NTNU – Trondheim
Norwegian University of
Science and Technology

Spacecraft Attitude and Orbit Estimation using GPS and Inertial Measurements

Tale Sundlisæter

Master of Science in Engineering Cybernetics
Submission date: July 2012
Supervisor: Thor Inge Fossen, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Engineering Cybernetics

MSC THESIS DESCRIPTION SHEET

Name:	Tale Sundlisæter
Department:	Engineering Cybernetics
Thesis title (Norwegian):	Estimering av attityde og baneparametere for satellitt med bruk av GPS og treghetsmålinger
Thesis title (English):	Spacecraft Attitude and Orbit Estimation using GPS and Inertial Measurements

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

The following items should be considered:

1) Present a mission analysis describing small satellite projects, including NUTS.
2) Investigate GPS attitude estimation, and GPS/INS integration methods. Suggest and implement (Matlab/Simulink) a solution method for the Integer Ambiguity Resolution problem that arises.
3) Implement (Matlab/Simulink) an Extended Kalman Filter for the combination of GPS and IMU measurements for attitude and orbit estimation.
4) Develop a spacecraft simulator that generates all measurements needed for numerical computation of the Extended Kalman Filter.
5) Present your results in a report and include simulation case studies.

Start date:	2012-02-01
Due date:	2012-07-04

Thesis performed at:	Department of Engineering Cybernetics, NTNU
Supervisor:	Professor Thor I. Fossen, Dept. of Eng. Cybernetics, NTNU

Abstract

This report studies the development of a Multiplicative Extended Kalman Filter in Matlab/Simulink, for orbit and attitude estimation for the $10 \times 10 \times 20$ cm CubeSat at the Norwegian University of Science and Technology (NTNU). The filter was developed in a tightly coupled manner with respect to the GPS attitude solution, based on data from differential carrier phase measurements. These measurements are aided by measurements from a three-axis magnetometer, and inertial measurements from a gyroscope.

Four antennas are virtually mounted on the satellite to obtain three baselines of 1 m each. The MEKF is complemented by an integer ambiguity resolution method, which makes sure that the solution for a GPS signal is not accepted until the integrity check value for all baselines is below the acceptance threshold. Until the ambiguities are resolved, the Multiplicative EKF is reliant upon the gyro measurements, and the magnetometer. The filter has been simulated with various attitude maneuvers.

The MEKF performs orbit estimation based on measurements from GPS position, velocity, and timing data, from which it estimates the Keplerian orbital parameters to determine the orbit of the craft. It operates as an ordinary EKF for this purpose.

Simulation shows that the filter is able to determine the attitude and orbit of the spacecraft from the given measurements, and that it is robust to a temporary loss of the GPS measurements. However, the orbit estimator assumes a circular orbit. The quality of orbit estimates are therefore dependent on the eccentricity of the orbit.

Sammendrag

I denne mastergradsavhandlingen er det blitt utviklet et Muliplikativt Extended Kalman Filter i Matlab/Simulink, for estimering av attityde og baneparametere for en dobbel CubeSat, i forbindelse med studentsatellitprosjektet NUTS, ved Norges Teknisk-Naturvitenskapelige Universitet (NTNU). Filteret ble utviklet i tett kobling for å løse for GPS attityde løsningen, som er basert på data fra differensielle fasemålinger fra GPS signal. Disse målingene er støttet av målinger fra et tre-akse magnetometer, samt treghetsmålinger fra et gyroskop.

Fire antenner er virtuelt plassert på satellitten for å oppnå tre baselinjer på èn meter hver. Filteret er komplementert med en metode for å løse tvetydigheten i heltallsoppløsningene. Det garanterer at løsningen for et GPS signal ikke aksepteres før kontrollverdien for heltall for alle baselinjer er under akseptert terskel. Før tvetydigheten er løst, avhenger filteret av målinger fra gyro og magnetometer. Flere forskjellige attitydemanøvrer ble utført.

Det Multiplikative filteret estimerer baneparametere basert på målinger fra GPS posisjon, hastighet, og tid. For dette formålet virker filteret som et ordinært EKF.

Simuleringer viser at filteret er i stand til å bestemme attityden samt banen til satellitten, basert på de gitte målingene. Filteret er også robust overfor midliertidig tap av GPS målinger. Estimering av baneparametere antar imidlertid en sirkulær bane, og kvaliteten på disse estimatene avhenger derfor av banens eksentrisitet.

Preface

This work is the concluding thesis of the Master of Science (Siv.Ing) at the Norwegian University of Science and Technology (NTNU), carried out at the Department of Engineering Cybernetics. I wish to acknowledge Professor Thor Inge Fossen for serving as my supervisor, and Professor Jan Tommy Gravdahl, Professor Odd Gutteberg, and Dr. Ola Erik Fjellstad for their advice and support during the initial stages of this thesis. Appreciation also to Professor Emeritus Gunnar Stette, and Professor Emeritus Börje Forssell for thoughtful conversation, to Roger Birkeland for patiently serving as project manager for the student satellite project, and to the other members of the satellite project group.

Special thanks go to Professor Anders Johnsson for his continuous support, enthusiasm, and thoughtful conversations throughout my education at NTNU. I would also like to give thanks to Dr. Wiley J. Larson at Stevens Institute of Technology, and to his wonderful wife, Carol, for taking the time to share their experienced advice and support during, and after my participation at the International Space University's Space Studies Program (SSP) 2011 in Graz, Austria. Overall, I wish to thank the Space Studies Program class, faculty, staff, and external experts, for fueling my passion. In this context I also wish to express my utmost gratitude to the European Space Agency's scholarship committee, and Kongsberg Seatex AS, for granting the necessary resouces for my SSP participation. I also owe many thanks to the Norwegian Space Centre for their subsequent encouragement and funding of my efforts to present the results at the 62nd International Astronautical Congress in Cape Town, South Africa. It is with inspiration from all this that I have managed to conduct this thesis.

A special word of appeciation goes to my fellow students and friends, with whom I have shared many great moments and cabin trips during my studies at NTNU. I know we will stay in touch.

Lastly, I wish to dedicate my most special thanks to my parents, Marit and Olav, for their life-long support, encouragement, and love.

Trondheim, July 3rd, 2012

Tale Sundlisæter

Contents

List of Abbreviations

ADOP	Attitude Dilution of Precision
AIAA	American Institute of Aeronautics and Astronautics
ANSAT	Norwegian Student Satellite Programme
CoCom	Coordinating Committee for Multilateral Export Controls
UTC	Coordinated Universal Time
DCP	Differential Carrier Phase
ECEF	Earth Center Earth Fixed
ECI	Earth Center Inertial
EKF	Extended Kalman Filter
EQUEST	Extended QUaternaion Estimator
GDOP	Geometric Dilution of Precision
GENSO	Global Educational Network for Satellite Operations
GLONASS	Global'naya Navigatsionnaya Sputnikovaya Sistema
GMT	Greenwich Mean Time
GoSSP	Guidebook on Small Satellite Programs
GPS	Global Positioning System
IAGA	International Association of Geomagnetism and Aeronomy
IEEE	Institute of Electrical and Electronics Engineers
IGRF	International Geomagnetic Reference Field
INS	Inertial Navigation System
IR	Infra Red

IAC	International Astronautical Congress
IMU	Inertial Measurement Unit
ISU	International Space University
ITU	International Telecommunication Union
KF	Kalman Filter
LEO	Low Earth Orbit
LOS	Line Of Sight
MEKF	Multiplikative EKF
MHE	Moving Horizon Estimation
MNOK	Billion Norwegian Crones
MSC	Military Cosmodrome (Plesetsk, Russia)
NAROM	Norwegian Centre for Space-related Education
NASA	National Aeronautics and Space Administration
NCUBE-1	NTNU CubeSat-1
NCUBE-2	NTNU Cubesat-2
NED	North East Down
NTNU	Norwegian University of Science and Technology
NUTS	NTNU Test Satellite
PLATO	PLAnetary Transits and Oscillations of stars
PRN	Pseudo Random Noise
QUEST	QUaternion Estimator
RAAN	Right Ascension of the Ascending Node
SMAD	Space Mission Analysis and Design
SSETI	Space Exploration and Technology Initiative
SV	Space Vehicle
TLE	Two-Line Element
TT&C	Telemetry, Tracking and Command
UNOOSA	United Nations Office for Outer Space Affairs

US	United States
WGS84	World Geodetic System (1984)
XU	X Unit CubeSat

List of Figures

List of Tables

Part I

Preliminaries

Chapter 1

Introduction

1.1 Motivation

1.1.1 Attitude Estimation

Attitude and attitude motion of a spacecraft describes the orientation and rotational motion about its center of mass, and the computation of this orientation relative to either an inertial reference or some object of interest is referred to as attitude estimation. This process is necessary in order to provide a reference for the attitude control system, enabling it to perform command engineering functions. This could, for example, be directional pointing of antennas, solar panels, and nadir control for Earth observing payloads.

A common technique is to use body measurements to estimate the vehicle's orientation, and then compare these measurements with known reference observations. An example of such measurement is the angle between the craft and the Sun, measured with a Sun sensor, or line-of-sight measurements of the position to the stars, provided by a star tracker. These measurements must have a corresponding set of reference vectors in the reference frame, such as a known Sun reference model, and maps of known observed stars. However, we must keep in mind that the size and weight requirements of the satellite impose strict limitations on the choice of sensors. A star tracker is a large sensor to the extent that it is not feasible to implement in double CubeSat design such as the Norwegian University of Science and Technology Test Satellite (NUTS). The sensors of choice must be light weight, consume little power, and be of small size.

Another issue which must be taken into consideration in estimation techniques is the fact that vector measurements will in practice be contaminated with noise. Measurement noise in particular contributes significantly factor at high frequencies, and it is therefore essential to subsequently filter the noise by combining the measurements with models.

In essense, the key issues in attitude estimation avoiding singularities, and providing satisfactory bias estimation.

1.1.2 Orbit Estimation

Knowledge of the spacecraft's orbit is in general required for space missions to perform attitude determination and control functions. Orbit and attitude determination are interdependent parameters, and for a craft in a low altitude Earth orbit both the atmospheric density and the magnetic field strength are determined based on the position of the spacecraft in the orbit. The atmospheric density and the magnetic field strength will, in turn, affect the attitude. Orbit data can be supplied by a process external to the attitude system, such as radar, telemetry, optics, or the Global Positioning System (GPS) [36]. Deciding upon which type of data to use for orbit estimation depends on the selected orbit, as well as the accuracy requirements, and the weight restrictions on the spacecraft.

United States Strategic Command (USSTRATCOM)

Two-line elements (TLEs) containing the orbital parameters can be obtained from the United States Strategic Command (USSTRATCOM), which tracks and maintains a catalog of TLEs of all Earth-orbiting spacecrafts, rocket bodies, and debris. Orbit parameters are estimated by USSTRATCOM from the moment the spacecraft is detected by space surveillance radars. However, it usually takes several days of tracking for USSTRATCOM to have enough radar data to create accurate orbital parameters of the spacecraft. These data are most critically needed at the early stages when they are the least accurate, such that data published by USSTRATCOM should not be relied upon in this early stage. However, they could become useful for on-ground post analysis of spacecraft data.

1.2 Mission Analysis

There are many decisions and trade-offs to consider throughout a small satellite mission, both at a program, project, and system level. This mission analysis is mainly based on the results from the team project Small Satellites for Capacity Building in Space Technology Development which was conducted during the 2011 Space Studies Program at the Inernational Space University, 11. July to 9. September hosted by the Graz University of Technology. Supported by the United Nations Office for Outer Space Affairs (UNOOSA) through the Basic Space Technology Initiative, it is implemented through the United Nations Program on Space Applicatoins. The resulting Guidebook on Small Satellite Programs (GoSSP) was followed by a paper which condensed the results and was presented at the International Astronautical Congress in Cape Town, 2011 [9].

A small satellite design is, compared to a large-scale design, different in the sense that they are built around tighter constraints in mass, volume, and budget. This limits the possibility of adding complex spacecraft systems to the design. It lower the costs, increase the risk and reliability tolerance, and in addition a simpler system often allows for shorter development time. These characteristics make small spacecraft projects attractive as an initial stage for capacity building within educational, governmental, military, or other organizations. Successful space programs can contribute to increased security, increased prestige, knowledge, enhanced cabability, and can also assist in social development.

Regulation

Key regulators of small satellite programs are in particular the International Telecommunication Union (ITU), space agencies, national administrations and national import/export control organizations. The applicable legal framework includes the five United Nation treaties and principles, as well as international regulations and national laws. In particular, states are liable based on their fault for any damage caused to the surface of the Earth, aircraft in flight, or another object in outer space.

NTNU Test Satellite Project Background

The NTNU Test Satellite project is a part of the Norwegian Student Satellite Program (ANSAT), the objective of which is to increase the overall societal interest and awareness of technological studies, and to provide a platform for cooperation with the industry and the educational institutions involved. Importantly, the project provides hands-on experience in that it allows students to gain experience from a real space project. At NTNU, the ANSAT program as a whole includes the Norwegian University of Science and Technology, the University of Oslo and Narvik University College. The project at NTNU is funded through the Norwegian Centre for Space-related Education (NAROM) and the Norwegian Space Centre with a cost-specific budget of about 2.5 MNOK, not counting the workforce.

This test satellite project is the third one of its kind at NTNU, for which a double CubeSat design with the dimensions $10 \times 10 \times 20$ cm was chosen. Small satellite programs are well suited for technology demonstration, and the NTNU Test Satellite platform is designed as a double CubeSat with this in mind. The project eventually aims to build a platform on which new concepts can be tested at a reasonable cost. Starting up in January 2010 and with a total time span of 4 years, the planned launch date for the satellite is late 2014 or early 2015. Technically, the main goal for the project is to establish communication with the satellite from the ground station in Trondheim, a goal which both predecessors NCUBE-1 and NCUBE-2, each dimensioned $10 \times 10 \times 10$, failed to achieve. NCUBE-1 was launched from Plesetsk MSC on 26th July 2006 but

was destroyed due to a failure in the carrier rocket which caused the rocket to explode. NCUBE-2 was successfully launched earlier, on 27th October 2005, but confirmation signals from the satellite were never received at ground stations at Svalbard or in Narvik, and the satellite therefore failed to operate. The NCUBE-2 is listed as SSETI-Express space debris with Norad-number 28897 [28].

The planned payload for the test satellite includes an S-band radio and a small infrared camera meant for scientific detection of airglow (also referred to as night glow), a weak emission of light by the planetary atmosphere of the Earth. The images from the IR camera will be transmitted via an S-band radio.

Challenges for Small Satellite Projects

As with most small satellites, the NTNU Test Satellite will be launched as a secondary payload. This imposes further limitations on the mission design and planning, as well as on post launch operations due to the associated uncertainty and modest operations budget. For example, ne of the requirements for optimal telemetry, tracking, and communications (TT&C) is enlistment of a sufficient number of ground antennas, data providers and external tracking resources to provide globally distributed redundant tracking coverage and augmented TT&C access. This requirement is clearly not met by the NUTS mission, which has so far been granted resources to operate a single ground station located in Trondheim. This currently allows for approximately 10 minutes of downlink time each time the satellite passes within broadband sight of the ground station.

A particularly critical stage of the mission will arise immediately after satellite deployment, as orbit and attitude information will have most likely been estimated pre-launch. It might be possible to obtain state vectors provided by the launcher, but it is not yet known if this information will be available to the mission control. The availability of state vectors, trajectories and uncertainty metrics varies widely depending on the piggyback launch provider. However, it can be expected that these data will be limited to pre-launch positional -and uncertainty estimates of marginal accuracy, as it will most likely be an expendable booster launch, as opposed to on-orbit launch. It is of utmost importance that initial acquisition and commanding must function at a desired rate at this critical stage in order to avoid the risk of degradation of onboard batteries, which could lead to mission failure if initial signal acquisition is not accomplished within the life time of the battery. Another issue is the small bandwidth, which is a direct result of the small sized transmitter onboard the satellite combined with a relatively moderately sized receiving antenna on ground.

Some strategies can be implemented to minimize mission risk and optimize operational capabilities of the small satellite, such as preflight planning and astrodynamics analyses, incorporation of automatic search and tracking capabilities,

Figure 1.1: Debris in polar orbit around Earth. Graphics from Universe Today (2008)

assembly or access to a globally distributed and geographically diverse set of tracking stations using amateur networks, for example GENSO, to obtain health data, external and internal organizational coordination, data management, and modelling and propagation of satellite position uncertainty [11]. These issues and strategies must be handled and managed at a project and program level, keeping an eye on each subsystem.

Documented by Sundlisæter (2008) in an interview with the previous project manager for a planned student satellite at NTNU, Lars Løge, it was revealed that one of the greatest challenges regarding student satellite projects has proved to be lack of continuity, as each student only works on the project for half a year to a year. A countermeasure to this problem was applied to the project in spring 2012 by the current project manager, Roger Birkeland, successfully recruiting students at an earlier level of their education.

Space Debris and Reentry Hazards

Taking into consideration the end of life of the NUTS mission, we recognize that a dead satellite is a typical item referred to as space debris. Other typical items are rocket stages, small flecks of paint and other nonfunctioning human-made items. This issue is a growing concern for operators of space systems. Figure 1.1 illustrates debris in polar orbit around Earth.

Due to the very high velocities in orbit, even microscopic space debris can cause great damage upon impact. Being small of size, the NTNU Test Satellite will be of reduced threat to other spacecraft in terms of probability of impact, but as such small satellites are of relatively low cost, more of them are being launched into space. This causes the population of orbiting objects to increase, additionaly a single launch vehicle can carry and deploy several satellites, increasing the probablility of collision in that launch-specific region of space. The U.S. Air Force maintains a catalog known as the Resident Space Object catalog, in which all objects larger than 10 centimeters are tracked and the inherent hazard they pose to other objects is computed. However, most of this debris is composed of fragments from more dramatic incidents like the explosion of a Pegasus stage in 1996 and the Chinese antisatellite test that destroyed the Feng Yung weather satellite in 2007. The NTNU Test Satellite mission does not have any space debris avoidance maneuvers for this purpose and in terms of risk and budget the double CubeSat imposes more a threat as space debris as opposed to being threatened by it. However, the probability of either is very small.

Disposal

Having the orbital altitude of the spacecraft ensure that the hazard posed to other space objects ends due to natural decay within a 25 year span is an accepted disposal technique for space hardware. The small spacecraft mission budget has no room for incorporating an end-of-life disposal action plan, and is therefore dependent upon this disposal technique. The double CubeSat will decay in less than 25 years at a 95% atmosphere, given the deployment altitude is 700 km or below [11], imposing this altitude as a strict altitude boundary. The NUTS project is therefore currently looking for piggyback rides with a deployment altitude of 600 km to meet this requirement.

Documentation

In order to capture lessons learned and informing all team members of a project's current status, documentation is imperative. The student project at NTNU conveniently documents all work through project or master thesis reports.

1.3 Previous Work on Orbit and Attitude Estimation

Attitude estimation is performed in several fields of science, such as in medical science and navigation systems. At NTNU, Svartveit (2003) performed estimation of attitude for the 1U CubeSat NCUBE mission. Using a discrete Kalman Filter (KF) based on measurements from magnetometer and sun sensor (solar panels used as crude sun sensor). This work indicated that the sun sensor measurements were inaccurate due to the Earth's albedo effect. Svartveit also

designed an orbit propagator system based on knowledge of the Keplerian orbital elements at a single point in time with the same orbit assumpsions as the ongoing satellite mission at NTNU. The enhanced version of the orbit propagator by Svartveit also takes secular perturbations into account.

Ose (2004) took the next step basead on Svartveit (2003) and implemented an Extended Kalman Filter (EKF) in Matlab, and a Kalman filter on a microcontroller (due to the complexity of the EKF).

Rohde (2007) then described the implementation process of an EKF on a microcontroller, but did not manage to finish implementation and testing due to time constraints.

Lindegård Jenssen and Huseby Yabar (2011), considering the previous challenges and with inspiration from an extended Kalman filter originally developed by Sabatini (2006) for use in biomedical engineering, made a new approach to the attitude estimation problem. The extended Kalman Filter developed by Sabatini is based on accelerometer, gyroscope, and magnetometer, and Jenssen and Yabar derived their EKF for the test satellite from the biomedical concept. They further developed an extended QUEST (EQUEST) based on work by Psiaki and Markley, focusing on the integration of nonvectorized gyrsocope measurements.

1.4 Contribution and Thesis Organization

The purpose of this thesis is to develop a GPS/IMU (Inertial Measurement Unit) integrated navigation system to be used on the NTNU Test Satellite to provide estimates of the satellite's attitude, position and velocity. Part I presented a mission analysis of the NTNU Test Satellite mission and small satellite missions in general to give an overview of the current project, and to put it into perspective on a program level.

Part II contains information to remind the reader of essential theory, of which Chapter 3 contains essential Kinematics theory and Chapter 4 gives a description of the motion of the spacecraft in low Earth orbit (LEO) modeled as a rigid body. It also contains linearization of the model. Chapter 5 informs the reader about the sensors and the International Geomagnetic Reference Model used in the design. Chapter 6 explains GPS and inertial navigation system integration, followed by an explaination of GPS attitude determination in Chapter 7. This Chapter deals with attitude determination theory and solution method, including an explanation and solution of the integer ambiguity resolution problem that arises. This problem is solved usin the fast integer ambiguity resolution using integer searches, proposed by Lightsey, Crassidis, and Markley (1999).

The main contribution of this thesis is the design and simulation of the attitude estimation system that applies the Multiplicative Extended Kalman Filter (MEKF) to produce the heuristic approximation of the optimal estimate of the spacecraft attitude, given the past measurements and the available dynamic model of the spacecraft. The same filter is used to estimate the orbit parameters based on GPS measurements, for which purpose the MEKF functions as an ordinary extended Kalman filter. The multiplicative extended Kalman filter design is described in Chapter 8.

In Chapter 9 the Simulator design is described, followed by a presentation of the results in Part III.

Closing remarks with conclusions including discussions and recommendations for further work is given in Part IV.

Chapter 2

Nomenclature

General

c : Speed of light.

$\mathbf{I}_{m\times n} \in \mathbb{R}^{m\times n}$: The m-by-n identity matrix.

$\mathbf{0}_{m\times n} \in \mathbb{R}^{m\times n}$: The m-by-n zero matrix.

$\mathbf{S}(\cdot) \in \mathbf{SS}(3) \subset \mathbb{R}^{3\times 3}$: A skew-symmetric matrix representing the cross-product in $\mathbb{R}^3$. $\mathbf{S}(\mathbf{x})\mathbf{y} = \mathbf{x} \times \mathbf{y}$ if $\mathbf{x}$, $\mathbf{y} \in \mathbb{R}^3$

t: The time.

t_0: The initial time.

Rigid Body Dynamics

$\mathbf{R} \in SO(3) \subset \mathbb{R}^{3\times 3}$: Rotation matrix representing the orientation of the b-frame relative to the o-frame. If $\mathbf{r}$ is a vector in physical $\mathbb{R}^3$, then $\mathbf{r}^o = \mathbf{R}\mathbf{r}^b$.

$O(3) \subset \mathbb{R}^{3\times 3}$: All orthogonal matrices of order 3.

$\mathbf{r} \in \mathbb{R}^3$: The position of the spacecraft with respect to the center of the Earth.

$\mathbf{v} \in \mathbb{R}^3$: The velocity vector of the spacecraft.

$i = \begin{bmatrix} x^i & y^i & z^i \end{bmatrix}^T \in \mathbb{R}^3$: The celestial coordinates of the Earth Center Inertial (ECI) frame. $e = \begin{bmatrix} x^e & y^e & z^e \end{bmatrix}^T \in \mathbb{R}^3$: The coordinates in the Earth Center Earth Fixed frame.

$o = \begin{bmatrix} x^o & y^o & z^o \end{bmatrix}^T \in \mathbb{R}^3$: Orbit frame coordinates. $x^o = y^o \times z^o$.

$b = \begin{bmatrix} x^b & y^b & z^b \end{bmatrix}^T \in \mathbb{R}^3$: Body frame coordinates. Coincides with $o = \begin{bmatrix} x^o & y^o & z^o \end{bmatrix}^T \in \mathbb{R}^3$ when the attitude of the spacecraft is 0^o in roll, pitch, and yaw.

$\mathbf{r}^o = [\ x_b^o \quad y_b^o \quad z_b^o\ ]^T \in \mathbb{R}^3$: The position of the origin of the b-frame, described in the o-frame. $\mathbf{x}_o^i$ = Unit vector parallel with the x- axis of the orbit frame, decomposed in the ECI frame.

$\mathbf{q} = [\ \eta \quad \boldsymbol{\epsilon}^T\ ]^T \in \mathbb{R}^4$: Quaternion describing the orientation of the b-frame relative to the o-frame.

$\Phi = [\ \phi \quad \theta \quad \psi\ ]^R \in \mathbb{R}^3$:Vector of roll, pitch, and yaw angles.

$v_{b/o}^b$: The velocity of the b-frame relative to the o-frame, given in the b-frame.

$\omega_{b/o}^b$: The angular velocity of the b-frame relative to the o-frame, given in the b-frame.

$v = [\ v^{oT} \quad \omega^{bT}\ ]^T$:Generalized velocity of the body frame.

τ^b: Sum of all torques acting on the rigid body, given in the b-frame.

$\mathbf{T}(\mathbf{q}) \in \mathbb{R}^{4\times3}$: Transfomation matrix transforming $\mathbf{q}$ and $\boldsymbol{\omega}$to $\dot{\mathbf{q}}$.

$\boldsymbol{\Omega}(\boldsymbol{\omega}) \in \mathbb{R}^{4\times4}$: Transformation matrix transforming $\mathbf{q}$ and $\boldsymbol{\omega}$to $\dot{\mathbf{q}}$.

The Environment

G: Universal empirical physical constant involved in the calculation(s) of gravitational force between two bodies.

$M > 0 \in \mathbb{R}$: Mass of the Earth.

$\mu_g = GM$: Gravitational constant.

a_{igrf} : Mean radius of the Earth.

r_{igrf} : Distance from the center of the Earth.

ϕ_{igrf} : Longitude east of Greenwich.

θ_{igrf} : Colatitude (90^o-latitude).

$g > 0 \in \mathbb{R}$: The acceleration of gravity.

$\omega_E > 0 \in \mathbb{R}$: The angular rate of the Earth.

$\frac{1}{f} > 0 \in \mathbb{R}$: Reciprocal flatteing of the Earth.

$P_n^m cos(\theta)$: The Schmidt normalized associated Legendre functions of degree n and order m.

g_n^m, h_n^m: Gauss coefficients associated with the Schmidt normalized Legendre functions of degree n and order m.

γ: Vernal equinox. The point where the Sun passes on its way from south to north. Used as origin of the celestial Eearth Center Inertial (ECI) coordinate system.

Keplerian Orbital Parameters

$a > 0 \in \mathbb{R}$: Semi-major axis.

$e \in [0, 1 >$: Eccentricity of a celestial body.

$M \in [\ 0,\quad 2\pi\ ]$: Mean anomaly.

$\nu \in [\ 0,\quad 2\pi\ ]$: True anomaly

$i \in [\ 0,\quad \pi\ ]$: Inclination, the angle between the orbital plane and the equatorial plane.

$\Omega \in [\ 0,\quad 2\pi\ ]$: Right Ascension of the Ascending Node

$\omega \in [\ 0,\quad 2\pi\ ]$: Argument of Perigee

The Spacecraft

$m > 0 \in \mathbb{R}$:Mass of the spacecraft.

$I_{sat} = I_{sat}^T > 0 \in \mathbb{R}^{3\times 3}$:Spacecraft's moment of inertia in the body frame.

$r_g^b = [\ x_g^b \quad y_g^b \quad z_g^b\ ]^T$: Position of the spacecraft's center of gravity, given in the b-frame.

$a > 0 \in \mathbb{R}$: Angular velocity of the orbit frame.

Sensors and Measurements

$\boldsymbol{\omega}_{imu} \in \mathbb{R}^3$: The angular velocity measured by the IMU.

$\mathbf{b}_g \in \mathbb{R}^3$: The gyro bias.

$\mathbf{m}_{imu} \in \mathbb{R}^3$: The magnetometer measurement.

$\mathbf{b}_m \in \mathbb{R}^3$: Local magnetic disturbance.

$\boldsymbol{\alpha} \in \mathbb{R}^6$: Gyro misalignment angles.

$\boldsymbol{\kappa} \in \mathbb{R}^3$: Gyro scale factor error.

r_{ij} : Pseudorange to the satellite i at antenna j.

N_{ij} : Ambiguity number of whole periods from satellite i to antenna j.

$\triangle\partial = \partial_i^r - \partial_j^s$: ∂_i^r is the clock error in the receiver and ∂_j^s is the clock error in the GPS SV.

e_{ij} : Errors on the signal.

φ_{ij} : Measured phase of signal i at antenna j.

Part II

Modeling

Chapter 3

Kinematics

Introduction

The motion of a rigid spacecraft in orbit is specified by its position, velocity, attitude, and attitude motion. The position and the velocity are the subject of what is typically called celestial mechanics, orbit determination, or space navigation, describing the translational motion of the center of mass of the spacecraft [35]. To describe attitude, two coordinate systems are usually defined: One is defined on the reference frame and one is defined on the vehicle body.

3.1 Reference Frames

The coordinate systems represented here have orthogonal unit vectors that follow the right hand rule. This thesis uses the IGRF model of the magnetic field together with line of sight (LOS) vectors between GPS Space Vehicles (GPS SVs) and the satellite to calculate the needed reference vectors. This attitude determination solution is augmented by gyroscope measurements, also referred to as inertial guidance. Given known reference vectors, an attitude sensor mounted on the spacecraft body can measure the orientation of that vector (or some function of the given vector) in the body reference frame of the craft.

Information on the satellite orbit is acquired by the GPS system which provides the necessary navigation data to the system in order to estimate orbit and attitude data.

