Sensor modeling, attitude determination and control for micro-satellite

Svein Tohami El Moussaoüi Brembo

NTNU

NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for informasjonsteknologi,
matematikk og elektroteknikk
Institutt for teknisk kybernetikk

MASTEROPPGAVE

Kandidatens navn: Svein Tohami El Moussaoui Brembo

Fag: Teknisk Kybernetikk

-

Oppgavens tittel (norsk): Sensormodellering og estimering av orientering for mikrosatelitt.

Oppgavens tittel (engelsk): *Sensor modelling and attitude determination for micro-satellite.*

Oppgavens tekst:

Kongsberg Defensive & Aerospace er et eget forretningsområde innen Kongsberg Gruppen ASA. Resultatområdet Missile & Space, som er en del av Kongsberg Defence & Aerospace (KDA), deltar i en internasjonal studie på bruk av cluster satellitter (satellitter i gruppe). KDA har ansvaret for attitude control, determination og posisjons systemet. Satellittene skal utføre optiske målinger og radarmålinger. Mikrosatellittene skal ha 3-akse styring effektuert gjennom bruk av fire reaksjonshjul. Det ønskes en nøyaktighet på ± 0.1 ° eller bedre omkring hver akse og mikrosatellitten skal styres aktivt i azimuth, dvs omkring z-aksen. Flere satellitter observerer samme gjenstand og det er viktig at vi kjenner attitudevinklene med stor nøyaktighet, ønsket er 0.001 ° i alle akser. Oppgaven blir å utvikle en matematisk modell for satellitten og sensorene som skal brukes i satellitten. De viktigste sensorene vil være stjernesensor, solsensorer og jordsensorer og GPS. Stjernesensoren, solsensorene og jordsensorene kan brukes til attitude informasjon, GPS skal også benyttes til attitude informasjon. Simuleringer skal utføres i Simulink.

Satellittdata:

Satellitten skal gå rett over polene og ha en sirkulær bane med en høyde på 600 km.
Treghetsmomentene til satellitten: I_X=4 kgm^2, I_y=4 kgm^2, I_z=3 kgm^2

Oppgaver:

1. Modeller satellitten.
2. Sett opp en matematisk modell for magnetometerene. Det bør brukes et 3-akset magnetometer. Vinklene som måles i de 3 akser skal være gitt av IGRF modellen for jordas magnetfelt. Benytt 1 nT = 4 arcsec. Legg inn muligheter for støy.
3. Sett opp en matematisk modell for solsensoren.
4. Stjernesensoren har svært krevende modellering og denne kan modelleres som en målt vinkel addert støy.
5. Orienteringsnøyaktigheten skal være ≤ 0.001 grad om alle 3-akser. Det må velges sensorer som kan gi denne nøyaktigheten. En stjernesensor fra DTU (Danske tekniske universitetet) har en nøyaktighet på 1 arcsec i pitch og azimuth og 5 arcsec i rull. Vi kan kanskje benytte 2 stjernesensorer? DTU leverer magnetometre som måler alle 3 akser med en nøyaktighet på 2 arcsec dersom de sitter ute på en lang bom. Det er mer realistisk å anta magnetometer nøyaktigheter på 0.01 °. Magnetometrene kan ha en sampelfrekvens som er 100 Hz eller høyere. Stjernesensoren har en sampelfrekvens på rundt 0.5 Hz. Utvikle et Kalmanfilter. Filtret bør benytte magnetometerene og solsensor som hovedsensorer, men får korrigeringer fra stjernesensoren.
6. Finn hvilke oppdateringsrate de forskjellige sensorene kan klare seg med. Benytt realsistisk støy. Filteret skal være slik at det finner ut når/hvis stjernesensoren måler feil og filtrerer bort disse målingene.
7. Utform et filter som estimerer tilstandene slik at hvis magnetometerene og/ eller solsensoren svikter kan vi fortsatt få nøyaktige vinkelmålinger ved å bruke stjernesensoren. Benytt realistisk støy og finn hvilke nøyaktighet vi kan oppnå.
8. Bruk tilbakekobling fra de estimerte tilstandene for regulering av orientering.

Oppgaven gitt: 12/1-05

Besvarelsen leveres: 8/7-05

Besvarelsen levert:

Utført ved Institutt for teknisk kybernetikk

Veileder: Åge Skullestad, Kongsberg Defence & Aerospace

Trondheim, den 12/1-05

Jan Tommy Gravdahl

Jan Tommy Gravdahl
Faglærer

Preface

This Master Thesis has been carried out at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU). I would like to thank my supervisors Åge Skullestad at Kongsberg Defence & Aerospace (KDA), and associate professor Jan Tommy Gravdahl at Department of Engineering Cybernetics.

Svein Tohami El Moussaoüi Brembo
Trondheim 08.07.05

ABSTRACT

This thesis deals with the attitude estimation for a micro satellite in a polar, low earth orbit (LEO, 600km). The satellite is to be used for optical and radar measurements, which gives stringent demands on the attitude accuracy. The attitude is estimated with a continuous extended Kalman filter utilizing measurements from a magnetometer, sun sensor and star sensors.

This thesis gives a mathematical model of the magnetometer, the sun sensor and the star sensor, and describes how to implement each of them into a Kalman filter and how to combine the measurements from the magnetometer and star sensor into an attitude estimation with a Gauss-Newton method. Different ways to combine the sensors will be presented and simulated. A fault detection scheme, detecting and removing faulty measurement are also given.

By combining two star sensors in the Kalman filter, the filter will be able to estimate the attitude within 1.8 arcseconds in all 3-axis and it is able to estimate the attitude regardless of the of the initial estimation error. Combining the star sensors with the magnetometer and sun sensor adds redundancy to the system, and makes it possible to detect and remove faulty sensors from the system, making the attitude estimator more robust.

Contents

List of Figures

Chapter 1

Introduction

Kongsberg Defence and Aerospace (KDA) is part of a international study on cluster satellites. Their main responsibility is the attitude determination, control and position system of the satellite. The satellites are micro-satellites with a circular, low earth orbit(LEO) of 600km, passing above the poles.

The satellites mission is to conduct optical and radar measurements. This gives stringent demands on the attitude control and estimation. The attitude estimator have to be able to estimate the attitude within en error of $0.001°$ and the attitude control has to be within an error of $0.1°$.

This thesis aims to develop an attitude estimator for a the satellite, using an extended Kalman filter with measurement from a magnetometer, star sensor and sun sensor. It will also describe the mathematical model for the sun sensor and magnetometer used in the Kalman filter. An algorithm for detecting and removing faulty sensors from the attitude estimator will also be presented.

Outline of this report

- Chapter 2, describes theory and background material used in the rest of the thesis.
- Chapter 3, describes the mathematical satellite model.
- Chapter 4, describes the environment surrounding the satellite, such as the earth's magnetic field, the satellite's orbit and the earths gravitational pull ont the satellite.
- Chapter 5, gives a detailed description of the sensors used.
- Chapter 6, gives a short description of actuators used in space.
- Chapter 7, deals with the Kalman filter. It describes the discreet and continuous Kalman filter, how to implement the measurements in the Kalman filter, and how to use a unit quaternion in a Kalman filter.
- Chapter 8, simulates and describes different Kalman filters with different measurements This chapter also describes and simulates the faulty sensor detection algorithm.
- Chapter 9 is the Conclusion.

Tools

The simulation, and calculations has been done in Matlab 7.0(R14), Simulink 6.0(R14) and the GNC toolbox[1].

[1] NTNU-MSS, Marine Systems Simulator (2005). Norwegian University of Science and Technology, Trondheim, Norway. Available at <www.cesos.ntnu.no/mss>

Chapter 2

Background

2.1 Reference frames

To be able to orient the satellite, the satellite's orientation must be described within a reference frame.

BODY frame

The body frame is fixed to the vessel, with :

- x_b-axis pointing from the back to the front (a.k.a. forward).
- z_b-axis pointing from the top to the bottom (a.k.a. down, nadir).
- y_b-axis completing the right hand system.

The origin of the frame is located in the mass center of the vessel. The orientation of the satellite is described relative to the orbit frame.(Fossen 2002) (Ose 2004)

ORBIT frame

The orbit is defined by the motion of the satellite's mass center. The orbit frame is described with:

- x_o-axis pointing in the direction of motion, tangential to the orbit
- z_o-axis pointing to the center of the Earth (nadir).
- y_o-axis completing the right hand system.

The orbit frame rotates relatively to the ECI frame. If the orbit of the satellite is elliptic, with the Earth in one of the foci, the axis will not be aligned with the velocity vector of the satellite.(Ose 2004) (Fauske 2002)

ECI frame

The ECI frame, is a nonaccelerating reference frame in which Newtons laws of motion apply. The frame is described by :

- x_i-axis pointing in the vernal equinox direction (the line from Earths origin through the Sun on the first day of spring).

- z_i-axis pointing upwards from the origin through the geographical north pole
- y_i-axis completing the right hand system.

The origin of the frame is located at the center of the Earth. (Fossen 2002) (Sellers 2000)

ECEF

The ECEF frame is fixed to the Earth. The frame is described by :

- x_e-axis points form Earth center towards the point were Greenwich intersects with the equator.
- z_e-axis pointing upwards from the origin through the geographical north pole
- y_e-axis completing the right hand system.

The origin of the frame is located at the center of the Earth. Since the ECEF is fixed to the Earth, it rotates with an angular speed of $\omega_e = 7.2921 \cdot 10^{-5} rad/sek$ around the z_i axis of the ECI frame. (Ose 2004) (Fossen 2002)

2.2 Vector mathematics and transformations

This section contains some mathematics necessary to calculate the orientation of the satellite.

2.2.1 Skew symmetric matrixes

A skew symmetric matrix is defined as:

$$\boldsymbol{S}(\boldsymbol{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{2.1}$$

and has the following properties :

- $\boldsymbol{S}^T(\boldsymbol{x}) = -\boldsymbol{S}^T(\boldsymbol{x})$
- $(\boldsymbol{S}^2(\boldsymbol{x}))^T = \boldsymbol{S}^2(\boldsymbol{x})$
- $\boldsymbol{S}^T(\boldsymbol{x})\boldsymbol{y} = \boldsymbol{x} \times \boldsymbol{y}$
- $\boldsymbol{S}(\boldsymbol{x})\boldsymbol{y} = -\boldsymbol{S}(\boldsymbol{y})\boldsymbol{x}$

2.3 Euler parameter

2.3.1 Euler angles

Euler angels are usually used to describe the rotations of a rigid body system. Since the satellite is a rigid body system it is possible to describe the rotation and attitude of the satellite with the Euler angels :

$$\boldsymbol{\Theta} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{2.2}$$

For the satellite:

- ϕ is roll, the rotation around the x_o-axis.
- θ is pitch, the rotation around the y_o-axis.
- ψ is yaw angle, the rotation around the z_o-axis.

Pros and cons of the Euler parameters

- The Euler angel represent an intuitive representation for the attitude of a object in a 3D space. It is therefore easy to relate to.
- Using Euler parameters to describe the attitude, may result in singularities.

2.3.2 Unit quaternions

One way to avoid the singularity problems of the Euler angels are to use the Unit quaternions. The quaternion **q** has four parameters, one real η and three imaginary $\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \varepsilon_3 \end{bmatrix}$. A unit quaternion has to satisfy $\boldsymbol{q^T q = 1}$, and therefore must be in a set Q, defined by:

$$\boldsymbol{Q} = \{\boldsymbol{q} \mid \boldsymbol{q^T q} = 1,\, \boldsymbol{q} = \begin{bmatrix} \eta, & \boldsymbol{\varepsilon} \end{bmatrix}^T, \boldsymbol{\varepsilon} \in \Re^3 \, and \, \eta \in \Re\} \tag{2.3}$$

The restriction $\boldsymbol{q^T q} = 1$ also implies that :

$$\eta^2 + \varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 = 1 \tag{2.4}$$

where η is defined as :

$$\eta = \cos\frac{\theta}{2} \tag{2.5}$$

and $\boldsymbol{\varepsilon}$ is defined as

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \varepsilon_3 \end{bmatrix}^T = \boldsymbol{\lambda} \sin\frac{\theta}{2} \tag{2.6}$$

where $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{bmatrix}$ is a unit eigenvector satisfying:

$$\begin{aligned} \boldsymbol{\lambda} &= \pm\frac{\boldsymbol{\varepsilon}}{\sqrt{\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}}} && (2.7) \\ \sqrt{\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}} &\neq 0 && (2.8) \end{aligned}$$

The algorithm to transform a set of parameters from euler angels to unit quaternions, or from unit quaternions to euler angels can be found in both in Fossen (2002) and Egeland & Gravdahl (2002) where they use that the fact that $\boldsymbol{R(\theta)} = \boldsymbol{R(q)}$, where $\boldsymbol{R}$ are the rotation matrices defined below, to derive the conversion logarithms.

2.4 Rotation matrices

A rotation matrix $\boldsymbol{R}$ is either used to move from one coordinate frame to another or to rotate within one frame. The rotation matrix is an element in a *special orthogonal group of order 3*, SO(3), defined in equation (2.9). In more mathematical terms $R \in SO(3)$.

$$SO(3) = \{\mathbf{R} \mid \mathbf{R} \in \Re^{3\times 3}, \mathbf{R} \quad is \quad ortogonal \quad and \quad det(\mathbf{R}) = 1\} \tag{2.9}$$

This means that $\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I}$ and $\mathbf{R}^{-1} = \mathbf{R}^T$.

The simplest rotation matrix is a matrix representing a rotation (β) around one axis (λ) and is given by:

$$\boldsymbol{R}_{\lambda,\beta} = \boldsymbol{I}_{3\times 3} + \sin\beta \boldsymbol{S}(\boldsymbol{\lambda}) + (1-\cos\beta)\boldsymbol{S}^2(\boldsymbol{\lambda}) \tag{2.10}$$

This gives the following rotation matrixes for euler angels:

$$\boldsymbol{R}_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{2.11}$$

$$\boldsymbol{R}_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{2.12}$$

$$\boldsymbol{R}_{z,\psi} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ 0 & \cos\psi & 0 \\ \sin\psi & 0 & 1 \end{bmatrix} \tag{2.13}$$

In the rest of this paper the following notation will be used when moving from one coordinate frame to another using the rotation matrix:

$$r^{to} = \mathbf{R}^{to}_{from} r^{from} \tag{2.14}$$

One rotation matrix that will be used quite a lot in this paper is the rotation matrix from body to orbit , $\boldsymbol{R}^o_b(\boldsymbol{q})$, and the rotation matrix from orbit to body $\boldsymbol{R}^b_o(q)$. Fossen (2002) derives the rotation matrix $\boldsymbol{R}^o_b(q)$ as:

$$\boldsymbol{R}^o_b(q) := \boldsymbol{R}_{\eta,\varepsilon} = \boldsymbol{I}_{3\times 3} + 2\eta\boldsymbol{S}(\boldsymbol{\varepsilon}) + 2\boldsymbol{S}^2(\boldsymbol{\varepsilon}) \tag{2.15}$$

since

$$\boldsymbol{R}^b_o(q) = (\boldsymbol{R}^o_b(q))^T \tag{2.16}$$

this gives that

$$\boldsymbol{R}^b_o(q) = \boldsymbol{I}_{3\times 3} - 2\eta\boldsymbol{S}(\boldsymbol{\epsilon}) + 2\boldsymbol{S}^2(\boldsymbol{\epsilon}) \tag{2.17}$$

2.5 Time

Time is an important parameter when calculating the position of an object in orbit. The standard SI[1] unit(International System of Units) for time is a second(sec). The second is then used to define minutes[min], hours[h], days, months and years. There are a fixed ratio between seconds, minutes, hour, day and week, but no fixed ratio between days, months and years. This is due to the use of leap-days, it is therefore inconvenient to use for computer computations. A solution to this is to use the Julian Day.

[1] abbreviated from the French: Système International d'Unités

2.5.1 International atomic time(TAI)

International atomic time, is a very accurate and stable reference time. It uses the radiation period of a cesium nuclide, 133 Ce, where 9,192,631,770 periods constitute 1 second in the SI system.

2.5.2 Universal Time(UT)

Universal Time, also called Greenwich mean time(GMT), is the mean solar time at the Royal Greenwich Observatory near London in England, which coincide with the 0° longitude.

2.5.3 Coordinated Universal Time(UTC)

Coordinated Universal Time, also called Zulu time (Z), was introduced and broadcasted in January 1972 (Kristiansen 2000). A second in UTC is equal to a second in TAI, but it is kept within 0.90 seconds of the actual rotation of the Earth, by correcting it with 1 second steps, usually at the end of June an December (This gives rise to the leap year).

2.5.4 Civil Time

Different time zones, give rise to the Civil time (T_{civil}), which is the time observed by people on their clocks. Civil time differs from the UT, by a an integer number of hours, and can be roughly calculated by :

$$T_{civil} \approx UT \pm (L + 7.5)/15 \tag{2.18}$$

where L is the longitude, given in degrees (with positive sign in eastern direction), and Ut and T_{civil} given in integer numbers, and T_{civil} is irrespective of daylight-saving time.

2.5.5 Julian day

The Julian calender is a continnous time count from the number of days since Greenwich noon on January 1, 4713BC(Wertz & Larson 1999). This is the solution adopted for astronomical use, and was proposed by the Italian scholar Joshep Scaliger in 1582 AD. The Julian Day can be calculated with:

$$JD = day + \frac{153 - m}{5} + 365y + y/4 - 32083 \tag{2.19}$$

where:

$$m = month + 12a - 3 \tag{2.20}$$

$$y = year + 4800 - a \tag{2.21}$$

and

$$a = \frac{14 - month}{12} \tag{2.22}$$

2.5.6 Modified Julian day(MJD)

The Julian Date is, as mentioned above, adopted for astronomical use, but for space application it present a minor problem, because it starts at 12:00 UT,instead of 00:00 UT, as the civil calender does. To remedy this the Modified Julian Date is used. The MJD is given by:

$$MJD = JD - 2,400,000.5 \tag{2.23}$$

and starts at midnight.

2.6 The Orbit

The mathematical equations used to describe the motion of a satellite and planets to day, was described by Johannes Kepler [1571-1630]. They were founded on the observation of Tycho Brahe [1546-1601], and deduced from Newton's equation of motion. Kepler described the following three laws for the motion of a satellite (Sellers 2000) :

Kepler's First Law The orbit of the planets are ellipses with the sun at one focus.

Kepler's Second Law The line joining a planet to the sun sweeps out equal areas in equal times.

Kepler's Third Law The square of the orbital period, the time it takes to complete one orbit, is directly proportional to the cube of the mean or average distance between the Sun and the planet.

2.6.1 Orbit elements

To define and describe a satellites orbit, one need six parameters or elements (Wertz & Larson 1999). The classical orbit elements are, described for a satellite in Earth orbit(Vallado 2001, Sellers 2000):

Semimajor axis a **:** is the axis running from the center of the orbit (ellipse), through a focus, and to the edge of the orbit. It is used to describe the size of the orbit.

Eccentricity e **:** is used to describe the shape of the orbit. If $e = 0$ the orbit is circular, if $0 < e =< 1$ it is a ellipse if $e = 1$ it is a parabola and if $e > 1$ the orbit is a hyperbola.

Inclination i is the tilt of the orbit plane in respect to the plane of reference (for a satellite orbiting Earth this is the equatorial plane).

Right ascension of the ascending node Ω is the angle from the vernal equinox to the ascending node (the point where the satellite crosses the equatorial plane from north to south).

Argument of perigee ω is the angle along the orbital plane from the ascending node to the point (called perigee) in the orbit (ellipse) closest to Earth (Earth lies in a focus point) .

True anomaly ν **:** is the angle along the orbital path from perigee to the satellites position.

Other elements that often is used are the :

Mean motion n which is used to describe the satellite's average angular motion over one orbit.

Mean Anomaly M which is the uniform angular motion of a circle.

The ascension of the zero meridian θ is the angle from the vernal equinox to the zero meridian.

Chapter 3

Satellite model

3.1 The inertia matrix

The inertia matrix of the satellite is given by:

$$\boldsymbol{I} := \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zy} & -I_{zy} & I_z \end{bmatrix} \tag{3.1}$$

If the physical model is symmetric, or is assumed to be symmetric, the inertia matrix becomes :

$$\boldsymbol{I} := \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{3.2}$$

Where I_x-, I_y- and I_z-axes are the moment of inertia about x_b, y_b and z_b, and they are calculated like this :

$$I_x = \int_V (y^2 + z^2)\rho_m dV \tag{3.3}$$

$$I_y = \int_V (x^2 + z^2)\rho_m dV \tag{3.4}$$

$$I_z = \int_V (y^2 + x^2)\rho_m dV \tag{3.5}$$

3.2 Dynamics

The dynamics of the satellite can be derived by using a Newton-Euler formulation, where the equation of motion is derived from the definition of angular moment. This leads to the following models, according to Øverby (2004) :

$$\boldsymbol{I}\dot{\boldsymbol{\omega}}^b_{ib} + \boldsymbol{\omega}^b_{ib} \times (\boldsymbol{I}\boldsymbol{\omega}^b_{ib}) = \boldsymbol{\tau}^b, \boldsymbol{\tau}^b = \boldsymbol{\tau}^b_{grav} + \boldsymbol{\tau}^b_m \tag{3.6}$$

$$\boldsymbol{I}\dot{\boldsymbol{\omega}}^b_{ib} + \boldsymbol{S}(\boldsymbol{\omega}^b_{ib})\boldsymbol{I}\boldsymbol{\omega}^b_{ib} = \boldsymbol{\tau}^b \tag{3.7}$$

$$\boldsymbol{\omega}^b_{ib} = \boldsymbol{\omega}^b_{io} + \boldsymbol{\omega}^b_{ob} = R^b_o\boldsymbol{\omega}^o_{io} + \boldsymbol{\omega}^b_{ob} \tag{3.8}$$

where $\boldsymbol{\omega}^b_{ib}$, $\boldsymbol{\omega}^b_{io}$ and $\boldsymbol{\omega}^o_{io}$ are the angular velocity of the satellite from body to inertia decomposed in the body frame, from orbit to inertia decomposed in the body frame and from orbit to inertia decomposed in the orbit frame. $\boldsymbol{\tau}^b$ is the angular momentum decomposed in body frame.

Differentiating the last equation (3.8) gives

$$\dot{\boldsymbol{\omega}}_{ib}^{b} = \dot{\boldsymbol{\omega}}_{ob}^{b} - \boldsymbol{S}(\boldsymbol{\omega}_{ob}^{b})\boldsymbol{R}_{o}^{b}\boldsymbol{\omega}_{io}^{o} \tag{3.9}$$

and

$$\dot{\boldsymbol{\omega}}_{ob}^{b} = \dot{\boldsymbol{\omega}}_{ib}^{b} + \boldsymbol{S}(\boldsymbol{\omega}_{ob}^{b})\boldsymbol{R}_{o}^{b}\boldsymbol{\omega}_{io}^{o} \tag{3.10}$$

3.3 Kinematics

The kinematics of the satellite is derived by integrating the angular velocity of the satellite. It is used to describe the orientation of the satellite. In unit quaternions it will, according to Øverby (2004) and Egeland & Gravdahl (2002), be described by (3.11);

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{\eta} \\ \dot{\boldsymbol{\varepsilon}} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\boldsymbol{\varepsilon}^{T} \\ \eta\boldsymbol{I}_{3\times3} + \boldsymbol{S}(\boldsymbol{\varepsilon}) \end{bmatrix}\boldsymbol{\omega}_{ob}^{b} \tag{3.11}$$

and the angular velocity of the body frame relative to the orbit frame will be given by :

$$\boldsymbol{\omega}_{ob}^{b} = \boldsymbol{\omega}_{ib}^{b} - \boldsymbol{\omega}_{o}c_{1}^{b} = \boldsymbol{\omega}_{ib}^{b} - R_{o}^{b}\boldsymbol{\omega}_{ib}^{o} \tag{3.12}$$

or by integrating equation (3.10)

Chapter 4

Satellite Environment

4.1 Simple orbit estimator

The orbit of the satellite considered in this thesis, is polar and it is assumed to be circular. It is therefore sufficient to use a simple orbit estimator. Svartveit (2003) proposed a simple orbit estimator, which will be used in this thesis. It will be repeated her for convenience.

