Autonomous Navigation for Nano-satellites during Earth-Moon Transfer

MSc Thesis

Supervisor
dr.ir. Erwin Mooij
Author
F.A.R. Belien - 1221582

July 16, 2015

Faculty of Aerospace Engineering · Delft University of Technology

Faculty of Aerospace Engineering
Delft University of Technology

Table of Contents

Foreword

Here comes the foreword.

Abstract

Here comes the abstract.

Chapter 1

Introduction

The thesis question is:
How can a nano-satellite autonomously navigate during lunar transfer and in lunar orbit with sufficient performance to control and operate it?

New subquestions: To solve this main design question, a range of subquestions needs to be answered in the following order.

1. What are the requirements for an autonomous navigation system on a nano-satellite during lunar transfer and in lunar orbit?
2. What sensors and input data are available to obtain a position, velocity, attitude and rotational velocity estimate of a nano-satellite during lunar transfer and in lunar orbit?
3. How can the performance of filter/state variable combinations be evaluated for nano-satellite missions?
4. What is the influence of different measurement inputs, depending on pre-processing, on the correlation between position and attitude estimation of the nano-satellite.
5. What is the influence of the combination of state estimation filters (EKF and UKF) on position and velocity estimation?
6. What are the hardware and software requirements to implement this navigation system for real-time navigation on board of a nano-satellite?

In short: the goal is to find a concept of a system that works, show sufficient performance, remains flexible. This is a first step in a research that needs to be continued.

Old subquestions: To solve this main design question, a range of subquestions needs to be answered in the following order.

1. What are the requirements for an autonomous navigation system on a nano-satellite during lunar transfer and in lunar orbit?
2. What sensors and input data are available to obtain a position, velocity, attitude and rotational velocity estimate of a nano-satellite during lunar transfer and in lunar orbit?
3. What is the influence of state variables (Cartesian coordinates and MEE) on the implementation of position and velocity estimation of the nano-satellite?
4. How can the performance of filter/state variable combinations be evaluated for nano-satellite missions?
5. What is the influence of the combination of the aforementioned state variables and state estimation filters (EKF, UKF and UPF) on position and velocity estimation?
6. What are the hardware and software requirements to implement this navigation system for real-time navigation on board of a nano-satellite?

Chapter 2

Background

This chapter gives background information on the type of mission and the current developments, and provides input for the requirements analysis in section 3.1. The scope is limited to what is currently present in the field of nano-satellite missions at Delft University of Technology and plans for the future, with a focus on attitude and orbit determination and control. Firstly, the past and current nano-satellite missions are presented in section 2.1. Then, OLFAR and the future developments and goals are presented in section 2.2.

2.1 Past and current Missions

Delft University of Technology launched its first mission, Delfi-C^3, in 2008 as a part of their space education programme [15]. Their second mission, Delfi-n3Xt, was launched in 2013. The launch of nano-satellites provides increased in-house knowledge of system design and hands-on experience for students. Both satellites are 3U Cubesats. A Cubesat is a standard size for nano-satellites. A Cubesat consists of one or more cubes, with one cube measuring 10x10x10 cm. A 3U Cubesat consists of 3-units and measures 10x10x30 cm. In figure 2.1, Delfi-C^3 and Delfi-n3Xt are shown.

In the Delfi-C^3 mission, the attitude determination is performed by measuring the incidence of the angle of the Sun on the solar panels. Furthermore, two experimental Sun sensors are on board. The passive attitude control system consists of a permanent magnet and magnetic hysteresis material, using the presence of the Earth magnetic field. The main goal is to reduce the rotation rate of the satellite to under 2 rotations per minute [15]. There is no active attitude or orbit control present.

For Delfi-n3xt new Sun sensors have been developed at the university, based on a quadrant photodiode and a window in PEEK material. Furthermore, a commercially available three-axis digital compass gives the orientation using the magnetic field of Earth. The

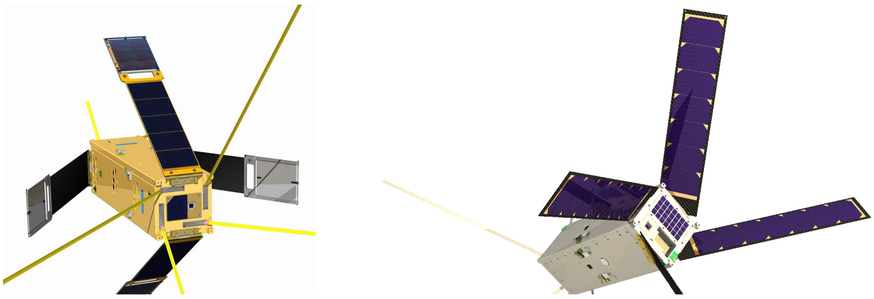

Figure 2.1: Delfi-C^3 (left) and Delfi-n3Xt (right) [15]

compass has an accuracy of 3° after calibration. These sensors are coupled with a b-dot controller for detumbling and safe mode, and a Kalman filter for advanced attitude determination. Attitude control is performed using in-house developed reaction wheels and magnetorquers, which are controlled with a PID controller and a reaction wheel offloading algorithm.

Currently, the university is working on its third mission, DelFFi [15]. This mission comprises two satellites, Delta and Phi, which form a part of the QB50 mission. QB50 is an European FP7 project for facilitating access to space and consists of 50 Cubesats, delivered by universities around the world [7]. Those Cubesats are going to work together to map the upper layers of the atmosphere. It is one of the first missions to test swarm principles. The DelFFi satellites are designed to use GPS receivers for position determination and will demonstrate formation flying and swarm capabilities using various Guidance, Navigation and Control (GNC) algorithms. These algorithms are essential for the further development towards larger nano-satellite formations or swarms, as required for missions like OLFAR.

2.2 OLFAR and Future Development

The orbiting low-frequency array (OLFAR) is a project for a radiotelescope formed by an array of nano-satellites [8] [15]. The main scientific goal is to observe the redshift in hydrogen motion from the dark ages of radio astronomy by performing measurements in the frequency range between 10 kHz and 30 MHz. These dark ages occured around 100,000 years after the big bang. During this period the first stars were formed. These observations provide valuable information on the formation of the universe. [5]

Due to interference between the low-frequency signals with noise from Earth, the deployment of such an array is not possible close to Earth. Different deployment locations have been and are being considered, each with its advantages and disadvantages: the Earth-Moon L2 point, a lunar orbit at various altitudes, the Earth-Sun L2 point *et cetera*. All these deployment locations have one common property: they are far away from or shielded against Earth.

As a cost-effective solution, a large array of nano-satellites is seen as the solution. However, numerous challenges need to be tackled. These challenges come mainly from the combination of three aspects. Firstly the limited capabilities of a single nano-satellite, secondly the requirements on operating a large number of satellites and thirdly the location of deployment.

One solution is the application of swarm technology. A swarm of nano-satellites can be compared to a swarm of bees. Separately, the nano-satellites are not capable of performing great things, but with a large number they can complement each others capabilities. The three main characteristics of a swarm of nano-satellites are cooperation, autonomy and flexibility. Cooperation between the nano-satellites requires communication and relative position knowledge. The autonomy of each separate satellite is a necessity to get rid of expensive tracking, navigating and controlling from Earth. Flexibility implies the possibility to add new satellites to the swarm and have old or damaged satellites leave it. These three basic characteristics are the main design criteria for the satellites in the OLFAR project and form the basic inputs to the requirements analysis. [5]

To reach the ultimate goal of constructing an array of spacecraft, several precursor missions are being proposed. These include DelFFi to test propulsion systems and formation flying capabilities, and ends at the e-Moth nano-satellite mission concept to reach the Moon [14]. Further pre-cursor missions are planned to allow for a stepwise approach and verification of all critical systems before launching the complete system.

As already mentioned, several challenges remain. These challenges can be found in the areas of propulsion, navigation and communications. This thesis focuses on the navigation of the nano-satellites between Earth and Moon for OLFAR and its precursor missions primarily, although further applications exist. As described above, the main characteristics of the navigation system should comply to those of the main OLFAR satellite and swarm characteristics: cooperation, autonomy and flexibility. These three characteristics are to be incorporated with their own implications. The reasons for autonomy are cost savings and reduced operational complexity by reducing ground-based tracking. Cooperation can increase the accuracy of the system by using position information of nearby satellites. Flexibility is also required to increase robustness. Throughout the mission, certain sensors or input data for the navigation can be temporarily unavailable (*e.g.*, Sun sensors during eclipse or limited field-of-view (FOV)) and the system must be able to cope with less or more information. Since OLFAR is taken as the basis and is needs to be kept in mind during the requirements analysis.

Chapter 3

System Architecture

The first step in developing a system is analysing the requirements and deriving the design of the system. Therefore, this chapter starts with the requirements analysis in section 3.1. Then, hardware is selected, which is used as a reference during this research in section 3.2. These components are then combined in two interface charts in section 3.3. The system interface chart shows the complete guidance, navigation and control system as it is to be implemented in the satellite, including both hardware and software. In the development interface chart, an overview is given of the software, which is to be programmed to create a simulation environment for the navigation system.

3.1 Requirements Analysis

The requirements are drawn from the background information in chapter 2. The requirements are divided in three parts: one on mission level, one on system level and one on navigation subsystem level. The requirements are listed in table 3.1.

3.1.1 Mission Requirements

The system requirements come straight from the mission description of OLFAR. As this system is to serve two mission types, *i.e.* OLFAR and precursor missions, the requirements must fit both mission types. As the precursor missions are also being covered by the out-of-swarm operations of the satellites, there is no need for separate investigation. The main requirements for OLFAR are defined by the overall mission architecture, *i.e.* a swarm.

The mission requirements are derived from the swarm philosophy and the five operational modes: deployment, transfer to operational orbit, in-swarm operations, out-of-swarm operations and end-of-life. During all these mission phases, except the in-swarm operations, the satellite is assumed to be on its own. As the number of satellites is larger than would be possible to handle comfortably with ground-based operations, they need to be able to operate autonomously. This thesis focuses on the attitude and position determination during the transfer, with a limit on the transfer between Earth and Moon. However, extensions to the other non-swarm modes remains possible.

To be able to design proper attitude and orbit control, it is required to track the position and attitude of the spacecraft. As mentioned before, this navigation needs to be performed autonomously, which is the prime requirement on mission level. One of the goals of this research is to find out if it is possible to determine the attitude and position with sufficient accuracy to be able to control the satellite during its transfer from Earth to the Moon. Therefore, the accuracy of the position and attitude determination is not described by a numerical value, but what is achievable. As a goal, a position accuracy of better than 1000 m during the complete transfer should be reached.

As a result, there are two requirements on mission level: (1) autonomy and (2) sufficient accuracy for control during Earth-Moon transfer.

3.1.2 System Requirements

The system is restricted to a nano-satellite. This includes limitations and opportunities. The limitations are its limited size, mass and available power. Currently, the size of the satellites is estimated between 1 and 6 cubes of 10 by 10 cm, which cannot be completely used for the navigation system only. This limits the selection of components. The largest opportunity is the possibility of using *unconventional* hardware, like micro-processors from mobile phones with relatively high computational power, instead of radiation hardened, space-proven (large and heavy) hardware. These requirements will influence the possible hardware choices.

To reduce the cost of the system and make the study more realistic, all components must be commercially off-the-shelf (COTS).

The system requirements are finding a system with (1) a small mass, (2) minimal size and (3) limited power consumption, while (4) minimising the computational power. Furthermore, the components must be COTS.

3.1.3 Navigation Subsystem Requirements

The navigation subsystem needs to be flexible enough to include the available measurements and discard those which are not. Otherwise, the same requirements apply as on the system level.

3.2 Component Selection

First a selection is made of the available techniques to see which ones could be feasible. Then, a selection of components is presented of which the characteristics are being used during the research.

Mission Requirements	
Autonomy	The navigation must be performed autonomously
Navigation accuracy	Sufficient to allow for control during Earth-Moon transfer (1000 m)
System Requirements	
Mass	Minimal
Size	Minimal
Power	Minimal
Computational power	Minimal
Component selection	COTS
Navigation Requirements	
Input data	Flexible on available data

Table 3.1: Requirements overview

3.2.1 Technique Selection

The mission scenario of Earth-Moon transfer means that the system must be operable far away from Earth. This means that magnetometers are not to be used. GNSS receivers might be possible, although the main ones which are currently available for space are expensive and/or military grade equipment, especially those which can be used up to geostationary orbit and beyond. Therefore, these cannot be used. Radar is possible, although these require (in general) a lot of power. Laser ranging violates the autonomy requirement.

Therefore, the most suitable options are:

1. a star tracker
2. a sun sensor
3. an accelerometer
4. a gyro
5. an IMU, combining the accelerometer and a gyro
6. optical navigation

With further developments and use of certain techniques, it is possible that some of the techniques which are discarded can be used for part of the mission, *e.g.* GNSS can be used until geostationary altitude and beyond [12]. However, for limiting the scope of this thesis, the above selection of techniques is used.

The accelerometer and gyroscope are combined in one sensor, the IMU. For each component, one sensor is selected. Further research can improve the sensor selection, but this is part of further design iterations of the system. The main goal is to proof the concept and develop basic algorithms.

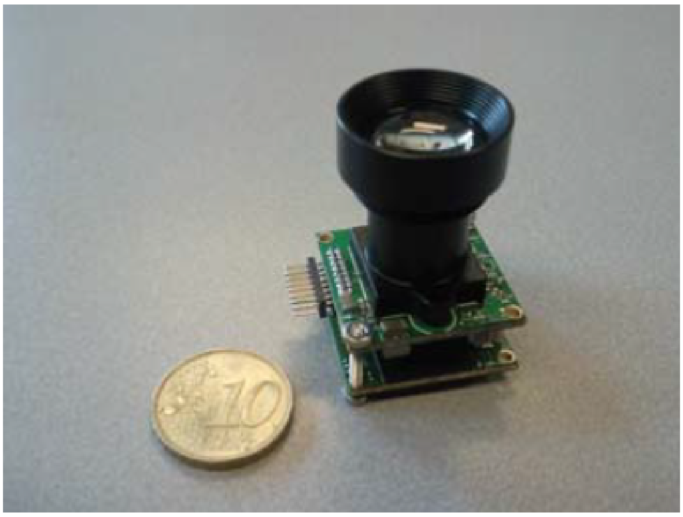

Figure 3.1: The ST-200 compared to a 10 eurocent coin [13]

3.2.2 Star Tracker - Pico Star Tracker ST-200

The Pico Star Tracker ST-200, see Figure 3.1, is a low-power star tracker and the successor of the Micro Star Tracker ST-100, designed by Berlin Space Technologies GmbH in cooperation with Syspa B.V. [13] The main changes with its predecessor are less mass, a lower power consumption, a reduced size and a shorter lifetime. This reduced lifetime is caused by a reduction of the aluminium shielding from 5 to 2 mm. This star tracker also has the possibility of acting as an optical camera. In this function it can be used for pattern matching, disc measurements and horizon scanning. This double use allows for a continuous use of the star tracker. In Table 3.2 more data can be found.

Parameter	Value
Mass	74 g
Power	0.7 W peak
Accuracy	$30''$
Design life	1 year
Market readiness	2011

Table 3.2: Characteristics of the ST-200 [13]

3.2.3 Sun Sensor - Micro Digital Sun Sensor

The micro Digital Sun Sensor of TNO Science and Industry [3] uses a masked CMOS Active Pixel Sensor. Its autonomy is increased by a wireless link to the avionics subsystem and a solar panel to power the sensor. Extra details can be found in Table 3.3. This Sun sensor has been tested on the Delfi-C^3 mission. Since it has been designed for low-Earth orbits, the radiation hardness is most likely not sufficient for lunar transfer. Sensors based on the same technique are used on other interplanetary missions, like BepiColombo, the capabilities of this sensor can be assumed to be realistic for future missions [ADD

SOURCE].

Parameter	Value
Mass	<30 g
Accuracy	<0.1°
FOV	120°
Dimensions	46x46x14 mm^3

Table 3.3: Characteristics of the micro Digital Sun Sensor [3]

3.2.4 IMU - ADIS16405

The ADIS16405, from Analog Devices [4], is a combination of an IMU and a magnetometer. The IMU part has three different dynamic modes to increase the resolution and dynamic range in different modes of operation. The magnetometer is included in the package, providing a lower cost for the system. The radiation resistance is unknown, but can again be assumed suitable for low-Earth orbit, but not for a high-Earth or lunar orbit. A summary of the specifications can be found in Table 3.4.

Parameter	Value
IMU	
Dynamic range	±300, ±150, ±75°/s
Scale factor	0.05, 0.025, 0.0125°/s
Initial bias error	±3°
Bias stability	$0.007°/s$
Angular random walk	$2°/\sqrt{hr}$
3dB bandwidth0	330 Hz
Magnetometer	
Dynamic range	±3.5 gauss
Scale factor	0.5 mgauss
Initial bias error	±4 mgauss
Output error	1.25 mgauss rms
General	
Dimensions	23x23x23 mm
Mass	16 g
Power	0.35 V

Table 3.4: Characteristics of the ADIS16405 [4]

3.2.5 Optical Navigation - Pico Star Tracker ST-200

As already mentioned in 3.2.2, the star tracker can also be used as an optical camera. The optical sensor of the ST-200 has 1024x1024 pixels and a field-of-view with an angular size of 20°. For further details, see 3.2.2.