Earth Center Inertial (ECI)

The Earth Center Inertial (ECI) reference frame is a system of celestial coordinates $i = \left[x^i,\, y^i\, z^i\right]^T$ and is usually applied for orbital analysis of an Earth orbiting satellite, for astronomy purposes, and for inertial motion analysis. In these cases, the ECI frame is considered a sufficiently inertial and non-accelerating

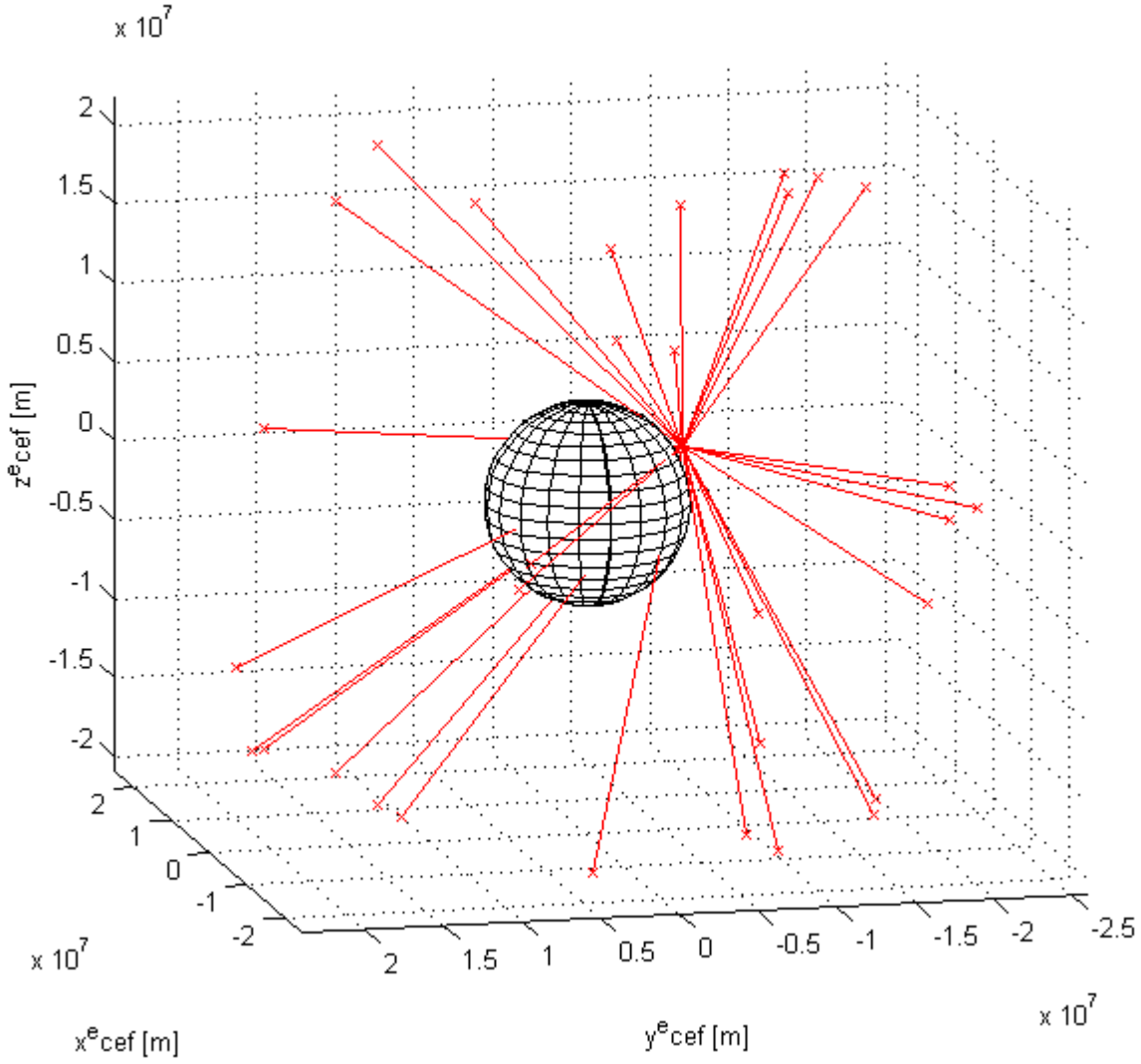

Figure 3.1: The current YUMA almanac from The U.S. Coast Guard Navigation Center of Excellence includes 31 GPS SVs. They are here used to simulate GPS line of sight vectors seen from the test satellite in a 600 km polar low Earth orbit. This simulation shows that the altitude is not sufficient to be in field of view of the complete GPS constellation. However, it does significantly enhance the visibility compared to the visibility for terrestrial applications.

reference frame in the sense that its acceleration due to the Earth orbiting the Sun can be disregarded. The ECI x-axis is defined to point towards the point where the sun crosses the Earth's equatorial plane when it goes from south to north, also known as the vernal equinoz, γ. The z-axis is aligned with the rotation axis of the Earth, pointing towards the celestial North Pole. The y-axis completes the right hand cartesian coordinate frame.

Earth Center Earth Fixed (ECEF)

The ECEF frame, $e = \begin{bmatrix} x^e & y^e & z^e \end{bmatrix}^T$ has its origin located in the center of the Earth. Its x -and y axes rotates with the Earth relative to the ECI frame with rate $\omega_E = 7.2921 * 10^{-5} [rad/s]$, about the z -axis. The x -axis is defined as pointing towards the intersection between the Greenwich meridian and the Equator $(0^o longitude, 0^o latitude)$, and the y -axis completes the right hand orthogonal system.

Orbit Frame

The origin of the orbit frame is located in the center of mass of the satellite, and is denoted o. The spacecraft's center of mass, $r_g^b = \begin{bmatrix} x_g^b & y_g^b & z_g^b \end{bmatrix}^T$ is located in the origin of the body frame (see below), i.e. $r_g^b = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$. The orbit frame rotates with an angular velocity ω_o relative to the ECI frame, and it follows that for a circular orbit ω_o will be constant. The orbit's vector space is spanned by the unit axis vectors x^o, y^o and z^o, with the z^o- axis defined to always point in *nadir* direction (towards the Earth's centroid) and the y^o - axis in the direction of the negative orbit normal. The x^o-axis, $x^o = y^o \times z^o$, will coincide along the linear velocity vector (i.e. direction of motion) if the satellite if the orbit is perfectly circular. The orbit frame will rotate once and have two flips for each orbit, completing a 360^o turn.

Body Frame

The body coordinate system is fixed to the spacecraft and is the system in which the attitude measurements are made. The origin of the body frame is defined to coincide with the origin of the orbit frame, namely in the center of mass of the satellite. The attitude of the satellite will be described relative to the orbit frame, and the body frame $x^b = \begin{bmatrix} x^b & y^b & z^b \end{bmatrix}^T$ axes coincide with the orbit frame x^o, y^o and z^o axes, respectively, when the attitude of the satellite equals 0^o in roll, pitch, and yaw. Note that this frame defines the orientation of the attitude determination and control hardware, which should not be confused with the location of the attitude sensing hardware within the spacecraft (it is the orientation of the field of view of the hardware in the spacecraft body coordinate system which is of importance).

North East Down (NED)

The NED coordinate system is a system which is usually defined in the tangent plane on the surface of the Earth, moving with the craft. It is defined relative to the World Geodetic System, 1984 (WGS84), a mathematically defined surface approximated to fit the shape of the Earth. The x-axis of the NED coordinate system points towards north, the z-axis points downwards, perpendicular to the WGS84 reference ellipsoid, while the y-axis completes the right handed orthogonal coordinate system, pointing towards East.

3.2 Rotation Matrices

A rotation matrix, also referred to as an attitude matrix or a direction cosine matrix, serves the purpose of describing the orientation of a frame b with respect to another frame. Applying a rotation matrix to transform a vector from one reference frame to another can also be interpreted as rotating the frame itself whilst keeping the vector still. The rotation matrix belongs to $SO(3)$, $\mathbf{R} \in SO(3)$, the special orthogonal group of order 3 for which:

$$SO(3) = \{\mathbf{R} | \mathbf{R} \in \mathbb{R}^{3\times 3}, \quad \mathbf{R} \text{ is orthogonal and det } \mathbf{R}=1\} \tag{3.1}$$

for which the group $SO(3)$ is a subset of $O(3)$ (all orthogonal matrices of order 3):

$$O(3) = \{\mathbf{R} | \mathbf{R} \in \mathbb{R}^{3\times 3}, \qquad \mathbf{R}\mathbf{R}^{\mathbf{T}} = \mathbf{R}^T\mathbf{R} = \mathbf{I}\} \tag{3.2}$$

where $\mathbf{R}$ satisfies:

$$\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I} \qquad \det \mathbf{R} = 1 \tag{3.3}$$

and the inverse rotation matrix is consequently given by:

$$\mathbf{R}^{-1} = \mathbf{R}^{\mathbf{T}} \tag{3.4}$$

Furthermore, a matrix $\mathbf{R}$ is a rotation matrix if and only if it is an element of the set $SO(3)$ **[6].**

3.2.1 Dynamics of the Rotation Matrix

Let us define skew-symmetry of a matrix, a property which is defined for a matrix that belongs to the set of skew-symmetric matrices of order n. The off-diagonal elements of $\mathbf{S}$ satisfy $s_{ij} = -s_{ji}$ for $i \neq j$ while the diagonal elements are zero:

$$\mathbf{S} = \mathbf{S}^T \tag{3.5}$$

By differentiating the equation [33]:

$$\mathbf{R}(\mathbf{R})^T = \mathbf{I} \tag{3.6}$$

with respect to time, the differential equation for the rotation matrix can be found. The result:

$$\dot{\mathbf{R}}(\mathbf{R})^T + \mathbf{R}(\dot{\mathbf{R}})^T = \mathbf{0}, \tag{3.7}$$

which can be written

$$\mathbf{S} + \mathbf{S}^T = \mathbf{0} \tag{3.8}$$

The property in Equation (3.8) reflects that:

$$\mathbf{S} = \dot{\mathbf{R}}(\mathbf{R})^T \tag{3.9}$$

is a skew-symmetric matrix. Moreover, for a skew-symmetric matrix exist a vector:

$$\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T, \tag{3.10}$$

such that:

$$\mathbf{S}(\omega) = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ \omega_2 & \omega_1 & 0 \end{bmatrix} \in \mathbb{SS}(3)$$

Postmultiplication of (3.9) with **R**gives the following differential equation of the rotation matrix:

$$\dot{\mathbf{R}} = \mathbf{S}(\omega)\mathbf{R}, \tag{3.11}$$

and using the similarity transform [33]:

$$\mathbf{RS}(\omega)\mathbf{S} = \mathbf{SR}(\omega) = \mathbf{S}(\omega) \tag{3.12}$$

Equation (3.11) can also be written:

$$\dot{\mathbf{R}} = -\mathbf{RS}(\omega) \tag{3.13}$$

3.2.2 Rotation Matrix Parameterization

Euler Angles

The set of parameters most commonly used to describe the motion of a rigid, freely moving body are the composite rotations using Euler angles, which are physically intuitive compared to, for example, quaternions. Using Euler angles, the rotation matrix may be described as a multiple of the three principal rotation

(one-axis rotation) matrices, rotating an angle ψ about the z-axis (yaw), θ about the y-axis (pitch), and φ about the x-axis (roll), respectively [8]:

$$\mathbf{R}_{\mathbf{x},\varphi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\varphi & -s\varphi \\ 0 & s\varphi & c\varphi \end{bmatrix} \tag{3.14}$$

$$\mathbf{R}_{\mathbf{y},\vartheta} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \tag{3.15}$$

$$\mathbf{R}_{\mathbf{z},\psi} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.16}$$

Note that there are many possible combinations of these rotations, and that the sequence of axial rotations is not arbitrary. Moreover, the rotation matrix R, in this case

$$\mathbf{R}_{\mathbf{x},\varphi}\mathbf{R}_{\mathbf{y},\vartheta}\mathbf{R}_{\mathbf{z},\psi} = \begin{bmatrix} c\theta\mathbf{c}\psi & s\theta s\varphi c\psi\text{-}c\varphi s\psi & s\theta c\varphi c\psi + s\varphi s\psi \\ c\theta s\psi & s\theta s\varphi s\psi + c\varphi c\psi & s\theta c\varphi s\psi - s\varphi c\psi \\ -s\theta & c\theta s\varphi & c\theta c\varphi \end{bmatrix} \tag{3.17}$$

Due to its orthogonality, there are six constraints imposed on the elements of this matrix and only three independent parameters are required to describe it. Furthermore, it is evident from 3.17 that the rotation matrix becomes singular for $\theta = \pm 90^o$, referred to as the *gimbal lock*. The singularity can not be avoided if the Euler angles are used in the mathematical description of attitude.

Angle Axis Representation

An alternative tool to the Euler angle representation is the angle axis representation, a useful representation for developing equations of motion and kinematic models for attitude estimation and control systems. Remembering the property of skew symmetry, we can now can describe the rotation matrix corresponding to a single rotation β about the single axis λ:

$$\mathbf{R}_{\lambda,\beta} = \mathbf{I}_{3\times 3} + sin(\beta)\mathbf{S}(\boldsymbol{\lambda}) + (1 - cos(\beta))\mathbf{S}^2(\boldsymbol{\lambda}), \tag{3.18}$$

Recalling that the attitude mapping is applied from orbit to body, we get:

$$\mathbf{v}^b_{b/o} = \mathbf{R}^b_o \mathbf{v}^o_{b/o}, \quad \mathbf{R}^b_o := \mathbf{R}_{\lambda,\beta} \tag{3.19}$$

where $\mathbf{v}^o_{b/o}$ is a vector fixed in the reference frame and $\mathbf{v}^b_{b/o}$ is a vector fixed in the body-frame. The superscript denotes in which frame the coordinates are given, and the subscript denotes which frame is relative to the other.

Unit Quaternions

As previously mentioned, a minimal three-parameter set with mutually independendent orientation parameters such as the rotation matrix is incapable of representing an arbitrary orientation in space. Such minimal parameterizations are therefore often avoided in practice. The attitude will therefore be represented by the four-dimensional singularity-free quaternion vector to avoid the representation singularity, as quaternion theory has many analysis techniques and tools which we can apply upon treating the Euler parameters as a unit quaternion vector. For a locally non-singular representation of SO(3), the four-component quaternion has the lowest dimensionality possible [18]. We have seen that every rotation can be described by a unit-length rotation axis $\boldsymbol{\lambda} \in \mathbb{R}^3$and a rotation angle β. The corresponding quaternion $\mathbf{q}$ is defined as a complex number with one real part η and three imaginary parts given by the vector ϵ [8]:

$$\mathbf{q} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} \tag{3.20}$$

where

$$\eta := \cos\left(\frac{\beta}{2}\right) \in \mathbb{R} \tag{3.21}$$

and

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 & \epsilon_2 & \epsilon_3 \end{bmatrix}^T = \boldsymbol{\lambda} \sin\left(\frac{\beta}{2}\right) \in \mathbb{R}^3 \tag{3.22}$$

$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 \end{bmatrix}^T$ is the unit vector satisfying:

$$\boldsymbol{\lambda} = \pm \frac{\boldsymbol{\epsilon}}{\sqrt{\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}}}; \sqrt{\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}} \neq 0 \tag{3.23}$$

and the unit quaternion is a quaternion of unit length, satisfying:

$$\mathbf{q}^T \mathbf{q} = \boldsymbol{\eta}^2 + \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = 1 \tag{3.24}$$

From this we can see that if η and ϵ are the Euler parameters, $\mathbf{q}$ is the unit quaternion corresponding to the rotation matrix $\mathbf{R}_{\eta,\epsilon}$, and the unit quaternion which corresponds to the identity matrix $\mathbf{R}_{1,0} = \mathbf{I}$ is the identity quaternion $\mathbf{q}_I$ defined by:

$$\mathbf{q_I} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.25}$$

The quaternion with the norm of one, $|q| = 1$, is a unit quaternion. Note that for a quaternion based Kalman filter design, the unit norm constraint which the unit quaternion must obey causes a singular covariance matrix, P. This problem is handled in Chapter 8 using the Multiplicative Extended Kalman Filter design.

Quaternion Differential Equation

Given a body-fixed angular velocity vector $\omega^b_{b/o} = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T$, the differential equation for the quaternion can be expressed:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\eta} \\ \dot{\boldsymbol{\epsilon}} \end{bmatrix} = \frac{1}{2}\mathbf{T}(\mathbf{q})\boldsymbol{\omega}^b_{b/o} \tag{3.26}$$

$$\mathbf{T}(\mathbf{q}) = \begin{bmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \eta & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \eta & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \eta \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\epsilon}^T \\ \eta\mathbf{I}_{3\times 3} + \mathbf{S}(\boldsymbol{\epsilon}) \end{bmatrix}, \tag{3.27}$$

or alternatively:

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}^b_{b/o})\mathbf{q} \tag{3.28}$$

where:

$$\boldsymbol{\Omega}(\boldsymbol{\omega}^b_{b/o}) = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -(\boldsymbol{\omega}^b_{b/o})^T \\ \boldsymbol{\omega}^b_{b/o} & -\mathbf{S}(\boldsymbol{\omega}^b_{b/o}) \end{bmatrix}, \tag{3.29}$$

or, alternatively:

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \oplus \begin{bmatrix} 0 \\ \boldsymbol{\omega}^b_{b/o} \end{bmatrix} \tag{3.30}$$

3.3 Transformation Between Frames

These rotations are imperative for comparison of measurements and their respective reference models.

ECEF to ECI

This rotation is about the z axis, $z_e \equiv z_i$, i.e. the ECEF and ECI z-axis coincide. The rotation can be expressed as and angle $\alpha = \omega_e t$, where ω_e is the rotation rate of the Earth and t is the time passed since initial alignment of the ECEF and the ECI frame. The rotation from ECEF to ECI is mathematically given by

$$\mathbf{R}_e^i = \mathbf{R}_{z,\alpha} = \begin{bmatrix} cos\alpha & sin\alpha & 0 \\ -sin\alpha & cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.31}$$

ECI to Orbit

Rotating from the Earth Center Inertial to the orbit frame is trivial. The rotation matrix is found from the definition of the orbit frame, giving the following result:

$$\mathbf{R}_i^o = \begin{bmatrix} \mathbf{x}_o^i & \mathbf{y}_o^i & \mathbf{z}_o^i \end{bmatrix}^T \tag{3.32}$$

where $\mathbf{y}_o^i = \frac{\mathbf{v}\times\mathbf{r}}{|\mathbf{v}\times\mathbf{r}|}$, $\mathbf{z}_o^i = -\frac{\mathbf{r}}{|\mathbf{r}|}$, $\mathbf{x}_o^i = \mathbf{y}_o^i \times \mathbf{z}_o^i$. $\mathbf{y}_o^i$ is parallel with the orbit anti-normal, which is found from the cross product of the velocity, $\mathbf{v}$, and position vector, $\mathbf{r}$. $\mathbf{z}_o^i$ is parallel with the negative of the position vector, and $\mathbf{x}_o^i$ completes the right handed coordinate system.

Body to Orbit

The rotation can be expressed using (3.18) and by substituting the Euler parameters $\boldsymbol{\lambda} = \boldsymbol{\epsilon}$ and $\beta = \eta$. From body to orbit this yields [22]:

$$\mathbf{R}_{\boldsymbol{\epsilon},\eta} = \mathbf{R}_b^o = \mathbf{1} + 2\eta\mathbf{S}(\boldsymbol{\epsilon}) + 2\mathbf{S}^2(\boldsymbol{\epsilon}) = \begin{bmatrix} 1-2(\boldsymbol{\epsilon}_2^2+\boldsymbol{\epsilon}_3^2) & 2(\boldsymbol{\epsilon}_1\boldsymbol{\epsilon}_2-\boldsymbol{\epsilon}_3\eta) & 2(\boldsymbol{\epsilon}_1\boldsymbol{\epsilon}_3+\boldsymbol{\epsilon}_2\eta) \\ 2(\boldsymbol{\epsilon}_1\boldsymbol{\epsilon}_2+\boldsymbol{\epsilon}_3\eta) & 1-2(\boldsymbol{\epsilon}_1^2+\boldsymbol{\epsilon}_3^2) & 2(\boldsymbol{\epsilon}_2\boldsymbol{\epsilon}_3-\boldsymbol{\epsilon}_1\eta) \\ 2(\boldsymbol{\epsilon}_1\boldsymbol{\epsilon}_3-\boldsymbol{\epsilon}_2\eta) & 2(\boldsymbol{\epsilon}_2\boldsymbol{\epsilon}_3+\boldsymbol{\epsilon}_1\eta) & 1-2(\boldsymbol{\epsilon}_1^2+\boldsymbol{\epsilon}_2^2) \end{bmatrix} \tag{3.33}$$

and the finite rotation is given by:

$$\mathbf{R}_o^b = (\mathbf{R}_b^o)^T \tag{3.34}$$

ECEF Cartesian to ECEF Ellipsoidal

The GPS system provides estimates of the antenna positions in the ECEF coordinate system, and in order to describe the location of a point in the ECEF frame, two different coordinate systems are commonly used. The cartesian version is described in 3.1. However, it is often more desirable to represent the ECEF coordinates using a geodetic system and the ellipsoidal longitude, latitude, and height coordinates.

The longitude, l, is easily computed:

$$l = arctan\left(\frac{y_e}{x_e}\right) \tag{3.35}$$

whereas the latitude and the height can be computed using the following algorithm [8]:

1. Compute $p = \sqrt{x_e^2 + y_e^2}$ and $e = \sqrt{1 - \frac{a}{b}}$

2. Compute the approximate value $\mu_{(0)}$ from

$$tan(\mu_{(0)}) = \frac{z_e}{p}(1 - e^2)^{-1} \tag{3.36}$$

3. Compute an approximate value $N_{(0)}$ from

$$N_{(0)} = \frac{a^2}{\sqrt{a^2 cos^2(\mu_{(0)}) + b^2 sin^2(\mu_{(0)})}} \tag{3.37}$$

4. Compute the ellipsoidal height by

$$h = \frac{p}{cos(\mu_{(0)})} - N_{(0)} \tag{3.38}$$

5. Compute an improved value for the latitude by

$$tan(\mu) = \frac{z_e}{p}\left(1 - e^2 \frac{N_{(0)}}{N_{(0)} + h}\right)^{-1} \tag{3.39}$$

6. Check for another iteration step: if $\mu = \mu_{(0)}$ then the iteration is completed. Otherwise set $\mu_{(0)} = \mu$ and continue from step 3. and onwards.

This thesis will use the WGS84 geodetic system. The WGS84 defining parameters are listed in Table 3.1.

Name	Symbol	Value	Units
Equatorial radius (semi-major axis)	a	6378137	m
Polar axis radius (semi-minor axis)	b	6356752	m
Reciprocal flattening	$\frac{1}{f}$	298.257223563	
Angular rate	ω_E	7.292115×10^{-5}	$\frac{rad}{s}$
Gravitational constant	$\mu_g = GM$	398600.5×10^9	$\frac{m^3}{s^2}$
Eccentricity	e	0.0818	

Table 3.1: WGS84 defining parameters

ECEF Ellipsoidal to ECEF Cartesian

The Cartesian coordinates can be expressed as

$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} (N+h)cos(\mu)cos(l) \\ (N+h)cos(\mu)sin(l) \\ (\frac{b^2}{a^2}N+h)sin(\mu) \end{bmatrix} \tag{3.40}$$

where N and $N_{(0)}$ in 3.3 and 3.3 is the radius of curvature in prime vertical. As with $N_{(0)}$ in 3.37, N is here obtained by computing

$$N = \frac{a^2}{\sqrt{a^2cos^2(\mu)+b^2sin^2(\mu)}} \tag{3.41}$$

Conversions between Cartiesian ECI coordinates and orbital parameters, and vice versa, are explained in Chapter 4.

Chapter 4

Motion for a Spacecraft in Low Earth Orbit

4.1 Environmental forces and moments

In low Earth orbit, the major environmental torques that affect the attitude are aerodynamic, magnetic, gravity gradient, and solar radiation.

In low Earth Orbit, the dominating force is the aerodynamic torque, whereas for high altitude spacecrafts the magnetic and gravity-gradient torques will become more important as the aerodynamic torque falls off exponentially with the distance from the Earth (see Table 4.1). Both the magnetic torque and the gravity gradient torque have the same functional dependence on the distance, and the structure of the spacecraft will impact on the relative strength between these forces to decide which one is dominant. The solar radiation torque is dominating for higher altitude orbits and throughout the interplanetary medium for interplanetary missions and is therefore not significant in a low Earth orbit. However, the solar radiation will contribute both to somewhat radiation pressure and differential heating of the spacecraft [35].

In a polar orbit special care must be taken because the longitude is not defined at the poles. Another source of concern is that the magnetic field is highly

Source	Dependence on distance from Earth	Region of Space where dominant
Aerodynamic	$e^{-\alpha r}$	Altitudes below ~500 km
Magnetic	$\frac{1}{r^3}$	~500 km to ~35000 km
Gravity gradient	$\frac{1}{r^3}$	~500 km to ~35000 km
Solar radiation	Independent	Interplanetary Space above synchronous altitude

Table 4.1: Environmental parameters

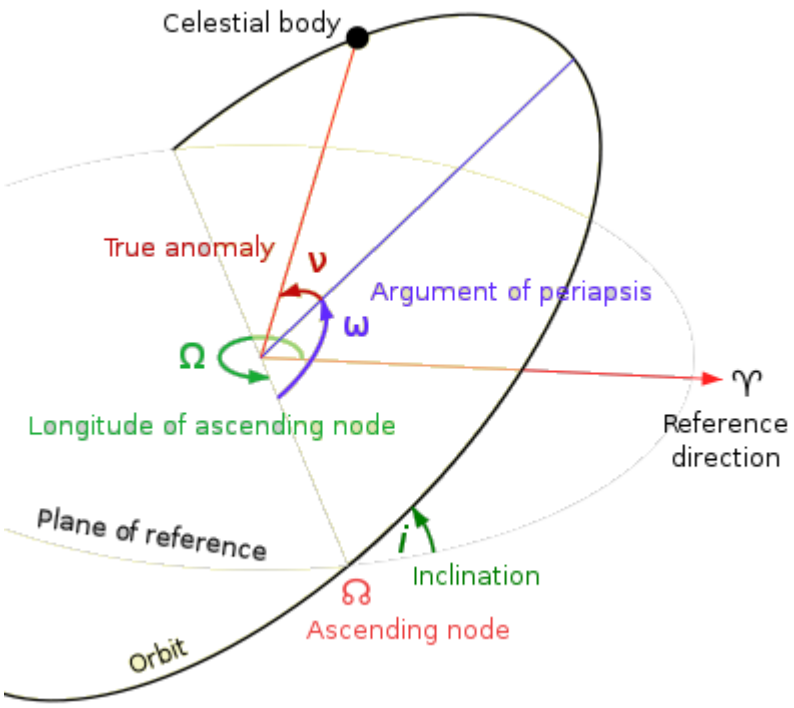

Figure 4.1: Keplerian Elements. Graphics by Lucas Snyder (2007)

fluctuating at the poles and has steep gradients in these areas.

4.2 Orbital Elements

In addition to representing an orbit using GPS position and velocity data, the six Keplerian orbital elements, introduced by Johannes Kepler (1570-1630) are more commonly used to represent the motion of a celestial body around its central body. They are estimated in this thesis using GPS navigation data. This is a problem which requires six quantities of integration in order to be solved for. The six classical orbital elements determine the size, shape and orientation of an orbit and are illusatrated in Figure 4.1 and described in this chapter.

Semi-major axis, a, describes the size of the ellipse. More specifically, this number represents half of the ellipses longest diameter. The semi-major axis runs from the ellipse center, through a focus and to the ellipse edge. In this case the orbit is circular, and the eccentricity represents the radius of the orbit, $e = 0$. The semi-major axis can also be replaced by the average angular velocity, the

mean motion, n, of the spacecraft, as these parameters are related via Keplers third law.

Eccentricity, e, is the number that describes the orbit's shape by indicating how much the orbit deviates from being a perfect circle. For a perfect circle the eccentricity is 0. In cases where the eccentricity reaches 1.0., the conic section becomes a parabolic trajectory and is no longer a closed orbit. Thus for orbital analysis we are only interested in eccentricity numbers between 0.0 and up to, but not including, 1.0. In the present case, we are only considering the case where e = 0.

Mean anomaly, M, describes the position of the satellite in the orbit. It is expressed as an angle, representing the fraction of an orbital period which has elapsed since perigee. In the case of a perfectly circular orbit, the mean anomaly equals the true anomaly, ν.

Inclination, i, is the angle between the orbital plane and the equatorial plane, by convention a number between 0 and 180 degrees.

Right ascension of the ascending node (RAAN), Ω, orients, together with the inclination, the orbital plane in space. The RAAN is the angle from the vernal equinox, where the RAAN is defined to be zero, to the ascending node, the point where the satellite passes through the equatorial plane moving from south to north. It is measured as a right-handed rotation about the Earth's pole, the ECI z-axis, and is by convention in the range 0 to 360 degrees. The RAAN is undefined for equatorial orbits.

Argument of perigee, ω, is the angle between the orbit's perigee (the point in the orbit which is closest to Earth) and the orbit's ascending node. This angle is measured in the direction of motion of the satellite in the orbital plane, hence orienting the orbit ellipse in the orbital plane. Its range of values is between 0 and 360 degrees, and it is undefined for equatorial and circular orbits.

4.2.1 Variations

Note that external forces acting on the satellite will induce variations in the satellite's orbital elements, causing the satellite to perturb away from the nominal Keplerian orbit due to which all the orbital elements will vary with time. The following secular variations neglect the variation caused by the changing orientation of the orbital plane with respect to the ecliptic plane and the Moon's orbital plane. However they are sufficiently accurate and included in the Simulink Chapter as perturbing forces [36]:

$$\dot{\Omega}_{Sun} = \frac{-0.00154 cos(i)}{orbits/day}, \tag{4.1}$$

Name of equation	Equation
Vis viva energy integral	$V^2 = \mu(\frac{2}{r} - \frac{1}{a})$
Angular momentum	$h = (\mu p)^{\frac{1}{2}}$
Kepler's equation	$M = E - esin\,E$
Radius equation, magnitude	$r = \frac{a(1-e^2)}{1+ecos\,\nu}$
Time rate of change of r	$\frac{dr}{dt} = \left(\frac{\mu}{p}\right)^{\frac{1}{2}} esin\,\nu$
Conversion of eccentric anomaly (E) to true anomaly (ν)	$cos\,\nu = \frac{cos\,E-e}{1-ecos\,E}$, $sin\,\nu = \frac{\left[(1-e^2)^{\frac{1}{2}} sin\,E\right]}{1-ecos\,E}$
Conversion of true anomaly, ν, to eccentric anomaly (E)	$cos\,E = \frac{(cos\,\nu+e)}{1+ecos\,\nu}$, $sin\,E = \frac{\left[\left(1-e^2\right)^{\frac{1}{2}} sin\,\nu\right]}{1+ecos\,\nu}$
Half-angle relation	$tan\left(\frac{\nu}{2}\right) = \left[\frac{1+e}{1-e}\right]^{\frac{1}{2}} tan\left(\frac{E}{2}\right)$
Flight-path angle, Υ	$tan\Upsilon = \frac{esin\,\nu}{1+ecos\,\nu}$
Mean anomaly at time, t	$M = M_0 + n(t - t_0)$
Perigee altitude	$h_p = a(1-e) - R_e$
Apogee altitude	$h_a = a(1+e) - R_e$

Table 4.2: Commonly Used Orbital Mechanics Equations

$$\dot{\Omega}_{Moon} = \frac{-0.00338cos(i)}{orbits/day}, \tag{4.2}$$

where i is the orbital inclination, set to $98^o * \frac{2\pi}{360}[rad]$.
$\dot{\Omega}$ is given in $\left[\frac{deg}{day}\right]$ and converted to $\left[\frac{rad}{second}\right]$ in the MATLAB code.

4.2.2 Conversion from Keplerian Orbital Parameters to ECI Cartesian Coordinates

The following equations are used to accomplish the conversion of classic orbit elements to ECI Cartesian coordinates [3]:

$$x = r(cos(\Omega)cos(u) - sin\,(\Omega)\,sin\,(u)\,cos\,(i)) \tag{4.3}$$

$$y = r(sin\,(\Omega)\,cos\,(u) + cos\,(\Omega)\,sin\,(u)\,cos\,(i)) \tag{4.4}$$

$$z = rsin\,(u)\,sin\,(i) \tag{4.5}$$

and:

$$\frac{dx}{dt} = V\left[\left(\frac{x}{r}\right) sin(\Upsilon) - cos(\Upsilon)(cos\,(\Omega)\,sin\,(u) + sin\,(\Omega)\,cos\,(i)\,cos\,(u))\right] \tag{4.6}$$

$$\frac{dy}{dt} = -V\left[\left(\frac{y}{r}\right) sin(\Upsilon) + cos(\Upsilon)(sin\,(\Omega)\,sin\,(u) - cos\,(\Omega)\,cos\,(i)\,cos\,(u))\right] \tag{4.7}$$

$$\frac{dz}{dt} = V[\left(\frac{z}{r}\right) sin(\Upsilon) + cos(\Upsilon)\,cos\,(u)\,sin\,(i))] \tag{4.8}$$

where V is the magnitude of the velocity.

Assuming a circular orbit with the perigee not defined and virtually chosen to be at the point of ascending node; $M \approx \nu$, $e \approx 0$, $\omega := 0$:

$$\begin{aligned} r &= a \\ u &= M \\ V &= \sqrt{\frac{GM}{a}} \\ \Upsilon &= 0 \end{aligned} \tag{4.9}$$

$$\begin{aligned} x^i &= a\,(cos(\Omega)cos(M) - sin(\Omega)sin(M)cos(i)) \\ y^i &= a\,(sin(\Omega)cos(M) + cos(\Omega)sin(M)cos(i)) \\ z^i &= asin(M)sin(i) \\ \dot{x}^i &= -V\,(cos(\Omega)sin(M) + sin(\Omega)cos(i)cos(M)) \\ \dot{y}^i &= -V\,(sin(\Omega)sin(M) - cos(\Omega)cos(i)cos(M) \\ \dot{z}^i &= Vcos(M)cos(i) \end{aligned} \tag{4.10}$$

These equations are linearized and used in the Multiplictive EKF.