The mean anomaly of a kepalerian orbit element is uniform in time. The prediction is therefore:

$$M(t_0 + t) = M(t_o) + n \cdot t \tag{4.1}$$

To solve Kepler's equation, the relation between the eccentric anomaly and the mean anomaly has to be known. This relation is given by (4.2).

$$E(t) = M(t) + e \sin E(t) \tag{4.2}$$

where e is the orbit eccentricity and t is the time. Given the eccentric anomaly, the vector from the center of the earth to the satellite is given by the following equation:

$$\boldsymbol{r}_{OC} = a \begin{bmatrix} \cos E - e \\ \sqrt{1 - e^2} \sin E \\ 0 \end{bmatrix} \tag{4.3}$$

$\boldsymbol{r}_{OC}$ can be transformed into the ECEF frame with :

$$\boldsymbol{r}_E = \boldsymbol{R}_z(-\Omega + \Theta)\boldsymbol{R}_x(-i)\boldsymbol{R}_z(-\omega)\boldsymbol{r}_{OC} \tag{4.4}$$

and into the ECI frame with:

$$\boldsymbol{r}_I = \boldsymbol{R}_z(-\Omega)\boldsymbol{R}_x(-i)\boldsymbol{R}_z(-\omega)\boldsymbol{r}_{OC} \tag{4.5}$$

where Θ is the ascension of the zero meridian, R_z and R_x are rotation matrixes.

4.2 Earth's magnetic field

To be able to control a satellite, with coils as actuators and magnetometer sensors, one needs a model of the Earths magnetic field. The complexity of the models can range from simple models to models which have a huge complexity.

4.2.1 Magnetic dipole

One of the simplest models represent the Earths magnetic field as a tilted magnetic dipole. The magnetic field is tilted with about 10°, as shown in figure (4.2). Because the magnetic field and the electric field is connected, it is natural to take a look at the electric dipole first.

Electric dipole

An electric field is produced by an electric charge that exerts a force on charged objects in its vicinity. In Sadiku (2001) an electric dipole is defined as:

Definition 1. *An electric dipole is formed when two points of charges of equal magnitude but opposite signs are separated by a small distance.*

The electric field generated by the dipole is:

$$\mathbf{E} = \frac{p}{4\pi\varepsilon_0 r^3}(2\cos\theta\mathbf{a}_r + \sin\theta\mathbf{a}_\theta) \tag{4.6}$$

Where:

- ε_0 is the permittivity of free space (F/m) $\approx \frac{10^{-9}}{36\pi}$
- r is the distance away from the center of the dipole
- θ is the angle from the dipole, to where the electric field is measured, se figure 4.1
- $\mathbf{a}_r$ and $\mathbf{a}_\theta$ are unit vectors
- $p = |\mathbf{p}|$=Qd where Q is the charge and d is the length of the dipole.

Further Sadiku (2001) defines the flux lines as :

Definition 2. *An electric flux line is an imaginary path or line drawn in such a way that its direction at any point is the direction of the electric field at that point.*

and the electric flux density is given by:

$$\mathbf{D} = \varepsilon_0\mathbf{E} \tag{4.7}$$

Magnetic dipole

A magnetic field is produced by a constant current flow (moving electric charges) that exerts a force on other moving charges. The connection between the magnetic field and the electric field is the moving charge Q, which is given by the lorentz force equation:

$$\boldsymbol{F} = Q(\boldsymbol{E} + \boldsymbol{u} \times \boldsymbol{B}) \tag{4.8}$$

where $\boldsymbol{u}$ is the velocity and Ampère-Maxwell's law:

$$\nabla \times \boldsymbol{B} = \mu_0\boldsymbol{J} + \mu_0\varepsilon_0\frac{\partial \boldsymbol{E}}{\partial t} \tag{4.9}$$

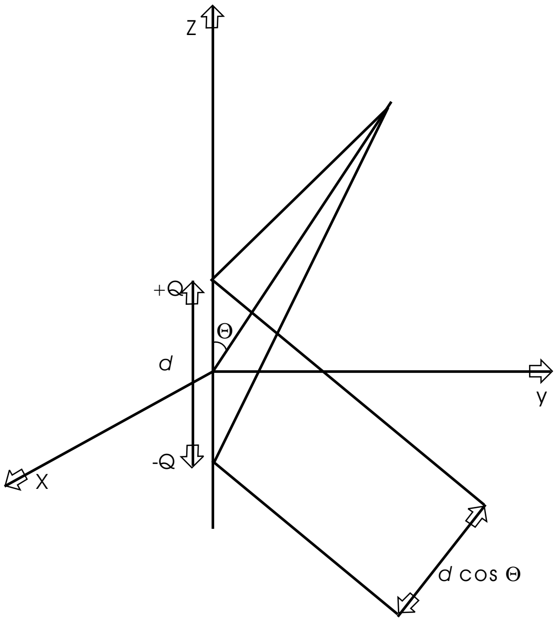


Figure 4.1: Explanation of the θ angle

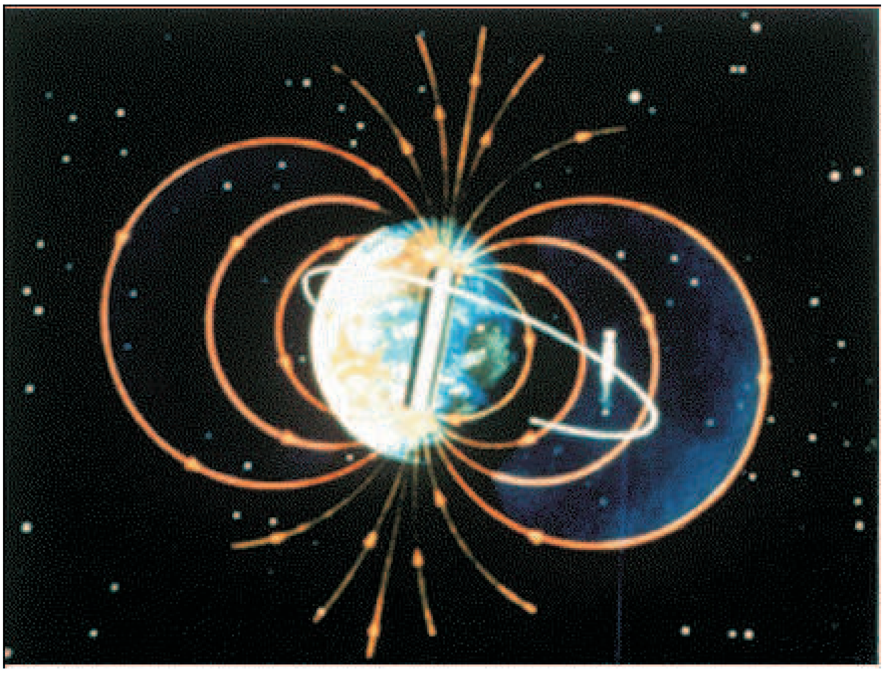


Figure 4.2: Earth represented as a tilted dipole.

The magnetic field due to a magnetic dipole is given by

$$\boldsymbol{B} = \frac{\mu_0 m}{4\pi r^3}(2\cos\theta \boldsymbol{a}_r + sin\theta \boldsymbol{a}_r) \tag{4.10}$$

where :

- μ_o is the permeability
- $\boldsymbol{J}$ is the current density
- ε_0 is the permittivity of free space (F/m) $\approx \frac{10^{-9}}{36\pi}$
- r is the distance away from the center of the dipole
- θ is the angle from the dipole, to where the magnetic field is measured, se figure 4.1
- $\mathbf{a}_r$ and $\mathbf{a}_\theta$ are unit vectors
- m is the mass of the particle.

4.2.2 International Geomagnetic Reference Field (IGRF)

The magnetic field of the Earth differs from the simplistic magnetic dipole. In fact the magnetic field is composite of several magnetic fields generated by a variety of sources, where the most important ones are the Earths fluid core, its crust and the upper mantel, the ionosphere and the magnetosphere. (www.esri.com 9.9.2004) To simulate this magnetic field The International Association of Geomagnetism and Aeronomy (IAGA) has provided the IGRF model. The IGRF model is a spherical harmonic equation, with Gauss coefficients,g_n^m, h_n^m

$$\boldsymbol{V} = a\sum_{n=1}^{N}\sum_{m=1}^{n}\left(\frac{a}{r}\right)^{n+1}(g_n^m \cos m\phi + h_n^m \cos m\phi)P_n^m(\cos\theta) \tag{4.11}$$

where V is the magnetic field given in the ECEF frame, a is the mean radius of the Earth, r is the distance form the center of the Earth, ϕ is the longitude east of Greenwich and θ is the colatitude (90°−latitude). The IGRF model is updated every 5 year. The newest IGRF model is IGRF2000, which have a order of 10. The IGRF2000 is illustrated in figure 4.3 and 4.4.

4.2.3 The magnetic field vector

To get the magnetic field from the ECEF to body frame it first has to be rotated to the Earth center orbit frame(Svartveit 2003):

$$\boldsymbol{B}^{oc} = \boldsymbol{R}_z(\omega)\boldsymbol{R}_x(i)\boldsymbol{R}_z(\Omega - \theta)\boldsymbol{V} \tag{4.12}$$

where Ω is the right ascension of the ascending node, ω is the Argument of perigee, i is the inclination and θ is the ascension of the zero meridian. In orbit frame the magnetic field can be calculated by :

$$\boldsymbol{B}^{o} = \boldsymbol{R}_x(\frac{\pi}{2})\boldsymbol{R}_z(\nu + \frac{\pi}{2})\boldsymbol{B}^{oc} \tag{4.13}$$

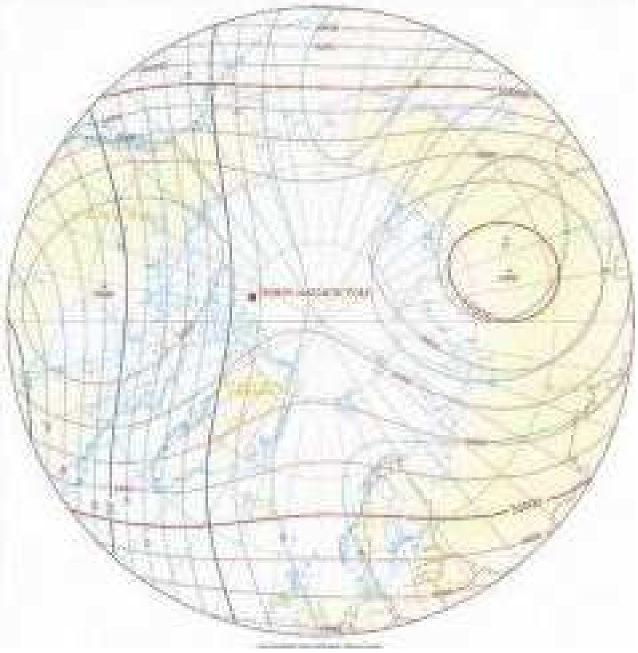


Figure 4.3: IGRF for the north pole

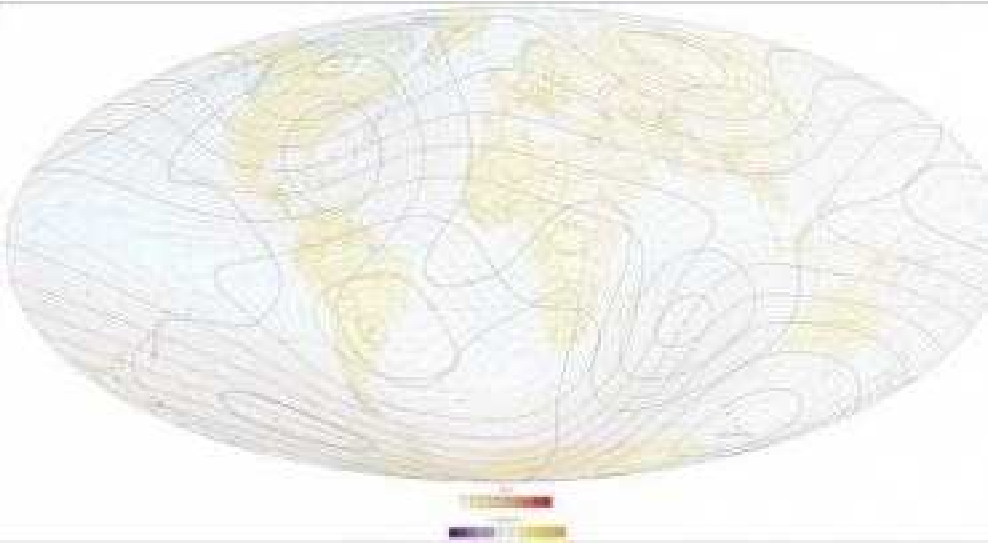


Figure 4.4: IGRF for the whole world

where ν is the true anomaly. It can further be calculated into the body frame of the satellite with :

$$\boldsymbol{B}^b = \boldsymbol{R}_o^b(\boldsymbol{q})\boldsymbol{B}^o \tag{4.14}$$

all of this can be done with one rotation matrix:

$$\boldsymbol{B}^b = \boldsymbol{R}_e^b\boldsymbol{V} \tag{4.15}$$

where

$$\boldsymbol{R}_e^b = \boldsymbol{R}_o^b(\boldsymbol{q})\boldsymbol{R}_x(\frac{\pi}{2})\boldsymbol{R}_z(\nu + \frac{\pi}{2})\boldsymbol{R}_z(\omega)\boldsymbol{R}_x(i)\boldsymbol{R}_z(\Omega - \theta) \tag{4.16}$$

4.3 Sun model

If the direction to the sun is known, it would provide a well defined reference vector, which can be utilized in the satellites attitude estimation. To be able to estimate the sun vector, the relationship between the Earth and the sun has to be known. It is well known that the Earth revolves around the sun, but it is more convenient to describe the relationship from the Earths point of view, as illustrated in figure 4.5. The elevation, ϵ_S, of the sun from the Earths equator varies with ±23°. Kristiansen (2000) proposed the following to calculate the elevation :

$$\epsilon_S = \frac{23\pi}{180}\sin\left(\frac{2\pi T_s}{365}\right) \tag{4.17}$$

where T_s is the time elapsed since the first day of spring. It has been assumed that orbit time is 365 days (a year) and that the Earth's orbit is circular. This leads to an error of (Svartveit 2003):

$$e = \arctan\left(\frac{R_o}{R_e}\right) \tag{4.18}$$

which is approximately $4.65 \cdot 10^{-5}$ [rad], where R_o is the radius of the satellite orbit, and R_e is the Earth orbit radius. The suns position, λ_s, in this imaginary orbit around the sun, is given by :

$$\lambda_S = \frac{2\pi T_s}{365} \tag{4.19}$$

4.3.1 The sun vector

When we know the elevation(ϵ_S given by (4.17)), and the suns poisson in relations to the Earth(λ_S given by (4.19)) it is possible to calculate a vector pointing ro the sun. The calculation, starts with the initial position

$$\boldsymbol{v}_{S0}^I = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{4.20}$$

given on the first day of spring (vernal equinox) (Svartveit 2003). Both the sun's elevation and the sun position describes the sun's imaginary rotation around the Earth. The position vector can be calculated as rotations:

$$\boldsymbol{v}_S^i = \boldsymbol{R}_y(\epsilon_S)\boldsymbol{R}_x(\lambda_S)\boldsymbol{v}_{S0}^i \tag{4.21}$$

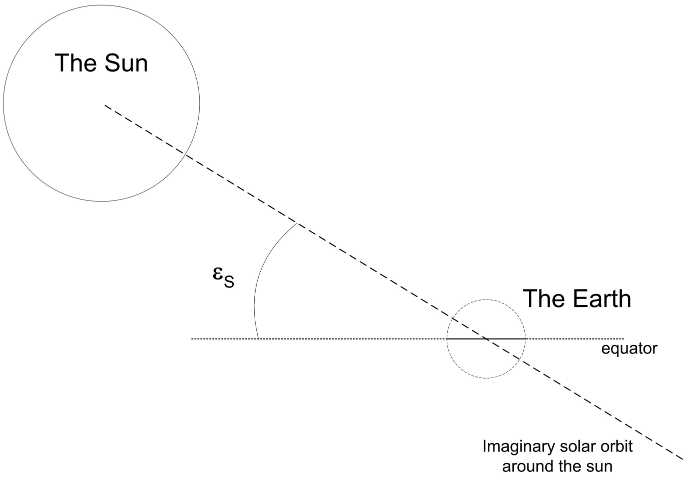


Figure 4.5: The suns elevation in an imaginary orbit around the Earth

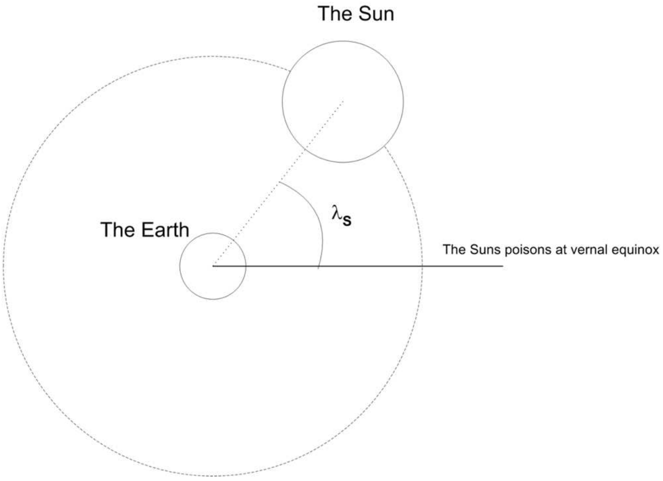


Figure 4.6: The suns position in an imaginary orbit around the Earth

which is equal to (substituting for the rotation matrixes, given by (2.11) and (2.12)):

$$\boldsymbol{v}_S^i = \begin{bmatrix} \cos\lambda_S \cos\epsilon_S \\ \sin\lambda_S \\ \cos\lambda_S \sin\epsilon_S \end{bmatrix} \tag{4.22}$$

and, by placing the $\cos\lambda_S$ on the outside yields:

$$\boldsymbol{v}_S^i = \begin{bmatrix} \cos\epsilon_S \\ \tan\lambda_S \\ \sin\epsilon_S \end{bmatrix} \cos\lambda_S \tag{4.23}$$

Since it is the vector direction that is important, the $\cos\lambda_S$ can be omitted, which gives:

$$\boldsymbol{v}_S^i = \begin{bmatrix} \cos\epsilon_S \\ \tan\lambda_S \\ \sin\epsilon_S \end{bmatrix} \tag{4.24}$$

The sun vector can easily be transformed into the orbit frame:

$$\boldsymbol{v}_S^o = \boldsymbol{R}_i^o(\boldsymbol{q})\boldsymbol{v}_S^i \tag{4.25}$$

and body frame:

$$\boldsymbol{v}_S^b = \boldsymbol{R}_o^b(\boldsymbol{q})\boldsymbol{v}_S^o \tag{4.26}$$

4.4 Star

Stars are self-luminous celestial bodies in space. All stars, except from the sun, are so far away[1] from the Earth that they appear as shining points fixed to specific locations in space. This makes it possible to utilize stars as fixed reference points in space.

4.5 The Earth albedo

The earth's reflection of the sun, the Earth albedo, is a significant source of light in a satellite's orbit. It would therefore contribute to deteriorate the measurement signals given from sun sensors or star tracers. To avoid this deterioration, the sensors either has to be shielded from the reflection or it has to be modeled and estimated.

4.6 Gravity torque

A satellite orbiting the earth will be subjected to the forces from the earth gravity field. If the mass of the satellite is distributed unevenly, the force from the earth gravity will poll unevenly on the satellite. This gived rise to the gravitational force. The gravitational force can, according to Kyrkjebø (2000), be modeled as :

$$\boldsymbol{\tau}_g^b = 3\omega_o^2 \boldsymbol{c}_3 \times \boldsymbol{I}\boldsymbol{c}_3 \tag{4.27}$$

[1] The star closest to the Earth is Proxima Centauri, 4.2 light years away

where $\boldsymbol{c}_3$ is directional cosine from the the rotational matrix $\boldsymbol{R}_o^b$ and ω_o is the angular velocity of the satellites orbit, and is calculated with:

$$\omega_o \approx \sqrt{\frac{GM}{R^3}} \tag{4.28}$$

where R is the orbit radius, G is Newton's specific gravity constant and M is the earth mass.

4.7 Ignored sources

In addition to the environmental condition listed earlier, there are some sources that, due to their small disturbance are ignored. These are the same as listed in Svartveit (2003), and are the following:

Atmospheric drag Due to the low amount of particles in the orbit, the atmospheric drag is assumed to be zero.

The gravity of the moon Because the gravity of the earth is dominant (it is closer and larger), the gravity of the moon, and the tidal force created by the earth-moon system is ignored.

Solar winds and pressure The sun's radiation of particles causes solar winds and pressure. Both the solar winds and pressure generates a torque on the satellite.

Satellite generated torques The satellite generated torques are generated by different sources, such as the deployment of the antenna and instruments onboard can generate magnetic fields that interact with the earth's magnetic field. These are either short lived, or have a low magnitude, and can therefore be ignored.

Chapter 5

Sensors

There are many types of sensors available for observing the attitude of a satellite. In this thesis the sensors used are two star trackers, a sun sensor and a magnetometer.

5.1 Star tracker

A star tracker is a lightsensitive precision instrument, that determines the attitude of a satellite by observing stars with high precision. The two major elements in a star tracker are a digital camera and processing unit.

The star tracker observes the stars with the camera, and compares this observations with an onboard star catalog. This gives the angles between the observed star an a reference frame in the satellite. If only one star (or clusters of stars) is observed, this will only give an accurate information about the attitude in 2 dimensions (Sellers 2000). By observing at least two different remote stars (or cluster of stars) the star tracker can determine the attitude in 3 dimensions(Chiang, Chang, Wang, Jan & Ting 2001).

How accurate the star tracker is, depends on the number of stars observed, the star catalog used, the quality of the optic and the resolution in the digital camera. A typically star sensor used in space application has a precision of a few arcseconds.

5.2 Sun sensor

A sun sensor is an instrument which measures the direction from the satellite to the sun. The direction to the sun can be measured in two different ways, both of them relying on photocells. The first one, the analog sun sensor, measures the intensity of the sun, and the second one, the digital sun sensor, uses a pattern where different photocells is exposed depending on the direction of the sun.

5.2.1 Analog sun sensor

As mentioned above the analog sun sensor, also called cosine sensor, measures the sun's intensity. More precisely it measures the energy flux through the surface area of photocell. The energy flux through a photocell is given by :

$$E = \boldsymbol{P} \cdot \boldsymbol{n} dA \tag{5.1}$$

where $\boldsymbol{P}$ is the pointing vector against the sun, $\boldsymbol{n}$ is a vector perpendicular to the cell, dA is the surface area of the cell and E is the energy flux. Equation (5.1) can be rewritten into :

$$E = \|\boldsymbol{P}\|\|\boldsymbol{n}\| \cos\Theta dA \tag{5.2}$$

where Θ is the angle of the pointing vector $\boldsymbol{P}$, and therefore is the angle from the perpendicular vector $\boldsymbol{n}$ towards the sun. The energy flux, E, is proportional to the electric current generated in the photocell. Therefore the easiest way to measure the angle of the pointing vector is to measure the current with :

$$I_c = I_{max} \cos\Theta \tag{5.3}$$

where I_c is the measured current, and I_{max} is the maximum current generated in the photocell.

