Integrated Attitude Determination System Using a Combination of Magnetometer and Horizon Sensor Data

E. Maass

Thesis presented in partial fulfilment
of the requirements for the degree of
Master of Science in Electronic Engineering
at the
University of Stellenbosch

Study Leader: **Mr. J. Treurnicht**

December 2003

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature:

Date:

Abstract

A different approach of employing attitude sensors with incomplete measurements in an attitude determination system is investigated. The amount of available attitude sensors on small satellites are limited, and the failure of sensors can be fatal when accurate attitude determination is necessary. The problem with sensors with incomplete measurements is that they must be used in combination with other sensors to obtain three dimensional attitude information. The aim is to enhance the possible number of sensor combinations that can be employed, in an attempt to improve the ability of the attitude determination system to tolerate sensor failures.

An alternative sensor structure consisting of a magnetometer and two horizon sensors is presented. A method to obtain vector observations of the attitude from a combination between magnetometer and horizon sensor measurements is derived and tested. A full state Extended Kalman Filter is used to determine the satellite's attitude, attitude rate and disturbance torque from these vector observations.

A second Extended Kalman Filter structure, using only magnetometer measurements, is implemented. The magnetometer Extended Kalman Filter and the horizon/magnetometer Extended Kalman Filter are integrated to obtain a single Extended Kalman Filter structure to determine the satellite's full attitude state. Integration is done by switching between the different pairs of vector information. A systematic analysis of the integrated filter's dynamic behaviour during the switching stages is done by means of a series of case studies.

Opsomming

Die gebruik van oriëntasiesensore met onvolledige metingsdata in oriëntasiebepalingsstelsels is ondersoek. Slegs 'n beperkte aantal oriëntasiesensore is beskikbaar op mikro satelliete. 'n Foutiewe sensor kan dus noodlottig wees wanneer akkurate oriëntasiebepaling nodig is. Die probleem met sensore met onvolledige metingsdata is dat dit in sensor kombinasies gebruik moet word om drie dimensionele oriëntasieinligting te verkry. Die doel is dus om die moontlike aantal sensor kombinasies sodanig te vermeerder dat die oriëntasiebepalingsstelsel beter bestand sal wees teen moontlike sensor falings.

'n Alternatiewe sensor struktuur, bestaande uit 'n magnetometer en twee horison sensore, is ondersoek. 'n Metode vir die verkryging van 3-as oriëntasie inligting vanaf 'n kombinasie van magnetometer en horison sensor metingsdata is afgelei en getoets. 'n Vol toestand uitgebreide Kalmanfilter is gebruik om the satelliet se oriëntasie, oriëntasie snelheid en versteurings draaimoment vanaf die vektor observasies af te lei.

'n Tweede uitgebreide Kalmanfilter struktuur, wat slegs magnetometer metingsdata gebruik, is geïmplementeer. Die magnetometer filter en die horison/magnetometer filter is geïntegreer sodat een uitgebreide Kalmanfilter struktuur volle oriëntasie inligting kan aflei vanaf verskillende pare vektors met oriëntasie inligting. Integrasie is gedoen deur te skakel tussen die verskillende vektorpare. 'n Sistematiese analise van die geïntegreerde filter se dinamiese gedrag gedurende die oorskakelingstye is gedoen deur middel van 'n reeks gevallestudies.

Acknowledgements

I would like to thank the following for their help, support and encouragement:

- My study leader, Mr. J. Treurnicht, for his patience and advice.
- Everyone in the ESL lab that are responsible for the smooth running of the lab. Their work is essential to create a comfortable working environment for the students.
- Japie Engelbrecht, from Sunspace, for his time and help.
- My colleague and friend, Jacques Rossouw, for his friendship and support during the long years of study.
- Hanelle Fourie, Nicelle Bresler and Marius Theunissen, for proofreading this thesis.
- My family and friends. Without their love, support and prayers this thesis would not be possible.

Contents

List of Abbreviations

EKF	Extended Kalman Filter
IGRF	International Geomagnetic Reference Field
DCM	Direction Cosine Matrix
LOS	Line of Sight
FOV	Field of View
RPY	Roll, pitch and yaw angles
MOI	Moment of Inertia
rpo	resolutions per orbit
M/HM	Magnetometer to Horizon/Magnetometer combination
HM/M	Horizon/Magnetometer combination to Magnetometer

List of Symbols

Coordinate frames:

x_b, y_b, z_b	Satellite body coordinate axes
x_o, y_o, z_o	Orbit coordinate axes
x_i, y_i, z_i	Inertially fixed coordinate axes
x_c, y_c, z_c	Geocentric Inertial coordinate axes

Orbit and space environment:

a	Semimajor axis of elliptical orbit
b	Semiminor axis of elliptical orbit
h_P	Perigee height
h_A	Apogee height
v	Orbit true anomaly
M	Orbit mean anomaly
M_o	Orbit mean anomaly at epoch
e	Orbit eccentricity
i	Orbit inclination
Ω	Right ascension of the ascending node of an orbital plane
ω	Argument of perigee
X_c, Y_c, Z_c	Satellite position in celestial coordinates
r	Distance from barycentre to orbit
P	Orbital period at Δt
t	time
q	Distance from barycenter to the orbit at perigee

ω_o	Mean orbital angular rate
$\tilde{\omega}_o$	True orbit angular rate
$GM_\oplus$	Earth gravitational constant
$R_\oplus$	Radius of the Earth
R_s	Geocentric orbital radius
ρ_a	Atmospheric density
$\mathbf{c}_p$	Vector between centre of mass and centre of pressure
d_o	Solar radiation constant
c	Velocity of light
V	Magnitude of spacecraft velocity vector
$\mathbf{V}$	Spacecraft velocity unit vector
A	Total projected area of spacecraft

Satellite Models

Magnetometer:

B_{cx}, B_{cy}, B_{cz}	Geomagnetic field components in Celestial Coordinates
B_r, B_θ, B_ϕ	Geomagnetic field components in local tangent coordinates
B_{ox}, B_{oy}, B_{oz}	Geomagnetic field components in orbit coordinates
B_{bx}, B_{by}, B_{bz}	Geomagnetic field components in body coordinates
δ	Declination
α	Right ascension
α_G	Right ascension of the Greenwich Meridian
θ	Coelevation
ϕ	East longitude from Greenwich

Horizon:

δ	Mounting angle of horizon sensor
$\mathbf{v}_{\text{bore}}$	Horizon sensor boresight vector in body coordinates
$\mathbf{v}_{\text{min}}$	Horizon sensor minimum field of view vector in body coordinates
$\mathbf{v}_{\text{max}}$	Horizon sensor maximum field of view vector in body coordinates
$(\mathbf{v}_o)_{\text{bore}}$	Horizon sensor boresight vector in orbit coordinates
$(\mathbf{v}_o)_{\text{min}}$	Horizon sensor minimum field of view vector in orbit coordinates

$(\mathbf{v}_o)_{\text{max}}$	Horizon sensor maximum field of view vector in orbit coordinates
$(\mathbf{v}_{ox})_{\text{bore}}$, $(\mathbf{v}_{oy})_{\text{bore}}$, $(\mathbf{v}_{oz})_{\text{bore}}$	Horizon sensor boresight vector components in orbit coordinates
ρ	Angular radius of the earth
λ	Geocentric latitude of observer's position
d	Distance from the centre of the earth to the observer
R	Distance from the centre of the earth to the subobserver
a	Radius of the earth at the equator
Ψ	Azimuth angle of the horizon vector in local tangent coordinates
f	Ellipticity factor
b	Boresight angle
$\mathbf{n}$	Nadir unit vector
α	Angle at centre of the earth between a planet and the sun
$\Delta\xi$	Correction term in the dark angle
ξ	Dark angle

Operators:

$N\{\}$	Noise system properties
$E\{\}$	Expected value

Satellite's attitude plus rate parameters:

ϕ	Yaw angle
θ	Pitch angle
ψ	Roll angle
$\mathbf{A}$	Attitude transfromation matrix (DCM)
A_{ij}	Component of matrix **A** at row i and column j
$\mathbf{q}$	Attitude quaternion vector (orbit referenced)
q_1, q_2, q_3, q_4	Quaternion components (orbit referenced)
Φ	Euler rotation angle
e_x, e_y, e_z	Euler axis components in orbit referenced coordinates

$\boldsymbol{\omega}_B^I$	Inertially reference body angular rate vector
$\omega_x, \omega_y, \omega_z$	Inertially referenced body rate components
$\boldsymbol{\omega}_B^O$	Orbit referenced body angular rate vector
$\omega_{ox}, \omega_{oy}, \omega_{oz}$	Orbit referenced body rate components
$\boldsymbol{\Omega}$	Angular rate matrix
$\mathbf{L}$	Total satellite angular momentum vector
$\mathbf{h}$	Reaction wheel angular momentum vector

Satellite MOI values:

$\mathbf{1}$	Identity matrix
$\mathbf{I}$	Moment of inertia tensor
I_{xx}, I_{yy}, I_{zz}	Principal satellite body axis moment of inertias
I_T	Transverse moment of inertia

Satellite torques:

$\mathbf{N}_{GG}$	Gravity gradient torque vector
$\mathbf{N}_M$	Magnetic torque vector
$\mathbf{N}_D$	External disturbance torque vector
n_{doy}	Disturbance torque around the orbital Y_o-axis
$\mathbf{N}_{AERO}$	Aerodynamic torque vector
N_{SOLAR}	Magnitude of torque caused by solar radiation pressure

Control system plus estimator parameters:

$\hat{\boldsymbol{\omega}}_B^I$	Estimated Inertially referenced body angular rate vector
$\hat{\mathbf{q}}$	Estimated attitude quaternion vector
$\hat{n}_{doy}$	Estimated disturbance torque
$\hat{\omega}_x, \hat{\omega}_y, \hat{\omega}_z$	Estimated inertially referenced body angular rate components
T_s	Discrete sampling period
$\mathbf{K}$	Kalman filter gain matrix
$\mathbf{Q}$	State weighting matrix or System noise covariance matrix

$\mathbf{R}$	Control weighting matrix or Measurement noise covariance matrix
$\mathbf{H}$	System Hamiltonian
$\mathbf{v}$	Measurement vector
$\mathbf{v}_{meas}$	Sensor measurement vector in body coordinates
$\mathbf{v}_{body}$	Modelled measurement vector in body coordinates
$\mathbf{v}_{orb}$	Modelled measurement vector in orbit coordinates
$\mathbf{s}$	System noise vector
$\mathbf{m}$	Measurement noise vector
$\mathbf{f}(\mathbf{x}(t), t)$	Non-linear continuous system model
$\mathbf{f}_k(\mathbf{x}_k, k)$	Non-linear discrete system model
$\mathbf{F}(\mathbf{x}(t), t)$	Linearized perturbation system model
$\mathbf{\Phi}$	Discrete system matrix
$\mathbf{x}_k$	Discrete state vector at sample k
$\delta\mathbf{x}_k$	Discrete state perturbation vector at sample k
$\hat{\mathbf{x}}_k$	Estimated state vector at sample k
$\mathbf{P}_k$	Discrete state covariance matrix at sample k
$\mathbf{h}(\mathbf{x}(t), t)$	Non-linear continuous output model
$\mathbf{H}_k$	Discrete output measurement matrix at sample k
$\mathbf{e}_k$	Discrete innovation error vector at sample k
$\mathbf{B}_{meas}$	Measured Geomagnetic field vector in body coordinates
$\mathbf{B}_{orb}$	Modelled Geomagnetic field vector in orbit coordinates
$\mathbf{A}_{meas}$	Direction Cosine matrix from measured Euler angles

List of Conventions

arctan4	four quadrant arctan

$$\text{arctan4}\left(\frac{y}{x}\right) = \begin{cases} \arctan\left(\frac{y}{x}\right) & x > 0 \\ \pi + \arctan\left(\frac{y}{x}\right) & x < 0,\ y \geq 0 \\ -\pi + \arctan\left(\frac{y}{x}\right) & x < 0,\ y < 0 \\ \frac{\pi}{2} & x = 0,\ y \geq 0 \\ -\frac{\pi}{2} & x = 0, y < 0 \end{cases}$$

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

In this thesis the following will be addressed:

1. A different approach of employing sensors with incomplete measurements in an attitude determination system. Sensors with incomplete measurements must be used in combination with other sensors in order to obtain three dimensional attitude information. A possible combination between a magnetometer and horizon sensors will be presented. An Extended Kalman Filter (EKF) design will be used to determine the full satellite state from these sensor combination measurements. Flexible sensor combinations will enable satellite control systems to recover more effectively from sensor failures, especially on small satellites with a limited amount of attitude sensors.

2. Different sensor structures will be integrated to obtain a single EKF structure for attitude determination. Integration will be done by switching between the different vector observations from the sensor combinations. The dynamic behaviour of the EKF during these switching stages will be investigated through simulation.

The aim is to use the presented sensor structures in future studies to improve the fault tolerance of attitude determination systems.

1.2 Background

Dr. W.H. Steyn (Steyn [7]) presented a new EKF design that used vector observations to obtain full attitude, attitude rate and disturbance torque information of a satellite. Various sensors can be employed to measure the attitude of a satellite. A magnetometer measures the magnitude and direction of the geomagnetic field of the earth. Comparing these measurements with a model of the earth's geomagnetic field, full attitude data can be acquired. The same can be done by using a star camera and a star catalogue. Both these sensors are easy to implement in a vector based EKF as both provide 3-axis attitude information.

Horizon sensors and sun sensors on the other hand can only provide attitude information in one axis. By placing the sensors perpendicular to each other, 3-axis attitude data can be obtained. The question is how to combine these measurements so that full attitude knowledge can be determined by the EKF. The method used by Steyn [7] was to iteratively update the filter with the measurements from the horizon and sun sensors. Each sensor provides additional attitude information due to its position, and will therefore improve the estimation error during each update. This method proved to be very accurate, with attitude accuracies of below 0.1^o expected.

The problem, however, is that small satellites can only carry a limited number of backup sensors. The consequence of this is that if a sensor (e.g a horizon or sun sensor) fails to supply valid measurements, the particular EKF will not be able to determine the full satellite attitude state.

This thesis will look at a way to compensate for such sensor failures by investigating additional sensor combinations where measurements from various sensors, with complete and incomplete measurements, can be used to provide a measurement vector to the EKF. Two combinations will be addressed in this study: one that uses only magnetometer measurements and one that uses a combination of two horizon sensors and a magnetometer.

One is faced with two important issues when implementing the proposed magnetometer and horizon sensor combination. The first of these is the difference in the nature of the measurements from each sensor. The magnetometer measurement is a vector measurement of the earth's magnetic field, while the horizon sensor provides an angle measurement of the horizon's elevation angle. The second problem is the difference in accuracy between the sensors. One advantage of horizon sensors is the accuracy of its measurements. The inaccuracy of the magnetometer may therefore decrease the precision of the EKF to a degree that the desired requirements, as specified by Steyn [7, page 1-2], cannot be met by the attitude determination system.

The integration, by means of switching, of separately implemented EKF systems, will also be investigated. The different vector observations are obtained from sensor measurements and used as input to a single EKF structure. The aim is to examine the dynamic behaviour of the EKF during the switching stages. Two sensor combinations will be used to provide the measurement vectors. The first consists of only a magnetometer and the second of two horizon sensors and a magnetometer.

1.2.1 Satellite

The attitude determination system developed in this thesis assumed a near cubical micro satellite with a deployed boom and tip-mass to earth stabilize it. The satellite's body z_b-axis was nadir pointing (pointed towards the earth). The satellite was further kept at a slow z_b-spin during normal operation.

1.2.2 Contributions

The main contributions of this research are stated below:

- Full attitude information was determined by an EKF from vector observations obtained from a combination of magnetometer and horizon sensor measurements. The accuracy of the EKF, however, was poor due to calculation errors in the mathematical models.

- The dynamic behaviour of an integrated EKF system during the switching stages between different vector observations was investigated by means of simulation.

1.3 Thesis Layout

Chapter 2 will introduce the various coordinate systems used throughout this thesis and the mathematical models employed to describe the orbital motion of the satellite. The simulation models for the satellite's sensors will be presented. The satellite's dynamic and kinematic equations of motion, and various external disturbance torques will also be discussed.

Chapter 3 describes the implementation of two EKFs that will be used to determine the satellite's full attitude state from vector observations of two different sensor combinations. The first combination consist of measurements from a magnetometer and the second of measurements from two horizon sensors and a magnetometer. The EKFs will be implemented and tested separately through simulation.

In Chapter 4 these sensor combinations are integrated into a single EKF by switching between the different vector observations. The effect of the switching on the filter's performance will be investigated by means of simulation.

Chapter 5 will summarize and reflect on the results obtained in this thesis.

Chapter 2

Satellite Motion and Space Environment Simulation Model

2.1 Introduction

This chapter describe the mathematical models used to simulate the satellite's attitude dynamics, orbital motion and the space environment in which the satellite operates.

The mathematical models used to simulate the satellite's attitude dynamics are presented by the equations of motion as documented by Wertz [8, Chapter 16]. The equations of motion are divided into the *dynamic equations of motion*, which relates the time derivative of the angular momentum vector to the applied torque, and the *kinematic equations of motion*, which is the study of the satellite's motion irrespective of the forces that bring about the motion.

Simulation of the satellite's orbital motion and space environment consists of mathematical models of the sun's orbit, the satellite's orbit around the earth and the most important environmental disturbance torques acting on the spacecraft.

The necessary parameters to define an orbit are found in Wertz [8, page 42 - 47] and are summarized in Section 2.2.1. The orbit propagator's output consist of the satellite's position in space at time, t, in terms of its latitudinal and longitudinal components, and its distance from the earth. The output data is used to model the earth's geomagnetic field and horizon, which are necessary for the simulation of the magnetometer and horizon sensors respectively. The mathematical models for the magnetometer and horizon sensors are derived in Section 2.4.

Section 2.3 defines the coordinate systems used throughout this document. It also discusses the attitude parameterization methods used to present the spacecraft's orientation and to simplify transformations between the different coordinate systems.

2.2 Orbit Propagator

2.2.1 Satellite Orbit Propagator

The elements of an orbit are the parameters needed to fully specify the motion of the satellite. These elements define the size and shape of the orbit, the orbital plane and the rotation of the orbit.

- The *semimajor axis*, a, and the *eccentricity*, e, define the size and shape of the orbit. The semimajor axis, a, of an elliptical orbit is calculated from the perigee height, h_P, the apogee height, h_A, and the radius of the Earth, $R_\oplus$, as (See Figure 2.1):

  $$a = R_\oplus + \frac{(h_P + h_A)}{2}. \tag{2.1}$$

  The eccentricity, e, specifies the shape of an ellipse and is defined as the ratio between the semimajor, a, and semiminor, b, axes. For an ellipse; $0 \leq e \leq 0$, and for a circle; $e = 0$.

- The *inclination*, i, and the *right ascension of the ascending node*, Ω, define the orbital plane. The inclination, i, is the angle between the orbital plane and a reference plane i.e. the equatorial plane. The right ascension of the ascending node, Ω, is the angle in the equatorial plane measured eastward from the *vernal equinox*[1] to the ascending node[2] of the orbit.

- The rotation of the orbit within the plane is defined by the *argument of perigee*, ω, which is the angle at the barycenter, measured in the orbital plane in the direction of the satellite's motion from the ascending node to perigee.

Once the orbit is fully defined, the satellite's exact coordinates in the celestial coordinate system at time, t, can be calculated as (See Wertz [8, page 135]):

$$X_c = r\left[\cos(\omega + v)\cos\Omega - \sin(\omega + v)\sin\Omega\cos i\right], \tag{2.2}$$

$$Y_c = r\left[\cos(\omega + v)\sin\Omega + \sin(\omega + v)\cos\Omega\cos i\right], \tag{2.3}$$

$$Z_c = r\left[\sin(\omega + v)\sin i\right], \tag{2.4}$$

where the *true anomaly*, v, is an indication of where the satellite is in its orbit at time, t, and is defined as the angle measured at the barycenter between the perigee point and the satellite. Thus,

$$v \approx M + 2\,e\sin M + \frac{5}{4}\,e^2\sin 2M, \tag{2.5}$$

where M is the *mean anomaly* defined as $360 \cdot (\Delta t/P)$ with P the orbital period and Δt the time since the satellite passed perigee. The *radius*, r, is the distance from the barycenter to the orbit at any time and can be calculated as:

$$r = q\left(\frac{1+e}{1+e\cos v}\right), \tag{2.6}$$

where q is the distance from the barycenter to the orbit at perigee (perifocal distance).

The satellite's latitude, longitude and altitude can be calculated by converting Equations 2.2 - 2.4 to spherical coordinates.

[1] The point where the *ecliptic*, or plane of the earth's orbit about the sun, crosses the equator going from south to north.

[2] For an earth satellite, the ascending node is the point in its orbit where a satellite crosses the equatorial plane going from south to north.

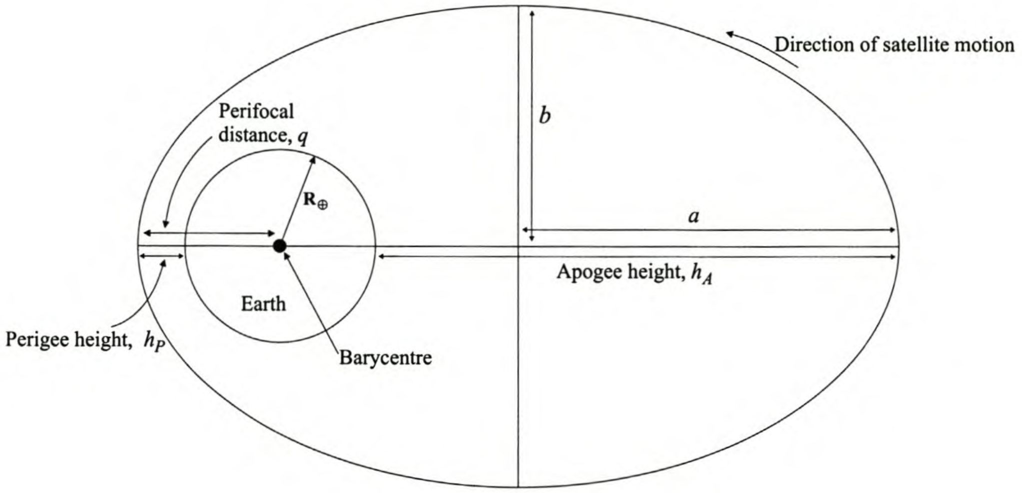


Figure 2.1: *Orbit terminology for an elliptical orbit.*

2.2.2 Sun Orbit Propagator

The parameters defined in Section 2.2.1 can also be used to model the orbit of the sun as the earth revolves around its own axis. The sun's coordinates in the celestial coordinates system can then be calculated from Equations 2.2 - 2.4.

2.3 Coordinate Systems

Three major coordinate systems are used to define the attitude of the satellite, that is the inertial, orbit and body coordinate systems. Wertz [8] made a further distinction between spacecraft-centered coordinate systems and nonspacecraft-centered coordinate systems.

2.3.1 Spacecraft-Centered Coordinate Systems

Three basic types of spacecraft-centered coordinates are defined by Wertz [8]:

- those fixed relative to the body of the spacecraft (body coordinates),
- those fixed in inertial space, and
- those defined relative to the orbit and not fixed relative to either the spacecraft or inertial space.

Body Coordinates

Body Coordinates are used to define the attitude of the satellite. It is also the system in which sensor measurements are made. The three components of the rectangular coordinate system will be presented by x_b, y_b and z_b. The body axes are defined as shown in Figure 2.2. The z_b-axis is parallel but opposite to the direction of boom deployment and the x_b and y_b axes are perpendicular to two of the side solar panels.

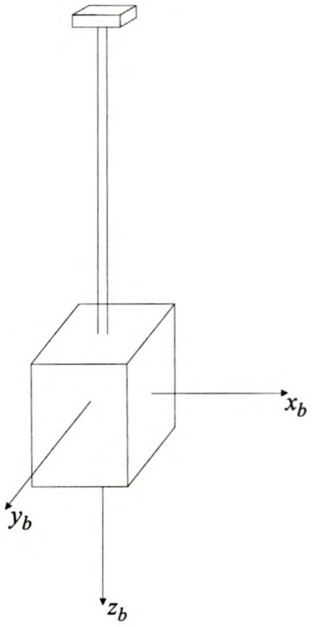


Figure 2.2: *Body Coordinate System.*

Orbit Coordinates

The Orbit-defined coordinate system maintains its orientation relative to the earth as the spacecraft moves in its orbit. The coordinates are defined as *roll*, *pitch* and *yaw* or *RPY* coordinates (see Figure 2.3), where the *yaw* axis or z_o-axis is directed towards nadir[3], the *pitch* axis or y_o-axis is directed towards the

[3] The centre of the earth.

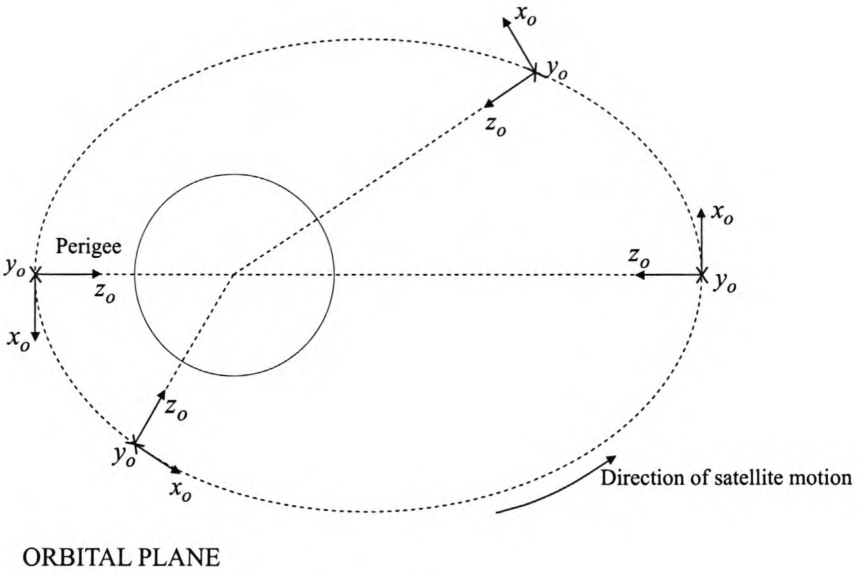


Figure 2.3: *The orbit coordinate system.*

negative orbit normal and the *roll* axis or x_o-axis completes the orthogonal set.

The attitude of the satellite can be defined by Euler angles. These angles are obtained from an ordered series of right hand positive rotations from the orbital axes to the body axes. This document used a Euler 3-2-1 sequence of rotations as shown by Figure 2.4. The first rotation is a yaw around the z_o-axis through an angle, ϕ. The second rotation is a pitch around the y'-axis through an angle, θ, and the last rotation is a roll around the x_b-axis through an angle, ψ. The corresponding transformation matrix, $\mathbf{A}$, also called the *direction cosine matrix* (DCM), for a full rotation from the reference orbital to the body coordinates is:

$$\mathbf{A} = \begin{bmatrix} C\theta C\phi & C\theta S\phi & -S\theta \\ -C\psi S\phi + S\psi S\theta C\phi & C\psi C\phi + S\psi S\theta S\phi & S\psi C\theta \\ S\psi S\phi + C\psi S\theta C\phi & -S\psi C\phi + C\psi S\theta S\phi & C\psi C\theta \end{bmatrix}, \tag{2.7}$$

where

C = cosine function, and

S = sine function.

Therefore,

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix}. \tag{2.8}$$

From Equation 2.7 the Euler angles can again be calculated as:

$$\text{roll}(\psi) = \arctan 4 \left[\frac{A_{23}}{A_{33}}\right], \tag{2.9}$$

$$\text{pitch}(\theta) = \arcsin\left[-A_{13}\right], \tag{2.10}$$

$$\text{yaw}(\phi) = \arctan 4 \left[\frac{A_{12}}{A_{11}}\right]. \tag{2.11}$$

Although the Euler angle presentation gives a clear physical interpretation of the roll, pitch and yaw angles, it suffers from singularities in the pitch angle, θ. This makes it undesirable to use in the control and estimation algorithms of this document. A better presentation which is more convenient to use for numerical computations would be the *Euler symmetric parameters*, q_1, q_2, q_3, q_4, which is defined as:

$$q_1 \equiv e_x \sin\frac{\Phi}{2}, \quad q_2 \equiv e_y \sin\frac{\Phi}{2}, \quad q_3 \equiv e_z \sin\frac{\Phi}{2}, \quad q_4 \equiv \cos\frac{\Phi}{2}, \tag{2.12}$$

where

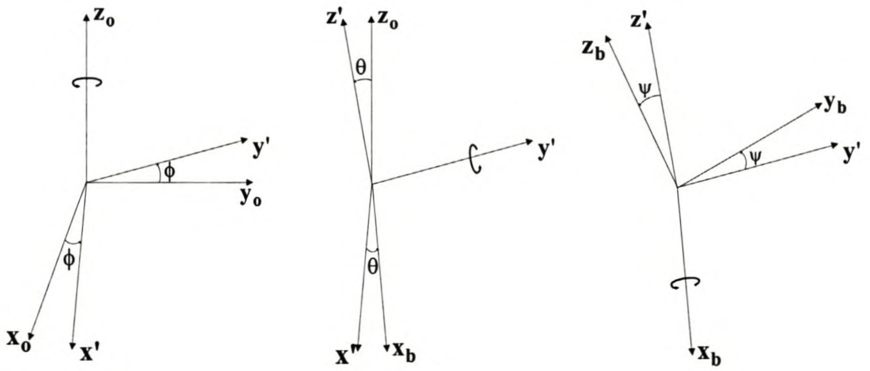


Figure 2.4: *3-2-1 Euler angle rotation.*

e_x, e_y, e_z = components of the unit Euler axis vector in orbit referenced coordinates, and

Φ = rotation angle around the Euler axis.