4.2.3 Conversion from ECI Cartesian Coordinates to Keplerian Orbital Parameters

To convert from ECI Cartesian coordinates to Keplerian orbital parameters we must first find some auxiliary vectors:

$$\mathbf{h} = \mathbf{r} \times \mathbf{v} \tag{4.11}$$

is a vector perpendicular to the orbital plane,

$$\mathbf{n} = \mathbf{z} \times \mathbf{h}, \tag{4.12}$$

where $\mathbf{n}$ is a vector pointing towards the ascending node, $\mathbf{z}$ is a unit vector parallel to the ECI z-axis,

Using these auxiliary vectors we can compute the orbital parameters, which, assuming a circular orbit, becomes:

$$a = \frac{GM}{2GM - |\mathbf{r}|v^2}|\mathbf{r}|, \tag{4.13}$$

$$e := 0 \tag{4.14}$$

where v is the magnitude of the velocity vector,

$$i = arccos\left(\frac{h_z}{|\mathbf{h}|}\right) \tag{4.15}$$

$$M = arccos\left(\frac{\mathbf{n}\cdot\mathbf{r}}{|\mathbf{n}||\mathbf{r}|}\right) (\text{if } \mathbf{n}\cdot\mathbf{v} > 0 \ , \ \text{then } M = 2\pi - M) \tag{4.16}$$

$$\Omega = arccos\left(\frac{n_x}{|\mathbf{n}|}\right) (\text{if } n_y < 0 \ , \ \text{then } \Omega = 2\pi - \Omega) \tag{4.17}$$

$$\omega := 0 \tag{4.18}$$

If the inclination is zero, $i = 0$, then:

$$M = arccos\left(\frac{r_x}{|\mathbf{r}|}\right) (\text{if } v_x > 0 \ , \ \text{then } M = 2\pi - M) \tag{4.19}$$

$$\Omega := 0 \tag{4.20}$$

4.3 Equations of Motion for the Test Satellite

Satellite Dynamics

Modelling the satellite as a rigid body, the dynamics can be derived from Euler's moment equation as [21]:

$$\mathbf{I}\dot{\boldsymbol{\omega}}^b_{b/i} + \boldsymbol{\omega}^b_{b/i} \times \mathbf{I}\boldsymbol{\omega}^b_{b/i} = \boldsymbol{\tau}^b \tag{4.21}$$

where $\boldsymbol{\tau}^b$ is the sum of all torques acting on the satellite, $\boldsymbol{\omega}^b_{ib}$ is the angular velocity of the body frame relative to the ECI frame expressed in the body frame and I is the identity matrix representing the inertia of the satellite, expressed in the body frame. $\boldsymbol{\omega}^b_{ib}$ can be expressed by:

$$\boldsymbol{\omega}^b_{b/i} = \boldsymbol{\omega}^b_{o/i} + \boldsymbol{\omega}^b_{b/o} = \mathbf{R}^b_o\boldsymbol{\omega}^o_{o/i} + \boldsymbol{\omega}^b_{b/o}, \tag{4.22}$$

and applying the skew-symmetric operator on the satellite dynamics gives:

$$\mathbf{I}\dot{\boldsymbol{\omega}} + \mathbf{S}(\boldsymbol{\omega}^b_{b/i})\mathbf{I}\boldsymbol{\omega}^b_{b/i} = \boldsymbol{\tau}^b, \tag{4.23}$$

and:

$$\boldsymbol{\omega}^o_{o/i} = \begin{bmatrix} \mathbf{0} & -\boldsymbol{\omega}_o & \mathbf{0} \end{bmatrix} \tag{4.24}$$

is the known angular velocity of the orbit frame relative to the ECI frame, expressed in the orbit frame. The angular velocity, ω_o, a function of the altitude of the orbit, can be calculated using:

$$\boldsymbol{\omega}_o = \sqrt{\frac{GM}{r^3}} \tag{4.25}$$

where G is the gravitational constant of the Earth, M is the mass of the Earth and r is the distance from the satellite to the centre of the Earth. The spacecraft will be nominally stabilized in an earth pointing attitude. The angular velocity in the body frame relative to ECI can thus be expressed:

$$\boldsymbol{\omega}^b_{b/i} = \boldsymbol{\omega}^b_{b/o} - \boldsymbol{\omega}^o_{o/i} \tag{4.26}$$

Furthermore, using 4.22 and 4.21 we can express $\dot{\boldsymbol{\omega}}^b_{b/o}$**,** the angular velocity of the body frame relative to the orbit frame:

$$\dot{\boldsymbol{\omega}}^b_{b/o} = \dot{\boldsymbol{\omega}}_{inertial} + \dot{\boldsymbol{\omega}}_{torque} + \dot{\boldsymbol{\omega}}_{add} \tag{4.27}$$

where:

$$\dot{\boldsymbol{\omega}}_{inert} = \mathbf{I}^{-1}\left[-(\boldsymbol{\omega}^b_{b/o} + \mathbf{R}^b_o\boldsymbol{\omega}^o_{o/i}) \times (\mathbf{I}(\boldsymbol{\omega}^b_{b/o} + \mathbf{R}^b_o\boldsymbol{\omega}^o_{o/i}))\right] \tag{4.28}$$

$$\dot{\boldsymbol{\omega}}_{torque} = \mathbf{I}^{-1}\boldsymbol{\tau}^b \tag{4.29}$$

$$\dot{\boldsymbol{\omega}}_{add} = \mathbf{S}(\omega^b_{b/o})\mathbf{R}^b_o\boldsymbol{\omega}^o_{o/i} \tag{4.30}$$

Satellite Kinematics

As argued above, the attitude estimates will be represented by quaternions to prevent the existence of singularities. The satellite kinematics differential equations are given by [21]:

$$\dot{\eta} = -\frac{1}{2}\boldsymbol{\epsilon}^T\boldsymbol{\omega}^b_{b/o} \tag{4.31}$$

and

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2}\boldsymbol{\eta}\boldsymbol{\omega}^b_{b/o} - \frac{1}{2}\boldsymbol{\omega}^b_{b/o} \times \boldsymbol{\epsilon}. \tag{4.32}$$

The latter can also be written as:

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2}\left[\eta\mathbf{I} - \mathbf{S}\left(\boldsymbol{\epsilon}\right)\right]\boldsymbol{\omega}_{b/o}^{b}, \tag{4.33}$$

when applying the skew-symmetric operator $\mathbf{S}$.

Chapter 5

Sensors and Reference Models

GPS

Mainly three reasons can be argued for installing a GPS receiver onboard this double CubeSat. For once, the magnetometer measurements need to be compared to the IGRF model in order to determine the attitude of the satellite, hence the need for position data which can be obtained using GPS. It is imperative for the attitude determination system that the position is known when using the magnetic field for this purpose. Second, using multiple GPS antennas enables for attitude determination using GPS signals, which is designed and simulated in this thesis. Third, the GPS receiver could be utilized as an additional payload to measure occultations in the lower atmosphere. However, utilization of a GPS receiver for this purpose would require a dual frequency receiver [29], which is not necessary for determining the orbit of the small satellite. Scientific experiements on occulatations in the lower atmosphere was therefore disregarded, as a dual frequency GPS receiver would cause the already tight budgets of the project to exceed their limits, in the sense of volume, weight, power, and cost.

There are some obstacles to be overcome in order to install a GPS receiver onboard a satellite in orbit around Earth. These limitations were imposed by the Coordinating Committee for Multilateral Export Controls (CoCom) in the 1960s, with the intention to avoid the use of GPS in intercontinental ballistic missile-like applications, demanding all GPS devices disabled for any GPS device detected to be travelling at speeds higher than 1,900 km/h (527,78 m/s) at altitudes higher than 18,000 m. (59,000 ft). Although CoCom ceased to function on March 31, 1994, most manufactureres still apply these limits to GPS receivers, when either one or both limits are reached [29].

5.1 Inertial Navigation Systems (INS)

Gyrocope

A gyroscope is an inertial sensor which measures the angular velocity about the sensor axis of the object to which it is attached, relative to the inertial frame. Gyroscopes can measure rapid changes in the attitude, and the satellite's orientation can be obtained by integrating the measured angular velocity. A potential source of error could be incorrectly initialization of the gyroscope, which is of great importance regarding the quality of the estimates. It is also important that the gyro bias is not too large. Due to the gyro bias, which is inherent within all gyroscopes, gyroscopes can never be used single handedly in order to obtain attitude estimates, and they should always be used in conjunction with other sensors [12]. Furthermore, gyro biases are typically low-dynamic or approximately constant, such that the gyro can track the subject body's orientation up to a certain point. However, the inherent bias of the gyro will cause the attitude estimates to drift and the gyroscope needs to be calibrated accordingly. The length of time a gyroscope will provide acceptable measurements can vary. However, most modern gyroscopes provide high quality measurement data at reasonable cost.

Gyro Measurement The gyroscope measurement is composed of a three component vector:

$$\boldsymbol{\omega}_{b/i}^{b} = \boldsymbol{\omega}_{o/i}^{b} + \boldsymbol{\omega}_{b/o}^{b} \tag{5.1}$$

where $\boldsymbol{\omega}_{o/i}^{b}$ is the orbit frame rotation relative to ECI and $\boldsymbol{\omega}_{b/o}^{b}$ is the rotational velocity of the body frame relative to the orbit frame.

For an orbit of 600 km, one revolution about the Earth$\approx$ 96.67minutes, giving an orbit frame rotation of 0.062 degrees per second. Due to sensor noise this rotation will be difficult to measure.

We consider the orbit frame to be the inertial frame for local navigation: $\boldsymbol{\omega}_{b/i}^{b} \approx \boldsymbol{\omega}_{b/o}^{b}$

Gyro Error Model The gyro error model is [33]:

$$\boldsymbol{\omega}_{b/i}^{b} = \left[\mathbf{I} + \boldsymbol{\Delta}(\boldsymbol{\kappa}, \boldsymbol{\alpha})\right] \boldsymbol{\omega}_{imu} + \mathbf{b}_g + \mathbf{w}_1 \tag{5.2}$$

where $\mathbf{b}_g \in \mathbb{R}^3$ represent gyro bias.

$\mathbf{w}_1 \in \mathbb{R}^3$ is bounded unmodeled errors and measurement noise. $\boldsymbol{\Delta} = \boldsymbol{\Delta}(\boldsymbol{\kappa}, \boldsymbol{\alpha}) = \begin{bmatrix} \kappa_x & \alpha_{xy} & \alpha_{xz} \\ \alpha_{yx} & \kappa_y & \alpha_{yz} \\ \alpha_{zx} & \alpha_{zy} & \kappa_z \end{bmatrix}$

$\boldsymbol{\kappa} = \begin{bmatrix} \kappa_x & \kappa_y & \kappa_z \end{bmatrix}^T$ are three gyro scale factor errors, and,

$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_{xy} & \alpha_{xz} & \alpha_{yx} & \alpha_{yz} & \alpha_{zx} & \alpha_{zy} \end{bmatrix}^T$ are six small gyro misalignment angles.

This error model is substituted into the following equations:

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} -\boldsymbol{\epsilon}^T \\ \eta \mathbf{I}_{3\times 3} + \mathbf{S}(\boldsymbol{\epsilon}) \end{bmatrix} \boldsymbol{\omega}^b_{b/i} - \frac{1}{2} \begin{bmatrix} -\boldsymbol{\epsilon}^T \\ \eta \mathbf{I}_{3\times 3} + \mathbf{S}(\boldsymbol{\epsilon}) \end{bmatrix} \boldsymbol{\omega}^o_{o/i}, \tag{5.3}$$

which are the inertial navigation system attitude determination equations using unit quaternions [33]. The substitution yields:

$$\dot{\mathbf{q}} = \mathbf{T}_q(\mathbf{q}) \left[(I + \triangle(\boldsymbol{\kappa}, \boldsymbol{\alpha})) \boldsymbol{\omega}_{imu} + \mathbf{b}_g + \mathbf{w}_1 \right] - \Xi(q) \boldsymbol{\omega}^o_{o/i}, \tag{5.4}$$

where:

$$\mathbf{T}_q(\mathbf{q}) = \frac{1}{2} \begin{bmatrix} -\boldsymbol{\epsilon}^T \\ \eta \mathbf{I}_{3\times 3} + S(\boldsymbol{\epsilon}) \end{bmatrix} \tag{5.5}$$

$$\Xi(\mathbf{q}) = \frac{1}{2} \begin{bmatrix} -\boldsymbol{\epsilon}^T \\ \eta \mathbf{I}_{3\times 3} + S(\boldsymbol{\epsilon}) \end{bmatrix} \tag{5.6}$$

The gyro error models are described by the 1st-order models [33]:

$$\dot{\mathbf{b}}_g = -\mathbf{T}_1^{-1} \mathbf{b}_g + \mathbf{w}_2 \tag{5.7}$$

$$\dot{\boldsymbol{\kappa}} = -\mathbf{T}_2^{-1} \boldsymbol{\kappa} + \mathbf{w}_3 \tag{5.8}$$

$$\dot{\boldsymbol{\alpha}} = -\mathbf{T}_3^{-1} \boldsymbol{\alpha} + \mathbf{w}_4 \tag{5.9}$$

where $\mathbf{w}_2, \mathbf{w}_3 \in \mathbb{R}^3$ and $\mathbf{w}_4 \in \mathbb{R}^6$ are Gaussian white noise signals, and,

$\mathbf{T}_1, \mathbf{T}_2 \in \mathbb{R}^{3\times 3}$ and $\mathbf{T}_3 \in \mathbb{R}^{6\times 6}$ are diagonal matrices of time constants.

$\boldsymbol{\omega}^b_{b/o}$ can be found from:

$$\begin{aligned} \boldsymbol{\omega}^b_{b/o} &= \boldsymbol{\omega}^b_{b/i} - \boldsymbol{\omega}^b_{o/i} \\ &= \boldsymbol{\omega}^b_{b/i} - \mathbf{R}^b_o \boldsymbol{\omega}^o_{o/i} \end{aligned} \tag{5.10}$$

where

$$\boldsymbol{\omega}^o_{o/i} = \begin{bmatrix} 0 & \omega_o & 0 \end{bmatrix}^T \tag{5.11}$$

and $\omega_o = \sqrt{\frac{GM}{a^3}}$ for a circular orbit.

Accelerometer

Another sensor inherent in an inertial navigation system is the accelerometer, which measures change in velocity. For a satellite in orbit the gravity force upon the satellite can be found by:

$$F_g = G\frac{Mm}{r^2} \tag{5.12}$$

where:

G is the universal gravity constant defined in Table 3.1,

M is the mass of the Earth.

m is the mass of the satellite, and,

r is the distance between the center of mass of the two bodies.

In low Earth orbit the acceleration due to the Earth's gravity is 8.19 $\left[\frac{m}{s^2}\right]$, and due to third bodies are e.g. $6 \cdot 10^{-4}$ $\left[\frac{m}{s^2}\right]$ due to the Sun, $3 \cdot 10^{-6}$ $\left[\frac{m}{s^2}\right]$ due to the Moon and $3 \cdot 10^{-8}$ $\left[\frac{m}{s^2}\right]$ due to Jupiter. The accelerations due to third bodies are very small, and we therefore consider the isolated case of the spacecraft orbiting Earth. The isolated resulting gravitational force is balanced out by the centripetal force, causing a resultant acceleration of approximately 0 g [12]. In this thesis we consider the case where the satellite is in a stable circular orbit around the Earth, meaning that the spacecraft is in continuous free fall towards the Earth. The total acceleration experienced by the spacecraft in orbit will be small and regular accelerometers can not be utilized for this purpose. Accelerometers may be used for testing on ground but they are not sensible to utilize in orbit.

Magnetometer

A magnetometer can be used to obtain measurements of the flux density of the local magnetic field at the current location of the satellite. By combining measurements from three mutually perpendicular magnetometers the magnetic field including two components of its direction will be given in the body frame. This information can also be used as a check on the measurement accuracy of the carrier phase measurements used by the GPS attitude determination solution method (see 7.1.3) and vice versa.

There are various methods to gain knowledge of the real magnetic field of the Earth. One way of doing this is by using a look-up table. This approach would require data uploaded from ground in order for it not to be too extensive for the storage capacity on-board. As the real magnetic field varies in magnitude and direction with its location and altitude, in addition to slowly changing with time, a stand-alone lookup table would have to compensate for with an infinite amount of combinations. An option would be to update a smaller table through ground communication with information about the magnetic field for a certain range of satellite orbit positions, easing the workload of the onboard computer. This would impose a need for continuous communication with ground stations, a potential and probable source of attitude estimation drop-out in the case of communication failure. A look-up table solution would also require communication bandwidth, which already is a limited resource for a small student satellite. It was therefore decided to use a mathematic model of the field; the International Geomagnetic Reference Field (IGRF), see Section 5.2.

Magnetometer Error Model The magnetometer error model is [33]:

$$\mathbf{m}^e_{imu} = \mathbf{m}^e + \mathbf{b}_m + \mathbf{w}_5, \tag{5.13}$$

where:

$\dot{\mathbf{b}}_m = -\mathbf{T}_4^{-1}\mathbf{b}_m + \mathbf{w}_6$ which is the local magnetic disturbance.

$\mathbf{w}_5, \mathbf{w}_6 \in \mathbb{R}^3$ are Gaussian white noise.

It is assumed that no residual magnetic field from the magnetic torquers are significant in the magnetometer measurements and that the necessary navigation data are provided for the IGRF model.

5.2 The International Geomagnetic Reference Field (IGRF)

The International Geomagnetic Reference Field is a mathematical model of the Earth's real magnetic field, which provides reference for the attitude estimator [12]. The geomagnetic field of the Earth provides a reference vector that can be obtained using a variety of mathematical models. The field is independent of orbit characteristics and it is generated by several sources and composed by several magnetic fields. The most important sources are the magnetosphere, the ionospere, the deep fluid interior of the Earth, the Earth's crust and its upper mantel [30]. The resulting magnetic field is simulated using one of the available models, the International Geomagnetic Reference Field model provided by the International Association of Geomagnetism and Aeronomy (IAGA). This model takes the form of a spherical harmonic equation:

$$\mathbf{V} = \sum_{n=1}^{N} \sum_{m=1}^{n} \left(\frac{a_{igrf}}{r_{igrf}} \right)^{n+1} (g_n^m cos(m\phi_{igrf}) + h_n^m cos(m\phi_{igrf})) P_n^m cos(\theta_{igrf}) \tag{5.14}$$

where:

g_n^m and h_n^m are Gauss coefficients.

a_{igrf} is the mean radius of the Earth.

r_{igrf} is the distance from the center of the Earth.

ϕ_{igrf} is the longitude east of Greenwich.

θ_{igrf} is the colatitude (90^o-latitude).

The resulting IGRF model, V, is an empirical representation of direction and magnitude of the Earth's main composite magnetic field, predicted to within approximately 100 nT. It holds maximum temporal variations of the field of about 1% per year [2]. The model is updated every 5 years, the 11^{th} generation IGRF model is of order 13 with coefficients finalized in December 2009. Available at IAGA V-MOD Geomagnetic Field Modeling, the IGRF 2010.0 coefficients are downloaded and used in this thesis. The resulting field is illustrated in Figure 5.1, 5.2, and 5.3. The desired local magnetic field vector is found from this model using positioning data acquired from the GPS system, and spacecraft attitude is then obtained by rotating the IGRF vector from ECEF to the orbit frame via ECI in the Kalman filter. Figure 5.4 shows the IGRF resulting Matlab simulation.

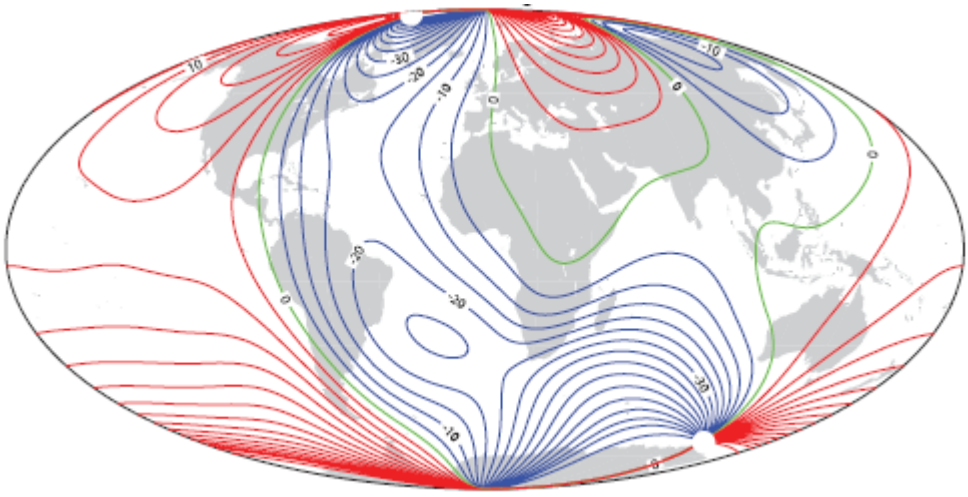

Figure 5.1: Declination D in degrees in 2010

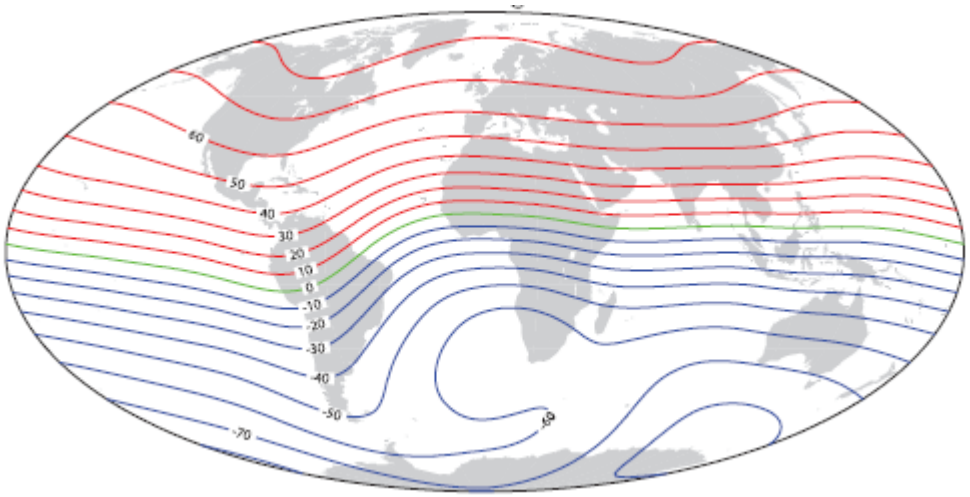

Figure 5.2: Inclination I in degrees in 2010

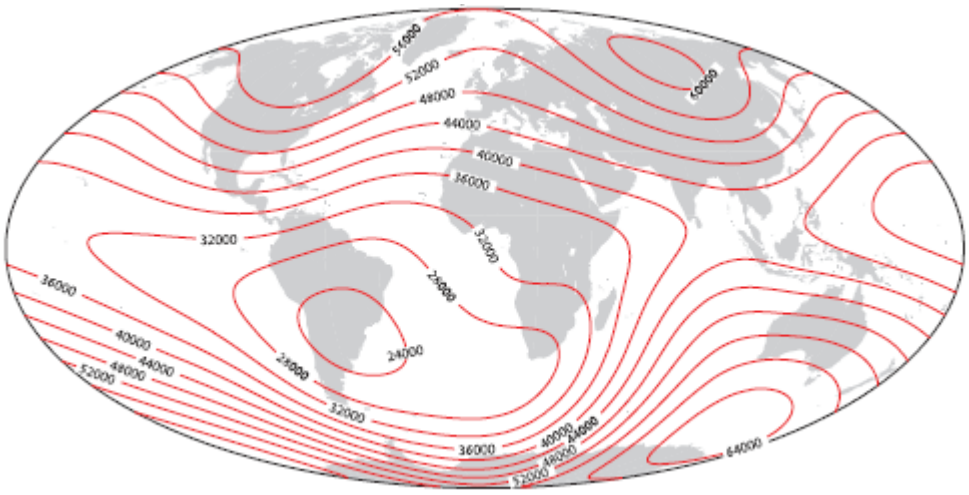

Figure 5.3: Total Intensity F in nT in 2010

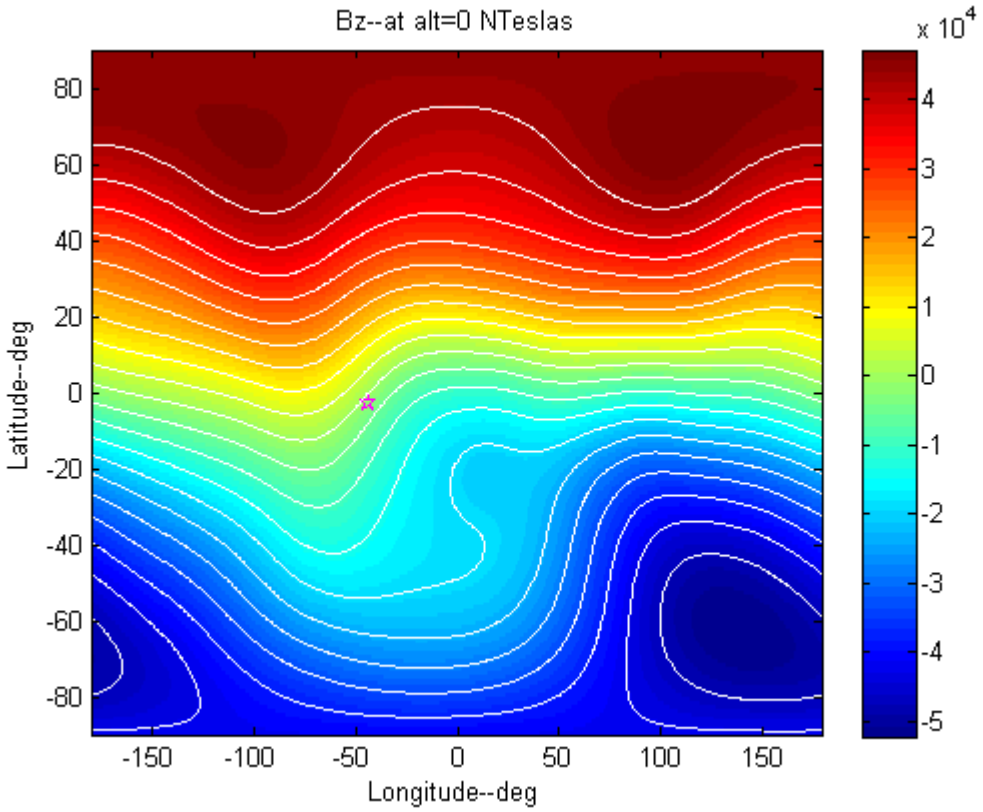

Figure 5.4: Matlab simulation the resulting International Geomagnetic Reference Field

Chapter 6

GPS and Inertial Navigation System Integration

6.1 Position, Velocity, and Attitude Aiding

In a strapdown inertial navigation system the sensors are rigidly attached to the vehicle in the body frame, and as the vehicle moves, the gyros experience the full rotational rate of the movement. Therefore the bias accuracy, as well as the gyro scale factor accuracy, is important. By integrating GPS with rate gyro, better calibration of gyro errors can be achieved. Integrating GPS with rate gyro also enables using gyro attitude data to reduce the integer search space for integer ambiguity resolution to shorten the time spent for reinitialization of integer ambiguities in case of GPS loss of lock (See 7.1.4). Several architectures may be used in the integration design, depending on the purpose and system requirements, with respect to complexity, redundancy, and flexibility [33]. This subchapter discusses the various integration methods, and concludes that the preferred method for the small satellite GPS/INS design is the loosely coupled design. Note that for any design, the instruments should be carefully placed in the satellite to avoid interference from other components.

6.1.1 Uncoupled Integration

The uncoupled integration architecture is the simplest and fastest method. It subtracts the estimated INS errors from the INS measurements in an open-loop manner to give the output signal. It is also the potentially cheapest, as INS and GPS sensors can be easily replaced. A high degree of tolerance to subsystems failure is achieved, as well as a high degree of redundancy. However, the validity of the filter models and their performance will degrade due to INS output drift over time. This architecture is only applicable to short duration applications and is therefore not a sufficiently robust method for a satellite in orbit.

6.1.2 Loosely Coupled Integration

The loosely coupled configuration method operates in a closed-loop, feeding the estimated INS errors back to the INS strapdown computation block. Several possible variations of feedback exist for a loosely coupled configuration [33]. For terrestrial use, acceleration information from the INS can be used to improve the smoothing properties of the GPS Kalman filter, but applications to be used in a low Earth orbit will not inherit this advantage, as the velocity is approximately constant throughout orbit lapses and the result acceleration is close to zero. By using the loosely coupled integration design, the position and velocity information is fed back to the GPS receiver. This helps to decrease the reacquisition time after loss of lock of GPS signals. Error estimates of the GPS error estimates and the INS error estimates are fed back from the integration filter to the GPS Kalman filter and to the strapdown computation block, respectively, serving to sustain small input values to the integration filter such that the validity of the models are maintained for a longer period of time. This feedback, or reset, as it is often denoted in literature, also ensures continuous calibration of the INS in case of GPS dropout. An example of loosely coupled integration filter can be found in Grip, Fossen, Johansen, and Saberi (2012).

6.1.3 Tightly Coupled Integration

A tightly coupled integration scheme uses raw accelerometer, gyro, pseudorange, and deltarange measurements from the sensors, instead of position, velocity, and attitude data which require at least 4 GPS SV's within line of sight. The pseudoranges and deltaranges acquired from GPS SVs will provide information to calibrate the INS during GPS dropout, even in the case where less than the necessary amount of GPS SVs are avilable for a position, velocity, timing, and attitude solution is available. As opposed to loosely coupled integration where the inputs to the integration filter are correlated, injecting raw measurements to the integration filter gives a better solution in terms of accuracy, as the raw measurements are uncorrelated. Tight coupling also provides velocity aiding of the GPS receiver loops, lowering the bandwidth of the GPS tracking loops and hence increasing jamming resistance. A drawback, however, is that this solution increases complexity and computational requirements, and a tightly coupled integration scheme is not necessary in terms of GPS SV visibility in low Earth orbit. Neither is it likely that the experimental student satellite will be subject to targeted jamming.

6.1.4 Deeply Coupled Integration

The deeply coupled integration method allows for the most optimal use of raw GPS and IMU data, as well as for control of all loop and filter bandwidths to optimize filter and correlator bandwidths. Due to its complexity, this design was not considered in this thesis.

6.1.5 Direct and indirect Integration

The modeling of position, velocity, and attitude variables can also be separated into *direct* and *indirect* integration methods:

Direct Integration

The direct integration approach estimates the states in the filter, and is a well suited approach for tightly coupled integration (see 6.1.3). Direct integration does not require implementation and calculation of error models, and hence less computation need to be done compared to the indirect integration approach. However, this argument does not hold if the integration filter is a Kalman filter, as the covariance matrix in a Kalman filter must be updated at a high iteration rate. Using a Kalman filter, the direct approach requires more computer througput and higher iteration rate, as the dynamics of the states is significanly larger than the dynamics of the error states. [33] The latter are used in the indirect approach, as described in the following section.

Indirect Integration

Indirect integration filters feed error estimates back to the strapdown equations in order to update the whole estimates, as opposed to estimates of the whole estimates. As the dynamics of the error estimates are smaller and slower than the dynamics of the whole states, a slower update rate can be used. This significantly reduces the load on covariance matrix computation in the integration filter when a Kalman filter is used for this purpose, see (6.1.5), which is implemented in this thesis in the Multiplicative Extended Kalman Filter design.

