



NTNU – Trondheim
Norwegian University of
Science and Technology

Design of Attitude Estimation and Control System for a Cube Satellite

Fredrik Sola Holberg

Master of Science in Engineering Cybernetics

Submission date: July 2012

Supervisor: Jan Tommy Gravdahl, ITK

Co-supervisor: Esten Grøtli, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

MASTER THESIS

Design of Attitude Estimation and Control System for a Cube Satellite

Fredrik Sola Holberg

Spring 2012

Master's Committee:

Advisors:

Professor J.T. Gravdahl

Postdoc Esten Ingar Grøtli

Project Manager:

Roger Birkeland



Department of Engineering Cybernetics

Abstract

The NUTS (NTNU Test Satellite) is a satellite being built in a student CubeSat project at the Norwegian University of Science and Technology. The project was started in September 2010 as a part of the Norwegian student satellite program run by NAROM (Norwegian Centre for Space-related Education). The NUTS project goals are to design, manufacture and launch a double CubeSat by 2014. As payload an IR-camera observing waves in the air-glow layer is planned, as well as a short-range RF experiment. The satellite will fly two transceivers in the amateur radio bands. Final year master students from several departments are the main contributors in the project and most of the system components are designed and built by students.

Described in this paper is an assesement of different strategies for selecting and designing the Attitude Determination and Control System, ADCS, for the NUTS satellite, effectively providing the next iteration from the work that was conducted last year, placing much emphasis on the special requirements of a system that can not be repaired or maintained in a traditional way.

Different options for sensors and actuators are explored and evaluated, looking for tried and tested products with a low chance of failure and concluding that Analog devices should be a safe and reliable provider of the inertial measurement units for our satellite.

While the Extended Kalman Filter is the filtering method that is probably the most used in almost all applications, it has some very serious limitations that are especially important in such a demanding environment as a cube satellite. Two other methods for attitude estimation are evaluated, namely the Extended QUaternion ESTimator and newly developed nonlinear observers. While they both are very attractive options with many properties that fit with our requirements, there are still some questions as to wether they are as reliable as the proven EKF or not.

The open source electronic prototyping platform Arduino is gaining foothold with hobbyists all over the world, and it is explored if it can also be used for rapid prototyping and if it is suitable for frequent design iterations in our satellite project. The freedom it provides coupled with a standard breadboard seems to make it a very good choice for any project needing to frequently change or improve it's design according to other hardware or software considerations.

Preface

This report is the concluding thesis of my Master of Science degree at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology. I would like to thank both my supervisors during this project, professor J.T. Gravdahl and E.I. Grøtli for the helpful input. The rest of the satellite group, lead by project manager Roger Birkeland, have been a great source of motivation and help during the work on this thesis and I wish to thank them all for the things I learned from them along the way.



MSc thesis assignment

Name of the candidate: Fredrik Sola Holberg

Subject: Engineering Cybernetics

Title: Design of Attitude Estimation and Control System for a Cube Satellite

Background

The NUTS (NTNU Test Satellite) project was started in September 2010. The project is part of the Norwegian student satellite program run by NAROM (Norwegian Centre for Space-related Education). The projects goal is to design, manufacture and launch a double CubeSat by 2014.

Assignment:

- 1) Evaluate different methods for attitude estimation for the cubesat, both sensors and algorithms.
- 2) Investigate if a setup using an Arduino and a breadboard can be used for rapid prototyping and design iterations for the ADCS board.
- 3) Further improve on the ADCS prototype previously made, to accommodate for three actuators and analog sensor input from solar cells.

To be handed in by:

Trondheim,

Esten I. Grøtli
Post doc, supervisor

Contents

| | | |
|-------|--|----|
| 1 | Introduction | 8 |
| 1.1 | Motivation | 8 |
| 1.2 | Previous work | 8 |
| 1.3 | Thesis outline | 9 |
| 2 | Theory, definitions, notations | 11 |
| 2.1 | Coordinate frames | 11 |
| 2.1.1 | Earth-centered Inertial | 11 |
| 2.1.2 | Body | 11 |
| 2.1.3 | Orbit | 12 |
| 2.2 | Attitude representations | 12 |
| 2.2.1 | Rotation matrix | 12 |
| 2.2.2 | Euler angles | 13 |
| 2.2.3 | Unit quaternions | 14 |
| 2.3 | Estimation methods | 15 |
| 2.3.1 | Extended Kalman Filter | 15 |
| 2.3.2 | QUEST | 16 |
| 2.3.3 | EQUEST | 17 |
| 2.3.4 | Further development of the EQUEST method | 17 |
| 2.3.5 | Nonlinear Observers | 18 |
| 2.4 | Mathematical modeling | 18 |
| 2.4.1 | Satellite model | 18 |
| 3 | Sensors and actuators | 20 |
| 3.1 | Sun sensor | 20 |
| 3.1.1 | Solar panels | 20 |
| 3.1.2 | Dedicated sun sensors | 22 |

| | | |
|-------|---|----|
| 3.2 | Gyroscope and magnetometer | 23 |
| 3.2.1 | Gyroscope | 23 |
| 3.2.2 | Magnetometer | 23 |
| 3.2.3 | Sensor choice | 23 |
| 3.3 | GPS | 24 |
| 3.4 | Magnetic coils as actuators | 24 |
| 3.4.1 | International Geomagnetic Reference Field | 25 |
| 4 | Estimator considerations | 26 |
| 4.1 | Extended QUaternion ESTimator | 26 |
| 4.2 | Extended Kalman Filter | 27 |
| 4.3 | Nonlinear Observers | 27 |
| 5 | Software | 29 |
| 5.1 | EQUEST | 29 |
| 5.2 | Extended Kalman Filter | 30 |
| 5.3 | Nonlinear observer | 30 |
| 6 | Prototyping | 31 |
| 6.1 | Arduino | 31 |
| 6.2 | Rapid prototyping with Arduino | 31 |
| 6.2.1 | Fritzing | 32 |
| 6.3 | H-bridge | 34 |
| 6.4 | ADC and solar panels | 34 |
| 6.5 | Magnetic coils | 38 |
| 6.6 | Programming the microcontroller | 38 |
| 6.6.1 | Using Arduino with Atmel Studio | 38 |
| 6.6.2 | ADC | 38 |
| 6.7 | Prototype board | 40 |
| 7 | Simulations and tests | 43 |
| 7.1 | Reading the solar panels | 43 |
| 7.2 | H-bridge output | 43 |
| 7.3 | Discussion | 47 |
| 8 | Recommendations for further improvements to the ADCS module and concluding remarks | 48 |
| 8.1 | Sensors | 48 |
| 8.2 | Estimation algorithm | 49 |

| | | |
|-------|---|----|
| 8.3 | Board design | 49 |
| A | Conference submissions | 50 |
| A.1 | The 25th Annual AIAA/USU Conference on Small Satellites, Utah State University | 50 |
| A.1.1 | Poster Submission | 51 |
| A.2 | The 4th European Cubesat Symposium, Ecole Royale Militaire Brussels | 52 |
| A.2.1 | Abstract | 52 |
| A.2.2 | Poster Submission | 53 |
| B | Using Arduino with AVR Studio | 54 |
| B.1 | Setting up external tools to flash Arduino . | 54 |
| C | C-code | 57 |
| D | Norsk sammendrag | 62 |

List of Figures

| | | |
|-----|---|----|
| 2.1 | Orbit frame | 12 |
| 3.1 | Comparison of various IMUs | 24 |
| 4.1 | Experimental results presented by T.I. Fossen et. al. | 28 |
| 6.1 | Arduino MEGA front | 32 |
| 6.2 | Arduino and breadboard with H-bridge | 33 |
| 6.3 | SN754410 with heat sink | 35 |
| 6.4 | Pin map and function table for the SN754410 H-bridge | 36 |
| 6.5 | Fritzing generated graphics showing h-bridge and solar panel wiring | 37 |
| 6.6 | Pin map Arduino MEGA vs ATmega 2560 | 39 |
| 6.7 | REFS1:0 bits in the ADMUX register | 40 |
| 6.8 | ADC reading | 41 |
| 7.1 | Arduino test bed with solar panel | 44 |
| 7.2 | PWM output with one full H-bridge | 45 |
| 7.3 | PWM output with two full H-bridges in parallel | 46 |
| B.1 | External tools in Atmel Studio | 55 |
| B.2 | Atmel Studio 6 toolbar: build “Release” | 56 |

Chapter 1

Introduction

1.1 Motivation

The NTNU Test Satellite, NUTS, is a double CubeSat project that was started in the fall of 2010 and aims to be ready for launch by 2014. The satellite includes both an IR-camera as payload and radio antennas that are to some degree dependent on the Attitude Determination and Control Systems, from here referred to as ADCS throughout the text, working as intended. While all subsystems are critical for operation of the satellite, the ADCS and radio are important because they need to work at least to a certain extent if we are to be able to establish contact with the satellite to determine whether any of the other subsystems works at all. In other words we have no way of knowing if the other subsystems works or not if radio contact can not be established with the satellite. Either if the radio system fails or the satellite is tumbling with great rotational velocity because of ADCS malfunction, the success criteria of the satellite will be impossible to determine.

1.2 Previous work

The ADCS of our satellite will be the result of incremental improvements that have been made since the project was started, and this is where we stand so far.

Last year a new method for attitude estimation based on QUaternion ESTimation (QUEST) was developed by members of our project group. The Extended QUEST (EQUEST) method enables the fusing of vectorized and non-vectorized measurements making it more suitable for attitude estimation. Simulations showed that for cubesats, which often uses magnetorquers for both attitude estimation and control, in addition to the often limited computational power, the EQUEST method offers several advantages over the well known Extended Kalman Filter, and it was suggested to go forwards with this algorithm. [24]

There was also some work done on the control part of the ADCS [22] looking at magnetic coils as actuators and some investigation into control algorithms. It was found that a simple 8-bit, 16MHz microcontroller had sufficient processing power to continuously compute the geomagnetic field using the complex International Geomagnetic Reference Field model while simultaneously maintaining correct coil actuation.

Additionally two projects has been done on the control system, looking at optimal and nonlinear control respectively. These themes will be further expanded in other reports, with the final decision on control algorithms and coil dimensions still open. [13][4]

In the previous term the EQUEST method was further developed to be more coherent in a mathematical sense. The original method made use of subtractions between estimated and measured quaternions, which does not result in new attitude error quaternions. A new optimization problem was found that made use of quaternion products in place of subtraction which was more computationally demanding, but resulted in a more accurate mathematical model. [18]

1.3 Thesis outline

Chapter 2 introduces some theoretical background for principles that are used throughout the text.

Chapter 3 presents evaluations of available sensors and actuators that is most suitable for use in ADCS of a CubeSat such as ours.

Chapter 4 contains evaluations of 3 different attitude estimation algorithms, which all provide distinct pros and cons relative to each other.

Chapter 5 discusses differences in software implementations between the algorithms presented in chapter 4.

Chapter 6 describes and expands on various considerations regarding prototyping and testing, presenting an attractive solution for rapid prototyping suitable for many design iterations.

Chapter 7 consists of test results relating to the prototyping methods described in chapter 7.

Chapter 8 rounds off the report with recommendations for further iterations of the ADCS module.

Appendix A shows an abstract and two poster presentations for the ADCS module presented at two conferences on satellites.

Appendix B expands upon section 7.6, regarding the use of an Arduino MEGA with Atmel Studio 6.

Appendix C contains a framework in C-code for using the ATmega 2560 MCU, with PWM, ADC and Serial connections in place and initialized.

Appendix D contains a summary of the thesis in norwegian

Chapter 2

Theory, definitions, notations

This chapter serves as an overview of the the theoretical backgrounds for principles and methods that are used and referenced later in the report.

2.1 Coordinate frames

There are 3 main coordinate frames that are relevant when designing the ADCS of a satellite, which are the earth-centered inertial, body and orbit frames.

2.1.1 Earth-centered Inertial

The ECI frame originates in the centre of the earth, with the x-axis pointing towards the vernal equinox, the z-axis along the Earth's rotation axis at some initial time and the y-axis completes the right-handed orthogonal system. [23] This frame is considered an inertial frame where Newton's laws of motion apply and will be represented by the letter i throughout this paper.