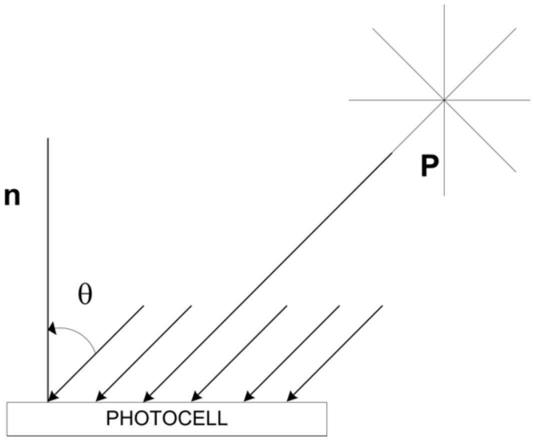


Figure 5.1: The analog photocell

5.2.2 Digital sun sensor

A digital sun sensor is built up of a pattern of photocells. The photocells are placed inside an installation that restricts which photocells that are illuminated and make this depended on the direction of the sun. This is illustrated in figure 5.2.

The photocells can be put into many different patterns. One example of this is a binary pattern, illustrated in fig 5.3, but it is more usual to use a Gray-coded pattern. From both of these patterns the angle of the sun can be found in the output of the sensors, since the photocells that is illuminated generate a higher energy level, either higher voltage or higher current, than the sensors that is in the shadow. This can in turn be converted into digital ones and zeros, which gives the angle of the sun either in binary or Gray scale.

5.2.3 The measured sun vector

The sun sensor measures the sun vector in body frame. Without noise the sun vector will be given by (4.26). With noise the measured sun vector can be given as

$$\boldsymbol{v}_S^b = \boldsymbol{R}_o^b \boldsymbol{v}_S^o + \boldsymbol{w} \tag{5.4}$$

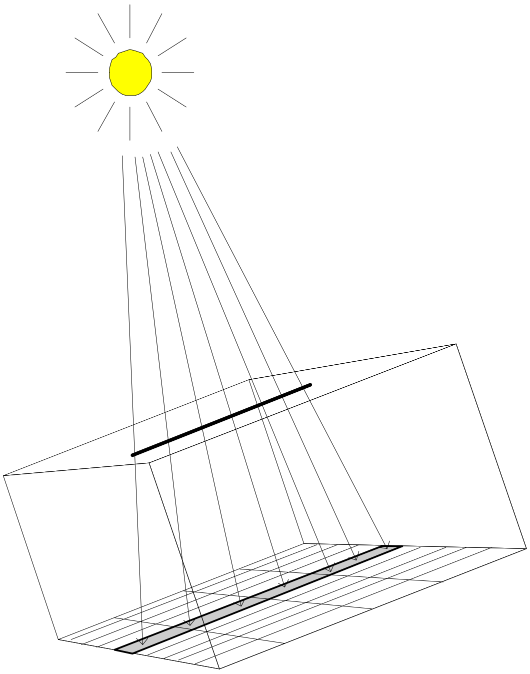


Figure 5.2: The digital photocell

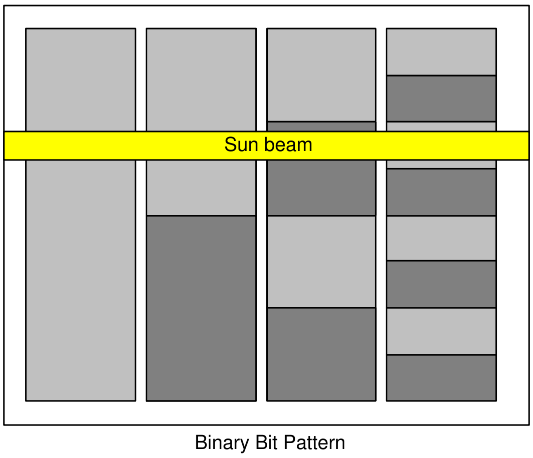


Figure 5.3: The binary bit pattern of a digital sun sensor

where $\boldsymbol{w}$ is an additive noise term, or it can be given by

$$\boldsymbol{v}_S^b = \boldsymbol{R}_o^b(\boldsymbol{q}_{w/noise})\boldsymbol{v}_S^o \tag{5.5}$$

where $\boldsymbol{q}_{w/noise}$ is the quaternion with noise. The rotation $\boldsymbol{R}_o^b(\boldsymbol{q}_{w/noise})$ can be replaced with $\boldsymbol{R}_o^b(\boldsymbol{\Theta}_{w/noise})$ if the attitude is given in Euler angels. Whether to use equation (5.4) or (5.5), depends on how the noise is given.

5.3 The magnetometer

A magnetometer is an instrument which measurers the flux density of the magnetic field it is placed in. A three axis magnetometer placed inside a satellite, will measures the geomagnetic intensity and direction surrounding the satellite. In a low orbiting satellite this can be used as a low cost, low weight, and reliable attitude sensors, with an accuracy of 0.5 to 3 degrees and a weight of 0.3 to 1.2 kg, according to (Wertz & Larson 1999). It can also be used to calculate the control input when using coils as actuator, and to estimate the orbit, as proposed by Zhao, Peng, Zeng, Shi, Huang & Li (2004)

The most common magnetometer used in space is the flux-gate magnetometer, where each axis has a sensor. Each sensor consists of a transformer wound around a core of high-permeability material, illustrated in figure 5.4 . By exciting the primary winding with a high frequency , this will induce a frequency on the secondary winding, where the amplitude and phase of the even harmonics are linearly proportional to the ambient magnetic field. For one axis this can give:

$$b \propto \hat{U}_{seco} \tag{5.6}$$

where $\hat{U}_{seco}$ is the voltage amplitude generated on the secondary winding. This can be rewritten into

$$b = K_{prop}\hat{U}_{seco} \tag{5.7}$$

where K_{prop} is the proportionality constant, and b is the ambient magnetic field. K_{prop} is determined by the material in the core, and the number of windings on the primary and secondary winding. When the three sensors are mounted perpendicular to each other, it will be possible to use (5.7), to calculate the magnetic field vector:

$$\boldsymbol{B}_{meas}^b = \begin{bmatrix} b_1^b \\ b_2^b \\ b_3^b \end{bmatrix} = \begin{bmatrix} K_{prop}\hat{U}_{seco1} \\ K_{prop}\hat{U}_{seco2} \\ K_{prop}\hat{U}_{seco3} \end{bmatrix} \tag{5.8}$$

where b_1^b is the ambient magnetic field of sensor nr 1 measured in body frame (it has been assumed that the sensor are aligned whit the body of the satellite), and $\hat{U}_{seco1}$ is the voltage amplitude on the secondary winding of the first sensor. Mathematically $\boldsymbol{B}_{meas}^b$ will, in a noise free environment, be equal to

$$\boldsymbol{B}_{meas}^b = \boldsymbol{R}_o^b(\boldsymbol{q})\boldsymbol{B}^o \tag{5.9}$$

where $\boldsymbol{B}^o$ is the magnetic field given in orbit frame. With noise $\boldsymbol{B}^o$ will be equal to

$$\boldsymbol{B}_{meas}^b = \boldsymbol{R}_o^b(\boldsymbol{q})\boldsymbol{B}^o + \boldsymbol{w} \tag{5.10}$$

where $\boldsymbol{w}$ is the noise vector.

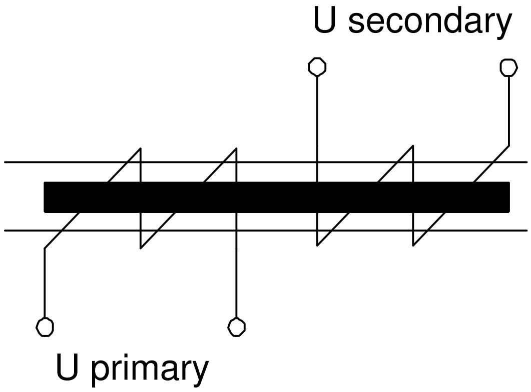


Figure 5.4: Flux-gate magnetometer

Chapter 6

Actuator

There are basically three categories of actuators available to control a satellite. These are thrusters, reaction wheels and magnetic coils.

6.1 Thruster

A thruster is basically a small rocket engine placed on the body of the satellite to adjust either the attitude or orbit parameters. The thruster is precise but it has to carry the fuel with itself when launched.

6.2 Reaction wheel

A reaction wheel is a rotor with high inertia that is accelerated. This acceleration will produce a torque on the reaction wheel:

$$\boldsymbol{\tau}_r = \boldsymbol{I}_r \dot{\boldsymbol{\omega}}_r \tag{6.1}$$

where $\boldsymbol{I}_r$ is the wheel's moment of inertia and $\dot{\boldsymbol{\omega}}_r$ is the angular acceleration. The torque on the wheel will generate a torque with opposite sign on the satellite (6.2), that can be used to control the angular velocity of the satellite.

$$\boldsymbol{\tau}_b = -\boldsymbol{\tau}_r = \boldsymbol{I}_b \dot{\boldsymbol{\omega}}_b \tag{6.2}$$

The torque is the same as the time derivative of the moment(h_r):

$$\boldsymbol{\tau}_r = \frac{d\boldsymbol{h}_r}{dt} \tag{6.3}$$

which, according to (Kaplan 1976), is equal to :

$$\boldsymbol{\tau}_r = \left(\frac{d\boldsymbol{h}_{\boldsymbol{r}}}{dt}\right)^b + \boldsymbol{\omega}_{ib}^b \times \boldsymbol{h}_r \tag{6.4}$$

By adding friction this becomes :

$$\boldsymbol{\tau}_r = \left(\frac{d\boldsymbol{h}_{\boldsymbol{r}}}{dt}\right)^b + \boldsymbol{\omega}_{ib}^b \times \boldsymbol{h}_r - \boldsymbol{\tau}_{friction}^b \tag{6.5}$$

6.3 The coils

To generate the force needed to rotate the satellite, one can use coils as actuators. A coil will generate a moment, and the moment will induce a torque which can be used to control the rotation of the satellite, and hence the attitude.

More precis, Sadiku (2001) defines the the dipole moment as:

Definition 3. *The magnetic dipole moment is the product of current and area of the loop; its direction is normal to the loop.*

The magnetic dipole moment for one rectangular planar loop in a uniform magnetic field:

$$\boldsymbol{m} = iA\mathbf{a}_n \tag{6.6}$$

and for n loops:

$$\boldsymbol{m} = niA\mathbf{a}_n \tag{6.7}$$

In a field with uniform magnetic field, the torque generated by the dipole will be :

$$\boldsymbol{\tau} = \boldsymbol{m} \times \boldsymbol{B} \tag{6.8}$$

This gives the following torque:

$$\boldsymbol{\tau} = niA\mathbf{a}_n \times \boldsymbol{m} \tag{6.9}$$

Equation (6.8) can be rewritten, using a skew symmetric matrix, to :

$$\boldsymbol{\tau} = \boldsymbol{S}(\boldsymbol{m})\boldsymbol{B} \tag{6.10}$$

and

$$\boldsymbol{\tau} = -\boldsymbol{S}(\boldsymbol{B})\boldsymbol{m} \tag{6.11}$$

By placing a coil in all three axis it is possible to control the satellite by controlling the electric current in the coils. It is important to notice that it is only possible to control the rotation of the satellite in an approximately homogenous magnetic field.

Chapter 7

Kalman filter

The Kalman filter was proposed by R. E. Kalman[1930-] for discrete systems in 1960[1] and for continuous system in 1961[2].A Kalman filter is a way of formulating the minimum mean-square error(MMSE) filtering problem. This makes the Kalman filter into a optimal state estimator, with minimum variance, making it possible to estimate the state of a dynamical system.

7.1 Discrete Kalman Filter

The discrete Kalman filter described in this section is based on a discrete and linear system given by:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{\Phi}\boldsymbol{x}_k + \boldsymbol{\Delta}\boldsymbol{u}_k + \boldsymbol{\Gamma}\boldsymbol{w}_k && (7.1) \\ \boldsymbol{y}_k &= \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{v}_k && (7.2) \end{aligned}$$

where $\boldsymbol{\Phi}$ is the discrete state transition matrix given by

$$\boldsymbol{\Phi} = e^{\boldsymbol{A}h} \tag{7.3}$$

$\boldsymbol{\Delta}$ is the discrete control transition matrix given by

$$\boldsymbol{\Delta} = \boldsymbol{A}^{-1}(\boldsymbol{\Phi} - \boldsymbol{I})\boldsymbol{B} \tag{7.4}$$

$\boldsymbol{\Gamma}$ is the discrete noise transition matrix given by

$$\boldsymbol{\Gamma} = \boldsymbol{A}^{-1}(\boldsymbol{\Phi} - \boldsymbol{I})\boldsymbol{E} \tag{7.5}$$

and h is the sampling time of the system given by $h = t_{k+1} - t_k$. Both the process noise $\boldsymbol{w}_k$ and the measurement noise $\boldsymbol{v}_k$ is assumed to be uncorrelated Gaussian with noise with covariance

$$\begin{aligned} \boldsymbol{Q} &= E(\boldsymbol{w}_k\boldsymbol{w}_k^T) && (7.6) \\ \boldsymbol{R} &= E(\boldsymbol{v}_k\boldsymbol{v}_k^T) && (7.7) \end{aligned}$$

[1] The paper "A New Approach to Linear Filtering and Prediction Problems" was published in ASME Journal of Basic Engineering, series D, 82

[2] The paper, coauthor with R.S Bucy, "A New Approach to Linear Filtering and Prediction Theory" was published in ASME Journal of Basic Engineering, series D, 83

The discrete Kalman filter can be divided into to filter equations (the estimation) and predictions (Henriksen 1998). The filter part consist of the state estimation

$$\hat{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_k + \boldsymbol{K}_k[\boldsymbol{y}_k - \boldsymbol{H}_k\bar{\boldsymbol{x}}_k] \tag{7.8}$$

the estimation of the Riccati equation

$$\hat{\boldsymbol{P}}_k = [\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k]\bar{\boldsymbol{P}}_k[\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k]^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^T \tag{7.9}$$

and the calculation of the Kalman gain

$$\boldsymbol{K}_k = \bar{\boldsymbol{P}}_k\boldsymbol{H}_k^T[\boldsymbol{H}\bar{\boldsymbol{P}}_k\boldsymbol{H}_k^T + \boldsymbol{R}_k]^{-1} \tag{7.10}$$

The prediction part consist of the state prediction

$$\bar{\boldsymbol{x}}_{k+1} = \boldsymbol{\Phi}\hat{\boldsymbol{x}}_k + \boldsymbol{\Delta}\boldsymbol{u}_k \tag{7.11}$$

and the prediction of the Riccati equation

$$\bar{\boldsymbol{P}}_{k+1} = \boldsymbol{\Phi}_k\hat{\boldsymbol{P}}_k\boldsymbol{\Phi}_k^T + \boldsymbol{\Gamma}_k\boldsymbol{Q}_k\boldsymbol{\Gamma}_k^T \tag{7.12}$$

7.1.1 Discrete Extended Kalman filter

For a nonlinear discrete system the ordinary discrete Kalman filter will not be sufficient. The system equation for the nonlinear discrete system is given by:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{g}_k(\boldsymbol{w}_k) \tag{7.13}$$

$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{7.14}$$

where $\boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k)$, $\boldsymbol{g}_k(\boldsymbol{x}_k)$ and $\boldsymbol{h}_k(\boldsymbol{x}_k)$ are nonlinear functions. The filter equations for the discrete Extended Kalman filter(DEKF) is given by the state estimation:

$$\hat{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_k + \boldsymbol{K}_k[\boldsymbol{y}_k - \boldsymbol{h}_k(\bar{\boldsymbol{x}}_k)] \tag{7.15}$$

the estimation of the Riccati equation

$$\hat{\boldsymbol{P}}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\bar{\boldsymbol{P}} \tag{7.16}$$

and the Kalman gain is given by

$$\boldsymbol{K}_k = \bar{\boldsymbol{P}}_k\boldsymbol{H}_k^T(\boldsymbol{H}_k\bar{\boldsymbol{P}}_k\boldsymbol{H}_k^T + \boldsymbol{R})^{-1} \tag{7.17}$$

where $\boldsymbol{H}$ is the linearized discrete measurement matrix, given by:

$$\boldsymbol{H}_k = \frac{\partial \boldsymbol{h}_k}{\partial \boldsymbol{x}_k^T}(\bar{\boldsymbol{x}}_k) \tag{7.18}$$

The predictions are given by state prediction

$$\bar{\boldsymbol{x}}_{k+1} = \boldsymbol{f}_k(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k) \tag{7.19}$$

and the prediction or the Riccati equation

$$\bar{\boldsymbol{P}}_{k+1} = \boldsymbol{\Phi}_k\hat{\boldsymbol{P}}_k\boldsymbol{\Phi}_k^T + \boldsymbol{\Gamma}_k\boldsymbol{Q}_k\boldsymbol{\Gamma}_k^T \tag{7.20}$$

where $\boldsymbol{\Phi}_k$ is the linearized discrete state transfer matrix, given by :

$$\boldsymbol{\Phi}_k = \frac{\partial \boldsymbol{f}_k}{\partial \boldsymbol{x}_k^T}(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k) \tag{7.21}$$

7.2 Continuous Kalman Filter

The continuous Kalman filter is based on a continuous and linear system. The linear equations are given by :

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} + \boldsymbol{Ew} \tag{7.22}$$

$$\boldsymbol{y} = \boldsymbol{Hx} + \boldsymbol{v} \tag{7.23}$$

where $\boldsymbol{A}$ is the state transition matrix, $\boldsymbol{B}$ is the measurement transition $\boldsymbol{E}$ is the noise transition matrix and $\boldsymbol{H}$ is the measurement matrix. $\boldsymbol{w}$ and $\boldsymbol{v}$ are assumed to be uncorrelated Gaussian with noise process with covariance

$$\boldsymbol{R} = E(\boldsymbol{vv}^T) \tag{7.24}$$

$$\boldsymbol{Q} = E(\boldsymbol{ww}^T) \tag{7.25}$$

The filter equation consist of the state estimation :

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{Bu} + \boldsymbol{K\nu} \tag{7.26}$$

where $\boldsymbol{\nu}$ is the innovation process given by :

$$\boldsymbol{\nu} = \boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}} \tag{7.27}$$

The Kalman gain is calculated from

$$\boldsymbol{K} = \boldsymbol{PH^TR^{-1}} \tag{7.28}$$

The continuous Riccati equation is given by

$$\dot{\boldsymbol{P}} = \boldsymbol{AP} + \boldsymbol{PA}^T + \boldsymbol{EQE^T} - \boldsymbol{PH^TR^{-1}HP} \tag{7.29}$$

which means that $\boldsymbol{P} = \int \dot{\boldsymbol{P}}$

7.2.1 Continuous Extended Kalman filter

For a nonlinear system the ordinary continuous Kalman filter will not be sufficient. Instead the continuous Extended Kalman(CEKF) filter will be used. The nonlinear continuous system is given by:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{w} \tag{7.30}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v} \tag{7.31}$$

where $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$, $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{h}(\boldsymbol{x})$ are nonlinear functions. Filter equation for the CEKF consist of the state estimation :

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u}) + \boldsymbol{K\nu} \tag{7.32}$$

where the innovation is given by

$$\boldsymbol{\nu} = \boldsymbol{y} - \boldsymbol{h}(\hat{\boldsymbol{x}}) \tag{7.33}$$

and the Kalman gain is calculated with

$$\boldsymbol{K} = \boldsymbol{P}\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{R}^{-\boldsymbol{1}} \tag{7.34}$$

The continuous Riccati equation is given by

$$\dot{\boldsymbol{P}} = \boldsymbol{A}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^{T} + \boldsymbol{E}\boldsymbol{Q}\boldsymbol{E}^{\boldsymbol{T}} - \boldsymbol{P}\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{R}^{-\boldsymbol{1}}\boldsymbol{H}\boldsymbol{P} \tag{7.35}$$

which means that $\boldsymbol{P} = \int \dot{\boldsymbol{P}}$. A is the linearized state transfer matrix, given by :

$$\boldsymbol{A} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^{T}}(\hat{\boldsymbol{x}}, \boldsymbol{u}) \tag{7.36}$$

and H is the linearized measurement matrix, given by:

$$\boldsymbol{H} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}^{T}}(\hat{\boldsymbol{x}}) \tag{7.37}$$

7.3 Unit Quaternions in Kalman Filter

The definition of the quaternion (2.3) gives rise to some difficulties when they are used in a Kalman filter. This difficulties was addressed by Lefferts, Markley & Shuster (1982), and most of this section will be based on this article.

7.3.1 Covariance singularity

The first problem is the fact that the unit quaternion parameter is dependent on each other, this is because the norm constrained to 1 ($\|\boldsymbol{q}\| = 1$). This constraint result in a singularity in the covariance matrix of the quaternion. According to Lefferts et al. (1982)the norm constrain gives :

$$\triangle q^{T}\hat{q} \simeq 0 \tag{7.38}$$

where $\hat{q}$ is the estimated of q and $\triangle q = q - \hat{q}$. This then leads to the following null vector of P:

$$\begin{bmatrix} \hat{q} \\ 0 \end{bmatrix} \tag{7.39}$$

This singularity is difficult to maintain numerically, and is therefore a source of errors in computer computations. Lefferts et al. (1982) proposes three ways to avoid this numerically errors, all of which are based on reducing the covariance matrix from a 7 × 7 to a 6 × 6 matrix. The three different ways are the Truncated Covariance Representation, Reduced Representation of the Covariance Matrix and the body-Fixed Covariance Representation. This thesis will use the Truncated Covariance Representation which therefore will be described below. For the other two reading the article is recommend.

The Truncated Covariance Representation. The idea with the truncated covariance representation, is to truncate the quaternion vector used in the calculation of the variance matrix. The most obvious state to remove is the η element, but in principal, any component is as good.

7.3.2 Maintaining the unity of the quaternion

The second problem is maintaining the unity of the unit quaternion. In theory a unit quaternion should remain as a unit quaternion. As the following shows(Egeland & Gravdahl 2002) :

$$\frac{d}{dt}(\boldsymbol{q}^T\boldsymbol{q}) = \boldsymbol{q}^T \begin{bmatrix} \eta & -\boldsymbol{\epsilon}^T \\ \boldsymbol{\epsilon} & \eta\boldsymbol{I}_{3\times3} + \boldsymbol{S}(\boldsymbol{\epsilon}) \end{bmatrix} \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} = 0 \tag{7.40}$$

where the norm of a unit quaternion will stay identical. But this property is hard to maintain in numerical calculation. It is also difficult to maintain in an ordinary Kalman filter, both discrete and continuous. There are many ways to remedy this problem. The most intuitive way is to normalize the output directly:

$$\boldsymbol{q} = \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|} \tag{7.41}$$

An other way, described by Egeland & Gravdahl (2002), is to insert a normalizing term

$$\frac{\lambda}{2}(1 - \boldsymbol{q}^T\boldsymbol{q})\boldsymbol{q}^T\boldsymbol{q} \tag{7.42}$$

into the kinematic equation (3.11). This will give :

$$\dot{\boldsymbol{q}} = \frac{1}{2}\begin{bmatrix} -\boldsymbol{\varepsilon}^T \\ \eta\boldsymbol{I}_{3\times3} + \boldsymbol{S}(\boldsymbol{\varepsilon}) \end{bmatrix} \boldsymbol{\omega}_{ob}^b + \frac{\lambda}{2}(1 - \boldsymbol{q}^T\boldsymbol{q})\boldsymbol{q}^T\boldsymbol{q} \tag{7.43}$$

where λ is a positive integer, which controls how fast the normalizing will converge[3].

7.4 Star sensor in Kalman filter

The star sensor gives out measurements in either Euler parameters or in quaternions. This means that it is easy to incorporate in to the Kalman filter. Since the Kalman filter in this thesis uses quaternions, it is assumed that sensor data from the star sensor is given in quaternions. The innovation process will be

$$\boldsymbol{\nu} = \boldsymbol{q}_{meas} - \hat{\boldsymbol{q}} \tag{7.44}$$

where $\boldsymbol{q}_{meas}$ is the measured quaternion vector and $\hat{\boldsymbol{q}}$ is the estimated quaternion vector. Since the star sensor measure the quaternion directly, the measurement matrix is straight forward. For a truncated system, without η the measurement matrix will be :

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \boldsymbol{0} \end{bmatrix} \tag{7.45}$$

and the innovation process will reduced to

$$\boldsymbol{\nu}_{\varepsilon} = \boldsymbol{\varepsilon}_{meas} - \hat{\boldsymbol{\varepsilon}} \tag{7.46}$$

[3] $\lambda = 100$ will give a convergence time of 0.01 sec.