6.1.6 Choice of Method

In this thesis, a loosely coupled system for position and velocity aiding was chosen for convenience. A tightly coupled system for attitude aiding was chosen based on its advantages in robustness and accuracy.

6.1.7 Lever Arm Compensation (Position, Velocity, Acceleration)

The inertial navigation system of the craft should be mounted as close as possible to the center of gravity of the platform, such as to minimize the error in level arm compensation. This can be seen from the expressions [33]:

$$\mathbf{r}_0^o = \mathbf{r}_1^o + \mathbf{R}_b^o \triangle \mathbf{r}^b \tag{6.1}$$

$$v_0^o = \mathbf{r}_1^o + \mathbf{R}_b^o \mathbf{S}(\boldsymbol{\omega}_{b/o}^b) \triangle \dot{\mathbf{r}}^b \tag{6.2}$$

and:

$$\mathbf{a}_0^o = \mathbf{a}_1^o + \mathbf{R}_b^o \mathbf{S}^2(\boldsymbol{\omega}_{b/o}^b) \triangle \mathbf{r}^b + \mathbf{R}_b^o \mathbf{S}(\dot{\boldsymbol{\omega}}_{b/o}^b) \triangle \mathbf{r}^b \approx a_1^o + \mathbf{R}_b^o \mathbf{S}^2(\boldsymbol{\omega}_{b/o}^b) \triangle \mathbf{r}^b, \quad (6.3)$$

which represents the constant lever arm $\triangle \mathbf{r}^b$ relative to a point with position $\mathbf{r}_0^o$, and the lever arm compensation for position, velocity, and acceleration, respectively. When the platform of the spacecraft changes attitude, the GPS receiver will measure a different position than the INS.

Chapter 7

Attitude Determination

7.1 GPS Attitude Determination

Signals from all the Global Navigation Satellite Systems (GNSSs), including the Russian Global'naya Navigatsionnaya Sputnikovaya Sistema (GLONASS) at 19,140 km, the Chinese Beidou at 21,150 km, the European Galileo at 23,222 km and the U.S. GPS system at 20,200 km altitude may in theory be utilized for attitude determination. For practical reasons, only the GPS system is considered in this thesis.

Attitude determination using GPS was first proposed by Spinney (1976), who developed the basic idea of determining the attitude of a craft exploiting the accuracy of differential range GPS carrier phase measurements, a function of the vehicle attitude. The concept is illustrated in Figure 7.1. In his work, Spinney also examined the basic performance of the solution in terms of differential carrier phase (DCP) measurement error, Geometric Dillution of Precision (GDOP), Attitude Dillution of Precision (ADOP), antenna baseline length, and antenna baseline knowledge accuracy. GPS attitude determination specifically applied for spacecraft was first suggested by Ellis (1979), whereas the first real-time experiments of GPS attitude determination was documented by Kruczynski (1989), based on experiments on the Ticonderoga-class missile cruiser in 1988. Following this, Van Graas and Braasch (1991) conducted the first aircraft attitude tests on a Douglas DC3 at Ohio University. The first practical, commercially available attitude receiver was produced as a result of research by Cohen (1992) at Stanford University, and several demonstrations of attitude determination based on GPS signals have been performed since the first spacecraft GPS attitude experiment flown on the Air Force RADCAL satellite launched in 1993 [1].

The direction of the incoming GPS signal is given by the unit length line-of-sight vector, $\mathbf{l}^e$, with respect to a baseline, $\mathbf{b}$. The LOS vector $\mathbf{l}^e$ is known from

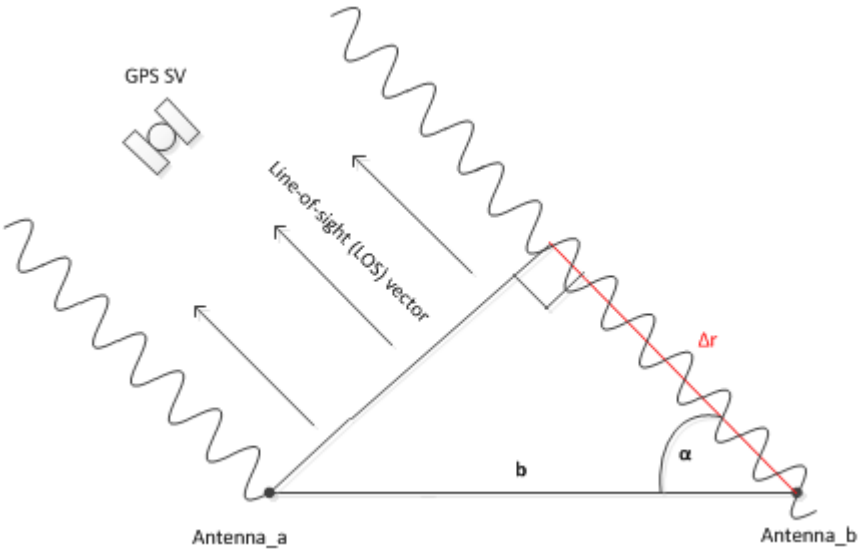

Figure 7.1: Attitude Observation Geometry

the GPS almanac and the test satellite's position and is always given in ECEF coordinates, i.e. $\mathbf{l}^e = \mathbf{l}$. The baseline is known from the predetermined and geometrical placement of the antennas illustrated in Figure 7.2. Figure 7.1 also illustrates that the line of sight vector is always considered to be parallel for all GPS antennas due to the comparatively large distance from the antennas to the GPS SV. The attitude angle α can be found by observing the relation:

$$\triangle r_{ij} = |\mathbf{b}_j| cos(\alpha_{ij}) = \mathbf{l}_i^T \mathbf{b}_j^e \tag{7.1}$$

where we can see that finding the delta range, $\triangle r_{ij}$, is the only obstacle to accessing α. The delta range can can not be measured directly, but the phase difference between the two antennas a and b can be measured by comparing the received signals.

This introduces the problem of finding the whole number of wavelengths, referred to as ambiguity parameter resolution, of the incoming GPS signal between the two antennas a and b. The ambiguity parameter is denoted N_{ij} for signal i and baseline j [33]:

$$N_{ij} = round\left(\frac{\triangle r_{ij} - \lambda \triangle \varphi_{ij}}{\lambda}\right) \tag{7.2}$$

where λ is the wavelength of the GPS signal, for which we have used the GPS L_1 frequency corresponding to a wavelength of $\lambda_{L_1} = 0.1903[m]$. The function $round(\bullet)$ finds the integer number closest to its argument. The integer ambiguity resolution problem is explained in Chapter 7.1.4, for which the Fast Integer

Ambiguity Resolution algorithm proposed in [16] by Lightsey, Crassidis, and Markley (1999) is implemented.

As the integer ambiguity can be resolved we can now resolve the attitude accordingly. The final GPS attitude determination equation for an arbitrary number of satellites is more conveniently written in a matrix form:

$$\lambda(\triangle\Phi + N) = L^T \mathbf{R}_b^e \mathbf{B}^b, \tag{7.3}$$

for which $\mathbf{R}_b^e$ represents the rotation matrix representing the unknown attitude we want to solve for.

The signal structure of relevance for this thesis is the L_1 carrier frequency:

$$L_1(t) = A_{L1}p(t)d(t)cos(f_1 t) + A_{L1}c(t)d(t)sin(f_1 t) \tag{7.4}$$

where:

$f_1 = 1575.42$ [MHz].

A_{L1} is the signal amplitude.

$c(t)$ and $p(t)$ are Pseudo Random Noise (PRN) sequences modulated onto L_1 at10.23 [MHz].

$d(t)$ is the data message containing information about satellite parameters, clock and clock errors for calculation of user position and velocity. This structure allows for distinguishment of signals sent from different satellites at the same carrier frequency.

A common oscillator approach described by (Vik, 2011) and a single GPS receiver is used to attain information for attitude calculation. The line-of-sight dependent, common mode carrier phase measurement disturbances can be removed by differencing, which requires at least four GPS SVs in sight [1] In low Earth orbit, the GPS signals are visible with an enhanced visibility of approximately 12-16 signals, compared to 6-8 for terrestrial applications[1]. Note that due to high orbital speeds in low Earth orbit an increased signal Doppler shift will be in the order of $\pm 40KHz$ compared to $\pm 5KHz$ on Earth, and that a receiver in LEO will be subject to rapidly shifting transmitter line-of-sights to the GPS satellites. A single GPS SV will be visible for ≈ 30 minutes compared to ≈ 6 hours on the ground. Consequently, the time and demands for signal acquisition will be higher.

The GPS measurement observables are the pseudorange, the signal Doppler shift, and the integrated carrier phase. During a cold start, the GPS receiver may search for PRN codes which are not visible in the given moment. Note also that spacecraft attitude motion can involve larger angles than will typically occur for terrestrial applications.

7.1.1 GPS Attitude Determination Equations with Clock Line Biases

The attitude information obtained from the differential phase measurements is given by [33]:

$$\triangle\varphi = N\lambda_{L1} + \beta = (\mathbf{l})^T R_b^o \mathbf{b}^b \tag{7.5}$$

where

$\triangle\varphi$ is the scalar differential phase measurement from the receiver.

N is the integer ambiguity.

λ_{L1} is the wavelength of the L1 carrier.

β is the line bias.

$\mathbf{b}^b$is the baseline vector decomposed in the body frame.

7.1.2 GPS Error Sources

There are several sources of error entering the carrier phase signal. The largest error source for attitude applications is multipath, which are errors in the phase measurement caused by reflections reaching the antennas. The only source of reflection in this case is the spacecraft itself, and the multipath effects are therefore small. Other errors entering the carrier phase signal is ionosphere/troposphere errors, which can be removed by differencing. There are also errors due to phase center variation of the antennas, receiver noise, baseline length error, and line and clock bias. These errors are combined to form one single error source in Matlab/Simulink.

The goal is to achive continuous attitude solution maintainance over the entire orbit of the spacecraft. An inertial navigation system is used as as an aid to achieve this goal. Integrated GPS/INS provides calibration of gyro errors, in addition to an increased observability of the states which significantly reduces alignment time. INS attitude data can also be used to reduce the integer search space, shortening the time spent for reinitialization of integer ambiguities after GPS loss of lock [33]. As a dedicated GPS receiver approach is suggested, β

here represents line bias. The line bias is the bias/error due to the electrical lengths of each baselines. These electrical lengths will vary but can be measured. The errors can be predetermined together with the body frame baselines using a static survey, but that may not be trivial, as temperature variations are significant so that line biases will change significantly due to these temperature differences between antenna cables. If so, this issue must be resolved and taken into account for using line bias estimation techniques [4] and is not considered further in this thesis.

7.1.3 Attitude Solution Method

A baseline, **b**, is the vector between two antennas, and one of the antennas are selected as a master antenna. In this thesis a configuration of four antennas has been simulated with A_1 , given by the body-frame coordinates [-0.05, -0.05, -0.10] m., as the master antenna. See Figure 7.2. This antenna is used as a reference giving the three baselines A_0 to A_1 , A_0 to A_2 and A_0 to A_3, a geometric configuration which avoids parallel baselines. Any parallell baselines will be surplus as they do not provide any new information. The four antenna configuration illustrated in Figure 7.2 has been implemented and provides a full non-coplanar solution which increases the performance of the GPS receiver in terms of reliability and accuracy [33]. Note that the baselines are not to scale in the figure.

Since we are here using only the L1 frequency, the measurement is:

$$\varphi = n_1\varphi_1 \tag{7.6}$$

where $\varphi = ft$ and $f = \frac{c}{\lambda}$. The measurement of GPS SV i (sightline) from antenna j (baseline) can be expressed as [33]:

$$\varphi_{i,j} = -N_{ij} + \frac{r_{ij}}{\lambda} + \frac{c}{\lambda}(\partial_i^r + \partial_j^s) + e_{ij} \tag{7.7}$$

where

r_{ij} is the pseudorange to the satellite i at antenna j

c is the speed of light

N_{ij} is the ambiguity number of whole periods from satellite i to antenna j

$\triangle\partial = \partial_i^r - \partial_j^s$, where ∂_i^r is the clock error in the receiver and ∂_j^s is the clock error in the GPS SV

e_{ij} is the error on the signal

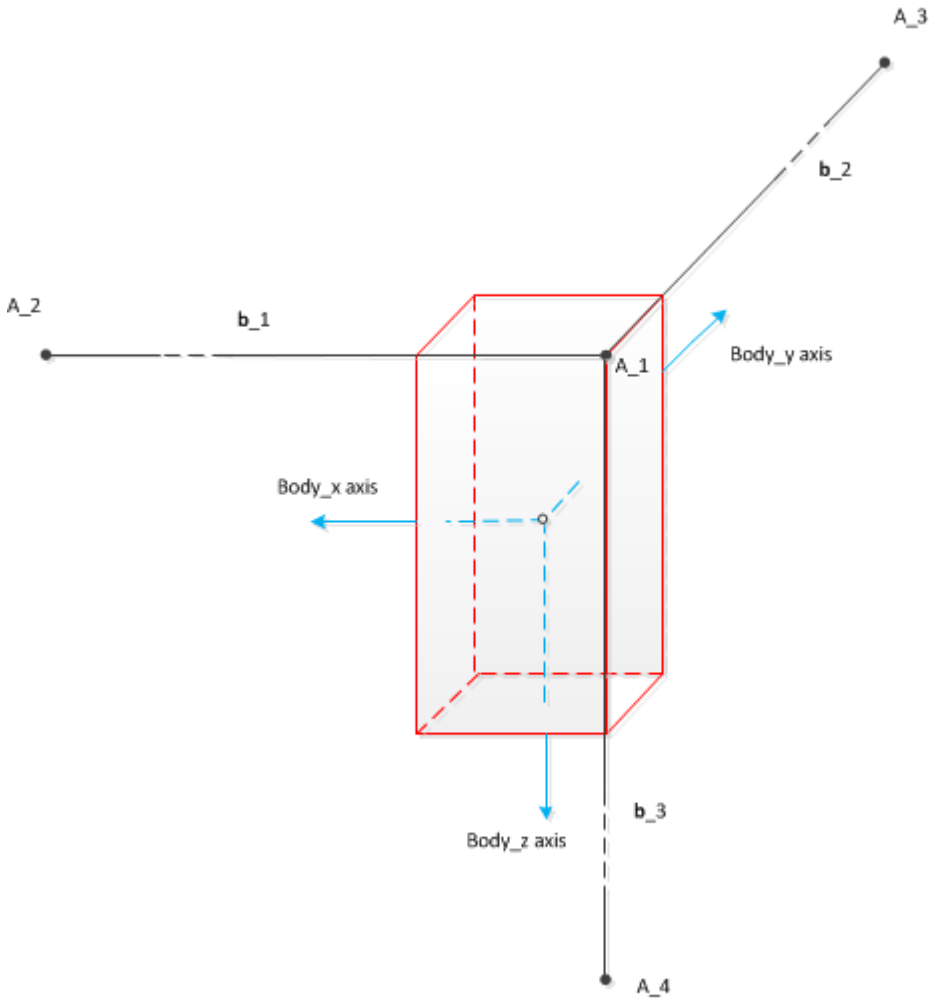

Figure 7.2: NUTS Antenna Configuration

φ_{ij} is the measured phase of signal i at antenna j

In this differencing technique, the difference between the measurements is used to eliminate noise and errors in the equations. Errors, such as the clock error in the GPS SV and the atmospheric and ionospheric noise related to the GPS signal are further reduced by differencing the signal with a reference antenna of choice onboard the small spacecraft. The single differentiated version is obtained [33]:

$$\triangle\varphi_{ij} = \triangle\varphi_{ij} - \triangle\varphi_{0j} = -\triangle N_{ij} + \triangle\frac{r_{ij}}{\lambda} + \frac{c}{\lambda}\triangle\partial_i^r + \triangle e_{ij} \tag{7.8}$$

The error terms associated with the GPS receiver onboard the test satellite, such as the receiver clock error, may also be removed. This further improvement of the measurement is obtained by differentiating with a reference GPS SV of choice, giving the following result:

$$\nabla\triangle\varphi_{ij} = \triangle\varphi_{ij} - \triangle\varphi_{i0} = -\nabla\triangle N_{ij} + \nabla\triangle\frac{r_{ij}}{\lambda} + \nabla\triangle e_{ij} \tag{7.9}$$

When measurements for double differencing are given, error residuals and uncorrelated noise remain. The most significant contributor to this is usually multipath, which is uncorrelated between the respective satellites as well as between the antennas. The spacecraft is small of size and the only source of reflection. Multipath noise is therefore simulated as part of the uncorrelated noise in Matlab.

Eq. (7.9) may also be written on matrix form as [33]:

$$\mathbf{A}\lambda(\triangle\varphi + N) = \mathbf{A}L^T\mathbf{b}^e \tag{7.10}$$

where A represents the differencing matrix which transform the n single differenced measurements into a $n-1$ double differenced system. Since we want to both keep the integer property of the ambiguity parameter N while exploiting the advantage of having uncorrelated noise in the Kalman filter, we choose to split A into one integer part and one scaling part:[33]:

$$\mathbf{A} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{6}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{12}} \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & -2 & 0 \\ 1 & 1 & 1 & -3 \end{bmatrix} \tag{7.11}$$

The phase measurement are made up of the differential range measurements between any baseline pair and a GPS SV. An indicator of the performance of a GPS attitude determination system is given by the GPS satellite constellation and baseline configuration dependent ADOP value:

$$ADOP = \left[trace \left(\left(\mathbf{L}^b \right)^T \mathbf{A}\mathbf{A}^T \mathbf{L}^b \right) I_{2\times 2} - A^T \mathbf{L}^b \left(\mathbf{L}^b \right)^T \mathbf{A} \right]^{-1} \tag{7.12}$$

From Cohen (1996) it was found that ideal geometries for the baseline configuration with the property:

$$\mathbf{B}\mathbf{B}^T = \mathbf{l}\mathbf{I}_{3\times 3} \tag{7.13}$$

results in the ADOP value:

$$ADOP = \sqrt{trace(n\mathbf{I} - \mathbf{L}^T\mathbf{L})^{-1}} \tag{7.14}$$

where n is the number of LOS vectors

L is a $n \times 3$ matrix of LOS vectors.

Precision is dependent on the noise level on the measurements and the ADOP value, whereas the most important factors determining the ADOP value are the baseline configuration and the geometric properties of the GPS Space Vehicle constellation. This thesis uses ADOP values to select four GPS satellites for carrier phase measurements.

7.1.4 Integer Ambiguity Resolution

When phase differences are used to determine a vehicle's attitude, it is necessary to first resolve the integer ambiguity problem, finding the correct number of integer wavelenghts between a given pair of antennas [16]. There are several approaches to integer ambiguity resolving when determining the attitude in an GPS/IMU manner. First it has been noted that the L1 GPS frequency of 1575.42 MHz corresponds to a wavelength, λ_{L1}, of 0,1903 [m] , approximately a factor of five to the baseline of 1 [m]. This indicates that integer search methods can get very computationally demanding, as it requires calculation throughout all the solutions [33].

First, the complete set of ambiguity parameters must be determined. The following equation generates a boundary of the ambiguity parameters for each baseline:

$$-\frac{\|\mathbf{b}\|}{\lambda_{L1}} < N_j < \frac{\|\mathbf{b}\|}{\lambda_{L1}} \tag{7.15}$$

For baselines of one meter, $N_j \in \pm 5$. For this boundary of three baselines measuring one meter each, the total number of solutions is $11^3 = 1331$ for each signal, as this boundary gives 11 possible solutions including $N_j = 0$. Here we will solve for the ambiguity parameters first and subsequently perform double differentiating. The double differentiated integer ambiguities will then be utilized.

Fast Integer Ambiguity Resolution for GPS Attitude Determination

It is possible to reduce the subset from cubic to quadratic using normality and geometric constraints. This approach is based on Lightsey, Markley, and Crassidis (1999) where it is assumed that either three noncoplanar sightlines or (preferably) three noncoplanar baselines are available. The latter is applied in Matlab.

First, the set is sequencially reduced to two baselines and two sightlines. The consequence to this approach is a significant reduction in the integer search space from $N^3 \in O(N^3)$ to $3N^2 \in O(N^2)$, a subset for which it can be geometrically shown that the following inequality must be true for baselines $\mathbf{b}_1$ and $\mathbf{b}_2$ [16]:

$$\|\mathbf{b}_1\|^2\|\mathbf{b}_2\|^2 > (\mathbf{b}_1\mathbf{b}_2)^2+\|\mathbf{b}_2\|^2(\triangle\varphi_{i1}-N_{i1})^2-2(\triangle\varphi_{i1}-N_{i1})(\triangle\varphi_{i2}-N_{i2})(\mathbf{b}_1\mathbf{b}_2)+\|\mathbf{b}_1\|^2(\triangle\varphi_{2j}-N_{2j}) \tag{7.16}$$

for which the same inequality can be applied using sightlines s_1 and s_2:

$$\|\mathbf{b}_1\|^2 > \|\mathbf{b}_1\|^2(s_1s_2)^2+(\triangle\varphi_{1j}-N_{1j})^2-2(\triangle\varphi_{1j}-N_{1j})(\triangle\varphi_{2j}-N_{2j})(\mathbf{s}_1\mathbf{s}_2)^2+(\triangle\varphi_{2j}-N_{2j}).$$

Both inequalities may be used, and as the baselines are better known, Equation 7.16 is applied in this thesis. In order to extract attitude information outside of the $\mathbf{b}_1, \mathbf{b}_2$ plane, the following condition:

$$\left[\left(\mathbf{R}_e^b s_i\right)\left(\mathbf{b}_1 \times \mathbf{b}_2\right)\right]^2 > 0 \tag{7.17}$$

must apply, down to which Eq. (7.16) has been reduced if the integers have been properly resolved. This condition indicates that $R_e^b s_i$, $\mathbf{b}_1$ and $\mathbf{b}_2$ must not lie in the same plane, which is almost always satisfied if the integers pass the test in Eq. (7.16) [16]. The parallelepiped is spanned by the vectors $\mathbf{R}_e^b s_i$, $\mathbf{b}_1$ and $\mathbf{b}_2$ in Eq. (7.17). This is illustrated in Figure 7.3; where the sign is positive, if and only if, the vectors form a right-handed system [16]. Since only two baselines are considered at a time, we use Equation 7.16 to significantly reduce the search space. We then search through all remaining candidates to directly determine the integers withouth pre-computing the sightline vector in the body frame. This is done using:

$$\mathbf{J}(n_i) = \frac{1}{2}\sum_{k=1}^{L}\left\{\frac{1}{\sigma_i^2(k)}\left[||\mathbf{B}_i^{-1}\Gamma_i(\Phi_i(k)-n_i)||^2 - ||s_i(k)||^2 + trace\left\{\mathbf{B}_i^{-1}\right\}\right]^2 + log\sigma_i^2(k)\right\} \tag{7.18}$$

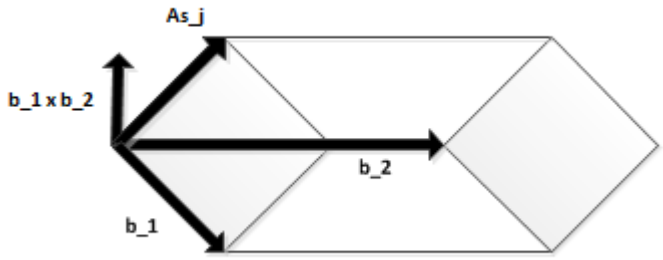

Figure 7.3: Parallelepiped Spanned by Three Vectors. The volume is zero if the three vectors are co-planar, meaning that by zero volume the non-coplanar condition is not satisfied.

where $n_i = \begin{bmatrix} n_{i1} \\ n_{i2} \\ n_{i3} \end{bmatrix}$, $\Phi_i = \begin{bmatrix} \triangle\phi_{i1} \\ \triangle\phi_{i2} \\ \triangle\phi_{i3} \end{bmatrix}$ and:

$$\sigma_i^2(k) = (\Phi_i(k) - n_i)^T \Gamma_i^T \mathbf{B}_i^{-3} \Gamma_i (\Phi_i(k) - n_i) - trace^2 \left\{ B_i^{-1} \right\} \tag{7.19}$$

Now, a unique solution which minimizes Equation 7.19 can now be determined with minimal vehicle motion since the solutions for the components of n_i are constrained to be integers. The loss function in Equation 7.18 involves a scalar check on the norm vector residuals. The remaining integers that have passed the inequality condition in Equation (7.16) are checked and the integer set which minimizes the loss function for each signal is chosen.

In order to secure that the ambiguity solution we have found is the correct one, an integrety check must be performed. This is done by calculating the estimate error covariance:

$$P_i = \left\{ \sum_{k=1}^{L} \frac{4}{\sigma_i^2(k)} \left[\underline{\Phi}_i(k) - \underline{n}_i \right] \left[\underline{\Phi}_i(k) - \underline{n}_i \right]^T \right\}^{-1} \tag{7.20}$$

The ambiguity solution for each signal is accepted when the square root of every diagonal element is real and less than some threshold, ζ:

$$\sqrt{P_{kk}}|_{k=1,2,3} < \zeta \tag{7.21}$$

The filter specific design parameter is chosen $\zeta = 0.1$. This check makes sure that any false ambiguity solutions are disregarded.

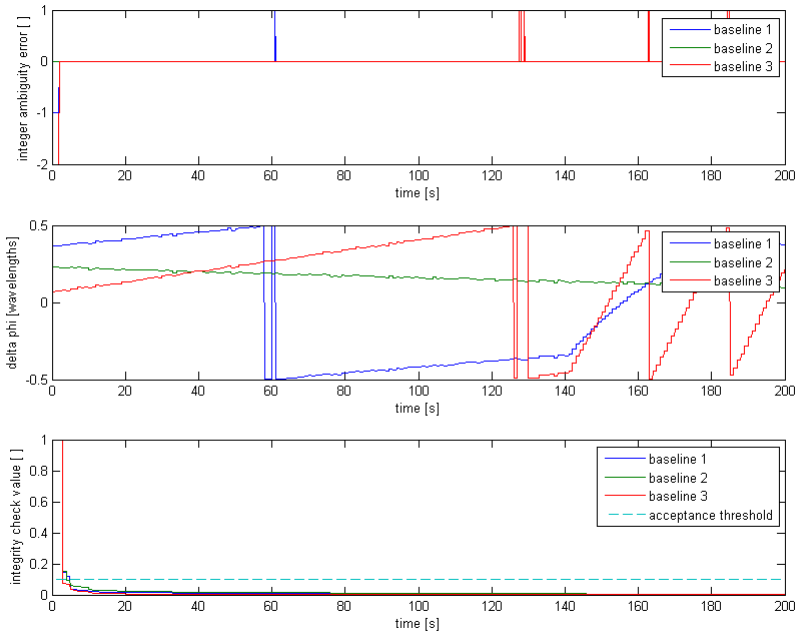

Figure 7.4: This figure shows the integer ambiguity error, delta phi and the integrity check value, respectively, simulated for a GPS signal which arbitrarily chosen out of the GPS SVs visible to the test satellite. The GPS attitude data (delta phi) are updated every 5 seconds. It was necessary to include the integrity check in order to reject false solutions in the fast integer ambiguity resolution method.

7.1.5 Time Measurements

There are two main categories of time measurements to be aware of when estimating the attitude of a spacecraft. It is for once necessary to measure the time intervals between events, such as the length of time a sensor sees the Sun. Second, there are absolute times of specific events such as at which calendar time did some particular spacecraft sensing occur [35]. The uniformly adopted solution to the fact that time zones are different throughout the world is to use the local time corresponding to 0 degrees longitude, such as Coordinated Universal Time (UTC) or Greenwich Mean Time (GMT), as the standard absolute time for events anywhere in the world or in space.

Chapter 8

Kalman Filter Design

The Kalman Filter is a recursive algorithm which produces statistically optimal estimates of unknown variables based on a series of noisy measurements observed over time, under the assumption of Gaussian white noise on the measurements, and that a linear observable system is available. These estimates are normally more precise than estimates based on single measurements, and by using a Kalman filter we can obtain a heuristic approximation of the optimal estimate of the spacecraft attitude.

However, the strapdown inertial navigation equations are nonlinear. It is therefore desirable to utilize a nonlinear version of the Kalman filter, known as the Extended Kalman Filter. The EKF solves nonlinear problems by linearizing the nonlinear system about its currently best estimate and, based on past measurements and the available dynamic model of the spacecraft, it **c**an be used to produce position, velocity, and attitude estimates. The EKF has become a widely used filter in control systems in general, and for his work on co-invention and development of the Kalman filter, U.S. President Barack Obama awarded Rudolf E. Kálmán with the National Medal of Science on October 7, 2009.

Stability and convergence of the Extended Kalman Filter was proved by Jouffroy and Fossen (2010), under the assumption of a lower and upper bounded covariance matrix.

8.1 Multiplicative Extended Kalman Filter (MEKF)

As outlined in this thesis, an attitude aiding sensor approach is implemented, and a dynamic model of an inertial measurement unit is incorporated in the Kalman filter to combine the GPS, gyro, and magnetometer attitude information. Because the rigid body dynamics of a spacecraft is of predictable character, this deterministic information about the vehicle dynamics is used to smooth

the GPS attitude solution during degraded transmitter visibility or GPS-signal drop-out.

