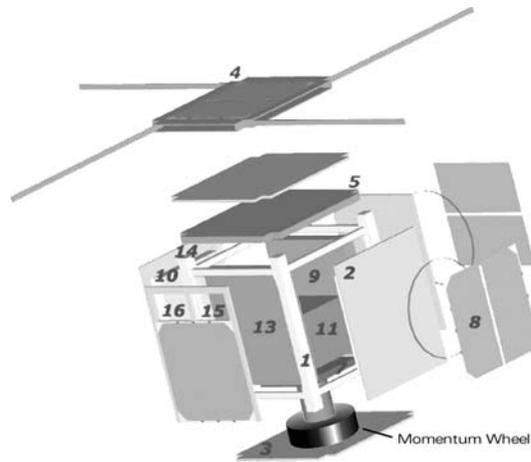


Figure A.2: Placement of momentum wheel

The motor will have to be placed in the inside the CubeSat as shown in figure A.2. The space provided by the structural design team, is a cylindric space with a hole in the middle. The diameter of the hole is 40 mm.

The constrains on the motor is that the weight must not excite 45 g. The weight of the rotor will be designed to be approximately 30% of the total weight, this gives the rotor a maximum weight of 13.5 g.

With the above constraints a motor has been designed using Autocad. This motor is shown in figure A.3. The rotor has a inner diameter of 46 mm and outer diameter of 54 mm and the width is 10 mm. Using the mass densities of aluminium for the rotor and iron for the permanent magnets, the mass of the rotor is found to be 12.7 g, which is below the limit of 13.5 g. The outer diameter of the stator is 66.82 mm and the width of the two halves is 5 mm. the combined weight of the stator is found to be 17.3 g using the mass density of aluminium, calculated with Autocad.

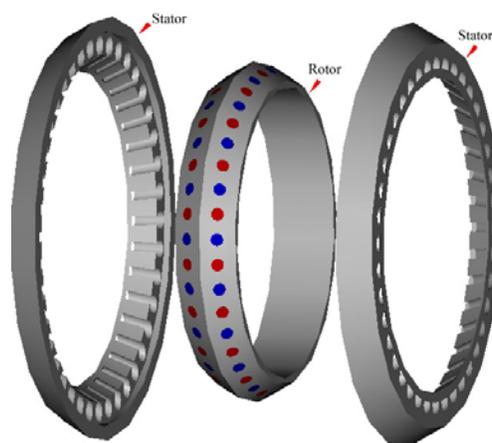


Figure A.3: Motor shape.

The moment of inertia of the rotor was calculated by Autocad and is $7.59 \cdot 10^{-6}$ Nm. Given this and equation A.4 the necessary rotational speed can be found using the following equation:

$$\omega \approx \frac{H}{I} = \frac{2.66 \cdot 10^{-3}}{7.59 \cdot 10^{-6}} = 350.46 \text{ [rad/s]} \approx 3350 \text{ RPM} \quad (\text{A.5})$$

This rotational speed is found feasible, however at point the work was discontinued due to resistance from the structural design team, who feared that the design would end up being too heavy when including magnetic shielding and mounting.

This fear might well have been justified considering that the total weight of the design already is 29.7 g not including the stator coils. However even if the motor could be designed within the given weight limit, the manufacture of it would still be complicated.

appendix B

Description of external disturbance torques

In this chapter the external disturbance torques will be described, this includes the type of disturbance, the source of the disturbance and the characteristics of the disturbance. The momentary disturbance torque, and the accumulated angular momentum over an orbit. At the end of the chapter the total disturbance torque and accumulated angular momentum will be calculated.

Disturbance overview

In general three different types of disturbance torque acts on the satellite. The type, source and characteristic of each disturbance is listed in table B.1 [Larson and Wertz, 1992].

Type	Source	Characteristic
Aerodynamic	Earth atmosphere	Drag and lift on the satellite. Torque is cyclic for inertial pointing satellite.
Gravity gradient	Distribution of inertia	If the satellite has off-diagonal inertia elements it will produce a body fixed torque. The torque will be cyclic for inertial pointing satellite.
Solar radiation	The Sun	The torques and forces cause by the sun are harmonics of the orbit rate, and dependent on the satellite geometry and the surface reflectivity. The torque is constant for a solar oriented satellite.