7.5 Magnetometer in Kalman filter

The measurements from the magnetometer($\boldsymbol{B}^b_{meas}$) is given by (5.10). This measurement can be compared with the predicted magnetic field

$$\bar{\boldsymbol{B}}^b = \boldsymbol{R}^b_o(\hat{\boldsymbol{q}})\boldsymbol{B}^o \tag{7.47}$$

where $\hat{\boldsymbol{q}}$ is the estimated quaternion.

The standard innovation process would be:

$$\boldsymbol{\nu} = \boldsymbol{B}^b_{meas} - \bar{\boldsymbol{B}}^b \tag{7.48}$$

but since it is a pointing vector, and the interesting difference is in the vector part, it is more practical to normalize both the measured and the estimated signal, which gives the following innovation

$$\boldsymbol{\nu} = \frac{\boldsymbol{B}^b_{meas}}{\|\boldsymbol{B}^b_{meas}\|} - \frac{\bar{\boldsymbol{B}}^b}{\|\bar{\boldsymbol{B}}^b\|} \tag{7.49}$$

which according to Kyrkjebø (2000) ensures a more proper filter behavior, regardless of the unit of the magnetic field.

Since the prediction of the magnetic field (7.47) is nonlinear, the measurement matrix(H) must be linearized according to (7.37). By linearizing it around $\hat{\boldsymbol{q}}$, as proposed by Steyn (1994), the measurement matrix for a truncated system without η , will become:

$$H = \begin{bmatrix} \frac{\partial \boldsymbol{R}^b_o(\bar{q})}{\partial \varepsilon_1}\boldsymbol{B}_{orbit} & \frac{\partial \boldsymbol{R}^b_o(\bar{q})}{\partial \varepsilon_2}\boldsymbol{B}_{orbit} & \frac{\partial \boldsymbol{R}^b_o(\bar{q})}{\partial \varepsilon_3}\boldsymbol{B}_{orbit} & \boldsymbol{0} \end{bmatrix} \tag{7.50}$$

An easier measurement matrix proposed by Bak (1999) is :

$$H = \begin{bmatrix} 2\boldsymbol{S}(\bar{\boldsymbol{B}}) & \boldsymbol{0} \end{bmatrix} \tag{7.51}$$

This is also based on linearizing equation (7.47) around $\hat{\boldsymbol{q}}$, but to get this result one has to assume that η is close to 1 and $\boldsymbol{\epsilon}$ is close to $\boldsymbol{0}$. It is worth noticing that $\boldsymbol{H}$ given by (7.51) only has the rank of 2, which indicates that at any particular time it will only be possible to estimate 2 directions with the magnetometer.

7.6 Sun sensor in Kalman filter

Because both the sun sensor and magnetometer are pointing vectors they can be treated similarly. Doing the same for the sun sensor as for the magnetometer will give the following prediction of the sun vector:

$$\hat{\boldsymbol{v}}^b_S = \boldsymbol{R}^b_o(\hat{\boldsymbol{q}})\boldsymbol{v}^o_S \tag{7.52}$$

where $\boldsymbol{v}^b_{meas}$ is the sun vector measured by the sun sensors. The sun sensor can use the same form of innovation process as the magnetometer, where both the measurement and the estimated signals are normalized:

$$\boldsymbol{\nu} = \frac{\boldsymbol{v}^b_{meas}}{\|\boldsymbol{v}^b_{meas}\|} - \frac{\bar{\boldsymbol{v}}^b_S}{\|\bar{\boldsymbol{v}}^b_S\|} \tag{7.53}$$

This yield the following measurement matrixes for a truncated system without η

$$H = \left[\begin{array}{cc} 2\boldsymbol{S}(\bar{\boldsymbol{v}}_S^b) & \mathbf{0} \end{array} \right] \tag{7.54}$$

$$H = \left[\begin{array}{cccc} \frac{\partial \boldsymbol{R}_o^b(\bar{q})}{\partial \varepsilon_1}\boldsymbol{v}_S^o & \frac{\partial \boldsymbol{R}_o^b(\bar{q})}{\partial \varepsilon_2}\boldsymbol{v}_S^o & \frac{\partial \boldsymbol{R}_o^b(\bar{q})}{\partial \varepsilon_3}\boldsymbol{v}_S^o & \mathbf{0} \end{array} \right] \tag{7.55}$$

where $\boldsymbol{v}_S^o$ is given by (4.25). It is also worth noticing that $\boldsymbol{H}$ given by (7.54) only has the rank 2, which indicates that it only will be possible to estimate 2 directions with the sun at any particular time.

7.7 The Gauss-Newton Method

Instead of using the vector measurements directly in the Kalman filter, one can make use of the Gauss-Newton method described by Marins, Yun, Bachmann, McGhee & Zyda (2001). This method finds the quaternion best relating to the measured vectors by minimizing a nonlinear objective function given by:

$$\begin{aligned} \boldsymbol{Q} &= \boldsymbol{\epsilon}^T\boldsymbol{\epsilon} && (7.56) \\ \boldsymbol{\epsilon} &= \boldsymbol{y}^o - \boldsymbol{M}(\hat{\boldsymbol{q}}^{GN})y_{meas}^b && (7.57) \end{aligned}$$

where $\boldsymbol{\epsilon}$ is the vector error given in body frame, $\boldsymbol{y}^o$ the reference vector given in orbit frame, $\boldsymbol{y}_{meas}^b$ is the measured vector field given in body frame, and $M(\hat{\boldsymbol{q}}^{GN})$ is a matrix that transforms the measured vector from the body frame to the orbit frame. Since the measurement is given in the body frame and the error is given in orbit frame $M(\hat{\boldsymbol{q}}^{GN})$ will be given by:

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{R}_b^o(\hat{\boldsymbol{q}}^{GN}) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{R}_b^o(\hat{\boldsymbol{q}}^{GN}) \end{bmatrix} \tag{7.58}$$

where $\boldsymbol{R}_b^o(\hat{\boldsymbol{q}}^{GN})$ is given by (2.15), and $\hat{\boldsymbol{q}}^{GN}$ is the quaternion calculated by the Gauss-Newton equation. Since the the nonlinear objective function($\boldsymbol{Q}$) depends on the estimated quaternion vector $\hat{\boldsymbol{q}}^{GN}$, the Gauss-Newton method will yield the quaternion with the least minimum error, corresponding to the vector measurements.

The iterative Gauss-Newton method given by Marins et al. (2001) is :

$$\hat{\boldsymbol{q}}_{k+1}^{GN} = \hat{\boldsymbol{q}}_k^{GN} - [\boldsymbol{J}^T(\hat{\boldsymbol{q}}_k^{GN})\boldsymbol{J}(\hat{\boldsymbol{q}}_k^{GN})]^{-1}\boldsymbol{J}^T(\hat{\boldsymbol{q}}_k^{GN})\boldsymbol{\epsilon}(\hat{\boldsymbol{q}}_k^{GN}) \tag{7.59}$$

where $\boldsymbol{J}$ is the Jacobian matrix defined as:

$$\boldsymbol{J} = -\left[\begin{array}{cccc} \left(\frac{\delta \boldsymbol{M}}{\delta \hat{\eta}^{GN}}\boldsymbol{y}_0\right) & \left(\frac{\delta \boldsymbol{M}}{\delta \hat{\epsilon_1}^{GN}}\boldsymbol{y}^0\right) & \left(\frac{\delta \boldsymbol{M}}{\delta \hat{\epsilon_2}^{GN}}\boldsymbol{y}^0\right) & \left(\frac{\delta \boldsymbol{M}}{\delta \hat{\epsilon_3}^{GN}}\boldsymbol{y}^0\right) \end{array} \right] \tag{7.60}$$

The continuous form of the Gauss-Newton method can be deduced from the iterative method(A), and is given by :

$$\dot{\hat{\boldsymbol{q}}}^{GN} = -[\boldsymbol{J}^T(\hat{\boldsymbol{q}}^{GN})\boldsymbol{J}(\hat{\boldsymbol{q}}^{GN})]^{-1}\boldsymbol{J}^T(\hat{\boldsymbol{q}}^{GN})\boldsymbol{\epsilon}(\hat{\boldsymbol{q}}^{GN}) \tag{7.61}$$

The matrix , $\boldsymbol{J}^T(\hat{\boldsymbol{q}}^{GN})\boldsymbol{J}(\hat{\boldsymbol{q}}^{GN})$, will be badly conditioned if only one measurement is available (or a set of measurements with the same directional information). This will make it problematic to take the inverse of the matrix. This can be avoided by taking the pseudoinverse. It will also result in problems with the estimation in one direction.

Using the Gauss-Newton method to estimate a quaternion from the vector and using this estimation in the Kalman filter, will make the measurement equation of the Kalman filter linear. This reduces the computational requirements. For the Gauss-Newton method to be used in the Kalman filter it has to converge rapidly. The iterative Gauss-Newton method was shown by Marins et al. (2001) to converge in 3-4 iterations, even when the initial estimate had a large error. Marins et al. (2001) do not show that the Gauss-Newton method converge theoretically, but this can be found textbooks about numerical optimization technics, such as Nocedal & Wright (1999).

7.8 Continuous Extended Kalman Filter for a Satellite

The Kalman filter for a Satellite starts with the dynamic and kinematic equations of the satellite, given in chapter 3 equation (3.7) and (3.11). The Kalman equation copies the dynamic and kinematic equations and adds a correction term. This will give :

$$\dot{\hat{\boldsymbol{q}}} = \frac{1}{2}\begin{bmatrix} -\hat{\boldsymbol{\varepsilon}}^T \\ \eta \boldsymbol{I}_{3\times 3} + \boldsymbol{S}(\hat{\boldsymbol{\varepsilon}}) \end{bmatrix} \hat{\boldsymbol{\omega}}^b_{ob} + \boldsymbol{K_q \nu} \tag{7.62}$$

$$\dot{\hat{\boldsymbol{\omega}}}^b_{ib} = \boldsymbol{I}^{-1}[\boldsymbol{\tau}^b - \boldsymbol{S}(\hat{\boldsymbol{\omega}}^b_{ib})\boldsymbol{I}\hat{\boldsymbol{\omega}}^b_{ib})] + \boldsymbol{K_\omega \nu} \tag{7.63}$$

where $\hat{\boldsymbol{\omega}}^b_{ob}$ can be calculated with :

$$\hat{\boldsymbol{\omega}}^b_{ob} = \hat{\boldsymbol{\omega}}^b_{ib} - R^b_o(\hat{\boldsymbol{q}})\boldsymbol{\omega}^o_{ib} \tag{7.64}$$

or by integrating :

$$\dot{\hat{\boldsymbol{\omega}}}^b_{ob} = \dot{\hat{\boldsymbol{\omega}}}^b_{ib} + \boldsymbol{S}(\hat{\boldsymbol{\omega}}^b_{ob})\boldsymbol{R}^b_o\boldsymbol{\omega}^o_{io} \tag{7.65}$$

$K_{\boldsymbol{q}}$ and $K_{\boldsymbol{\omega}}$ are Kalman gains for the kinematic equation and for the dynamic and $\boldsymbol{\nu}$ is the innovation process.

The calculation of the Kalman gain $\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K_q} & \boldsymbol{K_\omega} \end{bmatrix}^T$, calculated with equation (7.34), and the innovations depend on which measurements used. The measurement matrix and the innovation used will be given in each case. Since the quaternion have to maintain the unity (described in section 7.3.2) it will not work to only integrate $\dot{\hat{\boldsymbol{q}}}$, it has to be normalized. The easiest will be to integrate equation (7.63) and normalize it like this:

$$\hat{\boldsymbol{q}} = \frac{\int \dot{\hat{\boldsymbol{q}}}}{\| \int \dot{\hat{\boldsymbol{q}}} \|} \tag{7.66}$$

But doing only this, will give no control with the value of $\| \int \dot{\hat{\boldsymbol{q}}} \|$, which may be a source of error. A way to avoid this is to include the normalizing term given in (7.42) in to the dynamic equation before integrating and normalizing it. This will lead to :

$$\dot{\bar{\boldsymbol{q}}} = \frac{1}{2}\begin{bmatrix} -\hat{\boldsymbol{\varepsilon}}^T \\ \eta \boldsymbol{I}_{3\times 3} + \boldsymbol{S}(\hat{\boldsymbol{\varepsilon}}) \end{bmatrix} \hat{\boldsymbol{\omega}}^b_{ob} + \frac{\lambda}{2}(1 - \bar{\boldsymbol{q}}^T\bar{\boldsymbol{q}})\bar{\boldsymbol{q}}^T\bar{\boldsymbol{q}} + \boldsymbol{K_q \nu} \tag{7.67}$$

$$\bar{\boldsymbol{q}} = \int \dot{\bar{\boldsymbol{q}}} \tag{7.68}$$

$$\hat{\boldsymbol{q}} = \frac{\bar{\boldsymbol{q}}}{\|\bar{\boldsymbol{q}}\|} \tag{7.69}$$

Since the dynamic equations depends on $\boldsymbol{\omega}_{ob}^{b}$ instead of $\boldsymbol{\omega}_{ib}^{b}$ the equation (3.10) can be used directly in the Kalman filter, this gives the following estimation of the angular velocity :

$$\dot{\hat{\boldsymbol{\omega}}}_{ob}^{b} = \dot{\hat{\boldsymbol{\omega}}}_{ib}^{b} + \boldsymbol{S}(\hat{\boldsymbol{\omega}}_{ob}^{b})\boldsymbol{R}_{o}^{b}\boldsymbol{\omega}_{io}^{o} + \boldsymbol{K}_{\boldsymbol{\omega}}\boldsymbol{\nu} \tag{7.70}$$

This gives the following Kalman estimation for the satellite for a full state estimation:

$$\begin{aligned}
\dot{\bar{\boldsymbol{q}}} &= \frac{1}{2}\begin{bmatrix} -\hat{\boldsymbol{\varepsilon}}^{T} \\ \eta \boldsymbol{I}_{3\times 3} + \boldsymbol{S}(\hat{\boldsymbol{\varepsilon}}) \end{bmatrix} \hat{\boldsymbol{\omega}}_{ob}^{b} + \frac{\lambda}{2}(1 - \bar{\boldsymbol{q}}^{T}\bar{\boldsymbol{q}})\bar{\boldsymbol{q}}^{T}\bar{\boldsymbol{q}} + \boldsymbol{K}_{\boldsymbol{q}}\boldsymbol{\nu} && (7.71)\\
\bar{\boldsymbol{q}} &= \int \dot{\bar{\boldsymbol{q}}} && (7.72)\\
\hat{\boldsymbol{q}} &= \frac{\bar{\boldsymbol{q}}}{\|\bar{\boldsymbol{q}}\|} && (7.73)\\
\dot{\hat{\boldsymbol{\omega}}}_{ib}^{b} &= \boldsymbol{I}^{-1}[\boldsymbol{\tau}^{b} - \boldsymbol{S}(\hat{\boldsymbol{\omega}}_{ib}^{b})\boldsymbol{I}\hat{\boldsymbol{\omega}}_{ib}^{b})] + \boldsymbol{K}_{\boldsymbol{\omega}}\boldsymbol{\nu} && (7.74)\\
\dot{\hat{\boldsymbol{\omega}}}_{ob}^{b} &= \dot{\hat{\boldsymbol{\omega}}}_{ib}^{b} + \boldsymbol{S}(\hat{\boldsymbol{\omega}}_{ob}^{b})\boldsymbol{R}_{o}^{b}\boldsymbol{\omega}_{io}^{o} && (7.75)
\end{aligned}$$

When the quaternion in the state vector are truncated to avoid the covariance singularities (7.3.1) the dimension of the Riccati equation will result in a reduction of the Kalman gain where $K_{\boldsymbol{q}}$ will be reduced from a $4\times n$ to a $3\times n$ matrix, where n depends on the number of measurements used in the Kalman filter. The dimension of the innovation $\boldsymbol{\nu}$ will also be reduced . If the truncation removes the η from the state vector the Kalman filter will become :

$$\begin{aligned}
\dot{\bar{\boldsymbol{q}}} &= \frac{1}{2}\begin{bmatrix} -\hat{\boldsymbol{\varepsilon}}^{T} \\ \eta \boldsymbol{I}_{3\times 3} + \boldsymbol{S}(\hat{\boldsymbol{\varepsilon}}) \end{bmatrix} \hat{\boldsymbol{\omega}}_{ob}^{b} + \frac{\lambda}{2}(1 - \bar{\boldsymbol{q}}^{T}\bar{\boldsymbol{q}})\bar{\boldsymbol{q}}^{T}\bar{\boldsymbol{q}} + \begin{bmatrix} 0 \\ \boldsymbol{K}_{\boldsymbol{q}}\boldsymbol{\nu} \end{bmatrix} && (7.76)\\
\bar{\boldsymbol{q}} &= \int \dot{\bar{\boldsymbol{q}}} && (7.77)\\
\hat{\boldsymbol{q}} &= \frac{\bar{\boldsymbol{q}}}{\|\bar{\boldsymbol{q}}\|} && (7.78)\\
\dot{\hat{\boldsymbol{\omega}}}_{ib}^{b} &= \boldsymbol{I}^{-1}[\boldsymbol{\tau}^{b} - \boldsymbol{S}(\hat{\boldsymbol{\omega}}_{ib}^{b})\boldsymbol{I}\hat{\boldsymbol{\omega}}_{ib}^{b})] + \boldsymbol{K}_{\boldsymbol{\omega}}\boldsymbol{\nu} && (7.79)\\
\dot{\hat{\boldsymbol{\omega}}}_{ob}^{b} &= \dot{\hat{\boldsymbol{\omega}}}_{ib}^{b} + \boldsymbol{S}(\hat{\boldsymbol{\omega}}_{ob}^{b})\boldsymbol{R}_{o}^{b}\boldsymbol{\omega}_{io}^{o} && (7.80)
\end{aligned}$$

Chapter 8

Implementation and Simulation results

There are many ways of combining magnetometer, sun-sensor and star tracker data in a Kalman filter. In this chapter different Kalman filters, using one or more of this sensors, will be presented and simulated.

8.1 System overview and general assumptions

Before the different filters are presented, a short system overview and the assumptions made for all the different filters will be given. The most important assumption is that the satellite has been detumbled before the attitude estimation begins. This means that the initial angular velocity is limited. It has also been assumed that all the sensors works independent of each other, and that none of them have problems coping with the angular velocity of the satellite after it has been detumbled.

The system overview will use the simulink diagram of the system as basis. The top level of the simulink diagram is shown in figure B.1 and consist of the following sections:

Environmental Model The Environmental model calculates the environment around the satellite. It calculates the gravity torque on the system, the magnetic field surrounding the satellite and the suns position.

Controller The Controller block contains the system controller. The controller used is a purely mathematical PD controller, with out physical limitations. It is only used to keep the system at rest during simulations, or to check how the filter and system behaves with feedback from the estimated stats.

Sensors The sensors block contains the sensor models and calculates the sensor output.

Satellite Nonlinear Dynamics The Satellite Nonlinear Dynamics block contains the satellites system equation, and calculates the real attitude and angular velocity of the satellite.

Kalman filter The Kalman filter block contains the different Kalman filters used during simulation. It contain the Gauss-Newton algorithm, calculating an attitude estimation from the vector measurements, the innovation section, the Kalman update section which calculates the Kalman gain, and the dynamic and Kinematic equations used in the Kalman filter.

8.1.1 Implementing the satellite model.

The satellite model implements the dynamic and kinematic equation given i chapter 3 with additive withe noise. The process noise represent unmodelled noise and errors in the model. This gives the following process model:

$$\begin{aligned} \dot{\boldsymbol{\omega}}^b_{ib} &= \boldsymbol{I}^{-1}[\boldsymbol{\tau}^b - \boldsymbol{S}(\boldsymbol{\omega}^b_{ib})\boldsymbol{I}\boldsymbol{\omega}^b_{ib}] + \boldsymbol{w}_{d-noise} \qquad (8.1) \\ \begin{bmatrix} \dot{\eta} \\ \dot{\boldsymbol{\varepsilon}} \end{bmatrix} &= \frac{1}{2}\begin{bmatrix} -\boldsymbol{\varepsilon}^T \\ \eta\boldsymbol{I}_{3\times 3} + \boldsymbol{S}(\boldsymbol{\varepsilon}) \end{bmatrix}\boldsymbol{\omega}^b_{ob} + \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix}\boldsymbol{w}_{k-noise} \qquad (8.2) \end{aligned}$$

where $\boldsymbol{w}_{d-noise}$ and $\boldsymbol{w}_{k-noise}$ is assumed to be white noise with standard deviation of $\sigma_{d-noise} = 10^{-8}$ and $\sigma_{k-noise} = 10^{-12}$. This gives makes the real process covariance:

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{\sigma}_{d-noise}\boldsymbol{I}_{3\times 3} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\sigma}_{k-noise}\boldsymbol{I}_{3\times 3} \end{bmatrix} \qquad (8.3)$$

8.1.2 Sensor implementation

The sensors are described in chapter 5, and they are implemented accordingly, except that the sensors are sampled with sample and hold. The sample frequency is given in each case, but what the noise is and how it is implemented in the different sensors is described in this section. The star sensor, and magnetometer data used in this thesis is given in the assignment, and the data for the sun sensor is from (Sunde 2005).

8.1.3 Sensor noise

The sensor noise represent errors in the sensors, due to inaccurate measurements, errors in reference models and digitalization errors. The noise will be implemented differently for the different sensors, this section will describe how the noise is implemented and what the noise is assumed to be.

8.1.4 Sun sensor

The error in the sun sensor is assumed to be white noise with a standard deviation of 0.2°. Since the noise is given in degrees and the true attitude is given i quaternions, the quaternion with noise has to be calculated. This is done by converting the true quaternion to Euler angels, adding the noise term, and converting it back to quaternions. This will make it possible to calculate a sun vector measurement with equation (5.5).

8.1.5 Magnetometer

The error in the magnetometer is calculated with equation (5.10) where the noise vector $\boldsymbol{\omega}_{mag}$ is assumed to be white noise. The magnetometer is assumed to have a error of 0.01°[1], but the $\boldsymbol{\omega}_{mag}$ is given in nT. Since the error is given in degrees, it is necessary to calculate the corresponding error in nT. In the assignment it is given that $1nT = 4arcseconds$ which, since $0.01° = 0.01 \cdot 3600arcsec.$, gives an error of $9nT$.

[1] given in the assignment

8.1.6 Star sensor

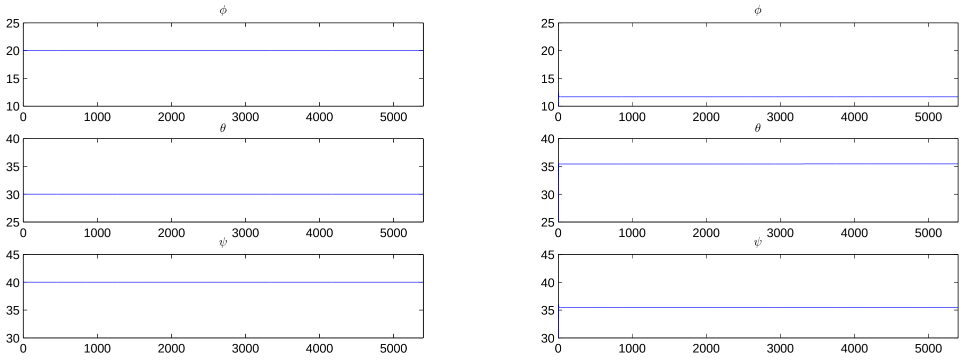
The error of the star sensor is assumed to be white noise, with an accuracy of 1 arcseconds in two directions, and 5 arcseconds in the third. The measurement signal from the star sensor is calculated by converting the true attitude into Euler angels, adding the noise and converting it back to quaternions. The star sensors used in the simulation will be star sensor 1 and star sensor 2. Star sensor 1 will have have the accuracy of $\left[\begin{array}{ccc} 5 & 1 & 1 \end{array}\right]^T$ arcseconds and star sensor 2 will have an accuracy of $\left[\begin{array}{ccc} 1 & 5 & 1 \end{array}\right]^T$ arcseconds.