The Multiplicative EKF is a version of the EKF which has been used on board several NASA spacecraft for attitude estimation, for the first time in the Space Precision Attitude Reference System (SPARS) in 1969 [6], and has been discussed in details in References [15] and [17]. The MEKF solves the singluar covariance matrix problem that arises due to the constraints on the quaternions (see Subchapter 3.2.2) by reducing the dimension of the covariance matrix. A dilemma which is faced by the Kalman filter is namely that it uses an attitude representation that is either singular (by using a minimal three-parameter representation such as Euler angles and Rodrigues parameters) or redundant (e.g. using the four-dimensional quaternion). One of the ways to implement an extended Kalman filter and a preferred strategy to avoid this dilemma of representation singularities is using a correctly normalized four-component globally non-singular representation for a reference attitude and a three-parameter set, $\delta\boldsymbol{\epsilon}$, to represent the deviations from this reference. It is computed as an unconstrained estimate of the three-component [19]. The attitude is here represented as the quaternion product:

$$\mathbf{q} = \mathbf{q}_{ref} \otimes \delta\mathbf{q}(\delta\boldsymbol{\epsilon}), \tag{8.1}$$

where $\mathbf{q}_{ref}$ is some unit reference quaternion, a correctly normalized four-component vector which provides a globally non-singular attitude representation. The unit quaternion $\delta\mathbf{q}(\delta\boldsymbol{\epsilon})$ represents the rotation from $\mathbf{q}_{ref}$ to the true attitude $\mathbf{q}$, and the rotation is parameterised by $\delta\boldsymbol{\epsilon}$, a three-parameter vector representing the deviations from the reference. Furthermore, if $\delta\boldsymbol{\epsilon}$ can be estimated as $\delta\hat{\boldsymbol{\epsilon}} = E\{\delta\boldsymbol{\epsilon}\}$ of $\delta\boldsymbol{\epsilon}$, it results from Eq. 8.1 that:

$$\mathbf{q}_{ref} \otimes \delta\mathbf{q}(\delta\hat{\boldsymbol{\epsilon}}) = \hat{\mathbf{q}}, \tag{8.2}$$

the corresponding estimate of the true attitude quaternion $\mathbf{q}$. For the MEKF this estimate of the true quaternion is the same as $\mathbf{q}_{ref}$, the reference quaternion (the reference quaternion is the best estimate of the true quaternion). This technique chooses the reference quaternion $\mathbf{q}_{ref}$ such that $\delta\hat{\boldsymbol{\epsilon}}$ is identically to zero, $\delta\mathbf{q}(\mathbf{0})$ being the identity quaternion. By choosing $\delta\boldsymbol{\epsilon}$ indentically zero, any singluarity or discontinuity of the three-component representation of $\delta\boldsymbol{\epsilon}$ is avoided and the redundancy in the attitude representation is removed:

$$\hat{\mathbf{q}} = \mathbf{q}_{ref} \tag{8.3}$$

resulting in the product

$$\mathbf{q} = \hat{\mathbf{q}} \otimes \delta\mathbf{q}(\delta\boldsymbol{\epsilon})$$

where $\hat{\mathbf{q}}$ is the unit estimated quaternion, $\delta\boldsymbol{\epsilon}^{-1} = -\delta\boldsymbol{\epsilon}$, and:

$$\delta\mathbf{q}(\delta\boldsymbol{\epsilon}) = \begin{bmatrix} \sqrt{1 - \delta\boldsymbol{\epsilon}^T\delta\boldsymbol{\epsilon}} \\ \delta\boldsymbol{\epsilon} \end{bmatrix} \tag{8.4}$$

Design matrices	$\mathbf{Q}_k = E[\mathbf{w}_k \mathbf{w}_k^T] = cov(\mathbf{w}_k) \approx \sigma_{w_k}^2$ $\mathbf{R}_k = E[\mathbf{v}_k \mathbf{v}_k^T] = cov(\mathbf{v}_k) \approx \sigma_{v_k}^2 \mathbf{I}$
Initial conditions	$\bar{\mathbf{x}}_0 = \mathbf{x}_0$ $\bar{\mathbf{P}}_0 = \mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T]$ $\hat{b} = \mathbf{q}_0$
Kalman gain matrix State estimate update Error covariance update	$\mathbf{H}_k = \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k}\|_{\mathbf{x}_k = \bar{\mathbf{x}}_k}$ $\mathbf{K}_k = \bar{\mathbf{P}}_k \mathbf{H}_k^T [\mathbf{H}_k \bar{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k]^{-1}$ $\hat{\mathbf{x}}_k = \bar{\mathbf{x}}_k + \mathbf{K}_k [\mathbf{y} - \mathbf{h}(\bar{\mathbf{x}}_k)]$ $\hat{\mathbf{P}}_k = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \bar{\mathbf{P}}_k [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k]^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T$
Move error	$\hat{\mathbf{q}}_k = \hat{\mathbf{q}}_{k-1} \otimes \delta \mathbf{q}(\delta \hat{\boldsymbol{\epsilon}}_k)$ $\delta \hat{\boldsymbol{\epsilon}}_k = 0$
State estimation propagation Error covariance propagation	$\Phi_k = \frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_k}\|_{\mathbf{x}_k = \hat{\mathbf{x}}_k}$ $\bar{\mathbf{x}}_{k+1} = \mathbf{f}_k(\hat{\mathbf{x}}_k, \mathbf{u}_k)$ $\bar{\mathbf{P}}_{k+1} = \Phi_k \hat{\mathbf{P}}_k \Phi_k^T + \Gamma_k \mathbf{Q}_k \Gamma_k^T$

Table 8.1: Discrete Multiplicative Extended Kalman Filter

where we here choose to denote $\sqrt{1 - \delta\boldsymbol{\epsilon}^T \delta\boldsymbol{\epsilon}} = \delta\eta$, giving the resulting attitude error representation:

$$\delta\mathbf{q}(\delta\boldsymbol{\epsilon}) = \begin{bmatrix} \delta\eta \\ \delta\boldsymbol{\epsilon} \end{bmatrix} \tag{8.5}$$

This provides a consistent treatment of the attitude error statistics, with the covariance of the attitude error angles in the body frame (in radians squared) represented by the covariance of $\delta\boldsymbol{\epsilon}$ [18]. The fundamental conceptual advantage of the MEKF is that $q_{ref} = \hat{q}$ is a unit quaternion by definition, and therefore the three-vector $\delta\epsilon$ will never approach a singularity [19]. Note that the reference quternion in the MEKF is not considered a random variable, and that its estimate is not an expectation.

Theory and evaluation of the multiplicative EKF is found in various sources, e.g. [6] and [16] - [19].

8.1.1 States of the Multiplicative EKF

The model of the MEKF is written:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \delta\dot{\epsilon}^T & \dot{\mathbf{b}}_g^T & \dot{a} & \dot{M} & \dot{i} & \dot{\Omega} \end{bmatrix}^T \tag{8.6}$$

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k) + \boldsymbol{\Gamma} w_k \tag{8.7}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \tag{8.8}$$

where $\mathbf{v}$ and $\mathbf{w}$ are Gaussian white noise. Furthermore, we choose the state variables:

$$\mathbf{x} = \begin{bmatrix} \delta\boldsymbol{\omega}^T & \mathbf{b}_g^T & a & M & i & \Omega \end{bmatrix}^T \tag{8.9}$$

and the measurements:

$$\mathbf{y} = \begin{bmatrix} (\mathbf{m}^b)^T & \triangle r^T & (r^i)^T & (\mathbf{v}^i)^T \end{bmatrix}^T \tag{8.10}$$

8.1.2 Attitude Model

From Equation (8.1) we find the atttude representative [20]:

$$\delta\mathbf{q} = \hat{\mathbf{q}}^{-1} \otimes \mathbf{q} \tag{8.11}$$

Differentiating Equation (8.11) with respect to time yields

$$\begin{aligned} \delta\dot{\mathbf{q}} = \begin{bmatrix} \delta\dot{\eta} \\ \delta\dot{\boldsymbol{\epsilon}} \end{bmatrix} &= \hat{\mathbf{q}}^{-1} \otimes \dot{\mathbf{q}} \\ &= \frac{1}{2}\hat{\mathbf{q}}^{-1} \otimes \mathbf{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{b/i}^b \end{bmatrix} \\ &= \frac{1}{2}\delta\mathbf{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{b/i}^b \end{bmatrix} \\ &= \frac{1}{2}\Omega(\boldsymbol{\omega}_{b/o}^b)\delta\mathbf{q} \\ &= \frac{1}{2}\begin{bmatrix} 0 & -(\boldsymbol{\omega}_{b/i}^b)^T \\ \boldsymbol{\omega}_{b/o}^b & -\mathbf{S}(\boldsymbol{\omega}_{b/i}^b) \end{bmatrix}\begin{bmatrix} \delta\eta \\ \delta\boldsymbol{\epsilon} \end{bmatrix} \end{aligned} \tag{8.12}$$

This gives the following vector part:

$$\begin{aligned} \delta\dot{\boldsymbol{\epsilon}} &= \frac{1}{2}\left[\boldsymbol{\omega}_{b/i}^b\delta\eta - \mathbf{S}(\boldsymbol{\omega}_{b/i}^b)\delta\boldsymbol{\epsilon}\right] \\ &= \frac{1}{2}\left[\mathbf{I}_{3\times3}\delta\eta + \mathbf{S}(\delta\boldsymbol{\epsilon})\right]\boldsymbol{\omega}_{b/i}^b \end{aligned} \tag{8.13}$$

And the gyro bias derivative:

$$\dot{\mathbf{b}}_g = \mathbf{0}_{3\times1} \tag{8.14}$$

Remark: The rotation has been implemented in Matlab representing the attitude from body to ECI. Subsequently multiplying q with q_{orbit} will give the attitude represented from body to orbit, where q_{orbit} is the rotation from the ECI frame to the orbit frame.

8.1.3 Orbit Model

The orbit states $\mathbf{x}_{orbit}$ subset is:

$$\dot{\mathbf{x}}_{orbit} = \begin{bmatrix} 0 & n_{orbit} & 0 & 0 \end{bmatrix}^T, \tag{8.15}$$

where n_{orbit} is the orbit mean motion:

$$n_{orbit} = \sqrt{\frac{GM}{a^3}} \tag{8.16}$$

8.1.4 Measurement Equation

Assuming that the ambiguities are already resolved and recalling that only L_1 is used, the measurement vector, Eq. (8.8) now becomes:

$$\triangle r_{ij} = \begin{bmatrix} \triangle r_{11} \\ \vdots \\ \triangle r_{1j} \\ \vdots \\ \triangle r_{ij} \end{bmatrix} \tag{8.17}$$

$$= \begin{bmatrix} \triangle r_{11} & ... & \triangle r_{ij} \end{bmatrix}^T \tag{8.18}$$

for i satellite observations. Furthermore, we get:

$$\mathbf{h}_k(\mathbf{x}) = \begin{bmatrix} \mathbf{m}^b \\ \triangle r_{k11} \\ \vdots \\ \triangle r_{k1j} \\ \vdots \\ \triangle r_{kij} \\ \mathbf{r}^e \\ \mathbf{v}^e \end{bmatrix} \tag{8.19}$$

where:

$\mathbf{m}^e$ is the magnetometer measurement in ECEF coordinates:

$$\mathbf{m}^b = \mathbf{R}^b_{\hat{b}}(\delta\boldsymbol{\epsilon}^{-1})\mathbf{R}^{\hat{b}}_i(\hat{\mathbf{q}}^{-1})\mathbf{R}^i_e\mathbf{m}^e \tag{8.20}$$

$$= (\mathbf{I}_{3\times 3} + 2\mathbf{S}(\delta\boldsymbol{\epsilon}^{-1}))\mathbf{R}^{\hat{b}}_i(\hat{\mathbf{q}}^{-1})\mathbf{R}^i_e\mathbf{m}^e \tag{8.21}$$

r is the position vector of the NUTS spacecraft,

v is the velocity of the spacecraft and:

$$\triangle r_{kij} = \mathbf{l}_i^T \mathbf{R}_b^e(\mathbf{q}_k)\mathbf{b}_j^b \tag{8.22}$$

Now the linearized observation matrix $\mathbf{H}_k$ becomes:

$$\mathbf{H}_k = \begin{bmatrix} \frac{\partial(\mathbf{m}^b)}{\partial \mathbf{x}^T} \\ \frac{\partial(\triangle r_{k11})}{\partial \mathbf{x}^T} \\ \vdots \\ \frac{\partial(\triangle r_{ki1})}{\partial \mathbf{x}^T} \\ \vdots \\ \frac{\partial(\triangle r_{kij})}{\partial \mathbf{x}^T} \\ \frac{\partial(\mathbf{r}_k^e)}{\partial \mathbf{x}^T} \\ \frac{\partial(\mathbf{v}_k^e)}{\partial \mathbf{x}^T} \end{bmatrix} \tag{8.23}$$

The lever arm is 0,125 m., which does not exceed the precision of the GPS measurement. The position and velocity contributions of the lever arm may therefore be neglected, and the linearized orbital parameter measurement sensitivity matrices become:

$$\frac{\partial(\mathbf{r}^e)}{\partial \mathbf{x}^T} = R_i^e \frac{\partial(\mathbf{r}^i)}{\partial \mathbf{x}^T} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{\partial(\mathbf{r}^i)}{\partial a} & \frac{\partial(\mathbf{r}^i)}{\partial M} & \frac{\partial(\mathbf{r}^i)}{\partial i} & \frac{\partial(\mathbf{r}^i)}{\partial \Omega} \end{bmatrix} \tag{8.24}$$

$$\frac{\partial(\mathbf{v}^e)}{\partial \mathbf{x}^T} = R_i^e \frac{\partial(\mathbf{v}^i)}{\partial \mathbf{x}^T} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \frac{\partial(\mathbf{v}^i)}{\partial a} & \frac{\partial(\mathbf{v}^i)}{\partial M} & \frac{\partial(\mathbf{v}^i)}{\partial i} & \frac{\partial(\mathbf{v}^i)}{\partial \Omega} \end{bmatrix} \tag{8.25}$$

$$\frac{\partial(\mathbf{r}^i)}{\partial a} = \begin{bmatrix} cos(\Omega)cos(M) - sin(\Omega)sin(M)cos(i) \\ sin(\Omega)cos(M) + cos(\Omega)sin(M)cos(i) \\ sin(M)sin(i) \end{bmatrix} \tag{8.26}$$

$$\frac{\partial(\mathbf{v}^i)}{\partial a} = \begin{bmatrix} \frac{\sqrt{\frac{GM}{a}}}{2a} cos(\Omega)sin(M) + sin(\Omega)cos(M)cos(i) \\ \frac{\sqrt{\frac{GM}{a}}}{2a} sin(\Omega)sin(M) - cos(\Omega)cos(M)cos(i) \\ -\frac{\sqrt{\frac{GM}{a}}}{2a} cos(M)sin(i) \end{bmatrix} \tag{8.27}$$

$$\frac{\partial(\mathbf{r}^i)}{\partial M} = \begin{bmatrix} -a(cos(\Omega)sin(M) + sin(\Omega)cos(M)cos(i)) \\ -a(sin(\Omega)sin(M) - cos(\Omega)cos(M)cos(i)) \\ a(cos(M)sin(i)) \end{bmatrix} \tag{8.28}$$

$$\frac{\partial(\mathbf{v}^i)}{\partial M} = \begin{bmatrix} -\sqrt{\frac{GM}{a}}(cos(\Omega)cos(M) - sin(\Omega)sin(M)cos(i)) \\ -\sqrt{\frac{GM}{a}}(sin(\Omega)cos(M) + cos(\Omega)sin(M)cos(i)) \\ -\sqrt{\frac{GM}{a}}(sin(M)sin(i)) \end{bmatrix} \tag{8.29}$$

$$\frac{\partial(\mathbf{r}^i)}{\partial i} = \begin{bmatrix} a(sin(\Omega)sin(M)sin(i)) \\ -a(cos(\Omega)sin(M)sin(i)) \\ a(sin(M)cos(i)) \end{bmatrix} \tag{8.30}$$

$$\frac{\partial(\mathbf{v}^i)}{\partial i} = \begin{bmatrix} \sqrt{\frac{GM}{a}}(sin(\Omega)cos(M)sin(i)) \\ -\sqrt{\frac{GM}{a}}(cos(\Omega)cos(M)sin(i)) \\ \sqrt{\frac{GM}{a}}(cos(M)cos(i)) \end{bmatrix} \tag{8.31}$$

$$\frac{\partial(\mathbf{r}^i)}{\partial \Omega} = \begin{bmatrix} -a(sin(\Omega)cos(M) + cos(\Omega)sin(M)cos(i)) \\ a(cos(\Omega)cos(M) - sin(\Omega)sin(M)cos(i)) \\ 0 \end{bmatrix} \tag{8.32}$$

$$\frac{\partial(\mathbf{v}^i)}{\partial \Omega} = \begin{bmatrix} \sqrt{\frac{GM}{a}}(sin(\Omega)sin(M) - cos(\Omega)cos(M)cos(i)) \\ -\sqrt{\frac{GM}{a}}(cos(\Omega)sin(M) + sin(\Omega)cos(M)cos(i)) \\ 0 \end{bmatrix} \tag{8.33}$$

$\frac{\partial(\triangle r_{kij})}{\partial \mathbf{x}^T}$ can be found from the relationship:

$$\frac{\partial(\triangle r_{kij})}{\partial \mathbf{x}^T} = \begin{bmatrix} \frac{\partial(\triangle r_{kij})}{\partial(\delta \boldsymbol{\epsilon}^T)} & \mathbf{0}_{3\times 9} \end{bmatrix} \tag{8.34}$$

$$\begin{aligned} \triangle r_{k,ij} &= \mathbf{l}_{k,i}^T \mathbf{R}_i^e \mathbf{R}_b^i(\mathbf{q}_k)\mathbf{b}_j^b \\ &= \mathbf{l}_{k,i}^T \mathbf{R}_i^e \mathbf{R}_{\hat{b}}^i(\hat{\mathbf{q}}_k)\mathbf{R}_b^{\hat{b}}(\delta\boldsymbol{\epsilon}_k)\mathbf{b}_j^b \\ &= \mathbf{l}_{k,i}^T \mathbf{R}_i^e \mathbf{R}_{\hat{b}}^i(\hat{\mathbf{q}}_k)(\mathbf{I}_{3\times 3} + 2\mathbf{S}(\delta\boldsymbol{\epsilon}_k))\mathbf{b}_j^b \end{aligned} \tag{8.35}$$

$$\begin{aligned} \frac{\partial \triangle r_{kij}}{\partial(\delta\boldsymbol{\epsilon}^T)} &= -2\frac{\partial(\mathbf{l}_{ki}^T \mathbf{R}_i^e \mathbf{R}_e^i(\hat{\mathbf{q}}_k)\mathbf{S}(\mathbf{b}^b)\delta\boldsymbol{\epsilon}_k)}{\partial(\delta\epsilon^T)} \\ &= -2\mathbf{l}_{ki}^T \mathbf{R}_i^e \mathbf{R}_b^i(\hat{\mathbf{q}}_k)\mathbf{S}(\mathbf{b}^b) \end{aligned} \tag{8.36}$$

The linearized magnetometer observation matrix becomes:

$$\frac{\partial(\mathbf{m}^b)}{\partial(\mathbf{x}^T)} = \begin{bmatrix} \frac{\partial(\mathbf{m}^b)}{\partial(\delta\boldsymbol{\epsilon}^T)} & \mathbf{0}_{3\times 9} \end{bmatrix} \tag{8.37}$$

where:

$$\frac{\partial(\mathbf{m}^b)}{\partial(\delta\boldsymbol{\epsilon}^T)} = 2\mathbf{S}(\mathbf{R}_i^{\hat{b}}(\hat{\mathbf{q}}^{-1})\mathbf{R}_e^i\mathbf{m}^e) \tag{8.38}$$

8.1.5 Linearized Propagation Matrix

The linearized propagation matrix becomes:

$$\Phi = I_{10\times 10} + h\frac{\partial \mathbf{f}}{\partial \mathbf{x}^T} = \begin{bmatrix} \frac{\partial(\delta\dot{\boldsymbol{\epsilon}})}{\partial(\delta\boldsymbol{\epsilon})} & \frac{\partial(\delta\dot{\boldsymbol{\epsilon}})}{\partial(\delta\mathbf{b}_g)} & \mathbf{0}_{3\times 4} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 4} \\ \mathbf{0}_{4\times 3} & \mathbf{0}_{4\times 3} & \frac{\partial \dot{\mathbf{x}}_{orbit}}{\partial \mathbf{x}_{orbit}} \end{bmatrix} \tag{8.39}$$

where:

$$\frac{\partial(\delta\dot{\epsilon})}{\partial(\delta\epsilon)} = -\frac{1}{2}(\mathbf{S}(\omega^b_{imu} - \hat{\mathbf{b}}_g)) \tag{8.40}$$

$$\frac{\partial(\delta\dot{\epsilon})}{\partial(\delta\mathbf{b}_g)} = -\frac{1}{2}\mathbf{I}_{3\times 3} \tag{8.41}$$

$$\frac{\partial \dot{\mathbf{x}}_{orbit}}{\partial \mathbf{x}_{orbit}} = \begin{bmatrix} 0 & & \\ -\frac{3}{2}\sqrt{\frac{GM}{a^5}} & & \mathbf{0}_{4\times 3} \\ 0 & & \\ 0 & & \end{bmatrix}. \tag{8.42}$$

Chapter 9

Simulator

The equations of motion and kinematics of the satellite are implemented in Simulink, as are the orbit dynamics. This generates position, velocity, and attitude data, which are used to simulate sensor measurements.

9.1 Simulation of the Spacecraft in Low Earth Orbit

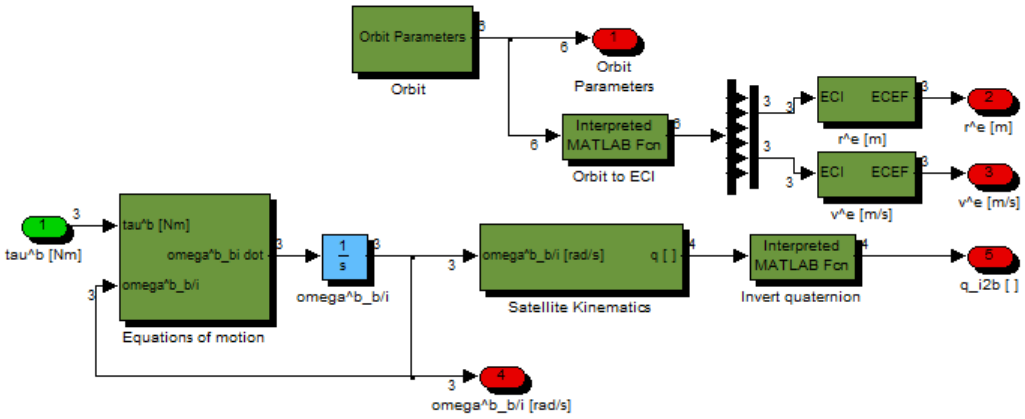

Figure 9.1: Satellite subsystem

9.1.1 Attitude

The satellite simulation subsystem shown in Figure 9.1 accepts a vector of torques, calculates and integrates the angular accelerations to give the angu-

lar velocity. The angular velocity is then used in the quaternion differential equation, which is subsequently integrated to give the attitude quaternion.

9.1.2 Orbit

The chosen Keplerian orbital elements with disturbances are propagated and then transformed into velocity and position data, using Equations (4.3 - 4.8).

9.2 Simulated Instrumentation

9.2.1 Gyro

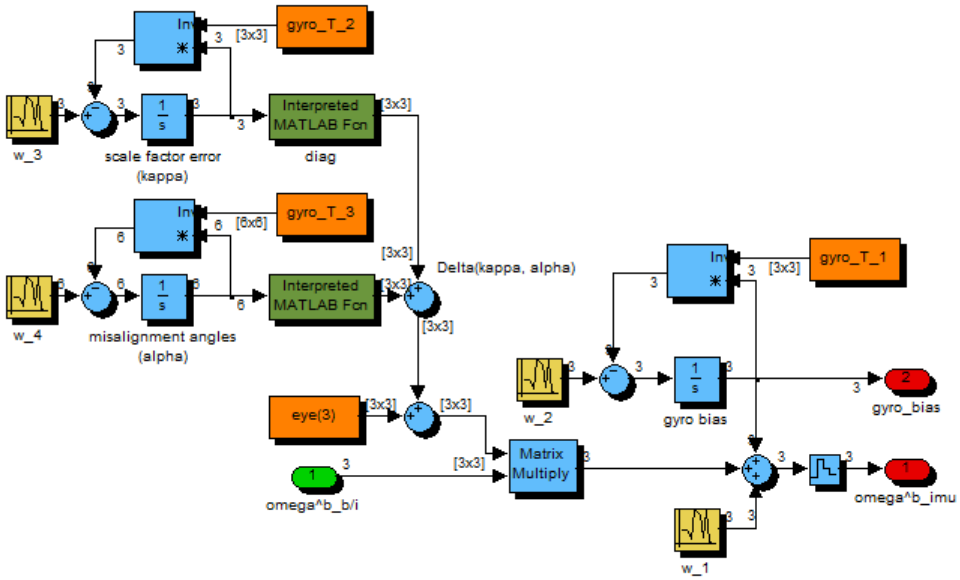

Figure 9.2: Gyro subsystem

The gyro subsystem accepts the actual angular velocity of the spacecraft, and adds error terms due to gyro bias, scale factor, misalignment angles, and measurement noise. These terms are modelled according to Equation (5.2).

9.2.2 Magnetometer

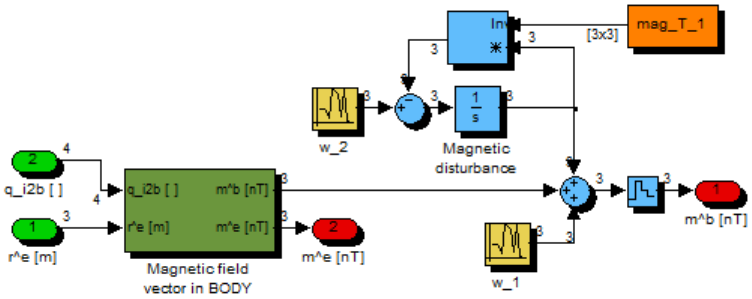

Figure 9.3: Magnetometer subsystem

The magnetometer subsystem simulates the local magnetic field vector measurement at the current position of the spacecraft. It accepts the actual position and attitude of the spacecraft, from which it calculates the local magnetic field vector using the IGRF model, and rotates it to the body frame. It then adds a local magnetic disturbance and measurement noise using Equation (5.13).

9.2.3 GPS Position and Velocity

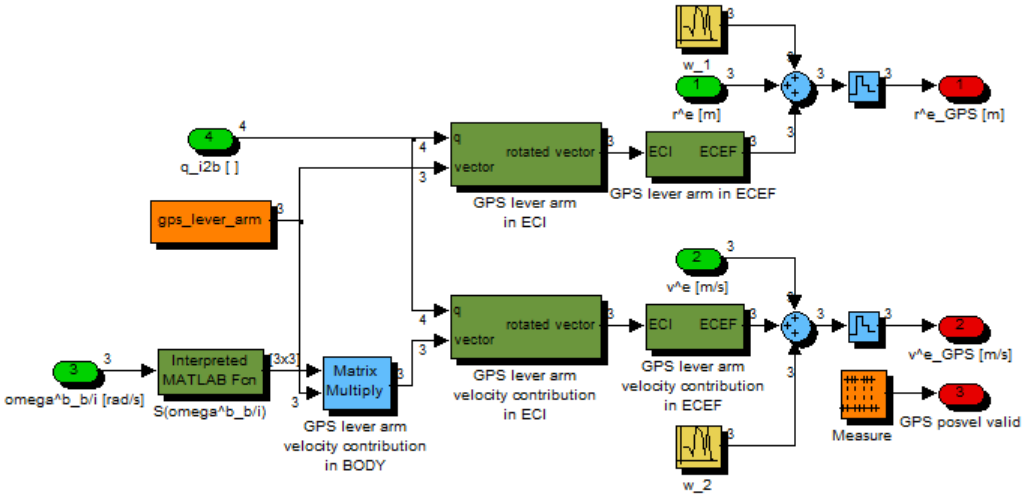

Figure 9.4: GPS position and velocity subsystem

The GPS subsystem takes in actual position and velocity. It then adds measurement noise, and the contribution from the GPS lever arm. Also generated is a signal that is true when a new measurement is made, and false otherwise.

9.2.4 GPS Attitude

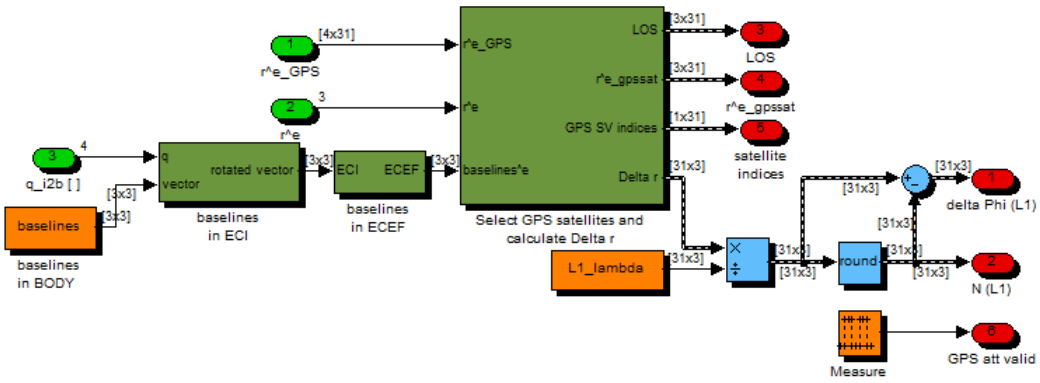

Figure 9.5: GPS attitude subsystem

The GPS attitude subsystem must simulate a number of measurements. Most important are the phase difference measurements, $\triangle\phi$, but the subsystem must also provide the positions and unique identifiers of the GPS Space Vehicles which are used to obtain the respective $\triangle\phi$ measurements. To do this, the orbits of the GPS SVs are retrieved from a GPS YUMA almanac. The GPS SVs orbits are then simulated in the same manner as in the satellite subsystem to provide the positions of the GPS SVs. Then the visibility of each GPS SV with respect to the test satellite is evaluated, and only the position of the visible GPS SVs are used to simulate $\triangle\phi$ measurements.

The $\triangle\phi$ measurements are now simulated by rotating the matrix of baseline vectors to the ECEF frame by using the actual attitude, and then multiplying the transposed matrix of line-of-sight vectors with the matrix of baseline vectors in the ECEF frame, to obtain the delta range matrix, $\triangle\mathbf{r}$. $\triangle\mathbf{r}$ is then divided by the wavelength of the GPS L1 signal, giving the range difference in wavelengths. The fractional part, $\triangle\phi$, is then found by subtracting the closest integer from the range difference.

Also generated is a signal that is true when a new measurement is made, and false otherwise.

Part III

Results

Chapter 10

Simulation Results

The filter was first simulated without any prior information, i.e. no initial values. The filter is then required to estimate the initial values based on the first set of measurements. The integer ambiguities must be resolved before the attitude solution can be obtained, a process which takes several samples. The filter therefore has to wait until the ambiguities are found, upon which it converges to the actual attitude.

The filter was then simulated given initial attitude values close to the actual attitude. Resolution of integer ambiguities must still be done, during which time the attitude estimates drift off due to the unknown gyro bias. This approach gave slightly better results.

Then the filter was tuned, and subsequently simulated under the conditions of no prior information. The new tuning parameters gave better results concerning initial attitude estimates and convergence of the filter.

The filter was also simulated with GPS outage, through which the attitude estimates stayed close to the actual values. The filter was robust to the attitude maneuvers which were performed.

Lastly, orbit simulations were performed for a perfectly circular orbit, as well as for a slightly elliptic orbit, $e = 0.0001$. Both simulations were performed with and without GPS outage.

Attitude maneuvers were performed for all simulations. The observations are described in the figures in this Chapter.

10.1 Attitude

No Initial Values

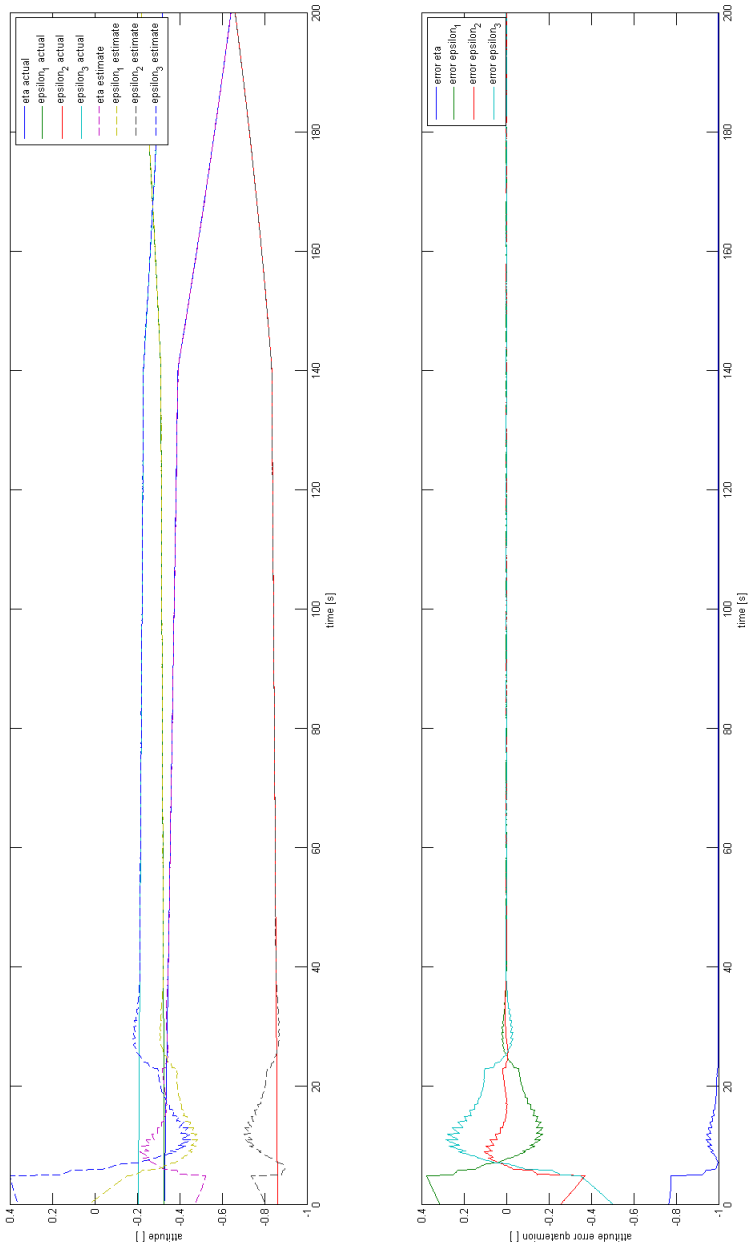

Figure 10.1: Here we see the quaternion attitude estimates with noise and no initial values. The filter converges to the correct attitude with some overshoot. After 140 seconds an attitude maneuver is simulated, applying torque of 0.01 Nm about the body y-axis for one second.