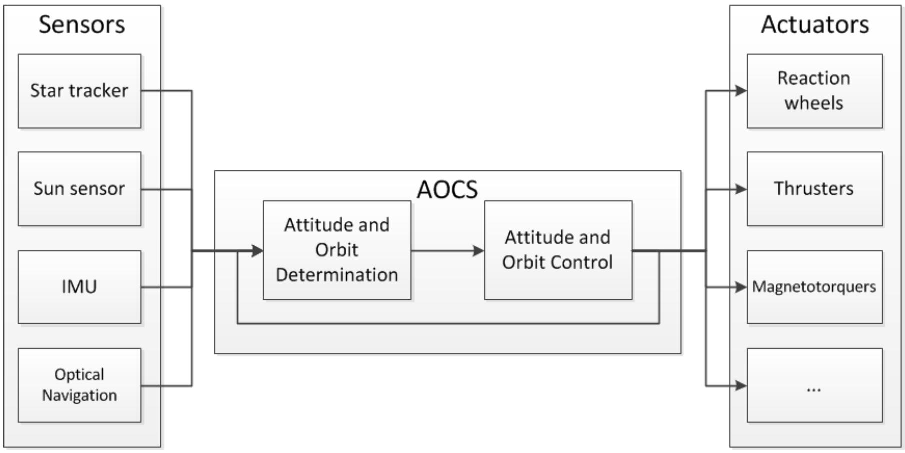

Figure 3.2: The system interface chart

3.3 Interface Chart

This section explains two interface charts: the system interface chart (see section 3.3.1) and the development interface chart (see 3.3.2). The first one describes the system and its basic interfaces between the hardware and software in the attitude and orbit determination and control system on a high-level. The second is the development interface chart, see section 3.3.2, which depicts the simulation environment to simulate the system.

3.3.1 System Interface Chart

The high-level attitude and orbit determination and control system is shown in figure 3.2. Basically, the goal of the system is to obtain measurements, process them and send commands to actuators. On the right in figure 3.2, the sensors are shown, as discussed in section 3.2. The outputs of the sensors are gathered and used for determining the orbit and attitude. A control algorithm uses the knowledge of the orbit and attitude to send commands to the actuators. A feedback is given from the controller to the attitude and orbit determination on which commands are being given, so these can be incorporated in the orbit and attitude estimation. The actuators, in combination with the environment, have an influence on the orbital dynamics. These translate in new measurements, which are begin processed in the satellite again. The focus of this research is on the attitude and orbit determination. The environment for the development of the attitude and orbit determination is presented in the next section, with some further details on what is required for simulating this system.

3.3.2 Development Interface Chart

The development interface chart, see figure 3.4 is based on the system interface chart. The starting point is like in the system interface chart, the measurements generation. These are generated from sensor models, which are reused for the filter algorithms. The measurements are being fed to the filter algorithms. As explained further, different implementations of the Kalman filter are chosen. Here the orbit and attitude determination takes place. The filter shares the sensor models with the measurements and the propagators with the orbit and attitude simulation. From the filter, the states are fed to the controller, which provides feedback to the filter and commands the actuators. The actuators feed their output to the orbit and attitude simulation, which updates the dynamics. The loop is closed with the input back to the measurements generation. The environment and ephemeris data are coupled to the loop by linking them to the sensor and propagation library.

In figure 3.4, the controller and actuator have been added. Normally, the actuators should be modelled similar to the sensors, resulting in an extra actuators library and a link to the filter. Since this research focuses on the navigation and the influence of the sensor inputs to the filters, the actuators and controller are not going to be modelled. Therefore, an open-loop system can be constructed, as can be seen in figure **??**. This means that the propagation of the states and the generation of measurements can be performed first. These inputs can then be reused by different filter implementations.

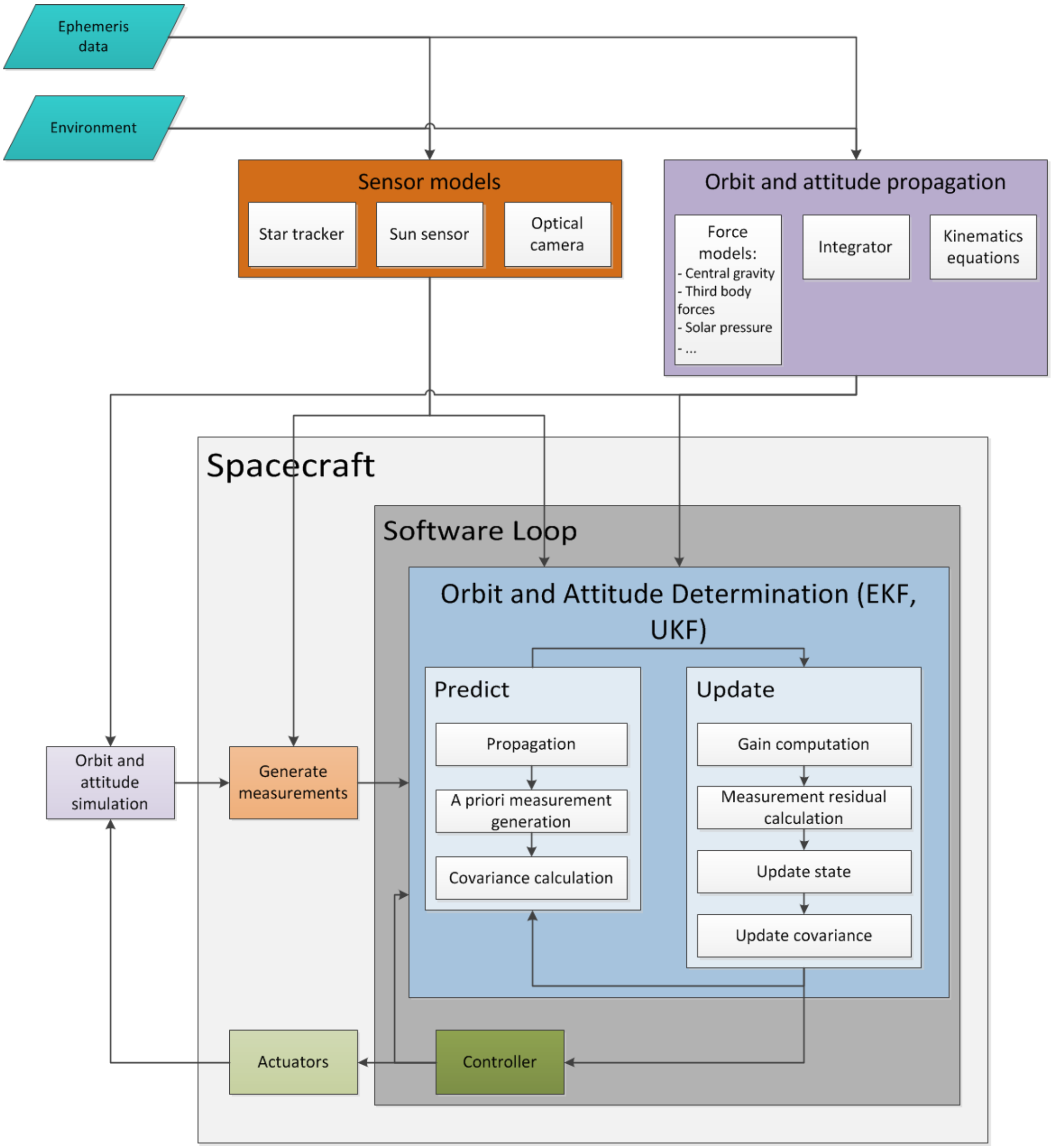

Figure 3.3: The closed loop development interface chart

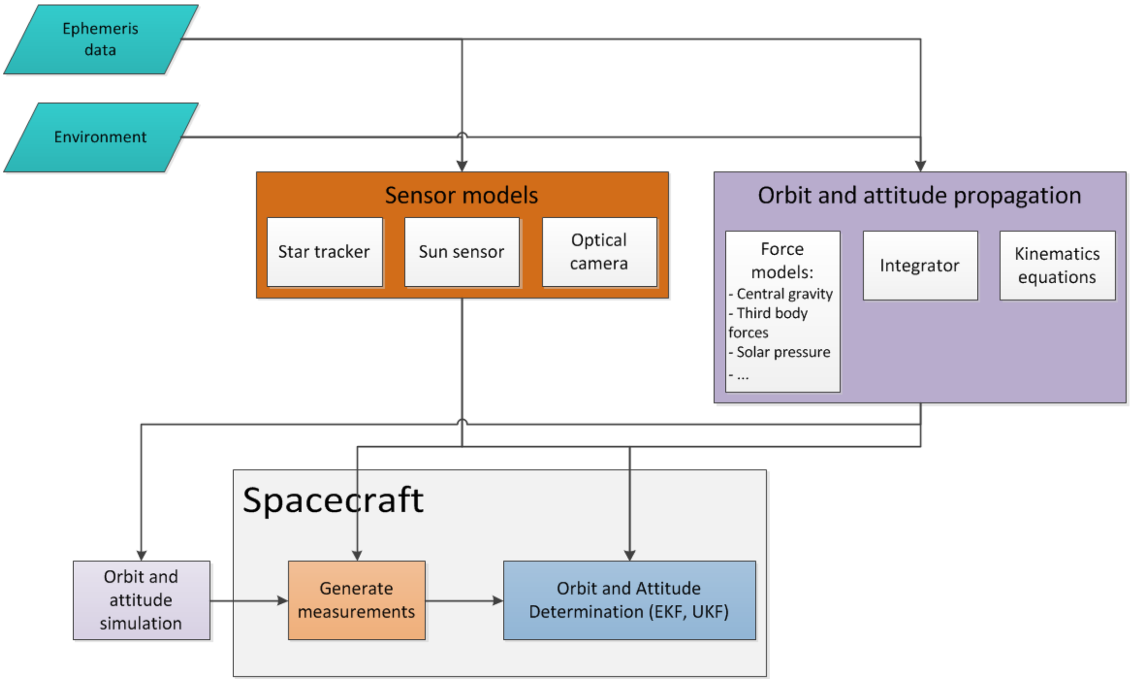

Figure 3.4: The simplified open loop development interface chart

Chapter 4

Functional Description

This chapter describes the theory behind the implementation. This starts with an explanation of the reference frames and astrodynamics in sections 4.1 and 4.2. Section 4.3 describes the sensor implementations. The used filter algorithms are explained in section 4.5. The stepwise implementation of these algorithms and functions is presented in chapter 5.

4.1 Reference Frames Description

THe reference frames which are being used are shortly explained.

4.1.1 Planet-centred Inertial Reference Frames

An inertial reference frame is a reference frame in which an object on which no forces are being applied moves in a straight line with constant velocity. Newton's laws are always applicable in inertial reference frames. Furthermore, inertial reference frames can always be transformed from one to another by a simple transformation, as they are not-rotating and have a constant velocity with respect to each other.

The definition of an inertial reference frame requires these frames to be defined over the complete universe. Therefore, they are Utopian, since it is currently impossible to define the complete universe. However, in reference frames centred in large bodies, like the Sun, Earth and other planets, motion approximates the motion as in a two-body system. They are called pseudo-inertial reference frames.

The most important for this research are the Earth-fixed and Moon-fixed inertial coordinate system. In the Earth-fixed system, the X-axis points to the vernal equinox and the Z-axis to the celestial pole. Currently, the vernal equinox coincides with the First Point of Aries, but moves with a period of 26,000 years. Therefore a version of the vernal equinox

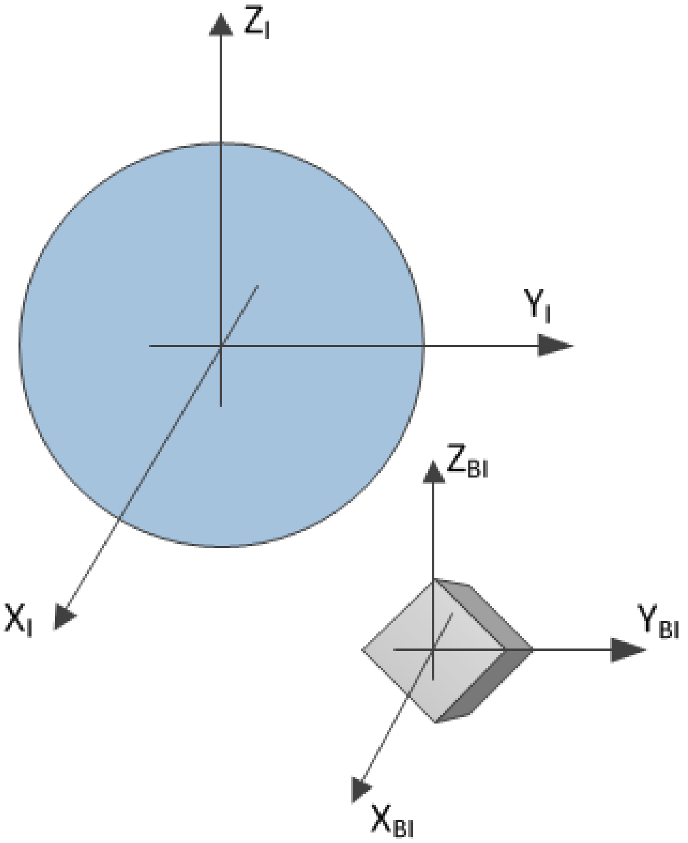

Figure 4.1: The spacecraft-centred inertial frame with respect to the planet-centred inertial frame

needs to be defined, like 1950, 2000 or true of date (TOD). The J2000 frame, *i.e.*, with the vernal equinox of 01/01/2000, is used during the research, because this is currently a commonly used one.

Since the relative positions of the Moon with respect to Earth is given in the J2000 frame, this is used for the Moon too, making changes in central body (from Earth to Moon and vice versa) easier.

Inertial reference frames are, due to the possibility of applying Newton's laws very convenient for the description of forces and accelerations.

4.1.2 Spacecraft-Centred Inertial Reference Frame

The spacecraft-centred inertial frame is defined here as the J2000 reference frame, but translated to the centre of the spacecraft. This is shown in figure **??**.

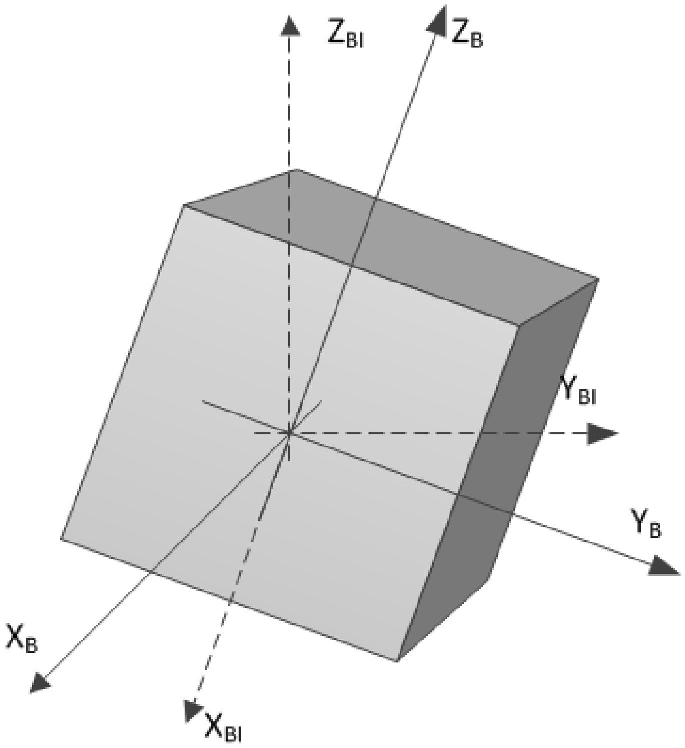

Figure 4.2: The spacecraft-fixed inertial frame with respect to the spacecraft-centred inertial frame

4.1.3 Spacecraft Body-Fixed Reference Frame

The spacecraft body-fixed reference frame is a non-inertial reference frame and is therefore not bound to Newton's laws. The attitude of the spacecraft is described by the rotation to go from the inertial to this spacecraft-centred frame. The axes are shown in 4.2.

4.1.4 Sensor Reference Frame

The sensor reference frame is the sensor-fixed frame. The Z-axis is the boresight axis of the sensor. The X- and Y-axis are chosen separately for each sensor. This frame is used to describe the measurements from the point of view of the sensor. The rotation between the sensor frame and the spacecraft-fixed frame normally remains constant. This reference frame is shown in figure 4.3

4.2 Astrodynamics

4.2.1 Force Models

The orbits under consideration are very high Earth orbits or lunar transfer orbits. Due to the distance to Earth, magnetic and aerodynamic forces can be neglected, as well as

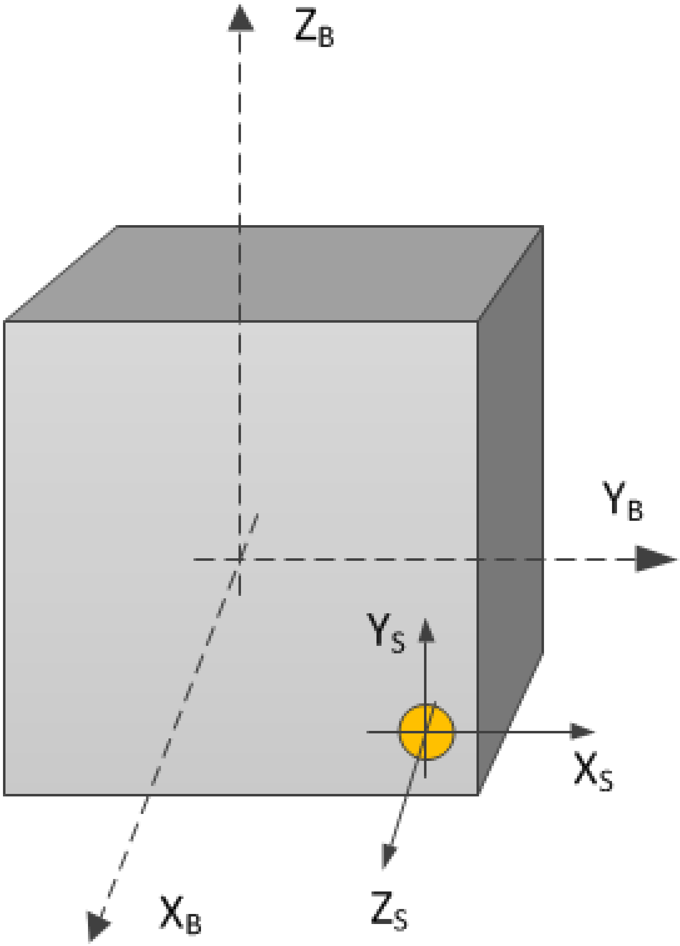

Figure 4.3: The sensor frame in the spacecraft-fixed frame

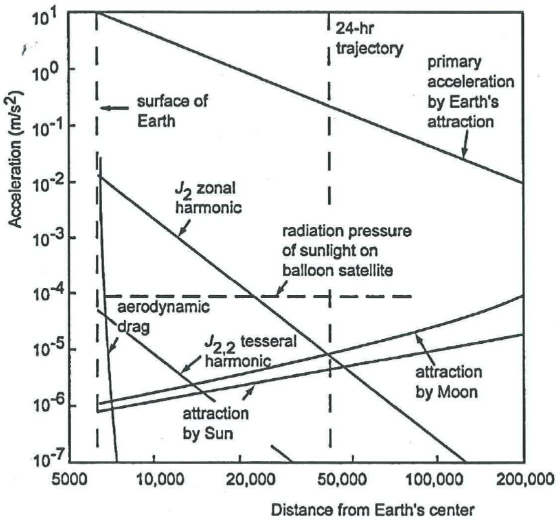

Figure 4.4: An overview of the disturbance forces with respect to the altitude of the satellite [19]

the influence of the gravitational disturbances. This can also be seen from figure 4.4 Therefore, the forces under consideration are the central gravity field of the main body (Earth or Moon), the gravity of third bodies (Sun, Moon or Earth, possibly others) and the solar pressure.