The Euler symmetric parameters (quaternions) are not independent, but satisfy the following constraint:

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1. \tag{2.13}$$

The direction cosine matrix expressed in quaternion form is:

$$\mathbf{A} = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix}. \tag{2.14}$$

From Equation 2.14 the corresponding quaternion elements can be calculated as[4]:

$$q_4 = \frac{1}{2}\left(1 + A_{11} + A_{22} + A_{33}\right)^{\frac{1}{2}}, \tag{2.15}$$

$$q_1 = \frac{1}{4q_4}\left(A_{23} - A_{32}\right), \tag{2.16}$$

$$q_2 = \frac{1}{4q_4}\left(A_{31} - A_{13}\right), \tag{2.17}$$

$$q_3 = \frac{1}{4q_4}\left(A_{12} - A_{21}\right). \tag{2.18}$$

Inertial Coordinates

This coordinate system is used as the reference frame for the motion of the satellite in inertial space. The spacecraft-centered inertial coordinate system used in this thesis was defined by Steyn [7, page 1-9] and is shown in Figure 2.5. The coordinate system coincides precisely with the orbit-defined coordinates at perigee. Since the orbital plane experiences a slow precession, this inertial coordinate system is not strictly inertial. However, since this precession is slow enough, it has a negligible effect on the dynamics of the satellite. The matrix used to transform

[4] Note that this is only one of four possible ways to calculate the quaternion elements. If q_4 becomes too small, q_1, q_2 or q_3 can be calculated from the DCM diagonal values, and used to calculate the remaining elements

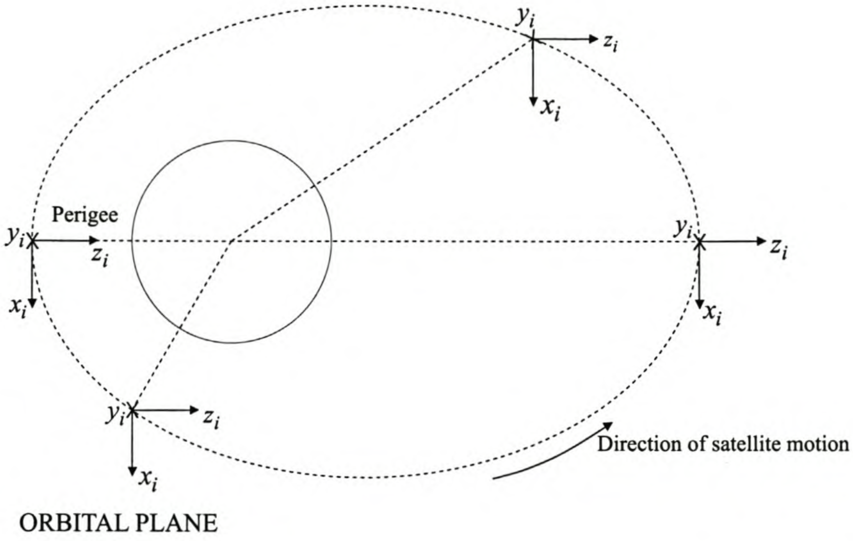


Figure 2.5: *The spacecraft-centered inertial coordinate system.*

from inertial to orbital coordinates is:

$$\begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} = \begin{bmatrix} \cos v(t) & 0 & \sin v(t) \\ 0 & 1 & 0 \\ -\sin v(t) & 0 & \cos v(t) \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}, \tag{2.19}$$

where

$v(t)$ = the true anomaly at time, t, and

t = time.

2.3.2 Nonspacecraft-Centered Coordinate Systems

Nonspacecraft-centered coordinate systems are convenient as a means of obtaining reference vectors such as the magnetic field vector or position vectors to objects seen by the spacecraft. Wertz [8] define a number of possible centre references resulting in different possible coordinate systems. In this document only the earth centered or *geocentric inertial coordinates* will be used.

Geocentric Inertial Coordinates

These coordinates are also known as the celestial coordinate system and is defined relative to the spinning axis of the earth as shown in Figure 2.6. The z_c-axis is

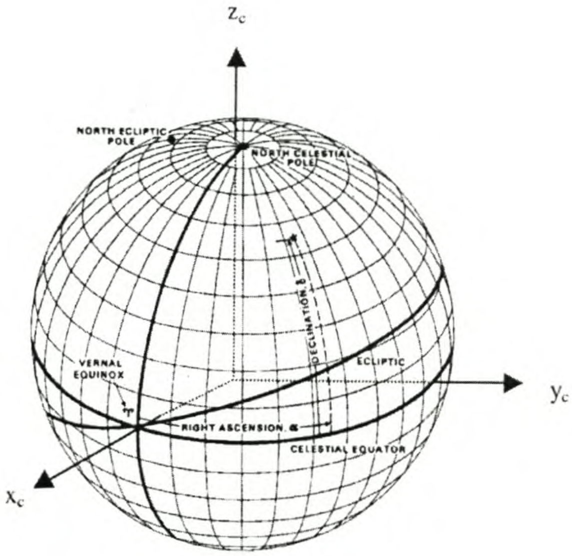


Figure 2.6: *The geocentric inertial coordinate system.*

parallel to the spinning axis of the earth with positive in the direction of the geometric north pole. The x_c-axis is parallel to the line connecting the centre of the earth and the vernal equinox[5]. The y_c-axis completes the orthogonal set.

2.4 Sensor Models

2.4.1 Magnetometer

The magnetometer measures the strength and direction of the geomagnetic field at the location of the satellite. By comparing the measurement with geomagnetic field models, full attitude data can be obtained. The following mathematical model is used to simulate the magnetometer measurement. The axes of the magnetometer are aligned with the satellite body axes, so the measured geomagnetic

[5] The point where the *ecliptic*, or plane of the earth's orbit about the sun, crosses the equator going from south to north.

field is in body coordinates.

1. Obtain the subsatellite latitude, longitude and geocentric distance from the orbit propagator in Section 2.2.1.

2. An eight order IGRF model is now used to calculate the geomagnetic field model in local tangent coordinates. See Appendix A for a detailed calculation.

3. The geomagnetic field vector in local tangent coordinates must be converted to the satellite body coordinates. The transformation consists of the following sequence of conversions:

   (a) local tangent coordinates to celestial coordinates:

   $$\begin{aligned} B_{cx} &= (B_r \cos\delta + B_\theta \sin\delta)\cos\alpha - B_\phi \sin\alpha, \\ B_{cy} &= (B_r \cos\delta + B_\theta \sin\delta)\sin\alpha + B_\phi \cos\alpha, \\ B_{cz} &= (B_r \sin\delta - B_\theta \cos\delta), \end{aligned} \tag{2.20}$$

   where $\delta = 90° - \theta$ is the declination and $\alpha = \phi + \alpha_G$ the right ascension. α_G is the right ascension of the Greenwich meridian at Greenwich. θ and ϕ is the coelevation and East longitude from Greenwich respectively.

   (b) celestial coordinates to orbit coordinates:
   An Euler angle rotation that consists of four rotations is used for the transformation from celestial coordinates to orbit coordinates. The transformation matrix, developed by Jacobs [9], is represented by:

   $$\mathbf{T} = \begin{bmatrix} C(-i)C(\Omega')C(-\omega')+S(\Omega')S(-\omega') & C(-i)C(\Omega')C(-\omega')-C(\Omega')S(-\omega') & -S(-i)C(-\omega') \\ -S(-i)C(\Omega') & -S(-i)S(\Omega') & -C(-i) \\ C(-i)C(\Omega')S(-\omega')-S(\Omega')C(-\omega') & C(-i)S(\Omega')S(-\omega')+C(\Omega')C(-\omega') & -S(-i)S(-\omega') \end{bmatrix}. \tag{2.21}$$

Therefore,

$$\begin{bmatrix} B_{ox} \\ B_{oy} \\ B_{oz} \end{bmatrix} = \mathbf{T} \begin{bmatrix} B_{cx} \\ B_{cy} \\ B_{cz} \end{bmatrix}, \tag{2.22}$$

with

$$\begin{aligned} \Omega' &= 90^\circ + \Omega, \\ \omega' &= \omega + \upsilon, \end{aligned} \tag{2.23}$$

where

B_{ox},B_{oy},B_{oz} = components of the geomagnetic field vector in orbit coordinates,

i = orbit inclination,

Ω = orbit right ascension of the ascending node,

ω = orbit argument of perigee, and

υ = orbit true anomaly.

(c) orbit coordinates to body coordinates:
The geomagnetic field vector is converted from orbit coordinates to body coordinates with the direction cosine matrix from Equation 2.7:

$$\begin{bmatrix} B_{bx} \\ B_{by} \\ B_{bz} \end{bmatrix} = \mathbf{A} \begin{bmatrix} B_{ox} \\ B_{oy} \\ B_{oz} \end{bmatrix}. \tag{2.24}$$

2.4.2 Horizon Sensor

Two CCD sensors with a field of view (FOV) of ±15^o each are used to obtain orthogonal measurements of the sunlit earth horizon. Figure 2.7 shows the placement of the sensors on the satellite. The sensors are mounted at an elevation angle of $\delta = 27.31^o$, which is equal to the nominal horizon angle at an altitude of 800 km. The -X-horizon sensor measures pitch angle rotations of the satellite, while the Y-horizon sensor measures roll angle rotations. The measurements are, however, only valid for valid fields of view (limited to ±15^o around the nominal horizon, δ) and valid horizon illumination. The mathematical model for the -X- and Y-horizon sensors is derived as follows:

1. Three unit vectors in body coordinates can be defined: one to describe the boresight of the sensor at angle, δ, and two to describe the FOV limitations in minimum, $\delta - 15^o$, and maximum, $\delta + 15^o$, FOV vectors.

   The three unit vectors for the -X-Horizon sensor are:

$$\mathbf{v}_{\mathrm{boreX}} = \begin{bmatrix} -\cos(\delta) \\ 0 \\ \sin(\delta) \end{bmatrix}, \tag{2.25}$$

$$\mathbf{v}_{\mathrm{minX}} = \begin{bmatrix} -\cos(\delta - 15^o) \\ 0 \\ \sin(\delta - 15^o) \end{bmatrix}, \tag{2.26}$$

$$\mathbf{v}_{\mathrm{maxX}} = \begin{bmatrix} -\cos(\delta + 15^o) \\ 0 \\ \sin(\delta + 15^o) \end{bmatrix}, \tag{2.27}$$

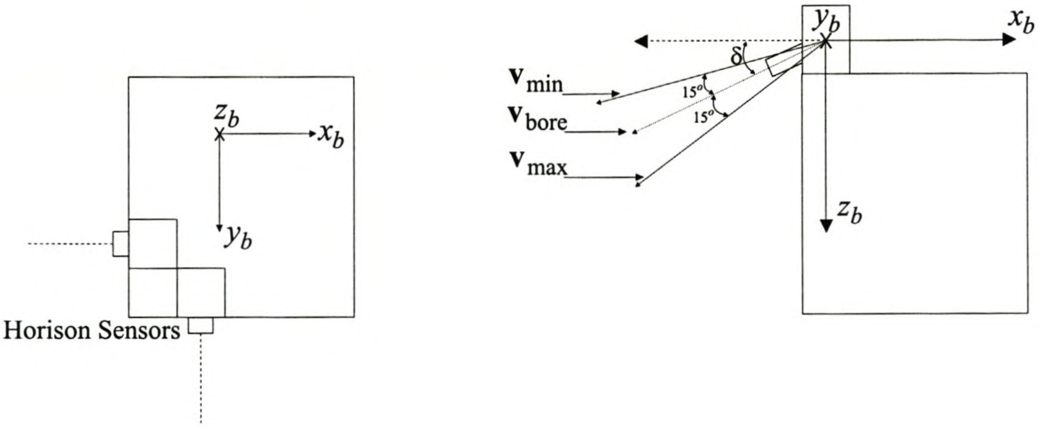

z_b
x_b
y_b
Horison Sensors
y_b
x_b
δ
$\mathbf{v}_{\mathrm{min}}$
15^o
$\mathbf{v}_{\mathrm{bore}}$
15^o
$\mathbf{v}_{\mathrm{max}}$
z_b

Figure 2.7: *Placement of the horizon sensors on the satellite.*

and for the Y-Horizon sensor:

$$\mathbf{v}_{\text{boreY}} = \begin{bmatrix} 0 \\ \cos(\delta) \\ \sin(\delta) \end{bmatrix}, \tag{2.28}$$

$$\mathbf{v}_{\text{minY}} = \begin{bmatrix} 0 \\ \cos(\delta - 15^o) \\ \sin(\delta - 15^o) \end{bmatrix}, \tag{2.29}$$

$$\mathbf{v}_{\text{maxY}} = \begin{bmatrix} 0 \\ \cos(\delta + 15^o) \\ \sin(\delta + 15^o) \end{bmatrix}. \tag{2.30}$$

2. Transform the boresight vector and maximum and minimum FOV vectors from body coordinates to orbital coordinates by using the transpose of the DCM, **A**:

$$\begin{aligned} (\mathbf{v}_o)_{\text{bore}} &= \mathbf{A}^T \mathbf{v}_{\text{bore}}, \\ (\mathbf{v}_o)_{\text{min}} &= \mathbf{A}^T \mathbf{v}_{\text{min}}, \\ (\mathbf{v}_o)_{\text{max}} &= \mathbf{A}^T \mathbf{v}_{\text{max}}. \end{aligned} \tag{2.31}$$

3. The azimuth of the boresight vector in orbit coordinates are obtained by:

$$Azim_{\text{bore}} = \arctan 4 \left(\frac{(v_{oy})_{\text{bore}}}{(v_{ox})_{\text{bore}}} \right). \tag{2.32}$$

4. The angular radius of an elliptical Earth, ρ, at the azimuth of the boresight vector, is calculated from the earth oblateness model as defined by Wertz [8, page 102, Equation 4-24]:

$$\begin{aligned} \rho &= \operatorname{arccot}\left\{ \left[\frac{(d^2 - R^2)}{a^2} \left(1 + \frac{(2-f) f R^2 \cos^2 \lambda}{(1-f)^2 a^2} \sin^2 \Psi \right) \right]^{\frac{1}{2}} \right. \\ &\quad \left. + \frac{(2-f) f R^2 \sin 2\lambda}{2(1-f)^2 a^2} \sin \Psi \right\}, \end{aligned} \tag{2.33}$$

where

λ = the geocentric latitude of the observer's (satellite's) position,

d = distance from the centre of the earth to the observer (satellite),

R = distance from the centre of the earth to the subobserver (subsatellite point on the earth),

a = the equatorial radius,

Ψ = the azimuth angle in local tangent coordinates, measured from the east direction to the horizon boresight vector, and

f = the ellipticity factor.

For a spherical Earth, $f = 0$, Equation 2.33 reduces to (See Figure 2.8)

$$\rho = \arcsin\left(\frac{a}{d}\right). \tag{2.34}$$

5. The difference between the angular radius of the earth, ρ, and boresight angle, b, will approximately be the resulting roll and pitch angles for small roll and pitch rotations (see Figure 2.8). The boresight angle is calculated by using the dot product between the boresight vector in orbit coordinates and the nadir unit vector, $\mathbf{n} = [0 \;\; 0 \;\; 1]^T$, therefore:

$$\begin{aligned} \text{roll}(\psi) &= \rho - \arccos((\mathbf{v}_o)_{\text{boreY}} \cdot \mathbf{n}), && (2.35)\\ \text{pitch}(\theta) &= \rho - \arccos((\mathbf{v}_o)_{\text{boreX}} \cdot \mathbf{n}). && (2.36) \end{aligned}$$

The calulated roll and pitch angles are only valid if it falls within the sensor's line of sight (LOS). In other words, the angular radius of the earth must fall between the maximum and minimum LOS vector angles. These angles are also calculated by using the dot product between the LOS vectors and the nadir unit vector, $\mathbf{n}$. Therefore,

$$\arccos((\mathbf{v}_o)_{\text{max}} \cdot \mathbf{n}) < \rho < \arccos((\mathbf{v}_o)_{\text{min}} \cdot \mathbf{n}). \tag{2.37}$$

The second constraint is that the horizon must be illuminated. For the planet to be fully illuminated the following equation holds (Wertz [8, page 89]):

$$\Psi' > \pi - \rho, \tag{2.38}$$

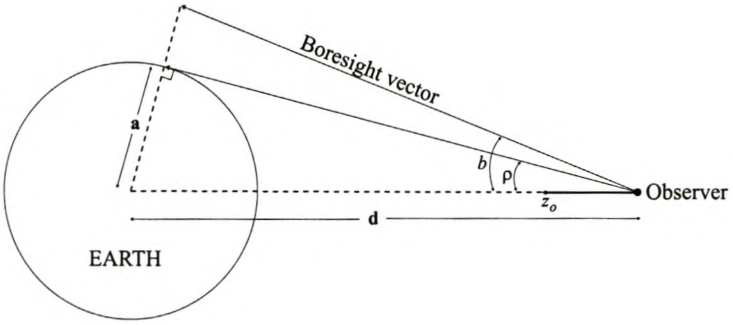


Figure 2.8: *Angular radius, ρ, of the earth and boresight angle, b, of the horizon sensor as seen from the satellite.*

where

$\Psi' \quad = \quad \pi - \alpha + \Delta\xi,$

$\alpha \quad = \quad$ the angle at the centre of the earth between the spacecraft and the sun, and

$\Delta\xi \quad = \quad$ the correction terms in the dark angle, $\Delta\xi \equiv \frac{\pi}{2} - \xi$, where the dark angle, ξ, is approximately 89.15^o for the earth.

2.5 Equations of Motion

This section will describe the mathematical models used to simulate the attitude dynamics of the satellite. The equations of motion for the attitude dynamics can be divided into two sets: the *dynamic equations of motion* and the *kinematic equations of motion.*

2.5.1 Dynamic Equations of Motion

The basic differential equations of attitude dynamics can be expressed in vector form as:

$$\mathbf{I}\dot{\boldsymbol{\omega}}_B^I = \mathbf{N}_{GG} + \mathbf{N}_M + \mathbf{N}_D - \boldsymbol{\omega}_B^I \times (\mathbf{I}\boldsymbol{\omega}_B^I + \mathbf{h}) - \dot{\mathbf{h}}, \tag{2.39}$$

where

$\mathbf{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$ = moment of inertia (MOI) tensor in body coordinates,

$\boldsymbol{\omega}_B^I = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$ = body angular rate vector inertially referenced,

$\mathbf{N}_{GG}$ = gravity gradient torque vector in body coordinates,

$\mathbf{N}_M$ = magnetic torque vector in body coordinates,

$\mathbf{N}_D$ = external disturbance torque vector in body coordinates, and

$\mathbf{h}$ = reaction wheel angular momentum vector in body coordinates.

For an axially symmetric satellite the off-diagonal products of inertial elements in the MOI tensor, **I**, will be zero. The deployed boom also increase the I_{xx} and I_{yy} to a much larger and equal value. This value is called the transverse MOI, I_T. The simplified MOI tensor is:

$$\mathbf{I} = \begin{bmatrix} I_T & 0 & 0 \\ 0 & I_T & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \tag{2.40}$$

2.5.2 Kinematic Equations of Motion

The quaternion representation of the spacecraft kinematics proved to be the most useful in spacecraft work. The kinematics of the spacecraft in orbit-defined coordinates are therefore expressed in vector form as:

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}\mathbf{q}, \tag{2.41}$$

where

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & \omega_{oz} & -\omega_{oy} & \omega_{ox} \\ -\omega_{oz} & 0 & \omega_{ox} & \omega_{oy} \\ \omega_{oy} & -\omega_{ox} & 0 & \omega_{oz} \\ -\omega_{ox} & -\omega_{oy} & -\omega_{oz} & 0 \end{bmatrix}, \tag{2.42}$$

with

$$\boldsymbol{\omega}_B^O = \begin{bmatrix} \omega_{ox} \\ \omega_{oy} \\ \omega_{oz} \end{bmatrix} = \text{body angular rate vector in orbit coordinates.}$$

The body angular rate vector in orbit coordinates can be calculated from the angular rate vector in inertial coordinates and the direction cosine matrix, **A**, as:

$$\boldsymbol{\omega}_B^O = \boldsymbol{\omega}_B^I + \mathbf{A} \begin{bmatrix} 0 \\ \tilde{\omega}_o(t) \\ 0 \end{bmatrix}, \tag{2.43}$$

with

$$\tilde{\omega}_o(t) \approx \omega_o\{1 + 2e\cos(\omega_o t + M_o)\} \quad \text{for small eccentricities, } e, \tag{2.44}$$

where

$\tilde{\omega}_o(t)$ = true orbit angular rate,
ω_o = orbit mean motion,
M_o = orbit mean anomaly at epoch, and
e = orbit eccentricity.

2.6 Disturbance Torques

The main sources of external attitude disturbance torques are the earth's gravitational and magnetic fields, solar radiation pressure and aerodynamic drag.

2.6.1 Gravity-Gradient Torque

The gravity-gradient torque is the result of a variation in the earth's gravitational force field, which tends to keep the satellite nadir pointing. It is expressed in vector form as:

$$\mathbf{N}_{GG} = \frac{3GM_\oplus}{R_s^3}\left[I_{zz} - \frac{I_{xx} + I_{yy}}{2}\right](\mathbf{n}_b \cdot \mathbf{z}_b)(\mathbf{n}_b \times \mathbf{z}_b), \tag{2.45}$$

where

$GM_{\oplus}$ = earth's gravitation constant,

R_s = geocentric spacecraft position vector length,

$\mathbf{n}_b = \begin{bmatrix} A_{13} \\ A_{23} \\ A_{33} \end{bmatrix}$ = nadir pointing unit vector in body coordinates, and

$\mathbf{z}_b = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ = z-axis unit vector in body coordinates.

2.6.2 Aerodynamic Torque

This torque is the result of an interaction between the upper atmosphere and the satellite's surface. The aerodynamic torque is defined as:

$$\mathbf{N}_{AERO} = \rho_a V^2 A_p (\mathbf{c}_p \times \mathbf{V}), \tag{2.46}$$

where

ρ_a = atmospheric density,

V = magnitude of spacecraft velocity vector,

$\mathbf{V}$ = unit spacecraft velocity vector,

A = total projected area of spacecraft, and

$\mathbf{c}_p$ = vector between centre of mass and centre of pressure.

2.6.3 Solar Radiation Torque

The solar radiation torque is expressed as:

$$N_{SOLAR} = \frac{d_o}{c} V^2 A_p (\mathbf{c}_p \times \mathbf{V}), \tag{2.47}$$

where

d_o = average solar radiation constant, and

c = velocity of light.

Chapter 3

Extended Kalman Filter (EKF)

3.1 Introduction

Two Extended Kalman Filters (EKF) will be used to estimate the inertial angular rate, the quaternion representation of the attitude and the aerodynamic disturbance torque. The two filters are based on the same EKF structure designed by Steyn [7]. The main difference is that only two sensor types, a magnetometer and two horizon sensors, will be used. The aim is to investigate an alternative sensor structure consisting of a combination between magnetometer and horizon sensor measurements.

The first EKF will therefore use only measurements from a magnetometer and an eight order IGRF model of the geomagnetic field to obtain the desired vector observations. The second EKF will use measurements from both the horizon sensors and the magnetometer. The measurement vector will consist of the roll and pitch angle measurements from the horizon sensors, and the yaw angle calculated from the magnetometer measurement. The modelled innovation vector will comprise of the estimated roll, pitch and yaw angles. If this method proves to be successful, it will expand the possible sensor combinations that can be used in attitude determination systems.

Measurements from the magnetometer are available throughout the orbit, while horizon sensor measurements are only available during a fraction of the orbit due to limitations on the FOV. The magnetometer EKF will therefore be used throughout the orbit, while the horizon/magnetometer EKF will only be used when valid measurements from both the horizon sensors are available. For simulation and testing purposes, however, it will be assumed that the earth is fully illuminated, the horizon/magnetometer EKF will therefore run throughout the orbit.

The magnetometer EKF runs at a sampling period of 10 seconds, while the horizon/magnetometer EKF is employed at a sampling time of 1 second. Note, however, that measurements from the magnetometer are only available every 10 seconds, therefore magnetometer measurement updates in the horizon/magnetometer EKF are only possible every 10 seconds.

During the implementation of the EKFs, the following assumptions are made:

- The satellite will be axially symmetric with a deployed boom along the spin-axis (z_b-axis). The simplified MOI tensor is:

$$\mathbf{I} = \begin{bmatrix} I_T & 0 & 0 \\ 0 & I_T & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \tag{3.1}$$

- The satellite orbit will be circular with a radius of 800 km. The Gravity Gradient torque can therefore be approximated as:

$$\mathbf{N}_{GG} = 3\omega_o^2 \left(I_T - I_{zz}\right) A_{33} \begin{bmatrix} -A_{23} \\ A_{13} \\ 0 \end{bmatrix}. \tag{3.2}$$

- The satellite will nominally be earth pointing with a certain z_b spin rate.
- The disturbance torque, $\mathbf{N}_D$, is generated mainly by aerodynamic pressure on the satellite, and is modelled as a slowly varying disturbance torque, n_{doy}, around the orbital y_o-axis:

$$\mathbf{N}_D = \mathbf{A} \begin{bmatrix} 0 & n_{doy} & 0 \end{bmatrix}^T, \qquad \text{where} \qquad \dot{n}_{doy} = 0. \tag{3.3}$$

3.2 EKF Implementation

Figure 3.1 shows the blockdiagram of the basic EKF structure. The continuous full state vector to be estimated is:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\boldsymbol{\omega}}_B^I \\ \hat{\mathbf{q}} \\ \hat{n}_{doy} \end{bmatrix}. \tag{3.4}$$

The EKF algorithm, as employed by the two filters, will now be presented. A full derivation of the system and measurement models can be found in Steyn [7, Chapter 5].

3.2.1 EKF Algorithm

1. Propagate the dynamic and kinematic equations of motion (Equations 2.39 and 2.41):

$$\hat{\mathbf{x}}_{k+1/k} = \hat{\mathbf{x}}_{k/k} + \int_k^{k+1} \mathbf{f}(\hat{\mathbf{x}}_{k/k}, \mathbf{u}_k, k)dt \tag{3.5}$$

2. Compute the linearised perturbation state matrix, $\mathbf{F}(\hat{\mathbf{x}}(t_{k+1}), t_{k+1})$:

$$\mathbf{F}(\hat{\mathbf{x}}(t_{k+1}), t_{k+1}) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1/k}} \tag{3.6}$$

3. Obtain the discrete system matrix $\Phi_{k+1/k}$:

$$\Phi_{k+1/k} \approx \mathbf{I} + \mathbf{F}(\hat{\mathbf{x}}(t_{k+1}), t_{k+1})T_s + \frac{1}{2}\left[\mathbf{F}(\hat{\mathbf{x}}(t_{k+1}), t_{k+1})T_s\right]^2 \tag{3.7}$$

4. Propagate the perturbation covariance matrix, $\mathbf{P}_{k+1/k}$:

$$\mathbf{P}_{k+1/k} = \Phi_{k+1/k}\mathbf{P}_{k/k}\Phi_{k+1/k}^T + \mathbf{Q} \tag{3.8}$$

5. Obtain the sensor measurement vector, $\mathbf{v}_{meas,k+1}$, and the modelled measurement vector in orbit coordinates, $\mathbf{v}_{orb,k+1}$, for the appropriate sensor combination. If no valid sensor measurements are available, return to step 1 at the next sampling interval.