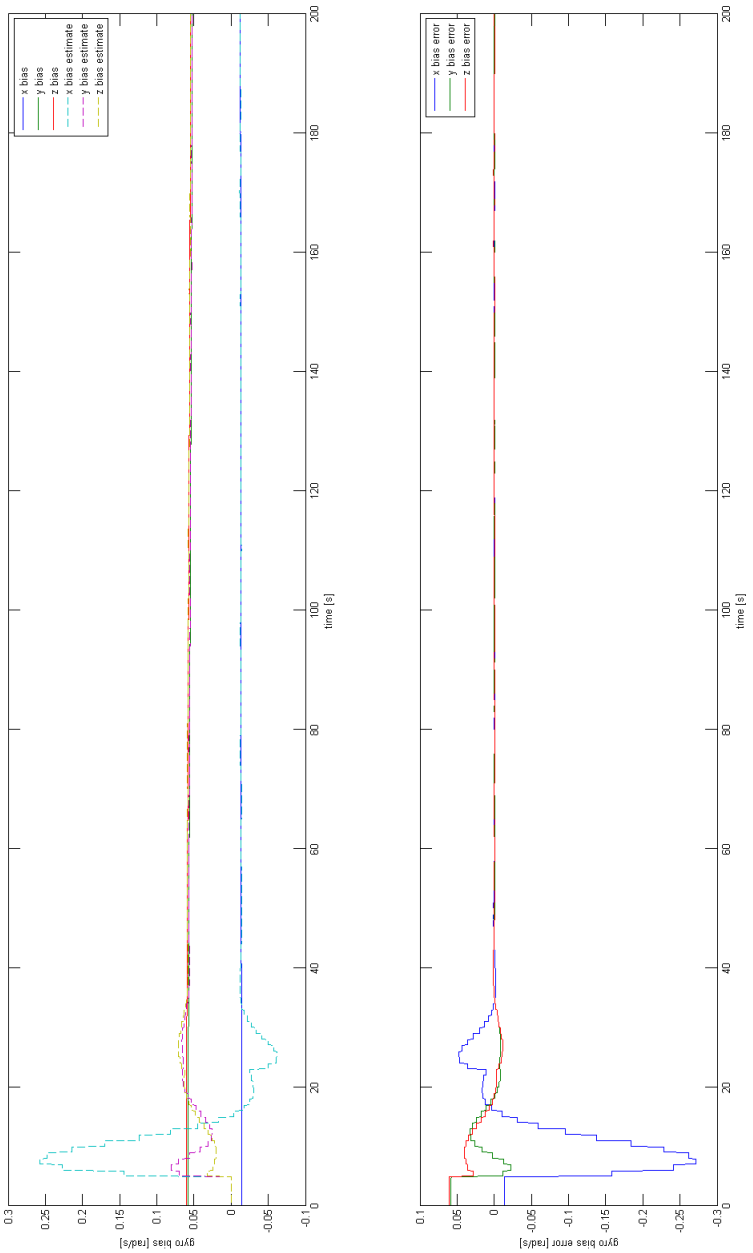

Figure 10.2: We see that the large error in the estimated initial values causes wind-up in the gyro bias during the first 40 seconds. This wind-up causes the attitude overshoot seen in Figure 10.1.

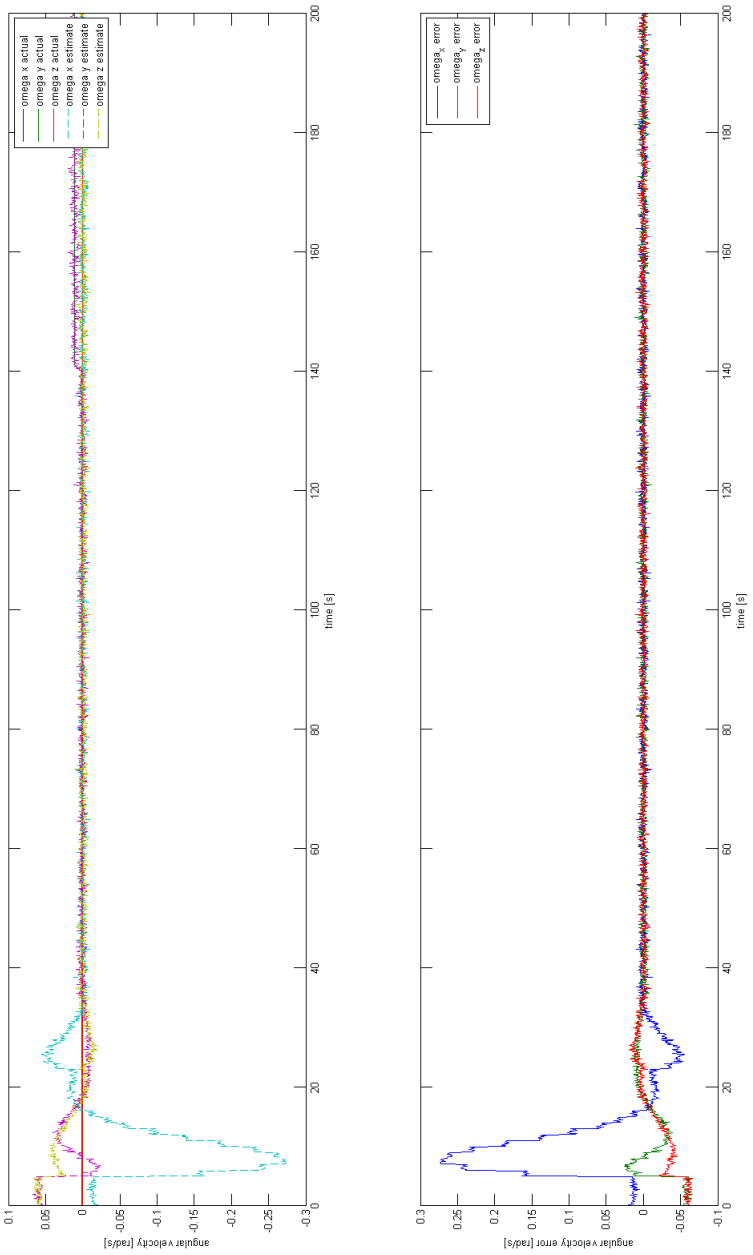

Figure 10.3: The angular velocity estimate converges to the actual angular velocity when the wind-up in gyro bias dissipates. We can also see how the attitude maneuver after 140 seconds gives a rise in angular velocity and no rise in the angular velocity error. The filter is able to follow the maneuver.

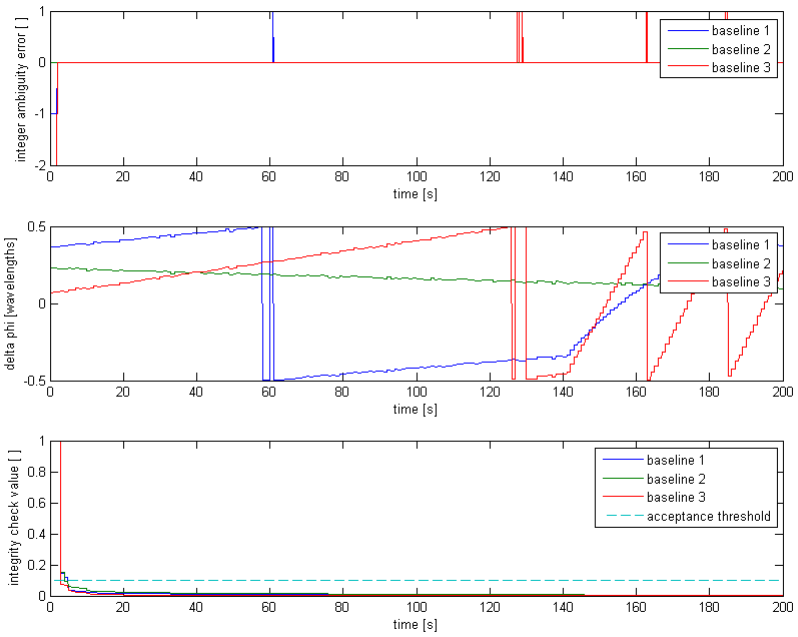

Figure 10.4: The integer ambiguity resolution for a single GPS signal. The correct integer ambiguity solution is found and kept locked through integer changes due to delta phi crossing 0.5 or -0.5. The solution for a signal is not accepted until the integrity check value for all baselines is below the acceptance threshold ζ , which was set to 0.1.

Good Initial Values

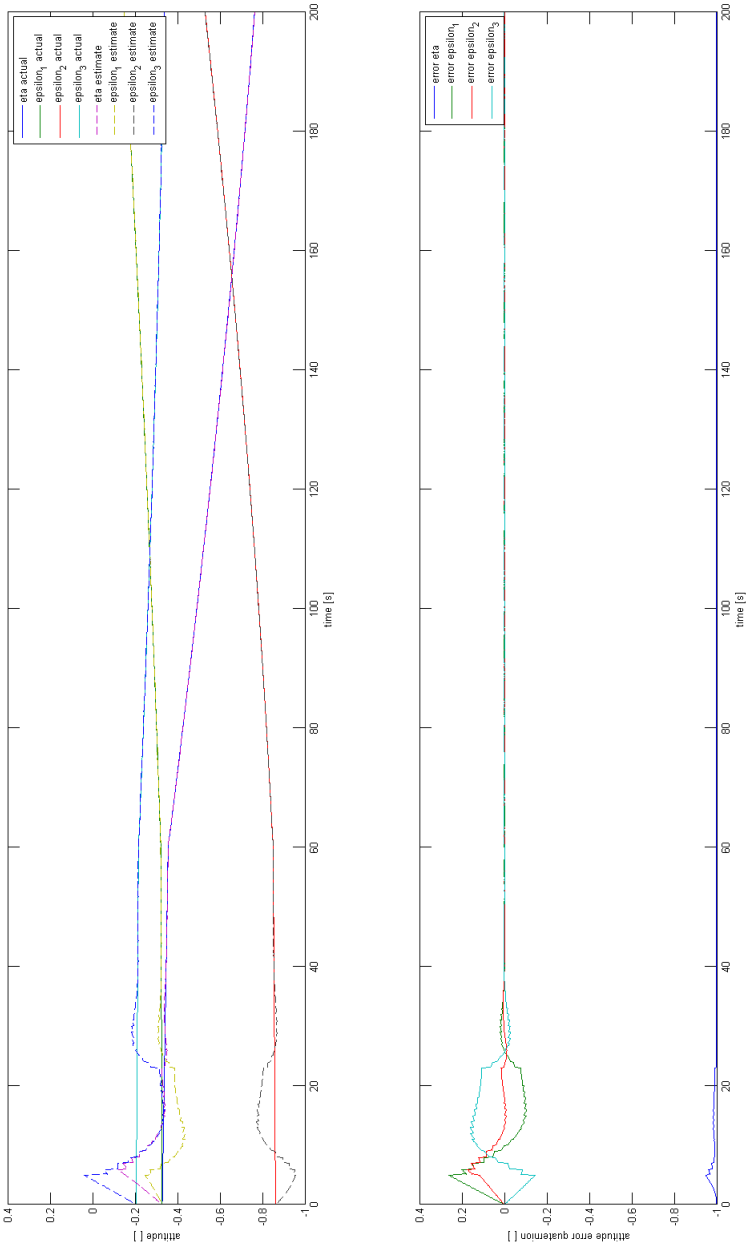

Figure 10.5: Until the ambiguities are resolved, the filter is reliant upon the gyro measurement and the magnetometer. The magnetometer measurement gives one vector, about which axis the attitude estimate can still rotate freely and must therefore rely on gyro measurement. Gyro bias makes the estimates drift away from the correct attitude. When the ambiguities are resolved, the attitude estimate converges to the correct attitude. After 60 seconds we can observe the effect of the applied torque of 0.01 Nm.

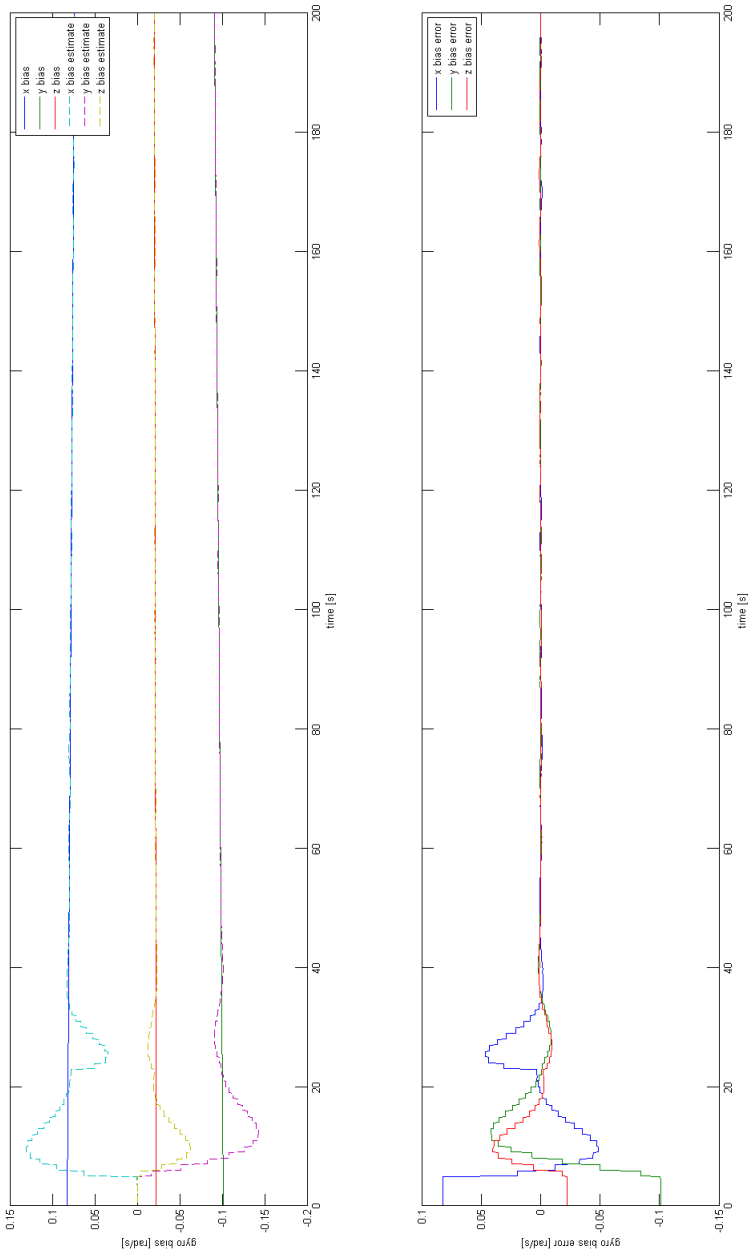

Figure 10.6: We can still see some wind-up but once the integer ambiguities are resolved and the correct attitude is found, the gyro bias estimate converges.

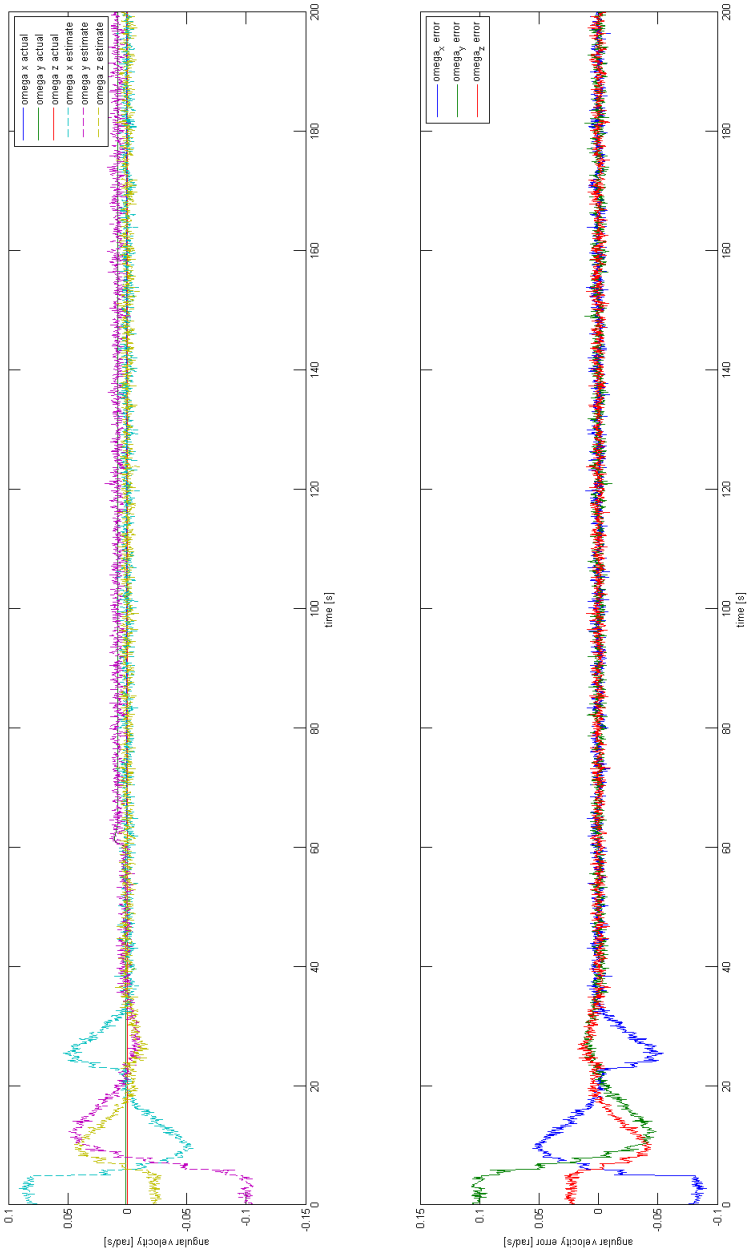

Figure 10.7: The more rapid swing-in of the gyro bias results in a faster convergence of the angular velocity estimates.

10.1.1 No Initial Values and Better Tuning

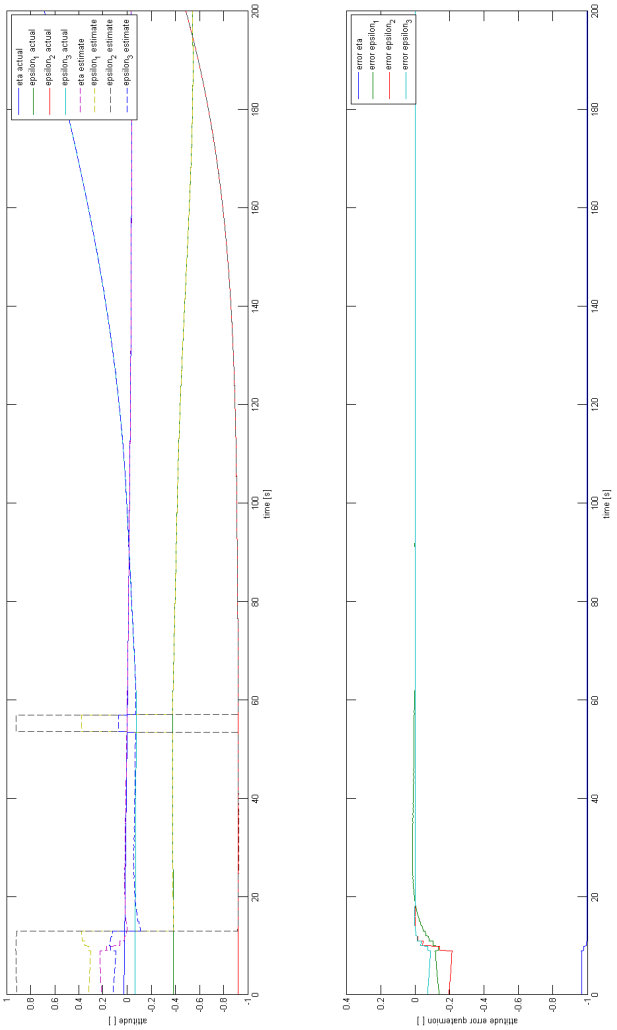

Figure 10.8: Attitude quaternion. The Multiplicative EKF is subject to extensive tuning of filter parameters. The filter parameters were tuned and another simulation was performed. Here we have better tuning and the filter finds a better initial value and converges to the correct attitude with less overshoot. There are three step changes in estimated quaternion, one between 10 and 15 seconds and two between 50 and 60 seconds. This is because the sign of the quaternion in this graph is defined such that the sign of the estimated η follows the sign of the actual η , which crosses zero at these points. These step changes do not affect the actual attitude error, as seen in the lower graph. This is because the quaternion $+\mathbf{q}$ represents the same rotation as $-\mathbf{q}$. An attitude maneuver is simulated for one second at 60 seconds, and for 100 seconds between 100-200 seconds, applying a torqe about all axis.

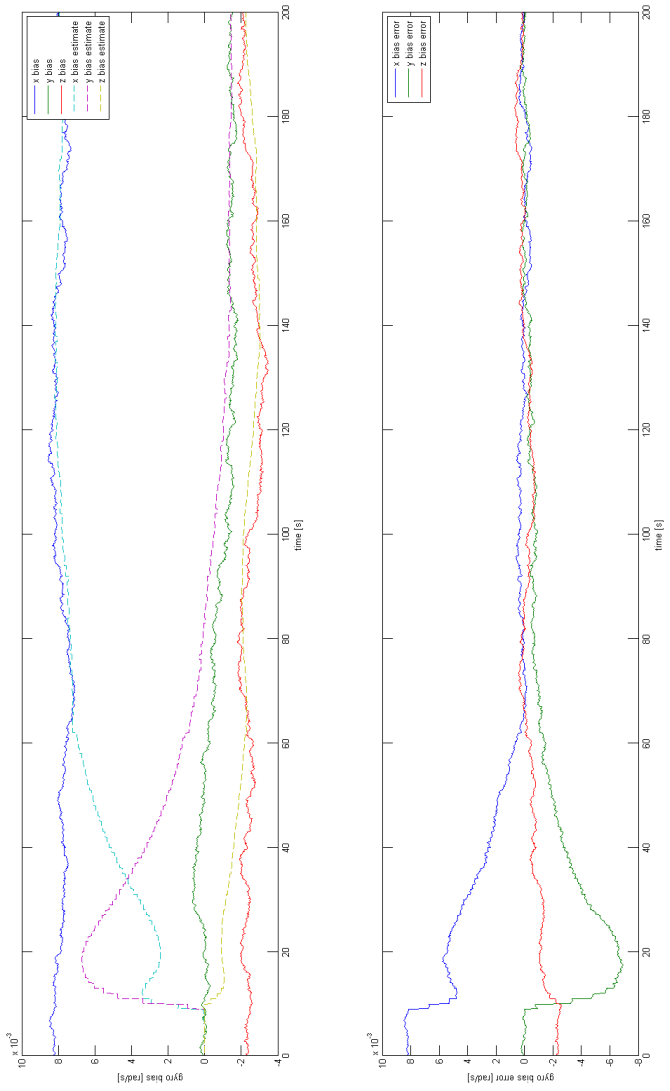

Figure 10.9: A slowly time varying gyro bias is simulated. We see the same wind-up during swing-in but of significantly smaller character than Figures 10.2 and 10.6.

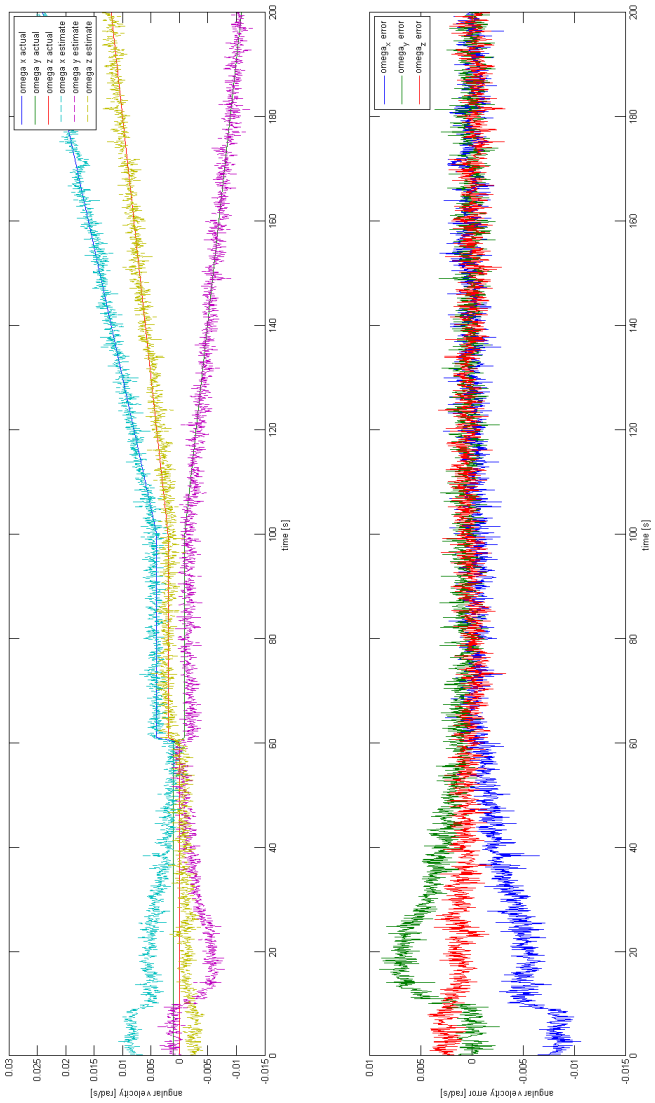

Figure 10.10: When the gyro bias is determined with sufficient accuracy, the angular velocity estimates follows the actual values well. The high frequency noise can be filtered out to improve the estimates. We can also see the attitude maneuvers at 60 seconds and 100-200 seconds.

10.1.2 No Initial Values, Better Tuning, and GPS Outage

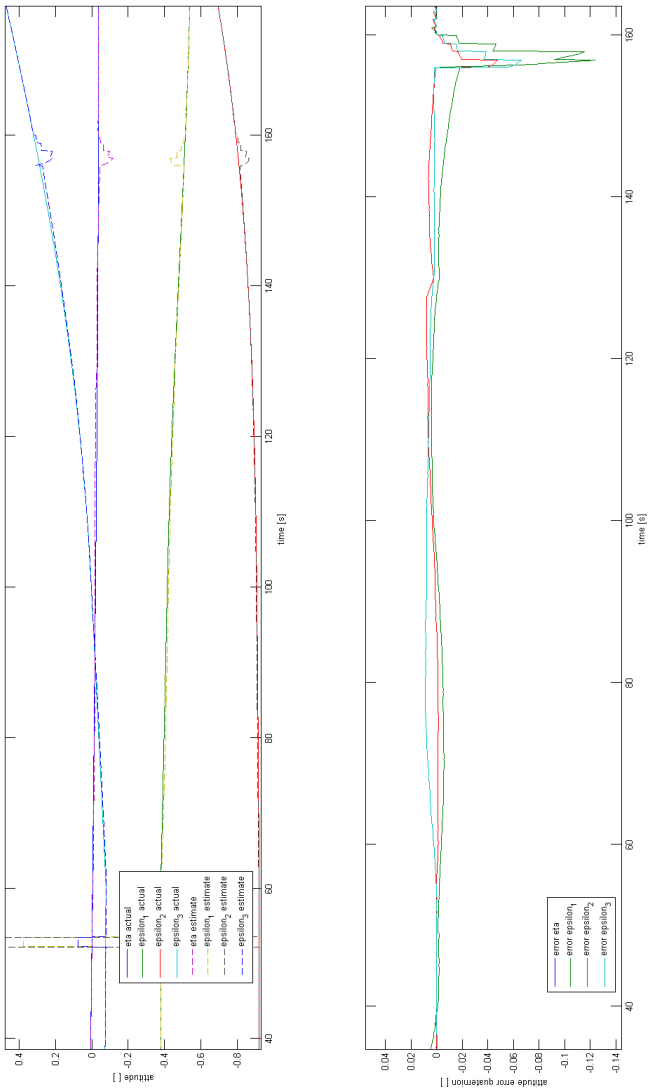

Figure 10.11: GPS outage for both attitude, velocity and position data was simulated from 50 to 150 seconds. We see the estimates stay close to the actual values through the attitude maneuvers. At 150 seconds the GPS data are available and the ambiguities must be resolved again.

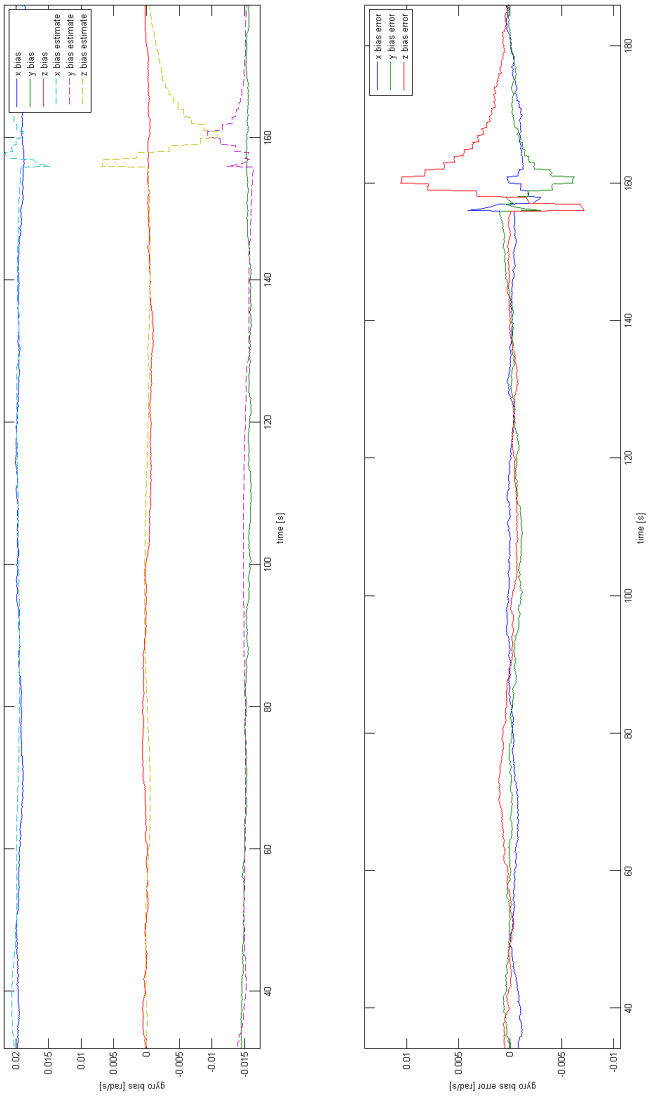

Figure 10.12: The gyro bias estimates stay close to the actual value through the GPS outage and attitude maneuvers. When the GPS data return, the ambiguities are reset and the gyro bias experiences wind-up.