Table B.1: External disturbance torques

Aerodynamic drag

In a low earth orbit a satellite is subject to an aerodynamic drag torque, due to atmospheric molecules. The collision of the satellite and the molecules produce a torque about the satellite CoM. The aerodynamic torque acting on a satellite composed by n surface elements is described by [Larson and Wertz, 1992]:

$$\mathbf{N}_{drag} = \frac{1}{2} C_D \rho V^2 \sum \mathbf{r}_{s,i} \times (\hat{\mathbf{n}}_i^T \hat{\mathbf{V}}) \hat{\mathbf{V}} A_i \quad (\text{B.1})$$

where,

C_D is the drag coefficient (usually between 2 and 2.5)

ρ is the atmospheric density

$\mathbf{r}_{s,i}$ is a vector from the satellite CoM to the i 'th surface element A_i

$\hat{\mathbf{n}}_i$ is the outward normal of A_i

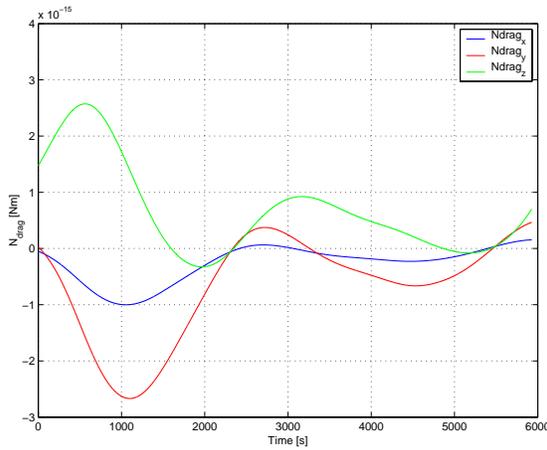
$\hat{\mathbf{V}}$ is the unit vector in the direction of the translational velocity

V is the length of the vector $\hat{\mathbf{V}}$

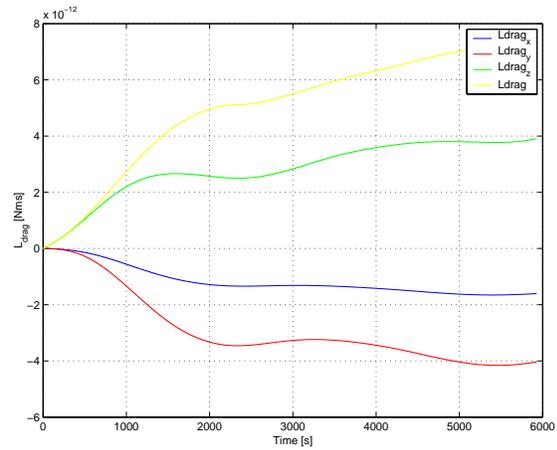
The aerodynamic drag is mainly influenced by the orbit altitude, the satellite structure and the location of CoM.

To get an idea of the influence of the aerodynamic drag torque, a simulation of one orbit is performed, where both the momentary drag torque and accumulated angular momentum is calculated.

In figure B.1 the results of the simulation is shown. In figure B.1(a) the aerodynamic disturbance torque in the body frame is shown, and in figure B.1(b) the accumulated aerodynamic disturbance torque over one orbit is shown. Over one orbit the drag torque accumulates to 7.567 pNm.s.



(a) Momentary drag torque.



(b) Accumulated angular momentum.

Figure B.1: Momentary aerodynamic drag torque, and accumulated angular momentum over one orbit.

Gravity gradient

The gravity gradient torque results from the inverse squared gravitational force field. A satellite that is not perfectly symmetric will be subject to a gravitational torque due to variations in the earth gravitational force over the satellite. The gravity gradient torque is given by [Larson and Wertz, 1992]:

$$\mathbf{N}_{gg} = \frac{3\mu}{R_{cm}^3} \left[\hat{\mathbf{R}}_{cm} \times \mathbf{I} \hat{\mathbf{R}}_{cm} \right] \quad (\text{B.2})$$

where,

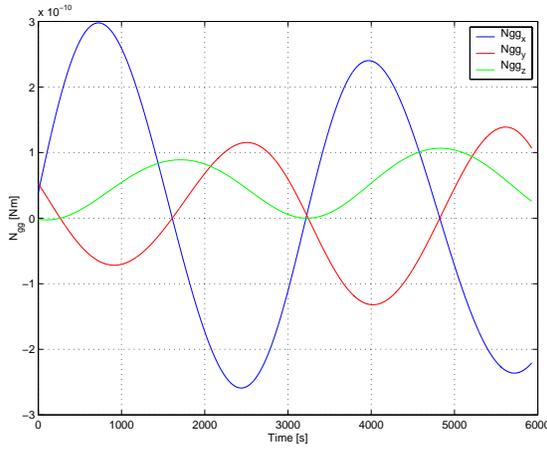
μ is the earth gravitational constant

$\hat{\mathbf{R}}_{cm}$ is a unit vector in direction of Nadir

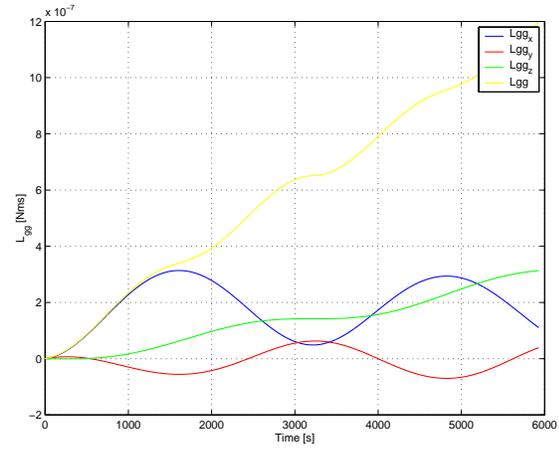
\mathbf{I} is the satellite principal moments of inertia