The only rotational disturbance would be the solar pressure and internal disturbances, like sloshing. However, due to the high density of nano-satellites and the limited surface, it is assumed these are negligible. Therefore, the attitude of the spacecraft is assumed to be unperturbed. If there should be any further disturbances, it is assumed these are small enough to be within the noise of the measurements and system.

The Universal Law of Gravitation

This law is stated as follows:

“Two particles attract each other with a force directly proportional to their masses and inversely proportional to the square of the distance between them.”

The vectorised form of this equation is [18]:

$$\boldsymbol{F} = -G\frac{m_1 m_2}{r^3}\boldsymbol{r} \tag{4.1}$$

For orbital mechanics, these equations are simplified by introducing $\mu = Gm_1$, the gravitational constant. m_1 is taken as mass of the major body, *i.e.*, Earth, Moon or Sun. The acceleration of a satellite under the influence of one of these bodies is the result of dividing the gravitational force by the mass of the satellite, m_2:

$$\boldsymbol{a} = -\frac{\mu}{r^3}\boldsymbol{r} \tag{4.2}$$

Body	Case1	Case2	Case3	Case4	Case5
Sun	$1.489e^{-5}$	$2.395e^{-4}$	$1.661e^{-4}$	$4.948e^{-6}$	$1.177e^{-7}$
Mercury	$1.087e^{-11}$	$1.748e^{-10}$	$1.213e^{-10}$	$3.6128e^{-12}$	$8.594e^{-14}$
Venus	$1.715e^{-9}$	$2.758e^{-8}$	$1.913e^{-8}$	$5.697e^{-10}$	$1.356e^{-11}$
Mars	$3.326e^{-11}$	$5.348e^{-10}$	$3.710e^{-10}$	$1.105e^{-11}$	$2.623e^{13}$
Jupiter	$1.919e^{-10}$	$3.086e^{-9}$	$2.141e^{-9}$	$6.375e^{-11}$	$1.517e^{-12}$
Saturn	$6.836e^{-12}$	$1.099e^{-10}$	$7.626e^{-11}$	$2.271e^{-12}$	$5.403e^{-14}$
Uranus	$1.087e^{-13}$	$1.747e^{-12}$	$1.212e^{-12}$	$3.610e^{-14}$	$8.590e^{-16}$
Neptune	$3.138e^{-14}$	$5.045e^{-13}$	$3.500e^{-13}$	$1.042e^{-14}$	$2.480e^{-16}$

Table 4.1: The ratio of disturbance force over central gravitational force for five cases. Case1: geostationary altitude; case 2: Earth orbit at 100,000 km; case 3: lunar orbit at 20,000 km; case 4: lunar orbit at 5,000 km; case 5: lunar orbit at 200 km

Sun and Planets

The gravitational influence of third bodies is determined by their position and their standard gravitational parameter. INCLUDE REFERENCE TO EPHEMERIS MODELS

To give an indication of the influence, one can use the following equation [18]:

$$\left(\frac{a_d}{a_m}\right)_{max} = 2\frac{m_d}{m_M}\left(\frac{r_s}{r_d}\right)^3 \tag{4.3}$$

Here m_d is the mass of the disturbing body, m_M is the mass of the main body, *i.e.*, the Earth or Moon, r_s the range of the satellite to the main body and r_d is the distance of the satellite to the disturbing body. The distance to the disturbing body is minimised to maximise the disturbance. Five cases have been used. The same as for the spherical harmonics: at geostationary altitude, Earth orbit at 100,000 km, lunar orbit at 20,000 km, 5,000 km and 200 km. The result can be found in table 4.1.

This maximum is at the minimal distance to the Earth, which is $(r_{planet} - 1)AU$, with r_{planet}, the semi-major axis of the planet.

The only force of importance is the Sun. Especially during the transfer, the influence of the Sun becomes important. Other planets have an influence of at most $2.758e^{-8}$ and can be neglected.

Solar Pressure

The solar pressure is computed by [19]:

$$\bar{f} = f_p\bar{e_s} = C_R\frac{WS}{Mc}\bar{e_s} \tag{4.4}$$

Here f_p is the size of the solar pressure, e_s is the unit vector pointing from the Sun to the satellite, W is the power density of the solar radiation in the proximity of the Earth ($1360W/m^2$), M is the mass of the satellite, S is the effective cross-sectional area, c is the speed of light in vacuum and C_r is the satellite's reflectivity.

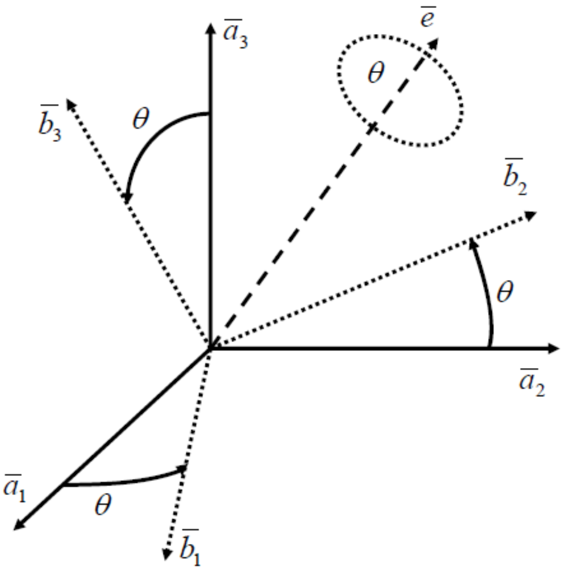

Figure 4.5: Euler's eigenaxis rotation [2]

4.2.2 Translational Equations of Motion

The Cartesian coordinate system is the reference system for the research. The dynamics and kinematics in a Cartesian coordinate system are described by:

$$\boldsymbol{v}(t) = \boldsymbol{v}_0 + \int_{t_0}^{t} \boldsymbol{a}(t)dt \tag{4.5}$$

$$\boldsymbol{r}(t) = \boldsymbol{r}_0 + \int_{t_0}^{t} \boldsymbol{v}(t)dt \tag{4.6}$$

In combination with the second law of Newton, see equation 4.2, the motion of a body can be described.

4.2.3 Attitude State Variables

To get to a good understanding of quaternions, it is necessary to first understand the Eular eigenaxis rotation.

Euler Angles

Each (right-handed) coordinate system can be rotated to any other using one rotation. This rotation is called the Euler eigenaxis rotation. The Euler eigenaxis, $\boldsymbol{e} = (e_1, e_2, e_3)$, is a unit vector and the axis around which the coordinate system needs to rotate. The Euler angle, θ, is the total rotation around this axis. This derivation is required to come to the definition of quaternions.

The eigenaxis shown in figure 4.5 can be described in two reference frames, $\boldsymbol{A}(\boldsymbol{a_1}, \boldsymbol{a_2}, \boldsymbol{a_3})$ and $\boldsymbol{B}(\boldsymbol{b_1}, \boldsymbol{b_2}, \boldsymbol{b_3})$.

$$\begin{aligned} \boldsymbol{e} &= e_1\boldsymbol{a_1} + e_2\boldsymbol{a_2} + e_3\boldsymbol{a_3} \qquad (4.7) \\ &= e_1\boldsymbol{b_1} + e_2\boldsymbol{b_2} + e_3\boldsymbol{b_3} \qquad (4.8) \end{aligned}$$

By rearranging, the derivation of the direction cosine matrix, $\boldsymbol{C}$, can be initiated.

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{a_1} & \boldsymbol{a_2} & \boldsymbol{a_3} \end{bmatrix} \begin{bmatrix} \boldsymbol{b_1} \\ \boldsymbol{b_2} \\ \boldsymbol{b_3} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \qquad (4.9)$$

Now the elements in the direction cosine matrix need to be derived. Three rotations are required. The first rotation rotates frame A with rotation matrix $\boldsymbol{R}$ in such a way that $\boldsymbol{a_1}$ coincides with $\boldsymbol{e}$, forming the new frame A', as can be seen in figure 4.6. The second rotation is the rotation of A' around $\boldsymbol{e} = \boldsymbol{a_1'}$ through θ and form A''. This rotation is the same rotation to go from A to B. Due to this 'parallel' rotation and the common rotation axis $\boldsymbol{e}$, the rotation from A to A' is the same as a rotation from B to A'', both with the rotation matrix R. Use the inverse of the rotation from B to A'' as the third rotation with rotation matrix R^T. Now the total rotation from A to B can be described by:

$$\begin{bmatrix} \boldsymbol{b_1} \\ \boldsymbol{b_2} \\ \boldsymbol{b_3} \end{bmatrix} = \boldsymbol{R}^T\boldsymbol{C_1}(\theta)\boldsymbol{R} \begin{bmatrix} \boldsymbol{a_1} \\ \boldsymbol{a_2} \\ \boldsymbol{a_3} \end{bmatrix} \qquad (4.10)$$

with $C_1(\theta)$ as the rotation matrix of the second rotation, *i.e.*, a rotation around $\boldsymbol{a_1'}$ with angle θ. The three rotation matrices are:

$$\boldsymbol{R} = \begin{bmatrix} e_1 & e_2 & e_3 \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \qquad (4.11)$$

$$\boldsymbol{C_1(\theta)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \qquad (4.12)$$

$$\boldsymbol{R^T} = \begin{bmatrix} e_1 & R_{21} & R_{31} \\ e_2 & R_{22} & R_{32} \\ e_3 & R_{23} & R_{33} \end{bmatrix} \qquad (4.13)$$

To solve for the unknown parameters R_{ij}, two properties of direction cosine matrices can be used. The first one is the orthonormality:

$$\boldsymbol{R_T R} = \begin{bmatrix} e_1 & R_{21} & R_{31} \\ e_2 & R_{22} & R_{32} \\ e_3 & R_{23} & R_{33} \end{bmatrix} \begin{bmatrix} e_1 & e_2 & e_3 \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \boldsymbol{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (4.14)$$

The second is that the direction cosine matrices can be expressed by their co-factors:

$$e_1 = R_{22}R_{33} - R_{23}R_{32}; \qquad e_2 = R_{23}R_{31} - R21R33; \qquad e_3 = R_{21}R_{32} - R_{22}R_{31} \quad (4.15)$$

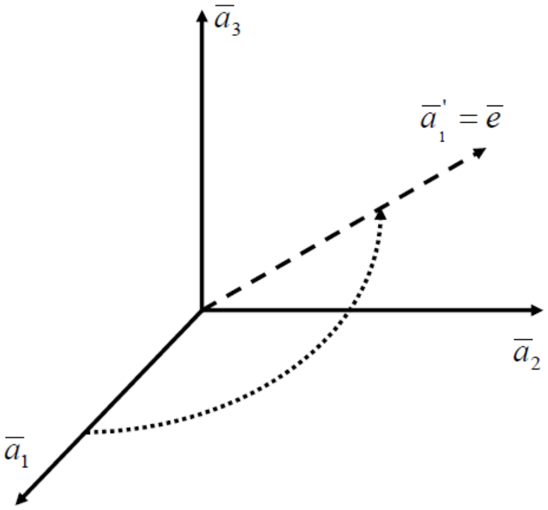

Figure 4.6: The rotation of $\boldsymbol{a_1}, \boldsymbol{a_2}, \boldsymbol{a_3}$

Using these two properties and equation 4.10, the direction cosine can be calculated for the Euler eigenaxis rotation:

$$\boldsymbol{C} = \begin{bmatrix} \cos\theta + e_1^2(1-\cos\theta) & e_1e_2(1-\cos\theta) + e_3\sin\theta & e_1e_3(1-\cos\theta) - e_2 sin\theta \\ e_2e_1(1-\cos\theta) - e_3\sin\theta & \cos\theta + e_2^2(1-\cos\theta) & e_2e_3(1-\cos\theta) + e_1 sin\theta \\ e_3e_1(1-\cos\theta) + e_2 sin\theta & e_3e_2(1-\cos\theta) - e_1 sin\theta & \cos\theta + e_3^2(1-\cos\theta) \end{bmatrix} \tag{4.16}$$

Or in a more compact form

$$\boldsymbol{C} = \boldsymbol{I}\cos\theta + (1+\cos\theta)\boldsymbol{e}\boldsymbol{e}^T - \boldsymbol{E}\sin\theta \tag{4.17}$$

With I as the identity matrix and:

$$\boldsymbol{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \tag{4.18}$$

$$\boldsymbol{E} = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \tag{4.19}$$

In this relation e_1, e_2 and e_3 are related by $e_1^2 + e_2^2 + e_3^2 = 1$. θ can be found from a given direction cosine matrix by:

$$\cos\theta = \frac{1}{2}(C_{11} + C_{22} + C_{33} - 1) \tag{4.20}$$

Quaternions

To avoid singularities a rotation can also be described using four Euler parameters or quaternions. The definition of the four quaternions is based on Euler's eigenaxis rotations, see Subsection **??** [2].

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \sin(\theta/2) \tag{4.21}$$

$$q_4 = \cos(\theta/2) \tag{4.22}$$

Due to their relation to e_1, e_2 and e_3 the quaternions depend on each other.

$$\boldsymbol{q}^T\boldsymbol{q} + q_4^2 = q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \tag{4.23}$$

The direction cosine matrix in terms of quaternions can be derived by substituting equations 4.21 and 4.22 in theh direction cosine matrix for Euler angles, see equation 4.16. As an example, the three first elements are derived:

$$\begin{aligned} C_{11} &= \cos\theta + e_1^2(1-\cos\theta) = 1 - 2\sin^2\frac{\theta}{2} + 2e_1^2\sin^2\frac{\theta}{2} \\ &= 1 - 2(1-e_1^2)\sin^2\frac{\theta}{2} = 1 - 2(e_2^2+e_3^2)\sin^2\frac{\theta}{2} \\ &= 1 - 2(q_2^2+q_3^2) \\ C_{12} &= e_1e_2(1-\cos\theta) + e_3\sin\theta = 2e_1e_2\sin^2\frac{\theta}{2} + 2e_3\sin\frac{\theta}{2}\cos\frac{\theta}{2} \\ &= 2\left(e_1\sin\frac{\theta}{2}e_2\sin\frac{\theta}{2} + e_3\sin\frac{\theta}{2}\cos\frac{\theta}{2}\right) \\ &= 2(q_1q_2 + q_3q_4) \\ C_{13} &= e_1e_3(1-\cos\theta) - e_2\sin\theta = 2e_1e_3\sin^2\frac{\theta}{2} - 2e_2\sin\frac{\theta}{2}\cos\frac{\theta}{2} \\ &= 2\left(e_1\sin\frac{\theta}{2}e_3\sin\frac{\theta}{2} - e_2\sin\frac{\theta}{2}\cos\frac{\theta}{2}\right) \\ &= 2(q_1q_3 - q_2q_4) \end{aligned}$$

This can be continued for the other elements. This results in the direction cosine matrix:

$$\boldsymbol{C}^{B/A} = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2+q_3q_4) & 2(q_1q_3-q_2q_4) \\ 2(q_1q_2-q_3q_4) & 1-2(q_1^2+q_3^2) & 2(q_2q_3+q_1q_4) \\ 2(q_1q_3+q_2q_4) & 2(q_2q_3-q_1q_4) & 1-2(q_1^2+q_2^2) \end{bmatrix} \tag{4.24}$$

This can be rewritten in more compact form as:

$$\boldsymbol{C} = (q_4^2 - \boldsymbol{q}^T\boldsymbol{q})I + 2\boldsymbol{q}\boldsymbol{q}^T - 2q_4\boldsymbol{Q} \tag{4.25}$$

with $\boldsymbol{Q}$ defined as:

$$\boldsymbol{Q} = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{4.26}$$

A more compact form of this equation is

$$\boldsymbol{C} = \boldsymbol{I}\cos\theta + (1+\cos\theta)\boldsymbol{e}\boldsymbol{e}^T - \boldsymbol{E}\sin\theta \tag{4.27}$$

With I as the unity matrix and:

$$\boldsymbol{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \tag{4.28}$$

$$\boldsymbol{E} = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \tag{4.29}$$

In this relation e_1, e_2 and e_3 are related by $e_1^2 + e_2^2 + e_3^2 = 1$. θ can be found from a given direction cosine matrix by:

$$\cos\theta = \frac{1}{2}(C_{11} + C_{22} + C_{33} - 1) \tag{4.30}$$

4.2.4 Rotational Equations of Motion

To get to the equations of motion of quaternions, first the equations of motion based on a direction cosine matrix are explained. Based on those, the dynamics of the Euler Eigenaxis rotation are explained, to finish with the equations of motion of quaternions.