2.1.2 Body

The BODY frame is moving and rotating with the satellite. Its origin lies with the centre of mass of the satellite and rotates at an angular velocity ω_O relative

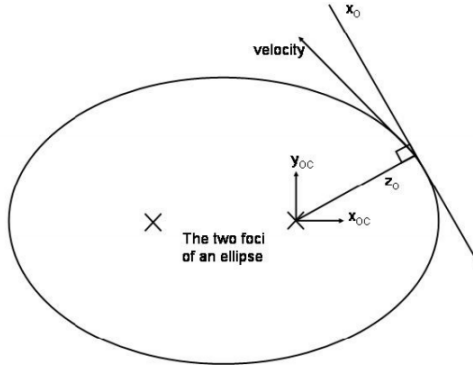


Figure 2.1: Orbit frame

to the ECI frame. The x-axis points in the forward direction, the z-axis points down, and the y-axis completes the right-handed orthogonal system. This frame will be represented by the letter b throughout this paper.

2.1.3 Orbit

The orbit frame shares its origin with the body frame and the z-axis points toward the centre of the earth. For circular orbits, the x-axis points in the spacecraft direction of motion, tangential to the orbit. The y-axis completes the right-handed coordinate system. Notice that the tangent is only perpendicular to the radius vector in the case of a circular orbit. For elliptic orbits the x-axis does not align with the satellites velocity vector, Figure 2.1 shows this. This coordinate frame is denoted o , and the rotation between the orbit- and body frames is used to represent the attitude of the satellite.

2.2 Attitude representations

2.2.1 Rotation matrix

The coordinate transformation of a vector from frame b to frame a is given by

$$\mathbf{v}^a = \mathbf{R}_b^a \mathbf{v}^b \quad (2.1)$$

where

$$\mathbf{R}_b^a = \{\vec{a}_i \cdot \vec{b}_j\} \quad (2.2)$$

is the rotation matrix from b to a as described in [6].

The rotation matrix is orthogonal and satisfies

$$\mathbf{R}_a^b = (\mathbf{R}_b^a)^{-1} = (\mathbf{R}_b^a)^T \quad (2.3)$$

The orthogonality property $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ can be used to find the time derivative of the rotation matrix [6]:

$$\dot{\mathbf{R}}_b^a = \boldsymbol{\omega}_{ab}^a \times \mathbf{R}_b^a \quad (2.4)$$

where $\boldsymbol{\omega}_{ab}^a$ is the angular velocity of frame b, relative to frame a, represented in frame a. This report will mostly make use of the skew-symmetric cross product operator $\mathbf{S}(\boldsymbol{\omega})$ to denote the cross product:

$$\boldsymbol{\omega} \times = \mathbf{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \quad \boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (2.5)$$

This notation can be found in a different, very useful parametrization of the rotation matrix, the angle-axis parametrization which represents a rotation θ around the axis λ :

$$\mathbf{R}_{\lambda,\theta} = \mathbf{I}_{3 \times 3} + \sin\theta \mathbf{S}(\lambda) + (1 - \cos\theta) \mathbf{S}^2(\lambda) \quad (2.6)$$

2.2.2 Euler angles

A widely used set of parameters for the rotation matrix is the Euler angles, where the matrix is given as a composite rotation of selected combinations of rotations about the x, y and z-axes. The rotation matrix thus consists of three successive rotations given by the roll-pitch-yaw angles.

$$\mathbf{R}_b^a = \mathbf{R}_z(\psi) \mathbf{R}_y(\theta) \mathbf{R}_x(\phi) \quad (2.7)$$

This representation isn't used in any calculations, it is merely used to find a more physically relatable representation of the satellite's attitude. The initial values are given in Euler angles before being converted to a quaternion representation.

2.2.3 Unit quaternions

Unit quaternions or Euler parameters is a four-parameter method that arises as an alternative to the Euler angle representation[10]. The rationale behind using quaternions is to avoid representation singularities which may arise from calculations using Euler angles. A quaternion q is a complex number consisting of a real part η and three imaginary parts given by the vector ε . A rotation β around the λ -axis can be expressed on quaternion form

$$\mathbf{q} = \begin{bmatrix} \eta \\ \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix} = \begin{bmatrix} \cos(\frac{\beta}{2}) \\ \lambda \sin(\frac{\beta}{2}) \end{bmatrix} \quad 0 \leq \beta \leq 2\pi \quad (2.8)$$

this implies that the unit quaternion satisfy the constraint $\mathbf{q}^T \mathbf{q} = 1$, in other words that

$$\eta^2 + \varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 = 1 \quad (2.9)$$

and the set Q of unit quaternions is

$$Q := \{\mathbf{q} | \mathbf{q}^T \mathbf{q} = 1, \mathbf{q} = [\eta, \varepsilon^T]^T, \varepsilon \in \mathbb{R}^3 \text{ and } \eta \in \mathbb{R}\} \quad (2.10)$$

Using this and equation 2.6 a unit quaternion representation of the rotation matrix from body to orbit frame is obtained

$$\mathbf{R}_b^o(\mathbf{q}) = \mathbf{I}_{3 \times 3} + 2\eta \mathbf{S}(\varepsilon) + 2\mathbf{S}^2(\varepsilon) \quad (2.11)$$

where $\mathbf{I}_{3 \times 3}$ is the identity matrix. For further calculations, it is helpful to view the rotation matrix from orbit to body, which is the transpose of \mathbf{R}_b^o , on the column vector form

$$\mathbf{R}_o^b = (\mathbf{R}_b^o)^T = [\mathbf{c}_1^b \ \mathbf{c}_2^b \ \mathbf{c}_3^b] \quad (2.12)$$

where \mathbf{c}_i^b are the projections of the x_o -, y_o - and z_o -axes in the body frame.

2.3 Estimation methods

A wide range of methods exist to filter and fuse measurements for sensors into useable data. Here the relevant methods for this paper are presented, emphasizing the methods that have seen developments during the course of ADCS design processes in our team.

2.3.1 Extended Kalman Filter

Rudolf. E. Kalman introduced the Kalman filter in 1960[15], which is an optimal recursive data processing algorithm.[23] The Kalman filter is optimal with respect to most criteria, but most importantly with respect to minimum variance. The Kalman filter uses all information that can be provided to it, processing all available measurements.

Since most real world systems are nonlinear and the Kalman filter relies on linearity, a Kalman filter that could handle nonlinear systems was developed by NASA in the early sixties.[8] This development came at the expense of losing the property of general optimality.

In contrast to the Kalman filter, the state transition and observation models can be differential functions.

Given the nonlinear equations on the form

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \quad (2.13)$$

$$\mathbf{z}_{k-1} = h(\mathbf{x}_k) + \mathbf{v}_k \quad (2.14)$$

with the assumption that the process and measurement noise \mathbf{w}_{k-1} and \mathbf{v}_k are additive and Gaussian distributed, we want to find the optimal estimate \mathbf{x}_k . The predict and update equations that are central to the operation of the Kalman filter can then be written as [23]

Predict:

$$\hat{\mathbf{x}}_{\bar{k}} = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}) \quad (2.15)$$

$$\mathbf{P}_{\bar{k}} = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}_{kal} \quad (2.16)$$

Update:

$$\mathbf{K}_k = \mathbf{P}_{\bar{k}} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{\bar{k}} \mathbf{H}_k^T + \mathbf{R}_{kal})^{-1} \quad (2.17)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{\bar{k}} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{\bar{k}}) \quad (2.18)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{\bar{k}} \quad (2.19)$$

where $\hat{\mathbf{x}}$ is the estimated state vector, \mathbf{P} the error covariance matrix, \mathbf{K} calculated Kalman gain, \mathbf{R}_{kal} the measurement covariance matrix, \mathbf{Q}_{kal} the process covariance, and

$$\mathbf{F}_k = \left. \frac{\delta f}{\delta \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{u}} \quad (2.20)$$

is the derivative of the nonlinear system with respect to the states \mathbf{x} , and

$$\mathbf{H}_k = \left. \frac{\delta h}{\delta \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1}} \quad (2.21)$$

is the derivative of the measurement equations with respect to the states.

2.3.2 QUEST

The QUaternion ESTimator is method for estimating the rotation matrix by minimizing the cost function [17]

$$\begin{aligned} J_1(q) &= \frac{1}{2} \sum_{j=1}^n \frac{1}{\sigma_j^2} (b_j - R_b^i(q) r_j)^\top (b_j - R_b^i(q) r_j) \\ &= \frac{1}{2} \sum_{j=1}^n \frac{1}{\sigma_j^2} (b_j^\top b_j - 2b_j^\top R_b^i(q) r_j + r_j^\top r_j) \end{aligned} \quad (2.22)$$

where r_j and b_j are known unit vectors in the NED frame and BODY frame respectively, and σ_j are the standard deviation of the measurement error. By using the fact that r_j and b_j are unit vectors, the cost function to be minimized can be reduced to

$$J_1(q) = \sum_{j=1}^n \frac{1}{\sigma_j^2} (1 - b_j^\top R_b^i(q) r_j) \quad (2.23)$$

2.3.3 EQUEST

The Extended QUaternion ESTimator was developed by members of the satellite project last year[24] to integrate non vectorized measurements in the cost function of the QUEST algorithm. In our case the non vectorized input stem from gyroscope measurements. This was achieved by adding another term to the cost function from equation 2.22 containing the gyroscope measurements

$$J_2 = \frac{1}{2} \sum_{j=1}^n \left\{ \frac{1}{\sigma_j^2} (b_j^T b_j - 2b_j^\top R_b^i(q) r_j + r_j^\top r_j) \right\} + \frac{1}{2} (q - \hat{q}_{gyro}) D (q - \hat{q}_{gyro}) \quad (2.24)$$

where \hat{q}_{gyro} is the estimated rotation matrix based on gyroscope tracking and D is a diagonal weighting matrix. Subtraction of two quaternions will not result in an attitude quaternion, a problem which will be discussed further in the next section.

In addition to the terms containing gyroscope measurements, a prediction term was added to the cost function

$$J_2 = \frac{1}{2} \sum_{j=1}^n \left\{ \frac{1}{\sigma_j^2} (b_j^T b_j - 2b_j^\top R_b^i(q) r_j + r_j^\top r_j) \right\} + \frac{1}{2} (q - \hat{q}_{gyro}) D (q - \hat{q}_{gyro}) + \frac{1}{2} (q - \hat{q}_{pre}) S (q - \hat{q}_{pre}) \quad (2.25)$$

where \hat{q}_{gyro} contains the predicted attitude based on previous observations.

2.3.4 Further development of the EQUEST method

Last term, the EQUEST method was further improved by replacing the quaternion subtraction in the cost function with quaternion products[18]. The resulting cost function can then be written on matrix form

$$J_3 = \frac{1}{2} \sum_{j=1}^n \left\{ \frac{1}{\sigma_j^2} (b_j^T b_j - 2b_j^\top R_b^i(q) r_j + r_j^\top r_j) \right\} + \frac{1}{2} q^\top N_d q + \frac{1}{2} q^\top N_s q \quad (2.26)$$

$$N_d = \hat{Q}_{gyro}^{*\top} D \hat{Q}_{gyro}^* \text{ and } N_s = \hat{Q}_{pre}^{*\top} S \hat{Q}_{pre}^*$$

where $\hat{Q}_{gyro}^{*\top}$ and $\hat{Q}_{pre}^{*\top}$ are matrix representations of the error quaternions ($\hat{q}_{gyro}^* \otimes q$) and ($\hat{q}_{pre}^* \otimes q$) respectively.

The resulting mathematical model is more accurate than the one resulting from equation 2.25, but it is also more computationally demanding.

2.3.5 Nonlinear Observers

Nonlinear observers are relatively new methods for attitude estimation, with the overall design goal to avoid the very high number of differential Riccati equations that are needed for the extended kalman filter. One of the most recent improvements of the method achieves a semi-global stable attitude observer with gyro bias state estimation extended to time-varying reference vectors[9].

There are various ways to implement these observers, based on what kinds of measurements are available, usually using a gyroscope, an accelerometer and a magnetometer, but by prof. Fossen's account, there should be no problem replacing the accelerometer with sun sensors.