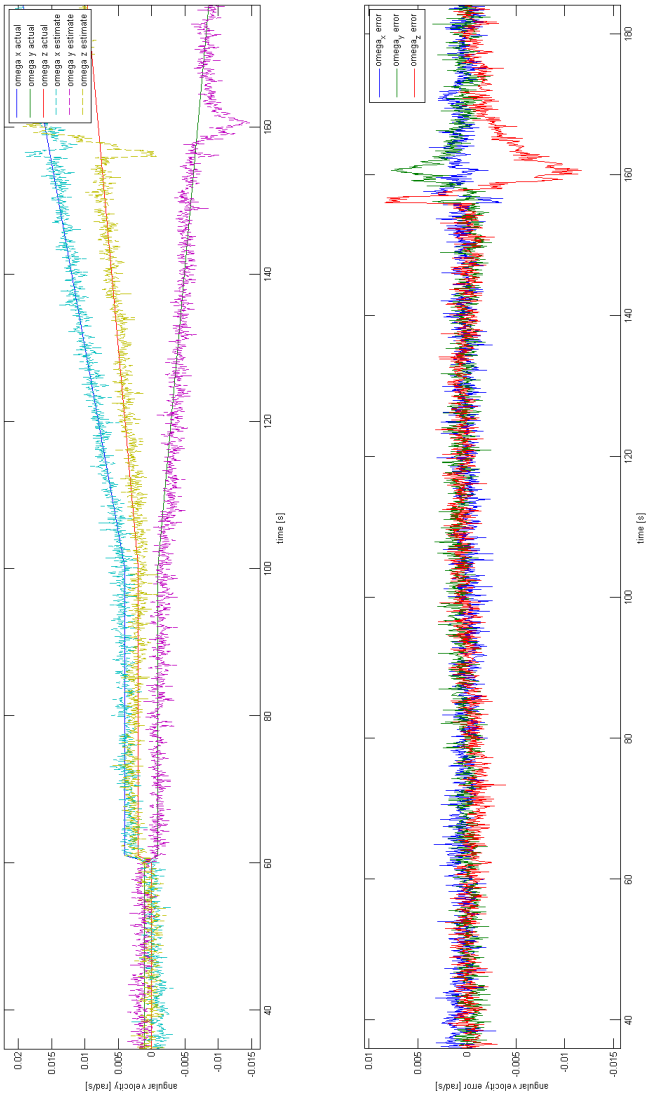

Figure 10.13: Due to the good gyro bias estimates the angular velocity estimates stay close to the actual value.

10.1.2.1 Fast Integer Ambiguity Resolution with GPS Outage

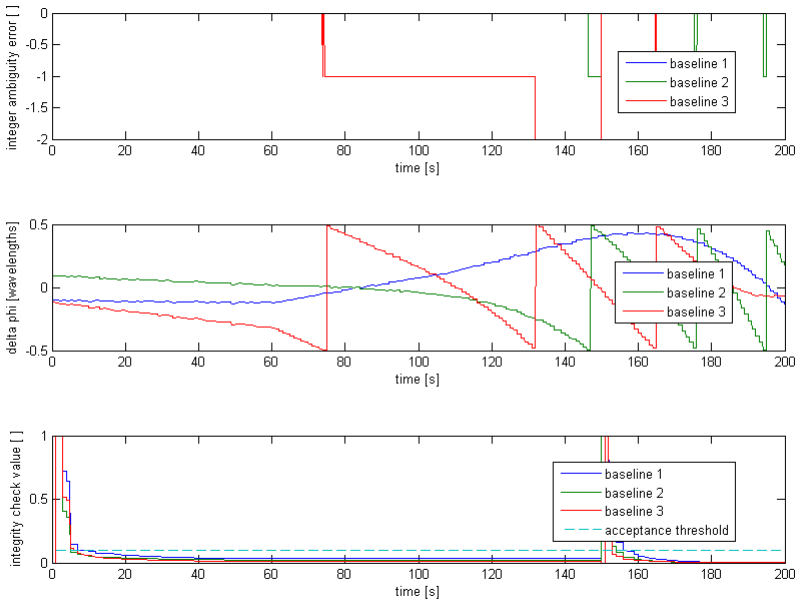

Figure 10.14: When GPS data return, the integer ambiguity resolution must be reset due to loss of lock of the ambiguity solution.

10.2 Orbit Simulation Results

10.2.1 Circular Orbit

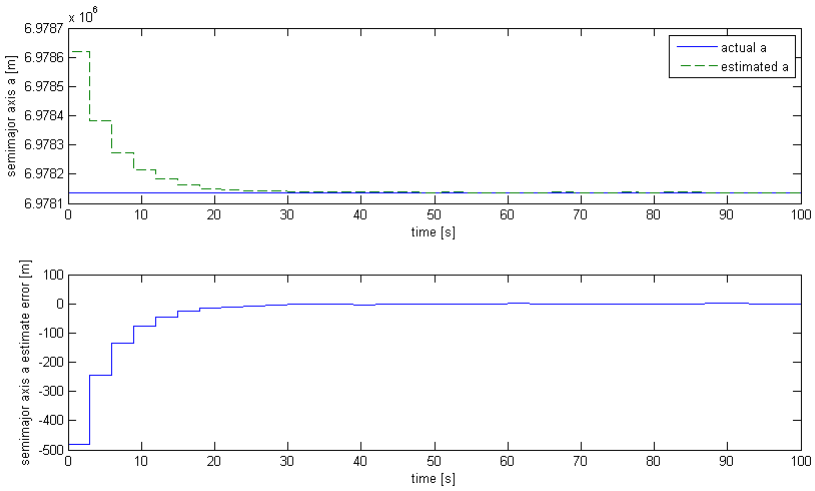

Figure 10.15: The estimate of semi-major axis, a, has a large initial error due to its sensitivity to errors in the velocity and position measurements. It converges step-wise to its actual value. The steps are due to the sampling time of the velocity and position measurements, which is set to 3 seconds for this simulation.

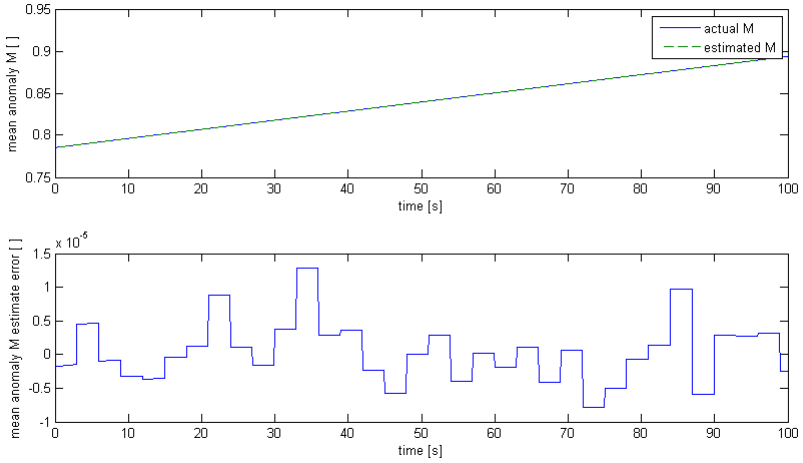

Figure 10.16: The mean anomaly, M, has a very small initial error and this error stays small, of the order 10^{-5} , throughout the simulation.

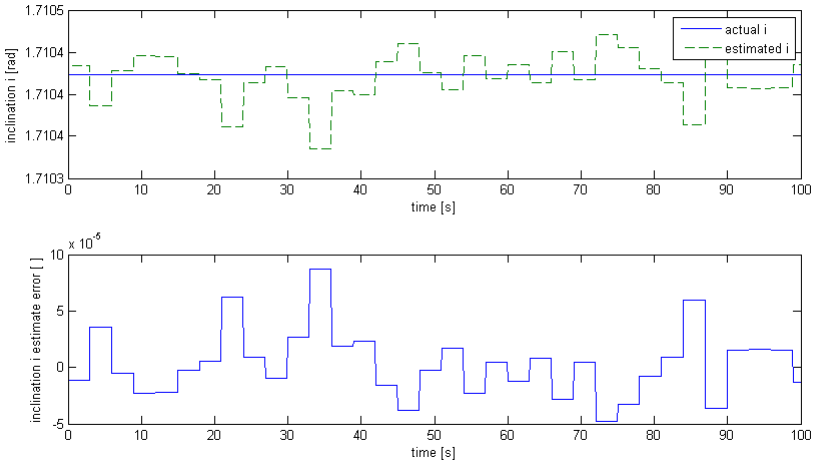

Figure 10.17: Similarly for the estimation of inclination, i, the error is very small, of the order 10^{-4} , throughout the simulation.

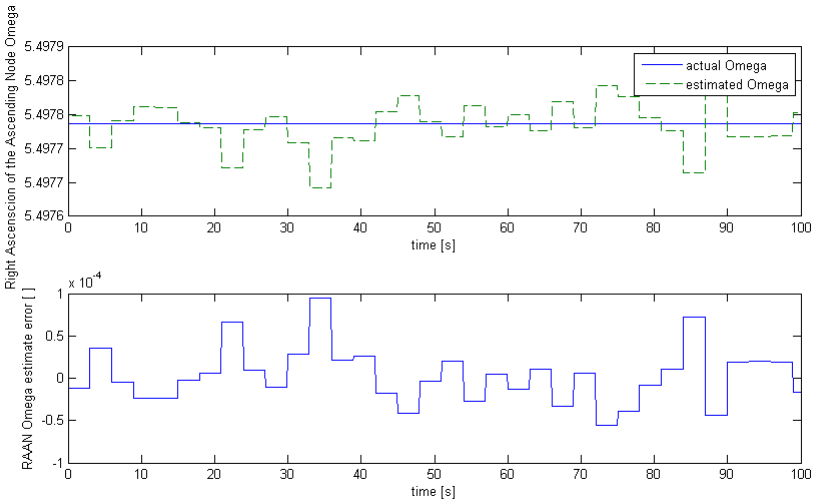

Figure 10.18: The Right Ascension of the Ascending Node, Ω , has a very small initial error and this error stays small, of the order 10^{-5} , throughout the simulation.

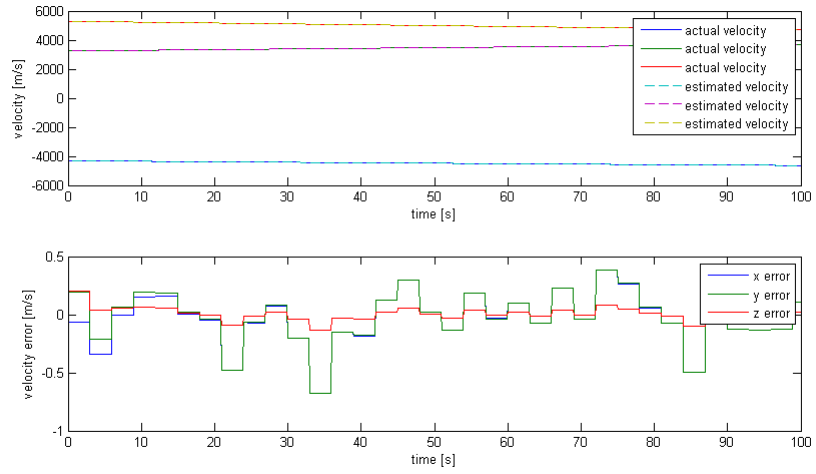

Figure 10.19: The velocity estimates for a circular orbit has a small initial error and the error stays small throughout the simulation period, in the range of $\frac{1}{2}\frac{m}{s}$ whereas the estimates are in the order of several $\frac{km}{s}$.

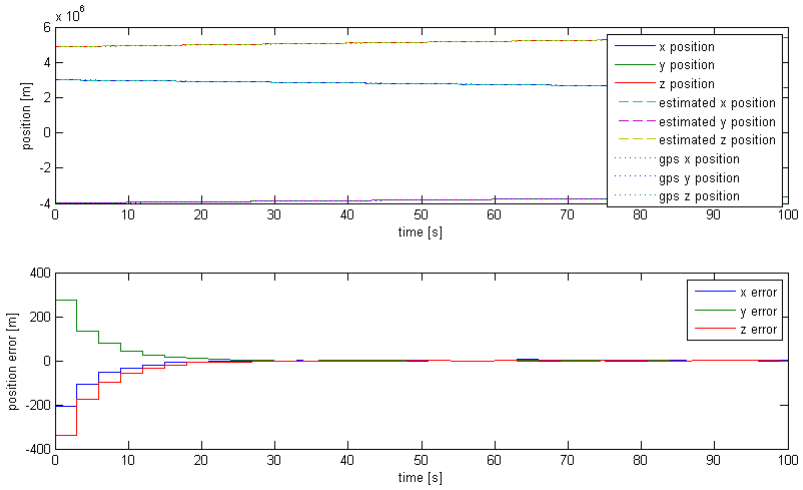

Figure 10.20: The position estimate is dependent on the orbital parameters, and due to the large error in initial value of a , the position estimate also has a large error in its simulated initial value. When a converges, so does the position estimate.

10.2.1.1 Circular Orbit with GPS Outage

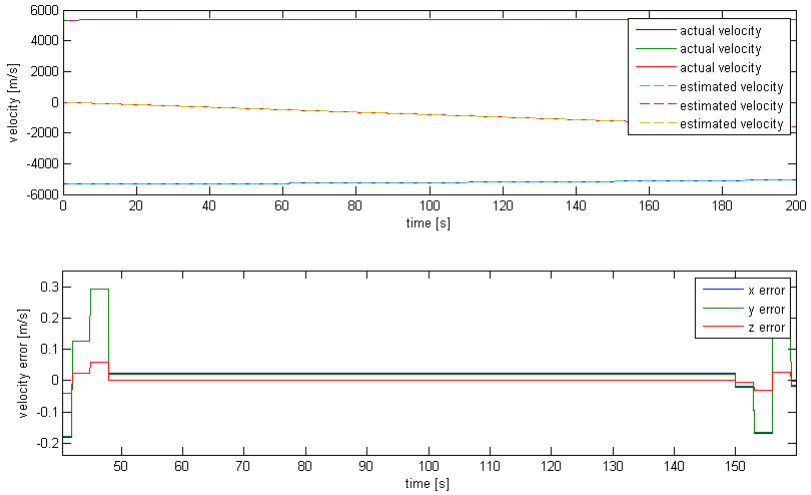

Figure 10.21: The filter keeps the estimated velocity error small during the outage.

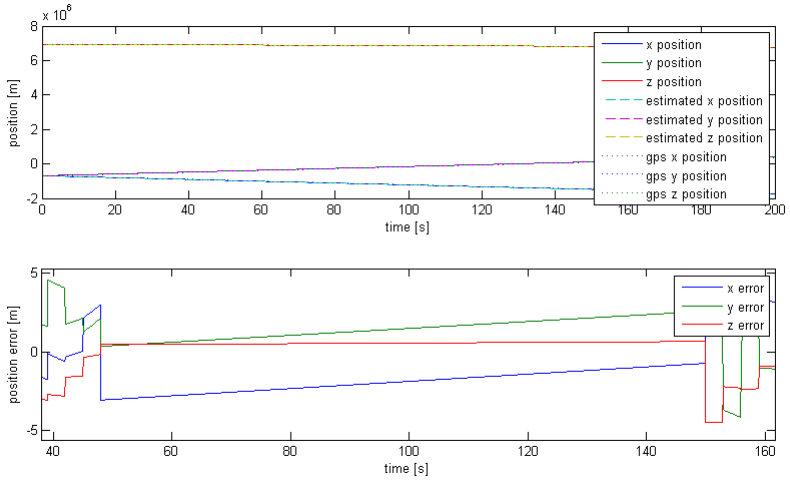

Figure 10.22: The filter ensures that also the estimated position error is kept small during the outage.

10.2.2 Elliptic Orbit, $e = 0.0001$

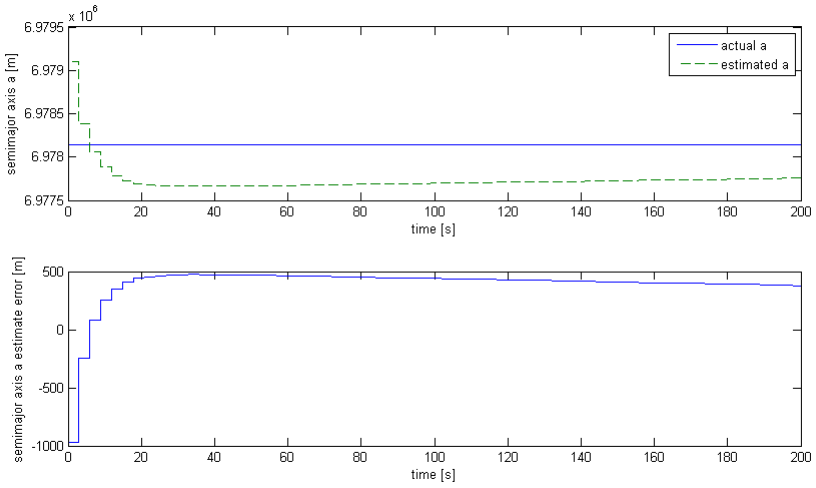

Figure 10.23: For a slightly elliptic orbit, we see that the estimated semi-major axis, a , starts out with a large initial error. Due to the circular orbit assumption in the model the estimated a does not converge to the actual value of a quickly, if at all.

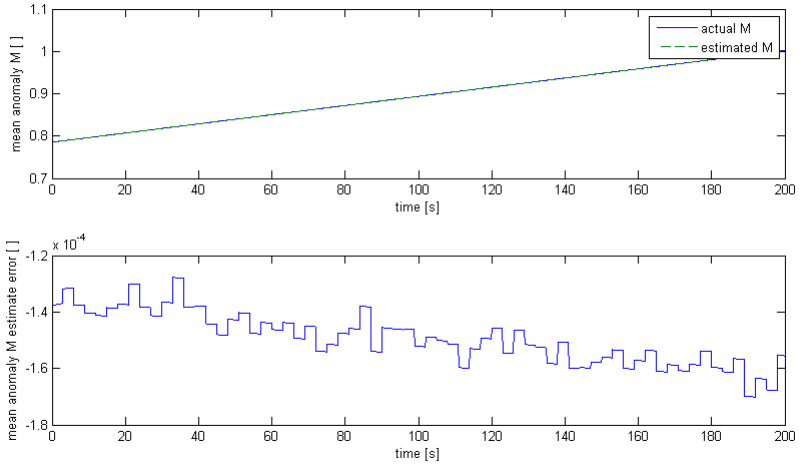

Figure 10.24: The mean anomaly estimate has a small steady-state error due to the eccentricity of the orbit, $e \neq 0$.

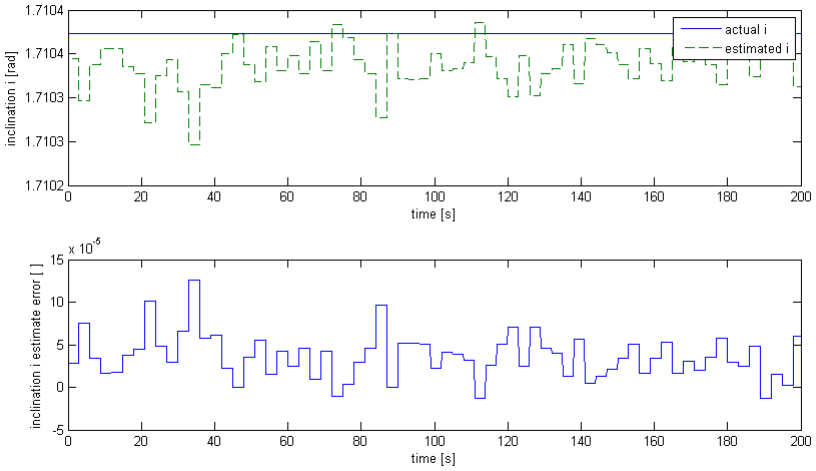

Figure 10.25: The estimate of i also inherits a small steady-state error due to the eccentricity.

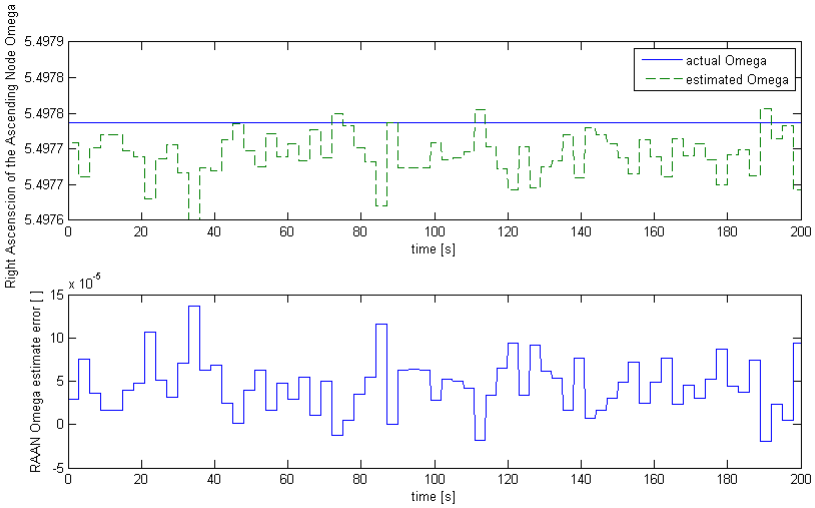

Figure 10.26: A small steady-state error is observed also in the estimate of the Right Ascension of the Ascending Node, Ω.

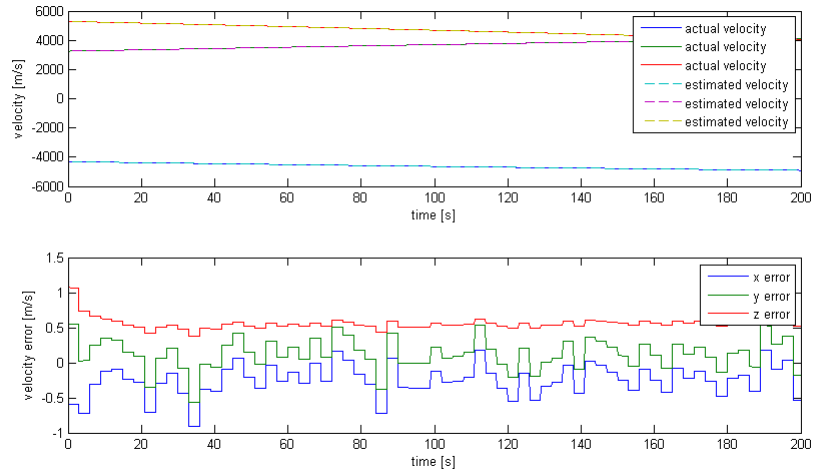

Figure 10.27: The velocity estimates are also subject to the steady-state error as they are functions of the Keplerian orbital elements.

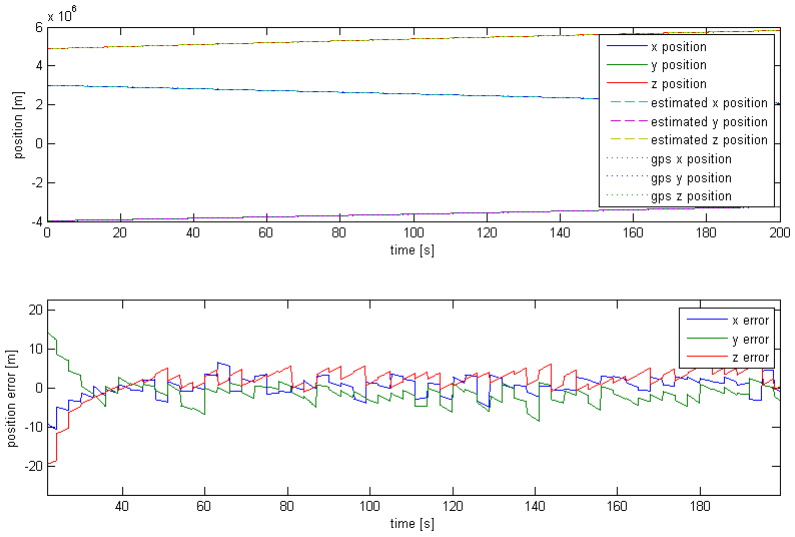

Figure 10.28: Position estimates are highly dependent on the estimate of the semi-major axis, a , therefore it inherits the large initial error. After converging to the actual values, the position estimates are highly dependent on continuous GPS position updates to keep the estimates close to the actual value.

10.2.2.1 Elliptic Orbit with GPS Outage

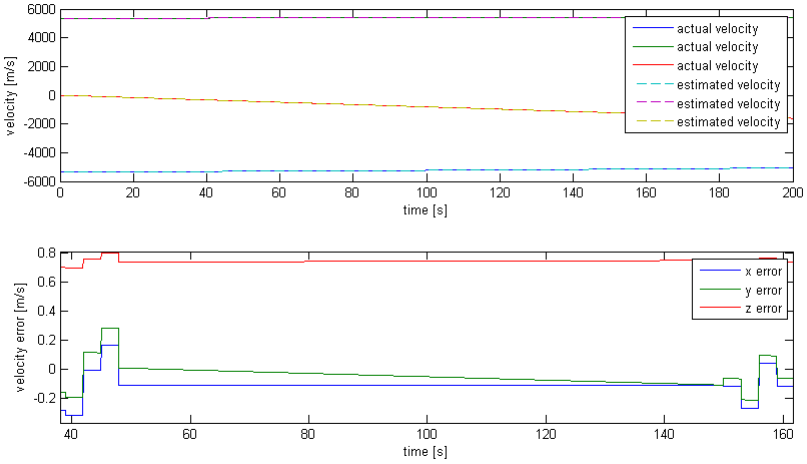

Figure 10.29: For this eccentricity, the velocity estimates do not suffer greatly from the GPS outage, as the estimator can rely on the mathematical model of the orbit.

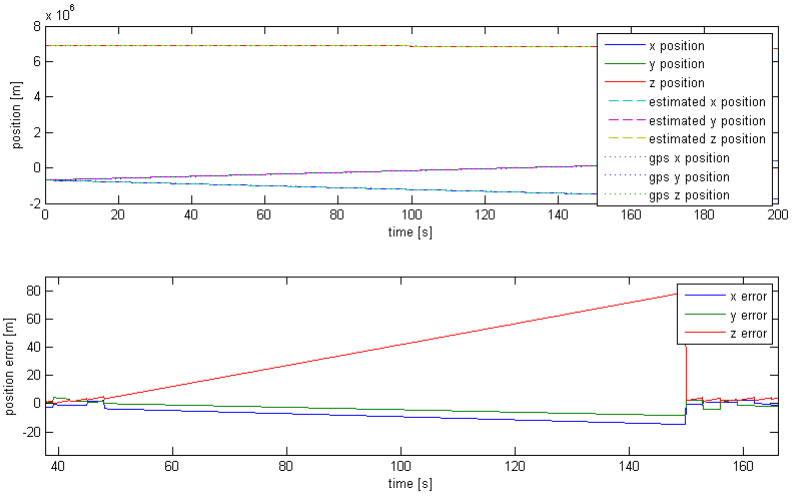

Figure 10.30: The circular orbit assumption causes the position estimates to diverge from the actual value in the absence of continuous GPS updates when the orbit is eccentric.

Part IV

Closing remarks

Chapter 11

Conclusions

11.1 Discussion and Recommendations for Further Work

A Multiplicative Extended Kalman Filter was developed in this thesis for the purpose of orbit and attitude estimation. The filter was developed in a tightly coupled manner with respect to the GPS attitude solution, and complemented by a fast integer ambiguity resolution method. The orbit estimator is also included in this design. Howeer, the MEKF now operates as an ordinary EKF, inherent in the MEKF.

The MEKF serves as a very robust filter for attitude estimation. It can combine measurements from multiple sensors in multiple reference frames, and it is also fairly robust with respect to modelling errors. Another important remark regarding the MEKF is its performance with respect to tuning of the filter parameters. The process of tuning the parameters of a (Multiplicative) Extended Kalman Filter may not be intuitive, and this task can be time consuming. A poorly tuned filter can cause overshoot and inaccuracy, or even instability. This is shown in the results where the filter was simulated using two different sets of tuning parameters.

It should also be noted that the MEKF is a computationally intensive algorithm, as several matrix multiplications are performed. A countermeasure to this can be found by noting that although the attitude estimator is reliant upon data from the orbit estimator, the two estimators are not directly reliant on each other computationally. Therefore an implementation of the estimators on a microcontroller in a small satellite could be split into a separate orbit estimator of dimension 4×4 (6×6 for an elliptic estimator) and a separate attitude estimator of dimension 6×6 , to reduce computational demands from the current 10×10 dimensioned MEKF/EKF. Alternatively, and perhaps more advantageously, the attitude estimator could be implemented as a nonlinear observer,

such as the semiglobal stability proven nonlinear observer design presented by Grip, Fossen, Johansen and Saberi (2012). The results from this work also proves convergence for all initial attitudes. Based on Vandersteen, Diehl, Aerts, and Swevers (2011), it is also concluded that optimization-based moving horizon state estimation (MHE) could be further investigated.

The usefulness of the circular orbit estimator design can be questioned, since it is accurate only for circular orbits. The actual orbit will never be perfectly circular, and from Figures 10.29 and 10.30 we can see that for eccentric orbits, frequent GPS updates are necessary in order to keep the position estimates close to the actual position. The required frequency of these updates is dependent on the degree of necessity of situational awareness in terms of position and velocity estimates, as well as on the required precision. If the NTNU Test Satellite project is able to expand its downlink infrastructure on ground through for example Kongsberg Satellite Services at Svalbard, as well as through global amateur satellite networks such as GENSO, this could open up for more advanced payloads and scientific experiments. It could then become a requirement that the GPS position and velocity engineering function should always be switched on during flight, meaning that it also should be switched on when the spacecraft is in the Earth shadow. It is therefore necessary to perform an analyses of the power consumption of spaceborne GPS receivers followed by a selection of hardware based on these analysis. The GPS receiver design could preferrably be carried out as a thesis assigment in order to optimize the project outcome in terms of educational benefits. However, an elliptic orbit estimator would require less frequent GPS velocity and position updates in order to maintain satisfactory estimates, and the circular orbit estimator design should therefore be expanded to an elliptic orbit estimator design.

A potential source of error to the design in this thesis is the length of the baselines. Longer baselines would reduce the effect of noise in the phase measurements on the attitude solution, and examining the effect of baseline length on the accuracy of the attitude estimates should be done in future work.

The integer ambiguity resolution could be improved by adding gyro -and magnetometer support to improve the performance of the fast integer ambiguity resolution in terms of robustness and speed of convergence. Gyro- and magnetometer support and the current attitude estimate could also be used to improve reacquisition after loss of lock due to GPS outage.

Another source of error is the uncertainties of the Gauss coefficients in the IGRF model, which can be estimated. The uncertainties in the IGRF model arise do the complicated dynamic nature of the geomagnetic field, a property which is particularly dominant over the polar regions of the Earth, where the field is highly dynamic. Induced residual magnetic fields within the spacecraft hardware could become a problem, causing the electronics to behave in undesirable

ways. The assumption regarding the magnetometer mesurements, that no residual magnetic field from the magnetic torquers would become significant, may become invalid. It could be useful to map the dynamic magnetic signature of the spacecraft.

The GPS signals could also contain errors which are not taken into account in this thesis. Of these, known errors are the effect of GPS signal polarization, antenna phase center variation, vehicle self-reflected multipath, and antenna structure phase pattern interactions. It could also be useful to replace the current naive ADOP search method with a more sophisticated way of selecting the GPS satelitte signals.