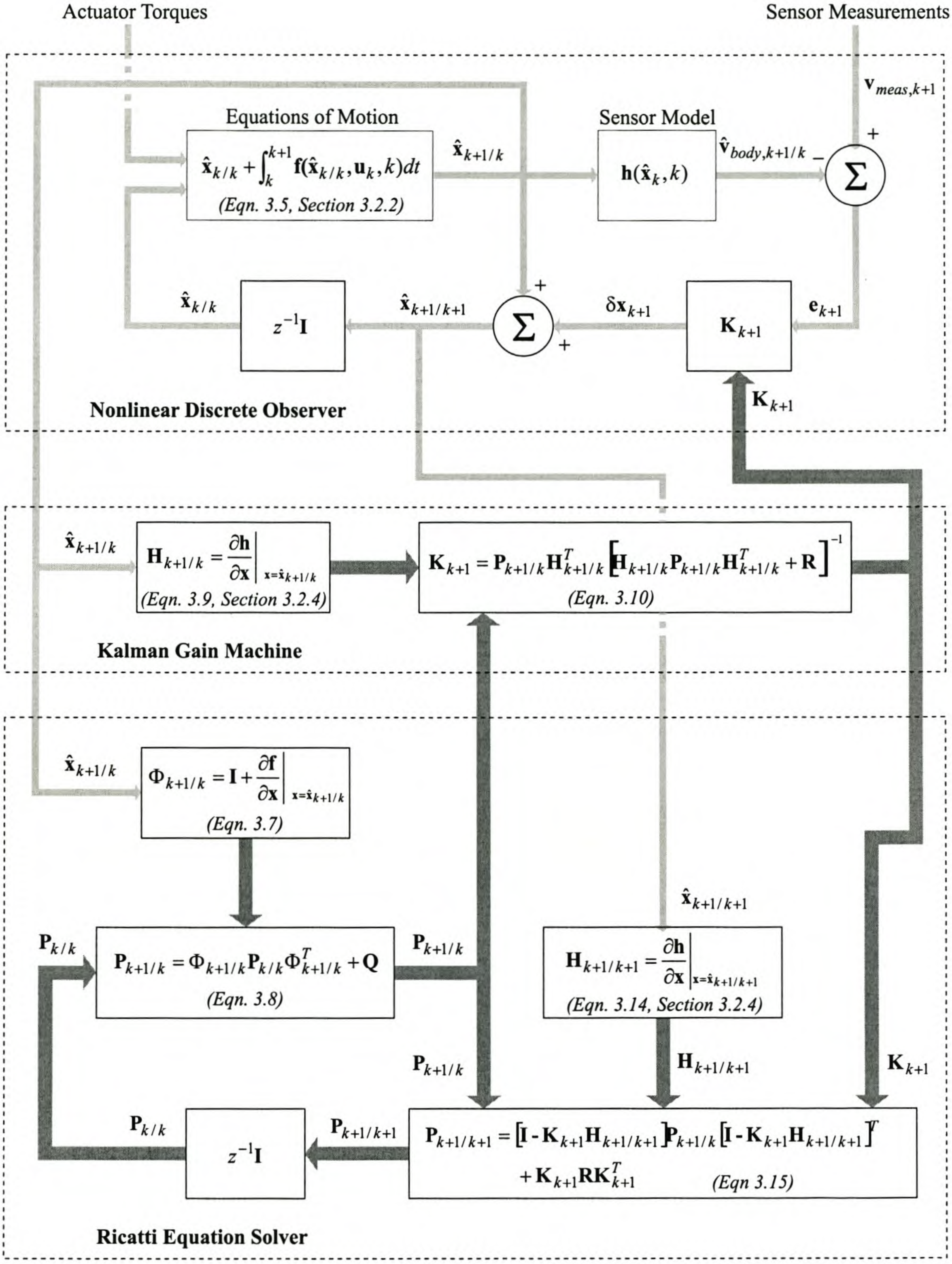

Actuator Torques
Sensor Measurements
$\mathbf{v}_{meas,k+1}$
Equations of Motion
$\hat{\mathbf{x}}_{k/k} + \int_k^{k+1} \mathbf{f}(\hat{\mathbf{x}}_{k/k}, \mathbf{u}_k, k)dt$
(Eqn. 3.5, Section 3.2.2)
$\hat{\mathbf{x}}_{k+1/k}$
Sensor Model
$\mathbf{h}(\hat{\mathbf{x}}_k, k)$
$\hat{\mathbf{v}}_{body,k+1/k}$
+
−
Σ
$\hat{\mathbf{x}}_{k/k}$
$z^{-1}\mathbf{I}$
$\hat{\mathbf{x}}_{k+1/k+1}$
+
Σ
+
$\delta\mathbf{x}_{k+1}$
$\mathbf{K}_{k+1}$
$\mathbf{e}_{k+1}$
$\mathbf{K}_{k+1}$
Nonlinear Discrete Observer
$\hat{\mathbf{x}}_{k+1/k}$
$\mathbf{H}_{k+1/k} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1/k}}$
(Eqn. 3.9, Section 3.2.4)
$\mathbf{K}_{k+1} = \mathbf{P}_{k+1/k}\mathbf{H}^T_{k+1/k}\left[\mathbf{H}_{k+1/k}\mathbf{P}_{k+1/k}\mathbf{H}^T_{k+1/k} + \mathbf{R}\right]^{-1}$
(Eqn. 3.10)
Kalman Gain Machine
$\hat{\mathbf{x}}_{k+1/k}$
$\Phi_{k+1/k} = \mathbf{I} + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1/k}}$
(Eqn. 3.7)
$\hat{\mathbf{x}}_{k+1/k+1}$
$\mathbf{P}_{k/k}$
$\mathbf{P}_{k+1/k} = \Phi_{k+1/k}\mathbf{P}_{k/k}\Phi^T_{k+1/k} + \mathbf{Q}$
(Eqn. 3.8)
$\mathbf{P}_{k+1/k}$
$\mathbf{H}_{k+1/k+1} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1/k+1}}$
(Eqn. 3.14, Section 3.2.4)
$\mathbf{P}_{k+1/k}$
$\mathbf{H}_{k+1/k+1}$
$\mathbf{K}_{k+1}$
$\mathbf{P}_{k/k}$
$z^{-1}\mathbf{I}$
$\mathbf{P}_{k+1/k+1}$
$\mathbf{P}_{k+1/k+1} = \left[\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1/k+1}\right]\mathbf{P}_{k+1/k}\left[\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1/k+1}\right]^T$
$+ \mathbf{K}_{k+1}\mathbf{R}\mathbf{K}^T_{k+1}$
(Eqn 3.15)
Ricatti Equation Solver

Figure 3.1: *Blockdiagram of the Extended Kalman Filter algorithm.*

6. Compute the discrete output measurement matrix, $\mathbf{H}_{k+1/k}$:

$$\mathbf{H}_{k+1/k} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1/k}} \tag{3.9}$$

7. Compute the Kalman Filter Gain, $\mathbf{K}_{k+1}$:

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1/k}\mathbf{H}_{k+1/k}^{T}\left[\mathbf{H}_{k+1/k}\mathbf{P}_{k+1/k}\mathbf{H}_{k+1/k}^{T} + \mathbf{R}\right]^{-1} \tag{3.10}$$

8. Calculate the innovation error vector, $\mathbf{e}_{k+1}$:

$$\mathbf{e}_{k+1} = \mathbf{v}_{meas,k+1} - \mathbf{A}(\hat{\mathbf{q}}_{k+1,k})\mathbf{v}_{orb,k+1} \tag{3.11}$$

9. Update the state vector with the innovation:

$$\hat{\mathbf{x}}_{k+1/k+1} = \hat{\mathbf{x}}_{k+1/k} + \mathbf{K}_{k+1}\mathbf{e}_{k+1} \tag{3.12}$$

   After the state vector has been updated, the quaternion elements of the state vector are normalised to ensure that the estimated quaternions still satisfy the quaternion property, $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$:

$$\hat{\mathbf{q}}_{norm,k+1/k+1} = \frac{\hat{\mathbf{q}}_{k+1/k+1}}{\|\hat{\mathbf{q}}_{k+1/k+1}\|} \tag{3.13}$$

10. Recompute the discrete output measurement matrix, $\mathbf{H}_{k+1/k+1}$, for the updated state vector, $\hat{\mathbf{x}}_{k+1/k+1}$:

$$\mathbf{H}_{k+1/k+1} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1/k+1}} \tag{3.14}$$

11. Update the perturbation covariance matrix:

$$\begin{aligned}\mathbf{P}_{k+1/k+1} &= \left[\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1/k+1}\right]\mathbf{P}_{k+1/k}\left[\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1/k+1}\right]^{T} \\ &\quad + \mathbf{K}_{k+1}\mathbf{R}\mathbf{K}_{k+1}^{T}\end{aligned} \tag{3.15}$$

3.2.2 Propagation of the Equations of Motion

The equations of motion, as described by the differential equations, Equations 2.39 and 2.41, are solved through numerical integration to obtain a discrete solution of the state at each sampling interval. The improved Euler numerical method[1] was used and is defined as:

$$y_{n+1} = y_n + h\frac{f(x_n, y_n) + f(x_{n+1}, y^*_{n+1})}{2}, \tag{3.16}$$

where

$$y^*_{n+1} = y_n + hf(x_n, y_n). \tag{3.17}$$

The dynamic and kinematic equations of motion can now respectively be solved in the following way.

Propagation of Dynamic Equations of Motion

$$\boldsymbol{\omega}^I_{B,k+1} = \boldsymbol{\omega}^I_{B,k} + T_s\frac{f(t_k, \boldsymbol{\omega}^I_{B,k}) + f(t_{k+1}, \boldsymbol{\omega}^{*I}_{B,k+1})}{2}, \tag{3.18}$$

where

$$f(t_{k+1}, \boldsymbol{\omega}^{*I}_{B,k+1}) = \boldsymbol{\omega}^{*I}_{B,k+1} = \boldsymbol{\omega}^I_{B,k} + T_s f(t_k, \boldsymbol{\omega}^I_{B,k}), \tag{3.19}$$

and

$$f(t_k, \boldsymbol{\omega}^I_{B,k}) = \dot{\boldsymbol{\omega}}^I_{B,k} = \mathbf{I}^{-1}[\mathbf{N_{GG}} + \mathbf{N_D} - \boldsymbol{\omega}^I_{B,k} \times (\boldsymbol{I}\boldsymbol{\omega}^I_{B,k} + \mathbf{h}) - \dot{\mathbf{h}}]. \tag{3.20}$$

Propagation of Kinematic Equations of Motion

$$\mathbf{q}_{k+1} = \mathbf{q}_k + T_s\frac{f(t_k, \mathbf{q}_k), f(t_{k+1}, \mathbf{q}^*_{k+1})}{2}, \tag{3.21}$$

where

$$f(t_{k+1}, \mathbf{q}^*_{k+1}) = \mathbf{q}^*_{k+1} = \mathbf{q}_k + T_s f(t_k, \mathbf{q}_k), \tag{3.22}$$

and

$$f(t_k, \mathbf{q}_k) = \dot{\mathbf{q}}_k = \frac{1}{2}\mathbf{\Omega}\mathbf{q}_k. \tag{3.23}$$

[1] See Zill & Cullen [10], page 271

3.2.3 Computing the F-matrix

The linearized perturbation state model used by the EKF is:

$$\delta\dot{\mathbf{x}}(t_k) = \mathbf{F}(\mathbf{x}(t_k), t_k)\delta\mathbf{x}(t_k) + \mathbf{s}(t_k), \tag{3.24}$$

where

$$\begin{aligned}
\delta\mathbf{x}(t_k) = \mathbf{x}(t_k) - \hat{\mathbf{x}}(t_k) &= \text{the state perturbation, or the difference between the actual state and the estimated state,}\\
\mathbf{F}(\mathbf{x}(t_k), t_k) &= \text{the } 8 \times 8 \text{ linearised perturbation state matrix, and}\\
\mathbf{s}(t_k) = N\{\mathbf{0}, \mathbf{Q}(t_k)\} &= \text{the process noise in the system (zero mean white noise with covariance matrix, } \mathbf{Q}).
\end{aligned}$$

The linearized perturbation state matrix, **F**, is defined as:

$$\mathbf{F}(\hat{\mathbf{x}}(t_k), t_k) = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}}. \tag{3.25}$$

From Equations 2.39, 2.41 and 3.3 then follows:

$$\mathbf{F}(\hat{\mathbf{x}}(t_k), t_k) = \begin{bmatrix} \frac{\partial \dot{\boldsymbol{\omega}}}{\partial \boldsymbol{\omega}} & \frac{\partial \dot{\boldsymbol{\omega}}}{\partial \mathbf{q}} & \frac{\partial \dot{\boldsymbol{\omega}}}{\partial n_{doy}} \\ \frac{\partial \dot{\mathbf{q}}}{\partial \boldsymbol{\omega}} & \frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}} & \frac{\partial \dot{\mathbf{q}}}{\partial n_{doy}} \\ \frac{\partial \dot{n}_{doy}}{\partial \boldsymbol{\omega}} & \frac{\partial \dot{n}_{doy}}{\partial \mathbf{q}} & \frac{\partial \dot{n}_{doy}}{\partial n_{doy}} \end{bmatrix}, \tag{3.26}$$

where

- the first row represents the derivatives of $\dot{\boldsymbol{\omega}}_B^I$ with respect to $\boldsymbol{\omega}_B^I$, $\mathbf{q}$ and n_{doy}:

$$\frac{\partial \dot{\boldsymbol{\omega}}}{\partial \boldsymbol{\omega}} =$$

$$\begin{bmatrix} 0 & \hat{\omega}_z\left(1-\frac{I_{zz}}{I_T}\right)-\frac{h_z}{I_T} & \hat{\omega}_y\left(1-\frac{I_{zz}}{I_T}\right)+\frac{h_y}{I_T} \\ -\hat{\omega}_z\left(1-\frac{I_{zz}}{I_T}\right)+\frac{h_z}{I_T} & 0 & -\hat{\omega}_x\left(1-\frac{I_{zz}}{I_T}\right)-\frac{h_x}{I_T} \\ -\frac{h_y}{I_{zz}} & \frac{h_x}{I_{zz}} & 0 \end{bmatrix}, \tag{3.27}$$

$$\frac{\partial \dot{\boldsymbol{\omega}}}{\partial \mathbf{q}} = \frac{\partial \mathbf{N}_{GG}}{\partial \mathbf{q}} + \frac{\partial \mathbf{N}_D}{\partial \mathbf{q}}, \tag{3.28}$$

$$\frac{\partial \dot{\boldsymbol{\omega}}}{\partial n_{doy}} = \frac{\partial \mathbf{N}_D}{\partial n_{doy}} = \begin{bmatrix} \hat{A}_{12} \\ \hat{A}_{22} \\ 0 \end{bmatrix}, \tag{3.29}$$

with

$$\frac{\partial \mathbf{N}_{GG}}{\partial \mathbf{q}} = 6\omega_o^2(I_T - I_{zz})\times$$

$$\begin{bmatrix} \hat{A}_{23}\hat{q}_1 - \hat{A}_{33}\hat{q}_4 & \hat{A}_{23}\hat{q}_2 - \hat{A}_{33}\hat{q}_3 & -\hat{A}_{33}\hat{q}_2 - \hat{A}_{23}\hat{q}_3 & -\hat{A}_{33}\hat{q}_1 - \hat{A}_{23}\hat{q}_4 \\ -\hat{A}_{13}\hat{q}_1 + \hat{A}_{33}\hat{q}_3 & -\hat{A}_{13}\hat{q}_2 - \hat{A}_{33}\hat{q}_4 & \hat{A}_{33}\hat{q}_1 + \hat{A}_{13}\hat{q}_3 & -\hat{A}_{33}\hat{q}_2 + \hat{A}_{13}\hat{q}_4 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.30}$$

and

$$\frac{\partial \mathbf{N}_D}{\partial \mathbf{q}} = 2\begin{bmatrix} \hat{q}_2\hat{n}_{doy} & \hat{q}_1\hat{n}_{doy} & \hat{q}_4\hat{n}_{doy} & \hat{q}_3\hat{n}_{doy} \\ -\hat{q}_1\hat{n}_{doy} & \hat{q}_2\hat{n}_{doy} & -\hat{q}_3\hat{n}_{doy} & \hat{q}_4\hat{n}_{doy} \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.31}$$

- the second row represents the derivatives of $\dot{\mathbf{q}}$ with respect to $\boldsymbol{\omega}_B^I$, $\mathbf{q}$ and n_{doy}:

$$\frac{\partial \dot{\mathbf{q}}}{\partial \boldsymbol{\omega}} = \frac{1}{2}\begin{bmatrix} \hat{q}_4 & -\hat{q}_3 & \hat{q}_2 \\ \hat{q}_3 & \hat{q}_4 & -\hat{q}_1 \\ -\hat{q}_2 & \hat{q}_1 & \hat{q}_4 \\ -\hat{q}_1 & -\hat{q}_2 & -\hat{q}_3 \end{bmatrix}, \tag{3.32}$$

$$\frac{\partial \dot{\mathbf{q}}}{\partial \mathbf{q}} = \frac{1}{2}\hat{\boldsymbol{\Omega}} = \frac{1}{2}\begin{bmatrix} 0 & \hat{\omega}_{oz} & -\hat{\omega}_{oy} & \hat{\omega}_{ox} \\ -\hat{\omega}_{oz} & 0 & \hat{\omega}_{ox} & \hat{\omega}_{oy} \\ \hat{\omega}_{oy} & -\hat{\omega}_{ox} & 0 & \hat{\omega}_{oz} \\ -\hat{\omega}_{ox} & -\hat{\omega}_{oy} & -\hat{\omega}_{oz} & 0 \end{bmatrix}, \tag{3.33}$$

$$\frac{\partial \dot{\mathbf{q}}}{\partial n_{doy}} = \mathbf{0}_{4\times1}. \tag{3.34}$$

- the third row represents the derivatives of $\dot{n}_{doy}$ with respect to $\boldsymbol{\omega}_B^I$, $\mathbf{q}$ and n_{doy}:

$$\frac{\partial \dot{n}_{doy}}{\partial \boldsymbol{\omega}} = \mathbf{0}_{1\times3}, \qquad \frac{\partial \dot{n}_{doy}}{\partial \mathbf{q}} = \mathbf{0}_{1\times4}, \qquad \frac{\partial \dot{n}_{doy}}{\partial n_{doy}} = 0. \tag{3.35}$$

Since a discrete version of the EKF is used, the $\mathbf{F}$ matrix must be converted to the discrete system matrix, Φ_k, so that:

$$\delta\mathbf{x}_{k+1} = e^{(\mathbf{F}(\hat{\mathbf{x}}(t_k),t_k)T_s)}\delta\mathbf{x}_k = \Phi_k\delta\mathbf{x}_k, \tag{3.36}$$

therefore,

$$\Phi_k = e^{(\mathbf{F}(\hat{\mathbf{x}}(t_k),t_k)T_s)}. \tag{3.37}$$

The discrete system matrix can be approximated with a second order Taylor series expansion as:

$$\Phi_k \approx \mathbf{I} + \mathbf{F}(\hat{\mathbf{x}}(t_k),t_k)T_s + \frac{1}{2}\left[\mathbf{F}(\hat{\mathbf{x}}(t_k),t_k)T_s\right]^2. \tag{3.38}$$

3.2.4 Computing the H-matrix

The output measurement matrix, $\mathbf{H}_k$, is a 3×8 matrix which relates the innovation error vector, $\mathbf{e}_k$, to the state perturbation vector, $\delta\mathbf{x}_k$, in the following way:

$$\mathbf{e}_k = \mathbf{H}_k(\hat{\mathbf{x}}_k)\delta\mathbf{x} + \mathbf{m}_k, \tag{3.39}$$

where

$$\mathbf{m}_k = N\{\mathbf{0}, \mathbf{R}_k\} \quad = \quad \text{Discrete zero mean white measurement noise with covariance matrix, } \mathbf{R}.$$

$\mathbf{H}_k$ is derived from the innovation error vector calculation in the following way:

$$\begin{aligned} \mathbf{e}_k &= \mathbf{v}_{meas,k} - \mathbf{A}(\hat{\mathbf{q}}_k)\mathbf{v}_{orb,k} \\ &= \left[\sum_{i=1}^{4} \frac{\delta\mathbf{A}(\hat{\mathbf{q}}_k)}{\delta q_{i,k}}\delta q_{i,k}\right]\mathbf{v}_{orb,k} + \mathbf{m}_k \\ &= \left[\sum_{i=1}^{4} \frac{\delta\mathbf{A}(\hat{\mathbf{q}}_k)}{\delta q_{i,k}}\mathbf{v}_{orb,k}\right]\delta\mathbf{q}_k + \mathbf{m}_k \\ &= \begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 & \mathbf{h}_4\end{bmatrix}\delta\mathbf{q}_k + \mathbf{m}_k \\ &= \begin{bmatrix}\mathbf{0}_{3\times3} & \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 & \mathbf{h}_4 & \mathbf{0}_{3\times1}\end{bmatrix}\delta\mathbf{x}_k + \mathbf{m}_k \\ &= \mathbf{H}_k(\hat{\mathbf{q}}_k)\delta\mathbf{x}_k + \mathbf{m}_k. \end{aligned} \tag{3.40}$$

From this it can be seen that:

$$\mathbf{H}_k = \begin{bmatrix}\mathbf{0}_{3\times3} & \mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3 & \mathbf{h}_4 & \mathbf{0}_{3\times1}\end{bmatrix}, \tag{3.41}$$

with

$$\mathbf{h}_i = \frac{\delta\mathbf{A}(\hat{\mathbf{q}}_k)}{\delta q_{i,k}}\mathbf{v}_{orb,k}, \qquad i = 1,2,3,4 \tag{3.42}$$

then

$$\begin{aligned} \mathbf{h}_1 &= 2\begin{bmatrix} \hat{q}_{1,k} & \hat{q}_{2,k} & \hat{q}_{3,k} \\ \hat{q}_{2,k} & -\hat{q}_{1,k} & \hat{q}_{4,k} \\ \hat{q}_{3,k} & -\hat{q}_{4,k} & -\hat{q}_{1,k} \end{bmatrix}\mathbf{v}_{orb,k}, \\ \mathbf{h}_2 &= 2\begin{bmatrix} -\hat{q}_{2,k} & \hat{q}_{1,k} & -\hat{q}_{4,k} \\ \hat{q}_{1,k} & \hat{q}_{2,k} & \hat{q}_{3,k} \\ \hat{q}_{4,k} & \hat{q}_{3,k} & -\hat{q}_{2,k} \end{bmatrix}\mathbf{v}_{orb,k}, \end{aligned}$$

$$\mathbf{h}_3 = 2\begin{bmatrix} -\hat{q}_{3,k} & \hat{q}_{4,k} & \hat{q}_{1,k} \\ -\hat{q}_{4,k} & -\hat{q}_{3,k} & \hat{q}_{2,k} \\ \hat{q}_{1,k} & \hat{q}_{2,k} & \hat{q}_{3,k} \end{bmatrix}\mathbf{v}_{orb,k},$$

$$\mathbf{h}_4 = 2\begin{bmatrix} \hat{q}_{4,k} & \hat{q}_{3,k} & -\hat{q}_{2,k} \\ -\hat{q}_{3,k} & \hat{q}_{4,k} & \hat{q}_{1,k} \\ \hat{q}_{2,k} & -\hat{q}_{1,k} & \hat{q}_{4,k} \end{bmatrix}\mathbf{v}_{orb,k}.$$

3.2.5 Innovation Computation

The innovation error vector is computed as the vector difference between a measured normalised vector, $\mathbf{v}_{meas}$, and a modelled normalised vector, $\mathbf{v}_{body}$. Both vectors are in body coordinates, although the modelling is actually done in orbit coordinates, $\mathbf{v}_{orb}$, and then transformed to $\mathbf{v}_{body}$ by the DCM, $\mathbf{A}$. In other words:

$$\mathbf{e}_k = \mathbf{v}_{meas,k} - \mathbf{A}(\hat{\mathbf{q}}_k)\mathbf{v}_{orb,k}. \tag{3.43}$$

The vectors are normalized to reduce the effects of any magnitude errors due to inaccurate modelling, while still preserving the directional information.

Innovation data can be obtained from any attitude sensor able to supply vector directional measurements. The EKFs presented used two different sensors to obtain innovation data. The first used magnetometer measurements of the direction and strength of the geomagnetic field in body coordinates and the second a combination of magnetometer measurements and horizon sensor measurements of the sunlit earth horizon.

Magnetometer Innovation

The magnetometer EKF relies fully on the magnetometer for 3-axis measurements at 10 second intervals to obtain the measured innovation vector. The normalised measured innovation vector then is:

$$\mathbf{v}_{meas,k} = \frac{\mathbf{B}_{meas,k}}{\|\mathbf{B}_{meas,k}\|}. \tag{3.44}$$

An eight-order IGRF model (Appendix A) was used to calculate the modelled geomagnetic field in orbit coordinates. The normalised modelled innovation vector in orbital coordinates will then be:

$$\mathbf{v}_{orb,k} = \frac{\mathbf{B}_{orb,k}}{\|\mathbf{B}_{orb,k}\|}. \tag{3.45}$$

Horizon/Magnetometer Combination Innovation

The horizon/magnetometer EKF used both the horizon sensors and the magnetometer to construct the measurement vector. The -X-horizon sensor and the Y-horizon sensor measure the pitch and roll angles respectively, while the measurement from the magnetometer can be used to calculate the yaw angle. Measurements from the horizon sensors are available every second, while measurements from the magnetometer are only available every 10 seconds.

The yaw angle can be calculated from the magnetometer measurement, $\mathbf{B}_{meas,k}$, in the following way:

1. Obtain the geomagnetic field vector in orbital coordinates, $\mathbf{B}_{orb,k}$, from the IGRF model in Appendix A.

2. Figure 3.2 shows the geomagnetic field vector, $\mathbf{B}$, mapped onto the orbit coordinates, $\mathbf{B}_{orb,k}$, and body coordinates, $\mathbf{B}_{meas,k}$. $\mathbf{B}_{meas,k}$ are obtained by rotating the field in orbit coordinates, $\mathbf{B}_{orb,k}$, through an Euler 3-2-1 sequence of rotations (yaw $= \phi$, pitch $= \theta$, roll $= \psi$). Note that it is assumed at this stage that any roll or pitch rotations are small enough to be ignored. That leaves only the yaw rotation, ϕ. The azimuth angles of the $\mathbf{B}_{orb,k}$ and $\mathbf{B}_{meas,k}$ vectors will then respectively be:

$$\text{Azim}_{orb,k} = \arctan\left(\frac{B_{orb,k(y)}}{B_{orb,k(x)}}\right), \tag{3.46}$$

$$\text{Azim}_{meas,k} = \arctan\left(\frac{B_{meas,k(y)}}{B_{meas,k(x)}}\right). \tag{3.47}$$

3. The difference between the azimuth angle of $\mathbf{B}_{meas,k}$ and the azimuth angle of $\mathbf{B}_{orb,k}$ are the resulting yaw angle, ϕ. In other words:

$$\text{yaw}(\phi) = \text{Azim}_{orb,k} - \text{Azim}_{meas,k}. \tag{3.48}$$

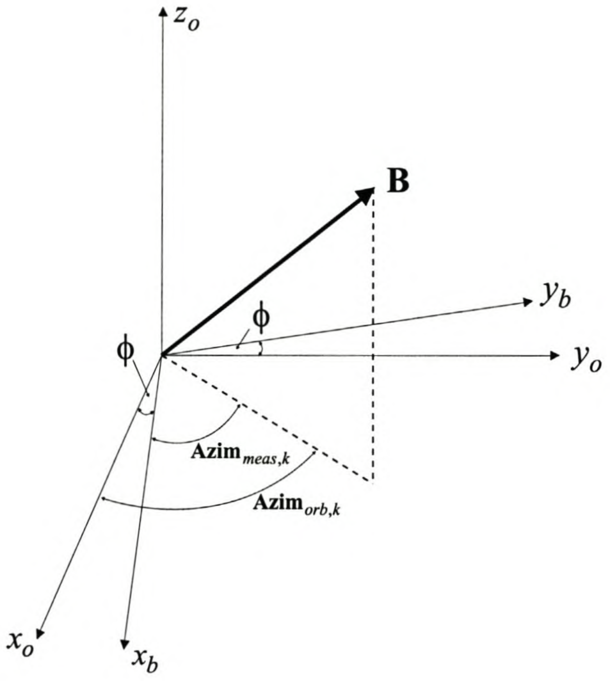


Figure 3.2: *Geomagnetic field vector mapped in orbit and body coordinates.*

The measured roll, pitch and yaw angles are now used to construct the measurement DCM, $\mathbf{A}_{meas,k}$. The measurement vector is normalised by transforming a unit vector in orbit coordinates to the body coordinates with $\mathbf{A}_{meas,k}$:

$$\mathbf{v}_{meas,k} = \mathbf{A}_{meas,k}\mathbf{u}, \tag{3.49}$$

where $\mathbf{u}$ is a unit vector in orbit coordinates of the form:

$$\mathbf{u} = \frac{\mathbf{v}}{\|\mathbf{v}\|}, \qquad \text{and} \qquad \mathbf{v} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T. \tag{3.50}$$

The modelled normalised innovation vector are obtained by transforming the unit vector, $\mathbf{u}$, from the orbit to body coordinates, but this time using the estimated DCM, $\mathbf{A}(\hat{\mathbf{q}}_k)$:

$$\mathbf{v}_{body,k} = \mathbf{A}(\hat{\mathbf{q}}_k)\mathbf{u}. \tag{3.51}$$

3.3 EKF Simulation Results

The Extended Kalman Filters were tested through simulation. Several assumptions with regard to satellite parameters, orbit parameters and sensor measurements were made.

- The satellite was assumed to be a near cubical micro satellite with a deployed gravity boom to earth stabilize it. The body z_b-axis of the satellite was nadir pointing and kept at a slow spin. The satellite was not actively controlled but left in a free librating mode with $I_T = 40$ kgm^2 and $I_{zz} = 2$ kgm^2. The initial satellite angular rate components were zero for the x_b and y_b axes, and 5 rpo (resolutions per orbit) for the z_b-axis (yaw spin), unless otherwise stated.

- The orbit was assumed to be circular with a radius of 800 km and an inclination of 45^o. The orbital period was approximately 100 minutes. To simplify the computations it was assumed that the earth was fixed about its spinning axis. It was further assumed that a valid horizon would be available throughout the orbit.

- Modelling errors, measurement noise and disturbances are the major causes of filter inaccuracy or divergence. According to Psiaki [11], IGRF modelling errors are the dominant contributing factor to the magnetometer EKF innovation. The IGRF modelled vector, $\mathbf{B}_{orb}$, was therefore obtained by adding uniformly distributed noise components, within the range -0.3 to 0.3 μT, to each calculated vector element. From Table 5-1 in Wertz [8] the expected maximum and RMS errors in the field magnitude, when using the IGRF model, are 0.54 μT and 0.18 μT respectively at an altitude of 445 km.

- The roll and pitch angle measurements from the horizon sensors were obtained by adding uniformly distributed noise, within the range -0.03^o to 0.03^o (0.5 mrad), to the true horizon sensor model.

- An offset-sinusoidal aerodynamic disturbance torque with an amplitude of $n_{doy} = 3$ μNm was employed during simulations.

The magnetometer EKF was implemented at a sampling time of 10 seconds, and the horizon/magnetometer EKF at 1 second intervals, although the magnetometer measurements were only available every 10 seconds.

The initial filter covariance matrix, $\mathbf{P}_o$, was initialized as a diagonal matrix with elements approximately 3 orders of magnitude higher than the system covariance matrix, $\mathbf{Q}$. The system and measurement noise covariance matrices, $\mathbf{Q}$ and $\mathbf{R}$, were chosen differently for each EKF to optimize the filter's performance and to reflect the difference in measurement noise for each sensor.

The stability and ability of the EKFs to converge in the face of large initial attitude errors, measurement noise and modelling errors were evaluated through the simulation tests. The magnetometer and horizon/magnetometer EKFs were tested separately.

Magnetometer EKF

- The stability of the magnetometer EKF was tested first. It was assumed that the satellite had a z_b-spin of 5 rpo and roll and pitch angles of both 0^o. The estimator state values were initialized with the correct values. Figure 3.3 show the estimated angular rate, roll and pitch angles and the estimated disturbance torque, n_{doy}.

- Secondly the ability of the EKF to converge from an unknown initial attitude state was tested. The roll and pitch angles were 10^o and 5^o respectively. The estimator state values were initialized with zero values, except $\hat{\omega}_z$, which was set to 90% of the correct value to speed up the conversion rate. Figure 3.4 show the estimated angular rate, roll and pitch angles and the estimated disturbance torque, n_{doy}.

- Another tested characteristic of the EKF was its ability to converge from large initial attitude errors. The roll and pitch angles were set to 50^o each. The initial estimator state values were all set to zero, except $\hat{\omega}_z$, which was initialized with the correct value. Figure 3.5 display the estimated roll and pitch angles.