2.4 Mathematical modeling

The satellite model is presented mostly as a reference point to understand the attitude estimators better.

2.4.1 Satellite model

By regarding the satellite as an ideal rigid body, a dynamic model of the satellite can be derived from a Newton-Euler formulation as described in [7]

$$\mathbf{I}\dot{\boldsymbol{\omega}}_{ib}^b + \mathbf{S}(\boldsymbol{\omega}_{ib}^b)\mathbf{I}\boldsymbol{\omega}_{ib}^b = \boldsymbol{\tau}^b \quad (2.27)$$

where \mathbf{I} is the satellite's moment of inertia, $\boldsymbol{\omega}_{ib}^b$ the angular velocity of the body frame relative the inertial frame and $\boldsymbol{\tau}^b$ the torque acting on the satellite, all of which are represented in the body frame.

The satellite's orientation in space is obtained through integration of the angular velocity. A vector representation using unit quaternions is found in [16]

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\boldsymbol{\eta}} \\ \dot{\boldsymbol{\varepsilon}} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\boldsymbol{\varepsilon}^T \\ \boldsymbol{\eta}\mathbf{I}_{3 \times 3} + \mathbf{S}(\boldsymbol{\varepsilon}) \end{bmatrix} \boldsymbol{\omega}_{ob}^b \quad (2.28)$$

The angular velocity relations between the orbit, body and inertial frames are described by the equation

$$\boldsymbol{\omega}_{ib}^b = \boldsymbol{\omega}_{io}^b + \boldsymbol{\omega}_{ob}^b = \mathbf{R}_o^b \boldsymbol{\omega}_{io}^o + \boldsymbol{\omega}_{ob}^b \quad (2.29)$$

where $\boldsymbol{\omega}_{io}^o = [0 \ \omega_0 \ 0]^T$ is the constant angular velocity of the orbit frame with respect to the inertial frame.

Chapter 3

Sensors and actuators

To make attitude estimations, the three main sensors that are used on cubesat are gyroscopes, magnetometers and some kind of sun sensor.[5][2] Gyroscopes are used to measure angular velocity and work independently from external influences, while magnetometers and solar sensors are dependent on models to make sense of the data they measure.

3.1 Sun sensor

3.1.1 Solar panels

The original design proposal for the attitude estimation system for the NUTS satellite included using the solar panels already present in the design to estimate the satellite's attitude in relation to the sun. Solar panels deliver current that is dependent on the area of the solar cell that is exposed to sunlight. The current is then dependent on the angle of the sun relative to the solar panel α_s by a cosine law, which gives the current from a solar panel as

$$I = I_{max} \sin \alpha_s \quad (3.1)$$

where I_{max} is the maximum current where the sun shines on the solar panel directly from above[21]. For a three dimensional satellite with solar panels on

5 of 6 sides it is then possible to find a sun vector as long as the sun is not shining directly on the bottom panel. In [21] the method for finding the sun vector in the body frame is described, and results are tested using a model. By using trigonometric expansions and assuming the solar panels have the same maximum current output, the current from panels on the x-, y- and z-axis are found to be

$$I_1 = I_{max} \sin\alpha_1 \cos\alpha_3 \quad (3.2)$$

$$I_2 = I_{max} \sin\alpha_2 \cos\alpha_3 \quad (3.3)$$

$$I_3 = I_{max} \sin\alpha_3 \quad (3.4)$$

While the sun shines on the positive x- and y-axis and negative z-axis the sun vector in the body frame then becomes

$$\mathbf{v}_S^B = \begin{bmatrix} \sin\alpha_1 \cos\alpha_3 \\ \sin\alpha_2 \cos\alpha_3 \\ -\sin\alpha_3 \end{bmatrix} \quad (3.5)$$

Combine with equations 3.23.3 and 3.4 to get

$$\mathbf{v}_S^B = \begin{bmatrix} I_1 \\ I_2 \\ -I_3 \end{bmatrix} \frac{1}{I_{max}} \quad (3.6)$$

I_{max} can be removed from the equation as only the direction of the sun vector gives attitude information, and the sun vector can be written as

$$\mathbf{v}_S^B = \begin{bmatrix} X & 0 & 0 \\ 0 & Y & 0 \\ 0 & 0 & Z \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} \quad (3.7)$$

where X, Y and Z are ± 1 depending on which side of the satellite the panels that deliver current is placed. This is the same results that are reported in [24], only expanded in a way that may be more useful for directly implementing on an MCU

$$\mathbf{v}_S^B = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} Int_{x+} \\ Int_{x-} \\ Int_{y+} \\ Int_{y-} \\ Int_{z+} \\ Int_{z-} \end{bmatrix} \quad (3.8)$$

where Int denotes the intensity of the measured sunlight on each side of the satellite.

During the design process of the EPS module we discussed placing sensors for retrieving the current drawn from the solar panels directly on the EPS subsystem, the results of which could be passed to the ADCS module over the internal data bus. This would have made the individual sun intensity measured on the solar panels readily available for the ADCS module in digital form without using external ADC units or wiring directly from the solar panels to ADC ports on the ADCS MCU, but the idea had to be scrapped because of space and complexity requirements on the EPS module.[14]

This leaves the option of using the ADC module on the ATmega2560/2561 MCU used in the ADCS subsystem to retrieve useful data from the solar panels, but may potentially introduce a weak point as there would need to be wires connected straight from the solar panels themselves.

3.1.2 Dedicated sun sensors

While the solar panels can indeed be used as coarse sun sensors, using dedicated units for this purpose frees the ADCS unit from needing direct access to the currents going out of the solar panels every time an attitude estimate is made, and also provide the benefits of increased accuracy. In addition to this it might be possible to fit a small dedicated sun sensor on all the sides of the satellite, removing the blind spot on the camera panel.

The calculations for finding the sun vector in the body frame by means of sun sensors are the same as when using the solar panels, but dedicated sun sensors usually provide better accuracy.

There are various options for sun sensors, and care should be taken when considering which one to buy, or even if they can be produced internally.

3.2 Gyroscope and magnetometer

3.2.1 Gyroscope

Gyroscopes are used to get an estimate of the angular velocity of the satellite.

While gyroscopes come in various forms, the design that is relevant for a project like this are MEMS gyros. These types of gyroscopes are usually based on the tuning fork gyro principle[23] where a vibrating fork is subjected to a steady rotation about it's axis and the response detected by the transducer results from the Coriolis force.

3.2.2 Magnetometer

A magnetometer measures the local magnetic field around the satellite, and needs to reference a model of the earth's magnet field to get a useable estimate of attitude from the measurements. The model that is most widely implemented and will be used in our satellite is the IGRF model.[22]

3.2.3 Sensor choice

Analog Devices is a well known manufacturer of MEMS-based sensors and one of the most prominent players in the market. They have a full range of different inertial sensors, but unfortunately they do not have a system which includes both a gyroscope and magnetometer without also including an accelerometer. Their sensors are aimed more towards the professional market than the CHIMU sensor that was used in a previous prototype [24] and that reflects in the ratings provided.

Another advantage to us while we are still in a stage where the ADCS board is still in the design process is that Analog Devices also offers a package with the ADIS 16405 sensor and a breakout board that facilitates rapid prototyping, at a cost of around \$750.

Various comparisons of IMUs in this range has been made, like the following table presented by prof. Thor I. Fossen, which confirms the ADIS 16405 as a viable sensor for our project.

Gyro bias stability for some IMUs

| MEMS Units | drift deg/s | drift deg/h | |
|---------------------------|--------------|---------------|--|
| Analog Devices ADIS 16385 | 0.0017 deg/s | 6.12 deg/h | Earth rotation: 0.0042 deg/s 15.12 deg/h |
| Analog Devices ADIS 16405 | 0.007 deg/s | 25.2 deg/h | |
| SBG Systems IG-500E | 0.1 deg/s | 360.0 deg/h | |
| MicroStrain 3DM-GX3@ -45 | 0.2 deg/s | 720.0 deg/h | |
| XSENS MTi | 1.0 deg/s | 3 600.0 deg/h | |

Gyro drift for the Litton LN-200 fiber-optical IMU is 1-10 deg/h

Figure 3.1: Comparison of various IMUs

3.3 GPS

In order to get a more accurate orbit estimate to get an accurate reading from the magnetometer by referencing the IGRF model, GPS can be useful. Another member of our team has been investigating the options in this area and the results can be found in [20].

3.4 Magnetic coils as actuators

The electromagnet was invented in 1824 by the principle that a current flowing through a conductor in a loop generates a magnetic field, where the strength of the field is proportional to the amount of current flowing through the conductor.

To be able to use magnetic coils as actuators for attitude control in a satellite, three coils are placed perpendicular to each other to generate magnetic fields that can interact with the geomagnetic field of the earth. This is an attractive mode of actuation for a small and lightweight cube satellite as no moving parts are needed, and the hardware is relatively simple in its design and construction.

A problem with this setup is that the coils can only produce a moment perpendicular to the area it encloses. [19] This means that we cannot create an arbitrary moment vector with the result that the optimal direction of vector can rarely be produced, and in the worst case the coils can theoretically be momentarily positioned in such a way that control is momentarily lost. [22]

3.4.1 International Geomagnetic Reference Field

In order to use the magnetic coils as actuators for attitude control, the algorithms need to have knowledge of the earth's magnetic field at the current orbit position.

The IGRF model is a standard mathematical model of the earth's magnetic field that is updated every 5 years by the International Association of Geomagnetism and Aeronomy. The IGRF model consists of the the Gauss coefficients which define a spherical harmonic expansion of the magnetic scalar potential, providing a very accurate estimate of the geomagnetic field.

Chapter 4

Estimator considerations

The mission goals of the NUTS satellite provides several criteria that should influence process of deciding on which estimation algorithm should be chosen, a system with no opportunities for hardware modification or repairs after completion places high demands on the reliability of the implemented solutions.

4.1 Extended QUaternion ESTimator

Previous work on the NUTS satellite has concluded that the EQUEST algorithm would be the best fit for our needs [24] and it has indeed been the algorithm that most work has been put into so far in the project, e.g. in [18]. The EQUEST method compares very favourable with the well known extended kalman filter, with EQUEST running more than 5 times faster and being significantly less computationally demanding, with a faster settling time. Another factor is that the slow and predictable nature of attitude change in a satellite makes it a good candidate for using the additional prediction term in equation 2.25.

However, since the QUEST method was first extended to be able to implement non-vectorized measurements by Yabar and Jenssen in [24] and first mathematically explored by Rinnan in [18] the method has only been tried and tested in lab environments and might pose a risk compared to more well known methods.

4.2 Extended Kalman Filter

It is then tempting to instead use the go-to method for filtering measurements the last 50 years, which has also been tested and given good results in many cube sats previously, the extended kalman filter.

While the value of testing novel techniques in the NUTS project should not be underestimated, the rest of the satellite relies on the ADCS module functioning properly, and every potential risk factor needs to be strictly evaluated with regards to this.

The testing of EQUEST vs. EKF done by Yabar and Jenssen indicates that the EKF provides a smoother estimation and is able to estimate the magnetometer and gyroscope bias in contrast to the EQUEST method.

4.3 Nonlinear Observers

There is also a third option for the attitude estimator, which is considered the current state of art in the field. As this attitude estimator is presented by Fossen et. al. [12] it provides an option that takes the best from both the QUEST algorithm and extended kalman filter.

Compared to the kalman filter, it promises to avoid the computationally complex Riccati equations, making the C-code more compact and guarantees asymptotic and exponential stability properties.

Compared to the QUEST algorithm it can estimate gyroscope and magnetometer bias and provides a more accurate and stable solution.

A final option that was suggested to our team by prof. Fossen was to use the quest algorithm for fast initial estimates and use that as a starting point for the nonlinear observer algorithm, an option that is currently being investigated theoretically.