8.2 Estimation with vector measurement

Many of the measurements available in a spacecraft are vector measurements, such as the sun sensor, earth sensor and magnetometer. This section will present and show simulation for continuous Kalman filters, where magnetometer and sun sensors are used.

Estimation with sun sensor

A single vector measurement can only be used to determine the attitude in 2 directions at any given time, because the rotation around the measurement vector is undetermined. This makes it difficult, if not impossible, to estimate the attitude, using only a sun sensor. To be able to use the sun sensor to estimate the attitude it has to be possible to measure all the angles in a given time span, for instance during one orbit. This might be possible for a specific orbit, but not for the satellite parameters used in this thesis. This can be shown by using the Gauss-Newton method, given in section 7.7, to convert the sun measurement into a measured attitude. By compering the measured attitude, given in figure 8.1(a), with the real attitude, given in figure 8.1(b), it is easy to se that the measurement error is constant during one orbit.



(a) Measured attitude

(b) Real attitude

Figure 8.1: Measured attitude vs real attitude

Estimation with magnetometer

Using a magnetometer to estimate the attitude of the satellite, gives rise to a lot of the same problems as for the sun sensor. It is also a vector measurement, an can only be used to determine two directions of the attitude at any given time. It is possible, how ever, to determine all directions of the attitude during one orbit using historical data. This makes it possible to make an observer that can estimate the attitude using only a magnetometer. This was proven and demonstrated for a discrete system in Psiaki, Martel & Pal (1989), but this proved difficult to adapt and do for a continuous Kalman filter. There may be many reasons for this, one of which is that the dynamics of the continuous Kalman filter is different from the dynamics of the discrete filter, and that the continuous filter therefore does not make the same use of old measurements as the discrete filter.

8.2.1 Estimation, combining sun senor and magnetometer

Combing two different vector measurements with different attitude information makes it possible to determine all three directions without the use of historical data. The magnetometer and the sun sensor will give different attitude information and this makes them suitable for use in a Kalman filter. Using the magnetometer and sun senor in a Kalman filter can be done in two different ways, either by using the measurements directly, as described in section 7.5 and section 7.6, or the quaternion that best relates to the measurement, can be calculated with Gauss-Newton method (given in section 7.7).

Filter description

Since the Gauss-Newton method makes the measurements equations linear this will simplify the filter equation, and will therefore be used in the filter.

The filter equation is given by (7.76)-(7.80) where the innovation $\boldsymbol{\nu}$ is truncated and given by:

$$\boldsymbol{\nu}_{G-N} = \boldsymbol{\varepsilon}_{G-N} - \bar{\boldsymbol{\varepsilon}} \tag{8.4}$$

The different parameters used during simulation are given in table 8.1 and the covariance used in the Kalman filter (not the real covariance) are given in table 8.2:

Initial attitude	Θ	$\begin{bmatrix} 20^\circ & 30^\circ & 40^\circ \end{bmatrix}^T$
Initial angular velocity	ω_{ob}^b	$\begin{bmatrix} 0.005 & 0.003 & 0 \end{bmatrix}^T$
Initial attitude estimation	$\hat{\Theta}$	$\begin{bmatrix} 0^\circ & 0^\circ & 0^\circ \end{bmatrix}^T$
Proportional controller gain	K_p	0.001
Derivative controller gain	K_d	1
Sample frequency, sun sensor	$f_{sample-sun}$	100Hz
Sample frequency, magnetometer	$f_{sample-mag}$	100Hz

Table 8.1: The parameter used in the Kalman filter with sun and magnetometer measurements

Measurement covariance 1	$\boldsymbol{R}$	$1e^{-5}\cdot$	$\begin{bmatrix} 9 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 9 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 9 \end{bmatrix}$
Process covariance	$\boldsymbol{Q}$		$\begin{bmatrix} 1e^{-18}\boldsymbol{I}_{3\times3} & \mathbf{0} \\ \mathbf{0} & 1e^{-10}\boldsymbol{I}_{3\times3} \end{bmatrix}$

Table 8.2: The covariance used in the Kalman filter

Performance with constant angular velocity

This section looks into the Kalman filters estimate, whiteout using the estimate state to control the satellite. The angular velocity is instead kept constant with feedback control from the true angular velocity. The initial values used when simulating is given in table 8.1 and table 8.2, except for K_p, which is set to zero. From the estimation errors, given in figure 8.2, it is easy to see that the estimated attitude converges to the real attitude, and has a an estimation error of ±2° after it has converged.

Performance with control feedback

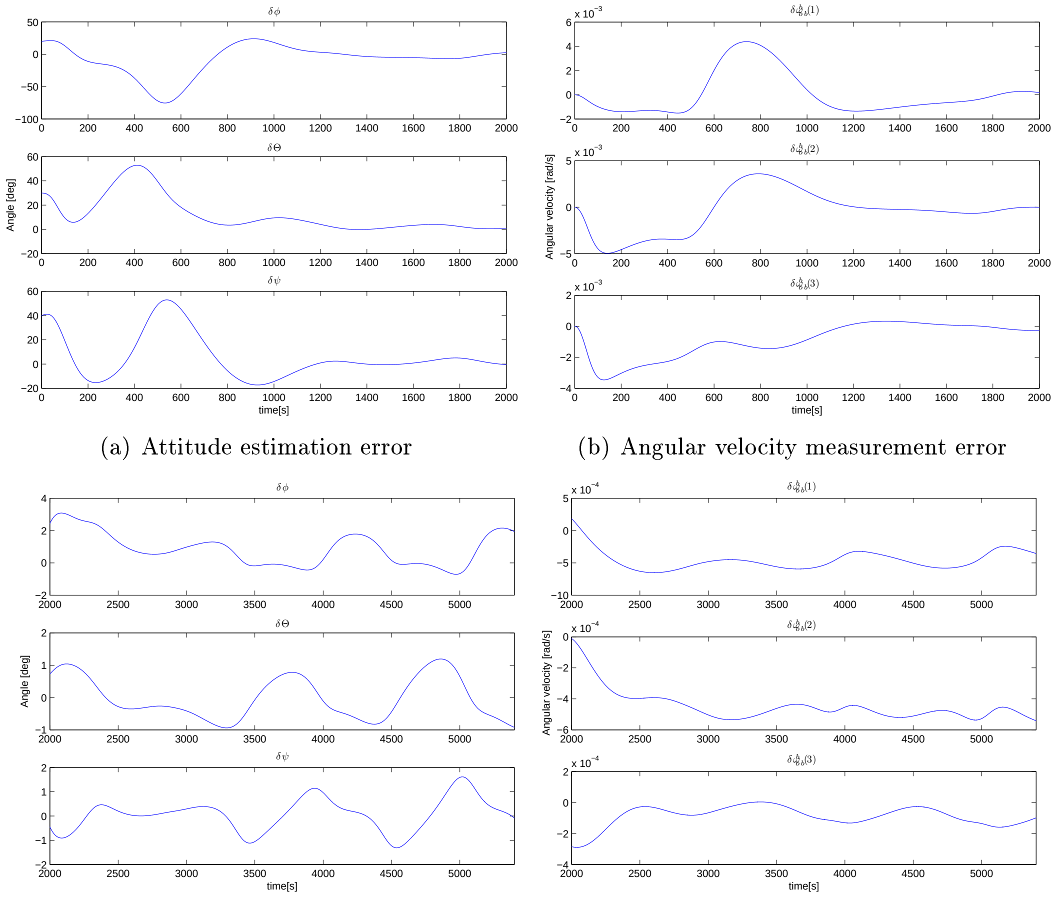
In this section the filter estimation is used to control the attitude to a desired attitude of $\begin{bmatrix} 0^\circ & 0^\circ & 0^\circ \end{bmatrix}$. The controller used is the one described in 8.1. Since the controller used is nonphysical the control error is of less interest and will not be shown. What is of interest is the estimation error and stability of the complete system, when the estimated states is used to control the system. From figure 8.3 one can see that both the estimated and real attitude, and the estimated and real angular velocity converges to zero and figure 8.4 shows that the estimation error stabilizes beneath 0.1°.

Filter comment

As seen from the simulations above, combing the magnetometer and sun sensor measurements with the Gauss-Newton method, and utilizing this in the continuous Kalman filter, gives an estimation error of ±0.1° for the feedback controlled system. This is not within the required estimation error of 0.001°, and the sun sensor and magnetometer can therefore not be the only attitude sensors used in the Kalman filter.

8.3 Estimation using star sensor

The star senors are the most accurate sensors used in this thesis, and they have an accuracy of 1 arcseconds in two directions, and 5 arcseconds in the third. To achieve an accuracy of 1 arcseconds in every directions, two stars senors, mounted in different bearings, will be used. Using two star sensors also gives measurement redundancy.



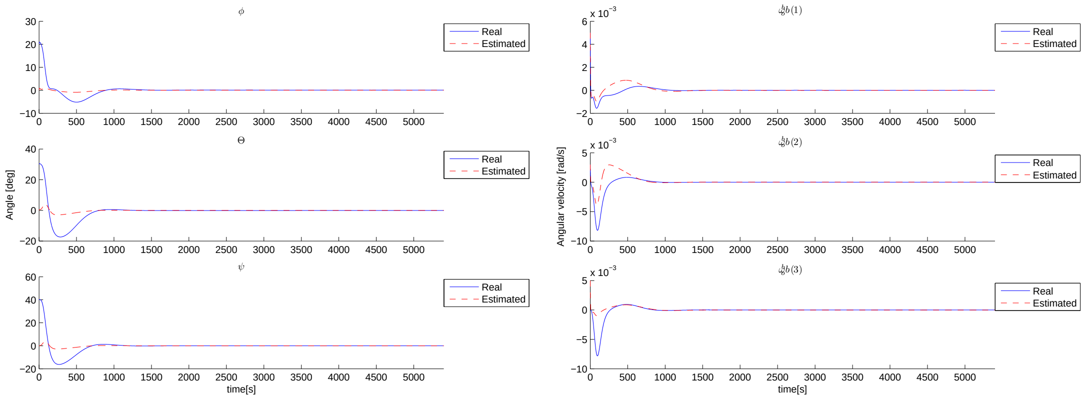
(a) Attitude estimation error

(b) Angular velocity measurement error

(c) Attitude estimation error zoom

(d) Angular velocity measurement error zoom

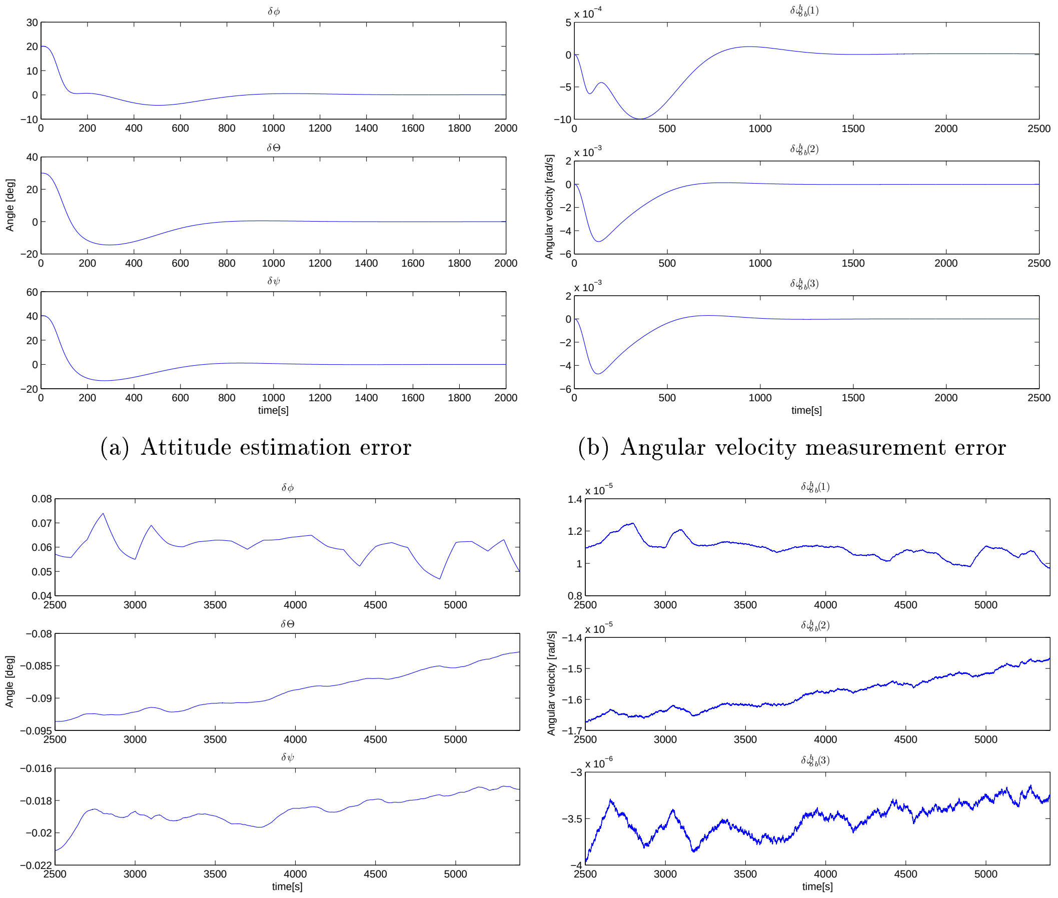
Figure 8.2: Estimation error with sun and magnetometer measurement without feedback.



(a) Estimation and real attitude

(b) Estimated an real angular velocity

Figure 8.3: Real and estimated attitude and angular velocity using measurement from magnetometer and sun sensor.



(a) Attitude estimation error

(b) Angular velocity measurement error

(c) Attitude estimation error

(d) Angular velocity measurement error

Figure 8.4: Estimation error with sun and magnetometer measurement with feedback from the estimated states.

8.3.1 Filter description

The filter equations is given by (7.76)-(7.80) where the innovation $\boldsymbol{\nu}$ is truncated and given by:

$$\boldsymbol{\nu}_{star} = \begin{bmatrix} \varepsilon_{star1} \\ \varepsilon_{star2} \end{bmatrix} - \begin{bmatrix} \hat{\varepsilon} \\ \hat{\varepsilon} \end{bmatrix} \tag{8.5}$$

and the different paraments used when simulating this Kalman filter is given in table 8.3 and the covariance used in the Kalman filter is (not the real covariance) is given in table 8.4.

Initial attitude	Θ	$\begin{bmatrix} 20^\circ & 30^\circ & 40^\circ \end{bmatrix}^T$
Initial angular velocity	ω_{ob}^b	$\begin{bmatrix} 0.005 & 0.003 & 0 \end{bmatrix}^T$
Initial attitude estimation	$\hat{\Theta}$	$\begin{bmatrix} 0^\circ & 0^\circ & 0^\circ \end{bmatrix}^T$
Proportional controller gain	K_p	0.001
Derivative controller gain	K_d	1
Sample frequency, star sensors	f_{star}	0.5Hz

Table 8.3: The parameter used in the Kalman filter with star sensor measurements

Measurement covariance	$\boldsymbol{R}$		$\begin{bmatrix} \boldsymbol{R}_{star1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R}_{star2} \end{bmatrix}$
Measurement covariance Star sensor 1	$\boldsymbol{R}_{star1}$	$1e^{-9}\cdot$	$\begin{bmatrix} 3.858 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0.772 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0.772 \end{bmatrix}$
Measurement covariance Star sensor 2	$\boldsymbol{R}_{star2}$	$1e^{-9}\cdot$	$\begin{bmatrix} 0.772 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 3.858 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0.772 \end{bmatrix}$
Process covariance	$\boldsymbol{Q}$		$\begin{bmatrix} 1e^{-18}\boldsymbol{I}_{3\times3} & \mathbf{0} \\ \mathbf{0} & 1e^{-10}\boldsymbol{I}_{3\times3} \end{bmatrix}$

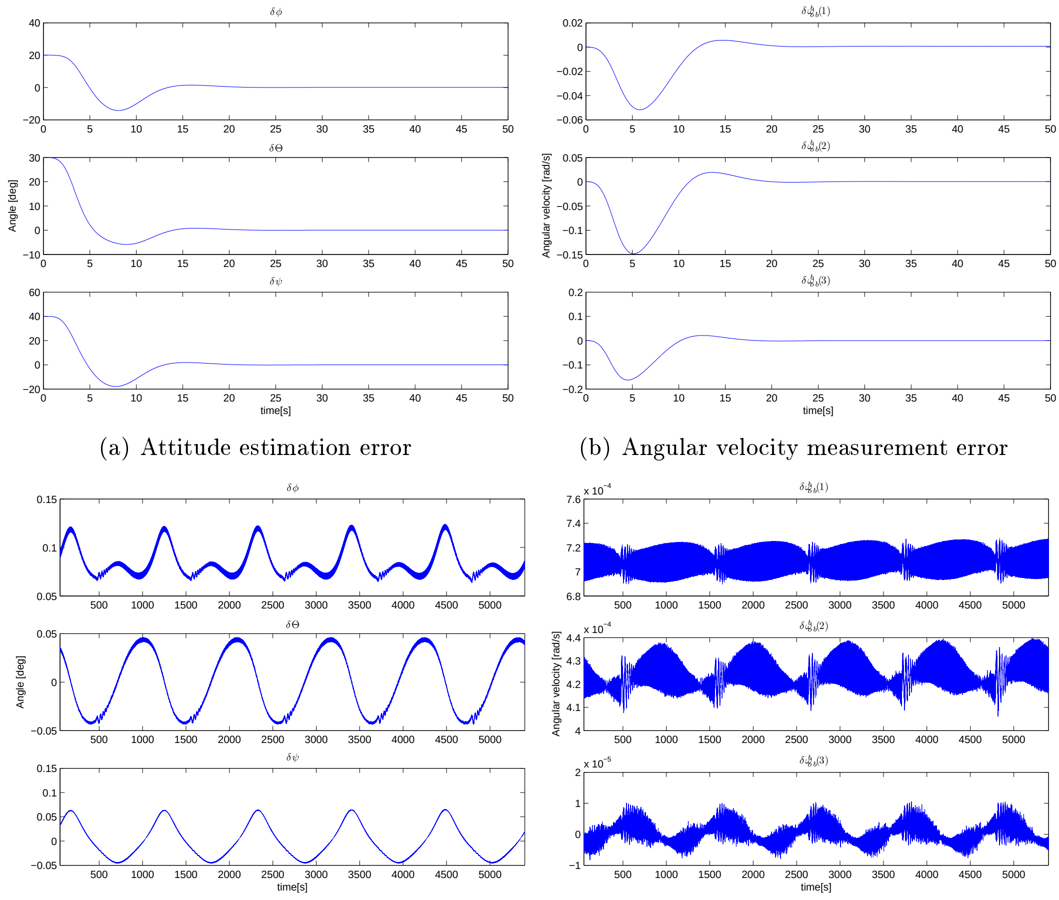
Table 8.4: The covariance used in the Kalman filter

8.3.2 Performance without control feedback

The performance of the Kalman filter, without using the estimated states to control the satellite is found by keeping the angular velocity constant with feedback control from the true states. The initial values used during simulations is given in table 8.3. From the estimation error, given in figure 8.5, it can be seen that the filter stabilizes with an oscillating error of 0.15°.

8.3.3 Performance with control feedback

In this section the filter estimation is used to control the attitude to a desired attitude of $\begin{bmatrix} 0^\circ & 0^\circ & 0^\circ \end{bmatrix}$. The controller used is the one described in 8.1. Since the controller



(a) Attitude estimation error

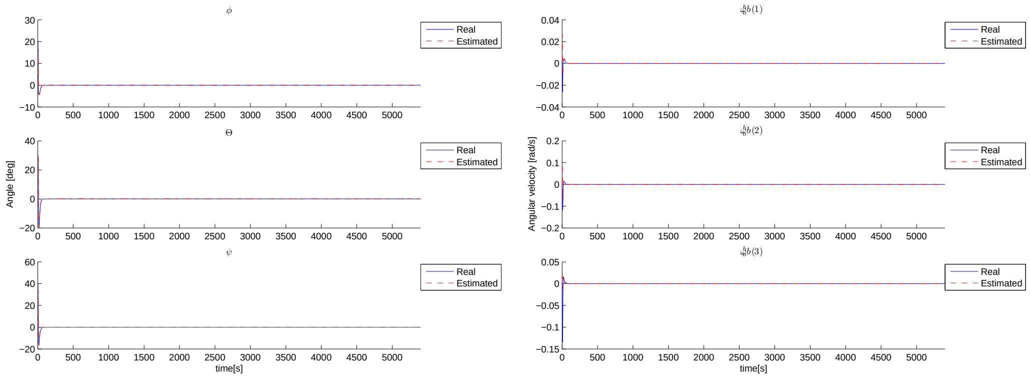
(b) Angular velocity measurement error

(c) Attitude estimation error

(d) Angular velocity measurement error

Figure 8.5: Estimation error with star measurement, without feedback.

used is nonphysical the control error is of less interest. What is of interest is the estimation error and stability of the complete system, when the estimated stat is used to control the system. From figure 8.6 one can se that both the estimated and real attitude and angular velocity converges to zero and figure 8.7 shows that the estimation error stabilizes beneath 0.0005°.



(a) Estimation and real attitude

(b) Estimated an real angular velocity

Figure 8.6: Real and estimated attitude and angular velocity using measurement from star sensors.

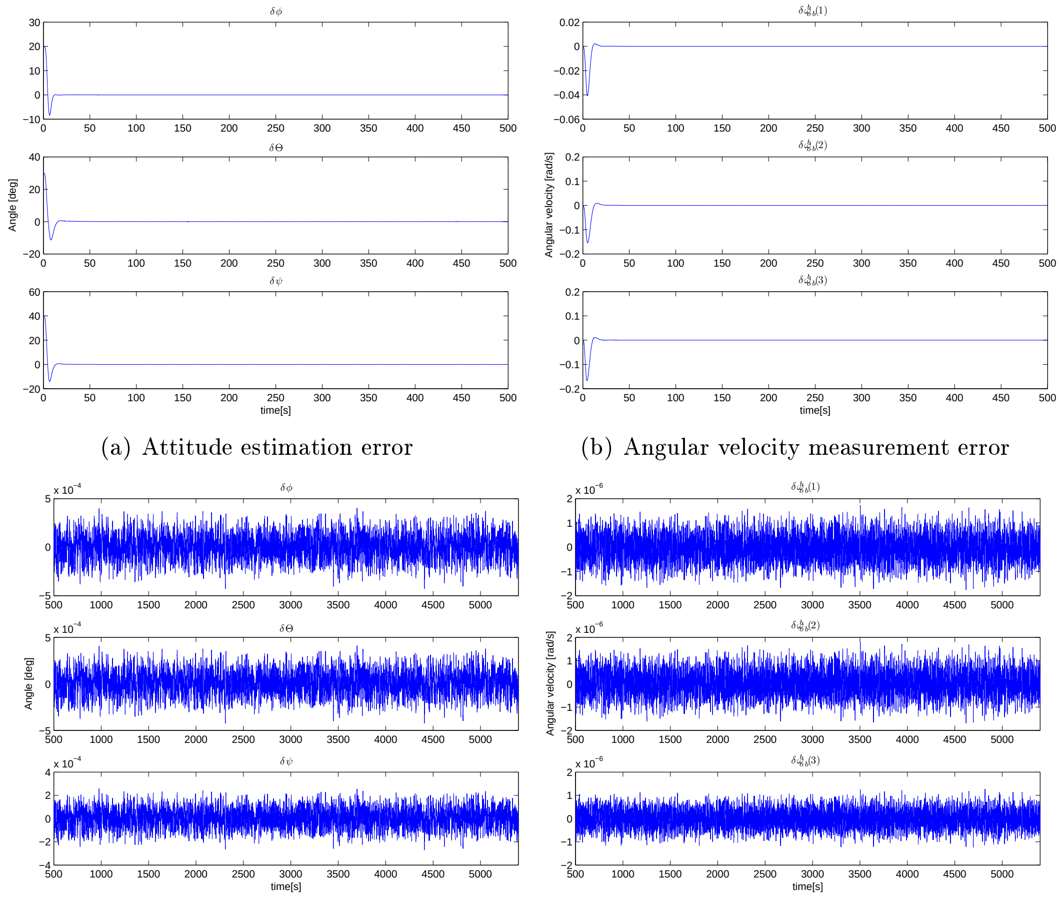
Filter comment

Combing two star sensors in a continuous extended Kalman filter, gives an estimation error of ±0.0005° for the feedback controlled system, and below 0.15° for the system with constant angular velocity. The estimation error is within the required limit when the attitude is controlled, and is limited if the attitude control fails.