Direction Cosine Matrix

Assume two right-hand sets of three orthogonal unit vectors, $\{\boldsymbol{a_1}, \boldsymbol{a_2}, \boldsymbol{a_3}\}$ and $\{\boldsymbol{b_1}, \boldsymbol{b_2}, \boldsymbol{b_3}\}$. The relation between these two coordinate systems is given by:

$$\begin{bmatrix} \boldsymbol{b_1} \\ \boldsymbol{b_2} \\ \boldsymbol{b_3} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{bmatrix} \boldsymbol{a_1} \\ \boldsymbol{a_2} \\ \boldsymbol{a_3} \end{bmatrix} = \boldsymbol{C}^{B/A} \begin{bmatrix} \boldsymbol{a_1} \\ \boldsymbol{a_2} \\ \boldsymbol{a_3} \end{bmatrix} \tag{4.31}$$

Here $\boldsymbol{C}_{ij} = \boldsymbol{b_i} \cdot \boldsymbol{a_j}$, which is the cosine of the angle between $\boldsymbol{a_j}$ and $\boldsymbol{b_i}$. Therefore $\boldsymbol{C}_{ij}$ is called the direction cosine matrix. The two sets of orthogonal vectors and the direction cosines are shown by figure 4.7.

To construct the kinematic differential equation, the angular velocity, $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$, and the time derivative of $\boldsymbol{C}$ are required. The time derivatives of $\dot{\boldsymbol{C}}$ is described by:

$$\dot{\boldsymbol{C}} = \begin{bmatrix} \dot{C}_{11} & \dot{C}_{12} & \dot{C}_{13} \\ \dot{C}_{21} & \dot{C}_{22} & \dot{C}_{23} \\ \dot{C}_{31} & \dot{C}_{32} & \dot{C}_{33} \end{bmatrix} \tag{4.32}$$

The elements in this matrix are described by:

$$\dot{C}_{11} = \omega_3 C_{21} - \omega_2 C_{31}; \quad \dot{C}_{12} = \omega_3 C_{22} - \omega_2 C_{32}; \quad \dot{C}_{13} = \omega_3 C_{23} - \omega_2 C_{33}; \tag{4.33}$$

$$\dot{C}_{21} = \omega_1 C_{31} - \omega_3 C_{11}; \quad \dot{C}_{22} = \omega_1 C_{32} - \omega_3 C_{12}; \quad \dot{C}_{23} = \omega_1 C_{33} - \omega_3 C_{13}; \tag{4.34}$$

$$\dot{C}_{31} = \omega_2 C_{11} - \omega_1 C_{21}; \quad \dot{C}_{32} = \omega_2 C_{12} - \omega_1 C_{22}; \quad \dot{C}_{33} = \omega_2 C_{13} - \omega_1 C_{23}; \tag{4.35}$$

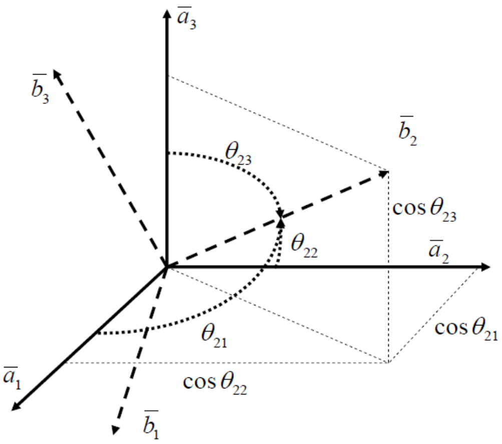

Figure 4.7: Reference frames and direction cosines [2]

The kinematic differential equation then becomes:

$$(\dot{\boldsymbol{C}}^T - \boldsymbol{C}^T\boldsymbol{\Omega})\begin{bmatrix}\boldsymbol{b_1}\\ \boldsymbol{b_2}\\ \boldsymbol{b_3}\end{bmatrix} = 0; \qquad \boldsymbol{\Omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{4.36}$$

Or in another form:

$$\dot{\boldsymbol{C}}^T - \boldsymbol{C}^T\boldsymbol{\Omega} = 0; \qquad \dot{\boldsymbol{C}} + \boldsymbol{\Omega}\boldsymbol{C} = 0; \qquad \boldsymbol{\Omega}^T = -\boldsymbol{\Omega} \tag{4.37}$$

Euler Eigenaxis Rotation

For the Euler eigenaxis rotation, the direction cosine matrix, $\boldsymbol{C}$, is expressed by:

$$\boldsymbol{C} = \boldsymbol{I}\cos\theta + (1+\cos\theta)\boldsymbol{e}\boldsymbol{e}^T - \boldsymbol{E}\sin\theta \tag{4.38}$$

With I as the identity matrix and:

$$\boldsymbol{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \tag{4.39}$$

$$\boldsymbol{E} = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \tag{4.40}$$

Based on this input and equation 4.37, $\boldsymbol{\Omega}$ can be expressed as a function of θ and $\boldsymbol{e}$ [2].

$$\boldsymbol{\Omega} = \dot{\theta}\boldsymbol{E} - (1-\cos\theta)(\boldsymbol{E}\dot{\boldsymbol{e}})^* + \sin\theta\dot{\boldsymbol{E}} \tag{4.41}$$

with the superscript $*$ representing the skew symmetric form. When rewriting the equation to vector form, this becomes:

$$\boldsymbol{\omega} = \dot{\theta}\boldsymbol{e} - (1-\cos\theta)\boldsymbol{E}\dot{\boldsymbol{e}} + \sin\theta\dot{\boldsymbol{e}} \tag{4.42}$$

From this equation, the following relation can be derived:

$$\boldsymbol{e}^T\boldsymbol{\omega} = \dot{\theta}\boldsymbol{e}^T\boldsymbol{e} - (1-\cos\theta)\boldsymbol{e}^T\boldsymbol{E}\dot{\boldsymbol{e}} + \sin\theta\boldsymbol{e}^T\dot{\boldsymbol{e}} = \dot{\theta} \tag{4.43}$$

After a derivation, the time derivative of the Euler axis can be given [2]:

$$\dot{\boldsymbol{e}} = \frac{1}{2}\left(\boldsymbol{E}\boldsymbol{\omega} - \frac{\cos\frac{\theta}{2}}{\sin\frac{\theta}{2}}\boldsymbol{E}\boldsymbol{E}\boldsymbol{\omega}\right) \tag{4.44}$$

Quaternions

For representing the attitude, quaternions are chosen. Quaternions have the advantage of avoiding singularities and relative ease of use. A quaternion represents an Euler eigenaxis rotation. This is based on the fact that each rotation between two orientations can be expressed by a single rotation around its eigenaxis, as explained before. The attitude quaternion is here defined as:

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{4.45}$$

q_1, q_2 and q_3 are the vector part of the quaternion with (e_x, e_y, e_z) representing the rotation axis. q_4 is the scalar part, representing the size of the rotation:

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} e_x \\ e_y \\ e_z \end{bmatrix} \sin\frac{\theta}{2} \tag{4.46}$$

$$q_4 = \cos\frac{\theta}{2} \tag{4.47}$$

The kinematic differential equation in terms of quaternions are based on those for direction cosine matrices [2]:

$$\dot{\boldsymbol{C}} + \boldsymbol{\Omega}\boldsymbol{C} = 0; \qquad \boldsymbol{\Omega} = -\dot{\boldsymbol{C}}\boldsymbol{C}^T$$

Now ω_1, ω_2 and ω_3 can be expressed as:

$$\omega_1 = \dot{C}_{21}C_{31} + \dot{C}_{22}C_{32} + \dot{C}_{23}C_{33} \tag{4.48}$$

$$\omega_2 = \dot{C}_{31}C_{11} + \dot{C}_{32}C_{12} + \dot{C}_{33}C_{13} \tag{4.49}$$

$$\omega_3 = \dot{C}_{11}C_{21} + \dot{C}_{12}C_{22} + \dot{C}_{13}C_{23} \tag{4.50}$$

By substituting the direction cosines in terms of quaternions, see equation 4.24, this becomes:

$$\omega_1 = 2(\dot{q}_1q_4 + \dot{q}_2q_3 - \dot{q}_3q_2 - \dot{q}_4q_1) \tag{4.51}$$

$$\omega_2 = 2(\dot{q}_2q_4 + \dot{q}_3q_1 - \dot{q}_1q_3 - \dot{q}_4q_2) \tag{4.52}$$

$$\omega_3 = 2(\dot{q}_3q_4 + \dot{q}_1q_2 - \dot{q}_2q_1 - \dot{q}_4q_3) \tag{4.53}$$

To obtain a fourth equation, differentiate the quaternion constraint:

$$0 = 2(\dot{q}_1 q_1 + \dot{q}_2 q_2 + \dot{q}_3 q_3 + \dot{q}_4 q_4) \tag{4.54}$$

Rewrite these in matrix form:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix} = 2 \begin{bmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \\ q_1 & q_2 & q_3 & q_4 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} \tag{4.55}$$

The matrix is orthonormal, so it can be solved with a matrix based on quaternions and on $\boldsymbol{\omega}$ as:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix} \tag{4.56}$$

$$= \frac{1}{2} \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix} \tag{4.57}$$

In the more compact form, this becomes:

$$\dot{\boldsymbol{q}} = \frac{1}{2}(q_4 \boldsymbol{\omega} - \boldsymbol{\omega} \times \boldsymbol{q}) = \frac{1}{2}(q_4 \boldsymbol{\omega} + \boldsymbol{q} \times \boldsymbol{\omega}) \tag{4.58}$$

$$\dot{q}_4 = -\frac{1}{2}\boldsymbol{\omega}^T \boldsymbol{q} = -\frac{1}{2}\boldsymbol{q}^T \boldsymbol{\omega} \tag{4.59}$$

4.3 Sensor Measurements

This chapter describes the functional implementation of the sensors, as they are implemented in the test environment. This includes the measurement equations and modelling of errors. The unit tests of the implementation can be found in **??**. The description of the reference frames can be found in section 4.1.

4.3.1 Planet detection

Planet detection is used for two bodies: Earth and Moon. The principle consists of making an image of a body with an optical camera and processing the image. From the image, three observations are derived: the elevation, the azimuth and the angular size of the body. The first two describe the position of the body in the sensor reference frame. The angular size contains information on the distance between the body and the sensor, and therefore also the spacecraft.

The centre of the coordinate system for the position of the spacecraft is either Earth or the Moon. To generate the measurements of the body, which is not located in the centre of the coordinate system, an ephemeris file is required to obtain the position of the non-centre body with respect to the body in the centre of the system.

First, the azimuth and elevation of the central and non-central body are treated. Then the angular size measurement algorithm and its errors are explained.

Azimuth and elevation

The only input for these measurements is the position of the planet in the sensor reference frame, $\boldsymbol{r}_S$. To obtain this position, the position and attitude of the spacecraft and the sensor on the spacecraft need to be known. For the non-central body, also the ephermis data of the non-central body with respect to the central body are required. The resulting transformation is:

$$\boldsymbol{r}_S = \boldsymbol{T}_{SB}\boldsymbol{T}_{BI}(-(\boldsymbol{r}_I - \boldsymbol{r}_{TC})) \tag{4.60}$$

Here, $\boldsymbol{r}_I$ is the inertial position of the spacecraft, which is part of the state. The matrices $\boldsymbol{T}_{SB}$ and $\boldsymbol{T}_{BI}$ are the transformation matrices from sensor (S) to body (B) frame and from body to inertial (I) frame. The first one is known from the orientation of the sensor on the spacecraft. The latter is to be taken from the attitude in the state. The vector which needs to be rotated is the negative of the position of the spacecraft minus $\boldsymbol{r}_{TC}$, the vector from the central body to the third body.

The azimuth, α, and elevation,ϵ can then be calculated by:

$$z_{X1} = \alpha = \text{ atan}\, 2(x_S, z_S) \tag{4.61}$$

$$z_{X2} = \epsilon = \text{ asin}(\frac{y_S}{r_S}) \tag{4.62}$$

with z_X representing the measurements of body X, which is Earth or Moon. In the equations above, no error is present in the measurements. Since the measurements are the result of post-processing images of the optical camera, it is assumed the noise is a white noise around the observation axis, the Z-axis, of the sensor. This can easily be modelled with quaternion rotations. An angle, ψ_e is randomly drawn from a uniform distribution between 0 and 360°. This angle is used to generate a unit vector in the XY-plane. This unit vector is used for a rotation. The size of the rotation, η is a random value from a normal distribution with the properties of the white noise. This is translated in a quaternion by:

$$\boldsymbol{q} = \begin{bmatrix} \cos(\psi_e) * \sin\left(\frac{\eta}{2}\right) \\ \sin(\psi_e) * \sin\left(\frac{\eta}{2}\right) \\ 0 \\ \cos\left(\frac{\eta}{2}\right) \end{bmatrix} \tag{4.63}$$

The first three elements are the rotation axis, where the scalar part of the quaternion, the fourth element, the size of the rotation describes.

The error description can only be used for relatively small field of views, since this rotation is around the Z-axis of the sensor. The further the observed body is from the Z-axis, the lower the error becomes. The optical camera has a half-cone angle of 20°. The reduction of the error at that point is about 6%. This is assumed to be acceptable, since the errors on the optical measurements are not well known, since this is depending on the

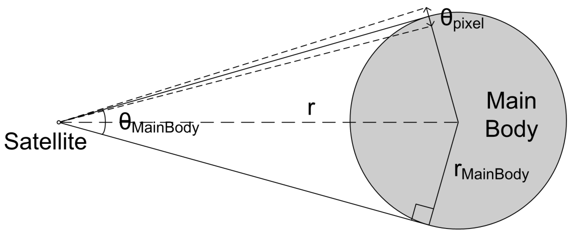

Figure 4.8: Trigonometry of a satellite imaging Earth [1]

pre-processing of the optical image. Therefore, the error on the error description is most likely much higher than this. The size of the noise is described after the measurements of the angular size of the central body.

Angular Size Measurement

Disc measurement is based on making an image of Earth or the Moon and by measuring the angular size, determine the distance to the main body. This technique is explained in [1] and is still being researched. By applying trigonometry on figure 4.8, one can estimate the distance to Earth by:

$$r = \frac{r_S}{\sin \frac{\theta_X}{2}} \tag{4.64}$$

with the subscript X referring to Earth or Moon. From this equation, the angular size can be derived as:

$$z_{X3} = \theta = 2\,\mathrm{asin}\left(\frac{R_X}{r_S}\right) \tag{4.65}$$

The measurement, as described here, is only valid if the complete main body is visible. However, by having only part of the Earth in view, the angular size can still be determined by using the visible arc. Different visibilities of Earth can be seen in figure 4.9.

The error on this measurement is depending on the altitude. By using only two pixels on opposing locations of the visible disc, the error is computed from figure ref-fig:discMeasurement:

$$\epsilon_r = r_S\left(\frac{1}{\sin\left(\frac{\theta_X - \theta_P}{2}\right)} - \frac{1}{\sin\left(\frac{\theta_X + \theta_P}{2}\right)}\right) \tag{4.66}$$

with the θ_P representing the angular size of one pixel. The performance can be improved by vectorising the disc using pixel averaging. This significantly lowers the errors on the measurement. By correlating the error over the number of pixels in view, the total error is being determined by:

$$\frac{1}{\frac{1}{\epsilon^2}} = \frac{1}{\frac{1}{\epsilon_1^2} + \frac{1}{\epsilon_2^2} + ...} \tag{4.67}$$

Here ϵ_1, ϵ_2, etc are the errors of the single pixels.

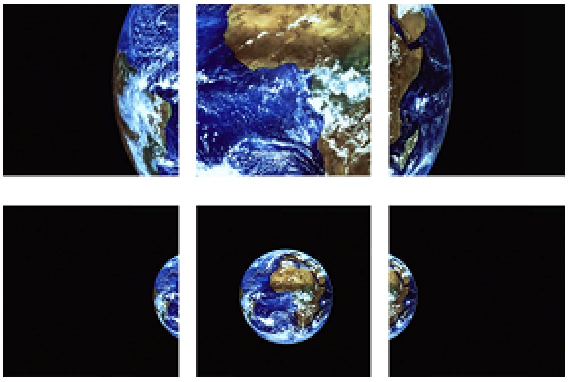

Figure 4.9: Visibility of the Earth from a low (upper series) and high (lower series) altitude as seen by the imager

4.3.2 Sun Sensor

The measurements from the Sun sensor are assumed to be an elevation and azimuth after processing. This is a simplification, since normally, the measurements are the voltages from the solar cells on the sensor. These are then converted to the position of the Sun. Based on these voltages, it is easy to obtain a certain attitude with respect to the Sun.

Since the complete attitude and position acquisition is required rather than a pure attitude controller, the choice has been made to simplify the measurements to elevation and azimuth in a common design with the planet measurements. Therefore, the implementation of the Sun sensor is the same as the azimuth and elevation measurements of the planet measurements. The main difference is the much lower accuracy. Therefore, the influence of the Sun sensor in the complete system is expected to be relatively low, compared to the planet measurements and the star tracker. However, it remains an important part of the spacecraft safe mode design.

4.3.3 Star tracker

The star tracker is a camera observing the sky. By combining its observations with a star catalogue, it generates a quaternion providing the orientation of the sensor in the inertial frame. This quaternion is the result of a quaternion multiplication of the attitude quaternion of the spacecraft, $\boldsymbol{q}$, and the quaternion describing the position of the sensor on the spacecraft, $\boldsymbol{s}$,.

$$\begin{bmatrix} z_{STR1} \\ z_{STR2} \\ z_{STR3} \\ z_{STR4} \end{bmatrix} = \boldsymbol{q} \cdot \boldsymbol{s} \tag{4.68}$$

EXPLANATION OF STR ERROR

4.3.4 IMU

The accelerometer measures all forces which are acting on the outside of the spacecraft. Forces which act on the complete body, like gravity, are not measured, since the accelerometer will be susceptible to the same acceleration as the rest of the spacecraft. Since the spacecraft is far away from Earth, there is no aerodynamic drag. The only force acting on the outside of the spacecraft is the solar pressure. The benefit of measuring this is limited. Therefore, the accelerometer is not modelled.

The gyroscope measures angular accelerations. On the short term, there are no significant forces acting on the spacecraft. Therefore, it is assumed that there are no rotational accelerations. From that point of view, there is no need for a gyroscope.

4.4 Integration Methods

For the integration in the filters, the Euler integration is one of the fastest methods. The Runge-Kutta methods are an extension to the Euler integration as they use a weighted average of multiple Euler integration. The Runge-Kutta 4 is commonly used for integration, due to thelow computational requirements and the reasonable accuracy. Below, the Euler integration (see section 4.4.1), the general Runge-Kutta methods (see 4.4.2) and the Runge-Kutta 4 integration (see 4.4.3) are explained.