However some of the same considerations as with the EQUEST algorithm comes into play with this method, as it is considered a new method for attitude estimation, especially compared to the kalman filter. The advantage to this over QUEST is that the current state of the art developments of this method, both theoretically and experimentally is situated at NTNU[12][11] where the results will be readily available for our group. One example of this can be seen in figure 4.1

Test 3: Traffic Pattern Maneuver

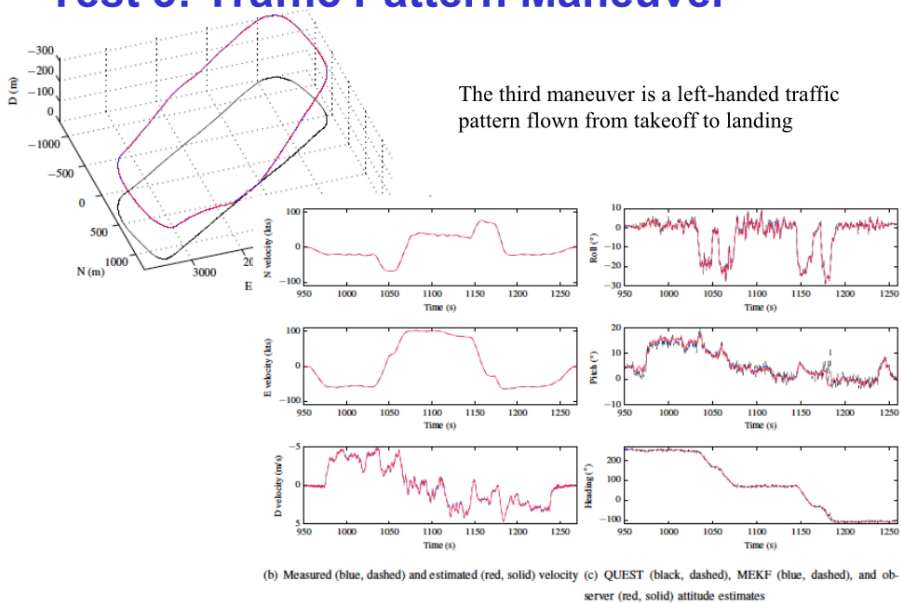


Figure 4.1: Experimental results presented by T.I. Fossen et. al.

Chapter 5

Software

An important part of selecting the right estimation algorithm is considerations concerning software implementation. In a cube satellite both space and available power is extremely limited, and it is highly important that any software that is put on the microcontrollers is optimized with regards to power consumption. Of course this consideration also has to be carefully balanced with the strict reliability demands of a system such as this, while the computational speed on the other hand is less important because the attitude of the satellite changes very slowly and can be predicted with a high degree of certainty.

5.1 EQUEST

A great advantage to the EQUEST method over the more commonly used Kalman filter is that it requires a great deal fewer arithmetic operations, while also being faster in settling on a solution. The algorithm does not require any linearization, and the total number of arithmetic operations averages at around 3200 for our system.

However the implementation in C-code is not trivial, and includes both matrix inversions and an eigenvalue problem[24], but it still runs around 5 times faster than the EKF.

5.2 Extended Kalman Filter

The extended kalman filter requires both matrix operations and linearizations, resulting in a very high number of arithmetic operations needed to estimate the attitude. In terms of software implementation, the extended kalman filter is a very poor performer, and in situations where resources are limited it would be best if one could avoid it. The EKF does around 40 000 operations to arrive on a solution, 12.5 times as many as what is needed for the EQUEST algorithm.

5.3 Nonlinear observer

The nonlinear observer method was, and still is designed with implementations on microcontrollers in mind. One of the main motivations of developing these methods is to produce methods that are minimum representations in terms of mathematical operations in software, floating point operations, which in turn reduces the need for maintenance by means of a reduction in software complexity and faults. A pleasant by effect of this is that CPU and power demands are also reduced compared to older methods.

Chapter 6

Prototyping

Since there are still a series of design decisions that needs to be made regarding the ADCS module, a final design will not be presented here. However both incremental updates to the prototype design as well as ways to speed up the prototyping process is investigated.

6.1 Arduino

Arduino is a tool for prototyping electronics based on Atmel microchips and open source software. In particular interest for this project is the Arduino MEGA board that is based on the ATmega2560 chip, which was selected for the prototype in the previous ADCS module iteration[22][24] and thereby offers full backwards compatibility. It offers a USB interface to communicate with a computer and easy access to all the pins present on an ATmega 2560.

6.2 Rapid prototyping with Arduino

Compared to the solution for prototyping previously used on the project [22][24], the Arduino MEGA board offers several advantages.

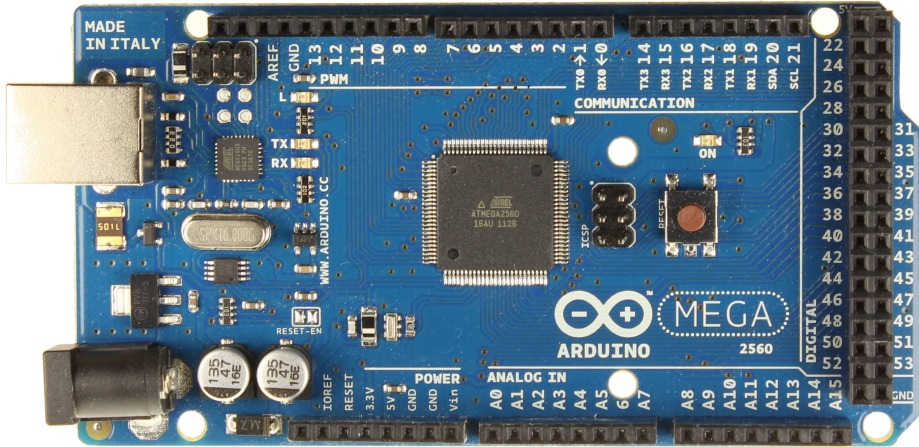


Figure 6.1: Arduino MEGA front

1. The Arduino board already comes with a set of standard matched electronics which reduces the prototype design process to adding the components that are specifically needed for what we want to accomplish.
2. No permanent connections need to be soldered to the board, which allows rapid iterations without having to print a new PCB and solder on parts.
3. No additional development tools, like the AVR Dragon are needed for programming the MCU.

By combining the Arduino board with a standard breadboard one can test different configurations on the fly by simply placing wires without having to permanently solder anything in place. The setup seen in figure 6.2 has all the elements of the prototype that was produced last spring, and is ready to be quickly modified.

6.2.1 Fritzing

By using the open source program Fritzing, one can more easily plan the wiring necessary for component testing and keep track of iterations in a clean and well arranged way.

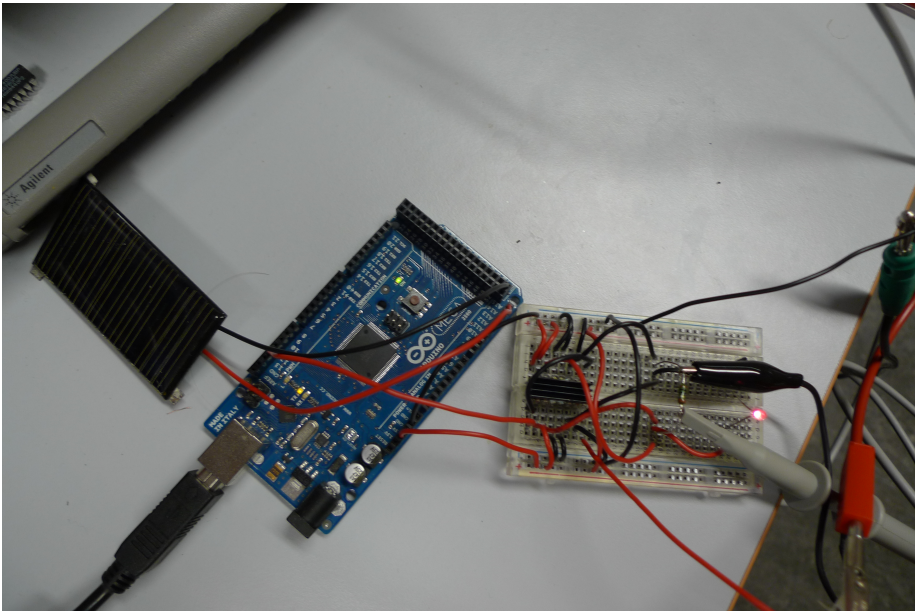


Figure 6.2: Arduino and breadboard with H-bridge

6.3 H-bridge

An H-bridge is needed between the MCU and the magnetic coils used for controlling the satellite, since the power output for the coils needs to be bi-directional

In the previous ADCS prototype iteration, an H-bridge was used that has now gone out of production [22], so a new choice had to be made.

A widely used H-bridge is the Texas Instruments SN754410 quadruple half H-driver and by looking at it's data sheet [3] we can see that it fits our requirements, providing bi-directional drive currents up to 1A, with a supply voltage range that includes the 5V available in the satellite and a separate input-logic supply.

An advantage to using a quadruple half H-driver such as this is that it can provide an easy way of adding some redundancy in a fragile system, without compromising the space requirements by simply wiring the output signal from the MCU to both the separate H-bridges that are in the chip, ensuring full functionality from the magnetic coils even if one of the bridges should fail.

If space permits, it is recommended to add heat sinks to the H-bridges to help dissipating surplus heat, as seen in figure 6.3

In figure 6.4 the function table of the H-bridge can be seen, showing a more or less intuitive relation between inputs and outputs.

Figure 6.5 shows a wiring diagram generated in fritzing where the separate drivers in the H-bridge are wired in parallel for redundancy. PWM output from the Arduino is wired to both the 1,2EN and the 3,4EN pins on the H-bridge, while pins 1A and 4A pins are pulled HIGH and pins 2A and 3A are grounded, logic LOW. The supply voltage VCC2 is connected to an external voltage generator, and the logic supply VCC1 is connected to the 5v output on the Arduino, which in turn is provided by a usb connection to a computer. Output pins 1Y and 4Y are wired to one side of a resistor, while pins 2Y and 3Y are connected to the other, providing a connection point for an oscilloscope.

6.4 ADC and solar panels

For testing purposes, it is a viable option to use the ATmega's ADC capabilities for utilizing the solar panels as crude sun sensors.

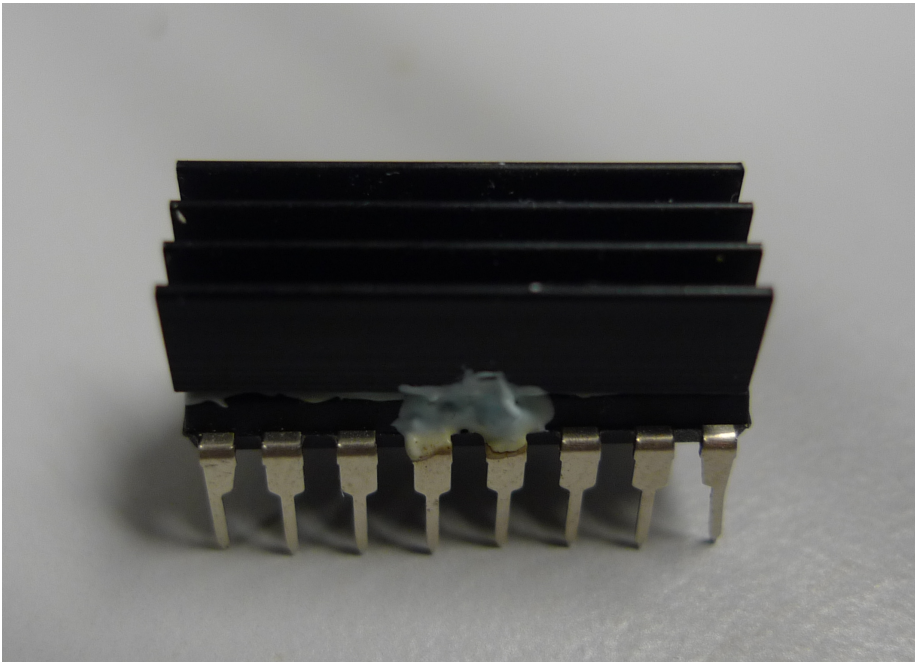
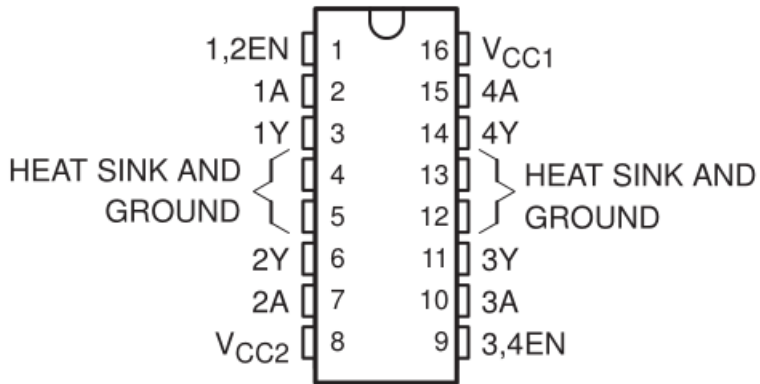


Figure 6.3: SN754410 with heat sink



FUNCTION TABLE
(each driver)

| INPUTS† | | OUTPUT |
|---------|----|--------|
| A | EN | Y |
| H | H | H |
| L | H | L |
| X | L | Z |

H = high-level, L = low-level

X = irrelevant

Z = high-impedance (off)

† In the thermal shutdown mode, the output is in a high-impedance state regardless of the input levels.