To get an idea of the influence of the gravity gradient torque, a simulation of one orbit is performed, where both the momentary gravity gradient torque and the accumulated angular momentum is calculated.

In figure B.2 the results of the simulation is shown. Figure B.2(a) shows the gravity gradient torque in the body frame, and figure B.2(b) shows the accumulated gravity gradient torque over one orbit. Over one orbit the gravity gradient torque accumulates to $122.45 \mu\text{Nms}$.



(a) Momentary gravity gradient torque.



(b) Accumulated angular momentum.

Figure B.2: Momentary gravity gradient torque, and accumulated angular momentum over one orbit.

Solar radiation

In an low earth orbit the surface of a satellite is subject to a radiation pressure, which results in a torque about the satellite CoM. The main factors in determining the radiation torque are the intensity and spectral distribution of the incident distribution, the geometry of the satellite, the optical properties, and the orientation of the sun relative to the satellite. In the model of the solar

radiation torque it is assumed that the radiation is completely absorbed. The solar radiation can be expressed as [Larson and Wertz, 1992]:

$$\mathbf{N}_{solar} = \frac{F_s}{c} C_a \sum \mathbf{r}_{s,k} \times \left(\hat{\mathbf{n}}_k^T \hat{\mathbf{S}} \right) \hat{\mathbf{S}} A_k \quad (\text{B.3})$$

where,

F_s is the solar constant

c is the speed of light

C_a is the absorption coefficient

$\mathbf{r}_{s,k}$ is the vector from the satellite CoM to the k'th surface element A_k

$\hat{\mathbf{n}}_k$ is the k'th outward surface normal

$\hat{\mathbf{S}}$ is the unit vector from the satellite to the sun

The primary influence on the solar radiation torque is the satellite structure, the surface reflectivity, and the satellite CoM.

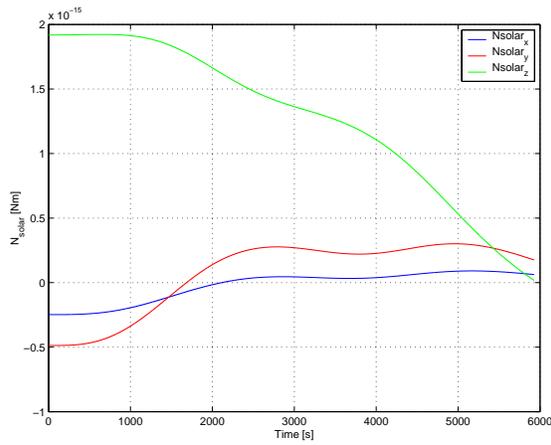
To get an idea of the influence of the solar radiation torque, a simulation of one orbit is performed, where both the momentary torque and accumulated angular momentum is calculated.

In figure B.3 the results of the simulation is shown. Figure B.3(a) shows the solar radiation torque in the body frame, and figure B.3(b) shows the accumulated solar radiation angular momentum over one orbit. Over one orbit the solar radiation angular momentum accumulates to 7.824 pNms.

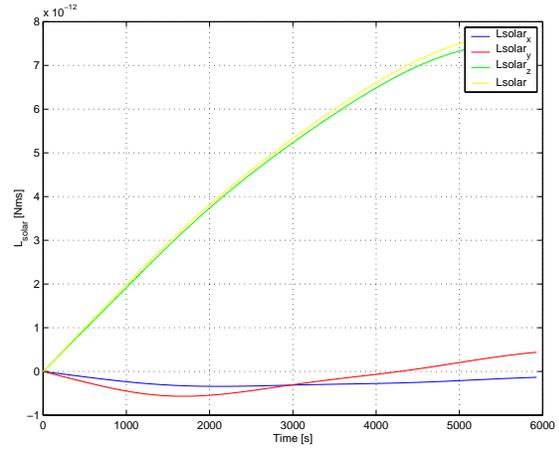
Total disturbance

In figure B.4 the total disturbance torque and accumulated angular momentum over one orbit is shown.

From the plots in figure B.1 to B.3 it is seen that the gravity gradient torque is by far the most significant torque acting on the satellite.