8.4 Estimation with sun sensor, magnetometer and star sensor.

While the star sensors give the accuracy asked for, there is a problem with the low sampling frequency, since low sampling frequency gives the estimator low bandwidth and therefore makes it difficult to estimate the attitude with acceptable error if the angular velocity is high. On way to remedy this are to combine the slow star sensor with faster measurements such as the sun sensor and magnetometer. Since both the sun sensor and magnetometer are pointing vectors and only can be used to determine 2 directions each, the filter will use the Gauss-Newton method to convert the magnetometer and sun senor measurements into a quaternion that best relates to the measurements. This makes the measurement equations linear and therefore simplifies the Kalman filter. The Kalman filter used is given by equations (7.76)-(7.80) where the innovation $\boldsymbol{\nu}$ is truncated and given by:

$$\boldsymbol{\nu}_{all} = \begin{bmatrix} \varepsilon_{star1} \\ \varepsilon_{star2} \\ \varepsilon_{G-N} \end{bmatrix} - \begin{bmatrix} \hat{\varepsilon} \\ \hat{\varepsilon} \\ \hat{\varepsilon} \end{bmatrix} \tag{8.6}$$



(a) Attitude estimation error

(b) Angular velocity measurement error

(c) Attitude estimation error

(d) Angular velocity measurement error

Figure 8.7: Estimation error with star measurements, with feedback from the estimated states.

and the Kalman filter parameter is given by table 8.5 and the covariance used in the Kalman filter (not the real covariance) is given in table 8.2.

Initial attitude	Θ	$\begin{bmatrix} 20^\circ & 30^\circ & 40^\circ \end{bmatrix}^T$
Initial angular velocity	ω_{ob}^b	$\begin{bmatrix} 0.005 & 0.003 & 0 \end{bmatrix}^T$
Initial attitude estimation	$\hat{\Theta}$	$\begin{bmatrix} 0^\circ & 0^\circ & 0^\circ \end{bmatrix}^T$
Proportional controller gain	K_p	0.001
Derivative controller gain	K_d	1
Sample frequency, star sensors	f_{star}	0.5Hz
Sample frequency, sun sensor	$f_{sample-sun}$	100Hz
Sample frequency, magnetometer	$f_{sample-mag}$	100Hz

Table 8.5: The parameter used in the Kalman filter with star sensor, magnetometer and sun sensor.

Measurement covariance	$\boldsymbol{R}$		$\begin{bmatrix} \boldsymbol{R}_{star1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_{star2} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_{G-N} \end{bmatrix}$
Measurement covariance Star sensor 1	$\boldsymbol{R}_{star1}$	$1e^{-9}\cdot$	$\begin{bmatrix} 3.858 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & 0.772 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & 0.772 \end{bmatrix}$
Measurement covariance Star sensor 2	$\boldsymbol{R}_{star2}$	$1e^{-9}\cdot$	$\begin{bmatrix} 0.772 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & 3.858 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & 0.772 \end{bmatrix}$
Measurement covariance Gauss-Newton 1	$\boldsymbol{R}_{G-N}$	$1e^{-9}\cdot$	$\begin{bmatrix} 9 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & 9 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & 9 \end{bmatrix}$
Process covariance	$\boldsymbol{Q}$		$\begin{bmatrix} 1e^{-18}\boldsymbol{I}_{3\times3} & \boldsymbol{0} \\ \boldsymbol{0} & 1e^{-10}\boldsymbol{I}_{3\times3} \end{bmatrix}$

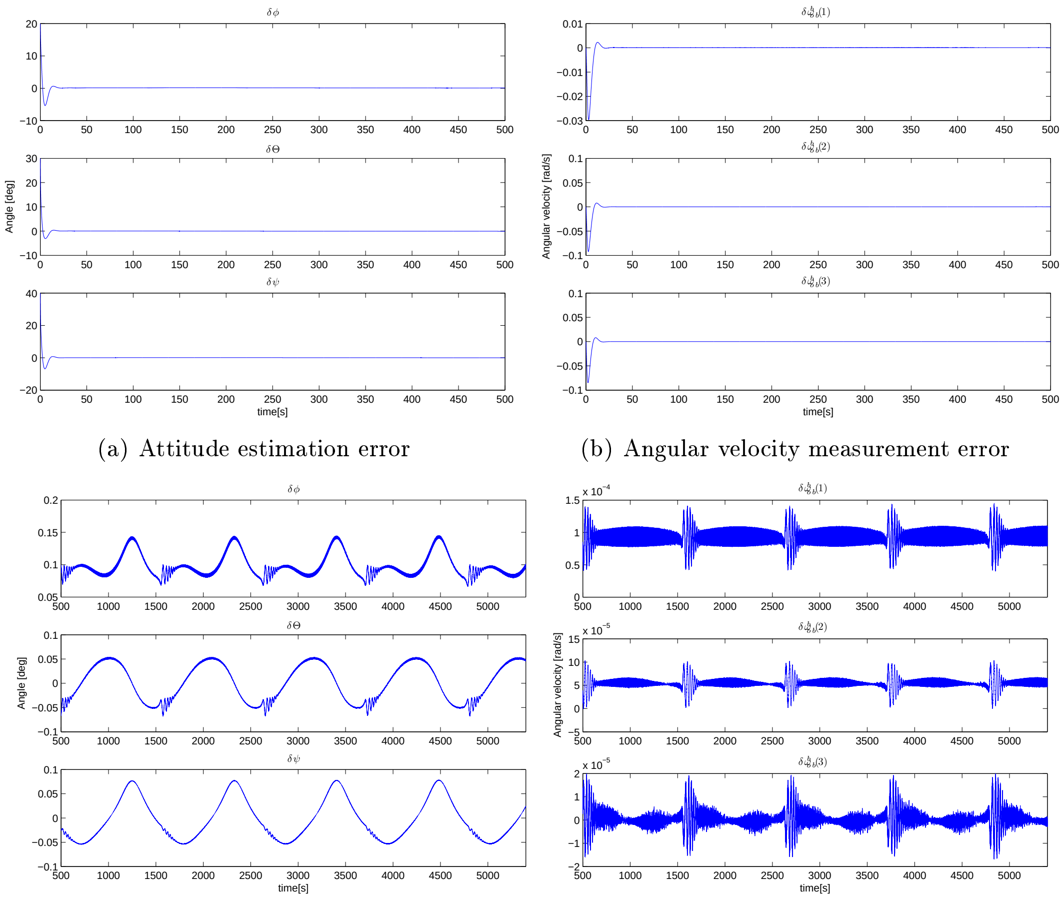
Table 8.6: The covariance used in the Kalman filter

8.4.1 Performance without control feedback

This section will look at the preforming of the estimator with a constant angular speed. To do this the angular velocity of the satellite is controlled with feedback form the real velocity. The initial values used in the simulation are given in table 8.5. From the estimation error, given in figure 8.8, it can be seen that the filter stabilizes with an oscillating error of 0.15°.

8.4.2 Performance with control feedback

This section will analyze how the system behaves and what the estimation error becomes when the estimated signal is used inn feedback control of the satellite. The



(a) Attitude estimation error

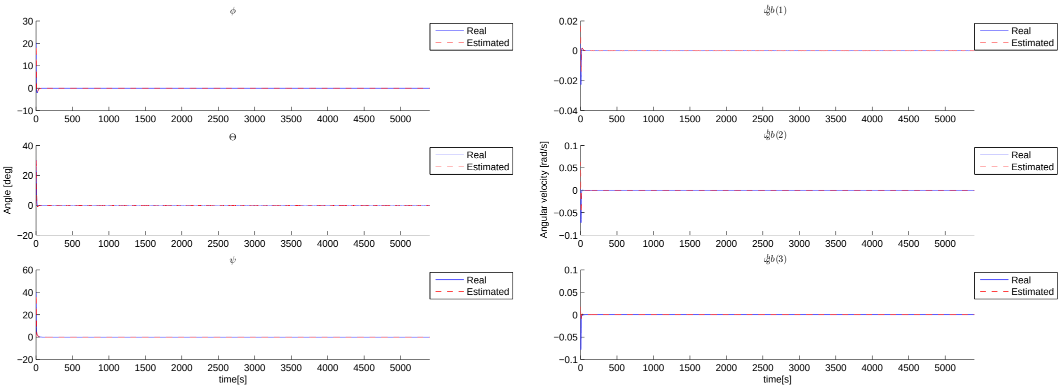
(b) Angular velocity measurement error

(c) Attitude estimation error

(d) Angular velocity measurement error

Figure 8.8: Estimation error with measurement from 2 star sensors, one magnetometer and sun sensor, without feedback.

controller described in section 8.1 is used to control the attitude to a desired attitude of $\begin{bmatrix} 0^\circ & 0^\circ & 0^\circ \end{bmatrix}^T$. The filter and simulation parameters is given in table 8.5 and 8.2. From figure 8.9 one can set that both the estimated and real attitude converges to zero, and figure 8.10 shows that the attitude error stabilizing beneath 0.0005°.



(a) Estimation and real attitude

(b) Estimated an real angular velocity

Figure 8.9: Real and estimated attitude and angular velocity using two star sensors, one magnetometer and one sun sensor, with feedback from the estimated states..

8.4.3 Detecting and removing faulty measurements in the Kalman filter

Detecting and removing faulty measurements is important to make the estimation more robust. To detect if a measurement gives out wrong attitude it can be compered with the estimated attitude and other measurement if available. After it has been detected it can easily be removed from the Kalman filter by setting the measurement matrix of the defect sensor to **0**.

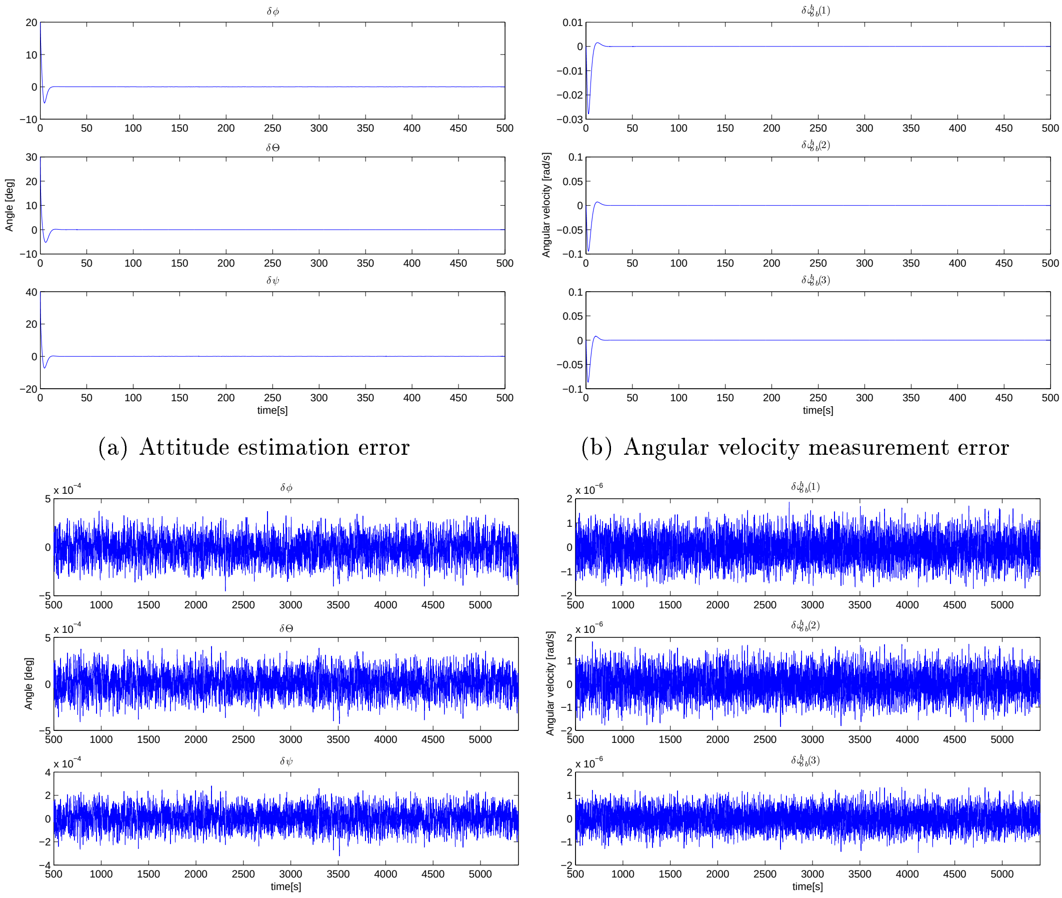
The Kalman filter used here will be the same as the one used in section 8.4, except that the measurements matrix, $\boldsymbol{H}$, will be replaced by a:

$$\boldsymbol{H}_{fault} = \begin{bmatrix} a_{star1}\boldsymbol{H}_{star1} & \mathbf{0} & \mathbf{0} \\ a_{star2}\boldsymbol{H}_{star2} & \mathbf{0} & \mathbf{0} \\ a_{G-N}\boldsymbol{H}_{G-N} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{8.7}$$

where a is put to zero if the sensor is faulty, and 1 if the sensor is operating.

8.4.4 Detecting a faulty sensor

To determine if a is one or zero, the faulty sensor must be detected. This can be done by compering the attitude given by the measurements with each other and with the estimated measurement. If one assumes that the initial error of the estimation has disappeared, and that the standard deviation of the estimation and measurement error is known, the easiest way to do this will be to compare the measured attitude($\boldsymbol{\Theta}_{meas}$) of the sensor with the estimated attitude($\hat{\boldsymbol{\Theta}}$), and conclude that the measurement has to



(a) Attitude estimation error

(b) Angular velocity measurement error

(c) Attitude estimation error

(d) Angular velocity measurement error

Figure 8.10: Estimation error with two star sensors, one magnetometer and one sun sensor, with feedback from the estimated states.

be wrong if it diverges to much from the estimation. This can be achieved by calculating a in the following manner:

$$a = \begin{cases} 1 & \text{if} \quad \|\mathbf{\Theta}_{meas} - \hat{\mathbf{\Theta}}\|_{\text{inf}} < p \\ 0 & \text{if} \quad \|\mathbf{\Theta}_{meas} - \hat{\mathbf{\Theta}}\|_{\text{inf}} \geq p \end{cases} \tag{8.8}$$

where p is a function between 0 and 360, that can depend on the angular velocity, the sample frequency of the measurement and the noise in the model and in the measurement. Since the attitude angel is forced to be between $-180°$ and $180°$, there will be a problem when the attitude is close to $\pm 180°$, but this can be factored in by changing equation (8.8) into:

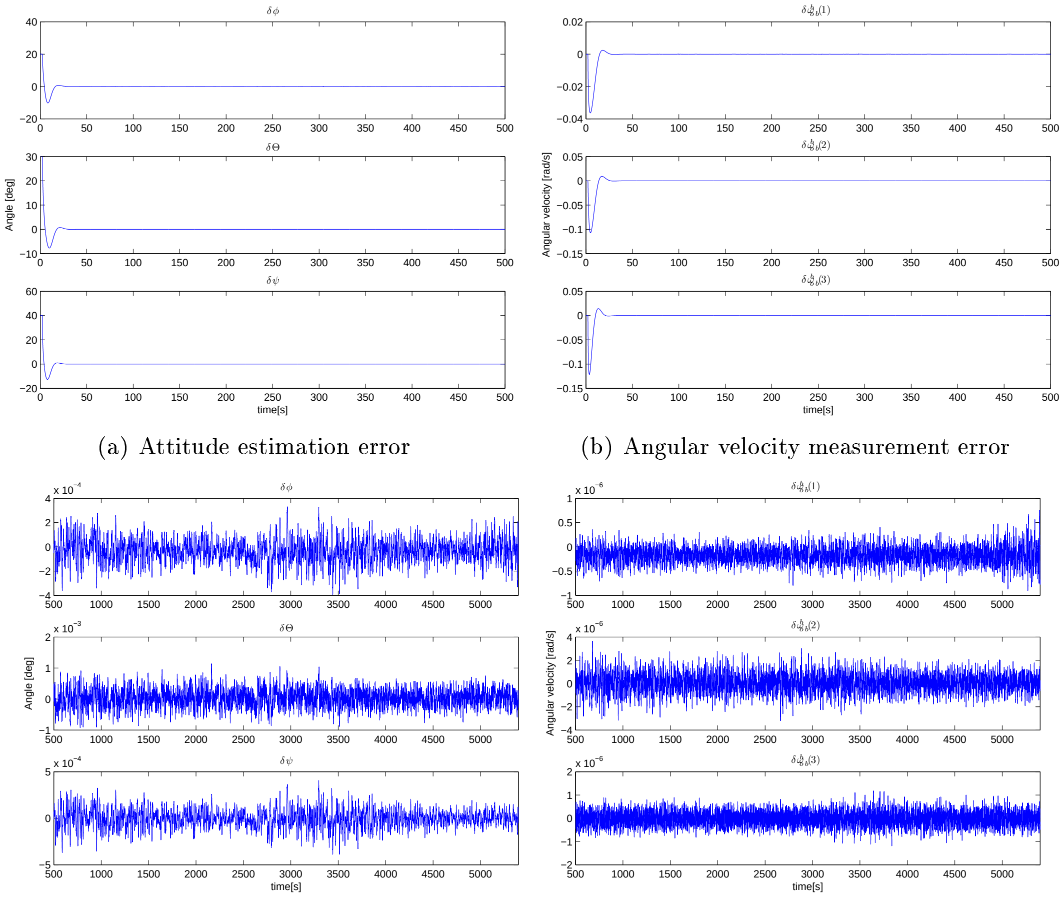
$$a = \begin{cases} 1 & \text{if} \quad \|\mathbf{\Theta}_{meas} - \hat{\mathbf{\Theta}}\|_{\text{inf}} < p \\ 1 & \text{if} \quad 360 - \|\mathbf{\Theta}_{meas} - \hat{\mathbf{\Theta}}\|_{\text{inf}} < p \\ 0 & \text{if} \quad \text{else} \end{cases} \tag{8.9}$$

This only works if the initial error of the estimation has disappeared, and do not work right after the estimation has started. To remedy this, the fault detection can be turned on after the initial estimation error has had time to settle. An other way to remedy this, is to make use of the redundancy of the system. Since there is four different sensors giving three different attitude measurement (star1, star2 and Gauss-Newton), it is unlikely that more than one of this is faulty at the same time and if they are, that they will give the same attitude. And the criteria for the sensor to be omitted can be that it has to diverge from the estimation and at least on other measurement. This can be done in the same manner as (8.9), and will be:

$$a_{meas1} = \begin{cases} 1 & \text{if} \quad \|\mathbf{\Theta}_{meas1} - \hat{\mathbf{\Theta}}\|_{\text{inf}} < p_1 \\ 1 & \text{if} \quad \|\mathbf{\Theta}_{meas1} - \mathbf{\Theta}_{meas2}\|_{\text{inf}} < p_2 \\ 1 & \text{if} \quad \|\mathbf{\Theta}_{meas1} - \mathbf{\Theta}_{meas3}\|_{\text{inf}} < p_3 \\ 1 & \text{if} \quad 360 - \|\mathbf{\Theta}_{meas1} - \hat{\mathbf{\Theta}}\|_{\text{inf}} < p_1 \\ 1 & \text{if} \quad 360 - \|\mathbf{\Theta}_{meas1} - \mathbf{\Theta}_{meas2}\|_{\text{inf}} < p_2 \\ 1 & \text{if} \quad 360 - \|\mathbf{\Theta}_{meas1} - \mathbf{\Theta}_{meas3}\|_{\text{inf}} < p_3 \\ 0 & \text{if} \quad \text{else} \end{cases} \tag{8.10}$$

where $\mathbf{\Theta}_{meas1}$ is the attitude of the the sensor tested and $\mathbf{\Theta}_{meas2}$ and $\mathbf{\Theta}_{meas3}$ is the attitude of the two other attitude measurements. p_1, p_2 and p_3 are the criteria for removing the sensor, and will depend on the same factor as p in equation (8.8).

The fault detection system is simulated with filter and system parameters given in table 8.5. Equation (8.10) is used to calculate a_{star1}, a_{star2} and a_{G-N} inn H_{fault} (8.7), where p_1, p_2 and p_3 is set to a constant value of 5. Figure 8.11 shows the estimation error when star sensor 1 is faulty, and figurer 8.12 when the magnetometer is faulty. From the figures it can be sen that the estimation error is below 0.001° when the star sensor is faulty and below 0.0005° when the magnetometer is faulty, and thus complying to the stringent estimation demands for the system.



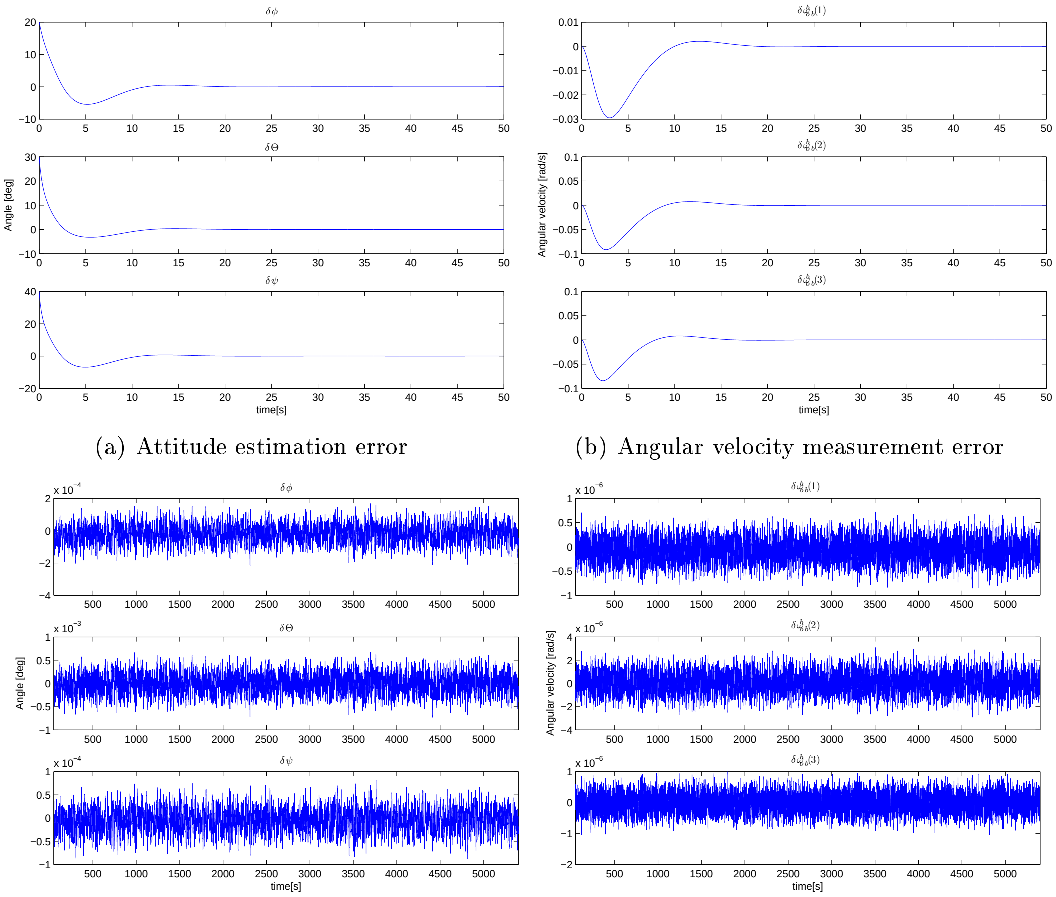
(a) Attitude estimation error

(b) Angular velocity measurement error

(c) Attitude estimation error

(d) Angular velocity measurement error

Figure 8.11: Estimation error with star sensor 1 faulty, without feedback.