- Finally the ability of the EKF to converge from an unknown torque induced on the satellite at time, t = 4000 sec, was evaluated. The satellite had an initial z_b-spin, ω_z, of 5 rpo. The estimator state values were all set to the correct values. Figure 3.6 show the estimated angular rate and roll and pitch angles after an unknown torque decreased the z_b-axis angular rate, ω_z, to 90% of its initial value.

All the above simulations were done over a period of two orbits ($\approx$ 12000 sec). The filter covariance matrix, $\mathbf{P}$, and system covariance matrix, $\mathbf{Q}$, were initialized with the following diagonal values:

$$\begin{aligned} \mathbf{P}_o &= \begin{bmatrix} 1e^{-1} & 1e^{-1} & 1e^{-1} & 1e^{5} & 1e^{5} & 1e^{5} & 1e^{5} & 5e^{-6} \end{bmatrix}, \\ \mathbf{Q} &= \begin{bmatrix} 2e^{-5} & 2e^{-5} & 2e^{-5} & 2e^{1} & 2e^{1} & 2e^{1} & 2e^{1} & 10e^{-9} \end{bmatrix}. \end{aligned} \tag{3.52}$$

Horizon/Magnetometer EKF

It was assumed that measurements from the horizon sensors were available throughout the orbit during the evaluation of the horizon/magnetometer EKF. This was to prevent the accumulation of estimation errors during the open loop period when no measurement updates are available.

- The first test performed on the horizon/magnetometer EKF was a stability test. It was assumed that the satellite has an initial z_b-spin, ω_z, of 5 rpo with roll and pitch angles of 0^o each. The estimator initial values were all set to the correct values. Figure 3.7 display the estimated angular rate, roll and pitch angles and the disturbance torque.

- The ability of the EKF to converge from an unknown initial z_b-spin rate, ω_z, was tested next. It was assumed that the satellite has an initial z_b-spin of 5 rpo. Roll and pitch angles were set to zero. The estimator initial values were all set to zero, except $\hat{\omega}_z$, which was initialized to 90% of the correct value. Figure 3.8 display the estimated angular rate, pitch and roll angles and disturbance torque.

- Finally, Figure 3.9 display the results of the EKF's convergence test from an unknown torque induced on the satellite at time, t = 4000 sec. The satellite had an initial ω_z-spin of 5 rpo. The estimator state values were all set to the correct values. The unknown torque reduced ω_z to 90% of its initial value.

The simulations were performed over a period of 2 orbits ($\approx$ 12000 sec). The filter covariance matrix, **P**, and system covariance matrix, **Q**, were initialized with the following diagonal values:

$$\begin{aligned} \mathbf{P}_o &= \begin{bmatrix} 4e^{-1} & 4e^{-1} & 4e^{-1} & 2e^{4} & 2e^{4} & 2e^{4} & 2e^{4} & 5e^{-5} \end{bmatrix}, \\ \mathbf{Q} &= \begin{bmatrix} 2e^{-5} & 2e^{-5} & 2e^{-5} & 5e^{1} & 5e^{1} & 5e^{1} & 5e^{1} & 10e^{-8} \end{bmatrix}. \end{aligned} \tag{3.53}$$

Result Summary

- The magnetometer EKF was able to extract full attitude information in all cases with an average convergence time of less than an orbit. Attitude errors of less than 1^o were obtained during normal operating conditions.

- The horizon/magnetometer EKF did not perform as well as would be expected. Although the filter was stable, and converged during all the tests performed, the convergence time was too slow. The EKF could only achieve an accuracy of approximately 5^o, which is even worse than the magnetometer EKF. The following section will discuss possible reasons for these bad results.

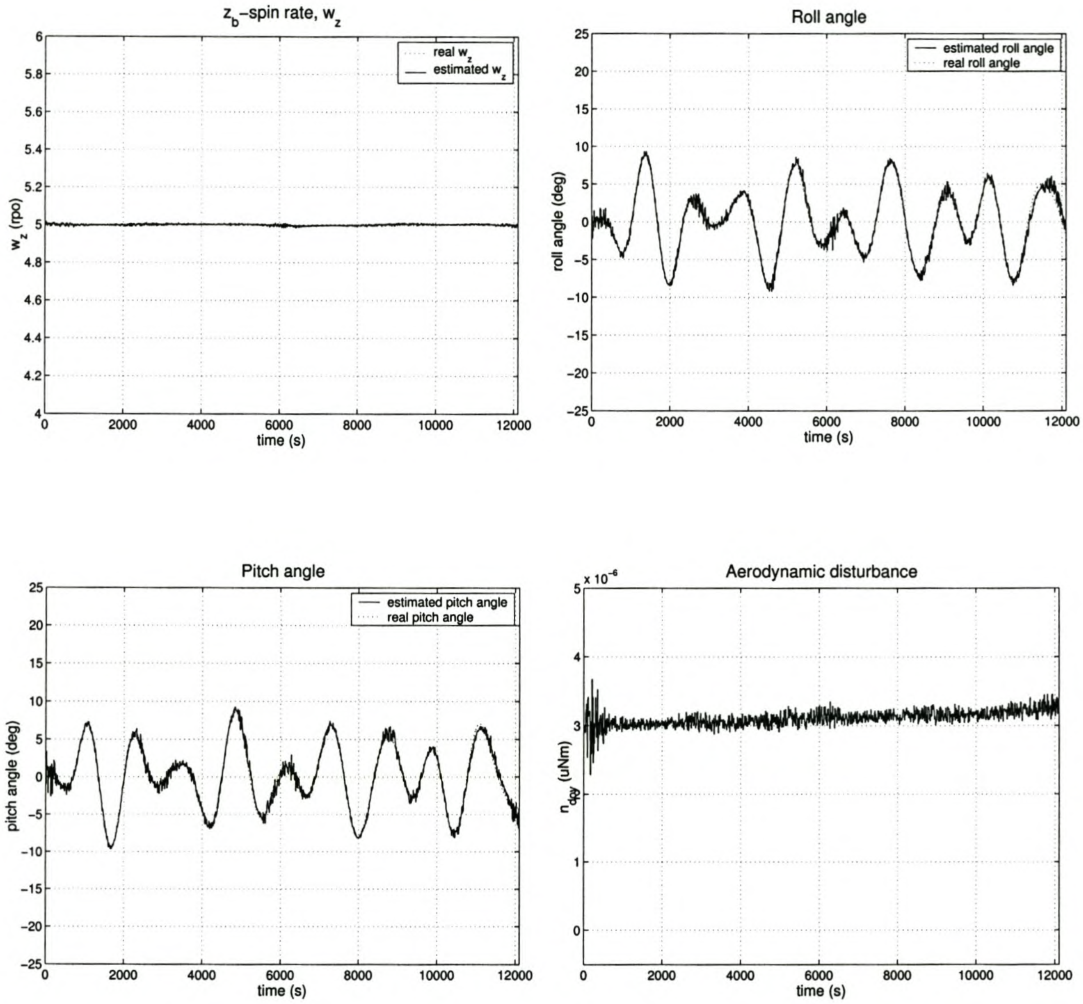


Figure 3.3: *Estimated angular rate, roll and pitch angles and disturbance torque from the magnetometer EKF.*

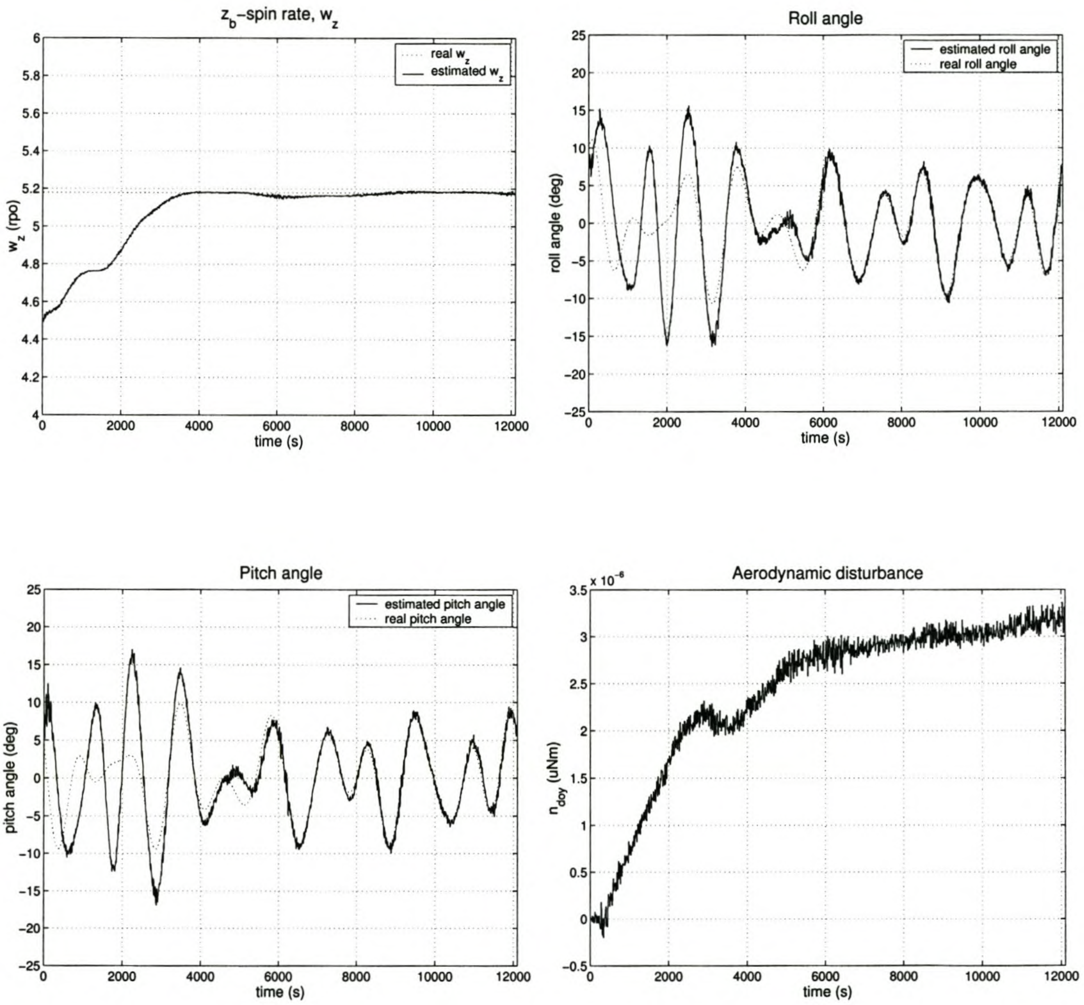


Figure 3.4: *Convergence performance of the magnetometer EKF from an unknown initial attitude state.*

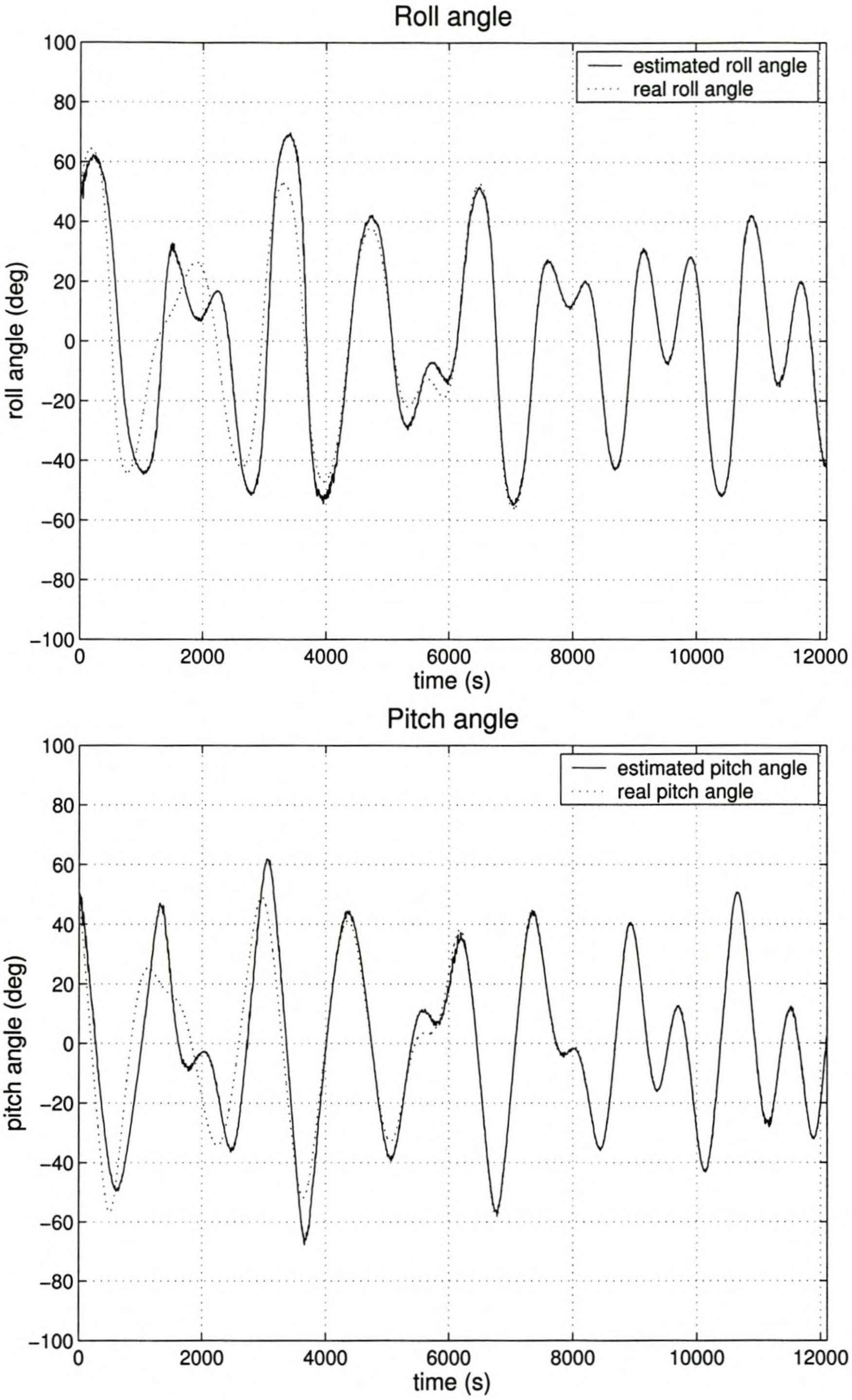


Figure 3.5: *Convergence performance of the magnetometer EKF from large initial attitude estimation errors.*

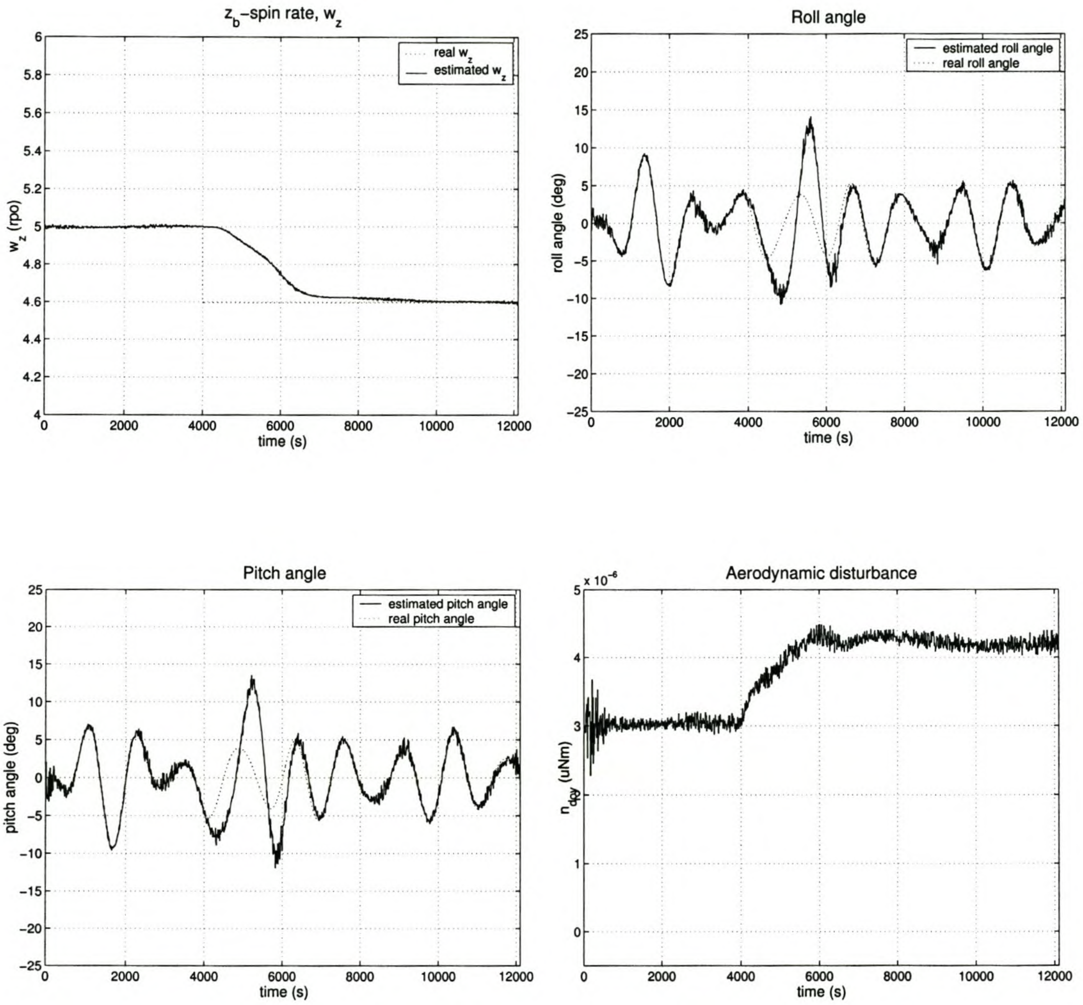


Figure 3.6: *Convergence performance of the magnetometer EKF from an unknown disturbance torque.*

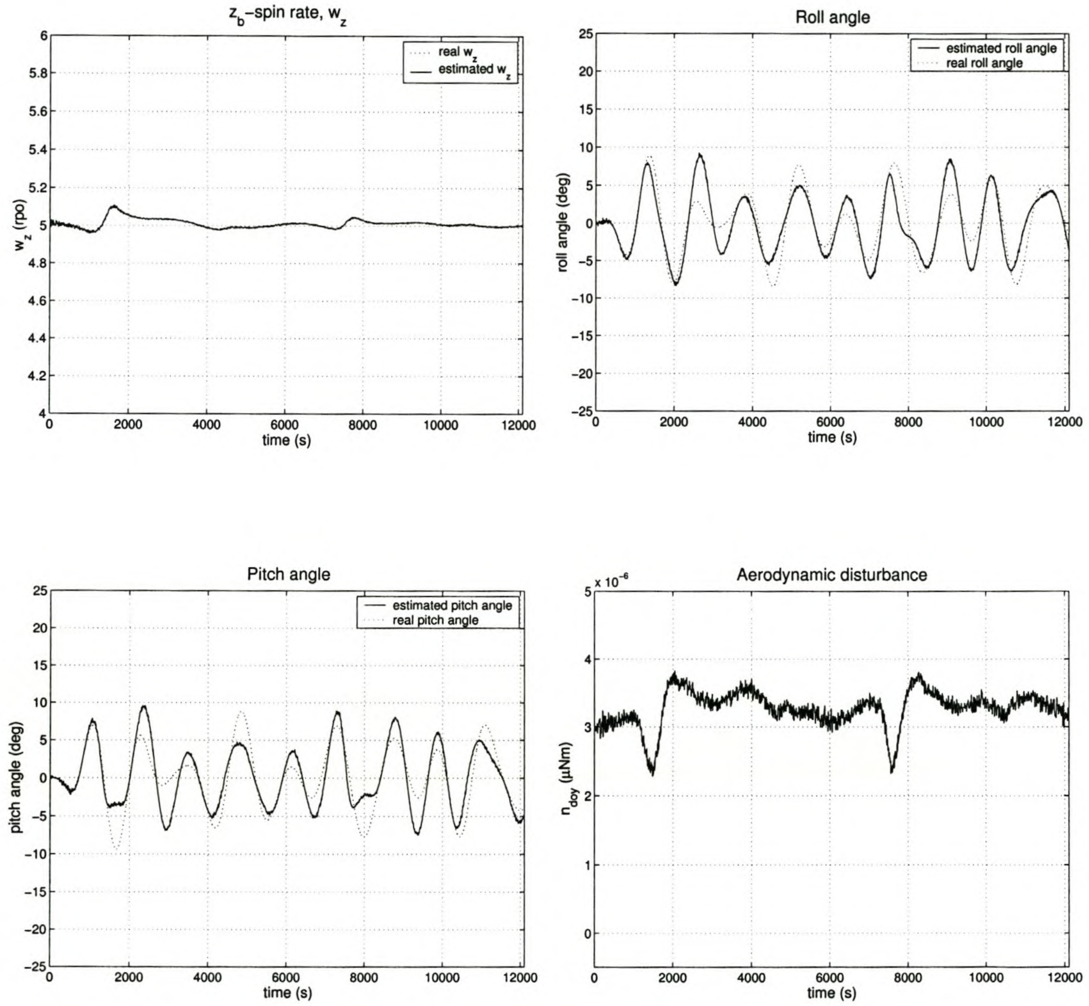


Figure 3.7: *Estimated angular rate, roll and pitch angles and disturbance torque from the horizon/magnetometer EKF.*

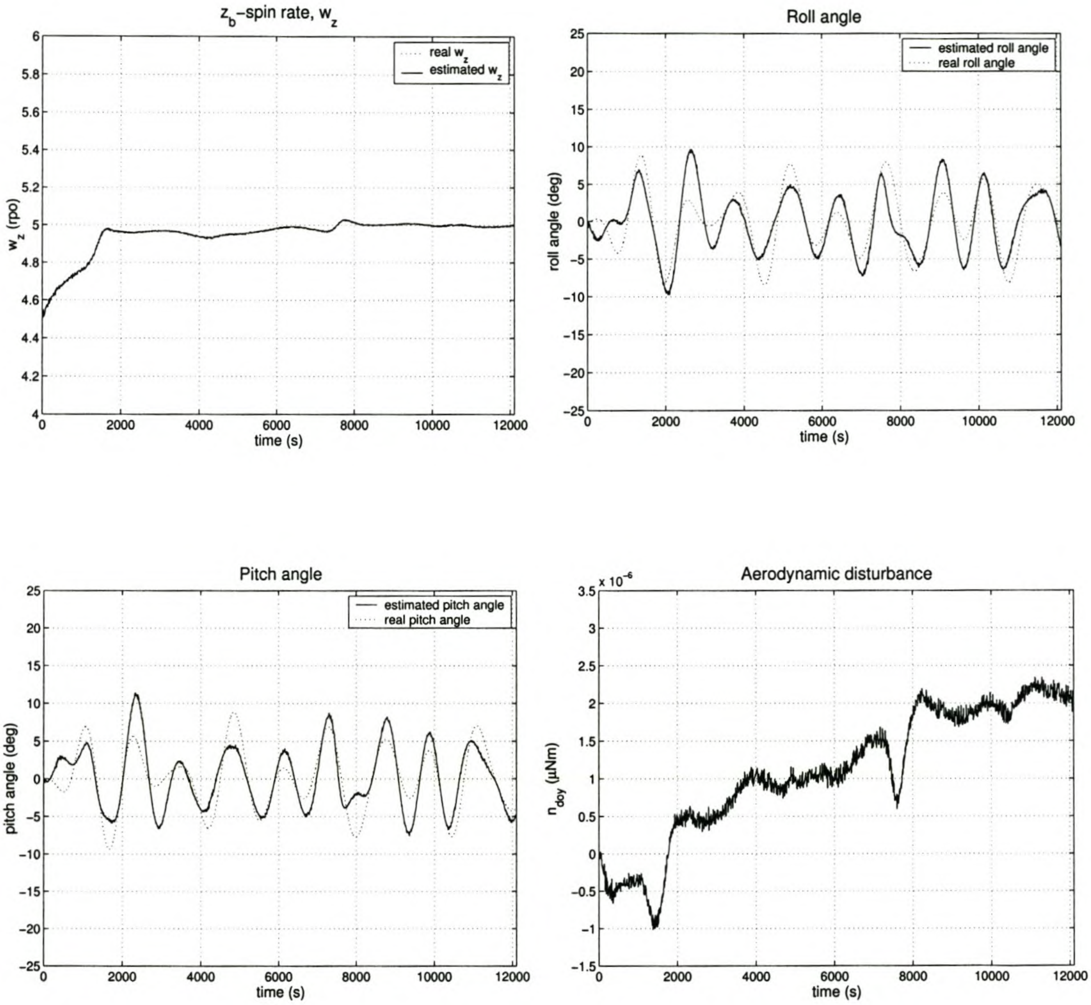


Figure 3.8: *Convergence performance of the horizon/magnetometer EKF from an unknown initial* z_b*-spin rate.*

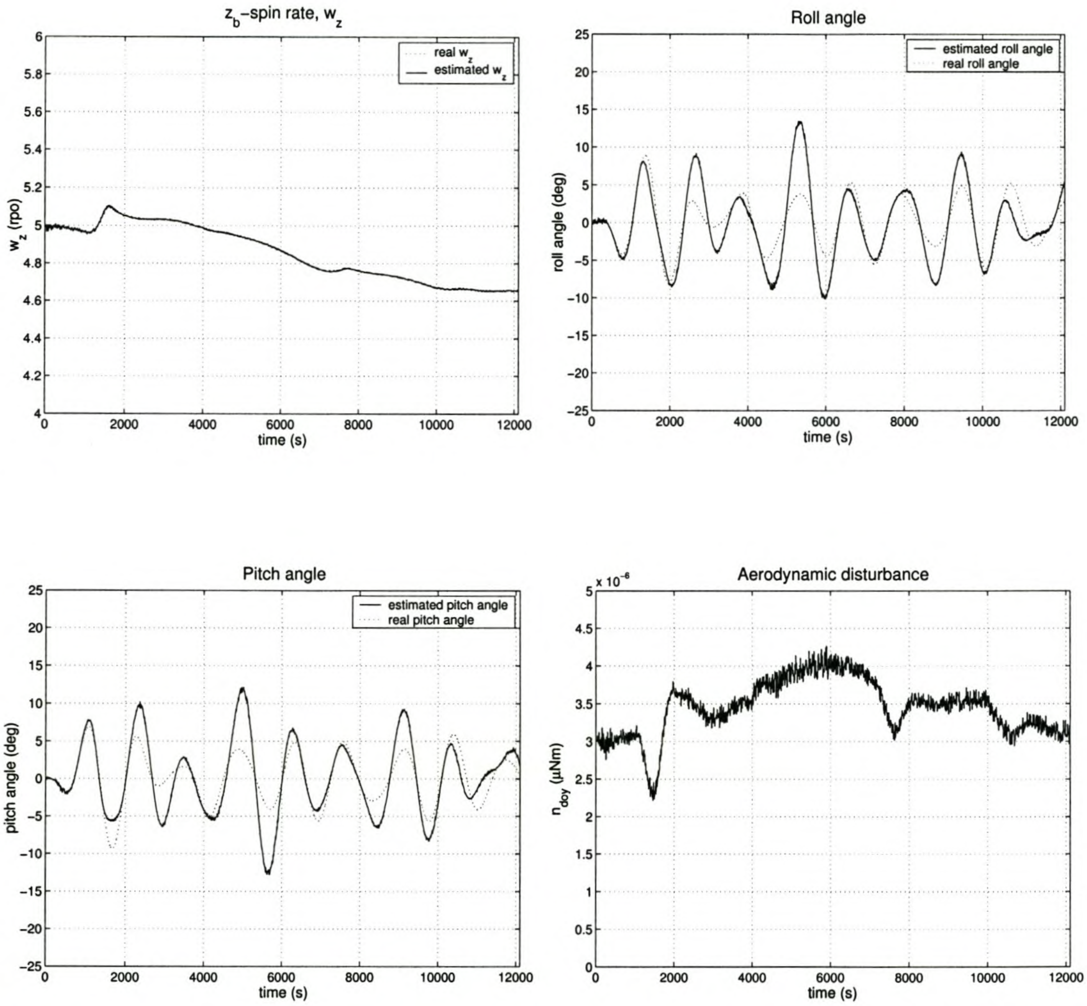


Figure 3.9: *Convergence performance of the horizon/magnetometer EKF from an unknown disturbance torque.*

3.4 Horizon/Magnetometer EKF Error Analysis

The first step in explaining the bad behaviour of this sensor combination is to identify the origin of the large attitude errors. Three possible causes could be identified:

1. Inaccuracy of the magnetometer measurements because of modelling errors
2. Calculation errors in roll, pitch and yaw angle calculations due to large roll and pitch rotations
3. Magnetometer measurements that are only available every 10 seconds

The contribution to the total attitude error of each of the possible causes was determined through simulation, and is summarized in Table 3.1. It is clear from Table 3.1 that the main contributors are the calculation errors in the roll, pitch and yaw angle calculations from the horizon sensor and magnetometer measurements. The Euler angle calculations from the mathematical models of Section 2.4.2 and Section 3.2.5 are only accurate for small roll and pitch angles, as it does not consider large roll and pitch rotations.

Possible solutions to improve the accuracy of the EKF will be to decrease the

Error Source	**RMS error contribution (deg)**
None	0.0314
Magnetometer measurements at 10 seconds sampling time	0
Calculation errors in roll & pitch angle calculation	0.1922
Calculation errors in yaw angle calculation	5.2051
Magnetometer modelling errors	0.0993
Total	5.2988

Table 3.1: *Error analysis results of the horizon/magnetometer EKF.*

size of the roll and pitch angles, to upgrade the mathematical models or to determine a model of the calculation errors. The error model can then be used to correct the calculated roll, pitch and yaw angles. The third option, that of creating an error model, will be considered first.