4.4.1 Euler Integration

The Euler integration is a very basic stepwise integration method.

Assume the initial value problem:

$$y' = f(t, y) \tag{4.69}$$

$$y(t_0) = y_0 \tag{4.70}$$

Then the Euler integration is defined as:

$$t_{n+1} = t_n + hy_{n+1} = \qquad y_n + y_n h \tag{4.71}$$

4.4.2 General Runge-Kutta Method

The Runge-Kutta methods compute the step in a stepwise integration process [6]. With the same initial value problem as for the Euler integration, the stepwise Runge-Kutta integration is defined by:

$$y_{n+1} = y_n + \sum_{i=1}^{s} b_i k_i \tag{4.72}$$

Here [6]:

$$k_1 = hf(t_n, y_n) \tag{4.73}$$

$$k_2 = hf(t_n + c_2h, y_n + a_{21}k_1) \tag{4.74}$$

$$k_3 = hf(t_n + c_3h, y_n + a_31k_1 + a_32k_2) \tag{4.75}$$

$$... \tag{4.76}$$

$$k_1 = hf(t_n + c_sh, y_n + a_{s1}k_2 + ... + a_{s,s-1}k_{s-1} \tag{4.77}$$

To specify a certain method, an integer s is required and the coefficients $a_{ij}(1 \leq j < i \leq s)$, $b_i(i = 1, 2, ...s)$ and $c_i(i = 2, 3, ..., s)$. These coefficients are arranged in a Butcher tableau, as seen in table 4.2.

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
$...$		$...$			
c_s	a_{s1}	a_{s2}	$...$	$a_{s,s-1}$	
	b_1	b_2	$...$	b_{s-1}	b_s

Table 4.2: The Butcher tableau [6]

The time step does not need to be fixed. Depending on the expected amount of non-linearity of the system, the time step can be reduced or increased. For example, in an elliptical orbit, the non-linearities are higher with a higher velocity. This means that the time step needs to be lowered around the perigee and increased around the apogee. However, decreasing the time step too much can make errors due to round-off larger than the integration errors. A way of adjusting the time step is by, for example, using the error estimate of the Cash-Karp method.

The difference between the fourth- and fifth-order integration is determined.

$$\delta = \frac{|y_{5th} - y_{4th}|}{t_{n+1} - t_n} \tag{4.78}$$

If δ is larger than a preset error, the step size needs to be reduced. If δ is much smaller (*e.g.* 3 times smaller) than the allowed error, the time step needs to be increased to reduce the computational load. This way, the computational load of the integration method can be optimised and the available computational load can be used optimally.

4.4.3 Runge-Kutta 4

One of the most known is the Runge-Kutta 4 (RK4) integrator. The integration is described by [6]:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{4.79}$$

$$t_{n+1} = t_n + h \tag{4.80}$$

$$k_1 = hf(t_n, y_n) \tag{4.81}$$

$$k_2 = hf(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \tag{4.82}$$

$$k_3 = hf(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2) \tag{4.83}$$

$$k_4 = hf(t_n + h, y_n + k_3) \tag{4.84}$$

Here y_{n+1} is the approximation of $y(t_{n+1})$.

4.5 Filter Algorithms

The linear Kalman filter is an optimal estimator for systems with linear models for both the process and the measurements. To be able to work with non-linear systems, several variations have been developed on the linear Kalman filter. The most commonly used in the field of navigation is the extended Kalman filter (EKF), which uses Taylor series expansions to linearise the system. Due to its well-known behaviour, the EKF is used as the baseline and is explained in section 4.5.2. Lately, the unscented Kalman filter shows great promise in solving non-linear problems. It generates a distribution based on the posterior estimate and its covariance and generates a new covariance by assuming a Gaussian noise over this distribution [9]. Due to the limited number of points in the distribution, the computational load is lower than pure Monte Carlo based systems. Since the UKF should be able to handle non-linear systems with larger errors, it is included in this research and explained in 4.5.3 [11]. First, a short introduction to general Kalman filtering is given in 4.5.1. This section also describes the nomenclature for the rest of the chapter.

4.5.1 Introduction to Kalman Filtering

Kalman filters are recursive filters. This means they use information from previous for estimating a new state [10]. For Kalman filters, this is not only limited to the reuse of the state, but also to include the previous noise in the shape of a covariance matrix. Finding the balance between reusing the previous step and putting more emphasis on the current measurements, is the main focus of tuning the filter.

The system is described by a process and measurement model:

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \boldsymbol{w}_{k-1}) \tag{4.85}$$

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k, \boldsymbol{v}_k) \tag{4.86}$$

The dynamics model, f, links the previous state, $\boldsymbol{x}_{k-1}$ to the current one, $\boldsymbol{x}_k$. The measurement model h is used to calculate the measurements $\boldsymbol{z}_k$ from the state. As extra inputs to the dynamics model, control inputs, $\boldsymbol{u}_k$, can be added. The variables $\boldsymbol{w}_{k-1}$ and $\boldsymbol{v}_k$ represent the process and measurement noise, respectively. The process and measurement covariance matrix, $\boldsymbol{Q}$ and $\boldsymbol{R}$, respectively, are given by:

$$\boldsymbol{Q} = E(\boldsymbol{w}\boldsymbol{w}^T) \tag{4.87}$$

$$\boldsymbol{R} = E(\boldsymbol{v}\boldsymbol{v}^T) \tag{4.88}$$

Kalman filters follow a general pattern [20]. Firstly there is a predict cycle, then a correction cycle. The predict cycle uses the *a posteriori* state of the previous loop, $\hat{\boldsymbol{x}}_{k-1}$, and the process to calculate a new state, the *a priori* state, $\hat{\boldsymbol{x}}_k^-$. From the *a priori* state, an *a priori* measurement, $\hat{\boldsymbol{z}}_k^-$, is calculated using the measurement model h. Finally, the *a priori* covariance can be calculated. The correction cycle uses the innovation, the difference between the *a priori* measurements, $\hat{\boldsymbol{z}}_k^-$, and the actual measurements, z, to correct the *apriori* state, $\hat{\boldsymbol{x}}_k^-$, to get the *a posteriori* state, $\hat{\boldsymbol{x}}_k$. The degree of correction is based on the different covariances and is the focus of tuning the filter, as mentioned before. The different types of Kalman filters use different techniques during this process. These techniques have a large influence on the capability of resolving non-linear systems, the required computational power and the complexity of the filter implementation and tuning.

The true measurements and states do not have a superscript (*e.g.* $\boldsymbol{x}_k$). The hat represents estimated values inside the filter. The superscript $-$ denotes that the variable is an *a priori* value. The subscript (*e.g.* k) denotes the evaluation step.

4.5.2 Extended Kalman Filter

In the predict cycle, the state can be updated by propagating the state using the non-linear process equations [21].

$$\hat{\boldsymbol{x}}_k^- = f(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_k, 0) \tag{4.89}$$

Note that the noise of the system is put to zero for the integration. The integration can be performed using an Euler integration, with one or multiple steps over the integration interval, or more advanced methods (*e.g.* Runge-Kutta integrations). This has an influence on the performance of the filter, in both numerical results and computational load. From $\hat{\boldsymbol{x}}_k^-$, the *a priori* measurements are calculated with the measurement model:

$$\hat{\boldsymbol{z}}_k^- = h(\hat{\boldsymbol{x}}_k^-, 0) \tag{4.90}$$

The problem is that a Kalman gain needs to be calculated for the non-linear system. In the EKF, the Riccati equations are used, as for the linear Kalman filter:

$$\boldsymbol{P}_k^- = \boldsymbol{\Phi}_k \boldsymbol{P}_{k-1} \boldsymbol{\Phi}_k^T + \boldsymbol{Q}_k \tag{4.91}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}^T (\boldsymbol{H} \boldsymbol{P}_k^- \boldsymbol{H}^T + \boldsymbol{R}_k)^{-1} \tag{4.92}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}) \boldsymbol{P}_k^- \tag{4.93}$$

Here, $\boldsymbol{\Phi}$ is the fundamental matrix, $\boldsymbol{P}$ is the covariance matrix of the filter, $\boldsymbol{H}$ is the measurement matrix, and $\boldsymbol{Q}_k$ is the discrete process covariance matrix. The fundamental matrix is based on the state transition matrix, $\boldsymbol{F}$. The state transition matrix, $\boldsymbol{F}$, and measurement matrix, $\boldsymbol{H}$, are the Jacobians from the process and measurement equations:

$$\boldsymbol{F} = \frac{\partial f(\hat{\boldsymbol{x}})}{\partial \hat{\boldsymbol{x}}} \tag{4.94}$$

$$\boldsymbol{H} = \frac{\partial h(\hat{\boldsymbol{x}})}{\partial \hat{\boldsymbol{x}}} \tag{4.95}$$

These matrices can vary each time step, but to increase readibility the subscript k is left out. Since the process is non-linear, the fundamental matrix, $\boldsymbol{\Phi}$ can be approximated with a Taylor series expansion:

$$\boldsymbol{\Phi} = I + \boldsymbol{F}T_S + \frac{\boldsymbol{F}^2 T_S^2}{2!} + \frac{\boldsymbol{F}^3 T_S^3}{3!} + ... \tag{4.96}$$

with T_S being the sample time. Accoring to Zarchan [21], an expansion of the Taylor series above the first-order, generally does not improve the performance of the filter, since the expansion is only used for computing the covariance and not for propagation of the states. As a result, the fundamental matrix becomes:

$$\boldsymbol{\Phi} = I + \boldsymbol{F}T_S \tag{4.97}$$

The discrete process covariance matrix can now be calculated from $\boldsymbol{\Phi}$ and $\boldsymbol{Q}$:

$$\boldsymbol{Q}_k = \int_0^{T_S} \boldsymbol{\Phi}(t)\boldsymbol{Q}\boldsymbol{\Phi}^T(t)dt \tag{4.98}$$

Now the *a posteriori* state can be computed using $\boldsymbol{K}_k$ from equation 4.92, $\hat{\boldsymbol{x}}_k^-$ from equation 4.89, $\hat{\boldsymbol{z}}_k^-$ from 4.90, and the actual measurements ($\boldsymbol{z}_k$).

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \hat{\boldsymbol{z}}_k^-) \tag{4.99}$$

4.5.3 Unscented Kalman Filter

With large non-linearities, the EKF becomes inaccurate due to the linearisation process. The unscented Kalman Filter (UKF), based on the unscented transformation, tries to provide a solution for this problem by using a minimal set of sample points, the sigma points. These sample points cover the mean and covariance of the system and, after propagating these points with the non-linear process, also covers the mean and covariance of the *a posteriori* mean and covariance up to the second order for all non-linearities [16]. This filtering technique was developed in the 1990s by Julier and Uhlmann [9].

By using the unscented transformation to generate the sigma points, the UKF reduces the number of points, compared to Monte Carlo like techniques, while keeping the characteristics of the system.

The state in the UKF is augmented with zeros and the covariance is augmented with the measurement covariance and the process noise:

$$\boldsymbol{x}^a = \begin{bmatrix}\boldsymbol{x} & 0 & 0\end{bmatrix} \tag{4.100}$$

$$\boldsymbol{P}_k^{a-} = \begin{bmatrix}\boldsymbol{P}_0 & 0 & 0 \\ 0 & \boldsymbol{Q} & 0 \\ 0 & 0 & \boldsymbol{R}\end{bmatrix} \tag{4.101}$$

To compute the sigma points, $\boldsymbol{S}$ needs to be computed from the covariance of the previous step [16]:

$$\boldsymbol{P}_{k-1}^a = \boldsymbol{S}^T\boldsymbol{S} \tag{4.102}$$

If $\boldsymbol{P}_{k-1}^a$ would be a 1-by-1 matrix, $\boldsymbol{S}$ would be $\sqrt{\boldsymbol{P}_{k-1}^a}$. Therefore, it is sometimes called the *square root*. There are multiple solutions to this problem. The Cholesky decomposition is often used in UKF design for calculating $\boldsymbol{S}$, since covariance matrices are positive definite and symmetric [11]. The Cholesky decomposition is explained in appendix **??**. The $2n+1$ sigma points, with n the number of states, are calculated by:

$$\boldsymbol{\mathcal{X}}_0^- = \hat{\boldsymbol{x}}_{k-1}^a \tag{4.103}$$

$$\boldsymbol{\mathcal{X}}_i^- = \hat{\boldsymbol{x}}_{k-1}^a + \left(\sqrt{L+\lambda}\boldsymbol{S}_i\right) \text{ for } i = 1, ...L \tag{4.104}$$

$$\boldsymbol{\mathcal{X}}_i^- = \hat{\boldsymbol{x}}_{k-1}^a - \left(\sqrt{L+\lambda}\boldsymbol{S}_i\right) \text{ for } i = L+1, ...2L \tag{4.105}$$

with $\boldsymbol{S}_i$ the i-th column of $\boldsymbol{S}$, L the number of elements in the augmented state and $\lambda = \alpha^2(L+\kappa)$. α is a constant representing the spread of the sigma points and κ is a secondary tuning parameter.

Each of these sigma points is propagated using the (non-linear) process model and measurements are generated using the measurement model.

$$\boldsymbol{\mathcal{X}}_i = f(\boldsymbol{\mathcal{X}}_i^-, \boldsymbol{u}_k, 0) \text{ for } i = 0, ...2n \tag{4.106}$$

$$\boldsymbol{\mathcal{Z}}_i = h(\boldsymbol{\mathcal{X}}_i, 0) \text{ for } i = 0, ...2n \tag{4.107}$$

The /textita priori state estimate and measurement are computed by taking a weighted mean:

$$\hat{\boldsymbol{x}}_k^{a-} = \sum_{i=0}^{2n} W_{m_i} \boldsymbol{\mathcal{X}}_i \tag{4.108}$$

$$\hat{\boldsymbol{z}}_k^{a-} = \sum_{i=0}^{2n} W_{m_i} \boldsymbol{\mathcal{Z}}_i \tag{4.109}$$

with the weights being:

$$W_{m_0} = \frac{\lambda}{L+\lambda} \tag{4.110}$$

$$W_{m_i} = \frac{1}{2(L+\lambda)} \text{ for } i = 1, ...2L \tag{4.111}$$

The *a priori* state and measurement covariance, $\boldsymbol{P}_k^{a-}$ and $\boldsymbol{P}_{zz}^{a-}$, respectively, are now calculated based on the differences between the sigma points and their weighted means:

$$\boldsymbol{P}_k^{a-} = \sum_{i=0}^{2n} W_{c_i} (\boldsymbol{\mathcal{X}}_i - \hat{\boldsymbol{x}}_k^-)(\boldsymbol{\mathcal{X}}_i - \hat{\boldsymbol{x}}_k^-)^T \tag{4.112}$$

$$\boldsymbol{P}_{zz}^{a-} = \sum_{i=0}^{2n} W_{c_i} (\boldsymbol{\mathcal{Z}}_i - \hat{\boldsymbol{z}}_k^-)(\boldsymbol{\mathcal{Z}}_i - \hat{\boldsymbol{z}}_k^-)^T \tag{4.113}$$

with the weights:

$$W_{c_0} = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \tag{4.114}$$

$$W_{c_i} = W_{m_i} \text{ for } i = 1, ...2L \tag{4.115}$$

Here, β is tunable and represents the extend to which the spread of $\boldsymbol{x}$ is known beforehand. For a Gaussian distribution, β is 2. As a result, the weights for Gaussian distributions are the same for calculating covariances and means. For non-Gaussian distributions, further tuning is possible. Now the cross-covariance can be calculated in a similar fashion:

$$\boldsymbol{P}_{xz} = \sum_{i=0}^{2n} W_{c_i}(\boldsymbol{\mathcal{X}}_i - \hat{\boldsymbol{x}}_k^-)(\boldsymbol{\mathcal{Z}}_i - \hat{\boldsymbol{z}}_k^-)^T \tag{4.116}$$

The first step for the correction cycle is calculating the innovation.

$$\boldsymbol{\nu}_k = \boldsymbol{z}_k - \hat{\boldsymbol{z}}_k^- \tag{4.117}$$

In the EKF, the *a priori* state covariance is used, in combination with $\boldsymbol{H}$ and $\boldsymbol{R}$, to calculate the Kalman gain. In the UKF, the information in $\boldsymbol{P}_k^-$ is incorporated in $\boldsymbol{P}_{zz}$, and therefore also in $\boldsymbol{P}_{xz}$ and $\boldsymbol{P}_{\nu\nu}$, due to the propagation of the sigma points and by taking the weighted mean. As a result, the $\boldsymbol{K}_k$ can be computed by:

$$\boldsymbol{K}_k = \boldsymbol{P}_{xz}\boldsymbol{P}_{\nu\nu}^{-1} \tag{4.118}$$

which is a similar expression as in 4.92.

Now the *a posteriori* state, $\hat{\boldsymbol{x}}_k$, and covariance, $\boldsymbol{P}_k$, can be updated:

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\boldsymbol{\nu}_k \tag{4.119}$$

$$\boldsymbol{P}_k = \boldsymbol{P}_k^- - \boldsymbol{K}_k\boldsymbol{P}_{\nu\nu}\boldsymbol{K}_k^T \tag{4.120}$$

There is also an simplified version of the UKF. This filter generates sigma points based only on the state and its related covariance, instead of the augmented state. In that case, the process and measurement covariance are added to $\boldsymbol{P}_{k-1}$ and $\boldsymbol{P}_{\nu\nu}$, respectively:

$$\boldsymbol{P}_{k-1} = \boldsymbol{P}_{k-1} + Q \tag{4.121}$$

$$\boldsymbol{P}_{\nu\nu} = \boldsymbol{P}_{zz} + R \tag{4.122}$$

The addition of the process noise is done before the sigma points are determined. The measurement noise is added to the measurement covariance, resulting in the innovation covariance. This replaces the measurement covariance for the calculation of the Kalman gain. The rest of the covariances and states are calculated as for the normal UKF. According to REFERENCE, the non-augmented state should have a better performance for high noise systems, although higher-order non-linearity, especially in the measurements is estimated less accurately.