Figure 6.4: Pin map and function table for the SN754410 H-bridge

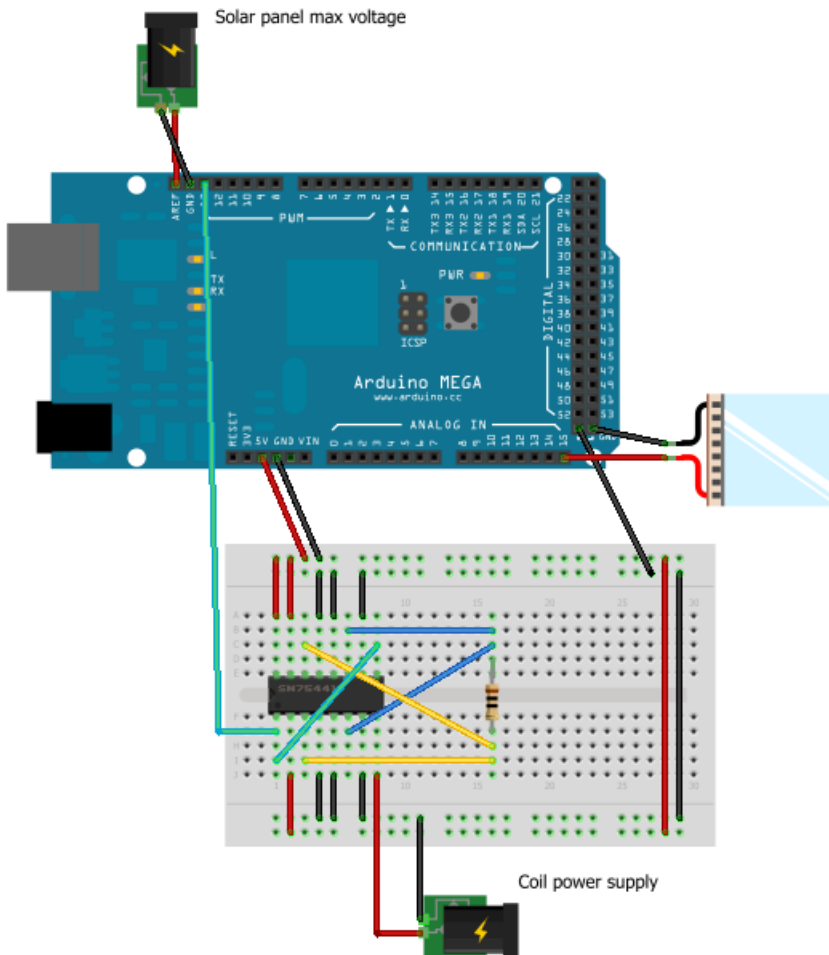


Figure 6.5: Fritzing generated graphics showing h-bridge and solar panel wiring

To do this, it is important to first measure the maximum voltage that the solar panels can deliver and connect a voltage generator set to that voltage to the “aref” pin on the MCU. This is done to make sure the digital output value is able to cover the entire range of the solar panel’s analog range.

6.5 Magnetic coils

Design of the magnetic coils is the subject of a master’s thesis currently in progress, for now we will just mention that in order to use 3 magnetic coils with the H-bridges set up for redundancy, 3 separate bridges are needed. If not, that number can be reduced to 2.

6.6 Programming the microcontroller

While Arduino provides it’s own programming language based on Wiring, it is desirable to use Atmel Studio and the C programming language to gain more direct control over the MCU, including the option of setting registers separately.

6.6.1 Using Arduino with Atmel Studio

Fortunately, it is quite easy to bypass the Arduino development environment and set up Atmel Studio to program the Arduino instead, and the process can be highly automated. The method for doing so as well as general tips for using Atmel Studio for the purposes described in this paper can be found in Appendix B.

Figure 6.6 shows the translations between the pin names that are used in the ATmega 2560 datasheet and the the pin names that are printed on the Arduino MEGA board.

6.6.2 ADC

To initialize the ADC functionality on an ATmega 2560 microcontroller, some register values needs to be set.

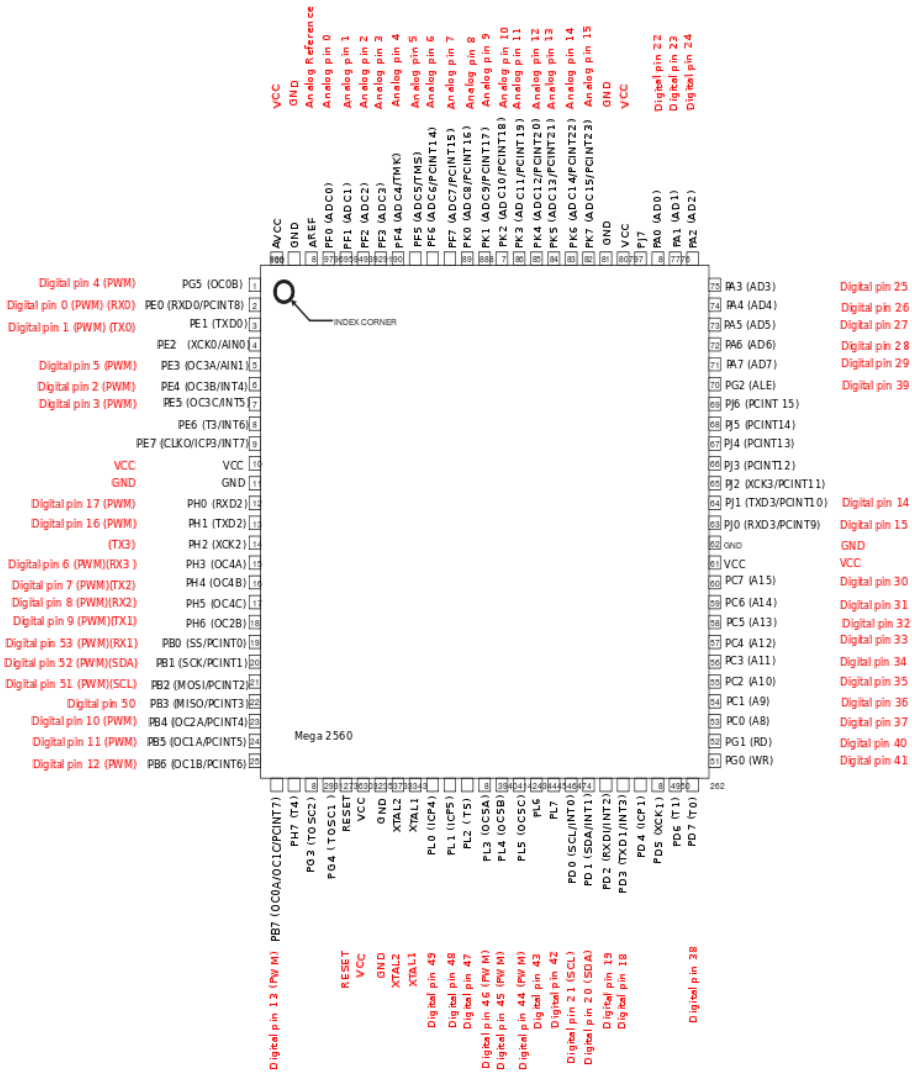


Figure 6.6: Pin map Arduino MEGA vs ATmega 2560

Table 26-3. Voltage Reference Selections for ADC

| REFS1 | REFS0 | Voltage Reference Selection ⁽¹⁾ |
|-------|-------|--|
| 0 | 0 | AREF, Internal V_{REF} turned off |
| 0 | 1 | AVCC with external capacitor at AREF pin |
| 1 | 0 | Internal 1.1V Voltage Reference with external capacitor at AREF pin |
| 1 | 1 | Internal 2.56V Voltage Reference with external capacitor at AREF pin |

Figure 6.7: REFS1:0 bits in the ADMUX register

First, for using the “aref” pin as voltage reference for the ADC unit, the REFS0 and REFS1 bits needs to be set at 1 and 0 respectively in the register ADMUX, see figure 6.7.[1]

The rest of the bits that needs to be set are found in the ADCSRA register, ADEN is set to 1 to initialize ADC, while ADPS2:0 are all set to 1. The ADPS2:0 bits determine the ADC prescaler select bits, which determines the ADC update frequency. By setting them all to 1, the slowest update frequency is selected, which is more than adequate for reading a solar panel often enough for testing purposes.

The problem of reading values on ADC pins is solved in software, and one possible implementation can be seen in figure 6.8. The important thing is to use if-sentences to set the right registers depending on which pin you want to be read, and making sure that the MCU keeps reading the pin as long as the “single conversion” bit, ADSC in the ADCSRA register, is kept high.

6.7 Prototype board

The previous prototype was designed using CadSoft Eagle, a program that is popular among hobbyists. It is desirable to make the switch to using Altium Designer to facilitate integration with the other subsystems of the satellite by using the same design platform in all areas.

Because there are still elements of the ADCS module that are not yet final the conversion of the design was only partially completed, placing the MCU, H-bridges and assorted electronics, while still leaving the sensors open.

```

uint16_t ADC_read(uint8_t ch)
{
    // select the corresponding channel 0~15

    // To account for ch between 7 and 16, need to set bit MUX5 in ADCSRB HIGH, else it
should be LOW
    if (0x08 & ch) { ADCSRB |= (1<<MUX5); }
    else { ADCSRB &= ~(1<<MUX5); }

    // ANDing with 0x07 will always keep the numerical value
    // of 'ch' between 0 and 7
    ch &= 0x07;

    ADMUX = (ADMUX & 0xF8)|ch;    // clears the last 3 bits before ORing

    // start single conversion
    // write '1' to ADSC
    ADCSRA |= (1<<ADSC);

    // wait for conversion to complete
    // ADSC becomes '0' again
    // till then, run loop continuously
    while(ADCSRA & (1<<ADSC));

    return (ADC);
}

```

Figure 6.8: ADC reading

The solution of using an Arduino and breadboard provided satisfactory performance for testing purposes, so there was no need to print a circuit board of an unfinished prototype at this stage of the design process.

Chapter 7

Simulations and tests

7.1 Reading the solar panels

A simple flashlight was used to simulate various angles of attack of light relative to the solar panels. The result of the analog to digital conversion done by the MCU was then reported over the serial port provided by the USB connection on the Arduino board. Because of this very simplistic test bed there wasn't sufficient data to confirm that the solar cell voltage relative to the light's angle of attack did indeed follow a cosine law, but it could be used as a confirmation that the ADC ports on the ATmega connected through an Arduino board functions with the solar cells we use for testing in this stage.

Figure 8.1 shows the the test bed with the MCU programmed to output the measured solar panel voltage as a PWM signal through an H-bridge connected to an oscilloscope.

7.2 H-bridge output

An H-bridge was tested with both using only two of the four half drivers, in effect one full driver, and by wiring two such "full drivers" in parallel to simulate loss of redundancy. The resulting PWM output was recorded using an oscilloscope and can be seen in figures 7.2 and 7.3, showing a minimal difference between output amplitudes.