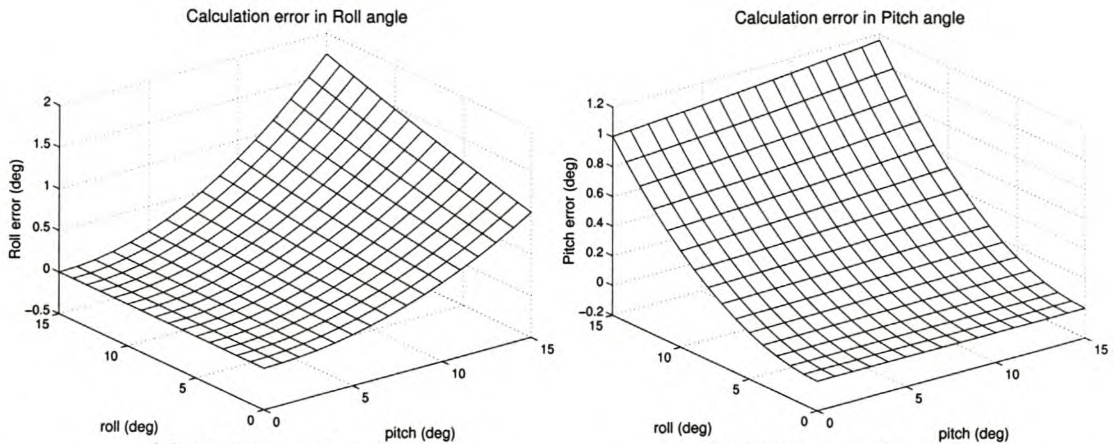
To get an idea of what the errors in the calculated roll, pitch and yaw angles look like and the character of these errors, a simulation program was implemented to calculate the difference between the computed Euler angles and the real Euler angles at specific roll and pitch rotations. The errors in the horizon sensor model were examined first. Figures 3.10 (a) and (b) show the respective errors in the roll and pitch angles for real roll and pitch rotations between 0^o and 15^o. The parabolic character of these errors made it fairly easy to derive a mathematical model. Therefore, the errors in the calculated angles in terms of the real angles are:

$$\begin{aligned} \text{error in roll angle (deg)} &= 4.444e^{-3}\theta^2 + 2.222e^{-4}\psi\theta^2, \quad (3.54) \\ \text{error in pitch angle (deg)} &= 4.444e^{-3}\psi^2 + 5.929e^{-5}\theta\psi^2, \quad (3.55) \end{aligned}$$

where ψ and θ are the respective roll and pitch angles in degrees.

An error model for the yaw calculation from the magnetometer measurements will be more complex to derive, as is evident from Figures 3.11(a), (b), (c), (d). Figures 3.11(a) and (b) plot the error in the yaw angle for roll and pitch rotations between 0^o and 10^o respectively. The real yaw angle was 0^o. Figures 3.11(c) and (d) kept the roll and pitch angles constant but vary the yaw angle between 0^o and 100^o. The roll angle was set at 10^o and pitch at 0^o in Figure 3.11(c), while the roll angle was fixed at 0^o and pitch at 10^o in Figure 3.11(d). From the results displayed in Figure 3.11 it is evident that the errors in the yaw calculation are not only dependent on the size of the roll and pitch angles, but also on the orientation of the geomagnetic field vector relative to the orbit. This dependency will complicate the derivation of the error model, and will, due to time constraints, not be dealt with in this document.

It was also mentioned earlier that the problem of large calculation errors can



(a) Y-Horizon Sensor

(b) X-Horizon Sensor

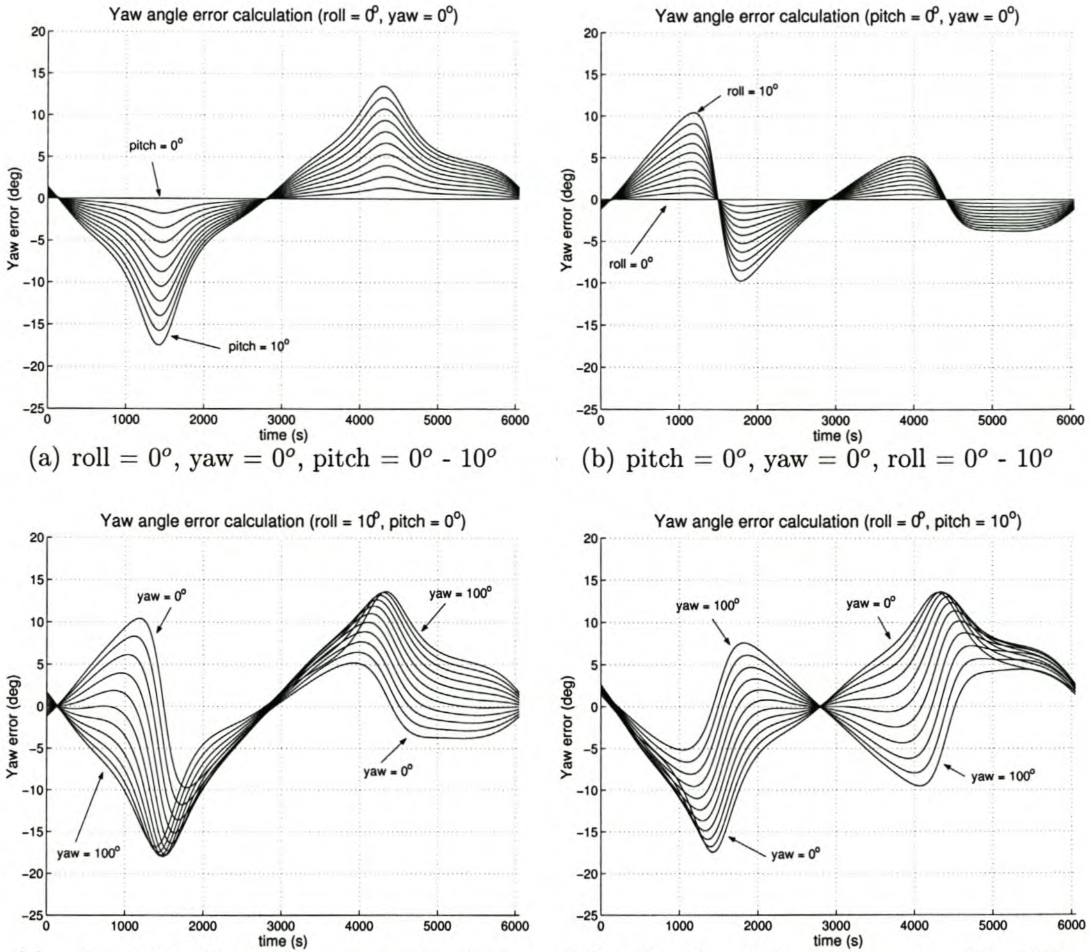
Figure 3.10: *Calculation errors in roll and pitch angle calculations from horizon sensor measurements.*

be solved by keeping the roll and pitch rotations small. The most effective way of achieving this will be to implement an attitude control system like the reaction wheel controller designed by Steyn [7, Chapter 3].

3.5 Conclusion

Two Extended Kalman Filter estimators were implemented to extract full attitude, body rate and disturbance torque information from vector observations. The first EKF used only magnetometer measurements and modelling data from a geomagnetic field model to obtain the filter innovations. Attitude errors of less than 1^o were obtained and convergence times of less than an orbit.

The second EKF investigated a new sensor combination consisting of two horizon sensors and a magnetometer. Measurements of the earth's horizon from the horizon sensors were used to calculate the roll and pitch rotations of the satellite, while measurements from the magnetometer were used to calculate the yaw angle rotation. From these a measurement innovation vector could be obtained, while the estimated Euler angles were used to construct a modelled innovation vector.



(a) roll = 0^o, yaw = 0^o, pitch = 0^o - 10^o

(b) pitch = 0^o, yaw = 0^o, roll = 0^o - 10^o

(c) roll = 10^o, pitch = 0^o, yaw = 0^o - 100^o

(d) roll = 0^o, pitch = 10^o, yaw = 0^o - 100^o

Figure 3.11: *Calculation errors in yaw angle calculation from magnetometer measurements.*

Although the filter was able to extract full attitude, body rate and disturbance torque information from vector observations, one problem was the inaccuracy of the filter during large roll and pitch rotations. The main reason for the inaccurate results of the filter is the calculation errors in the calculated RPY-angles from the sensor measurements. Possible solutions for the problem were presented in Section 3.4, but due to time constraints, it could not be thoroughly tested during this thesis.

The horizon/magnetometer combination presented a more flexible way of combining sensors with incomplete measurements to obtain suitable measurement data for attitude determination systems. It proved that different sensor combinations are possible, although some accuracy problems may occur that will need special attention.

Chapter 4

Integration of the Magnetometer and the Horizon/Magnetometer EKFs

4.1 Introduction

The previous chapter presented two Extended Kalman Filter estimators with different sensor combinations as input. The EKFs were implemented and tested separately, but can be integrated into one EKF by switching between the different sensor combinations that were used to obtain a measurement vector. Figure 4.1 shows the typical setup of such an integration. The sensor combinations consist of either a magnetometer (COMBINATION2) in the one instance or a combination of horizon and magnetometer (COMBINATION1) measurements. Measurements from the magnetometer are available throughout the orbit, while measurements from the horizon/magnetometer conjugation are dependent on a valid field of view and an illuminated horizon. Therefore, if data are available from the horizon sensors, COMBINATION1 will be used, otherwise the measurement vector will consist of COMBINATION2.

Differences in the accuracy, noise characteristics and modelling errors of the different sensor combinations will affect the performance of the EKF during switching.

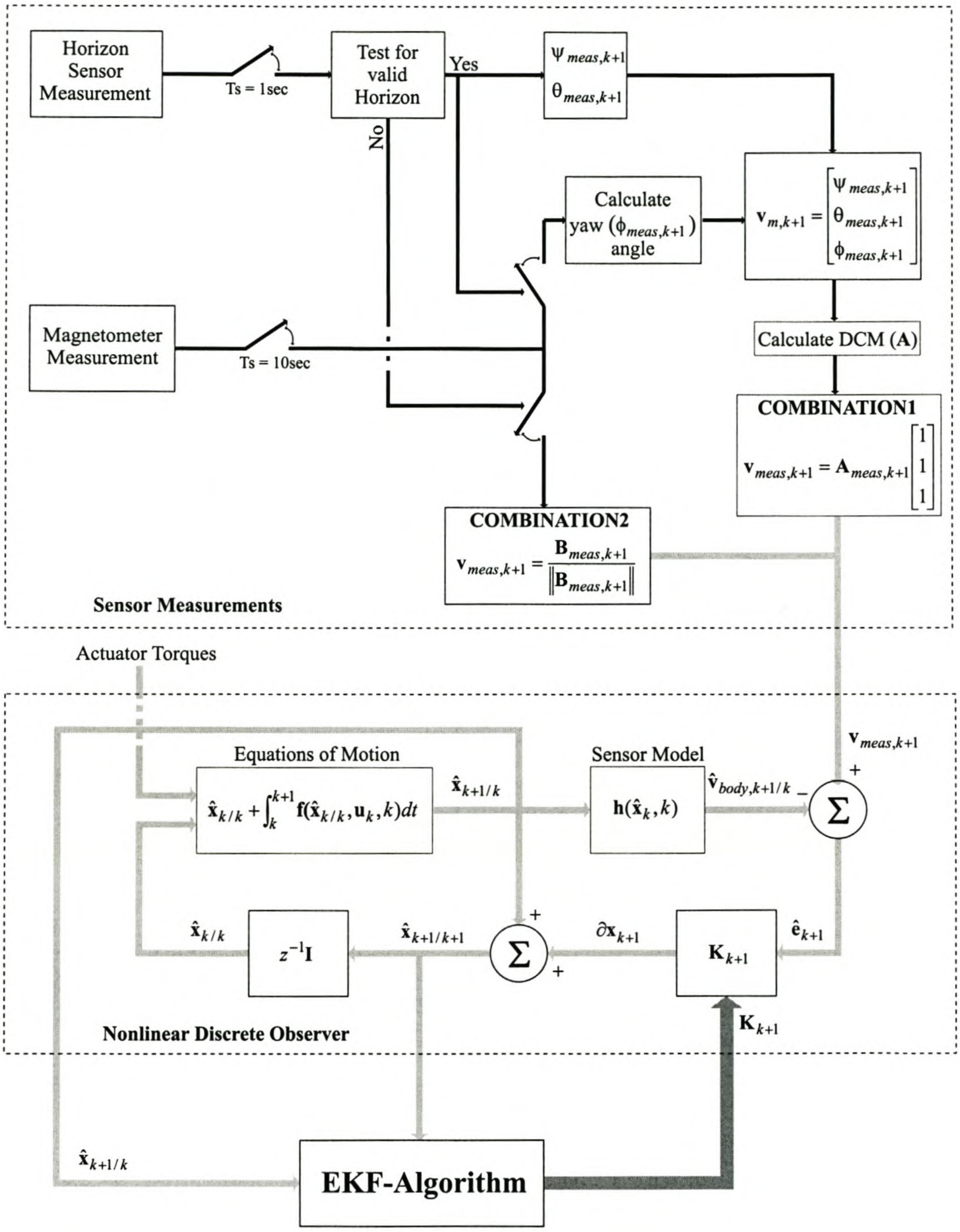

Horizon Sensor Measurement
Ts = 1sec
Test for valid Horizon
Yes
No
$\psi_{meas,k+1}$
$\theta_{meas,k+1}$
Calculate yaw ($\phi_{meas,k+1}$) angle
$\mathbf{v}_{m,k+1} = \begin{bmatrix} \psi_{meas,k+1} \\ \theta_{meas,k+1} \\ \phi_{meas,k+1} \end{bmatrix}$
Calculate DCM (A)
Magnetometer Measurement
Ts = 10sec
COMBINATION1
$\mathbf{v}_{meas,k+1} = \mathbf{A}_{meas,k+1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
COMBINATION2
$\mathbf{v}_{meas,k+1} = \frac{\mathbf{B}_{meas,k+1}}{\|\mathbf{B}_{meas,k+1}\|}$
Sensor Measurements
Actuator Torques
Equations of Motion
$\hat{\mathbf{x}}_{k/k} + \int_k^{k+1} \mathbf{f}(\hat{\mathbf{x}}_{k/k}, \mathbf{u}_k, k)dt$
$\hat{\mathbf{x}}_{k+1/k}$
Sensor Model
$\mathbf{h}(\hat{\mathbf{x}}_k, k)$
$\hat{\mathbf{v}}_{body,k+1/k}$
−
$\mathbf{v}_{meas,k+1}$
+
Σ
$\hat{\mathbf{e}}_{k+1}$
$\mathbf{K}_{k+1}$
$\partial\mathbf{x}_{k+1}$
+
+
Σ
$\hat{\mathbf{x}}_{k+1/k+1}$
$z^{-1}\mathbf{I}$
$\hat{\mathbf{x}}_{k/k}$
Nonlinear Discrete Observer
$\mathbf{K}_{k+1}$
$\hat{\mathbf{x}}_{k+1/k}$
EKF-Algorithm

Figure 4.1: *Block diagram of integrated EKF.*

The dynamic behaviour of the EKF during these switching periods was therefore investigated by means of simulation during a series of case studies.

The following assumptions regarding the orbit, sun-orbit and earth were made. Both the orbit and the earth were assumed to be circular. The orbit was positioned at an inclination of 45^o. The sun was set in a fixed position in its orbit, a fixed fraction of the earth was therefore illuminated during the satellite's orbit.

The satellite was assumed to have a z_b-axis, ω_z, spin of 5 rpo, roll and pitch angles of 0^o and an offset-sinusoidal disturbance torque, n_{doy}, of 3 μNm. For the estimator full attitude and angular rate knowledge was assumed, except where explicitly stated otherwise. The initial filter covariance matrix, **P**, the system covariance matrix, **Q**, and the noise covariance matrix, **R**, were chosen depending on the sensor combination used.

To improve the convergence performance of the Magnetometer EKF from Chapter 3 at different starting-points in the orbit, some of the **Q**-matrix values were slightly increased. The consequence of this was an increase in the bandwidth of the EKF, which again means a less effective filtering of the measurement noise, as is evident from the results displayed in Figure 4.2. The estimated values of Figure 4.2 will be used as a reference for the results obtained during the case studies in this chapter. The initial **P**-matrix and **Q**-matrix values for the respective sensor combinations were therefore:

$$\begin{aligned} \mathbf{P}_o &= \begin{bmatrix} 1e^{-1} & 1e^{-1} & 1e^{-1} & 1e^{5} & 1e^{5} & 1e^{5} & 1e^{5} & 5e^{-5} \end{bmatrix}, \\ \mathbf{Q} &= \begin{bmatrix} 2e^{-4} & 2e^{-4} & 2e^{-4} & 2e^{1} & 2e^{1} & 2e^{1} & 2e^{1} & 5e^{-8} \end{bmatrix}, \end{aligned} \tag{4.1}$$

for the magnetometer, and:

$$\begin{aligned} \mathbf{P}_o &= \begin{bmatrix} 4e^{-1} & 4e^{-1} & 4e^{-1} & 2e^{4} & 2e^{4} & 2e^{4} & 2e^{4} & 5e^{-5} \end{bmatrix}, \\ \mathbf{Q} &= \begin{bmatrix} 2e^{-5} & 2e^{-5} & 2e^{-5} & 5e^{1} & 5e^{1} & 5e^{1} & 5e^{1} & 10e^{-8} \end{bmatrix}, \end{aligned} \tag{4.2}$$

for the horizon/magnetometer combination.

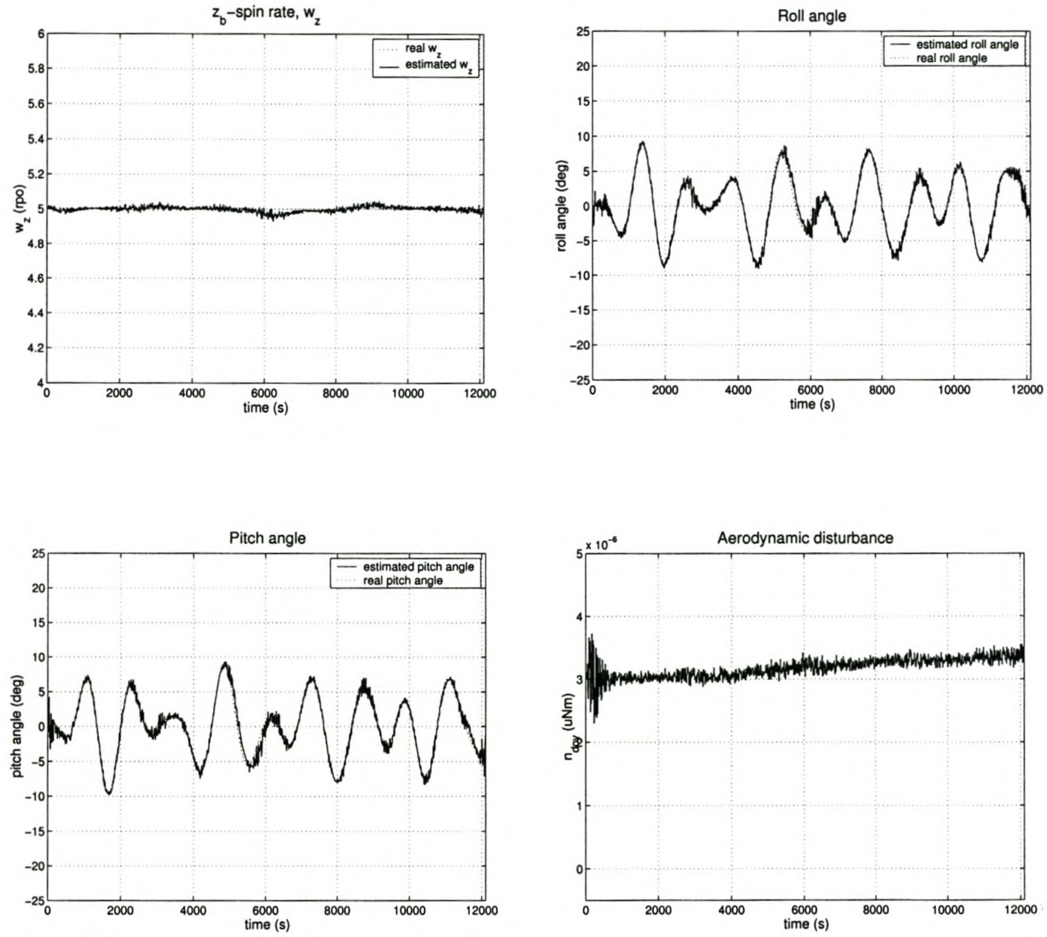


Figure 4.2: *Estimated rate and attitude of magnetometer EKF with improved* **Q**-*matrix.*

4.2 Systematic Analysis of the EKF's Behaviour During Switching

The systematic analysis of the behaviour of the integrated EKF during switching was done by means of case studies. Table 4.1 gives a summary of the different situations that were examined.

Case studies 1 to 4 looked at the overall performance and stability of the EKF. Case Study 1 employed the EKF in a noiseless environment to test the integration, while Case Study 2 examined the effects of noise on the performance and stability of the EKF. Case Study 3 started the simulation at different stages in the orbit to make sure stability occurs throughout the orbit. Finally Case Study 4 investigated the effect of the sun's relative position to the earth on the behaviour of the EKF. The results of Case Study 2 were used as a reference for case studies 3 and 4.

Case studies 5 and 6 looked at the convergence characteristics of the EKF by using different initial values and by employing unknown disturbance torques on the satellite during its flight. Finally Case Study 7 did a simple robustness test on the EKF by using different **Q**-matrix values.

The switching stages in each figure displaying simulation results will be indicated by two abbreviations: *M/HM* - indicates a switch from the magnetometer to the horizon/magnetometer combination, and *HM/M* - indicates a change from the horizon/magnetometer combination to the magnetometer. The simulations were executed over a period of two orbits ($\approx$ 12000 sec).

Case	Starting point (Longitude)	Sun position (Longitude)	Noise	$\mathbf{P}, \mathbf{Q}, \mathbf{R}$	Initial $\hat{\omega}_z$ (rpo)	Disturbance
1	0^o	0^o	No	Eq. 4.1, Eq. 4.2	ω_z	No
2	0^o	0^o	Yes	Eq. 4.1, Eq. 4.2	ω_z	No
3	180^o	0^o	Yes	Eq. 4.1, Eq. 4.2	ω_z	No
4	0^o	90^o	Yes	Eq. 4.1, Eq. 4.2	ω_z	No
5(a)	0^o	0^o	Yes	Eq. 4.1, Eq. 4.2	$0.9 \times \omega_z$	No
5(b)	180^o	0^o	Yes	Eq. 4.1, Eq. 4.2	$0.9 \times \omega_z$	No
5(c)	90^o	90^o	Yes	Eq. 4.1, Eq. 4.2	$0.9 \times \omega_z$	No
6(a)	0^o	0^o	Yes	Eq. 4.1, Eq. 4.2	ω_z	4500 sec
6(b)	0^o	0^o	Yes	Eq. 4.1, Eq. 4.2	ω_z	6500 sec
7(a)	0^o	0^o	Yes	Eq. 4.2	$0.9 \times \omega_z$	No
7(b)	0^o	0^o	Yes	Eq. 4.1	$0.9 \times \omega_z$	No

Table 4.1: *Summary of case studies.*

4.2.1 Case Study #1: Integration

The overall performance and stability of the integrated EKF were investigated in the first case study. Ideal sensors were assumed. The simulation was started at longitude = 0^o in the orbit. The sun's relative position to the earth was also fixed at longitude = 0^o. Only the first and last part the satellite's orbit was therefore illuminated, as is shown in Figure 4.3. Figure 4.4 show the estimated attitude state values during the simulation period.

Case Study Result

The EKF performed worse during the magnetometer period than during the horizon/magnetometer period as indicated by markers **A**. Comparison with Figure 4.2 indicates that this may be caused by the integration.

A possible reason for this is the large estimation errors that occur just before the switch from the horizon/magnetometer to the magnetometer combination, due to the inaccuracy of the horizon/magnetometer EKF. During the magnetometer period, the EKF must therefore converge from the initial estimation error.

The EKF, however, recovered quickly from these errors and reached convergence before the following switching stage.

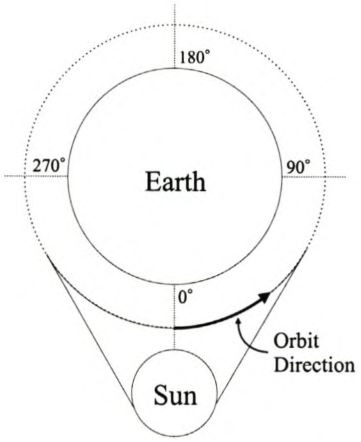


Figure 4.3: *Environmental setup for Case Study 1.*

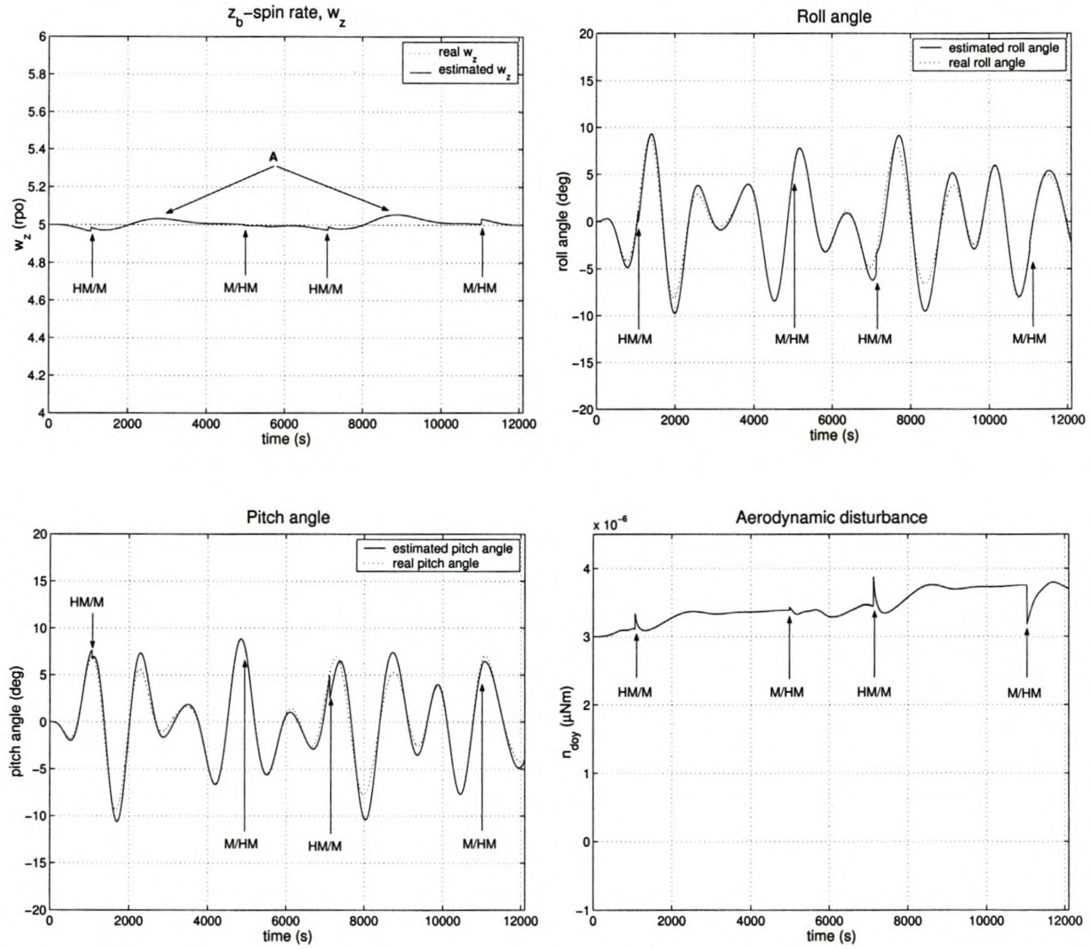


Figure 4.4: *Estimated rate and attitude of the integrated EKF in a noiseless environment.*

4.2.2 Case Study #2: Effect of Noise

Case Study 2 looked at the effect of noise on the EKF's performance and stability. The same noise models were used as defined in Chapter 3 (Section 3.3). The environmental setup (starting-point and sun's position) was the same as that of Case Study 1 (Figure 4.3). Figure 4.5 display the results.

Case Study Result

The only effect the added noise had, was a slight decrease in the EKF's accuracy as would be expected. The overall performance of the EKF compared good with that of Case Study 1.