An important note for the verification of the EKF and UKF compared to each other is that both filters should have the same performance for linear systems, and very similar for slightly non-linear systems.

To summarise, due to the spread of sigma points, the UKF has an advantage over the EKF, since the covariances are computed from a propagation of sigma points, and should therefore include a better mapping of non-linear relations in the state and in the measurement model. The disadvantage is the requirement to compute the *square root*, increasing the computational load. Due to the autonomy requirement, a low computational load is favourable. In combination with handling large noises better, the non-augmented state UKF is chosen over the normal UKF.

Chapter 5

Implementation and Verification

This chapter describes the system implementation. This is a step-wise process, starting with the falling-body reference case for the EKF and ending with the complete system for EKF and UKF. Unit tests for smaller units have been added in A. Large derivations are also included in the appendices.

First, the general implementation is described in 5.1. The other sections cover the step-wise approach.

5.1 General Implementation

The system is modelled in Matlab for the ease of implementation and debugging. For more optimised performance, an implementation in other coding languages (*e.g.* C++) could be better. However, for a relative performance comparison, this is sufficient.

Throughout the simulation campaign, a central library is used, so that incremental changes are applied to all simulations. This is not always possible, but maximised if possible to reduce the chance of errors.

The basic setup is straightforward and the same for all simulations:

1. Initialise the orbit, attitude and time for the true orbit
2. Load the simulation parameters: the filter tuning parameters, the description of the environment, the sensor characteristics, and the standard deviation of the initial conditions
3. Generate a trajectory: a propagation of the intialised conditions or a load of pre-generated trajectory
4. Generate measurements or load pre-generated measurements for the trajectory
5. Loop through the filter

6. Analyse the results

The analysis of the results is mainly based on plots. Two sets of plots are use mostly. The first type shows the difference between the actual and estimated state. In this plot, the covariance is often included. The diagonal of the covariance matrix is, in a linear system, the square of the standard deviation, σ. The $3 - \sigma$ value should contain about 99.7% of those differences. The second type of plots are measurement plots, containing the input measurements and the measurements generated inside the filter. When the filter converges, the internal measurements should stabilise on the average of the input measurements.

5.2 Falling-body Problem

The goal of the falling body problem is testing the implementation of the EKF algorithms.

5.2.1 Implementation Description

Process Description

The falling-body problem is specified by Zarchan [21]. The system consists of a falling body in a uniform gravity field, susceptible to aerodynamic drag. Only vertical motion is taken into account. The states are:

$$\boldsymbol{x} = \begin{bmatrix} x \\ dotx \end{bmatrix} \tag{5.1}$$

The acceleration is described by:

$$\ddot{x} = \frac{0.5\rho\dot{x}^2}{\beta} - g \tag{5.2}$$

where β is the ballistic coefficient of the body, g is the gravitational acceleration, and ρ is the atmospheric density. ρ is described as:

$$\rho = 0.0035e^{\frac{-x}{22000}} \tag{5.3}$$

The gravitational acceleration, g, is 32.2ft/s^2. The ballistic coefficient is set to $500\ \text{lb/ft}^2$.

The input trajectory is propagated using the ode45 function. Zarchan uses a second-order Runge-Kutta for the integration, but due to the simplicity of the system, the difference in performance between these two should be negligible. This also allows to test the correct implementation of the trajectory integration. Inside the filter, the state propagation is performed with an Euler integration.

The details of the derivation of the process covariance matrix, the fundamental matrix and the discrete process covariance matrix can be found in [21]. For completeness, the matrices are presented below. The process covariance matrix is:

$$\boldsymbol{F} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial \dot{x}} \\ \frac{\partial \ddot{x}}{\partial x} & \frac{\partial \ddot{x}}{\partial \dot{x}} \end{bmatrix} \tag{5.4}$$

$$= \begin{bmatrix} 0 & 1 \\ \frac{-\rho(\hat{x})g\hat{\dot{x}}^2}{44000\beta} & \frac{\rho(\hat{x})g\hat{\dot{x}}}{\beta} \end{bmatrix} \tag{5.5}$$

The first-order Taylor-series expansion of the fundamental matrix is:

$$\boldsymbol{\Phi}(t) = \boldsymbol{I} + \boldsymbol{F}t \tag{5.6}$$

$$= \begin{bmatrix} 1 & t \\ f_{21}t & 1 + f_{22}t \end{bmatrix} \tag{5.7}$$

The continuous process-noise matrix is:

$$\boldsymbol{Q}(t) = \begin{bmatrix} 0 & 0 \\ 0 & \Phi_S \end{bmatrix} \tag{5.8}$$

where Φ_S is the spectral density of the white noise on the calculation of the acceleration during the integration inside the filter. The discrete process-noise matrix can then be calculated from $\boldsymbol{Q}(t)$ and $\boldsymbol{F}$:

$$\boldsymbol{Q}_k = \int_0^{T_S} \boldsymbol{\Phi}(t)\boldsymbol{Q}\boldsymbol{\Phi}^T(t)\,\mathrm{d}t \tag{5.9}$$

$$= \Phi_S \begin{bmatrix} \frac{T_S^3}{3} & \frac{T_S^2}{2} + F_{22}\frac{T_S^3}{3} \\ \frac{T_S^2}{2} + f_{22}\frac{T_S^3}{3} & T_S + f_{22}T_S^2 + f_{22}^2\frac{T_S^3}{3} \end{bmatrix} \tag{5.10}$$

Measurements Description

There is one measurement, a radar measurement. This measurement measures the range from the radar to the object with a measurement accuracy with a standard deviation of 1000 ft. Measurements are taken at 10 Hz over a time period of 30 s. Initially, the object is at 200 000 ft above the radar.

Filter Implementation and Assumptions

Zarchan describes three EKF implementations in chapter 7 of [21]:

1. EKF without and with process noise, with a first-order Taylor series expansion for the fundamental matrix, and Euler integration for the process in the filter
2. EKF without process noise, second- and third-order Taylor series expansions for the fundamental matrix, and Euler integration for the process in the filter
3. EKF without process noise, with a first-order Taylor series expansion for the fundamental matrix, and more accurate integration methods for the process in the filter

The conclusions from those three implementations are, according to Zarchan, that the use of higher-order Taylor series for the fundamental matrix does not have a large influence on the stability of the filter. The addition of process noise and more accurate integration helps. Therefore, the first and third implementation are tested to check whether the EKF implementation is correct. The use of process noise and better integration can be used later on for trying to optimise the filters.

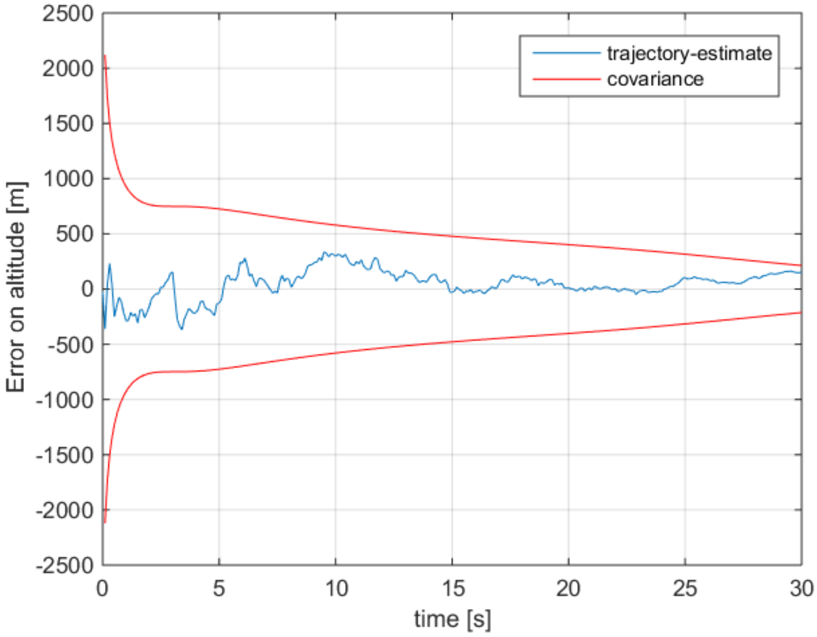

Figure 5.1: Difference between simulated and estimated altitude for the falling body problem with a standard deviation of 1000 ft

5.2.2 Results

First Implementation

The first test is performed with the standard deviation of the measurements on 1000 ft, without any process noise. The results are shown in figure 5.1. The filter converges and the errors on the states remain within the boundaries, as predicted by the covariance. The behaviour is the same as described by Zarchan [21]. The dotted red line is the square root the diagonal of the covariance matrix and represents the $1-\sigma$ value of the state. The full red line is the $3-sigma$ value. Ideally, the $1-\sigma$ and $3-\sigma$ bounds should cover 68% and 99.7% of the error distribution, respectively.

The second simulation uses a standard deviation of 25 ft for the measurements, without process noise. As shown in figure 5.2, the error on the states diverges after 20 s. This same behaviour is also observed by Zarchan and is caused by the Kalman gains being very low. As a result, the filter wil give a higher weight to the process and reduce the influence of the measurements as a correcting factor.

Based on the previous simulation, a third simulation is run with a process noise of 100 $(=\Phi_S)$. This is not an optimised value, but shows the influence on the stability of the filter, as can be seen in figure 5.3. The filter no longer diverges, but remains within the $3-sigma$ boundaries. Increasing the process noise, increases the Kalman gain. Therefore, the difference between the predicted measurement and the real measurement has a larger influence on the behaviour of the filter, as can be deduced from equation 4.99.

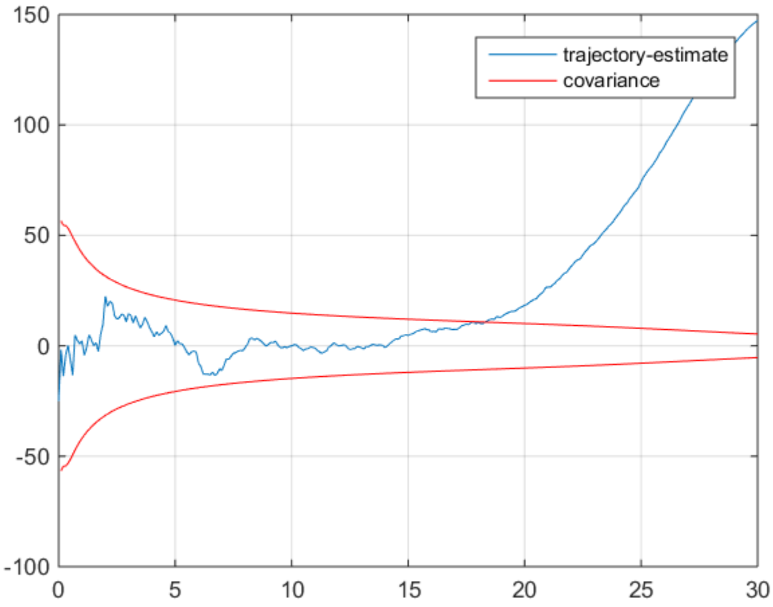

Figure 5.2: Difference between simulated and estimated altitude for the falling body problem with a standard deviation of 25 ft

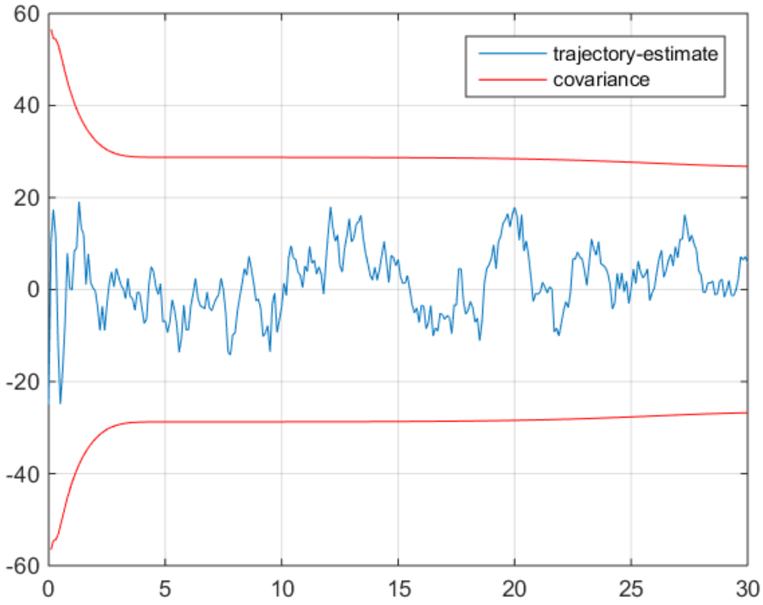

Figure 5.3: Difference between simulated and estimated altitude for the falling body problem with a standard deviation of 25 ft and $\Phi_S = 100$

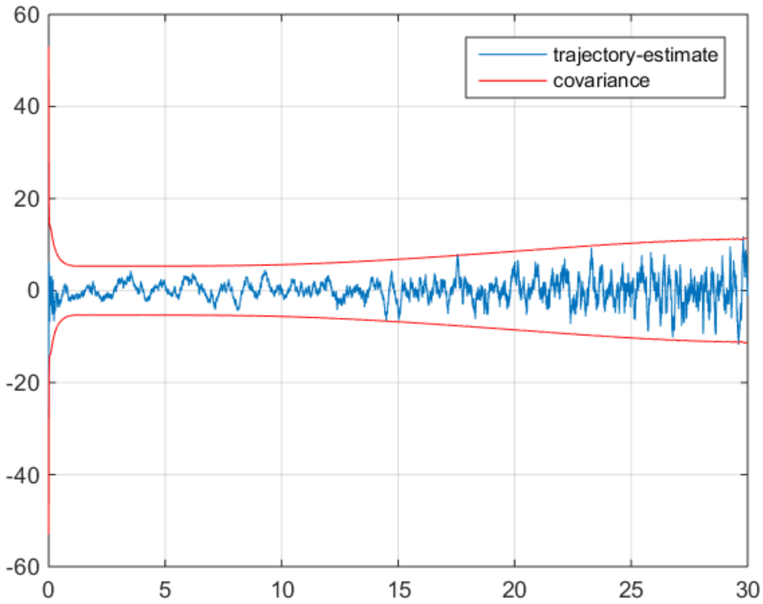

Figure 5.4: Difference between simulated and estimated altitude for the falling body problem with reduced integration steps

Third Implementation

In the third implementation, the step-size of the Euler integration is changed from 0.1 s steps to 0.001 s. As a result, the integration accuracy over 0.1 s inside the filter becomes more accurate. This simulation is run both without ($\Phi_S = 0$) and with process noise ($\Phi_S = 100$). The results are similar to those described by Zarchan, see [21]. The result of with $\Phi_S = 0$ is given in figure 5.4.

As mentioned by Zarchan, the difference between the simulations with process noise, but with a different integrator are not very large. Therefore, it could be useful to use a larger integration step to reduce the computational load and use process noise to avoid drifting.

5.3 Position Estimation in Orbit

The goal of the in-orbit position estimation is to check the implementation of the lateral equations of motion and verify the implementation of the UKF by cross-checking with the EKF.

5.3.1 Implementation Description

Process Description

The states are:

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{5.11}$$

The initial conditions are chosen such that a circular orbit is obtained with a semi-major axis of 40 000 km with an inclination of 45, in case no disturbances are present, and the ascending node on the X-axis. Earth is chosen as the central body. First, a system with only a central gravity acting on the spacecraft is implemented, then one with disturbances.

The dynamics of the undisturbed orbit are described by (see 4.2.1):

$$\ddot{\boldsymbol{x}} = -\frac{\mu}{|r|^3}\boldsymbol{r} \tag{5.12}$$

with $\boldsymbol{r}$ being the position vector.

The equations of motion for the disturbed orbit are described by:

$$\ddot{\boldsymbol{x}} = -\frac{\mu}{|r|^3}\boldsymbol{r} - \frac{\mu}{|r_M|^3}\boldsymbol{r_M} - \frac{\mu}{|r_S|^3}\boldsymbol{r_S} + C_r\frac{WS}{cm^2}\frac{\boldsymbol{r}_S}{|r_S|} \tag{5.13}$$

with the subscripts M referring to the Moon and S to the Sun. The second and third terms are the lunar and solar gravity, respectively, and the last term is the solar pressure.