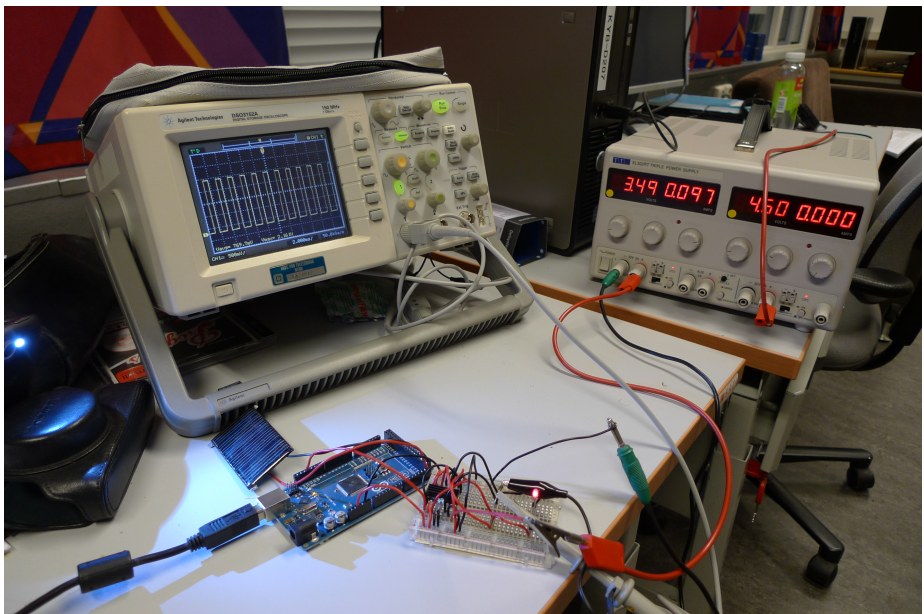


Figure 7.1: Arduino test bed with solar panel

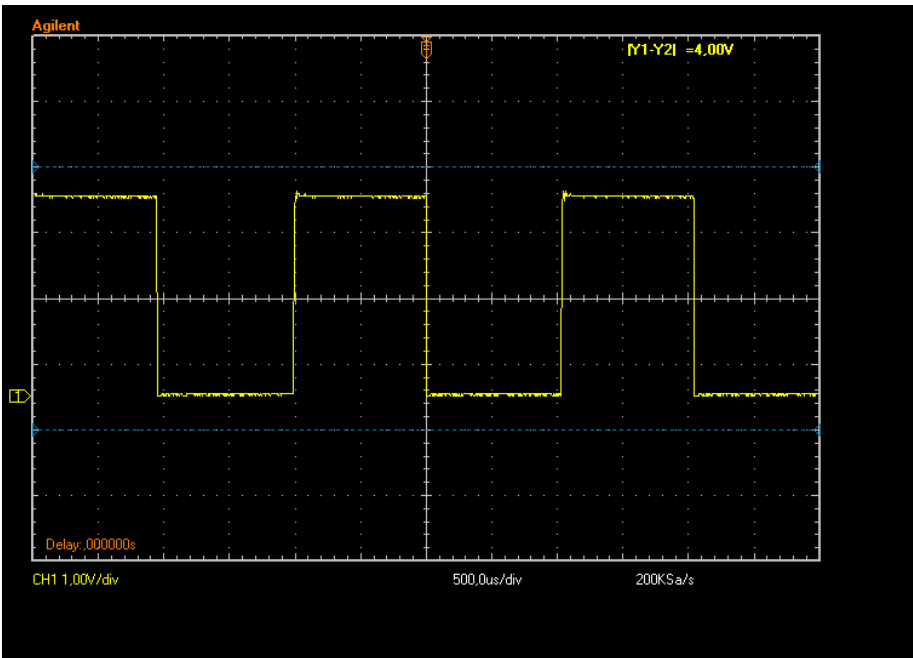


Figure 7.2: PWM output with one full H-bridge

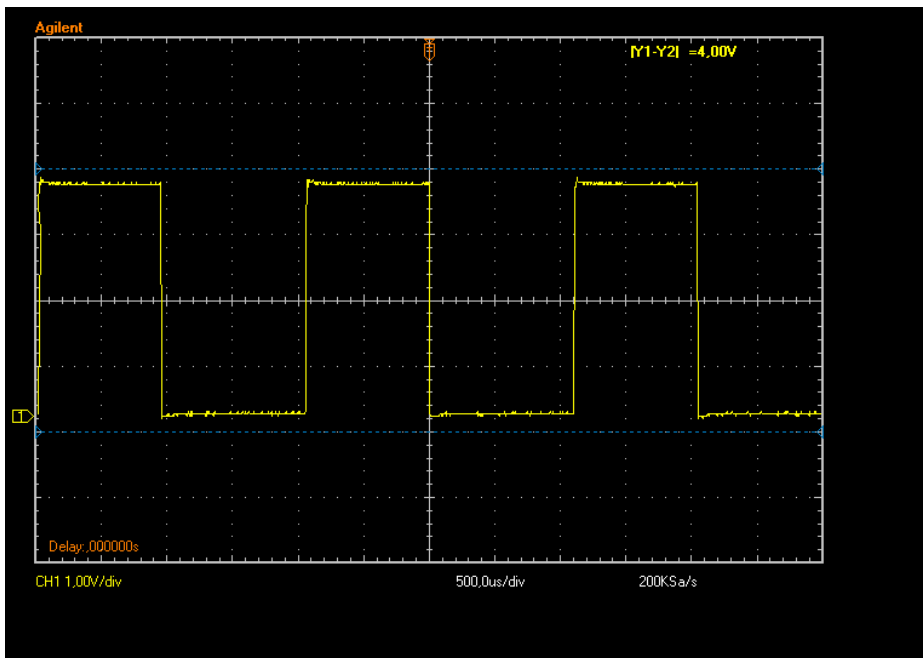


Figure 7.3: PWM output with two full H-bridges in parallel

7.3 Discussion

Because of the very simplistic test bed for the solar panels there wasn't sufficient data to confirm that the solar cell voltage relative to the light's angle of attack did indeed follow a cosine law. However it is a confirmation that the ADC ports on the ATmega connected through an Arduino board functions with the solar cells we use for testing in this stage.

Testing of the H-bridge suggests that difference in voltage output after removing one of the two drivers in parallel is minimal, and would indeed be sufficient to power the coils.

Chapter 8

Recommendations for further improvements to the ADCS module and concluding remarks

In this report a number of assessments regarding the design of the Attitude Determination and Control System for the NTNU Test Satellite has been made, but there is still a lot of work left to be done before the system is ready for launch, and this should be seen as a design iteration more than anything. Based on both this and previous work put into the ADCS design we can recommend further improvements to the module as a conclusion to this thesis.

8.1 Sensors

For reliability reasons, a sensor from a well known company such as Analog devices should probably be used for the inertial measurement unit. If there is room in the satellite, an all-in-one solution like the ADIS 16405 will probably offer the best compromise between operational reliability, power and space re-

quirements. By getting the package with the breakout board, it is also easily used for rapid prototyping and frequent design iterations.

For sun sensing it would be a much safer option to use dedicated sun sensors instead of trying to hack a solution where the solar panels are used as crude sun sensors. The goal should be to reduce liabilities wherever possible, and this represents a safer option without having to sacrifice much in regards to other design requirements. As an added benefit, sun sensors can be placed on all sides of the satellite, and the blind side you would otherwise get from the camera panel with the solar panels is eliminated.

8.2 Estimation algorithm

While a hybrid between the EQUEST method and nonlinear observers are currently being investigated, this is an area that is extremely vulnerable to errors and no matter the final decision here, very rigorous testing schemes needs to be constructed for this part of the satellite.

If some sort of collaboration with the groups that are working on the nonlinear observers can be established, and it shows stability and reliability properties that are very good, it indeed represents a very attractive algorithm compared to our needs and requirements.

However, if such testing schemes can not be completed, no matter how boring and old fashioned the Extended Kalman Filter may seem, it has an excellent track record in space. This is a fact that suggests the EKF in no way should be tossed aside lightly, even if it is slow and very computationally demanding.

8.3 Board design

The SN754410 H-bridge is tried and tested in many applications and should be a good match for our satellite. If there is room on the ADCS board, the easy way of getting some redundancy by using three separate units also seems like an attractive solution.

The important thing is that the final board again is tested to extreme specifications, as hardware errors can not be fixed once the satellite is finalized.

Appendix A

Conference submissions

A.1 The 25th Annual AIAA/USU Conference on Small Satellites, Utah State University

A.1.1 Poster Submission



ADCS for the NUTS satellite

Fredrik Holberg, Kaan Yabar, Kristian Jenssen, Zdenko Tudor, Gaute Bråthen
Norwegian university of science and technology

Introduction

As a part of the CUBESAT project at the Norwegian university of science and technology (NTNU) some work has been conducted on the ADCS system during this spring.

The actuators chosen for our cubesat project are three magnetic coils, which will be used both for attitude estimation and control.

This spring a new method for attitude estimation was developed and implemented based on the Quaternion Estimation (QUEST) method and it was later compared with an implementation of the well known Extended Kalman Filter (EKF).

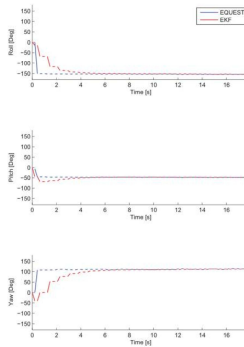
Different methods for nonlinear and optimal attitude control was compared, and an implementation of the detumbling controller and a reference controller for large deviations was tested.

| | |
|--|---|
| <p>EKF</p> <ul style="list-style-type: none"> — Well known — Estimates sensor bias — Good filtering effect — Requires about 40000 arithmetic operations | <p>EQUEST</p> <ul style="list-style-type: none"> — Finds solution in one iteration — Requires about 3200 arithmetic operations — About 5 times faster than implemented EKF — More intuitive tuning parameters than EKF |
|--|---|

Estimation

Substantial improvements have been made to the QUEST method in order to fuse vectorized and non-vectorized measurements such as gyroscope data, making the new extended QUEST (EQUEST) method more suitable for attitude estimation.

For comparison purposes both the EQUEST and EKF methods were implemented on an AVR microcontroller using C language.



Control

A dissipative controller was selected for the detumbling phase as the data needed for its computation is readily available from the determination module which makes calculation of the time derivative of the IGRF module needed for the popular B-dot detumbling controller redundant.

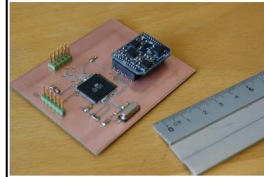
For the stabilization phase several different controllers were explored, using the dipole model of the earth's magnetic field. The controllers that yielded desirable results are shown in the following table.

| Control strategy | Energy used | Time used |
|---------------------------|-------------|------------|
| LQR, constant gain matrix | 0.7 Joule | 2.1 orbits |
| LQR | 33 mJoule | 1.5 orbits |
| Model independent PD | 90 Joule | 5 orbits |
| Sliding mode controller | 42 mJoule | 4 orbits |
| Angular velocity feedback | 0.75 Joule | 9 orbits |
| Attitude feedback | 0.8 Joule | 7 orbits |

Other controllers explored included a model dependent PD-controller and a backstepping controller, but these failed to stabilize the system.

Implementation

We have implemented a prototype running both the attitude determination and control systems using an AT-MEGA 2561 μ C. Previous CUBESAT



efforts by NTNU have assumed that the μ C unit did not have sufficient computing power to continuously compute controller gain matrices based on the IGRF model, but that seems to no longer be an issue as this prototype was found to be able to handle all the computations needed in real time.

Conclusions

Based on the work concluded so far, our satellite will be using the newly developed EQUEST method for attitude determination coupled with a dissipative detumbling controller and either a nonlinear sliding mode controller or an optimal LQR-controller for the stabilization phase. This will be implemented on an ATMEGA 2561 μ C combined with a CIMU IMU.

A.2 The 4th European Cubesat Symposium, Ecole Royale Militaire Brussels

A.2.1 Abstract

The NUTS (NTNU Test Satellite) is a satellite being built in a student CubeSat project at the Norwegian University of Science and Technology. The project was started in September 2010 as a part of the Norwegian student satellite program run by NAROM (Norwegian Centre for Space-related Education). The NUTS project goals are to design, manufacture and launch a double CubeSat by 2014. As payload an IR-camera observing waves in the air-glow layer is planned, as well as a short-range RF experiment. The satellite will fly two transceivers in the amateur radio bands. Final year master students from several departments are the main contributors in the project and most of the system components are designed and built by students.