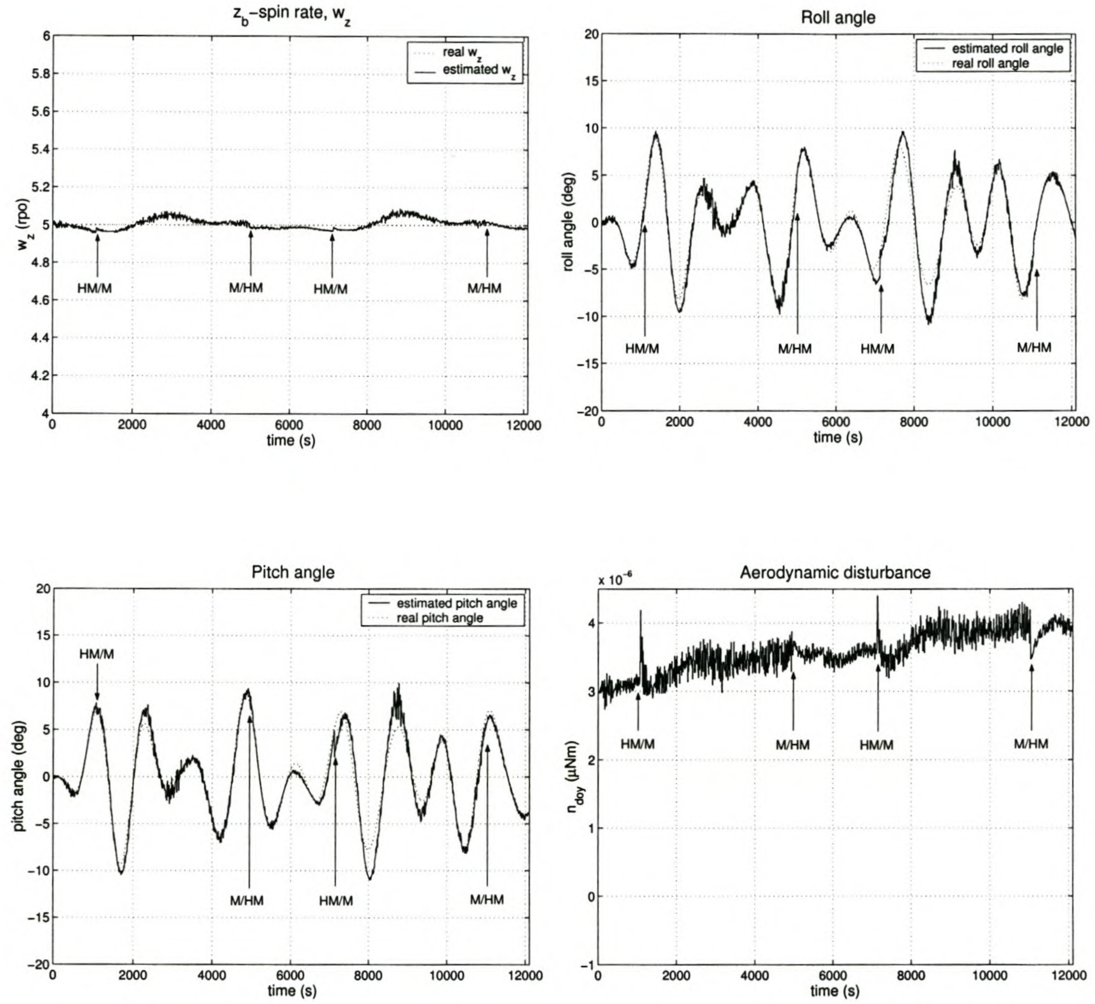


Figure 4.5: *Estimated rate and attitude of the integrated EKF with noise added to sensor measurements.*

4.2.3 Case Study #3: Different Starting Position

To be sure the previous cases studies' results were true independent of where the simulation was started in the orbit, the starting-point has been changed to longitude = 180^o. The sun's relative position to the earth was kept at longitude = 0^o (See Figure 4.6). The simulation therefore started in an unilluminated part of the earth. Figure 4.7 display the results.

Case Study Result

The behaviour of the EKF compared good with that of Case Study 2. The same deviation from the true values as seen in case studies 1 and 2 were found (marker **B**). Comparison with the results in Figure 4.4 showed that the deviation always occur at the same position in the orbit, around longitude = 180^o.

The results not only confirmed that the deviation are most probably due to estimation errors just before switching from the horizon/magnetometer to the magnetometer combination, but do also indicate a possible worst case scenario for the integrated EKF.

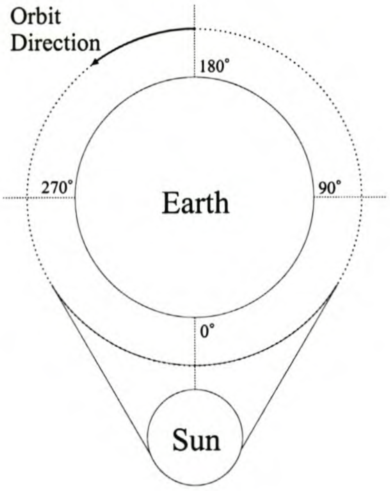


Figure 4.6: *Environmental setup for Case Study 3.*

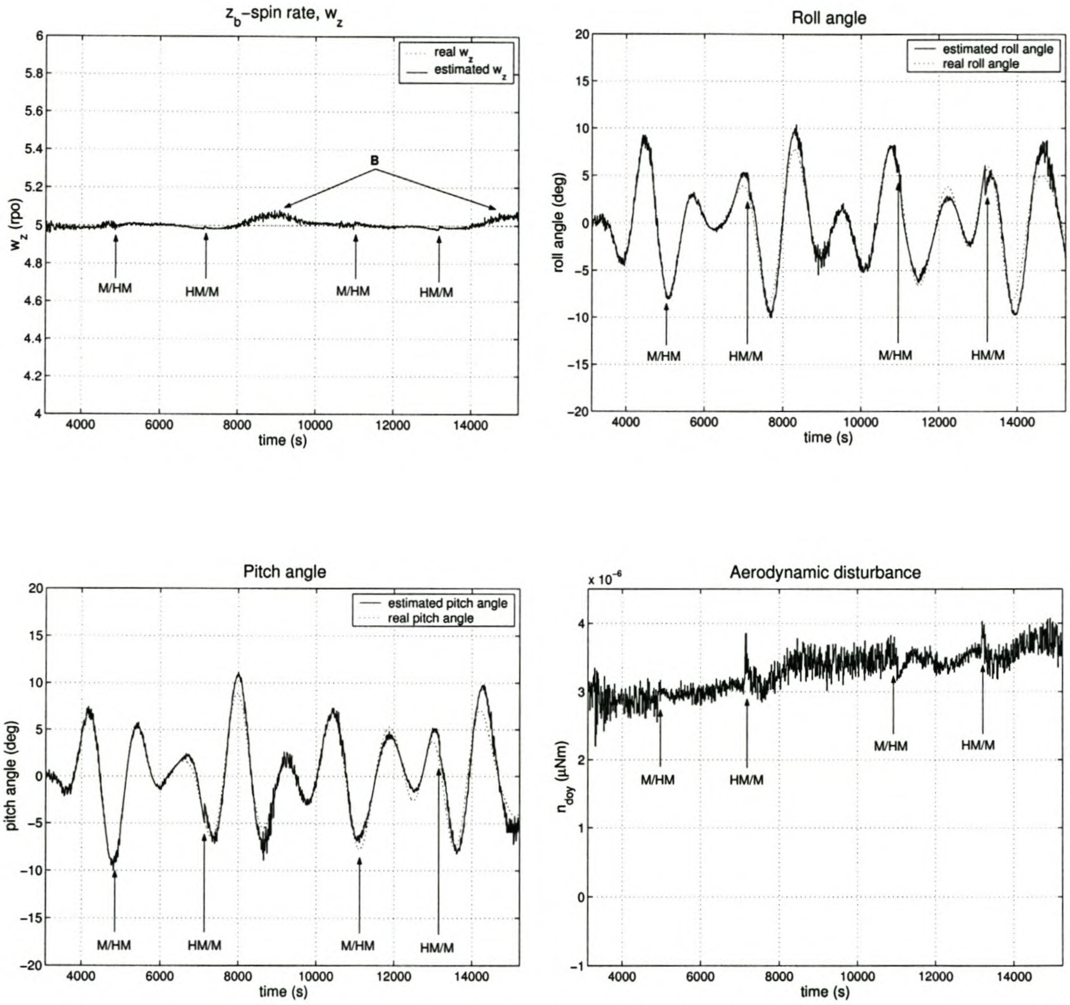


Figure 4.7: *Estimated rate and attitude of the integrated EKF with the simulation started at longitude = 180°.*

4.2.4 Case Study #4: Different Sun Position

Up until now, the sun was fixed in the same position. The same fraction of the earth was therefore illuminated in each case study. In a real space environment the sun's position will change continuously. To make sure the EKF's stability is independent from the sun's position, it has been shifted to longitude = 90^o. The horizon/magnetometer EKF was therefore running during a different part of the orbit. Figure 4.8 shows the sunlit part of the earth. The starting-point of the simulation was at longitude = 0^o.

Case Study Result

Figure 4.9 display the results obtained during the simulation. The results differ from those of the previous studies in that large deviations occurred during the time the horizon/magnetometer combination was running. From Figure 3.7, however, it is evident that this behaviour is not the result of the integrated EKF setup, but due to the calculation errors in the RPY-angle calculations.

Surprisingly the EKF performed better during the magnetometer period than in the first three case studies, thus indicating that the magnetometer combination was less sensitive to estimation errors from the horizon/magnetometer combination. This may mean that the effects of the switching on the EKF's performance are also orbit dependent.

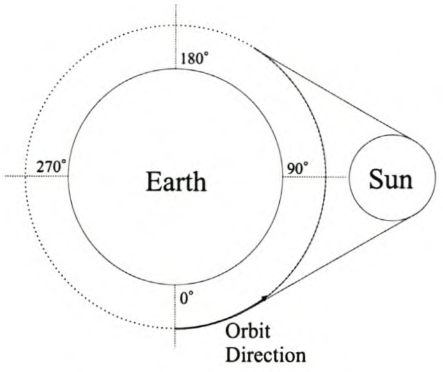


Figure 4.8: *Environmental setup for Case Study 4.*

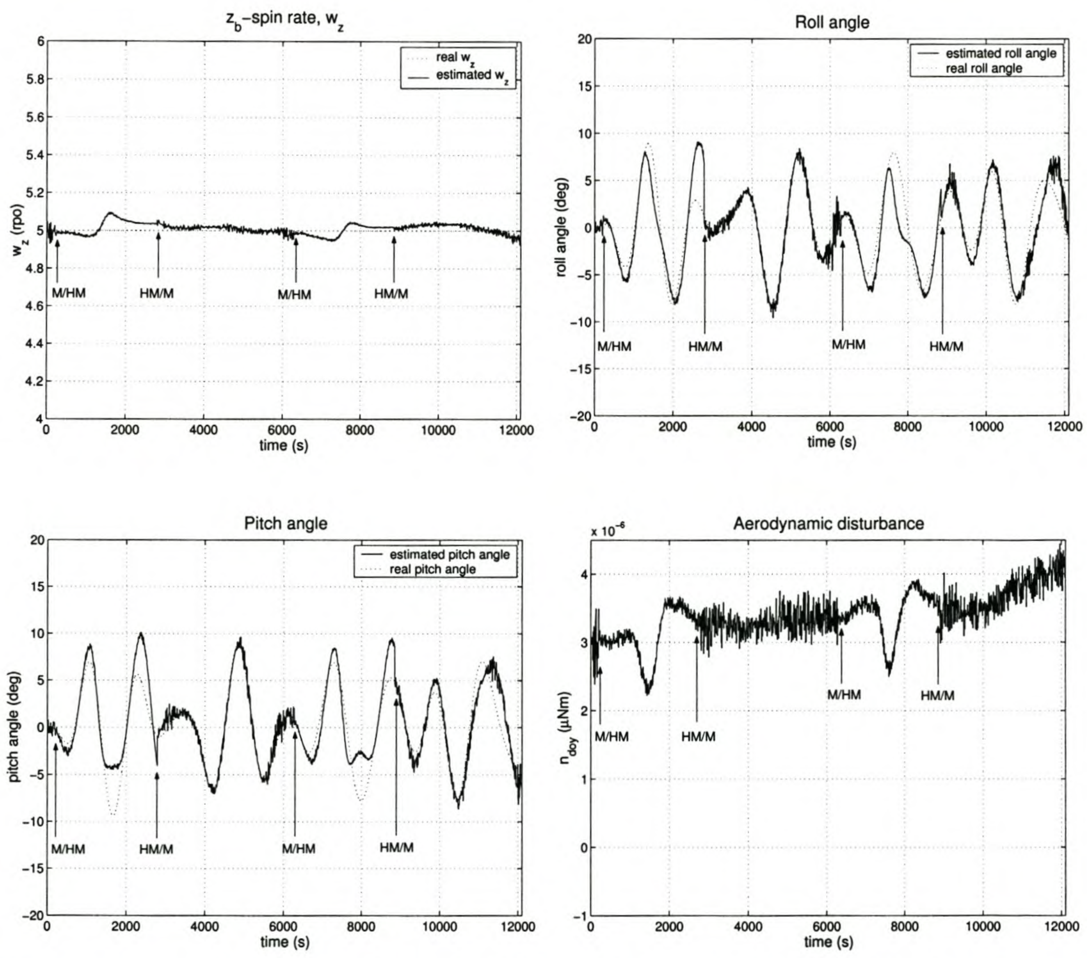


Figure 4.9: *Estimated rate and attitude of the integrated EKF with the sun's postion at longitude* $= 90^o$.

4.2.5 Case Study #5: Different Initial Values

The tests done in this case study investigated the convergence characteristics of the EKF from unknown initial state values. Three tests were run under three different environmental scenario's identified as possible problem areas during the previous case studies. Figure 4.10 show the different setups.

- The first test (a), starting at longitude = 0^o, investigated the convergence of the EKF when switching from the horizon/magnetometer to the magnetometer combination. The sun's position were therefore at longitude = 0^o.
- The second test (b) looked at the performance of the EKF when switching takes place from the magnetometer to the horizon/magnetometer combination. The simulation therefore started at longitude = 180^o with the sun's position still at longitude = 0^o.
- The last test (c) started in an area where the horizon/magnetometer combination showed a possible problem area as identified in Case Study 4. The simulation started at longitude = 90^o with the sun's relative position to the earth also at longitude = 90^o.

The estimator's initial values were all set to zero, except for the angular rate, $\hat{\omega}_z$, which was initialized to 90% of the real value of 5 rpo.

Case Study Result

Figures 4.11 to 4.13 show the results of tests (a), (b) and (c) respectively. In all three situations the EKF converged within an orbit. The behaviour of the EKF, during the convergence time, was dependant on where in the orbit the simulation started. This dependency was also found during the testing of the separate EKFs in Chapter 3, and are therefore not simply a characteristic of the integrated EKF.

The behaviour of the EKF during the switching stages further agreed with the results obtained in the case studies performed this far.

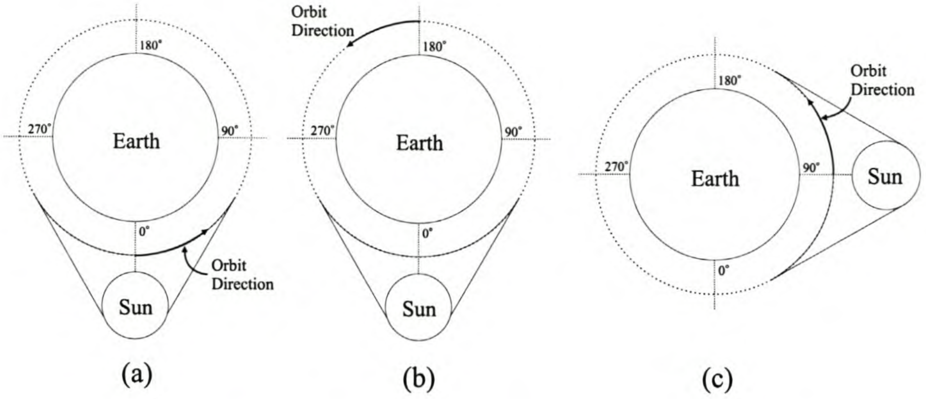


Figure 4.10: *Environmental setup for Case Study 5.*

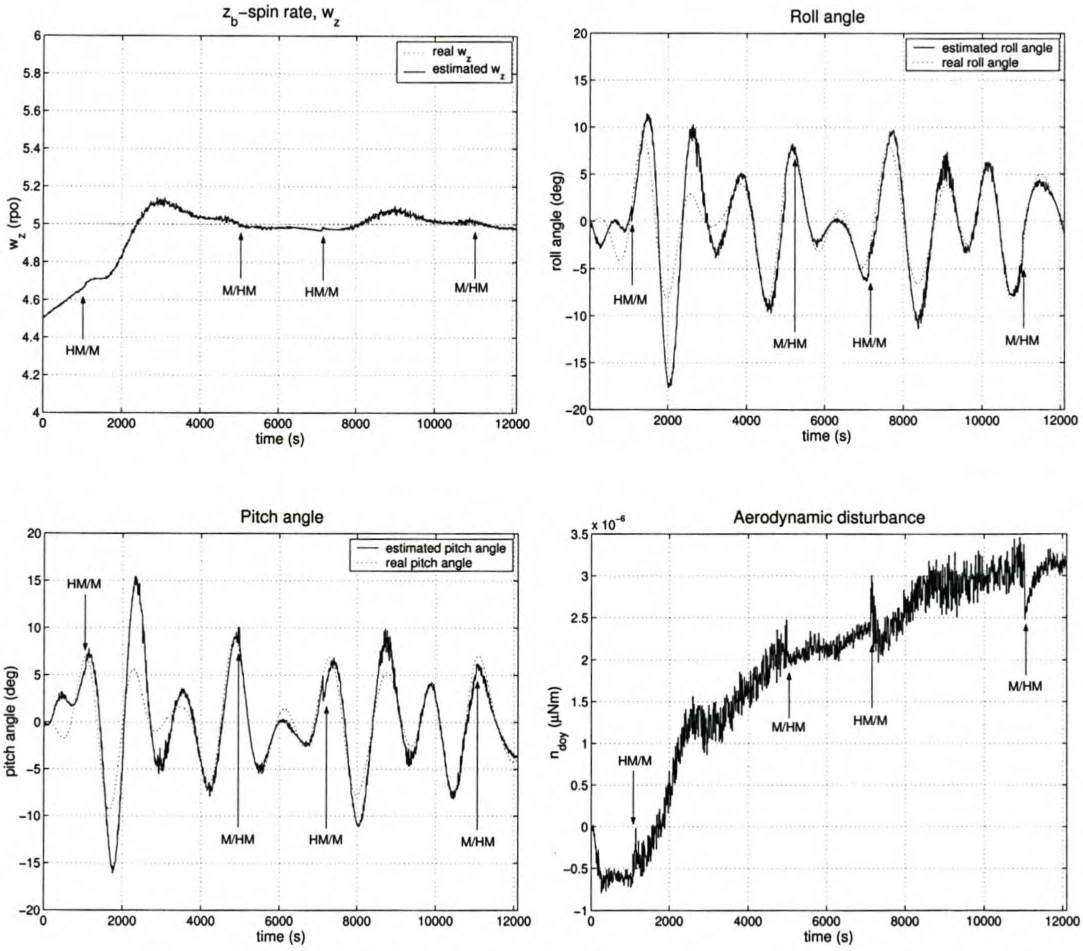


Figure 4.11: *Estimated rate and attitude of the integrated EKF with initial rate value 90% of real value. Simulation started at longitude* = 0^o, *with the sun's position at longitude* = 0^o.

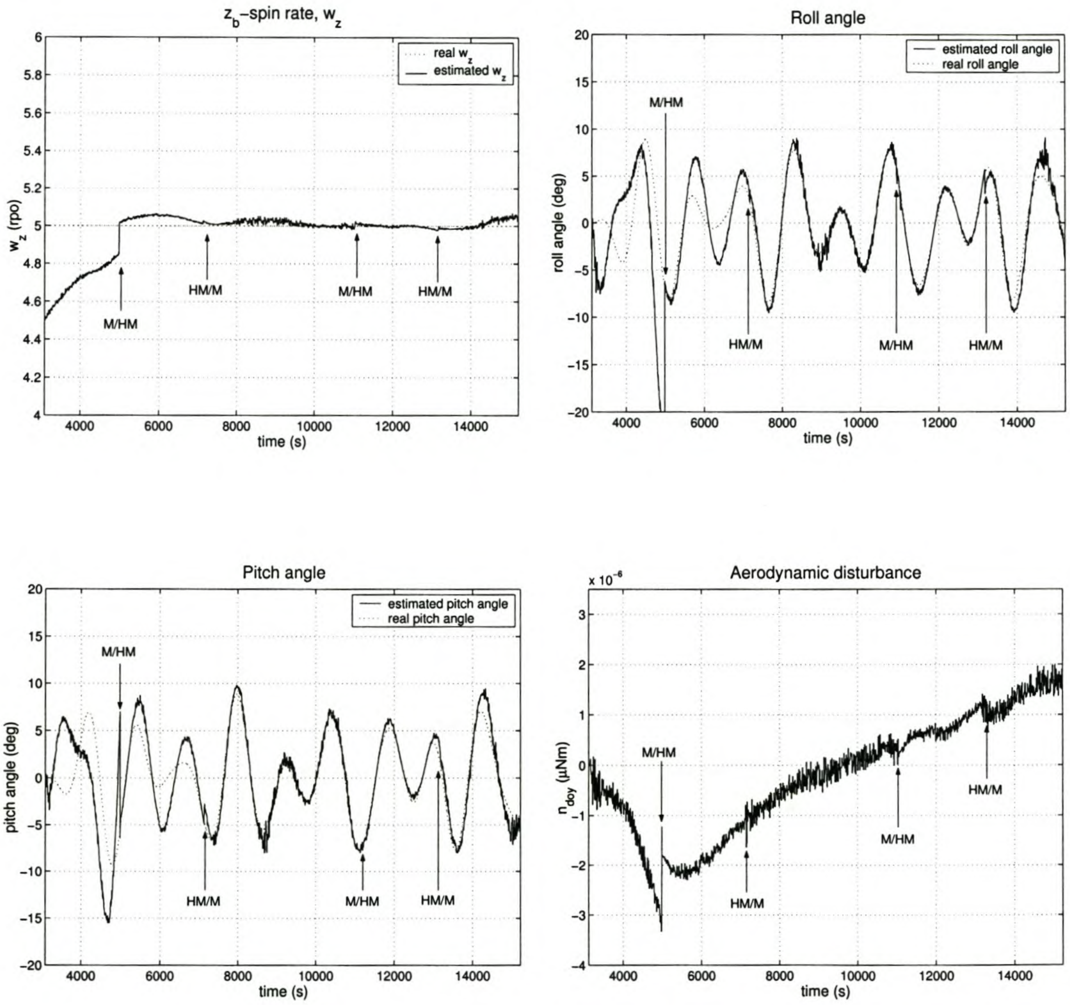


Figure 4.12: *Estimated rate and attitude of the integrated EKF with initial rate value 90% of real value. Simulation started at longitude* = 180^o, *with the sun's position at longitude* = 0^o.

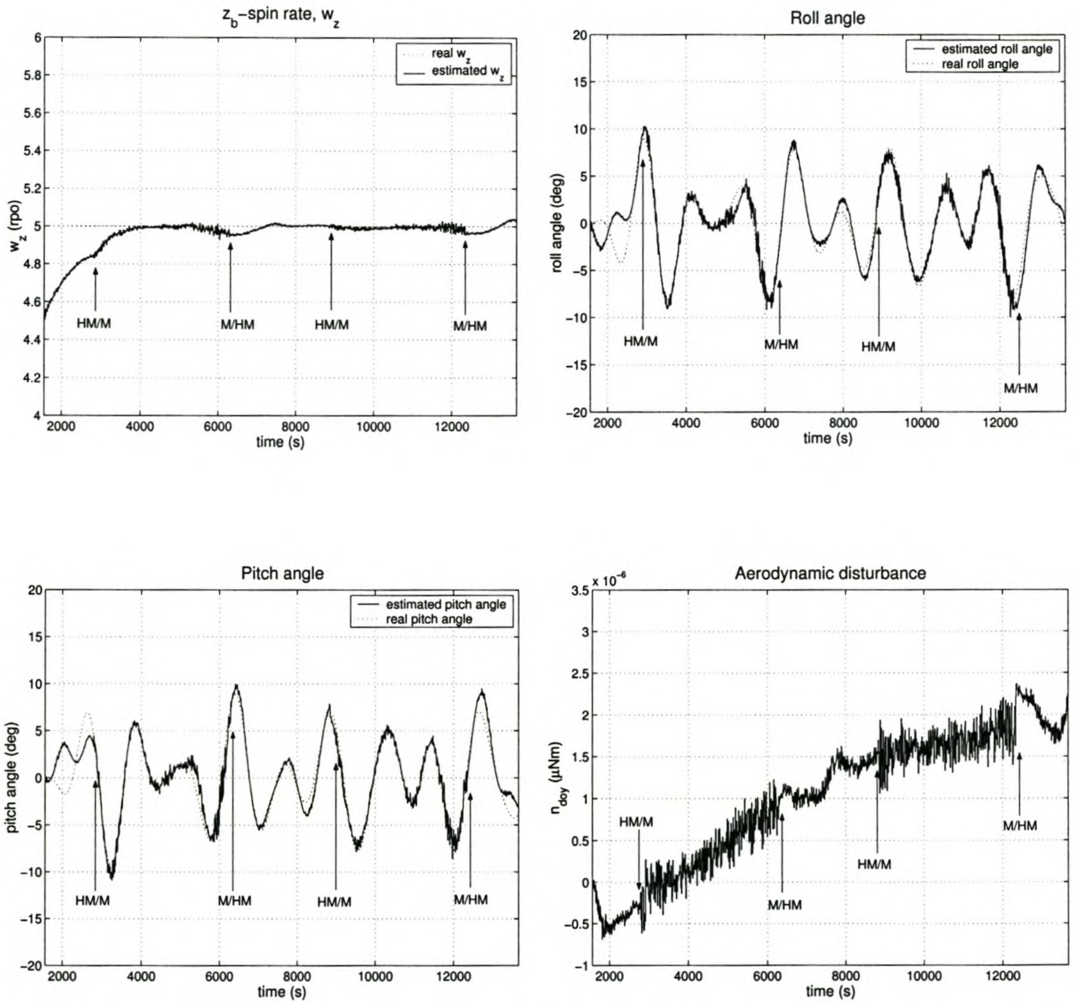


Figure 4.13: *Estimated rate and attitude of the integrated EKF with initial rate value 90% of real value. Simulation started at longitude = 90°, with the sun's position at longitude = 90°.*

4.2.6 Case Study #6: Disturbance

Another performance characteristic worth investigating, was the ability of the integrated EKF to converge from an unknown disturbance torque induced on the satellite. The effect of the disturbance torque was to decrease the satellite's attitude rate to 90% of its initial value.

Two critical stages would be just before the switch from the magnetometer to the horizon/magnetometer combination and back. The first simulation (a) therefore applied the disturbance torque at time, t = 4500 sec, and the second (b) at time, t = 6500 sec. Both simulations started at longitude = 0^o, with the sun's position also at longitude = 0^o. See Figure 4.14.

Case Study Result

Convergence in simulation (a) (Figure 4.15) was reached within an orbit, while simulation (b) (Figure 4.16) could only reach convergence after 6000 sec (more than an orbit).

The EKF in simulation (b) reacted very slowly to the induced disturbance torque at first. Faster tracking was only managed after switching to the magnetometer. Figure 3.9 showed that the horizon/magnetometer EKF has a poor convergence performance during a disturbance torque, thus explaining the poor tracking obtained in simulation (b).