The derivation of the Jacobians of the process matrix can be found in (REFERENCE TO APPENDIX). For completeness, the Jacobian is also given here:

$$\boldsymbol{F} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} & \frac{\partial \dot{x}}{\partial z} & \frac{\partial \dot{x}}{\partial \dot{x}} & \frac{\partial \dot{x}}{\partial \dot{y}} & \frac{\partial \dot{x}}{\partial \dot{z}} \\ \frac{\partial \dot{y}}{\partial x} & \frac{\partial \dot{y}}{\partial y} & \frac{\partial \dot{y}}{\partial z} & \frac{\partial \dot{y}}{\partial \dot{x}} & \frac{\partial \dot{y}}{\partial \dot{y}} & \frac{\partial \dot{y}}{\partial \dot{z}} \\ \frac{\partial \dot{z}}{\partial x} & \frac{\partial \dot{z}}{\partial y} & \frac{\partial \dot{z}}{\partial z} & \frac{\partial \dot{z}}{\partial \dot{x}} & \frac{\partial \dot{z}}{\partial \dot{y}} & \frac{\partial \dot{z}}{\partial \dot{z}} \\ \frac{\partial \ddot{x}}{\partial x} & \frac{\partial \ddot{x}}{\partial y} & \frac{\partial \ddot{x}}{\partial z} & \frac{\partial \ddot{x}}{\partial \dot{x}} & \frac{\partial \ddot{x}}{\partial \dot{y}} & \frac{\partial \ddot{x}}{\partial \dot{z}} \\ \frac{\partial \ddot{y}}{\partial x} & \frac{\partial \ddot{y}}{\partial y} & \frac{\partial \ddot{y}}{\partial z} & \frac{\partial \ddot{y}}{\partial \dot{x}} & \frac{\partial \ddot{y}}{\partial \dot{y}} & \frac{\partial \ddot{y}}{\partial \dot{z}} \\ \frac{\partial \ddot{z}}{\partial x} & \frac{\partial \ddot{z}}{\partial y} & \frac{\partial \ddot{z}}{\partial z} & \frac{\partial \ddot{z}}{\partial \dot{x}} & \frac{\partial \ddot{z}}{\partial \dot{y}} & \frac{\partial \ddot{z}}{\partial \dot{z}} \end{bmatrix} \tag{5.14}$$

$$= \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ & & & 0 & 0 & 0 \\ & F(4:6,1:3) & & 0 & 0 & 0 \\ & & & 0 & 0 & 0 \end{array}\right] \tag{5.15}$$

The lower left part of the matrix becomes:

$$F(4,1) = \frac{-\mu}{r^5}(r^2 - 3x^2) + \frac{-\mu}{r_M^5}(r_M^2 - 3x_M^2) + \frac{-\mu}{r_S^5}(r_S^2 - 3x_S^2) + C_r\frac{WS}{cm^2 r_S^3}(r_S^2 - x_S^2) \tag{5.16}$$

$$F(4,2) = 3\mu\frac{xy}{r^5} + 3\mu\frac{x_M y_M}{r_M^5} + 3\mu\frac{x_S y_S}{r_S^5} + C_r\frac{WS}{cm^2 r_S^3}(x_S y_S) \tag{5.17}$$

$$F(4,3) = 3\mu\frac{xz}{r^5} + 3\mu\frac{x_M z_M}{r_M^5} + 3\mu\frac{x_S z_S}{r_S^5} + C_r\frac{WS}{cm^2 r_S^3}(x_S z_S) \tag{5.18}$$

$$F(5,1) = F(4,2) \tag{5.19}$$

$$F(5,2) = \frac{-\mu}{r^5}(r^2 - 3y^2) + \frac{-\mu}{r_M^5}(r_M^2 - 3y_M^2) + \frac{-\mu}{r_S^5}(r_S^2 - 3y_S^2) + C_r\frac{WS}{cm^2 r_S^3}(r_S^2 - y_S^2) \tag{5.20}$$

$$F(5,3) = 3\mu\frac{yz}{r^5} + 3\mu\frac{y_M z_M}{r_M^5} + 3\mu\frac{y_S z_S}{r_S^5} + C_r\frac{WS}{cm^2 r_S^3}(y_S z_S) \tag{5.21}$$

$$F(6,1) = F(4,3) \tag{5.22}$$

$$F(6,2) = F(5,3) \tag{5.23}$$

$$F(6,3) = \frac{-\mu}{r^5}(r^2 - 3x^2) + \frac{-\mu}{r_M^5}(r_M^2 - 3z_M^2) + \frac{-\mu}{r_S^5}(r_S^2 - 3z_S^2) + C_r\frac{WS}{cm^2 r_S^3}(r_S^2 - z_S^2) \tag{5.24}$$

For the undisturbed orbit, only the first terms need to be taken into account.

The fundamental matrix is calculated according to 5.6 with the matrix for the process noise is given by:

$$\boldsymbol{Q}(t) = \begin{bmatrix} \Phi_{S1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \Phi_{S2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \Phi_{S3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \Phi_{S4}0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \Phi_{S5} & 0 \\ 0 & 0 & 0 & 0 & 0 & \Phi_{S6} \end{bmatrix} \tag{5.25}$$

with Φ_{S1} to Φ_{S6} representing the white noises on the process. The discrete process-noise matrix is calculated using equation 5.9.

Measurements Description

The measurements consist of the position with a white noise with a standard deviation of 1000 m. The measurements are kept simple to avoid further errors due to derivations of the measurement matrix while verifying the behaviour of the orbit dynamics equations and the implementation of the UKF.

For the EKF, the measurement matrix becomes:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.26}$$

Filter Implementation and Assumptions

Four implementations are implemented:

1. The EKF without disturbances: this is used as the reference case

2. The UKF without disturbances to verify the implementation of the UKF
3. The EKF with disturbances to verify the implementation of the disturbed orbit
4. The UKF with disturbances

The propagation of the orbit in both filters is performed using an Euler integration. In case of the disturbed orbit, the disturbances are also included in the propagation. In a later stage, removing the disturbances is tested.

The difference between the disturbed and undisturbed orbit are very small. The test on the performance of the propagation is performed in appendix A.2. The evaluation of the correct implementation of the disturbances is therefore based on the relatively small impact they have on the orbit. Therefore, the difference between the disturbed orbit and its estimation should remain almost the same as for the undisturbed orbit. Since there are no large non-linearities in this system, the difference between the EKF and UKF should also be minimal.

5.3.2 Results

The difference between simulated and estimated position are given in figure 5.5 and 5.5. As expected, the difference between the simulation and the state converges within the $3-\sigma$ boundaries. For the reference case, both position and velocity are given. For the other implementations, only the position is shown. In section 5.1, it is mentioned the internal measurements of the filter should settle around the average of the input measurements. As can be seen in figure 5.5, the noise on the measurement difference is about the noise of 1000 m which was put on the measurements.

The performance of the UKF is almost identical to the performance of the EKF for the undisturbed case, as can be seen in figure 5.8. In case of a linear system, the performance of the EKF and the UKF should be the same as for the linear Kalman filter. Since the non-linearities in the system are very low, it is normal that the system performs almost identical. Therefore, it can be assumed that the UKF is implemented correctly.

Since the disturbances should have a minor influence on the performance of the filter, the difference between the disturbed and undisturbed cases should be negligible. As can be seen in figures 5.9 and 5.10, the performance is very close to the undisturbed cases. Therefore, the implementation of the disturbances in the propagator can be assumed correct, as well as the derivation of the Jacobians of the disturbances.

5.4 Planet Measurements for Position

The goal is to verify the implementation of the planet measurements in the EKF and UKF. Therefore, the attitude is not included, but the position is measured with the sensors fixed in a spacecraft-centred inertial system. As a further simplification, the field-of-view of the sensors is not limited. This could include singularities, so the resulting measurements must be looked at with care.

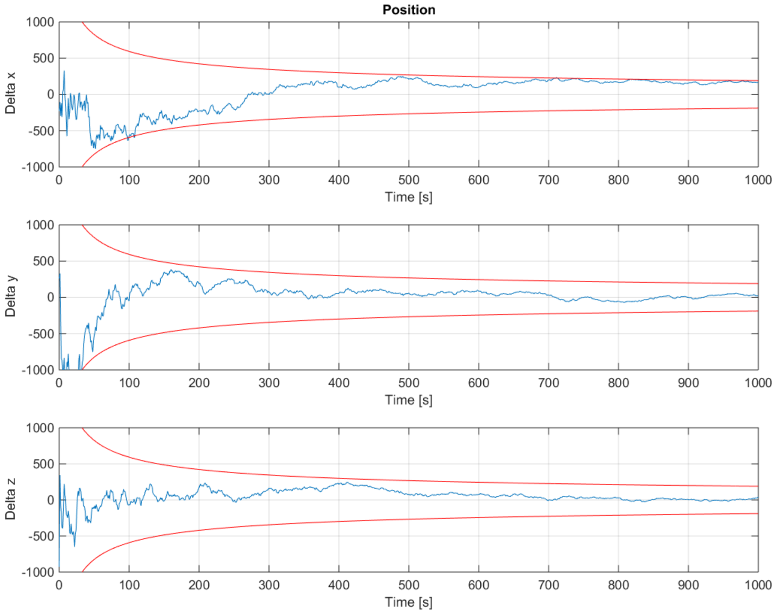

Figure 5.5: Difference between simulated and estimated position for the direct position estimation with EKF in a circular orbit without disturbances

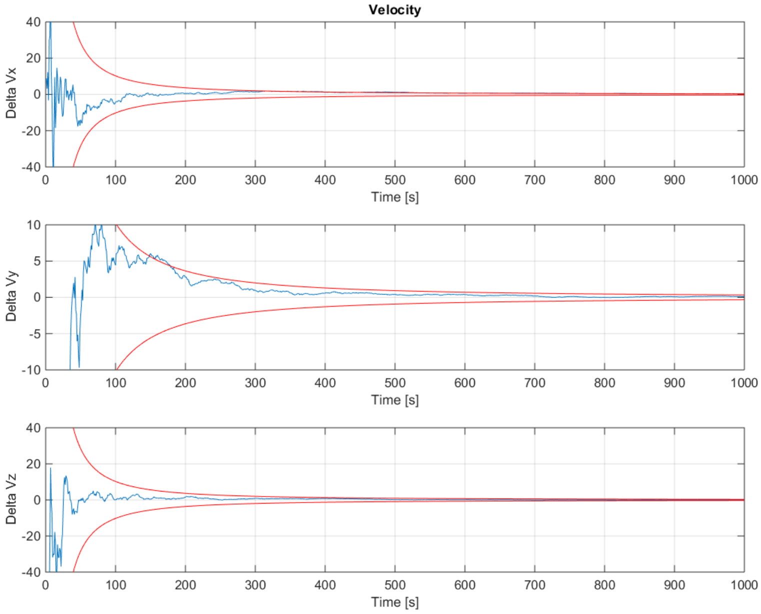

Figure 5.6: Difference between simulated and estimated velocity for the direct position estimation with EKF in a circular orbit without disturbances

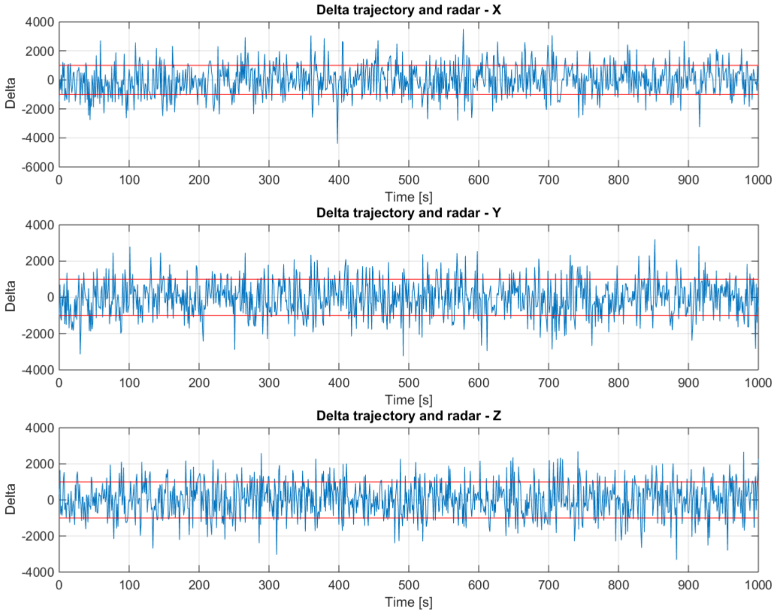

Figure 5.7: Difference between input measurements and the internal filter measurements for the direct position with EKF estimation in a circular orbit without disturbances

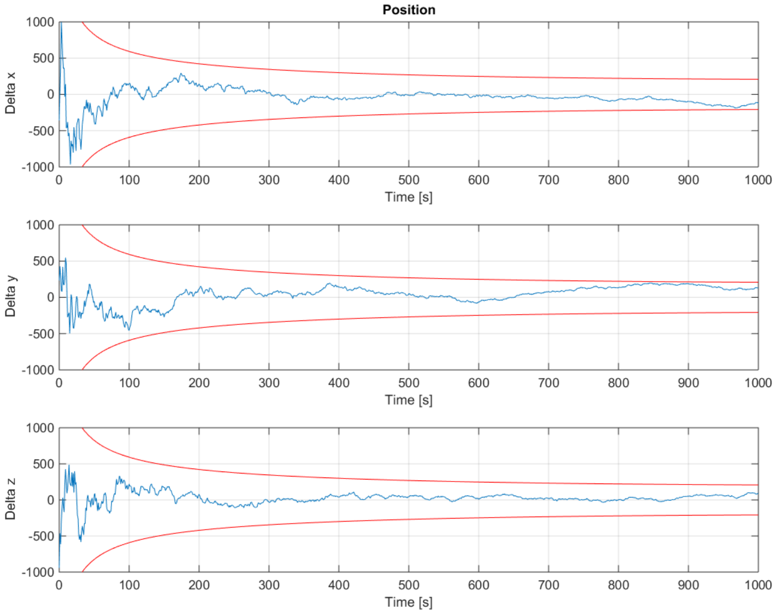

Figure 5.8: Difference between simulated and estimated position for the direct position estimation with UKF in a circular orbit without disturbances

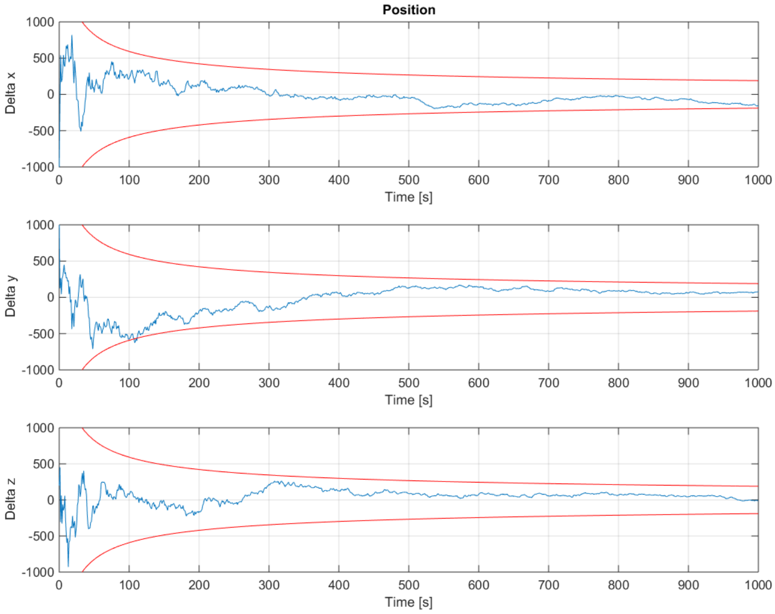

Figure 5.9: Difference between simulated and estimated position for the direct position estimation with EKF in a circular orbit with disturbances

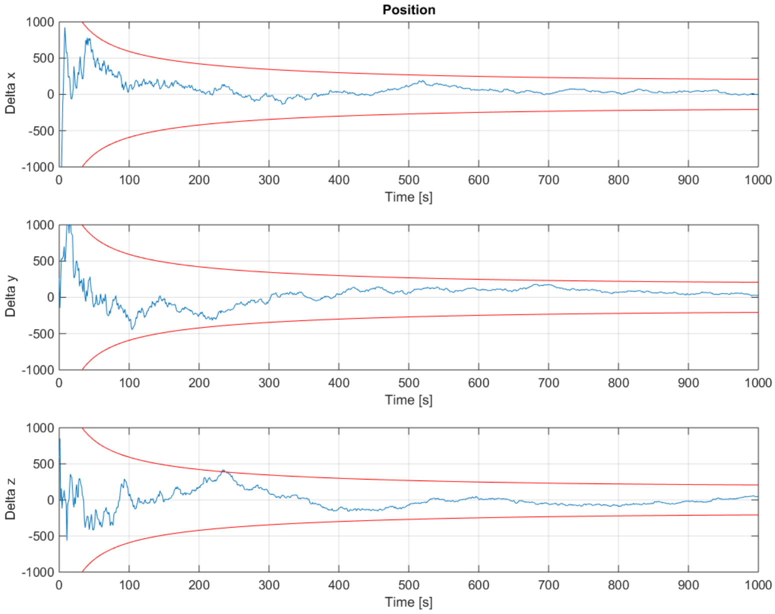

Figure 5.10: Difference between simulated and estimated position for the direct position estimation with UKF in a circular orbit with disturbances

Three sensors are being used for this test: planet measurements of Earth and Moon, and a Sun sensor. For the EKF, two different implementations are used for the planet measurements. The first one is based on the azimuth, elevation and angular size. The second one includes a further processing to obtain the position vector of the planet. For the UKF, only the azimuth, elevation and angular size measurements are implemented.

5.4.1 Implementation Description

Process Description

The states are the same as the one used in the previous section, see 5.3.1.

The sensors are oriented in the same direction as the inertial frame. The sensor observation axis is the Z-axis in the sensor frame, and so also in the inertial frame. The initial position is chosen on the -Z-axis, resulting in an initial azimuth and elevation of Earth of 0°. The semi-major axis is again chosen at 40 000 km. To check the behaviour of both azimuth and elevation, the velocity is:

$$\begin{bmatrix} v_x & v_y & v_z \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2}v_0 & -\frac{\sqrt{2}}{2}v_0 & 0 \end{bmatrix} \tag{5.27}$$

with v_0 being the velocity required for a circular orbit. With this initial velocity, the azimuth and elevation should start at zero and decrease (azimuth) or increase (elevation).

Measurements Description

Two types of measurements are to be implemented. One is a the angles measurement (azimuth, elevation and angular size), the other is the position vector measurement. To allow for a fair trade-off between the two systems, the position vector measurement is computed from the azimuth, elevation and angular size. This way, the noise on the position is the same, only described differently. This allows for an extra cross-check between the systems. Since the attitude is not modelled, the sensors are assumed to be centred in an inertial frame, centred in the spacecraft.

The measurements consist of:

1. Planet measurements of Earth
2. Planet measurements of the Moon
3. Sun sensor

The planet measurements are described in 4.3.1.