Last term a new method for attitude estimation based on QUaternion ESTimation (QUEST) was developed by members of our project group. The Extended QUEST (EQUEST) method enables the fusing of vectorized and non-vectorized measurements making it more suitable for attitude estimation. Simulations showed that for cubesats, which often uses magnetorquers for both attitude estimation and control in addition to the often limited computational power, the EQUEST method offers several advantages over the well known Extended Kalman Filter.



Described in this paper is an implementation of the EQUEST method using magnetorquers for 3-axis magnetometer data, solar panels for sun sensing and a 3-axis gyroscope. This information is further combined with continuous computation of the geomagnetic field using the IGRF-model on a single 8-bit, 16MHz microcontroller unit to apply correct actuation to the magnetorquers through the use of well known controller algorithms.

Finally simulation and prototype test results are presented which confirms the adequacy of this simplistic approach for keeping the satellite within the stability requirements of our payload.

A.2.2 Poster Submission

Implementation of three axis attitude determination and control system for a double cubesat

F.S. Holberg, R. Birkeland, J.T. Gravdahl
 Dept. of Eng. Cybernetics, Norwegian University of Science and Technology, N-7001 Trondheim, Norway
fredhol@stud.ntnu.no roger.birkeland@iet.ntnu.no

About the project

The NUTS (NTNU Test Satellite) is a satellite being built in a student Cubesat project at the Norwegian University of Science and Technology. The project was started in September 2008 and is a part of the Norwegian student satellite program run by NAROM (Newsgian Centre for Space-related Education). The NUTS project goals are to design, manufacture and launch a double Cubesat by 2014. As the payload an IMU-sensors observing ways in the air-glow layer is planned, as well as a short-range HF experiment. The satellite will fly two transceivers in the amateur radio bands. Final year master students from several departments are the main contributors in the project.

EQUEST

The EQUEST method has been developed from the QUEST algorithm in order to make use of non-vectorized gyroscope measurements for the attitude estimation. The main idea behind the EQUEST is to modify the cost function. This is done by adding another term, containing the gyroscope measurements. [Yalzer and Jensen (2011)]

$$J = \frac{1}{2} \sum_{i=1}^3 \left\{ \frac{1}{\sigma} (b_i - R_i^T(q)r_i)^T (b_i - R_i^T(q)r_i) \right\} + \frac{1}{2} (q - \hat{q}_{\text{prev}}) D (q - \hat{q}_{\text{prev}})$$

Further, a linear prediction term may be added due to the slow and predictable attitude change of a satellite to help filter out noise.

$$J = \frac{1}{2} \sum_{i=1}^3 \left\{ \frac{1}{\sigma} (b_i - R_i^T(q)r_i)^T (b_i - R_i^T(q)r_i) \right\} + \frac{1}{2} (q - \hat{q}_{\text{prev}}) D (q - \hat{q}_{\text{prev}}) + \frac{1}{2} (q - \hat{q}_{\text{prev}})^T S (q - \hat{q}_{\text{prev}})$$

The minimization problem is then put on standard form and solved by use of the Lagrangian multiplier method. The estimation method has been compared with the well known Extended Kalman Filter as well as an axes reference motion tracking system, and found to give accurate and reliable attitude information. This sensor method has been further developed by changing the quaternion substructure terms into quaternion products [Rinnan(2011)], which will represent proper rotations and find new attitude quaternions.

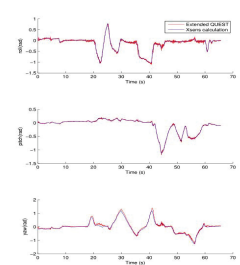


Figure 1: Comparison of EQUEST method to reference scene unit

Introduction

The main payload of our system is an IMU-camera which will take pictures of the atmosphere. The payload requires an accurate and reliable ADC system with a pointing accuracy of less than 10 degrees in every axis. We aim to complete these control goals by using magnetometers as actuators and a combination of a gyroscope, a magnetometer and the solar panels as sun sensors for attitude determination.

Control

The control part is comprised of two phases

- Detumbling. So far a dissipative controller has been explored and implemented which seems to be working. The choice for this controller is however still left open, as use of the more popular Euler controller might be considered a safer choice for cubesat with many other possibilities of failure.
- Stabilization. The choice of a stabilization controller is very much considered work in progress still, but for the purposes of prototype testing, a reference controller has been used that seems to be working during simulations. This controller is defined by the control law

$$m^* = -\frac{d}{\beta^2} \beta^* \times \omega_{IB}^* - \frac{k}{\beta^2} \beta^* \times \epsilon$$

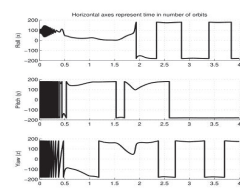


Figure 2: Euler angles for the full simulation.

Sun sensor

For this satellite we will utilize the solar panels already present for use as sun sensors. The values for power delivered from each of the panels will be provided by other parts of the satellite and is integrated into the EQUEST algorithm. For testing purposes an accelerometer has been used in the EQUEST method, a measurement which is easily replaceable by finding the directional sun intensity from the intensity measured by each of the panels to change the b and r vectors in the cost function. The sun vector is found via the intensity measured at each solar panel.

$$s_{\text{sun}} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} I_{\text{sun}_1} \\ I_{\text{sun}_2} \\ I_{\text{sun}_3} \\ I_{\text{sun}_4} \\ I_{\text{sun}_5} \\ I_{\text{sun}_6} \end{bmatrix}$$

Prototype

The prototype is built around a 64-pin Atmel ATMEGA2560 microcontroller unit and the use of a CHIMU IMU, containing a gyroscope, accelerometer and magnetometer. This prototype has been tested extensively, performing all the tasks expected from the ADC system of our satellite. The prototype was designed as a collaboration between group members working on determination and control, and these subsystems were implemented and tested separately. [Yalzer and Jensen(2011), Tolo(2011)]. These implementations have now been combined, and the complete ADC system runs on a single 256K MCU. Although the final design of the ADCS module is still incomplete the testing suggests this version is adequate for our mission.

Conclusion/Further Work

Simulations and testing so far suggests that the ADCS module designed by our group will fulfill the requirements from our main payload within the allowed power budget. As mentioned earlier there has been alterations made to the EQUEST method which still needs to be implemented on the prototype, as well as the need for a good earth albedo model for the sun sensor and final decisions for the controllers used. Even with these challenges remaining, our system is nearing completion and is en-route to launch in 2014.

References and other theses on this satellite project

- Yalzer, K.H., Jensen, K.L. "Development, Implementation and Testing of Two Attitude Estimation Methods for Cube Satellites", Master Thesis, Department of Engineering Cybernetics, NTNU, May 2011.
- Rinnan, T. "Development on the EQUEST method for attitude determination", Final year project, Department of Engineering Cybernetics, NTNU, December 2011.
- Tolo, Z. "Design and Implementation of Attitude Control for 3-axis Magnetic Coil Stabilization of a Spacecraft", Master Thesis, Department of Engineering Cybernetics, NTNU, May 2011.
- de Bruyn, D. "Power Distribution and Conditioning for a Small Student Satellite. Design of the NUTS Backplane EPS Module", Master Thesis, Department of Engineering Cybernetics, NTNU, June 2011.
- Brithen, G. "Nonlinear attitude control of a double Cubesat", Final year project, Department of Engineering Cybernetics, NTNU, May 2011.
- Holberg, F.S. "Optimal attitude control of a double cubesat using magnetometers", Final year project, Department of Engineering Cybernetics, NTNU, May 2011.
- Yalzer, K.H., Jensen, K.L., Gravdahl, J.T. "A comparison of attitude determination methods: theory and experiments". In proceedings of the 62nd International Astronautical Congress, Cape Town, South Africa, 3-7 October, 2011.
- Holberg, F.S., Yalzer, K.H., Jensen, Tolo, Z., Brithen, G. "ADCS for the NUTS satellite". Poster presented at SmallSat 2011, Logan, Utah

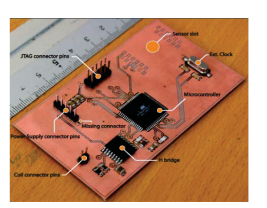



Figure 3: Prototype



Appendix B

Using Arduino with AVR Studio

B.1 Setting up external tools to flash Arduino

The easiest way to get this to work is by first installing the Arduino environment in the default folder.

Find “External tools” under the “Tools” tab in Atmel Studio, as seen in figure B.1

In the “Command” field, put the path to `avrdude.exe` in your Arduino installation. For example: `"C:/arduino-0022/hardware/tools/avr/bin/avrdude.exe"`

In “Arguments”, paste this line: `-CC:/arduino-0022/hardware/tools/avr/etc/avrdude.conf -v -v -patmega2560 -cstk500v1 -P\\.COM10 -b57600 -D -Uflash:w:"$(ProjectDir)Debug$(ItemF`
Edit the path to your Arduino installation and change the COM port and baud rate if necessary.

Make sure that the compiler is set to “release” instead of “debug” and that it outputs `.hex` files (found under properties for the current `.c` file you are editing.)

Then you need to hold the reset button on the Arduino board until right before you press the name of the external tool you just made, and it should flash the `.hex` file of your project to the Arduino board.

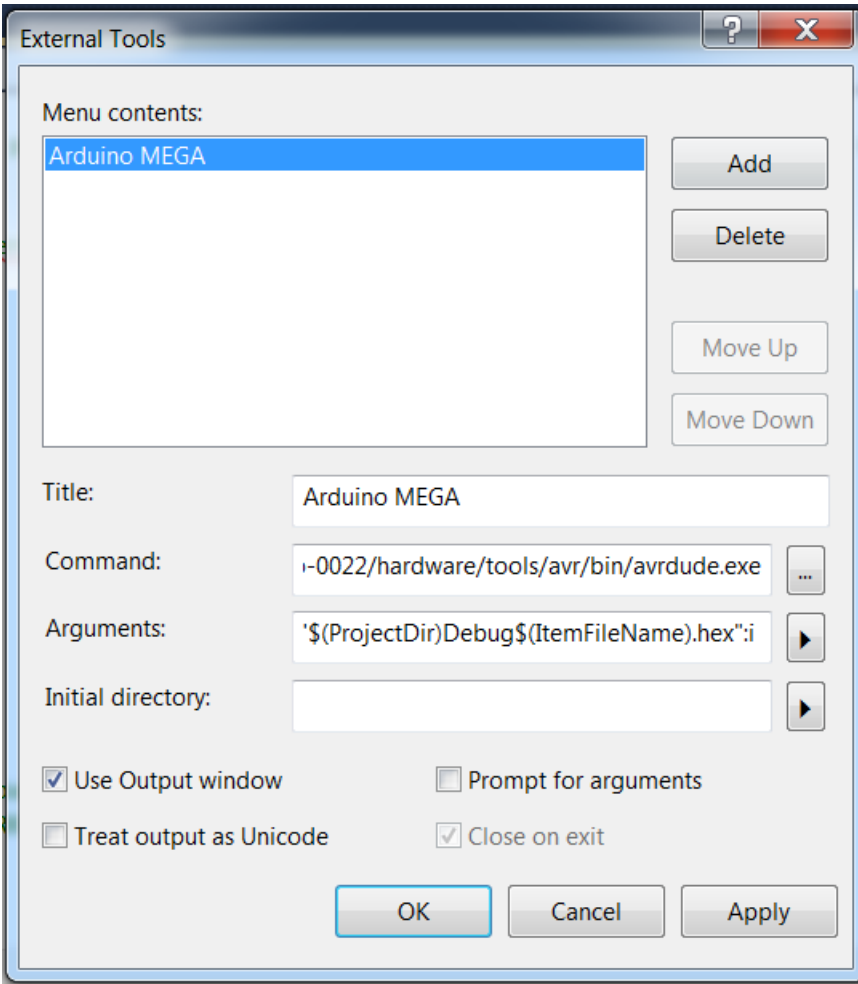


Figure B.1: External tools in Atmel Studio

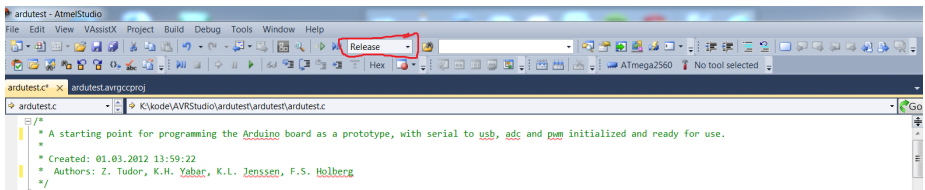


Figure B.2: Atmel Studio 6 toolbar: build “Release”