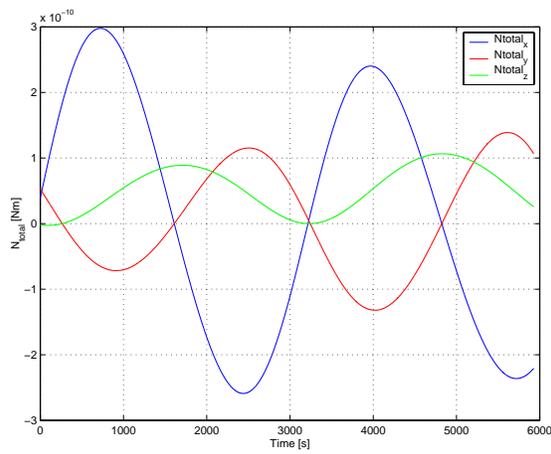


(a) Momentary solar radiation torque.

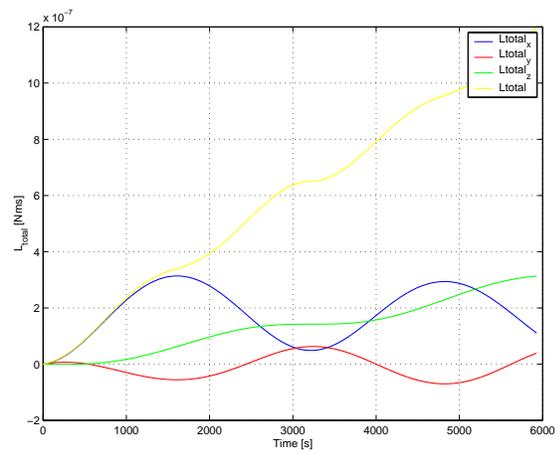


(b) Accumulated solar radiation angular momentum.

Figure B.3: Momentary solar radiation torque, and accumulated angular momentum over one orbit.



(a) Disturbance torque in one orbit.



(b) Accumulated angular momentum over one orbit.

Figure B.4: Total disturbance torque, and accumulated angular momentum over one orbit.

appendix C

Quaternions

This appendix describes the properties of quaternions, and is mainly based on [Wertz, 1978].

The quaternion representation of rigid body rotations leads to a convenient expression of kinematics. The representation is singularity free, as the quaternion is a four dimensional mapping of a 3-D space.

The four parameters $[q_1 \ q_2 \ q_3 \ q_4]^T$ form the components of the quaternion \mathbf{q} as follows

$$\mathbf{q} \equiv iq_1 + jq_2 + kq_3 + q_4 \quad (\text{C.1})$$

where i, j and k are the hyperimaginary satisfying the following conditions

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

The first three components, $\mathbf{q} = [q_1 \ q_2 \ q_3]^T$, form the vector part of the quaternion and q_4 is the scalar part. The conjugate or inverse of \mathbf{q} is defined as

$$q^* \equiv -iq_1 - jq_2 - kq_3 + q_4 \quad (\text{C.3})$$

The norm of a quaternion is given by

$$|\mathbf{q}| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} \quad (\text{C.4})$$

The unit quaternion can be interpreted as a combination of the components of the unit vector and the angle of rotation as

$$\begin{aligned}
q_1 &= e_1 \sin\left(\frac{\phi}{2}\right) \\
q_2 &= e_2 \sin\left(\frac{\phi}{2}\right) \\
q_3 &= e_3 \sin\left(\frac{\phi}{2}\right) \\
q_4 &= \cos\left(\frac{\phi}{2}\right)
\end{aligned} \tag{C.5}$$

$$\tag{C.6}$$

Quaternions provide simple methods for calculation of successive rotations. As an example the quaternion reflecting the rotation of the control CS in the orbit CS, and the quaternion describing the transformation from the inertial CS to the orbit CS is given as

$${}^c\mathbf{q} = \mathbf{R}({}^o\mathbf{q}) {}^c\mathbf{q} \tag{C.7}$$

where

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ -q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \tag{C.8}$$

The relation between the direction cosine and the attitude quaternion is also useful. As an example the direction cosine matrix from the orbit CS to the control CS is given as

$${}^c\mathbf{A} = [{}^c\mathbf{i}_o \ {}^c\mathbf{j}_o \ {}^c\mathbf{k}_o] \tag{C.9}$$

where ${}^c\mathbf{i}_o$, ${}^c\mathbf{j}_o$, ${}^c\mathbf{k}_o$ are the unit vectors of x, y and z axes of the orbit CS, projected on the control CS.