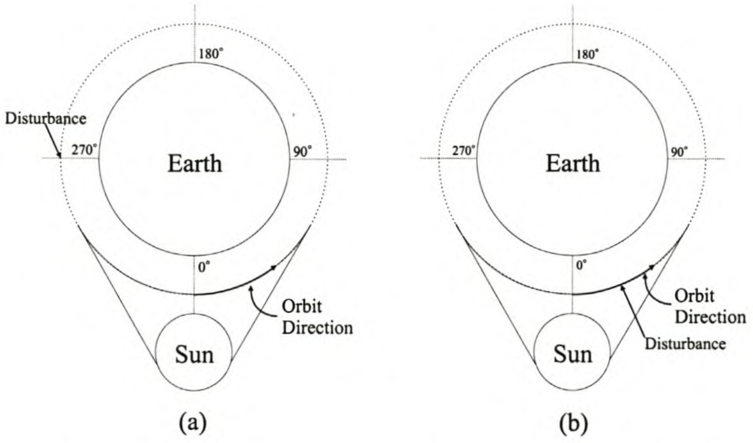


Figure 4.14: *Environmental setup for Case Study 6.*

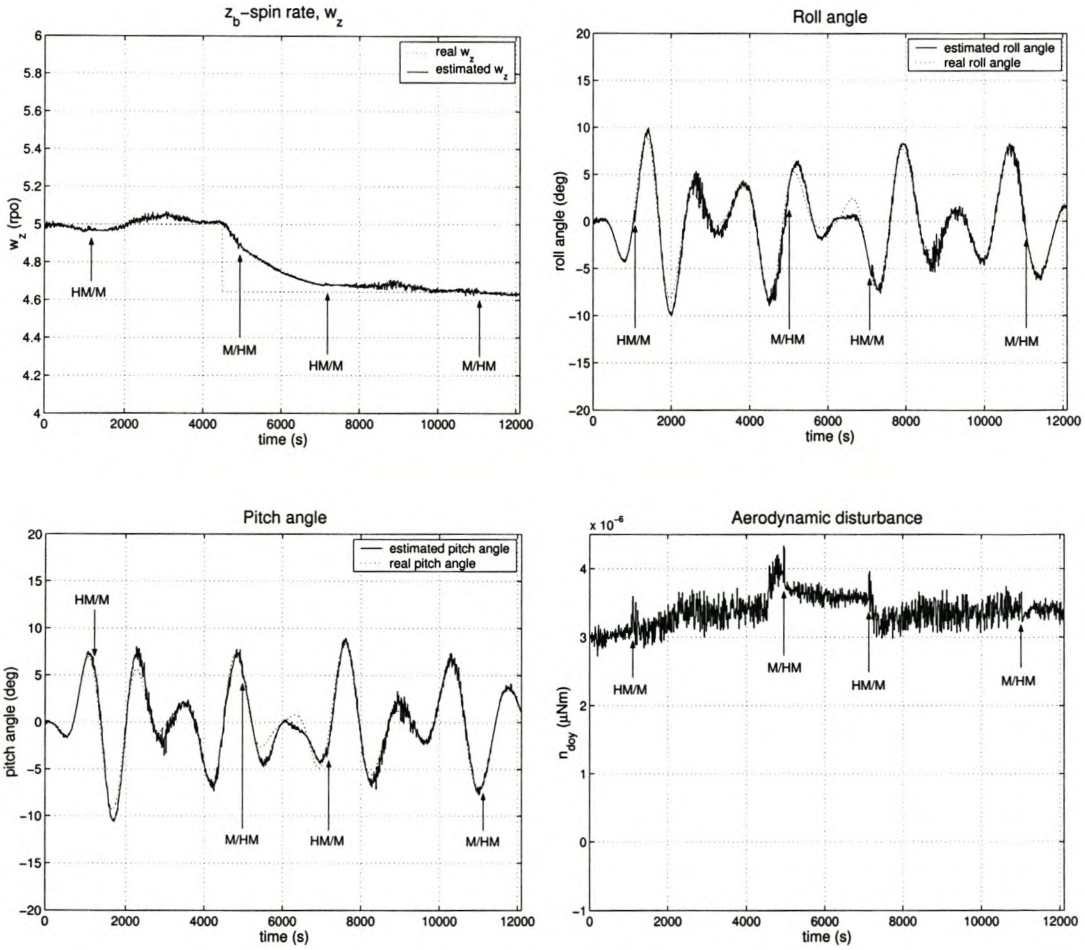


Figure 4.15: *Estimated rate and attitude of the integrated EKF with an unknown disturbance torque employed at t = 4500 sec.*

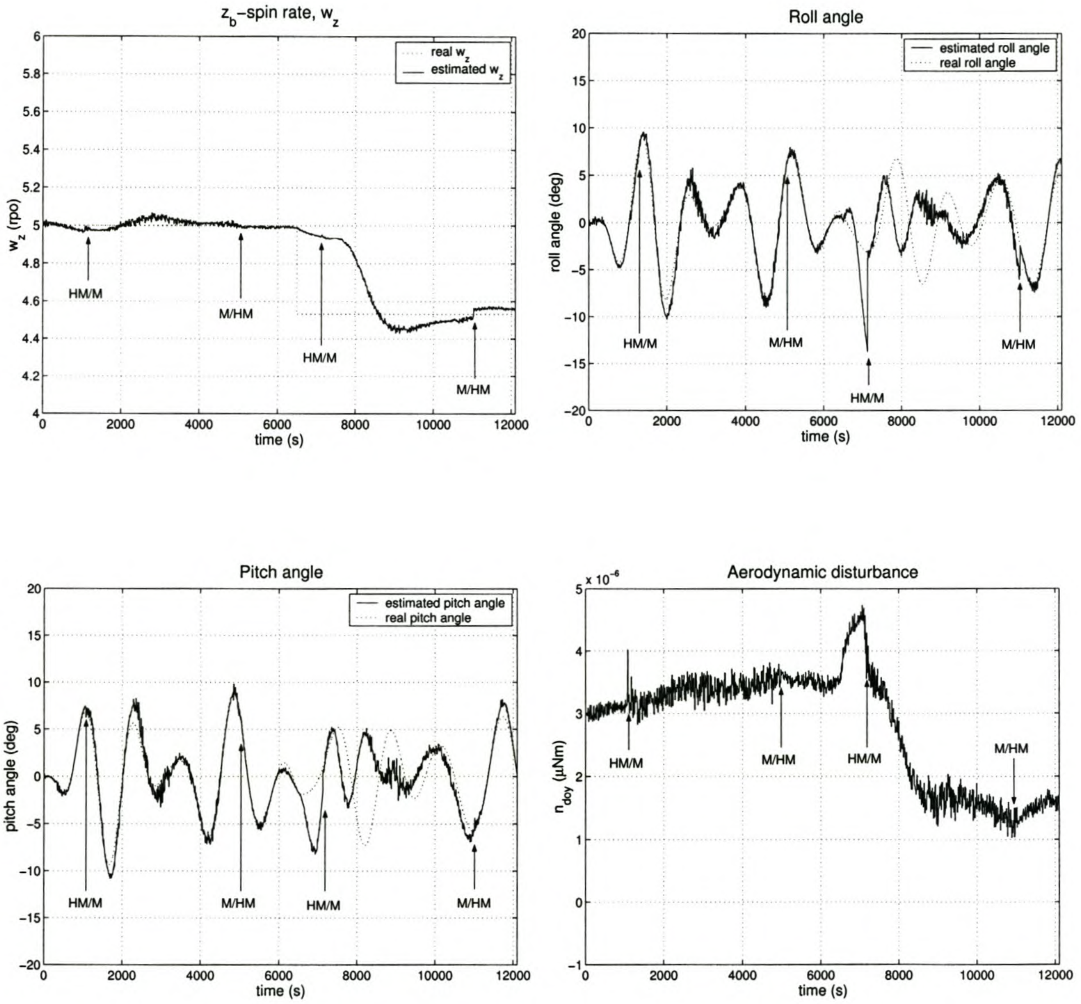


Figure 4.16: *Estimated rate and attitude of the integrated EKF with an unknown disturbance torque employed at t = 6500 sec.*

4.2.7 Case Study #7: Robustness Test

Finally, a simple robustness test was employed on the EKF. The robustness of a filter is an indication of the sensitivity of that filter to unknown disturbance torques and noise in the system dynamics. One goal of filter tuning is to achieve maximum accuracy in the light of the unknown disturbance torques and measurement noise. This is possible through the proper selection of the system noise covariance matrix, **Q**, and the measurement noise covariance matrix, **R**. The magnitudes of **Q** and **R** are representative of the expected disturbance inputs and measurement noise in the system.

Q and **R**, therefore, determine the tradeoff between the tracking of disturbance noise and the filtering of measurement noise. In the presented EKF system the measurement noise was known; a meaningful value could therefore be assigned to **R**. The disturbance torque level was based on the magnitude of possible attitude control torques in the system. These disturbances was, however, not always known. **Q** was therefore rather chosen in a pragmatic manner to optimize the filter's performance.

In the previous case studies, the respective system covariance matrixes, **Q**, for the different sensor combinations were used. This case study used the same **Q** throughout the simulation. Two tests were performed. Both started in an illuminated part of the earth at longitude $= 0^o$ (See Figure 4.17). The first (a), however used the **Q**-matrix weights of Equation 4.2 and the second (b) used the **Q**-matrix weights of Equation 4.1. In both cases the initial estimator angular rate, $\hat{\omega_z}$, was set to 90% of the real rate.

Case Study Result

Figure 4.18 display the results of simulation (a). Comparison of Figure 4.18 with that of study 5(a) (Figure 4.11) shows an improvement in the performance of the EKF, especially during the magnetometer period.

This result can be best explained considering the above discussion on filter tuning.

For smaller **Q**-matrix values, less disturbance noise are expected in the system. The result of this is that the EKF will fail to converge during large disturbance noise, but measurement noise will be filtered more effectively. The effect of the decreased **Q**-matrix is therefore evident in Figure 4.18. The performance of the EKF during the time the magnetometer was running was less noisy as was also found in Chapter 3, Figure 3.4. The problem with this setup, however, was that it experienced convergence problems during different starting-points in the orbit.

For larger **Q**-matrix values, more disturbance noise are expected in the system. The EKF, however, will also be more sensitive to measurement noise. This is evident from the results of simulation (b) as displayed in Figure 4.19. The EKF's performance are noisier, but also less accurate, indicating that **Q** may be too large.

The results obtained suggest that the optimal point of performance of the EKF may not yet have been reached. It further indicates that tuning of the EKF as an integrated setup may improve the overall performance of the EKF. It will, however, depend on both the performance specifications of the estimator and the noise characteristics of the different sensor combinations.

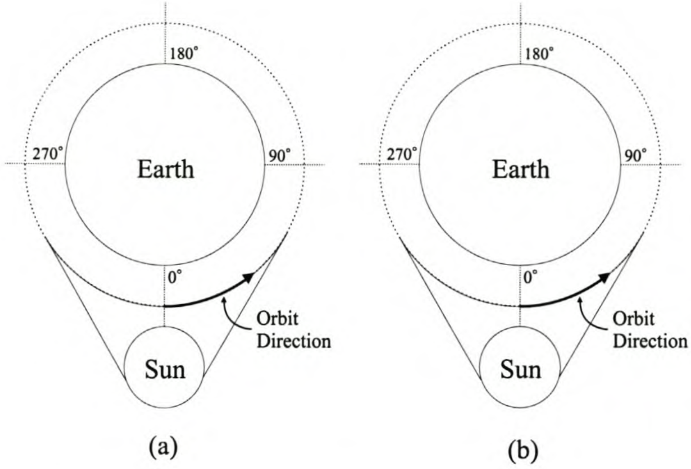


Figure 4.17: *Environmental setup for Case Study 7.*

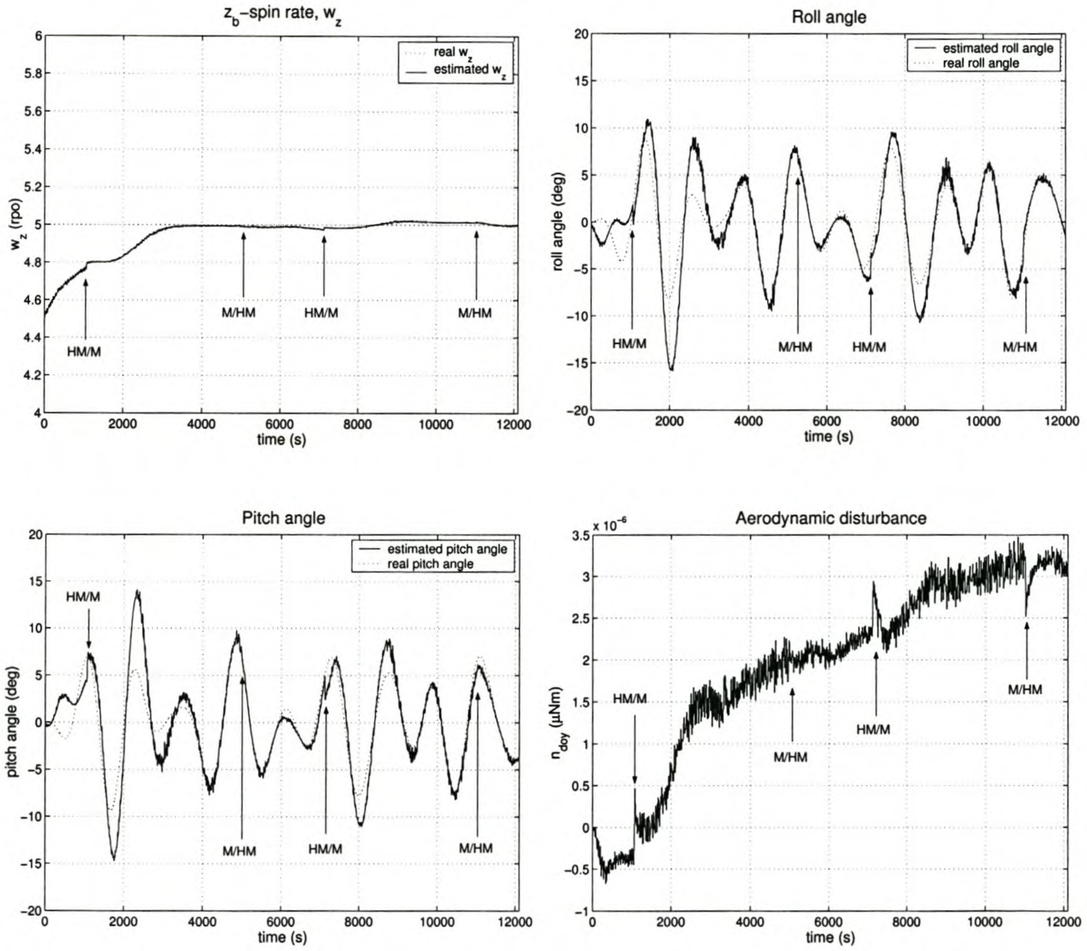


Figure 4.18: *Estimated rate and attitude of the integrated EKF with the* **Q** *matrix values of Equation 4.2.*

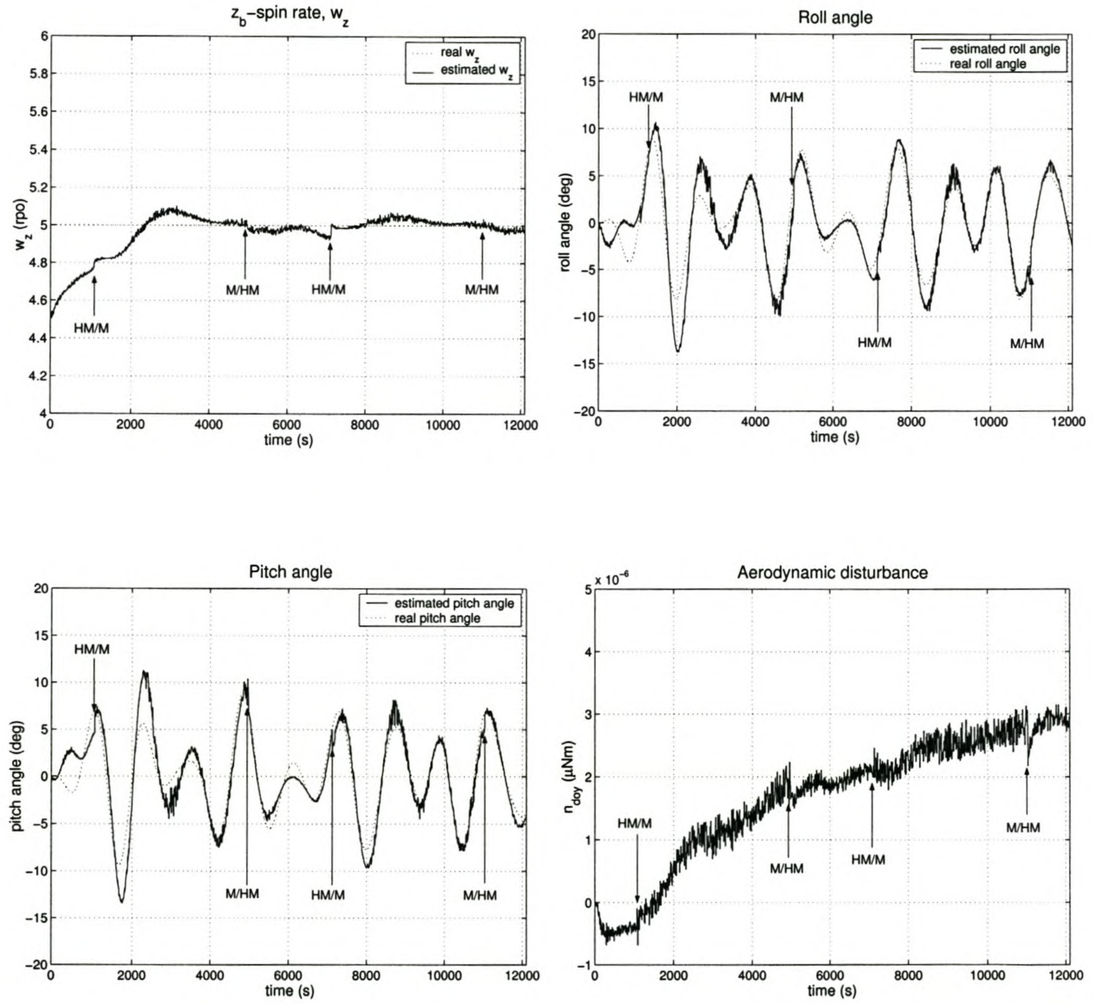


Figure 4.19: *Estimated rate and attitude of the integrated EKF with the* **Q** *matrix values of Equation 4.1.*

4.3 Summary of Case Study Results

A systematic analysis of the behaviour of an integrated Extended Kalman filter, during switching between different sensor combinations as input, was done by means of case studies. Two different sensor combinations were used. The first contained only a magnetometer and the second two horizon sensors and a magnetometer.

The integrated EKF performed stable during switching stages, and reached convergence times of less than an orbit in most of the convergence tests. The degree in which the EKF's performance was affected by the switching was largely dependent on both the state values just before the switching and the orbital position of the satellite and the sun.

Switching from the horizon/magnetometer combination to the magnetometer affected the EKF's behaviour the most. A possible reason for this is the inaccuracy of the horizon/magnetometer EKF, as was experienced in Chapter 3 during the simulation tests. During the magnetometer stage, the EKF thus had to recover from these errors.

To conclude: Two separately implemented Extended Kalman filters were successfully integrated by switching between different sensor combinations.

4.4 Hardware Implementation

The intention was to do a hardware implementation of the EKF setup presented in this chapter. The necessary hardware to enable interfacing between the different sensors and the PC were developed. A DSP Microcomputer, the ADSP-2189M, were used for the interfacing between the PC and the two CCD horizon sensors. The necessary drivers to enable transmission of data from the horizon sensors to the PC were written and tested. A separate board were developed and the necessary drivers written to transmit magnetometer data to the PC.

The hardware implementation could not be completed due to time constraints, and will, therefore, not be further discussed here. A block diagram of the full design can be found in Appendix B, as well as a discussion of the work that was completed.

Chapter 5

Conclusion

5.1 Summary of Contributions

Two areas were investigated in this thesis.

- The first investigated the use of sensors with incomplete measurements in an attitude determination system that determine the full attitude state of a satellite.
- The second considered the integration of Extended Kalman Filters with different vector observations.

5.1.1 EKF with Incomplete Sensor Combinations

Two EKFs were presented. The first used only magnetometer measurements to obtain vector observations of the attitude. The second used a combination between a sensor with incomplete measurements (horizon sensor) and a magnetometer. Horizon sensors can provide attitude information in only one axis. By positioning two horizon sensors orthogonal to each other, attitude information in two axes can be attained. Measurements from a magnetometer were used to provide attitude knowledge in the third axis.

Mathematical models used the measurements from the horizon sensors and the magnetometer sensor to calculate the roll, pitch and yaw angles of the satellite, thus obtaining a vector observation of the attitude. From this vector observation full attitude knowledge could be estimated.

The inaccuracy of the estimated values was, however, unacceptably large. The main reason for this was calculation errors in the RPY-angle calculations due to large roll and pitch rotations. Various possibilities to decrease these calculation errors were discussed, but could not be implemented and tested. It was, however, proved that an estimation of the full satellite attitude state is obtainable from a horizon/magnetometer sensor combination.

5.1.2 Integration of EKFs

Instead of separate EKF implementations with different sensors or sensor combinations, an integrated EKF was also implemented. Different sensor combinations can provide attitude information to the EKF by switching between the different measurement vectors. The second contribution of this thesis was a systematic analysis of the dynamic behaviour of the EKF during these switching stages.

Two sensor combinations were used. The first consisting of only magnetometer measurements and the second of horizon and magnetometer measurements. The analysis was done by a series of case studies. The simulation results showed that stability was obtained through each switching phase. Small disturbance effects mostly occurred during switching from a less accurate to a more accurate estimation of the state values. The EKF, however, was able to recover quickly from these disturbances.

Bibliography

[1] J. A. A. Engelbrecht, "A hardware-in-the-loop simulation facility for the attitude determination and control system of SUNSAT," Master's thesis, University of Stellenbosch, Stellenbosch, South Africa, December 1995.

[2] W. H. Steyn, "Attitude control algorithms and simulation programs for low earth orbit spacecraft," Master's thesis, University of Surrey, United Kingdom, December 1990.

[3] K. S. Miller, *An Introduction To Kalman Filtering with Applications.* Robert E. Krieger Publishing Company, 1987.

[4] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering.* John Wiley & Sons, inc, 1983.

[5] M. L. Psiaki, "Autonomous low-earth-orbit determination from magnetometer and sun sensor data," *Journal of Guidance, Control and Dynamics*, vol. 22, pp. 296–304, March - April 1999.

[6] M. L. Psiaki, "Autonomous orbit and magnetic field determination using magnetometer and star sensor data," *Journal of Guidance, Control and Dynamics*, vol. 18, pp. 584–592, May - June 1995.

[7] W. H. Steyn, *A Multi-Mode Attitude Determination and Control System for Small Satellites.* PhD thesis, University of Stellenbosch, Stellenbosch, South Africa, December 1995.

[8] J. R. Wertz, *Spacecraft Attitude Determination and Control.* D. Reidel Publising Company, 1978.

[9] M. J. Jacobs, "A low cost, high precision star sensor," Master's thesis, University of Stellenbosch, Stellenbosch, South Africa, December 1995.

[10] D. G. Zill and M. R. Cullen, *Advanced Engineering Mathematics.* Jones and Bartlett Publishers, second ed., 1997.

[11] M. L. Psiaki, F. Martel, and P. K. Pal, "Three-axis attitude determination via kalman filtering of magnetometer data," *Journal of Guidance, Control and Dynamics*, vol. 13, pp. 506–514, May - June 1990.

Appendix A

IGRF Model of Geomagnetic Field

The following theory was extracted from Appendix H in Wertz [8]. The earth's magnetic field, **B**, can be expressed as the gradient of a scalar potential, V, i.e:

$$\mathbf{B} = -\nabla V. \tag{A.1}$$

V can be conveniently represented by a series of spherical harmonics:

$$V(r,\theta,\phi) = a\sum_{n=1}^{k}\left(\frac{a}{r}\right)^{n+1}\sum_{m=0}^{n}(g_n^m \cos m\phi + h_n^m \sin m\phi)P_n^m(\theta), \tag{A.2}$$

where

a	=	the equatorial radius of the earth (6371.2 km),
g_n^m and h_n^m	=	gaussian coefficients of the IGRF model,
P	=	legendre functions (Scmidt normalized),
r	=	geocentric distance,
θ	=	coelevation(south positive), and
ϕ	=	east longitude from Greenwich.

The Gaussian coefficients are determined empirically by a least squares fit to measurements of the magnetic field. A set of these coefficients constitutes a model of the field. Table A.1 gives a set of the coefficients from the time 1990 to 1995. With these coefficients and a definition of the associated Legendre functions, P_n^m , the magnetic field at any point in space can be calculated from Equations A.1

and A.2 as:

$$\begin{aligned}
B_r &= -\frac{\partial V}{\partial r} \\
&= \sum_{n=1}^{k}\left(\frac{a}{r}\right)^{n+2}(n+1)\sum_{m=0}^{n}(g^{n,m}\cos m\phi + h^{n,m}\sin m\phi)P^{n,m}(\theta), \qquad (A.3) \\
B_\theta &= \frac{-1}{r}\frac{\partial V}{\partial \theta} \\
&= -\sum_{n=1}^{k}\left(\frac{a}{r}\right)^{n+2}\sum_{m=0}^{n}(g^{n,m}\cos m\phi + h^{n,m}\sin m\phi)\frac{\partial P^{n,m}(\theta)}{\partial \theta}, \qquad (A.4) \\
B_\phi &= \frac{-1}{r\sin\theta}\frac{\partial V}{\partial \phi} \\
&= \frac{-1}{\sin\theta}\sum_{n=1}^{k}\left(\frac{a}{r}\right)^{n+2}\sum_{m=0}^{n}m(-g^{n,m}\sin m\phi + h^{n,m}\cos m\phi)P^{n,m}(\theta). \qquad (A.5)
\end{aligned}$$

The coefficients of the IGRF assume that P_n^m is Schmidt normalised. The relation between the Gauss function, $P^{n,m}$, and the Schmidt function, P_n^m, is:

$$P_n^m = S_{n,m}P^{n,m}. \qquad (A.6)$$

The factors $S_{n,m}$ are best combined with the Gaussian coefficients because they are independent of r, θ and ϕ. It is therefore only necessary to calculate them once during a computer run. We define:

$$\begin{aligned}
g^{n,m} &\equiv S_{n,m}g_n^m, \\
h^{n,m} &\equiv S_{n,m}h_n^m. \qquad (A.7)
\end{aligned}$$

The following recursion relations can be derived for $S_{n,m}$:

$$\begin{aligned}
S_{0,0} &= 1, \\
S_{n,0} &= S_{n-1,0}\left[\frac{2n-1}{n}\right] \qquad \text{if} \qquad n \geq 1, \qquad (A.8) \\
S_{n,m} &= S_{n,m-1}\sqrt{\frac{(n-m+1)(\delta_m^1+1)}{n+m}} \qquad \text{if} \qquad m \geq 1.
\end{aligned}$$

$P^{n,m}$ can be obtained from the following recursion function:

$$\begin{aligned}
P^{0,0} &= 1, \\
P^{n,n} &= \sin\theta P^{n-1,n-1}, \qquad (A.9) \\
P^{n,m} &= \cos\theta P^{n-1,m} - K^{n,m}P^{n-2,m},
\end{aligned}$$

where

$$\begin{aligned} K^{n,m} &\equiv \frac{(n-1)^2 - m^2}{(2n-1)(2n-3)} \qquad \text{if} \qquad n > 1, \\ K^{n,m} &\equiv 0 \qquad \text{if} \qquad n = 1. \end{aligned} \tag{A.10}$$

The gradient in Equation A.1 will lead to partial derivatives of P_n^m. We therefore need:

$$\begin{aligned} \frac{\partial P^{0,0}}{\partial \theta} &= 0, \\ \frac{\partial P^{n,n}}{\partial \theta} &= (\sin\theta)\frac{\partial P^{n-1,n-1}}{\partial \theta} + (\cos\theta)P^{n-1,n-1}, \\ \frac{\partial P^{n,m}}{\partial \theta} &= (\cos\theta)\frac{\partial P^{n-1,m}}{\partial \theta} - (\sin\theta)P^{n-1,m} - K^{n,m}\frac{\partial P^{n-2,m}}{\partial \theta}. \end{aligned} \tag{A.11}$$

Also note that:

$$\cos m\phi = cos((m-1)\phi)\cos\phi - \sin((m-1)\phi)\sin\phi, \tag{A.12}$$

$$\sin m\phi = sin((m-1)\phi)\cos\phi + \cos((m-1)\phi)\sin\phi. \tag{A.13}$$

n	m	g (nT)	h (nT)	n	m	g (nT)	h (nT)
1	0	-29775	-	6	2	60	83
1	1	-1851	5411	6	3	-178	78
2	0	-2136	-	6	4	2	-52
2	1	3058	-2278	6	5	17	2
2	2	1693	-380	6	6	-96	27
3	0	1315	-	7	0	77	-
3	1	-2240	-287	7	1	-64	-81
3	2	1246	293	7	2	4	-27
3	3	807	-348	7	3	28	1
4	0	939	-	7	4	1	20
4	1	782	248	7	5	6	16
4	2	324	-240	7	6	10	-23
4	3	-423	87	7	7	0	-5
4	4	142	-299	8	0	22	-
5	0	-211	-	8	1	5	10
5	1	353	47	8	2	-1	-20
5	2	244	153	8	3	-11	7
5	3	-111	-154	8	4	-12	-22
5	4	-166	-69	8	5	4	12
5	5	-37	98	8	6	4	11
6	0	61	-	8	7	3	-16
6	1	64	-16	8	8	-6	-11

Table A.1: *Eigth order IGRF Gaussion Coefficients for EPOCH 1990-1995.*

Appendix B

Hardware Implementation

Figure B.1 shows the block diagram of the intended hardware implementation of the EKF setup described in Chapter 4.

B.1 Horizon Sensor Hardware Setup

The hardware setup of the horizon sensor consisted of a CCD horizon sensor, a DSP microcomputer and a FPGA board. Each part will be discussed shortly.

Horizon Sensor (KLI-2113)

The KLI-2113 is a 2098 × 3 tri-linear CCD Image sensor from Kodak. The hardware layout for the sensor were done by a colleague, Jacques Rossouw.

ADSP-2189M DSP Microcomputer

The ADSP-2189M DSP Microcomputer from Analog Devices was responsible for the signal processing of the horizon sensor measurement data. This involved the filtering of the data and the processing of the data to: 1) test for a valid horizon and 2) obtain the horizon elevation angle.

FPGA

The hardware layout and VHDL code for the FPGA board was designed by Jacques Rossouw as well. The board was responsible for the control of the com-

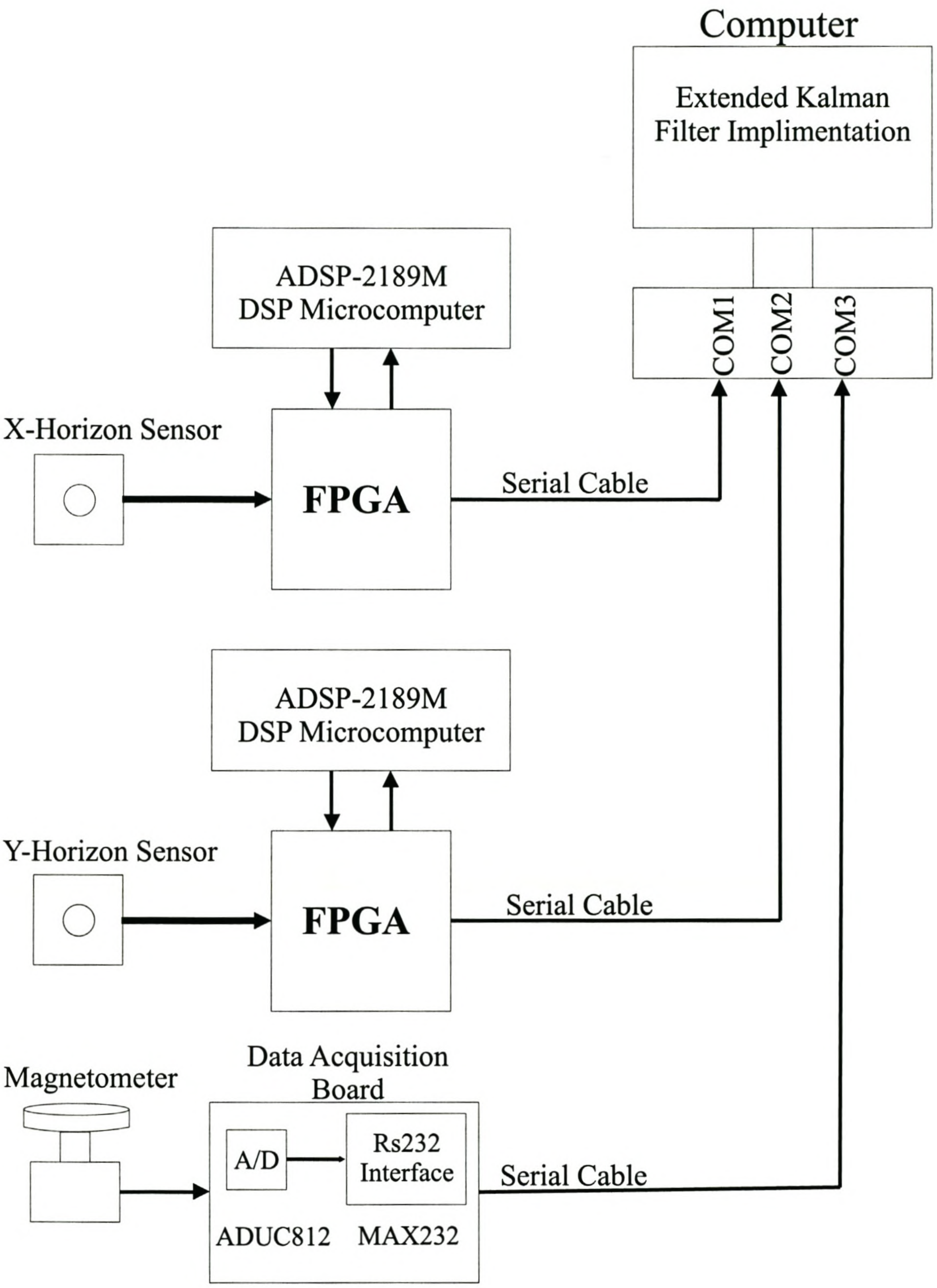


Figure B.1: *Hardware implementation.*

munication between the horizon sensor, the DSP Microcomputer and the computer. It provided the necessary clock signals to clock data from the horizon sensor to the DSP Microcomputer. An onboard UART, the MAX3110E, enabled serial communication between the DSP Microcomputer and the PC.