(a) Attitude estimation error

(b) Angular velocity measurement error

(c) Attitude estimation error

(d) Angular velocity measurement error

Figure 8.12: Estimation error with faulty magnetometer, without feedback.

Filter comment

As seen from the simulations above, combing the two star sensors with the sun sensor and magnetometer gives the accuracy of ±0.005°. This is below the demanded accuracy, and it is the same as for the filter with measurements from two star sensors. Combing the star sensors with the magnetometer and sun sensor, do not give an improved estimate accuracy, but the added redundancy of the magnetometer and sun sensor, makes it possible to use the simple fault detections scheme given below, and makes the system more robust.

8.5 Reducing the sampling frequency

The sun sensor, magnetometer and star sensor all gives out sampled measurement date. It is of interest to find out what the lowest sampling rate the Kalman filter needs from the different sensors and still be below the demanded estimation error of 0.001°. In the previous simulation it was shown that the magnetometer and sun sensor only adds redundancy and do not increase the accuracy of the feedback controlled system, this section will therefore only look at the result of reducing the frequency of the star sensor.

8.5.1 The filter

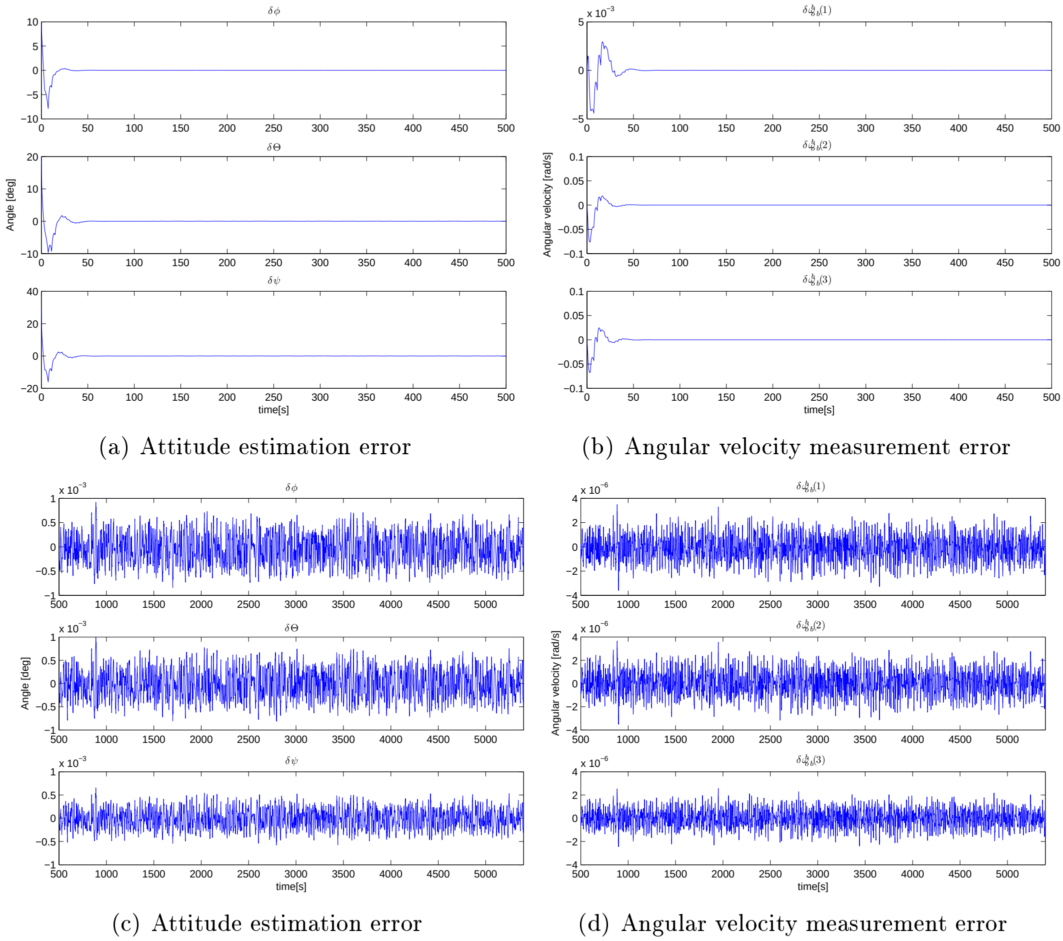
The filter used in this section will be the same as the one given in section 8.4. The filter equation is given by (7.76)-(7.80), and the innovation given by (8.4) and the initial values of the filter and simulation parameters is given by table 8.5 and 8.4. Since all the sensors will work during the simulation, the fault detection scheme described in section 8.4.4 is unnecessary, and will therefore be omitted.

8.5.2 Reducing the sampling frequency of the star sensors

The star sensors have the lowest samplings rate, and highest accuracy. Because the sampling rate of the star sensors is low to begin with, reducing their sampling frequency further will have a high impact on the estimation error. The aim is to find out how low the sampling frequency of the star sensor can be, before the estimation error become larger then 0.001° with feedback control from the estimated states. The controller used is described in section 8.1, and the desired attitude is $\begin{bmatrix} 0^\circ & 0^\circ & 0^\circ \end{bmatrix}^T$. The filter and simulation parameters are given in table 8.5. Figure 8.13 shows the estimation error with a star sampling frequency of $0.27Hz$. The estimation error of the system with this frequency is right below the demanded error of 0.001°, and reducing the frequency further will deteriorate the estimation further.

8.5.3 Filter comment

This section showed that Kalman filter is able to estimate the sensor with an error beneath 0.01° with the star sensors sampling frequency reduced from 0.5 to $0.27HZ$. This result is achieved with a PD regulator, using feedback from the estimated states and the simulation and filter parameters given in table 8.5 and table 8.6. This reduction is therefore not necessarily achievable with a controller with limitations. It will also



(a) Attitude estimation error

(b) Angular velocity measurement error

(c) Attitude estimation error

(d) Angular velocity measurement error

Figure 8.13: Estimation error with reduced sampling frequency on the star sensors.

be possible to achieve better results by using different control parameters and tuning parameters.

Chapter 9
Conclusion

In this thesis a continuous extended Kalman filter has been used to estimate the attitude of a low polar earth orbiting satellite. The satellite is intended to conduct optical and radar observations of the earths surface, which gave a demanded attitude accuracy of 0.001°. The Kalman filter used quaternions to represent the attitude, and utilized measurements from star sensors, a sun sensor and a 3-axis magnetometer to correct the estimations.

This thesis has given a mathematical model for the magnetometer and sun sensor. It has described how to implement the different sensor measurements in the Kalman filter and how to combine the magnetometer and sun sensor measurement into one attitude measurement, using a Gauss-Newton optimization algorithm.

Different Kalman filters, combining the different sensor measurements, have been presented and simulated. It was shown that a Kalman filter using two star sensor, with a sampling frequency of $0.5Hz$, is able to measure the attitude with an accuracy of 1.8 arcseconds (0.0005°), and therefore complies to the stringent attitude demands. An algorithm, using all the sensors, which detects an removes faulty measurements from the Kalman filter has also been presented.

References

Bak, T. (1999), Spacecraft Attitude Determination - a Magnetometer Approach, PhD thesis, Aalborg University - Department of Control Engineering.

Chiang, Y. T., Chang, F. R., Wang, L. S., Jan, Y. W. & Ting, L. H. (2001), 'Data fusion of three attitude sensors', *SICE* .

Egeland, O. & Gravdahl, J. T. (2002), *Modeling and Simulation for Automatic Control*, Marine Cybernetics.

Fauske, K. M. (2002), Ncub attitude controll,student projec, Technical report, NTNU, Department of Engineering Cybernetics.

Fossen, T. I. (2002), *Marine Control Systems*, Tapir Trykk.

Henriksen, R. (1998), Stokastiske systemer - analyse, estimering og regulering. NTNU, Department of Engineering Cybernetics.

Kaplan, M. H. (1976), *Modern spacecraft dynamics & control.*, Jhon Wiley & Sons New York.

Kristiansen, R. (2000), Attitude control of a microsatellite, Master's thesis, NTNU, Department of Engineering Cybernetics.

Kyrkjebø, E. (2000), Satellite attitude determination - three-axis attitude determination using magnetometer ad star tracker., Master's thesis, NTNU, Department of Engineering Cybernetics.

Lefferts, E., Markley, F. & Shuster, M. (1982), 'Kalman filtering for spacecraft attitude estimation', *Jurnal of guidence, control, and dynamics* .

Marins, J. L., Yun, X., Bachmann, E. R., McGhee, R. B. & Zyda, M. J. (2001), An extended kalman filter for quaternion-baased orientation estimation using marg sensors, IEEE/RSJ International Conference on Intelligent Robots and Systems.

Nocedal, J. & Wright, S. J. (1999), *Numerical Optimization*, Springer.

Ose, S. (2004), Attitude determination for the norwegian student satelite ncube, Master's thesis, NTNU, Department of Engineering Cybernetics.

Psiaki, M. L., Martel, F. & Pal, P. K. (1989), 'Three-axis attitude determination via kalman filterin of magnetometer data', *Jurnal of guidence, control, and dynamics* .

Sadiku, M. N. O. (2001), *elements of electromagnetics*, 3 edn, Oxford University Press.

Sellers, J. J. (2000), *Understanding Space*, 2nd edn, Edition, McGrawHill.

Steyn, W. H. (1994), Full satellite state determination from vectro oservations, IFA.

Sunde, B. O. (2005), Attitude determination for satellites., Master's thesis, NTNU, Department of Engineering Cybernetics.

Svartveit, K. (2003), Attitude determination of the ncube satellite, Master's thesis, NTNU, Department of Engineering Cybernetics.

Vallado, D. A. (2001), *Fundamentals of Astrodynamicss and Applications*, 2nd edn, Kluwer Academic Publishers.

Øverby, E. J. (2004), Attitude controlfor the norwegian student satellite ncube, Master's thesis, NTNU, Department of Engineering Cybernetics.

Wertz, J. R. & Larson, W. J. (1999), *Space mission analysis and design*, Published jointly by Microcosm Press and Kluwer Academic Publishers.

www.esri.com (9.9.2004), 'http://www.esri.com/mapmuseum/mapbook_gallery/volume16/geology2.html'.

Zhao, M., Peng, Q., Zeng, Y., Shi, M., Huang, Y. & Li, J. (2004), Adaptive orbital navigation algorithm using magnetometers, Fifth World Congress on Intelligent Control and Automation.

Appendix A

Deductions

This chapter contains deductions refereed to in the thesis

Deduction of the continuous Gauss-Newton method

Starting with the iterative equation (7.59) and using the assumptions :

$$\dot{\hat{\boldsymbol{q}}}^{GN} \approx \frac{\hat{\boldsymbol{q}}^{GN}_{k+1} - \hat{\boldsymbol{q}}^{GN}_{k}}{h} \tag{A.1}$$

and that h is the normalized step size, gives :

$$\dot{\hat{\boldsymbol{q}}}^{GN} \approx -[\boldsymbol{J}^T(\hat{\boldsymbol{q}}^{GN})\boldsymbol{J}(\hat{\boldsymbol{q}}^{GN})]^{-1}\boldsymbol{J}^T(\hat{\boldsymbol{q}}^{GN})\boldsymbol{\epsilon}(\hat{\boldsymbol{q}}^{GN}) \tag{A.2}$$

Appendix B

The simulink model of the system

This appendix will display the main simulink blocks. The first section will be the top level of the simulink diagram, and it will have one section for all of the blocks in the top level.

Simulink diagram, top level

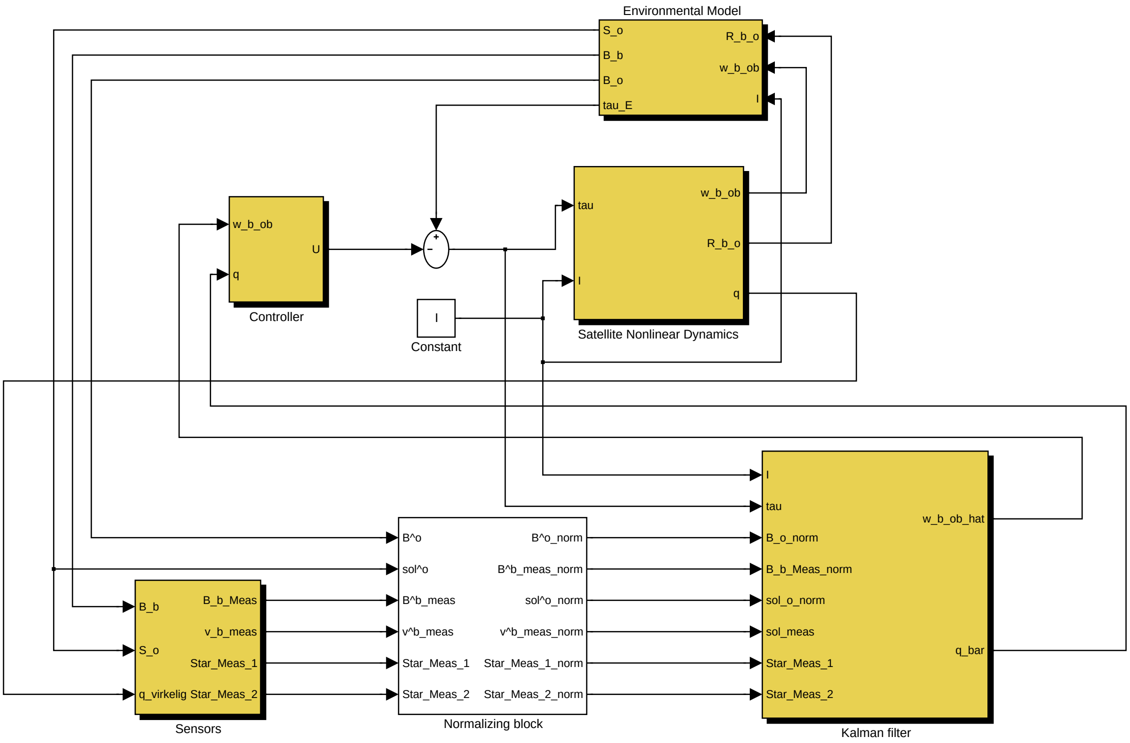


Figure B.1: Top level simulink diagram

The Satellites nonlinear dynamic

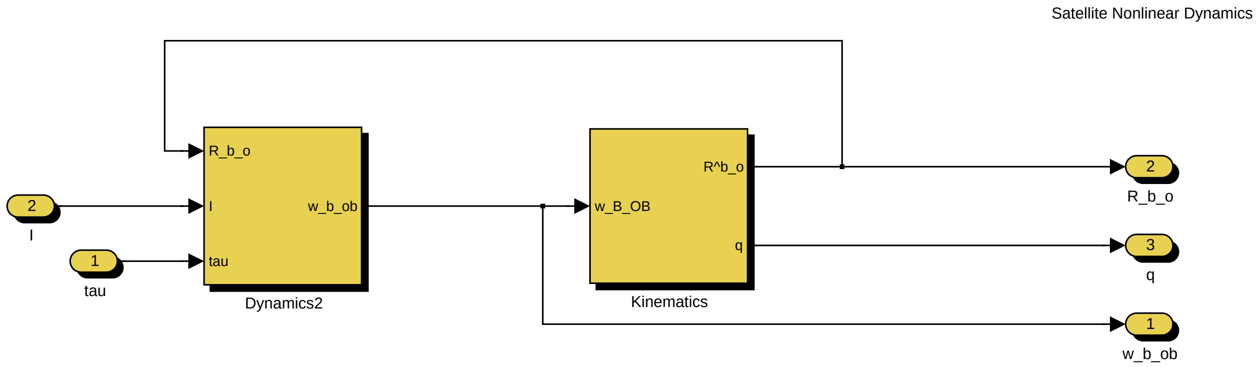


Figure B.2: The Satellite Nonlinear Dynamics block

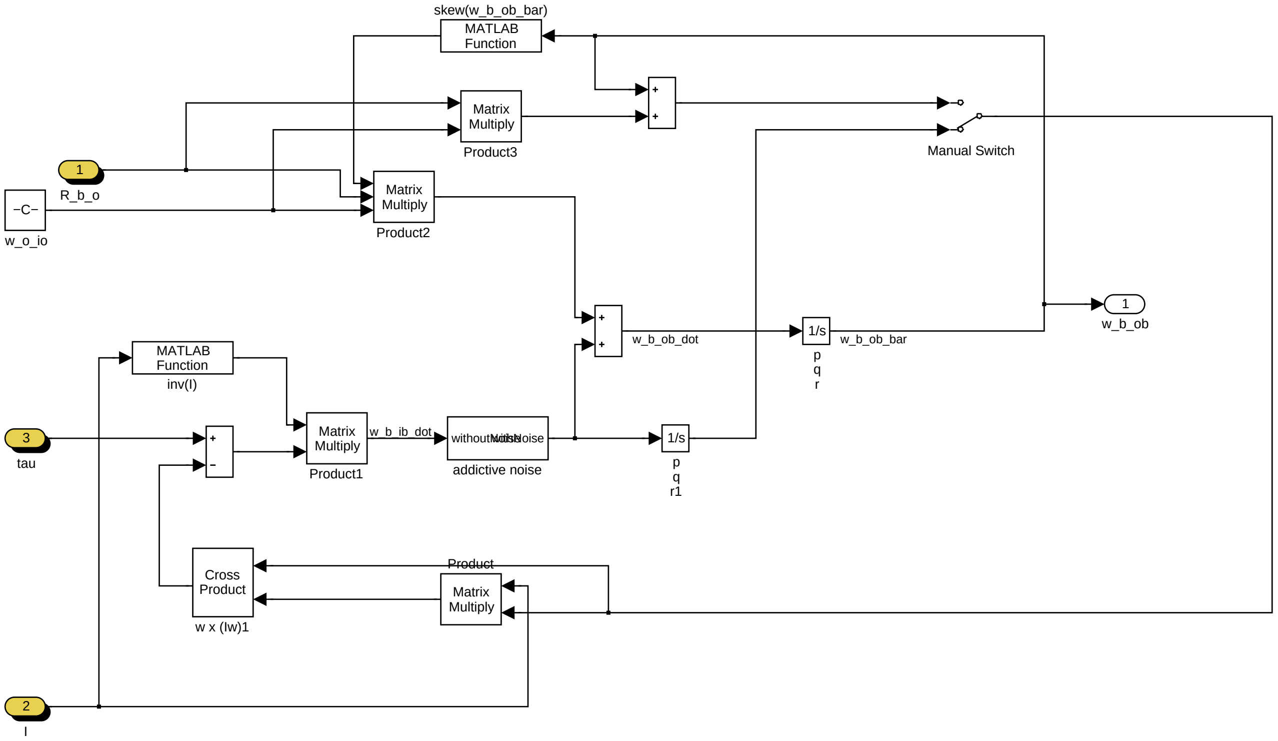


Figure B.3: The satellites dynamic.