The unit vectors may be parametrized by the attitude quaternion ${}^c\mathbf{q}$ as

$$\begin{aligned}
{}^c\mathbf{i}_o &= [q_1^2 - q_2^2 - q_3^2 + q_4^2 \quad 2(q_1q_2 - q_3q_4) \quad 2(q_1q_3 + q_2q_4)]^T \\
{}^c\mathbf{j}_o &= [2(q_1q_2 + q_3q_4) \quad -q_1^2 + q_2^2 - q_3^2 + q_4^2 \quad 2(q_2q_3 - q_1q_4)]^T \\
{}^c\mathbf{k}_o &= [2(q_1q_3 - q_2q_4) \quad 2(q_2q_3 + q_1q_4) \quad -q_1^2 - q_2^2 + q_3^2 + q_4^2]^T
\end{aligned} \tag{C.10}$$

appendix D

Hardware schematic diagram

appendix E

Printed circuit board layout

In this chapter the board layout of the ADCS subsystem will be described, this includes the design requirement given by the Structural Design Team and the ADS group. At the end a conclusion about the board status will be represented.

E.0.4 Requirements for the board

The Structural Design Team made a detailed technical drawing on how the board should be constructed. The drawing contained a description on where the mounting holes for both the circuit board and the batteries should be. The design space was restricted because the batteries will be mounted on one side of the board, see figure E.1. The mounting of the sun sensors also made a small restriction on the design space. The height of the components were restricted to 7 mm on one side and 13 mm on the side with the batteries.

The ADS group made a request for the DC-DC converter to be placed at a distance from magnetometer, to avoid magnetic disturbance. The track between the magnetometer and the DC-DC converter must be thick because of the high current peaks of up to 1 A to the set/reset circuit.

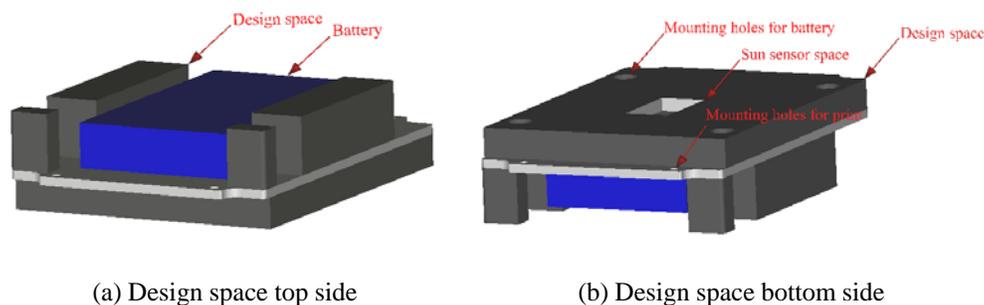
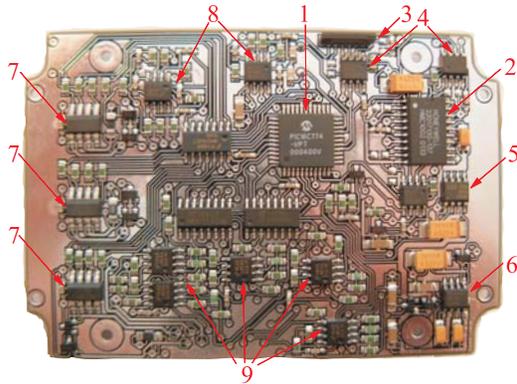


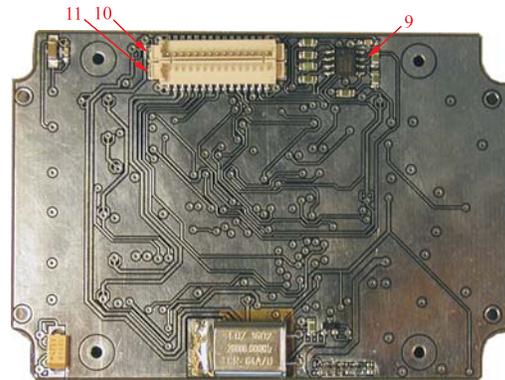
Figure E.1: Design space

E.0.5 Summary

The final board was constructed successfully by Sieker print see figure E.2. Unfortunately the board design contained small errors. The first error was that the mounting holes were miss placed



(a) The print seen form top



(b) The print seen form bottom

Figure E.2: The ADCS board seen form top and bottom

Number	Name
1	PIC micro processor (PIC16C774)
2	Magnetometer 1
3	Magnetometer 2
4	Magnetometer amplifier
5	Hex fet
6	DC-DC converter
7	H-bro coildriver
8	Current amplifier
9	Sun/temperature sensor interface
10	Connector 1
11	Connector 2

Table E.1: Board description

by 0.6 mm. The other error was that the crystal short circuited some of the tracks below it. This error was corrected by adding insulation tape under the crystal. Some of the capacitors were missing a plus sign on the silk overlay. These errors have been corrected and the design is ready for the final board. The restrictions made by the Structural Design Team turned out not to be a problem, since the highest components on the board were the capacitors, which uses only 3.1mm. The available space on the board did cause some small problems, but because the board was routed by hand, it turned out fine.