Results

The necessary software for the DSP Microcomputer to receive data from the horizon sensor was written and successfully tested. The software to send data from the DSP Microcomputer was also written, but not successfully debugged by the end of the thesis. Problems, like data loss, still occurred. Unfortunately, no time was left to write and test the software for the filtering and horizon detection.

B.2 Magnetometer Hardware Setup

An 3-axis analog magnetometer was used to take measurements of the geomagnetic field. The data acquisition board consisted of a 12-bit A/D converter (ADUC812) and a RS232 driver/receiver (MAX232) to enable serial communication with the computer. The software for the ADUC812 to receive data from the magnetometer and to send data to the computer was written and successfully tested.

B.3 Computer

The computer was to receive the measurement data from the horizon sensors and magnetometer. These measurements would be used to obtain vector observations of the attitude necessary for the Extended Kalman Filter attitude determination system. The Extended Kalman filter was implemented successfully in Matlab, but as the hardware and its corresponding software drivers could not be finished, this part of the integration was not tested.

Appendix C

Extended Kalman Filter Code

C.1 Code Listing

C.1.1 Procedures

Initialize.m
Initialization of all global constants. Is called by **main.m** at the beginning of the simulation.

main.m
Main procedure that starts simulation. Call **Initialize.m** to set global constants and set the flags that indicates 10 second intervals and a valid horizon sensor measurement. Call either **kalman2_hor.m**, **kalman_mag.m** or **kalman.m** to start the Kalman Filter procedures.

kalman2_hor.m
Implementation of the horizon sensor EKF. Called by **main.m**.

kalman_mag.m
Implementation of the magnetometer EKF. Called by **main.m**.

kalman.m
Implementation of the integrated EKF. Called by **main.m**.

DrawOrbit.m
Orbit propagator. Call the function **orbit.m** to calculate the necessary orbital parameters.

C.1.2 Functions

orbit.m	Implimentation of orbit propagator in Section 2.2.1.
DCMorbo.m	Calculate DCM from Euler angles. Equation 2.7.
DCMToEuler.m	Calculate Euler angles from DCM. Equations 2.9 to 2.11.
qua2dcm.m	Calculate DCM from quaternion elements. Equation 2.14.
quaternion.m	Calculate quaternion elements from DCM. Equations 2.15 to 2.18.
magfield.m	Implementation of magnetometer measurement model from Section 2.4.1.
horison.m	Implementation of horizon sensor measurement model from Section 2.4.2.
GravityTorque.m	Calculation of Gravity Gradient torque. Equation 2.45.
DisturbanceTorque.m	Calculation of the external disturbance torque. Equation 3.3.
EKF_propag.m	EKF propagation equations between measurements. Equations 3.5 to 3.8.
EKF_correct.m	EKF correction equations at the measurement time. Equations 3.9 to 3.15
PlantmodelPropagation.m	Propagation of the equations of motion. Section 3.2.2.
bodyrate2quaternion.m	Calculate the quaternion propagation. Equation 2.41.
mag_innovation.m	Calculate the innovation vectors for the magnetometer EKF. Equations 3.44 and Equation 3.45.

yaw_innovation.m	Calculate the yaw angle from the magnetometer measurements. Section 3.2.5.
kalfig.m	Called by **kalman2_hor.m**. Plot the results obtained by the horizon sensor EKF.
kalfig_mag.m	Called by **kalman_mag.m**. Plot the results obtained by the magnetometer EKF.
kalfigures.m	Called by **kalman.m**. Plot the results obtained by the integrated EKF.

C.2 Integrated EKF Code

The Extended Kalman Filter algorithm is presented in Chapter 3. The integrated EKF from Figure 4.1 can be implemented in Matlab as shown by procedure **kalman.m**.

C.2.1 Software Code: kalman.m

```
% ------------------------------------------------------------------ %
%     Implimentation of Integrated Extended Kalman Filter:           %
% ------------------------------------------------------------------ %

% ------------------------------------------------------------------ %
%       Initialization of Parameters                                 %
% ------------------------------------------------------------------ %

womean = ORBITn;
%   Initialize real satellite parameter values
roll = SATroll; pitch = SATpitch; yaw = SATyaw;
rollq = roll; pitchq = pitch; yawq = yaw;

% DCM %
Areal = DCMorbo(roll, pitch, yaw);
Aq = DCMorbo(rollq, pitchq, yawq);

% quaternions %
[q1, q2, q3, q4] = quaternion(Areal);
[eq1, eq2, eq3, eq4] = quaternion(Aq);
eqq = [eq1; eq2; eq3; eq4];
qreal = [q1; q2; q3; q4];
```

```
% Sampling Times %
dt = PredictorSampling;
dtPl = PlantSampling;

% Orbital angular rate %
woreal = [SATwox; SATwoy; SATwoz];
wo = [SATwox; SATwoy; SATwoz];

% Inertial angular rate %
wireal = woreal + Areal*[0; -womean; 0];
wi = wo + Aq*[0; -womean; 0];

% Disturbance Torque %
ndoy = 3e-6;
ndoyest = 3e-6;

% System Covariance matrix %
Pw = 1e-1; Pq = 1e5; Pndoy = 5e-5;
Pmatm = diag([Pw Pw Pw Pq Pq Pq Pq Pndoy]);
Pw = 4e-1; Pq = 2e4; Pndoy = 5e-5;
Pmath = diag([Pw Pw Pw Pq Pq Pq Pq Pndoy]);

Pmat = Pmath;

% System Noise Covariance matrix for horizon/magnetometer EKF %
Qw = 2e-5; Qq = 5e1; Qndoy = 10e-8;
Qmat = diag([Qw Qw Qw Qq Qq Qq Qq Qndoy]);

% Horizon Sensor Noise %
noiseh = ((0.5 - rand(1, length(timemat))))/1000;
% Magnetometer Noise %
noisemx = ((0.5 - rand(1, length(timemat)))*0.3/0.5);
noisemy = ((0.5 - rand(1, length(timemat)))*0.3/0.5);
noisemz = ((0.5 - rand(1, length(timemat)))*0.3/0.5);

% Measurement Noise Covariance matrix for horizon/magnetometer EKF %
Rmat = diag([(0.5e-3)^2 (0.5e-3)^2 0.1 ]);

% ------------------------------------------------------------------ %
%       Calling the Kalman Filter function                           %
% ------------------------------------------------------------------ %

Mcounter = 1;

Hcounter = 1;
```

```
for t = 1:length(timemat)

%    if timemat(t) == 6500
%       woreal = [SATwox; SATwoy; 0.9*SATwoz];
%       wireal = woreal + Areal*[0; -womean; 0];
%    end

    if (mod(t,1000) == 0),
        t
    end;


%   Propagation of real plant state variables    %
    [wireal, woreal, qreal, Areal, NGGreal] = ...
                     PlantmodelPropagation(dtPl, wireal, qreal, ndoy);

    % Normalising the q-matrix %
    q_abs = sqrt(sum(qreal.^2));

    qreal(1) = qreal(1)/q_abs; qreal(2) = qreal(2)/q_abs;
    qreal(3) = qreal(3)/q_abs; qreal(4) = qreal(4)/q_abs;

%   The extended Kalman Filter propagation loop %
    [wi, wo, eqq, Aq, NGG, PHI, Fmat, Pupdate] = ...
                       EKF_propag(dtPl, wi, eqq, ndoyest, Pmat, Qmat);

%  If true innovation exist, correct the estimated state variables  %
    if innovtruemag(t) == 1

        Rmat = diag([(0.6e-6)^2 (0.6e-6)^2 (0.6e-6)^2]);
        Qw = 2e-4; Qq = 2e1; Qndoy = 5e-8;
        Qmat = diag([Qw Qw Qw Qq Qq Qq Qq Qndoy]);


    %   Obtain innovation vectors, vmeas, vest
        [Br, Btheta, Bphi, Bc, Borbit] = ...
                magfield(SATAltitude(t,:), timemat(t), TrueAnomat(t));

        ermb = [noisemx(Mcounter) noisemy(Mcounter) noisemz(Mcounter)];
        erm = [0 0 0];

        Bmeas = Borbit' + erm;
        Bmodel = Borbit' + ermb;
    time(Mcounter) = timemat(t);

    [vmeas, vest, vorb] = mag_innovation(Aq, Areal, Bmeas, Bmodel);

%   The extended Kalman Filter correction loop  %
    [wi, eqq, ndoyest, Pmat, Hmat, Kmat, err, delta] = ...
    EKF_correct(wi, eqq, Pupdate, Rmat, Aq, vmeas, vest, vorb, ...
                                                         ndoyest);

    errmmat(Mcounter,:) = err';

    Mcounter = Mcounter + 1;

 elseif innovtruehor(t) == 2

    Rmat = diag([(0.5e-3)^2 (0.5e-3)^2 0.1 ]);
    Qw = 2e-5; Qq = 5e1; Qndoy = 10e-8;
    Qmat = diag([Qw Qw Qw Qq Qq Qq Qq Qndoy]);

%   Get the roll and pitch measurements from the Horison sensor, %
%   the yaw measurements will eventually come from the           %
%   magnetometer                                                 %

    num = 1;
    [XHSangle] = horison(timemat(t), dt, XHSelevation, XHShighel, ...
                 XHSlowel, XHSazimuth, Areal, num);
    [XHSmodel] = horison(timemat(t), dt, XHSelevation, XHShighel, ...
                 XHSlowel, XHSazimuth, Aq, num);

    num = 2;
    [YHSangle] = horison(timemat(t), dt, YHSelevation, YHShighel, ...
                 YHSlowel, YHSazimuth, Areal, num);
    [YHSmodel] = horison(timemat(t), dt, YHSelevation, YHShighel, ...
                 YHSlowel, YHSazimuth, Aq, num);


    xangle = YHSangle + noiseh(t);      % roll
    yangle = XHSangle + noiseh(t);      % pitch

    xmodel = YHSmodel;
    ymodel = XHSmodel;

    if innovtruemag(t) == 3
       [Br, Btheta, Bphi, Bc, Borbit] = ...
       magfield(SATAltitude(t,:), timemat(t), TrueAnomat(t));
```

```
            erh = [noisemx(t) noisemy(t) noisemz(t)];
            yawangle = yaw_innovation(Borbit', Areal, erh);
        else
            yawangle = yaw_innovation(Borbit', Areal, erh);
        end

        vnor = [1; 1; 1];

        vabs = sqrt(sum(vnor.^2));
        vorb = [vnor(1)/vabs; vnor(2)/vabs; vnor(3)/vabs];

    %   The measurement innovation vector in body axes  %
        hmeas = [xangle; yangle; yawangle];
        Ah = DCMorbo(xangle, yangle, yawangle);

        vmeas = Ah*vorb;
        vest = Aq*vorb;

    %   Transforming the estimated innovation vector in orbital axes %
        vorb = vorb';

    %   The extended Kalman Filter correction loop  %
        [wi, eqq, ndoyest, Pmat, Hmat, Kmat, err, delta] = ...
        EKF_correct(wi, eqq, Pupdate, Rmat, Aq, vmeas, vest, vorb, ...
                                                             ndoyest);


    %   The error between the measured and estimated innovation vector%
        timeh(Hcounter) = timemat(t);
        errhmat(Hcounter,:) = err';
        errmmat(Mcounter,:) = err';
        time(Mcounter) = timemat(t);

        Hcounter = Hcounter + 1;
        Mcounter = Mcounter + 1;
    else
        Pmat = Pupdate;
    end % if innovtruemag(t) == 1


%   Calculate the real and estimated roll, pitch and yaw angles %
    [erollr, epitchr, eyawr] = DCMToEuler(Areal);
    [eroll, epitch, eyaw] = DCMToEuler(Aq);

%   The aerodynamic disturbance torque  %
```

```
    ndoymat(t) = ndoyest;

%   Variable matrixes   %
    %   The real inertial angular rate and quaternion values    %
    wirealmat(t,:) = wireal';
    qrealmat(t,:) = qreal';

    %   The real roll, pitch and yaw angles %
    Eulermatreal(t,:) = [erollr, epitchr, eyawr];

    %   The estimated roll, pitch and yaw angles    %
    Eulermat(t,:) = [eroll, epitch, eyaw];

    %   The estimated angular rate in orbital coordinates   %
    womat(t,:) = wo';

    %   The estimated angular rate in inertial coordinates  %
    wimat(t,:) = wi';

    %   The estimated quaternions   %
    eqqmat(t,:) = eqq';

    %   The Kalman filter gain %
    knum = 1;
    for ry = 1:8
        for col = 1:3
            kstruct(knum).kmat(t) = Kmat(ry, col);
            knum = knum + 1;
        end %for col = 1:3
    end %for ry = 1:8

    %   The perturbation covariance matrix  %
    for ry = 1:8
        for col = 1:8
            if ry == col
                pstruct(ry).pmat(t) = Pmat(ry, col);
            end %if ry == col
        end %for col = 1:8
    end %for ry = 1:8

end % t = 1:length(timemat)

%   Convert angles from radians to degrees  %
wirealmat = wirealmat*3120/pi;
wimat = wimat*3120/pi;
```

```
womat = womat*3120/pi;
errmmat = errmmat*180/pi;
errhmat = errhmat*180/pi;
Eulermat = Eulermat*180/pi;
Eulermatreal = Eulermatreal*180/pi;

%   Figures of different parameters %
kalfigures(timemat, time, timeh, errmmat, errhmat, kstruct, ...
        pstruct, wirealmat, wimat, womat, Eulermatreal, Eulermat, ...
                                            qrealmat, eqqmat, ndoymat)
```

C.2.2 Software Code: PlantmodelPropagation.m

```
% ================================================================== %
% Propagation of the plantmodel                                      %
%     Input: Sampling Time (Ts)                                      %
%          : Inertial referenced angular rate at sampling, k (wi)    %
%          : Quaternion matrix at sampling, k (q)                    %
%          : Aerodynamic disturbance torque (ndoy)                   %
%    Output: Propagated inertially referenced angular rate at        %
%            sampling, k+1 (wi)                                      %
%          : Propagated orbitaly referenced angular rate at          %
%            sampling, k+1 (wo)                                      %
%          : Propagated quaternion matrix at sampling, k+1 (q)       %
%          : DCM matrix from updated quaternions (A)                 %
%          : Propagated Gravity-Gradient Torque at sampling, k+1 (NGG)
% ================================================================== %

function [wi, wo, q, A, NGG] = PlantmodelPropagation(Ts, wi, q, ndoy)

% ------------------------------------------------------------------ %
%       Globals                                                      %
% ------------------------------------------------------------------ %

global ORBITn
global MOI

% ------------------------------------------------------------------ %
%       Propagation                                                  %
% ------------------------------------------------------------------ %

womean = ORBITn;
I1 = MOI(1,1); I2 = MOI(2,2); I3 = MOI(3,3);

A = qua2dcm(q(1), q(2), q(3), q(4));
```

```
wo = wi - A*[0; -womean; 0];

NGG = GravityTorque(womean, A, MOI);
ND = DisturbanceTorque(A, ndoy);

% Calculate f(xn, yn) from won for the quaternion update
dqn = bodyrate2quaternion(wo, q);

% Calculate f(xn, yn)
dwix = 1/I1*(NGG(1) + ND(1) + (I2 - I3)*wi(3)*wi(2));
dwiy = 1/I2*(NGG(2) + ND(2) + (I3 - I1)*wi(3)*wi(1));
dwiz = 1/I3*(I1 - I2)*wi(1)*wi(2);

% Calculate yn+1_star
wi_star = wi + Ts*[dwix; dwiy; dwiz];

% Calculate wo from yn+1
wo = wi_star - A*[0; -womean; 0];

% Calculate yn+1_star for the quaternion update
qn_star = q + Ts * dqn;

% Calculate f(xn+1, yn+1_star) for the quaternion update
dqn_star = bodyrate2quaternion(wo, qn_star);

% Calculate NGG_star
A = qua2dcm(qn_star(1), qn_star(2), qn_star(3), qn_star(4));

NGG = GravityTorque(womean, A, MOI);
ND = DisturbanceTorque(A, ndoy);

% Calculate f(xn+1, yn+1)
dwix1 = 1/I1*(NGG(1) + ND(1) + (I2 - I3)*wi_star(3)*wi_star(2));
dwiy1 = 1/I2*(NGG(2) + ND(2) + (I3 - I1)*wi_star(3)*wi_star(1));
dwiz1 = 1/I3*(I1 - I2)*wi_star(1)*wi_star(2);

% Calculate yn+1
win = wi + Ts*0.5*[(dwix + dwix1); (dwiy + dwiy1); (dwiz + dwiz1)];

wi = win;

qn = q + Ts * 0.5 * (dqn + dqn_star);

q = qn;
```

```
A = qua2dcm(qn(1), qn(2), qn(3), qn(4));

% Calculate the body angular rates from yn+1
wo = wi - A*[0; -womean; 0];
```

C.2.3 Software Code: EKF_popag.m

```
% ================================================================== %
% Kalman Filter loop one:  Propagation                               %
%    Input: Sampling Time (dt)                                       %
%         : Estimated inertially referenced angular rate at          %
%           sampling, k (wi)                                         %
%         : Estimated quaternion matrix at sampling, k (eqq)         %
%         : Aerodynamic disturbance torque (ndoy)                    %
%         : Perturbation covariance matrix at sampling, k (Pmat)     %
%         : System covariance matrix (Qmat)                          %
%   Output: Propagated estimated inertially referenced angular       %
%           rate at sampling, k+1 (wi)                               %
%         : Propagated estimated orbitally referenced angular        %
%           rate at sampling, k+1 (wo)                               %
%         : Propagated estimated quaternion matrix at                %
%           sampling, k+1 (eqq)                                      %
%         : DCM matrix from quaternion matrix at sampling, k+1 (Aq)  %
%         : Propagated Gravity-Gradient torque at sampling, k+1 (NGG)%
%         : Linearised perturbation state matrix (Fmat)              %
%         : Updated perturbation covariance matrix at sampling,      %
%           k+1 (Pupdate)                                            %
% ================================================================== %

function [wi, wo, eqq, Aq, NGG, PHI, Fmat, Pupdate] = ...
                             EKF_propag(dt, wi, eqq, ndoy, Pmat, Qmat)


% ------------------------------------------------------------------ %
%       Globals                                                      %
% ----------------------- ------------------------------------------ %

global EarthEquRadius
global ORBITeccent ORBITn
global TransMOI zaxisMOI MOI GeoGrav

% ------------------------------------------------------------------ %
%   STEP 1: Propagate the full satellite state                       %
%       A = Propagate the aerodynamic disturbance torque                  %
%       B = Propagate the estimated angular rate vector                   %
%       C = Propagate the estimated quaternion                            %
% --------------------------------------------------------------------- %

    [wi, wo, eqq, Aq, NGG] = PlantmodelPropagation(dt, wi, eqq, ndoy);

% --------------------------------------------------------------------- %
%  STEP 2: Compute the linearised perturbation state matrix Fmat        %
% --------------------------------------------------------------------- %
gg = 6*ORBITn^2*(MOI(1,1) - MOI(3,3))/MOI(1,1);

c1 = 1 - MOI(3,3)/MOI(1,1);
c2 = 2*ndoy/MOI(1,1);
c3 = gg*Aq(1,3);
c4 = gg*Aq(2,3);
c5 = gg*Aq(3,3);
Fmat(1,1) = 0;
Fmat(1,2) = wi(3)*c1;
Fmat(1,3) = wi(2)*c1;
Fmat(2,2) = 0;
Fmat(2,1) = -wi(3)*c1;
Fmat(2,3) = -wi(1)*c1;
Fmat(3,1) = 0; Fmat(3,2) = 0; Fmat(3,3) = 0;
Fmat(1,4) = -eqq(4)*c5 + eqq(1)*c4 + eqq(2)*c2;
Fmat(1,5) = -eqq(3)*c5 + eqq(2)*c4 + eqq(1)*c2;
Fmat(1,6) = -eqq(2)*c5 - eqq(3)*c4 + eqq(4)*c2;
Fmat(1,7) = -eqq(1)*c5 - eqq(4)*c4 + eqq(3)*c2;
Fmat(2,4) = eqq(3)*c5 - eqq(1)*c3 - eqq(1)*c2;
Fmat(2,5) = -eqq(4)*c5 - eqq(2)*c3 + eqq(2)*c2;
Fmat(2,6) = eqq(1)*c5 + eqq(3)*c3 - eqq(3)*c2;
Fmat(2,7) = -eqq(2)*c5 + eqq(4)*c3 + eqq(4)*c2;
Fmat(3,4) = 0; Fmat(3,5) = 0; Fmat(3,6) = 0; Fmat(3,7) = 0;
Fmat(1,8) = Aq(1,2); Fmat(2,8) = Aq(2,2); Fmat(3,8) = 0;
Fmat(4,1) = 0.5*eqq(4);     Fmat(4,2) = -0.5*eqq(3);
Fmat(4,3) = 0.5*eqq(2);
Fmat(5,1) = 0.5*eqq(3);     Fmat(5,2) = 0.5*eqq(4);
Fmat(5,3) = -0.5*eqq(1);
Fmat(6,1) = -0.5*eqq(2);    Fmat(6,2) = 0.5*eqq(1);
Fmat(6,3) = 0.5*eqq(4);
Fmat(7,1) = -0.5*eqq(1);    Fmat(7,2) = -0.5*eqq(2);
Fmat(7,3) = -0.5*eqq(3);
Fmat(4,4) = 0; Fmat(4,5) = 0.5*wo(3); Fmat(4,6) = -0.5*wo(2);
Fmat(4,7) = 0.5*wo(1);
Fmat(5,4) = -0.5*wo(3); Fmat(5,5) = 0; Fmat(5,6) = 0.5*wo(1);
Fmat(5,7) = 0.5*wo(2);
Fmat(6,4) = 0.5*wo(2); Fmat(6,5) = -0.5*wo(1); Fmat(6,6) = 0;
Fmat(6,7) = 0.5*wo(3);
Fmat(7,4) = -0.5*wo(1);     Fmat(7,5) = -0.5*wo(2);
Fmat(7,6) = -0.5*wo(3);
Fmat(7,7) = 0;
Fmat(4,8) = 0; Fmat(5,8) = 0; Fmat(6,8) = 0; Fmat(7,8) = 0;
Fmat(8,1) = 0; Fmat(8,2) = 0; Fmat(8,3) = 0; Fmat(8,4) = 0;
Fmat(8,5) = 0;
Fmat(8,6) = 0; Fmat(8,7) = 0; Fmat(8,8) = 0;

%Fmat;

% ------------------------------------------------------------------ %
%  STEP 3: Obtain the discrete system matrix PHI                     %
% ------------------------------------------------------------------ %

PHI = eye(8,8) + Fmat*dt + 0.5*(Fmat*dt)^2;

% ------------------------------------------------------------------ %
%  STEP 4: Propagate the perturbation covariance matrix Pmat         %
% ------------------------------------------------------------------ %

Pupdate = PHI*Pmat*PHI' + Qmat;
```

C.2.4 Software Code: EKF_correct.m

```
% ================================================================== %
%  Kalman Filter loop two: Correction                                %
%       Input: Propagated estimated inertially referenced angular    %
%              rate at sampling, k+1 (wi)                            %
%            : Propagated estimated quaternion matrix at             %
%              sampling, k+1 (eqq)                                   %
%            : Updated perturbation covariance matrix at sampling,   %
%              k+1 (Pupdate)                                         %
%            : Measurement Noise covariance matrix (R)               %
%            : DCM matrix from quaternion matrix at sampling, k+1 (Aq)
%            : Measured innovation vector in body coordinates (vmeas)%
%            : Modelled innovation vector in body coordinates (vest) %
%            : Modelled innovation vector in orbit coordinates (vorb)%
%            : Aerodynamic disturbance torque (ndoy)                 %
%      Output: Corrected estimated inertially referenced angular     %
%              rate at sampling, k+1 (wi)                            %
%            : Corrected estimated quaternion matrix at              %
%              sampling, k+1 (eqq)                                   %
%             : Aerodynamic disturbance torque (ndoy)                 %
%             : Updated perturbation covariance matrix at sampling,   %
%               k+1,k+1 (Pupdate)                                     %
%             : Output measurement matrix from corrected quaternion   %
%               matrix (Hmat)                                         %
%             : Kalman Filter gain (Kmat)                             %
%             : Innovation error vector (err)                         %
%             : Matrix with all estimated state values (delta)        %
% =================================================================== %

function [wi, eqq, ndoy, Pmat, Hmat, Kmat, err, delta] = ...
      EKF_correct(wi, eqq, Pupdate, Rmat, Aq, vmeas, vest, vorb, ndoy)


% ------------------------------------------------------------------- %
%  STEP 5: Compute the discrete output measurement matrix Hmat        %
% ------------------------------------------------------------------- %

h1 = ...
[eqq(1) eqq(2) eqq(3); eqq(2) -eqq(1) eqq(4); eqq(3) -eqq(4) -eqq(1)];
h2 = ...
[-eqq(2) eqq(1) -eqq(4); eqq(1) eqq(2) eqq(3); eqq(4) eqq(3) -eqq(2)];
h3 = ...
[-eqq(3) eqq(4) eqq(1); -eqq(4) -eqq(3) eqq(2); eqq(1) eqq(2) eqq(3)];
h4 = ...
[eqq(4) eqq(3) -eqq(2); -eqq(3) eqq(4) eqq(1); eqq(2) -eqq(1) eqq(4)];

h0 = [0 0 0; 0 0 0; 0 0 0];
h1_1 = 2*h1*vorb';
h2_1 = 2*h2*vorb';
h3_1 = 2*h3*vorb';
h4_1 = 2*h4*vorb';
h5 = [0; 0; 0];

Hmat = [h0 h1_1 h2_1 h3_1 h4_1 h5];

% ------------------------------------------------------------------- %
%  STEP 6: Compute the Kalman filter gain                             %
% ------------------------------------------------------------------- %

Kmat = Pupdate*Hmat'*inv(Hmat*Pupdate*Hmat' + Rmat);

% ------------------------------------------------------------------- %
%  STEP 7: Calculate the innovation error vector ek                   %
% ------------------------------------------------------------------- %
```

```
err = vmeas - vest;

% ----------------------------------------------------------------- %
%  STEP 8: Update the state vector with the innovation              %
% ----------------------------------------------------------------- %

dwix = Kmat(1,:)*err; dwiy = Kmat(2,:)*err; dwiz = Kmat(3,:)*err;
deq1 = Kmat(4,:)*err; deq2 = Kmat(5,:)*err; deq3 = Kmat(6,:)*err;
deq4 = Kmat(7,:)*err; dndoy = Kmat(8,:)*err;

wixn = wi(1) + dwix;
wiyn = wi(2) + dwiy;
wizn = wi(3) + dwiz;
eq1n = eqq(1) + deq1;
eq2n = eqq(2) + deq2;
eq3n = eqq(3) + deq3;
eq4n = eqq(4) + deq4;
ndoyn = ndoy + dndoy;

wi(1) = wixn; wi(2) = wiyn; wi(3) = wizn;
eqq(1) = eq1n; eqq(2) = eq2n; eqq(3) = eq3n; eqq(4) = eq4n;
ndoy = ndoyn;

delta = [dwix dwiy dwiz deq1 deq2 deq3 deq4];

% ----------------------------------------------------------------- %
%  STEP 9: Normalize the quaternion q1 + q2 + q3 + q4 = 1           %
% ----------------------------------------------------------------- %

eqabs = sqrt(eqq(1)^2 + eqq(2)^2 + eqq(3)^2 + eqq(4)^2);
eqq(1) = eqq(1)/eqabs; eqq(2) = eqq(2)/eqabs; eqq(3) = eqq(3)/eqabs;
eqq(4) = eqq(4)/eqabs;

% ----------------------------------------------------------------- %
%  STEP 10: Recompute Hmat for the updated state vector             %
% ----------------------------------------------------------------- %

h1 = ...
[eqq(1) eqq(2) eqq(3); eqq(2) -eqq(1) eqq(4); eqq(3) -eqq(4) -eqq(1)];
h2 = ...
[-eqq(2) eqq(1) -eqq(4); eqq(1) eqq(2) eqq(3); eqq(4) eqq(3) -eqq(2)];
h3 = ...
[-eqq(3) eqq(4) eqq(1); -eqq(4) -eqq(3) eqq(2); eqq(1) eqq(2) eqq(3)];
h4 = ...
```

```
[eqq(4) eqq(3) -eqq(2); -eqq(3) eqq(4) eqq(1); eqq(2) -eqq(1) eqq(4)];

h0 = [0 0 0; 0 0 0; 0 0 0];
h1_1 = 2*h1*vorb';
h2_1 = 2*h2*vorb';
h3_1 = 2*h3*vorb';
h4_1 = 2*h4*vorb';
h5 = [0; 0; 0];

Hmat = [h0 h1_1 h2_1 h3_1 h4_1 h5];

% ------------------------------------------------------------------ %
%  STEP 11: Update the perturbation covariance matrix                %
% ------------------------------------------------------------------ %
cmat = (eye(8,8) - Kmat*Hmat);

Pnew = cmat*Pupdate*cmat' + Kmat*Rmat*Kmat';

Pmat = Pnew;
```