APPENDIX A Matlab Code

Multiplicative Extended Kalman Filter

```
function [q delta_eps gyro_b orbit pos vel N P_res ...
    chosen_sat]= MEKF_ao(gyro, mag, gps_pos, gps_vel,...
    gps_sat_pos,gps_delta_phi,new_att, new_posvel,...
    sat_indices, init, time)
%-------------------------------------------------------------
% MEKF_ao - Multiplicative Extended Kalman Filter for attitude
% and orbit estimation.
%
% x = [delta_eps gyro_b orbit_a orbit_M orbit_i orbit_RAAN]
% u = [gyro]
% y = [mag delta_r gps_pos gps_vel]
%
% Input         Description         Unit        Frame
% ..............................................
% gyro          gyro                [rad/s]     body
% mag           magnetometer        [nT]        body
% gps_pos       gps position        [m]         ecef
% gps_vel       gps velocity        [m/s]       ecef
% gps_delta_phi phase difference [rad]       n.a.
%
% Output        Description         Unit        Frame
% ..........................................................
% q             quaternion          [ ]         body_hat to eci
% delta_eps     delta epsilon       [ ]         body to body_hat
% gyro_b        gyro bias           [rad/s]     body
% pos           position            [m]         eci
% vel           velocity            [m/s]       eci
%
% Author:   Tale Sundlisæter (based on AP_MEKF.m by Harald
%           Nøkland
%           Nonlinear Observer Design for GNSS and IMU
%           Integration, 2011)
% Date:     May 2012
% -------------------------------------------------------------
global filter_ts gps_att_ts gps_ts gps_lever_arm;
global wgs84_GM res_accept ADOP_tol;

n = 10;
    % Number of states
h = filter_ts;
```

```
    % Sampling interval
global L1_lambda;
    % GPS L1 signal wavelength
global baselines;
    % Baselines for gps attitude determination
gps_arm = gps_lever_arm;
    % GPS lever arm, position of antenna in BODY [m]
nLOS = 4;
    % Number of GPS satellites to use for attitude estimation
nBL = size(baselines,2);
    % Number of baselines available for attitude estimation

% Double differencing matrix A, fractional and integer parts
dd_A_frac = diag([1/sqrt(2) 1/sqrt(6) 1/sqrt(12)]);
dd_A_int = [1 -1  0  0 ;1  1 -2  0; 1  1  1 -3];
dd_A = dd_A_frac*dd_A_int;

% Design matrices:

% State variances
var_delta_eps = [1e-4 1e-4 1e-4]; % Gyro variance
var_gyro_b = [1e-5 1e-5 1e-5];      % Gyro bias variance


var_a = 5;
         % Semimajor axis a variance
var_M = 1e-5;
         % Mean anomaly M variance
var_i = 1e-5;
         % Inclination i variance
var_RAAN = 1e-5;
         % Right ascencion of the ascending node RAAN variance

var_orbit = [var_a var_M var_i var_RAAN];
         % Orbit parameter variance

process  = [var_delta_eps var_gyro_b var_orbit];
Q = diag(process);                  % Process noise

% Measurement variances
var_mag  = [1e-4 1e-4 1e-4];
                % Magnetometer variance [rad]
var_delta_r = (0.0004^2).*ones(1,nBL*nLOS);
                % GPS deltarange variance [m]
```

```
var_gps_pos = [5 5 5];
                % GPS position variance [m]
var_gps_vel = [1e-1 1e-1 1e-1];
                % GPS velocity variance [m]

meas     = [var_mag var_delta_r var_gps_pos var_gps_vel];
R = diag(meas);                              % Measurement noise
% -------------------------------------------------------------


persistent q_hat delta_eps_bar gyro_b_bar orbit_bar P_bar...
    N_bar Res resdiag prev_indices init_ADOP last_att_time...
    last_posvel_time;

if(init == 1)
    % First run, set up initial values, initialize integer
    % ambiguity resolution and choose GPS SV to use for
    % attitude estimation

    delta_eps_bar = [0 0 0]';       % Initial delta epsilon
    gyro_b_bar = [0 0 0]';          % Initial gyro bias

    % Find rotation from ecef to eci
    R_ecef2eci = ecef2eci(time);

    % Convert GPS position and velocity to ECI
    pos_bar = R_ecef2eci*gps_pos;
    vel_bar = R_ecef2eci*gps_vel;

    % Find magnetic field reference vector from IGRF
    pos_sphere = cart2sphere(pos_bar);
                % = [radius, elevation, azimuth]
    IGRF_sphere = igrf11syn(2014, pos_sphere(1),...
        pos_sphere(2),pos_sphere(3));
    R_sph2ecef = Rsphere2ecef(pos_sphere(2), pos_sphere(3));
    mag_bar_ecef = R_sph2ecef*IGRF_sphere;

    % Normalize magnetic field reference vector for use in
    % q-method
    mag_bar_ecef = mag_bar_ecef/norm(mag_bar_ecef);


    % Find LOS vectors for all satellites
    LOS_all = gps_sat_pos - gps_pos*ones(1,size(gps_sat_pos,2));
```

```
LOS_all = colnormalize(LOS_all);

% Select satellites
[LOS_index init_ADOP] = choose_sat(LOS_all,nLOS);
LOS_bar = LOS_all(:,LOS_index);

prev_indices = sat_indices(LOS_index);

gps_delta_phi = gps_delta_phi(LOS_index,:);

% Resolve integer ambiguities
[N_bar Res] = intambres(gps_delta_phi',LOS_bar,...
[0 0 0 0],init);

N_bar = N_bar';

% Find deltaranges and baselines in ECEF
gps_delta_r = L1_lambda*(gps_delta_phi + N_bar);
B_ecef_bar = (LOS_bar*LOS_bar')\LOS_bar*gps_delta_r;




% Convert GPS position and velocity measurement in
% ECI to Kepler
% orbital elements
[a_bar e_bar M_bar i_bar RAAN_bar w_bar] =...
 eci2koeC(pos_bar,vel_bar);

% Assuming circular orbit -> e = 0 and w := 0
orbit_bar = [a_bar M_bar i_bar RAAN_bar]';

% Find initial attitude from magnetometer and baseline
%measurements using q-method

qm_weighting = [10000 1 1 1];
qm_weighting = qm_weighting/sum(qm_weighting);
q_hat = qmethod(qm_weighting,...
    colnormalize([R_ecef2eci*mag_bar_ecef...
    R_ecef2eci*B_ecef_bar]),...
    colnormalize([mag baselines]));

if(q_hat(1) < 0)
    q_hat = -q_hat;
```

```
    end

    resdiag = zeros(3,4);

    last_att_time = time;
    last_posvel_time = time;

    P_bar = diag([1e-7 1e-7 1e-7   1e-8 1e-8 1e-8 ...
        10 1e-3 1e-4 1e-4]);  % Initial error covariance
else
    % Calculate estimated ECI position from orbit parameters
    [pos_bar,vel_bar] = koe2eci(orbit_bar(1),0,orbit_bar(2),...
     orbit_bar(3),orbit_bar(4),0);
end



if(new_att == 1 && init == 0)


    rechoose = 0;
    temp_ind = [0 0 0 0]';
    reset = zeros(1,nLOS);
    % Check if chosen satellites are no longer visible
    for i = 1:nLOS
        ind = find(sat_indices == prev_indices(i), 1);
        if(isempty(ind))
            rechoose = 1;
            break;
        end
        temp_ind(i) = ind;
    end

    rechoose = 1;

%     % Check if ADOP has degraded
%     if(~rechoose)
%         LOS_bar = gps_sat_pos(:,temp_ind) -...
%         pos_bar*ones(1,nLOS);
%         LOS_bar = colnormalize(LOS_bar);
%         curr_ADOP = adop(LOS_bar);
%         if(curr_ADOP > init_ADOP + ADOP_tol)
%             rechoose = 1;
%         end
```

```
%     end

    % If necessary, choose new satellites
    if(rechoose)
        %LOS vectors
        LOS_all = gps_sat_pos -...
        pos_bar*ones(1,size(gps_sat_pos,2));
        LOS_all = colnormalize(LOS_all);

        % Select satellites
        [LOS_index init_ADOP] = choose_sat(LOS_all,nLOS);

        LOS_bar = LOS_all(:,LOS_index);
        new_indices = sat_indices(LOS_index);

        % Check for satellite change

        for i = 1:nLOS
            if(new_indices(i) ~= prev_indices(i))
                reset(i) = 1;
            end
        end
        prev_indices = new_indices;
        gps_delta_phi = gps_delta_phi(LOS_index,:);
    else
        gps_delta_phi = gps_delta_phi(temp_ind,:);
    end


    % Detect outage
    if(time > last_att_time + gps_att_ts + filter_ts)
        reset = ones(1,nLOS);
    end

    % Resolve integer ambiguities
    [N_bar Res] = intambres(gps_delta_phi',LOS_bar,reset,init);

    N_bar = N_bar';
    %Find delta_r
    gps_delta_r = L1_lambda*(gps_delta_phi + N_bar);
    last_att_time = time;
end

% Predicted rotation matrices:
```

```
delta_q_bar = qbuild(delta_eps_bar);
R_bar_body2eci = Rquat(qmult(delta_q_bar,q_hat));
R_bar_eci2body = R_bar_body2eci';

R_ecef2eci = ecef2eci(time);
R_eci2ecef = R_ecef2eci';

R_bar_ecef2body = R_bar_eci2body*R_ecef2eci;
R_bar_body2ecef = R_bar_ecef2body';

%Magnetic field reference vector:
pos_sphere = cart2sphere(pos_bar);
            % = [radius, elevation, azimuth]
IGRF_sphere = igrf11syn(2014, pos_sphere(1), pos_sphere(2),...
pos_sphere(3));
R_sph2ecef = Rsphere2ecef(pos_sphere(2), pos_sphere(3));
mag_bar_ecef = R_sph2ecef*IGRF_sphere;
mag_bar_body = R_bar_ecef2body*mag_bar_ecef;

if(new_att == 1)
    %Predicted delta_r
    Delta_r_bar = LOS_bar'*R_bar_body2ecef*baselines;
else
    Delta_r_bar = zeros(4,3);
    gps_delta_r = zeros(4,3);
    LOS_bar = zeros(3,4);
end

%Estimated measurement:
y_bar = [mag_bar_body;
         reshape(Delta_r_bar,nLOS*nBL,1);...
         pos_bar;...
         vel_bar]...
      + [meas_bias_mag(-delta_eps_bar,...
         Rquat(qinv(q_hat))*R_ecef2eci*mag_bar_ecef,P_bar);...
         meas_bias_dr(delta_eps_bar,baselines,LOS_bar,...
         R_eci2ecef*Rquat(q_hat),P_bar);...
        [0 0 0]';...
        [0 0 0]'];

% Real measurement:
y = [mag;...
     reshape(gps_delta_r,nLOS*nBL,1);...
     R_ecef2eci*gps_pos;...
```

```
     R_ecef2eci*gps_vel];

% Linearized observation matrix
H1 = [2*Smtrx(R_bar_ecef2body*mag_bar_ecef) zeros(3,7)];
if(new_att == 1)
    H2 = [Hdelta_r(R_bar_body2ecef,baselines,LOS_bar)...
    zeros(nBL*nLOS,7)];
else
    H2 = zeros(nBL*nLOS,10);
end

if(new_posvel == 1)
    H3 = [zeros(6) HorbitC(orbit_bar)];
else
    H3 = zeros(6,10);
end

H = [H1;H2;H3];

% Kalman gain
K = P_bar'*H'/(H*P_bar*H' + R);

% Dead-reckoning
if(new_att == 0)
    % If there is no new attitude data, do not use
    % attitude measurements
    K(:,4:4+(nLOS*nBL)-1) = zeros(n,nLOS*nBL);
else
    % If there is new attitude data, check the integrity of the
    % ambiguity solution before using the measurements
    for i = 1:nLOS
        % Integer ambiguity solution integrity check:
        % If three times the square root of the diagonals of
        % the estimate error covariance matrix of the integer
        % ambiguity solution is less than res_accept, then the
        % solution is accepted and the measurement is used
        resdiag(:,i) = 3*sqrt(diag(Res(:,:,i)));
        if(any(resdiag(:,i) > res_accept) ||...
                any(~isreal(resdiag(:,i))) ||...
                any(isinf(resdiag(:,i))))
            K(:,4+(i-1)) = 0;
            K(:,4+(i-1)+nLOS) = 0;
            K(:,4+(i-1)+2*nLOS) = 0;
        end
```

```
    end
end

if(new_posvel == 0)
    % If there is no new position and velocity data, do not
    % use the position and velocity measurements
    K(:,4+(nBL*nLOS):end) = 0;
end

% Update error state estimate with measurement
delta_eps_hat = delta_eps_bar + K(1:3,:)*(y - y_bar);
gyro_b_hat = gyro_b_bar + K(4:6,:)*(y - y_bar);
if norm(delta_eps_hat)>1, delta_eps_hat =...
        0.999*delta_eps_hat/norm(delta_eps_hat); end

% Update orbit state estimate with measurement
orbit_hat = orbit_bar + K(7:10,:)*(y - y_bar);

% Sanity checks on output
orbit_hat(2) = mod(orbit_hat(2),2*pi);
            % Mean anomaly defined from 0 to 2*pi
orbit_hat(3) = mod(orbit_hat(3),pi);
            % Inclination defined from 0 to pi
orbit_hat(4) = mod(orbit_hat(4),2*pi);
            % RAAN defined from 0 to 2*pi

% Calculate position and velocity from estimated orbit
[pos_hat,vel_hat] = koe2eci(orbit_hat(1),0,...
    orbit_hat(2),orbit_hat(3),orbit_hat(4),0);

% Output:
dq_hat = qbuild(delta_eps_hat);
q_hat = qmult(q_hat,dq_hat);
q_hat = q_hat/norm(q_hat);
delta_eps = delta_eps_hat;
q = q_hat;
gyro_b = gyro_b_hat;
pos = pos_hat;
vel = vel_hat;
orbit = orbit_hat;
N = N_bar;
P_res = resdiag;
chosen_sat = prev_indices;
```

```
% Reset error quaternion
delta_eps_hat = [0 0 0]';

% Compute error covariance for updated estimate:
IKH = eye(n) - K*H;
P = IKH*P_bar*IKH' + K*R*K';


% Project ahead:
delta_eps_dot = 0.5*(gyro - gyro_b_hat) +...
    prop_bias(delta_eps_hat,gyro,gyro_b_hat,P);
gyro_b_dot = [0 0 0]';
orbit_n_hat = sqrt(wgs84_GM/orbit_hat(1)^3);
orbit_dot = [0 orbit_n_hat 0 0]';

delta_eps_bar = delta_eps_hat + h*delta_eps_dot;
gyro_b_bar = gyro_b_hat + h*gyro_b_dot;
orbit_bar = orbit_hat + h*orbit_dot;
if norm(delta_eps_bar)>1, delta_eps_bar =...
        0.999*delta_eps_bar/norm(delta_eps_bar); end

PHI1 = [-0.5*Smtrx(gyro - gyro_b_hat) -0.5*eye(3) zeros(3,4)];
PHI2 = zeros(3,10);

dM_da = -(3/2)*(wgs84_GM/orbit_hat(1)^5);
PHI3 = [zeros(4,6) [0 dM_da 0 0]' zeros(4,3)];

PHI = eye(n) + h*[PHI1;PHI2;PHI3];
GAMMA = h*eye(n);
P_bar = PHI*P*PHI' + GAMMA*Q*GAMMA';
```

Integer Ambiguity Resolution

```
function [N P] = intambres(delta_phi,LOS,reset,init)

% [N P] = intambres(delta_phi,LOS,reset,init) resolves the
% integer ambiguity problem that rises when calculating
% deltaranges from phasse differences and line-of-sight vectors.
%
% The algorithm is based on the Fast Integer Ambiguity
% Resolution algorithm by Lightsey, Crassidis and Markley in
% which the set of all possible candidates {n1,n2,n3} is first
% reduced by a geometric inequality before searching through the
% remaining candidates.
%
% Inputs:
%   delta_phi:  nBL x nLOS matrix of phase differences
%   LOS:        3 x nLOS matrix of column LOS vectors
%   reset:      1 x nLOS boolean vector. If element i is true,
%               the integer ambiguity solution for LOS i is
%               reset
%   init:       initialization boolean, set to true (or 1) for
%               the first run, false (or 0) otherwise
%
% Outputs:
%   N:          nBL x nLOS integer matrix solving the ambiguity
%               problem
%   P:          nBL x nBL x nLOS error covariance matrix used
%
% Author:   Tale Sundlisæter (Based on Fast Integer Ambiguity
%           Resolution for GPS Attitude Determination by
%           Lightsey, Crassidis & Markley, 1999.)
% Date:     May 2012
% ------------------------------------------------------------
%
% Copyright (C) 2012 Tale Sundlisæter
%
% This program is free software: you can redistribute it
% and/or modify it under the terms of the GNU General Public
% License as published by the Free Software Foundation,
% either version 3 of the License, or (at your option) any
% later version.
%
% This program is distributed in the hope that it will be
% useful, but WITHOUT ANY WARRANTY; without even the
```

```
% implied warranty of MERCHANTABILITY or FITNESS FOR A
% PARTICULAR PURPOSE.  See the GNU General Public License
% for more details.
%
% You should have received a copy of the GNU General Public
% License along with this program.  If not, see
% <http://www.gnu.org/licenses/>.
%
% E-mail: tale.sundlisater@gmail.com

persistent N_candidates candycount J P_inv prev_index;
persistent delta_phi_prev Gamma B B_inv B_inv_cubed GBG;

global baselines L1_lambda;

% Baselines given in wavelengths
b1 = baselines(:,1)/L1_lambda;
b2 = baselines(:,2)/L1_lambda;
b3 = baselines(:,3)/L1_lambda;

% Precompute for speed
sqnorm_b1 = norm(b1)^2;
sqnorm_b2 = norm(b2)^2;
sqnorm_b3 = norm(b3)^2;
b1b2 = dot(b1,b2);
b1b3 = dot(b1,b3);
b2b3 = dot(b2,b3);
upper = floor(norm(b1));
lower = -floor(norm(b1));

%measurement noise
w_ij = 0.0004/L1_lambda; %[wavelengths] typical phase noise

nLOS = size(LOS,2);
nBL = 3;

if(init)
    % First run, preallocate and initialize,
    % set reset to all true
    N_candidates = -inf.*ones(3,nLOS,(upper-lower+1)^3);
    J = inf.*ones(nLOS,(upper-lower+1)^3);
    P_inv = inf.*ones(3,3,nLOS,(upper-lower+1)^3);
    candycount = zeros(nLOS,3);
    delta_phi_prev = inf*ones(3, nLOS);
```

```
    reset = ones(1,nLOS);
    Gamma = (w_ij^-2).*[b1 b2 b3];
    B = (w_ij^-2).*(b1*b1'+b2*b2'+b3*b3');
    B_inv = inv(B);
    B_inv_cubed = B_inv * B_inv * B_inv;
    GBG = Gamma'*B_inv_cubed*Gamma;
else
    % Subsequent runs, check delta_phi to detect integer
    % crossings, to keep
    % the solution locked
    for j = 1:nLOS
        for i = 1:nBL
            if(delta_phi(i, j) - delta_phi_prev(i, j) > 0.6)
                % Integer crossing downwards
                if(N_candidates(i,j,prev_index(j)) > lower)
                % if previous integer is above the lowest
                % possible, decrement it
                    N_candidates(i,j,prev_index(j)) =...
                        N_candidates(i,j,prev_index(j)) - 1;
                else
                    % else, loop around to highest
                    N_candidates(i,j,prev_index(j)) = upper;
                end
            elseif(delta_phi(i,j) - delta_phi_prev(i,j) < -0.6)
                % Integer crossing upwards
                if(N_candidates(i,j,prev_index(j)) < upper)
                 % if current integer is below the lowest
                 % possible, increment it
                    N_candidates(i,j,prev_index(j)) =...
                        N_candidates(i,j,prev_index(j)) + 1;
                else
                    % else, loop around to lowest
                    N_candidates(i,j,prev_index(j)) = lower;
                end
            end
        end
    end
end

% Check each GPS signal seperately
for j = 1:nLOS

    % Check if the signal has been reset
    if(reset(j))
```

```
% The signal has been reset and we must find a new
% ambiguity solution {n1j,n2j,n3j}. First we reduce
% the search space by using a geometric inequality
% that must be satisfied for a candidate
% solution {n1j,n2j,n3j} to be accepted.

% Reset ambiguity solution for signal j
N_candidates(:,j,:) = -inf;
J(j,:) = inf;
P_inv(:,:,j,:) = inf;
candycount(j,:) = 0;

% Preallocate candidate storage 1 for speed
N_temp1 = -inf.*(ones(2, (upper-lower+1)^2));

% Search through n1 and n2
for N1=lower:upper,
    for N2=lower:upper,

        % Check geometric inequality to accept
        % or reject current n1 and n2
        if(sqnorm_b1*sqnorm_b2 > (b1b2^2 +...
                sqnorm_b2*(delta_phi(1,j) + N1)^2 ...
                - 2*(delta_phi(1,j)...
                + N1)*(delta_phi(2,j) + N2)*(b1b2) ...
                + sqnorm_b1*(delta_phi(2,j) + N2)^2))

            % n1 and n2 accepted as candidates,
            % increment candidate
            % counter, add n1 and n2 to accepted
            % candidates
            candycount(j,1) = candycount(j,1) + 1;
            N_temp1(:,candycount(j,1)) = [N1 N2]';
        end
    end
end

% Preallocate candidate storage 2 for speed
N_temp2 = -inf.*(ones(3,...
    candycount(j,1)*(upper-lower+1)));

% Iterate through accepted n1,n2 pairs to find matching
% n3 candidates
```

```
 for iter = 1:candycount(j,1)

     N1 = N_temp1(1,iter);
     N2 = N_temp1(2,iter);

     if(N1 == -inf)
         % If n1 is -inf we have searched through
         % all candidates
         break;
     end

% Search for n3 candidates for the current n1 and n2
     for N3=lower:upper,
         % Check geometric inequality to accept or
         % reject current n3
         if(sqnorm_b1*sqnorm_b3 > b1b3^2 +...
                 sqnorm_b3*(delta_phi(1,j)+ N1)^2 ...
                 - 2*(delta_phi(1,j) + N1)*...
                 (delta_phi(3,j)+ N3)*(b1b3) ...
                 + sqnorm_b1*(delta_phi(3,j) + N3)^2)
             % n3 accepted as candidate, increment
             % candidate counter, add n1, n2 and n3
             % to accepted candidates
             candycount(j,2) = candycount(j,2) + 1;
             N_temp2(:,candycount(j,2)) = [N1 N2 N3]';
         end
     end
 end

 % Preallocate candidate storage 3 for speed
 N_temp3 = -inf.*(ones(3, candycount(j,2)));

 for iter = 1:candycount(j,2)
     N1 = N_temp2(1,iter);
     N2 = N_temp2(2,iter);
     N3 = N_temp2(3,iter);
     if(N1 == -inf)
         % If n1 is -inf we have searched through
         % all candidates
         break;
     end
     % Check geometric inequality to accept or reject
     % current candidate set {n1,n2,n3}
     if(sqnorm_b2*sqnorm_b3 > b2b3^2 +...
```

```
                    sqnorm_b3*(delta_phi(2,j) + N2)^2 ...
                    - 2*(delta_phi(2,j) + N2)*(delta_phi(3,j)...
                    - N3)*(b2b3) ...
                    + sqnorm_b2*(delta_phi(3,j) + N3)^2)
                % Candidate set accepted, increment candidate
                % counter and add set to candidates
                candycount(j,3) = candycount(j,3) + 1;
                N_temp3(:,candycount(j,3)) = [N1 N2 N3]';


            end
        end
        % Truncate candidate storage 3 to remove empty elements
        N_temp3 = N_temp3(:,1:candycount(j,3));
        % Move candidates to main candidate storage
        N_candidates(:,j,1:candycount(j,3)) = N_temp3;

        % Reset score for all candidates
        J(j,1:candycount(j,3)) = 0;

        % Reset estimate error covariances for all candidates
        P_inv(:,:,j,1:candycount(j,3)) = 0;
    end

    % Compute loss function scores and estimate
    % error covariances for all candidates
    for candy = 1:candycount(j,3)

        % Calculate sigma_j^2
        sigma_sq = (delta_phi(:,j) + N_candidates...
            (:,j,candy))'*GBG*(delta_phi(:,j) +...
            N_candidates(:,j,candy)) -...
            trace(B_inv)^2;

        % Calculate loss function
        J(j,candy) = J(j,candy) + 0.5.*((1/sigma_sq)...
            *(norm((B_inv)*Gamma*(delta_phi(:,j) +...
            N_candidates(:,j,candy)))^2 ...
            - norm(LOS(:,j))^2 + trace(B_inv))^2 +...
            log(sigma_sq));

        % Calculate estimate error covariance P
        P_inv(:,:,j,candy) = P_inv(:,:,j,candy) +...
            (4/sigma_sq)*(delta_phi(:,j)+N_candidates...
```

```
            (:,j,candy))*(delta_phi(:,j)+...
            N_candidates(:,j,candy))';
    end
end

% Preallocate outputs
N = inf*ones(3,nLOS);
P = inf*ones(3,3,nLOS);

% Select the best ambiguity solution {n1j,n2j,n3j} for
% each sightline j
for j = 1:nLOS

    % Find the candidate that minimizes the loss function
    [score index] = min(J(j,1:candycount(j,3)));



    % Add the chosen candidate to the output
    N(:,j) = N_candidates(:,j,index);

    % Add the estimate error covariance matrix to the output
    P(:,:,j) = inv(P_inv(:,:,j,index));

    % Store the current measurement and solution
    delta_phi_prev = delta_phi;
    prev_index(j) = index;
end
```

Bibliography

[1] Adams, J. C., (2000) Robust GPS Attitude Determination for Spacecraft, Ph.D thesis, Stanford University, Department of Aeronautics and Astronautics.

[2] Bak, T. (1999) Spacecraft Attitude Determination - a Magnetomoter Approach, Ph.D Thesis, Aalborg University, Department of Control Engineering.

[3] Chao, C. C., (2005) Applied Orbit Perturbation and Maintenance, American Institute of Aeronautics and Astronautics, Inc.

[4] Cohen, C. E., (1992) Attitude Determination Using GPS, Ph.D. Dissertation, Stanford University, Dept. of Aeronautics and Astronautics, Dec. 1992.

[5] Cohen, C. E., (1996) Attitude Determination, Global Positioning System: Theory and Applications Volume II, Parkinson, B. W. and Spilker Jr. J. J., American Institute of Aeronautics and Astronautics Inc.

[6] Crassidis, J. L., Markley, F. L., Cheng, Y. (2007) Survey of Nonlinear Attitude Estimation Methods, Journal of Guidancy, Control and Dynamics, Vol. 30, No. 1, January-February 2007.

[7] Ellis, J. F., (1979) Interferometric Attitude Determination with the Global Positioning System, Journal of Guidance and Control, colume 2, no. 6, Nov-Dec 1979.

[8] Fossen, T. I. (2011) *Handbook of Marine Craft Hydrodynamics and Motion Control*, John Wiley & Sons Ltd.

[9] Ghadaki, F., Alonsoperez, V., Sundlisæter, T., Romano, P., (2011) ISU Space Studies Programme 2011: Team Project on Small Satellite for Capacity Building in Space Technology Development, International Astronautical Federation International Astronautical Congress, October 2011.

[10] Grip, H.F., Fossen, T.I., Johansen, T.A., and Saberi, A., (2012) Attitude Estimation Using Biased Gyro and Vector Measurements With Time-Varying Reference Vectors, IEEE Transactions on Automatic Control, May 2012.

[11] Helvajian, H. and Janson, S. W., (2008) Small Satellites: Past, Present and Future, The Aerospace Press, American Institute of Aeronautics and Astronautics, Inc.

[12] Jenssen, K. L., Yabar, K., (2011) Development, Norwegian University of Science and TechnologyImplementation and Testing of Two Attitude Estimation Methods for Cube Satellites, Department of Engineering Cybernetics.

[13] Jouffroy, J. and Fossen, T., (2010) A tutorial on incremental stability analysis using contraction theory, Modeling, Identification and Control 31(3): pp 93-106.

[14] Kruczynski, L. R., Evans, A. G. and Hermann, B. R., (1989) Using GPS to Determine Vehicle Attitude: USS Yorktown Test Results, Proceedings of the Institute of Navigation ION GPS-89 Conference, Colorado Springs, CO, Sept. 1989

[15] Lefferts, E. J., Markley, F. L., and Shuster, M. D., (1982) Kalman Filtering for Spacecraft Attitude Estimation, Journal of Guidance, Control and Dynamics, Vol. 5, No. 5, 1982.

[16] Lightsey, E. G., Markley, F. L., Crassidis, J. L., (1999) Fast Integer Ambiguity Resolution for GPS Attitude Determination, American Institute of Aeronautics and Astronautics, Inc.

[17] Markley, F. L., (2003) Attitude Error Representations for Kalman Filtering, Journal of Guidance, Control and Dynamics, Vol. 63, No. 2, 2003.

[18] Markley, F. L., (2003) Attitude Estimation or Quaternion Estimation?, The Journal of the Astronautical Sciences, Vol. 52, No. 1-2, January-June 2004.

[19] Markley, F. L., (2004) Multiplicative vs. Additive Filtering for Spacecraft Attitude Determination, Proceedings, 6th Cranfield Conference on Dynamics and Control of Systems and Structures in Space, Cranfield University Press.

[20] Nøkland, H., (2011) Nonlinear Observer Design for GNSS and IMU Integration, M.Sc. thesis, NTNU, Department of Engineering Cybernetics.

[21] Ose, S. S., (2004) Attitude determination for the Norwegian student satellite nCube, M.Sc. thesis, Norwegian University of Science and Technology, Department of Engineering Cybernetics.

[22] Rohde, J., (2007) Kalman Filter for Attitude Determination of Student Satellite, M.Sc. thesis, Norwegian University of Science and Technology, Department of Engineering Cybernetics.

[23] Sabatini, A.M., (2006) Quaternion-Based Extended Kalman Filter for Determining Orientation by Inertial and Magnetic Sensing, IEEE Transaction on Biomedical Engineering, Vol.53, No. 7, July 2006.

[24] Schwab, A. L., (2006) Quaternions, Finite Rotation and Euler Parameters, Delft University of Technology, Laboratory for Engineering Mechanics.

[25] Schweiger, M., (2010) Orbiter User Manual, Space Flight Simulator, 2010 Edition, Copyright (c) 2000-2010 Martin Schweiger. www.orbitersim.com, last accessed 30. June 2012.

[26] Spinney, V. W. (1976) Applications of Global Positioning System as an Attitude Reference for Near Earth Users, ION National Aerospace Meeting, Naval Air Development Center, Warminster PA, April 1976.

[27] Sundlisæter, T., (2008) Norsk Studentsatellittprosjekt, Romfart Magazine, issue no. 2008-2, pp. 49-50.

[28] Sundlisæter, T., (2012) Literature Study for NTNUs Test Satellite, Department of Engineering Cybernetics, Norwegian University of Science and Technology.

[29] Svartveit, K., (2003) Attitude determination of the NCUBE satellite, M.Sc. thesis, NTNU, Department of Engineering Cybernetics.

[30] Tohami, S., Brembo, E.M., (2005) Sensor Modeling, Attitude Determination and Control for Micro-Satellite, M.Sc. thesis, Norwegian University of Science and Technology, Department of Engineering Cybernetics in cooperation with Kongsberg Defence and Aerospace.

[31] Vandersteen, J., Diehl, M., Aerts, C., and Swevers, J., (2011) A Novel Attitude Estimation Filter for the PLATO Space Mission Based on Moving Horizon Esitmation, 8th International ESA Conference on Guidance, Navigation and Control Systems, 5-10 June 2011.

[32] Van Graas, F. and Braasch, M., GPS Interferometric Attitude and Heading Determination: Initial Flight Test Results, Navigation, volume 38, no. 4, Winter 1991-1992.

[33] Vik, B., (2011) Integrated Satellite and Inertial Navigation Systems, Norwegian University of Science and Technology, Department of Engineering Cybernetics.

[34] Wahba, G., (1965) A Least Squares Estimate of Satellite Attitude, SIAM Review 7(3): 409.

[35] Wertz, J. R., (1978) Spacecraft Attitude Determination and Control, Microcosm, Kluwer Academic Publishers.

[36] Wertz, J. R. and Larson, W. J., (199) Space Mission Analysis and Design, 3rd edition, Space Technology Library.