The board has not yet been tested. This must be done before the board for the flight model is constructed.

appendix F

Manufacturing of magnetorquers

This appendix describes the construction of the magnetorquers designed in chapter 6. The thin wire and the high number of turns made it necessary to construct the coils by hand. It was first considered to design a machine for the job, but it was decided to be too time consuming a task, because of the high degree of precision required in order for the coils not to exceed the physical constraints placed on them by the structural design team. Preliminary tests also showed the wire to be very fragile, and great care needed to be taken when winding the coils, so that the wire wouldn't break. In order to ease the manufacture of the coils a simple scaffold was created to support the wire drum. An axis with a revolution counter was mounted in parallel. On this axis it was possible to mount a frame for the coils. See figure F.1.

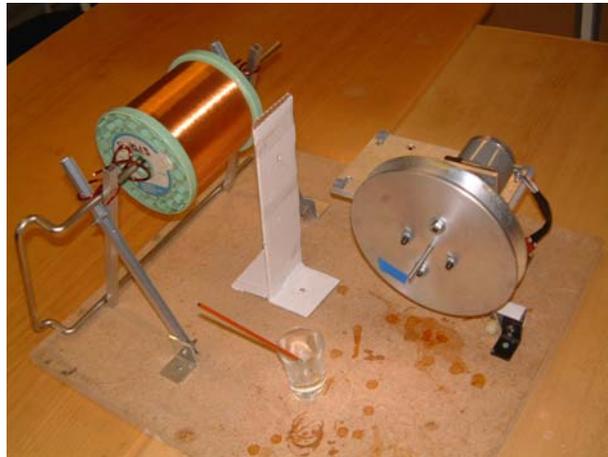
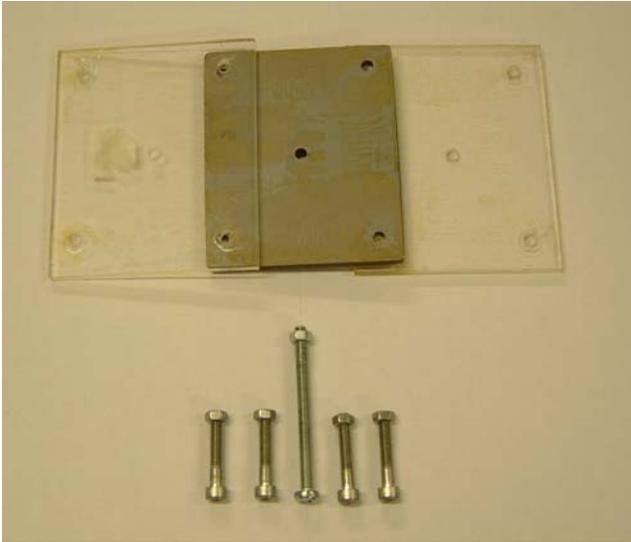


Figure F.1: Coil winding setup

A suitable frame for winding the coils needed to be constructed. The test frames were designed as a simple sandwich construction of three plates. A center plate made of plywood with the same thickness, height and width as the hole in the coils, and two larger transparent plates made of plexiglas, see figure F.2.

The test coils were held together with Shellac. This quickly turned out to be a problem, since the coils would not let go of the frames, and were destroyed if they were forced off. It was discovered that the Shellac would not bind to baking paper, and so the test frame was covered with baking paper. This however made it impossible to maintain the required precision.



(a)



(b)

Figure F.2: Test frame

A new and more accurate frame was needed for the construction of the coils for both the engineering and flight model. The structural design team offered to manufacture an aluminium frame with the exact dimensions of the panels on the CubeSat, see figure F.3. They were however concerned about the use of Shellac and offered to supply a small amount of NASA approved epoxy¹.

Before the manufacture of a coil could start the frame had to be treated with a special slip coat, that would prevent the epoxy from sticking to it. The slip coat had to be applied at least five times with 10 min drying time in between. When the last coat had been applied, the slip coat needed 30 min to cure before the frame could be used. However if cured for longer or at higher temperature the slip coat would last for several releases. Therefore the it was cured at 80°C for 45 min. The slip coat could then be reused 2-3 times.

Before the first turn of the coil was wound, a coat of epoxy was applied to the bottom of the frame with a small brush. When a layer was complete (about 30 turns), a new coat of epoxy was applied. This process was repeated until the desired number of turns was reached. A final coat of epoxy was then applied to the topside of the final layer. The winding of one coil took about 2.5 hours.

The epoxy required 4 hours to harden at room temperature, but only 45 min at 120°. The frame and coil was therefore placed in an oven and heated to 100° for 1 hour. In order to prevent

¹The epoxy and slip coat is manufactured by Micro Joining AB of Sweden. The product ID of the epoxy is U330-2.