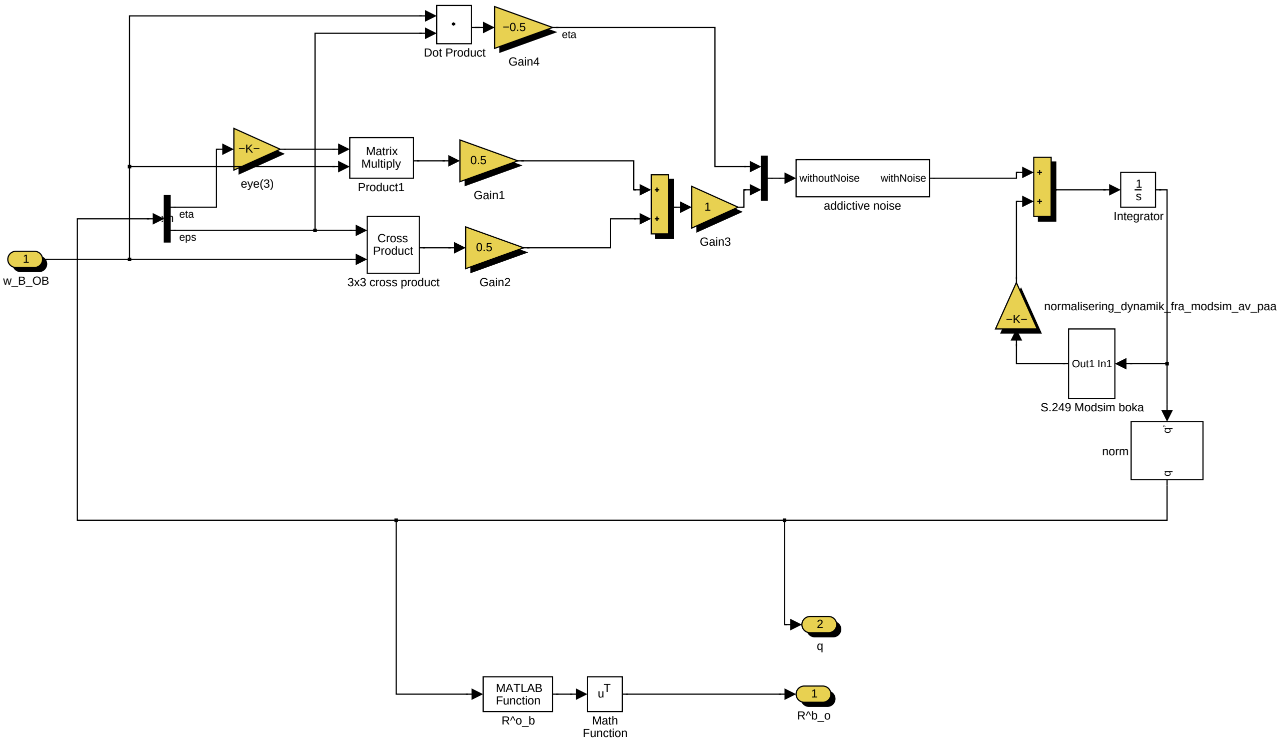


Figure B.4: The satellite Kinematics

The Sensor implementation

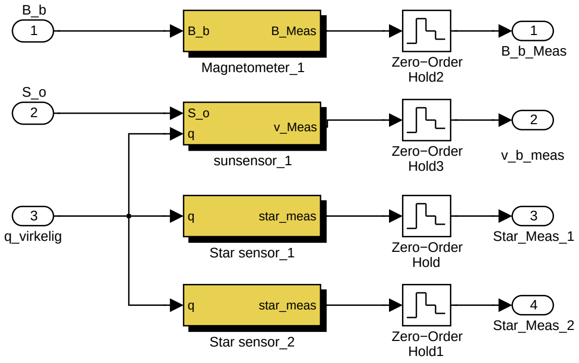


Figure B.5: The sensor block

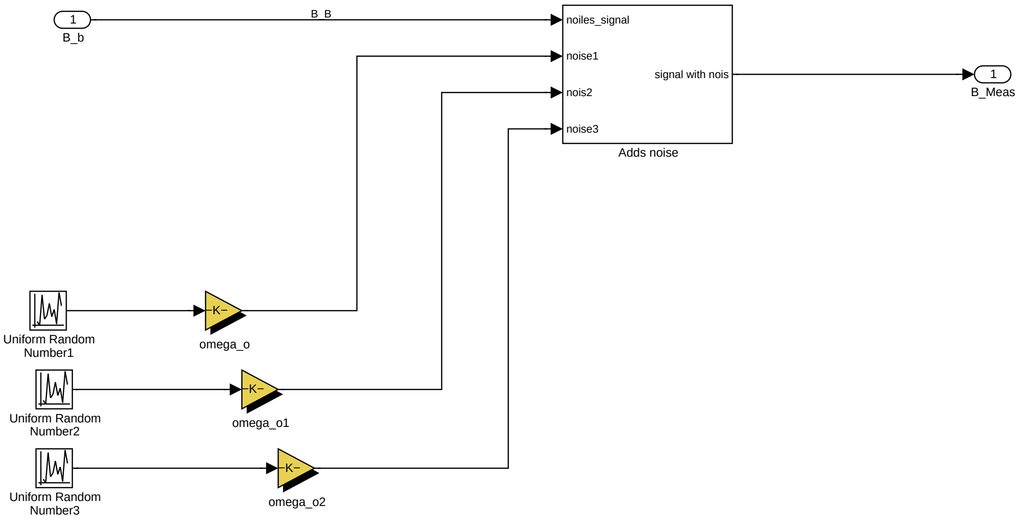


Figure B.6: The magnetometer

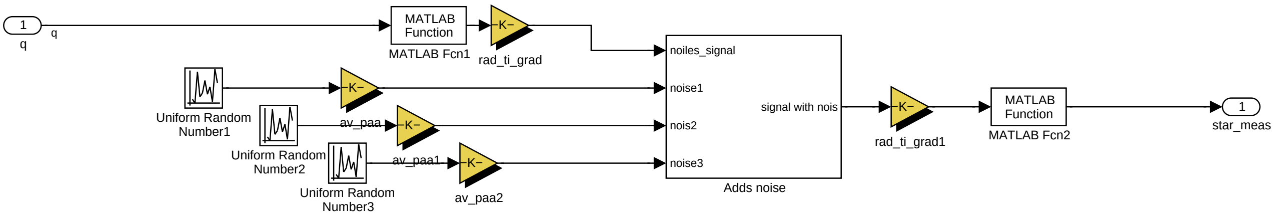


Figure B.7: The Star sensors

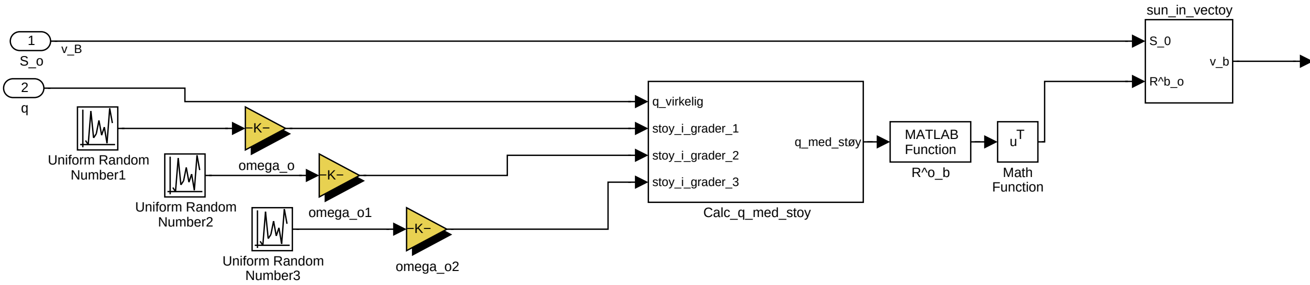


Figure B.8: The sun sensor

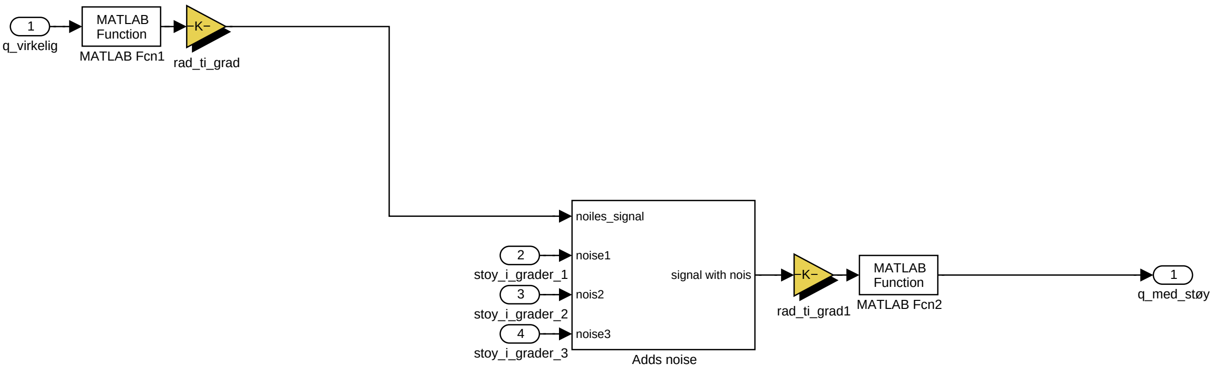


Figure B.9: Calculating the sun sensor noise

The Kalman filter block

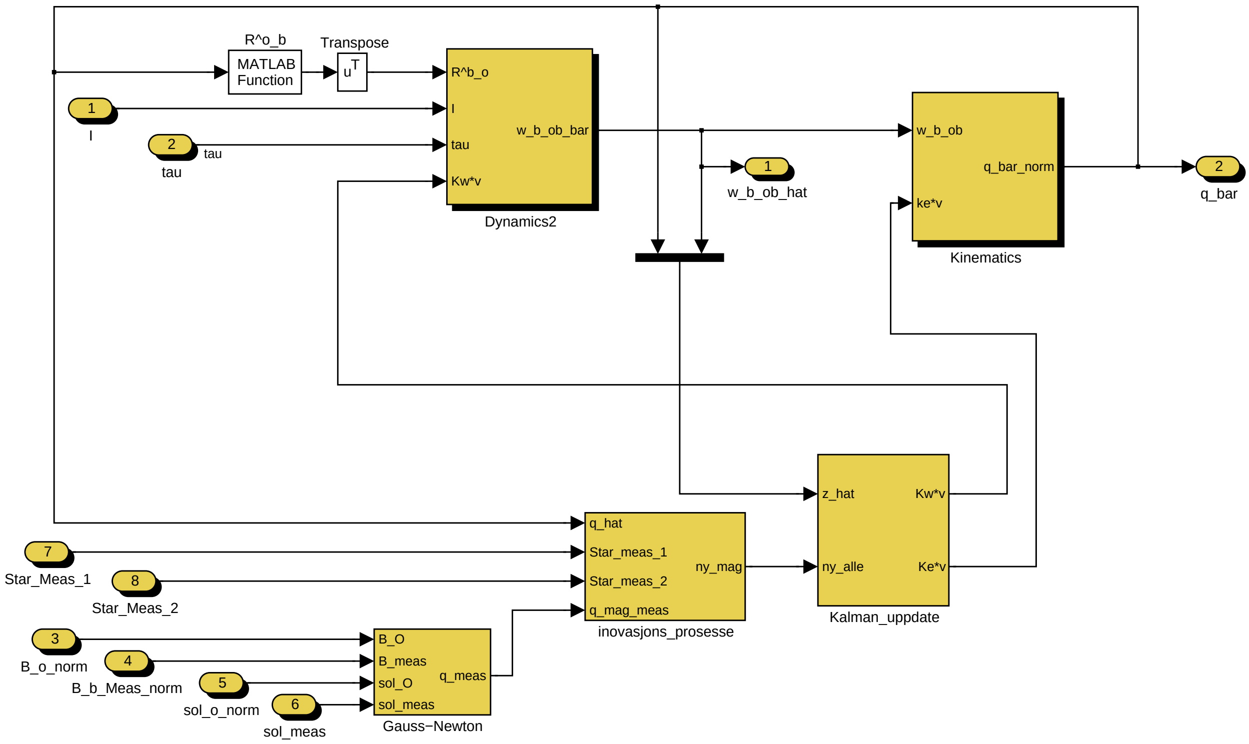


Figure B.10: The Kalman filter block

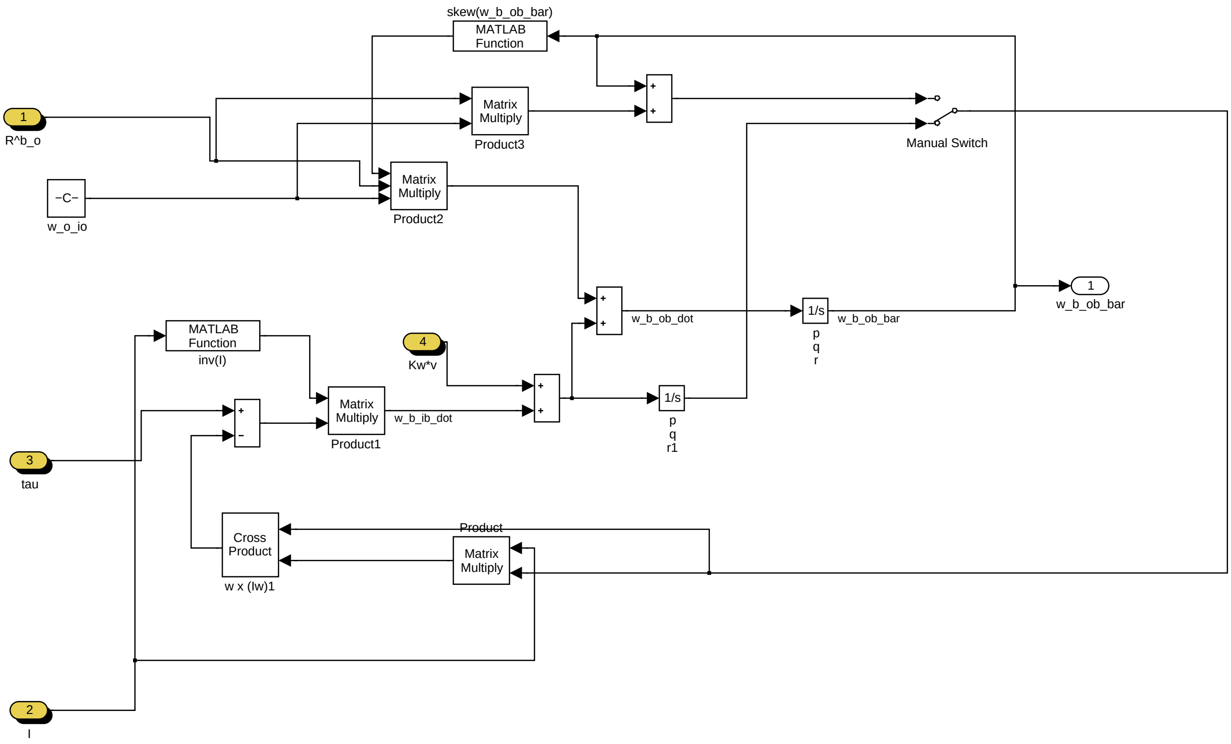


Figure B.11: The satellite dynamic in the Kalman filter

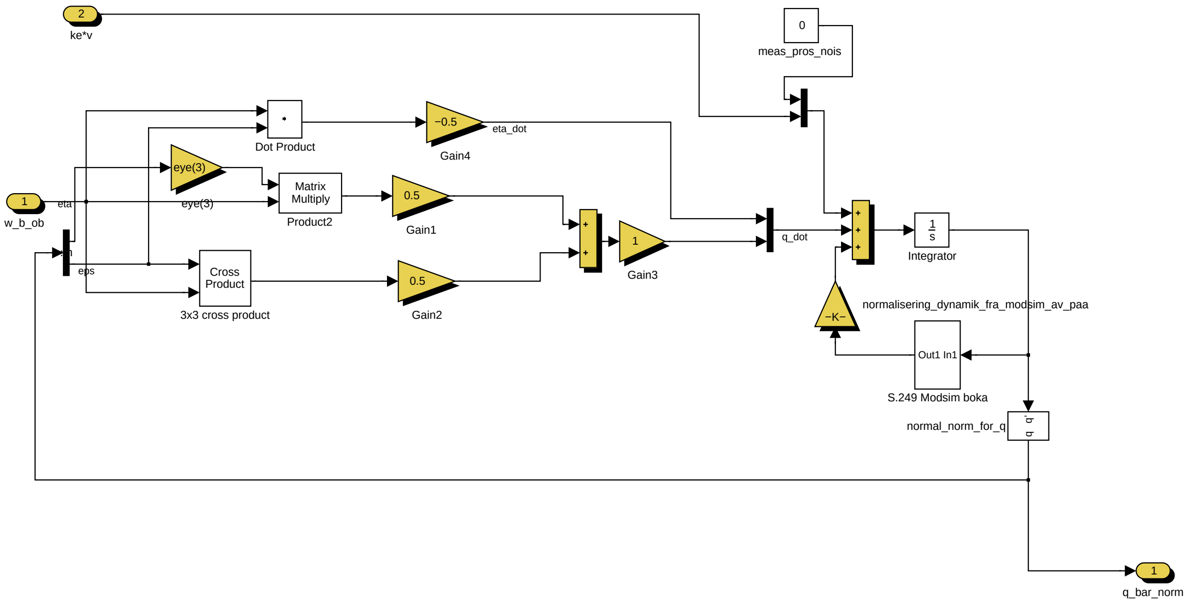


Figure B.12: The satellite kinematic in the Kalman filter

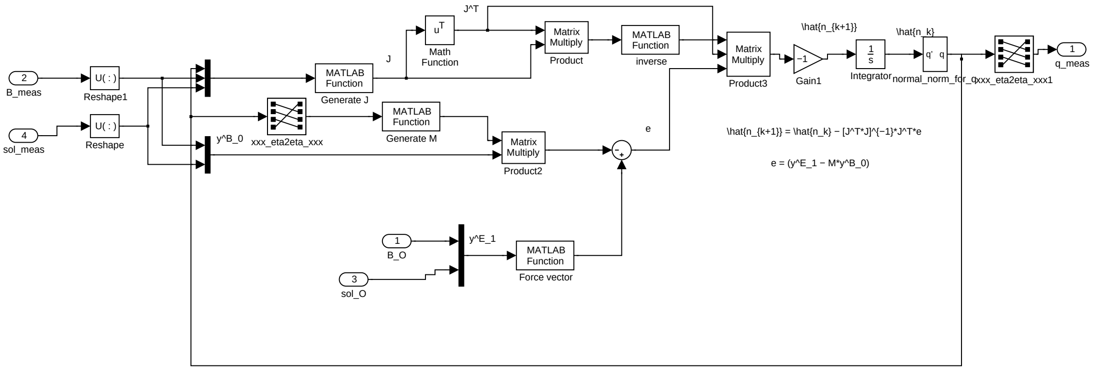


Figure B.13: The continuous Gauss-Newton, merging the magnetometer an sun sensor measurements

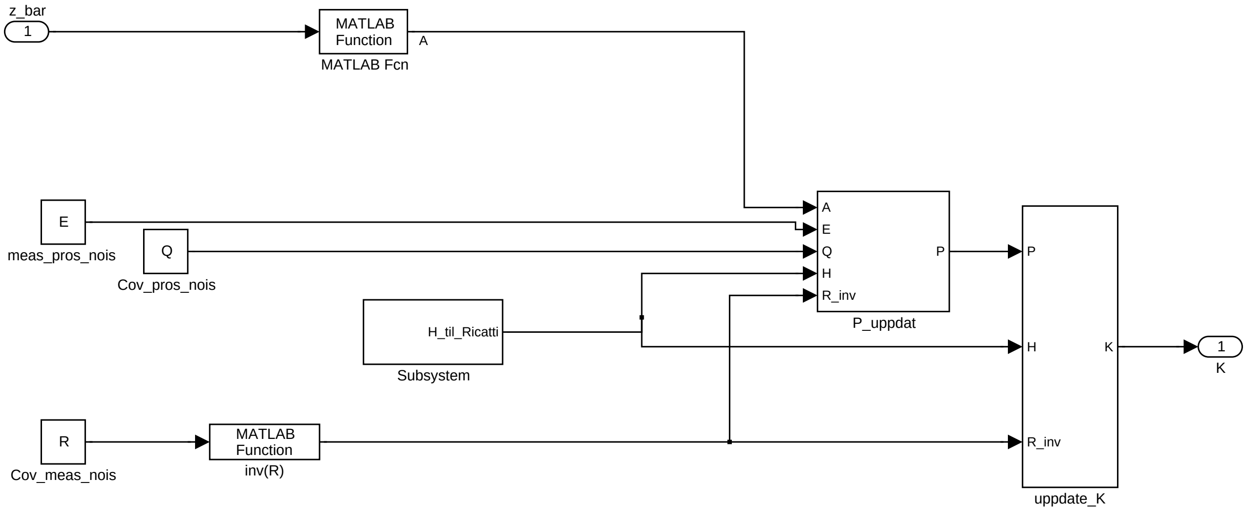


Figure B.14: Kalman gain calculation

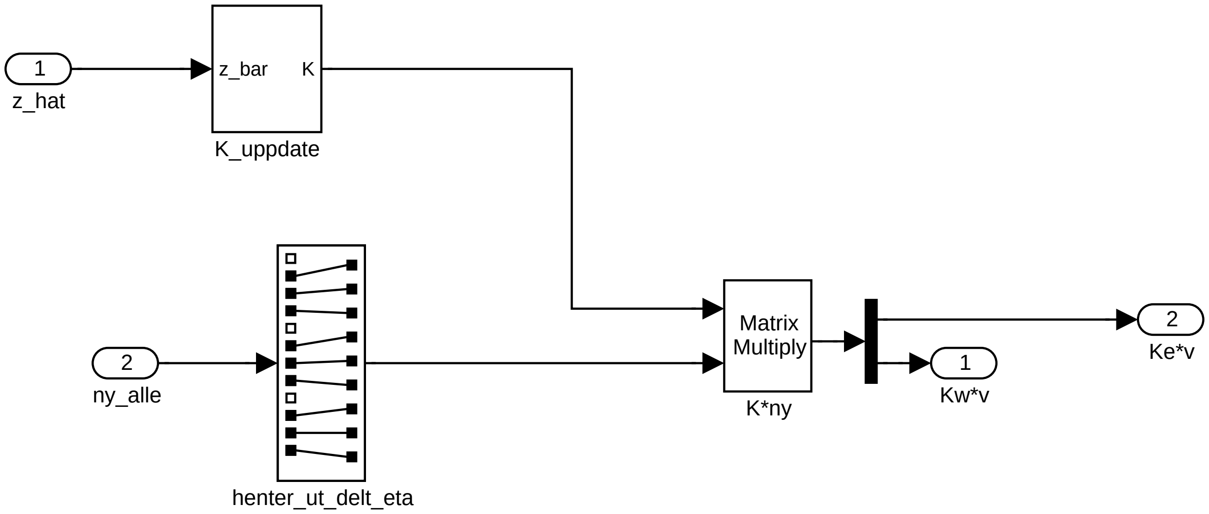


Figure B.15: Calculation the correction term

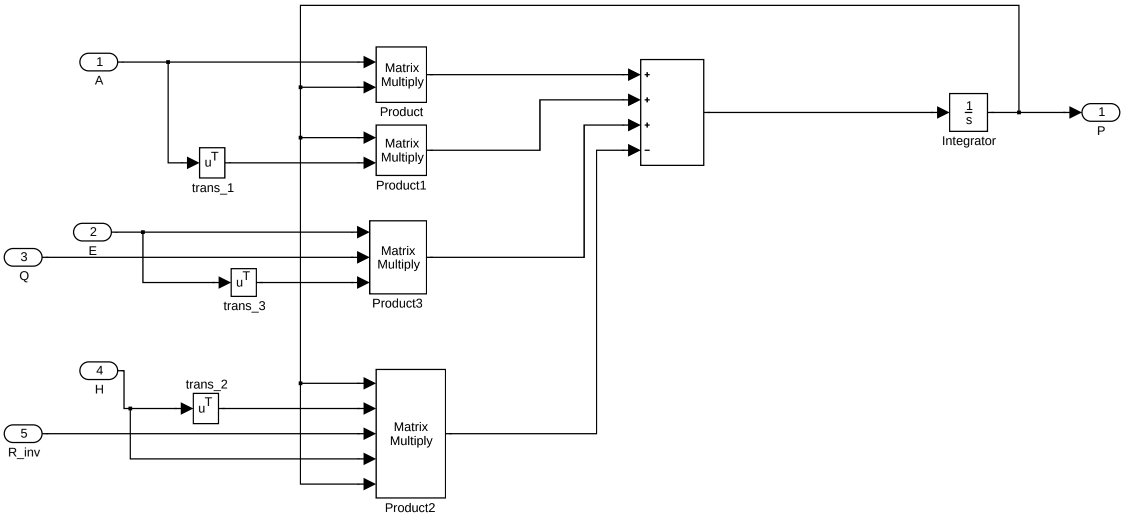


Figure B.16: The Riccati equation

Appendix C

The attached compact disc.

The cd contains the files used in this thesis. The name of the files and a description is given in the table on the next page.

File name	Description
start_bare_sol.m	initiates the simulation parameters for the filter in section 8.2
start_mag_og_sol_gaus.m	initiates the simulation parameters for the filter in section 8.2.1
start_bare_stjerne.m	initiates the simulation parameters for the filter in section 8.3
start_alle_gaus.m	initiates the simulation parameters for the filter in section 8.4
start_alle_gaus_presse_frekvens.m	initiates the simulation parameters for the filter in section 8.5
initSatelliteExt_svein_2.m	initiates the the satellite parameters
initSatelliteExt_svein_2_hoy_ang_velocity.m	initiates the the satellite parameters with and high angular velocity
fault.m	initiates the fault tolerance used in the fault detection
presse_freq.m	sets the star samplings frequency to 0.27
plott_resultat_bare_sol_gn.m	plots the simulation results for the filter in section 8.2
plott_resultat_sol_og_mag_ufeed.m	plots the simulation results for the filter in section 8.2.1
plott_resultat_sol_og_mag_mfeed.m	plots the simulation results for the filter in section 8.2.1
plott_resultat_stjerne_ufeed.m	plots the simulation results for the filter in section 8.3.2
plott_resultat_stjerne_mfeed.m	plots the simulation results for the filter in section 8.3.3
plott_resultat_alle_ufeed.m	plots the simulation results for the filter in section 8.4.1
plott_resultat_alle_mfeed.m	plots the simulation results for the filter in section 8.4.2
plott_resultat_fault.m	plots the simulation results for the filter in section 8.4.4
plott_resultat_alle_presse_Hz.m	plots the simulation results for the filter in section 8.5.2
linear_system.m	calculates $\boldsymbol{A} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T}(\hat{\boldsymbol{x}}, \boldsymbol{u})$
JGN.m	calculates the Jacobian in the Gauss-Newton method
JGN_bare_sol.m	calculates the Jacobian in the Gauss-Newton method for one vector measurement
MGN.m	calculates the transformation matrix used in the Gauss-Newton method
skew.m	calculates the skew symmetric matrix of a vector
euler2q_with_control.m	calculates the unit quaternions q from Euler angles and compeers it with a other quaternion
simplepropagator.m	calculates the orbit, with the simple orbit estimator
gravitytorque.m	calculates the gravity torque
magneticField.m	calculates the magnetic field in orbit(B^o)
Rquat	computes the quaternion rotation matrix
euler2q	computes the unit quaternions q from Euler angles
q2euler.m	computes the Euler angles from the unit quaternions q
igrf2000.m	calculates the IGRF model
Sattmodell_alle_gaus_disk.mdl	the simulink model for the Kalman filter with one star sensor, one magnetometer and two sun sensors.
Sattmodell_sol_gaus_plott.mdl	the simulink for the Kalman filter with one star sensor
Sattmodell_alle_gaus_disk_fault.mdl	the simulink for the Kalman filter with all the sensors and fault detection
Sattmodell_stjerne_med_disk.mdl	the simulink for the Kalman filter with two sun sensors
Sattmodell_mag_og_sol_gaus_disk.mdl	the simulink for the Kalman filter with one star sensor and one magnetometer
maketable.mdl	generates the magnetic field in the satellite orbit

Table C.1: The m files on the cd rom