Appendix C

C-code

```

/*
 * A starting point for programming the Arduino board as a prototype, with serial to usb,
 * adc and pwm initialized and ready for use.
 *
 * Created: 01.03.2012 13:59:22
 * Authors: Z. Tudor, K.H. Yabar, K.L. Jenssen, F.S. Holberg
 */
#define F_CPU 16000000UL

#include <avr/io.h>
#include <stdio.h>
#include <util/delay.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>
#include <avr/interrupt.h>
#include <avr/sleep.h>
#include <avr/wdt.h>

#define DBL_EPSILON 2.2204460492503131e-16
#define BAUDRATE 115200 //set desired baud rate
#define MYUBRR (F_CPU/8/BAUDRATE-1) //calculate UBRR register value
#define MAXSIZE 10
#define TINY 1.0e-20;

void WDT_off(void){
    wdt_reset();
    MCUSR &= ~(1<<WDRF);
    WDTCSR |= (1<<WDCE) | (1<<WDE);
    WDTCSR = 0x00;
}

void USART_Init(unsigned int ubrr){
    //Set UBRR register for USART0 and USART1
    UBRR0H = (unsigned char)(ubrr>>8);
    UBRR0L = (unsigned char)ubrr;
    UBRR1H = (unsigned char)(ubrr>>8);
    UBRR1L = (unsigned char)ubrr;
    UCSR0B |= (1<<RXEN0)|(1<<TXEN0); //Enable Transmitter\Receiver
for USART0
    UCSR1B |= (1<<RXEN1)|(1<<TXEN1)|(1<<RXIE1); //Enable Transmitter\Receiver and
interrupt for USART1
    UCSR1A |= (1<<U2X1); //Enable double speed
for USART1
    UCSR0A |= (1<<U2X0); //Enable double speed
for USART0
    sei();
    //Activate interrupt
}
}

```

```

//Send data to sensor
void USART_vSendByte(uint8_t u8Data){
    while((UCSR1A &(1<<UDRE1)) == 0);           // Wait if a byte is being transmitted
        UDR1 = u8Data;                           // Transmit data
    }

//Send data to computer computer
void USART_vSendByte_Pc(uint8_t u8Data, FILE *stream){
    while((UCSR0A &(1<<UDRE0)) == 0);           // Wait if a byte is being transmitted
        UDR0 = u8Data;                           // Transmit data
    }

//Receive data from sensor
char USART_vReceiveByte(void){
    while((UCSR1A&(1<<RXC1)) == 0);             // Wait until a byte has been
received
    return UDR1;                                // Return received data
}

/* Initialize PWM to desired operation by modifying its registers */
void PWM_init(){
    // Waveform generation mode: PWM, Phase and Frequency Correct
    TCCR1A |= (0<<WGM11)|(0<<WGM10);
    TCCR1B |= (1<<WGM13)|(0<<WGM12);

    // No prescaling
    TCCR1B |= (0<<CS12)|(0<<CS11)|(1<<CS10);

    // Clear on compare match when up-counting. Set on compare match when downcounting.
    TCCR1A |= (1<<COM1A1)|(0<<COM1A0);
    TCCR1A |= (1<<COM1B1)|(0<<COM1B0);
    TCCR1A |= (1<<COM1C1)|(0<<COM1C0);

    // No filtering
    TCCR1B |= (0<<ICNC1)|(0<<ICES1);

    // Resolution:
    ICR1 = 0xFFFF;

    // DDRB |= (1<<DDB0)|(1<<DDB4);
    DDRB = 0xFF;
}

```

```

/* Initialize ADC on PORTF to read solar cells */
void ADC_init()
{
    //AREF = Vcc
    ADMUX |= (1<<REFS0);

    // ADC Enable and prescaler of 128
    // 16000000/128 = 125000
    ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);
}

uint16_t ADC_read(uint8_t ch)
{
    // select the corresponding channel 0~15

    // To account for ch between 7 and 16, need to set bit MUX5 in ADCSRB HIGH, else it
    should be LOW
    if (0x08 & ch) { ADCSRB |= (1<<MUX5); }
    else { ADCSRB &= ~(1<<MUX5); }

    // ANDing with 0x07 will always keep the numerical value
    // of 'ch' between 0 and 7
    ch &= 0x07;

    ADMUX = (ADMUX & 0xF8)|ch;    // clears the last 3 bits before ORing

    // start single conversion
    // write '1' to ADSC
    ADCSRA |= (1<<ADSC);

    // wait for conversion to complete
    // ADSC becomes '0' again
    // till then, run loop continuously
    while(ADCSRA & (1<<ADSC));

    return (ADC);
}

//Send data to computer using printf
static FILE mystdout = FDEV_SETUP_STREAM(USART_vSendByte_Pc, NULL, _FDEV_SETUP_WRITE);

```

```

int main(void)
{
    USART_Init(MYUBRR);
    stdout = &mystdout;
    WDT_off();
    PWM_init();
    ADC_init();
    uint16_t adc_result0;
    uint16_t to_pwm;

    printf("%d\n\r",1);

    while(1)
    {
        adc_result0= ADC_read(15); //Read ADC channel 15
        OCR1B = adc_result0*64; //Output scaled ADC reading with PWM on PWM clock B
        //OCR1A = 0x0000; //PWM clock A
        //OCR1C = 0xFFFF; //PWM clock C
        printf("%u\n\r",to_pwm); //Output scaled ADC reading over serial USB
        _delay_ms(50);
    }
}

```


Appendix D

Norsk sammendrag

NUTS (NTNU Test Satellite) er en satellitt som skal bygges som del av et CubeSat prosjekt for studenter ved Norges teknisk-naturvitenskapelige universitet, NTNU. Prosjektet ble startet i september 2010 som en del av det norske studentsatellittprogrammet i regi av NAROM (Norsk senter for romrelatert opplæring). NUTS prosjektmål er å designe, produsere og lansere en dobbel CubeSat innen 2014. Som nyttelast vil det være et IR-kamera som observerer bølger i luft-glød-laget, samt et kortrekkevidde RF eksperiment. Satellitten vil fly med to transceivere i amatørradioband. Siste års masterstudenter fra flere institutter er de viktigste bidragsyterne til prosjektet og de fleste av systemkomponenter er designet og bygget av studenter.

Beskrevet i denne artikkelen er en vurdering av ulike strategier for valg og utforming av Attitude Estimation and Control System, ADCS, for NUTS satellitten, effektivt neste iterasjon av det arbeidet som ble gjennomført i fjor, med mye vekt lagt på de spesielle kravene for et system som ikke kan repareres eller vedlikeholdes på en tradisjonell måte.

Ulike alternativer for sensorer og aktuatorer er utforsket og evaluert, på jakt etter velprøvde produkter med lav sjans for svikt, og konkluderer med at Analog devices skal være en trygg og pålitelig leverandør av inertielle måleenheter for satellitt vår.

Mens Extended Kalman Filteret er den filtreringsmetoden som trolig er den mest brukte i nesten alle applikasjoner, har den noen svært alvorlige begrensninger som er spesielt viktig i en slik krevende miljø som en cube satellitt.

To andre metoder for estimering av orienteringen evalueres, nemlig Extended QUaternion ESTimator og nyutviklede lineære observatører. Mens de begge er svært attraktive alternativer med mange egenskaper som passer med våre krav, er det fortsatt noen spørsmål om de er like pålitelige som den velprøvde EKF.

Den åpen kildekode elektroniske prototype plattformen Arduino begynner å få fotfeste hos amatører over hele verden, og det er utforsket om det også kan brukes for hurtig prototyping og om det er egnet for hyppige design-iterasjoner i vår satellitt prosjekt. Friheten det gir kombinert med en standard breadboard synes å gjøre det til et meget godt valg for alle prosjekter som ofte skal endres eller forbedre designet i henhold til annen maskinvare eller programvarehensyn.

Bibliography

- [1] *Atmega 2560/2561 Datasheet*, retrieved January 2012, <http://www.atmel.com/Images/doc2549.pdf>.
- [2] *GOMspace NanoMind A712C ADCS Datasheet*, retrieved May 2012, <http://gomspace.com/documents/GS-DS-NM712C-1.1.pdf>.
- [3] *SN754410 Quad Half H-bridge Datasheet*, retrieved February 2012, <http://www.ti.com/lit/ds/symlink/sn754410.pdf>.
- [4] G. Bråthen. Nonlinear attitude control of a double cubesat using magnetorquers. Technical report, Dept. of engineering cybernetics, Norwegian university of science and technology, 2011.
- [5] B. Despont. System engineering and development and test of the adcs breadboard for swisscube. Master's thesis, EPFL Lausanne, 2007.
- [6] O. Egeland and J.T. Gravdahl. *Modeling and simulation for automatic control*. 2002.
- [7] K.M. Fauske. Attitude Stabilization of an Underactuated Rigid Spacecraft. *Siv-ing theis, Department of Engineering and Cybernetics, Norwegian University of Technology and Science*, pages 27–30, 2003.
- [8] L.A. McGee G.L. Smith; S.F. Schmidt. Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle. *National Aeronautics and Space Administration*, 1962.
- [9] Johansen Grip, Fossen and Saberi. Semi-global stable attitude observer with gyro bias state estimation extended to time-varying reference vectors (uses a projection algorithm). *IEEE Trans. on Automatic Control*, 2012.

- [10] W.R. Hamilton. On quaternions, or on a new system of imaginaries in algebra. *Philosophical Magazine*, 25, 1844.
- [11] T.A. Johansen A. Saberi H.F. Grip, T.I. Fossen. Globally exponentially stable attitude and gyro bias estimation with application to gnss/ins integration. *Submitted to IEEE Transactions on Automatic Control*, 2012.
- [12] T.A. Johansen A. Saberi H.F. Grip, T.I. Fossen. Integration of gnss and imu measurements with gyro bias estimation and explicit stability guarantees. In *Proc. American Control Conference, June 27-29, Montréal, Canada*, 2012.
- [13] F. S. Holberg. Optimal attitude control of a double cubesat using magnetorquers. Technical report, Dept. of engineering cybernetics, Norwegian university of science and technology, 2011.
- [14] L.E. Jacobsen. Electrical power system of the ntnu test satellite. Master's thesis, NTNU, 2012.
- [15] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82:35–45, 1960.
- [16] T.R. Kane, P.W. Likins, and D.A. Levinson. *Spacecraft dynamics*, volume 1. New York, McGraw-Hill Book Co, 1983.
- [17] M.L. Psiaki. Attitude-determination filtering via extended quaternion estimation. *Journal of Guidance, Control, and Dynamics*, 23(2):206–214, 2000.
- [18] T.B. Rinnan. Development on the equest method for attitude determination. Technical report, Dept. of engineering cybernetics, Norwegian university of science and technology, 2011.
- [19] P.K. Soglo. 3-aksestyring av gravitasjonsstabilisert satellitt ved bruk av magnetspoler. *Siv. Ing Thesis, Department of Engineering Cybernetics, NTNU (Norwegian)*, 1994.
- [20] T. Sundlisæter. Spacecraft attitude and orbit estimation using gps and inertial measurements. Master's thesis, Norwegian University of Science and Technology, 2012.
- [21] K. Svartveit. Attitude determination of the ncube satellite. Master's thesis, NTNU, 2003.

- [22] Z. Tudor. Design and implementation of attitude control for 3-axes magnetic coil stabilization of a spacecraft. Master's thesis, Norwegian University of Science and Technology, 2011.
- [23] B. Vik. Integrated Satellite and Inertial Navigation Systems. *Department of Engineering Cybernetics, NTNU*, 2009.
- [24] K.L. Jenssen & K.H. Yabar. Development, implementation and testing of two attitude estimation methods for cube satellites. Master's thesis, Norwegian University of Science and Technology, 2011.