(a)



(b)

Figure F.3: Aluminium frame supplied by the Structural Design Team.

the excess epoxy from collecting at one side of coil and thereby forming edges outside the design space, the frame was rotated continuously, while the epoxy hardened.

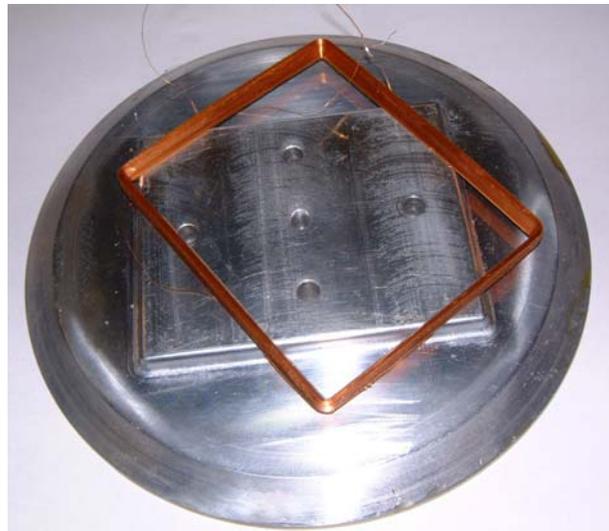


Figure F.4: Finished magnetorquer.

The finished magnetorquers all weighed between 17.7 g and 17.9 g which is well within the design criteria of a maximum weight of 23.3 g. The process of constructing one magnetorquer from start to end took about 5 hours. Because only one aluminium frame was available only one magnetorquer could be constructed at a time.

appendix G

NORAD Lines and the SGP4 orbit Model

NORAD Lines

NORAD (North American Aerospace Defense Command) is a joint defense organization between the United States and Canada and tracks all air traffic in north America using several ground and air based radar stations. NORAD also has the capability to track objects in orbit around the Earth. Using ground based electro-optical sensors NORAD can track objects as small as a basket ball in orbits as high 36000 Km. This capability is used to track all known man-made satellites and keep updated orbit parameters for these. Information on these parameters for all non classified satellites are made public in the form known as Two Line Elements (TLE) or NORAD lines.

NORAD lines have the following format:

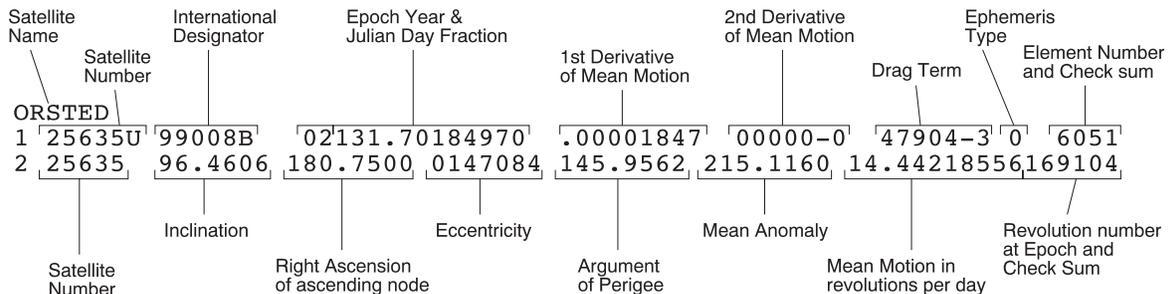


Figure G.1: Format of NORAD line.

A couple of the fields might need a little more explanation. Field 1.2, Satellite Number, is the NORAD Satellite Catalog number of the satellite.

Field 1.11, BSTAR drag term, is the aerodynamic term, B^* . In aerodynamic theory all objects have a ballistic coefficient, B , which is defined as the cross product between the coefficient of drag and the cross sectional area of the object divided by the mass:

$$B = \frac{(C_D \times A)}{m} \quad (G.1)$$

The ballistic coefficient shows how susceptible an object is to drag. The higher the number the more susceptible the object is to drag. B^* is an adjusted value of B that takes in to account the atmospheric density, P_0 :

$$B^* = \frac{BP_0}{2} \quad (\text{G.2})$$

B^* has units of $(\text{earth radii})^{-1}$. The value in field 1.11 can be either that of B^* or B depending on the orbit model used. B^* is used by the SGP4 orbit model and B is used by the SDP4 model.

Fields 1.10 and 1.11 have a somewhat different format than the other fields. In particular, they use a modified exponential notation with an implied leading decimal point. The first six columns of each field represent a decimal fraction less than 1 and the last two represent the exponent. For example, the value -12345-6 corresponds to $-0.12345 * 10^{-6}$. Each of these two fields can be blank, corresponding to a value of zero.

Field 1.12, Ephemeris type, shows which orbit model is used to generate the TLE. However this field is only used for internal analysis and distributed TLE's have a value of zero in this field. Normally the TLE's are generated with either the SGP4 or the SDP4 orbit model. To determine which model to use field 2.8, Mean Motion, can be used. For values greater than 6.4 the SGP4 model should be used and for values less or equal to 6.4 the SDP4 model should be used.

Field 1.13, Element Number, is the number of the current TLE. Ideally this number would be incremented each time a new TLE was generated. However switches between primary and backup space control centers some times lead to numbers being skipped or reused.

Field 1.14, Checksum, The checksum is generated by simply adding all the numbers in the line, ignoring all the letter, periods, plus sign and assigning the value 1 to all minus signs. The checksum is then the last digit of the sum.

Field 2.9, Revolution number at epoch, is the number of the current orbit at epoch. Starting from zero when the satellite is first launched the revolution number is incremented each time the satellite passes the ascending node.

The SGP4 Orbit Model

The SGP4 model is used to model the orbit of LEO satellites, which includes the CubeSats. The SGP4 model was developed in 1970 by Ken Cranford. The model is used to predict the position and velocity of a satellite and incorporates perturbations due to:

- Atmospheric drag.
- Forth-order zonal geopotential harmonics (J2, J3 and J4).
- Spin-orbit resonance effects for synchronous and semi-synchronous orbits.
- Solar and lunar gravitational effects to the first order.

Input	Example
Epoch [YYDDD]	02304 (01 NOV 2002)
Epoch [JD]	2452579,43200
First derivative of Mean Motion	0
Second derivative of Mean Motion	0
Drag	0
Inclination	98.16
Right	63,756298
Eccentricity	0,00105
Argument of Perigee	90
Mean Anomaly	0
Mean Motion	14,5788769

Table G.1: Input to orbit model.

In this project the model will be considered a black box, that takes the input shown in table G.1. For simulation purposes it is necessary to construct all these inputs from what is known about the orbit of the satellite before launch. That is a solar synchronous orbit with local time 01:30 and an orbit height of 700 km.

Bibliography

- [Boulet, 2002] Boulet, B. (2002). Linear systems. URL: [www.cim.mcgill.ca/ boulet/304-501A/304-501A.htm](http://www.cim.mcgill.ca/boulet/304-501A/304-501A.htm).
- [Bucelea et al., 2001] Bucelea, V., Lazar, R. D., chiubna, T., Loidi, A., and Formanek, L. (2001). Optimised design of power supply for cubesat satellite. Technical report, Aalborg University. URL: <http://www.cubesat.auc.dk/dokumenter/psu.pdf>.
- [CCS, 2002] CCS (2002). Pic c-compiler. URL: www.ccsinfo.com.
- [Conoly, 2000] Conoly, R. (2000). The p-pod payload planer's guide. Technical report, OSSS.
- [Gahninet et al., 1995] Gahninet, P., Nemirovski, A., Laub, A. J., and Chilali, M. (1995). *LMI Control Toolbox*. The Mathworks Inc.
- [Krak, 2002] Krak (2002). Krak kort. URL: www.krak.dk.
- [Krogh and Schreder, 2002] Krogh, K. and Schreder, E. (2002). Attitude determination system. Technical report, Aalborg University. URL: <http://www.cubesat.auc.dk/dokumenter/ads.pdf>.
- [Larson and Wertz, 1992] Larson, W. J. and Wertz, J. R. (1992). *Space mission analysis and design*. Microcosm inc. ISBN: 0-7923-1998-2.
- [Mohler, 1991] Mohler, R. R. (1991). *Nonlinear Systems*. Prentice Hall. ISBN: 0-13-623489-5.
- [Scherer and Weiland, 1999] Scherer, C. and Weiland, S. (1999). Lecture notes disc course on linear matrix inequalities in control. Technical report, Delft University of Technology. URL: www.er.le.tue.nl/SWeiland/lmi.htm.
- [Serway, 1982] Serway, R. A. (1982). *Physics*. Saunders College Publishing. ISBN: 0-03-015654-8.
- [Simiconductors, 2002] Simiconductors, N. (2002). Operational amplifier datasheet (lmc6462aim). URL: www.national.com.
- [Wertz, 1978] Wertz, J. R. (1978). *Spacecraft attitude determination and control*. Kluwer academic publisher. ISBN: 90-277-1204-2.
- [Wisniewski and Stoustrup, 2001] Wisniewski, R. and Stoustrup, J. (2001). Generalized h_2 control synthesis for periodic systems with application to magnetic attitude control. Technical report, Aalborg University.

[Zetex, 2002a] Zetex (2002a). H-bridge datasheet (zhb6790). URL: www.zetex.com.

[Zetex, 2002b] Zetex (2002b). Zener diode datasheet (zrc250f01). URL: www.zetex.com.