The angle measurements consist of an azimuth (α), elevation (ϵ) and a angular size measurement. For simplicity, the Sun sensor is implemented similarly with azimuth and elevation as outputs. For the first implementation, the measurements are:

$$\boldsymbol{z} = \begin{bmatrix} \alpha_E \\ \epsilon_E \\ \theta_E \\ \alpha_M \\ \epsilon_M \\ \theta_M \\ \alpha_S \\ \epsilon_S \end{bmatrix} \tag{5.28}$$

with α (see equation 4.61), ϵ (see equation 4.62) and θ (see equation 4.65) calculated by:

$$\alpha = \operatorname{atan2}({x_X}_S, {z_X}_S) \tag{5.29}$$

$$\epsilon = \operatorname{asin}({y_X}_S/|{r_X}_S|) \tag{5.30}$$

$$\theta = 2\operatorname{asin}\left(\frac{R_X}{{r_X}_S}\right); \tag{5.31}$$

with the subscript X_S referring to body X (Earth, Moon or Sun) in the sensor frame (S). For the UKF, these equations are sufficient. For implementation in the EKF, the measurement matrix, $\boldsymbol{H}$, needs to be calculated. The implementation of the sensors, centred in the spacecraft-centred inertial frame, is called the simplified model and the derivation can be found in appendix **??**. The elements of $\boldsymbol{H}$ related to the azimuth are:

$$\boldsymbol{H}(i,1) = \frac{-{z_X}_S}{{x_X}_S^{S^2} + {z_X}_S{}^2} \tag{5.32}$$

$$\boldsymbol{H}(i,2) = 0 \tag{5.33}$$

$$\boldsymbol{H}(i,3) = \frac{{x_X}_S}{{x_X}_S^{S^2} + {z_X}_S{}^2} \tag{5.34}$$

where the i-th row is the row of the respective measurement and the columns 1, 2 and 3 are related to the states x, y and z. The elements of $\boldsymbol{H}$ related to the elevation are:

$$\boldsymbol{H}(i,1) = \frac{{x_X}_S {y_X}_S}{{r_X}_S^2\sqrt{{r_X}_S^2 - {y_X}_S^2}} \tag{5.35}$$

$$\boldsymbol{H}(i,2) = \frac{{y_X}_S^2 - {r_X}_S^2}{{r_X}_S^2\sqrt{{r_X}_S^2 - {y_X}_S^2}} \tag{5.36}$$

$$\boldsymbol{H}(i,3) = \frac{{z_X}_S {y_X}_S}{{r_X}_S^2\sqrt{{r_X}_S^2 - {y_X}_S^2}} \tag{5.37}$$

The elements of $\boldsymbol{H}$ related to the angular size are:

$$\boldsymbol{H}(i,1) = \frac{2R_X {x_X}_S}{{R_E}^2\sqrt{{r_X}_S^2 - {R_E}^2}} \tag{5.38}$$

$$\boldsymbol{H}(i,2) = \frac{2R_X {y_X}_S}{{R_E}^2\sqrt{{r_X}_S^2 - {R_E}^2}} \tag{5.39}$$

$$\boldsymbol{H}(i,3) = \frac{2R_X z_{XS}}{{R_E}^2\sqrt{r_{XS}^2 - {R_E}^2}} \tag{5.40}$$

The azimuth, elevation and angular size are not related to the angular rate, so the columns 4 to 6 of the matrix are zero.

For the position vector measurements, the Earth and Moon position are calculated from the above measurements. In the filter, the system is easier, since the position vector of Earth and Moon are directly related to the position of the spacecraft:

$$H(i,1) = \frac{\partial x_{XS}}{\partial x} = -1 \tag{5.41}$$

$$H(i+1,1) = \frac{\partial y_{XS}}{\partial y} = -1 \tag{5.42}$$

$$H(i+2,1) = \frac{\partial z_{XS}}{\partial z} = -1 \tag{5.43}$$

All the other elements are zero.

Filter Implementation and Assumptions

In this test, the sensors are assumed to be in a spacecraft-centred inertial frame. This choice was made to avoid further complications due to the cross-correlation of position and attitude measurements.

The second assumption is the infinite field-of-view. To be able to check the convergence of the filter, continuous measurements are helpful. However, problems can occur if the measured elevation goes to ±90°. In that case, the tangent function goes to infinity and the error behaviour could be incorrect. This could potentially cause problems for the filter.

For the propagation inside the EKF and UKF, the Euler integration is used.

5.4.2 Results

The results are expected to be similar for all implementations, if there is no singularity due to measurements. Therefore, the measurements are being looked at first. The measurements for the EKF and UKF implementation are the same. The measurements for the position vector measurements are processed further. The measurements, both internal and inputs, for Earth, Moon and Sun sensor are given for the EKF in figures 5.11 to 5.13, respectively. In the Moon and Sun sensor measurements, there should be no singularities over the simulated time 25 000 s. In the measurements of the Earth sensor, there could potentially be a problem, due to the low noise on the elevation at about 20 000 s. The elevation is measured in YZ-plane of the sensor system. If the azimuth is 90°, the planet must be on the XY-plane. Due to the modelling of the noise, the standard deviation of the noise will reduces. Normally, this is not a problem, because the field-of-view of the sensors is limited to 20° half-cone. The deviation of the noise in that field-of-view from a constant is less thamn 10%.

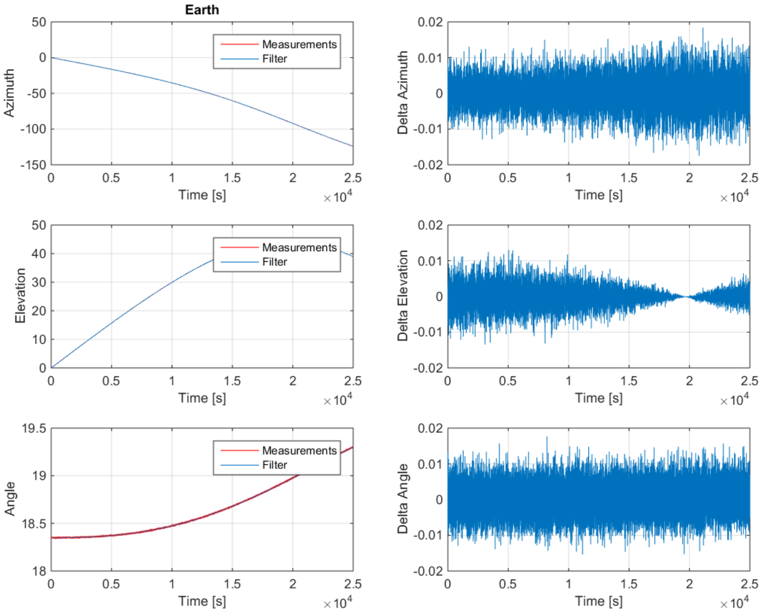

Figure 5.11: Measurements of the Earth sensor for the estimation with EKF with angle measurements in the inertial frame

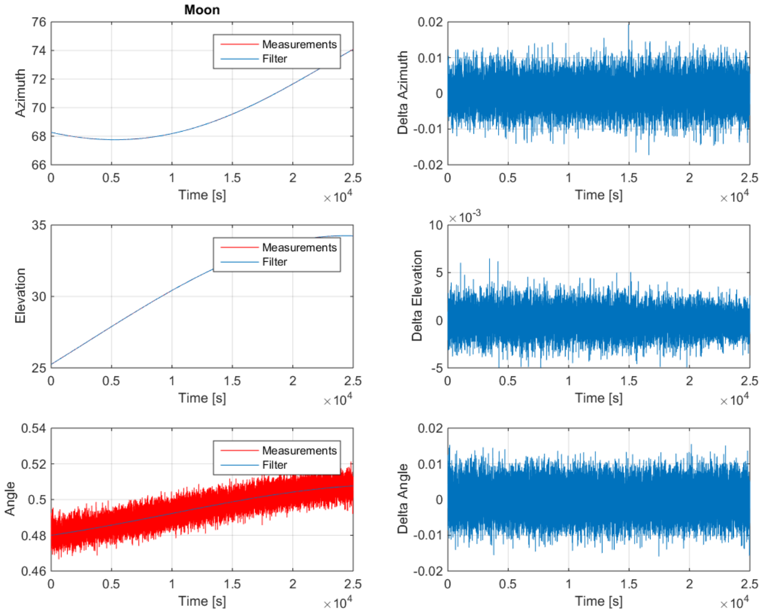

Figure 5.12: Measurements of the Moon sensor for the estimation with EKF with angle measurements in the inertial frame

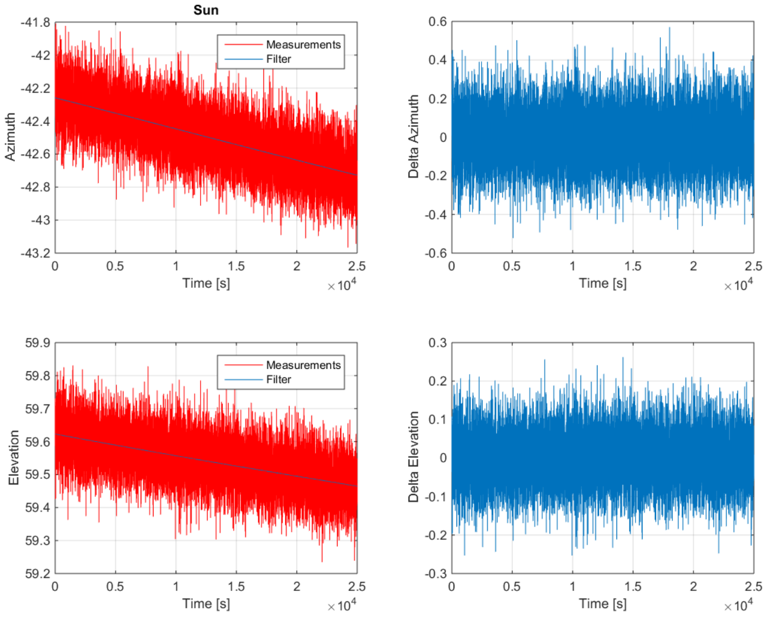

Figure 5.13: Measurements of the Sun sensor for the estimation with EKF with angle measurements in the inertial frame

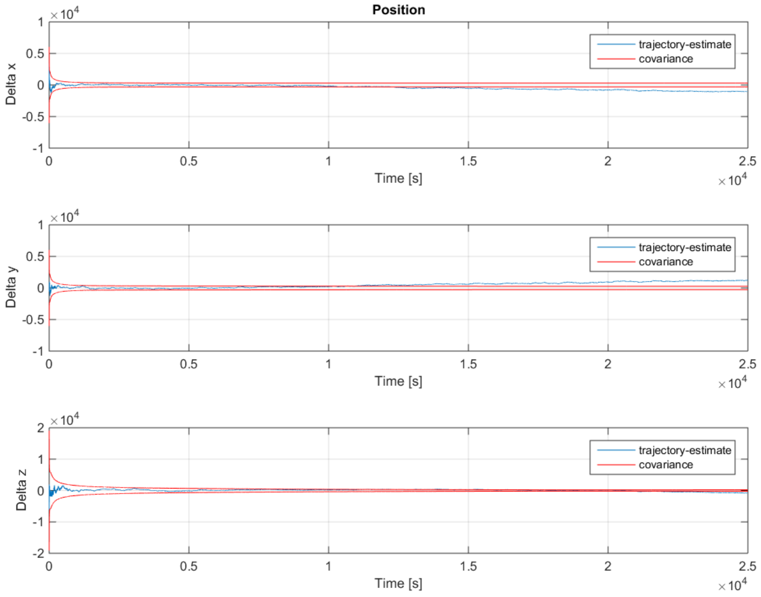

Figure 5.14: Difference between simulated and estimated position for the estimation with EKF with angle measurements in the inertial frame

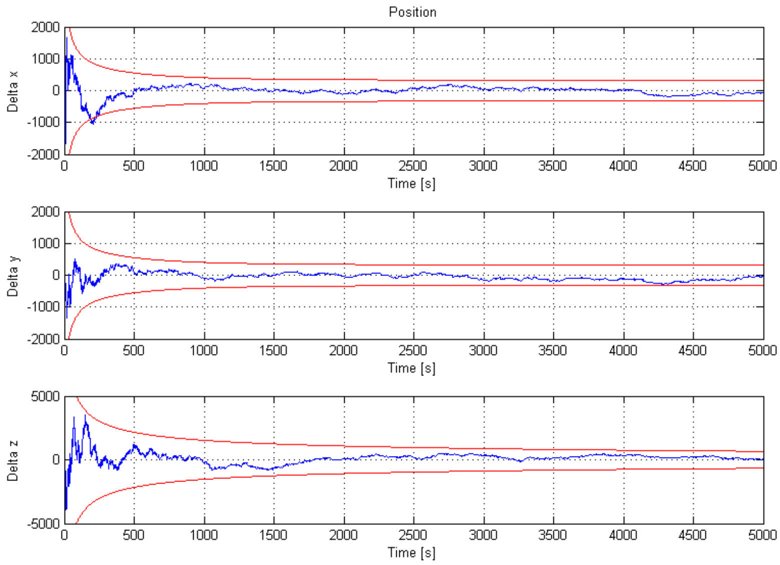

Figure 5.15: Difference between simulated and estimated position for the estimation with UKF with angle measurements in the inertial frame

None of the filters is capable of handling the lower noise. All have a similar behaviour as shown for the EKF in figure 5.14. However, if only the first 5000 s are looked at, with data that is not or less influenced by the reduction of noise on one axis, the differences between estimated and simulated position for the three implementations are converging within the $3-\sigma$ boundaries, as can be seen in figures 5.14 to 5.15. For this reason, another initial state is tested with Therefore, it can be assumed that the three implementations are properly implemented, with a focus on the simplified model in the EKF and UKF, the measurement matrix in the EKF and the position vector measurements for the EKF.

5.5 Planet Measurements and Star Tracker

The next step consists of adding the star tracker and observing the attitude. This is included in the three configurations from the previous test: the EKF with angle measurements, the EKF with position vector measurements and the UKF.

The implementation is also performed in steps.

1. The implementation of the previous test is expanded with the star tracker measuring the attitude and the other measurements measuring the position.
2. The attitude and position are linked by mounting the planet and Sun sensors on the attitude.
3. The final step is having the system rotate and limit the field-of-view.

These final tests should show the capabilities of the system to estimate both position and attitude with limited measurements.

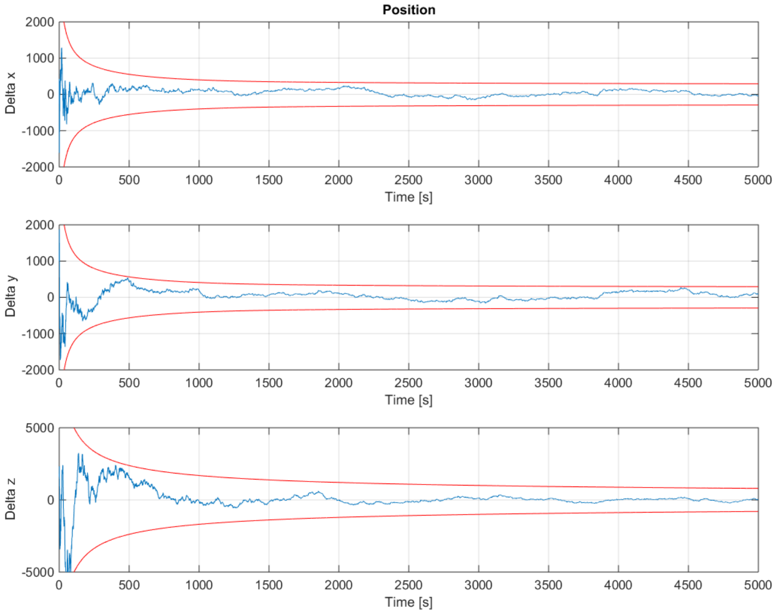

Figure 5.16: Difference between simulated and estimated position for the estimation with EKF with position vector measurements in the inertial frame

5.5.1 Implementation Description

Process Description

The state is now expanded to include the attitude:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{r} \\ \dot{\boldsymbol{r}} \\ \boldsymbol{q} \\ \boldsymbol{\omega} \end{bmatrix} \tag{5.44}$$

with

$$\boldsymbol{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad \dot{\boldsymbol{r}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{5.45}$$

$$\boldsymbol{r} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \qquad \boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{5.46}$$

The states for the attitude are described in section 4.2.4.

$$\dot{q}_1 = 0.5(\omega_x q_4 + \omega_y q_3 - \omega_z q_2) \tag{5.47}$$

$$\dot{q}_2 = 0.5(\omega_y q_4 + \omega_z q_3 - \omega_x q_3) \tag{5.48}$$

$$\dot{q_3} = 0.5(\omega_z q_4 + \omega_x q_2 - \omega_y q_1) \tag{5.49}$$

$$\dot{q_4} = 0.5(-\omega_x q_1 - \omega_y q_2 - \omega_z q_3) \tag{5.50}$$

$$\dot{\omega_x} = 0 \tag{5.51}$$

$$\dot{\omega_y} = 0 \tag{5.52}$$

$$\dot{\omega_z} = 0 \tag{5.53}$$

The derivatives of the rotational rates are assumed to be zero with the disturbances on the rates assumed to be negligible for filter performance. On longer simulations, the difference is notable in the attitude and rates. However, the disturbances are very low, so the rates changes slowly. As a result, this should be easily filtered out with the measurements from the star trackers with minimal influence on the filter.

For the UKF, the above description is sufficient. For the EKF, the state transition matrix needs to be derived. The complete derivation is given in **??**.

$$F_q = \begin{bmatrix} 0 & -0.5\omega_z & 0.5\omega_y & 0.5\omega_x & 0.5q_4 & 0.5q_3 & -0.5q_2 \\ 0.5\omega_z & 0 & -0.5\omega_x & 0.5\omega_y & -0.5q_3 & 0.5q_4 & 0.5q_1 \\ -0.5\omega_y & 0.5\omega_x & 0 & 0.5\omega_z & 0.5q_2 & -0.5q_1 & 0.5q_4 \\ -0.5\omega_x & -0.5\omega_y & -0.5\omega_z & 0 & -0.5q_1 & -0.5q_2 & -0.5q_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.54}$$

Since no disturbances are taken into account for the attitude, there is no correlation between attitude and position. As a result, the fundamental matrix is:

$$F = \begin{bmatrix} F_x & 0 \\ 0 & F_q \end{bmatrix} \tag{5.55}$$

with F_x described in 5.3.1.

Measurements Description

The star tracker is modelled as a quaternion multiplication of the attitude and the sensor quaternion (see equation 4.68):

$$z_{STR1} = q_4 s_1 + q_1 s_4 - q_2 s_3 + q_3 s_2 \tag{5.56}$$

$$z_{STR2} = q_4 s_2 + q_1 s_3 + q_2 s_4 - q_3 s_1 \tag{5.57}$$

$$z_{STR3} = q_4 s_3 - q_1 s_2 + q_2 s_1 + q_3 s_4 \tag{5.58}$$

$$z_{STR4} = q_4 s_4 - q_1 s_1 - q_2 s_2 - q_3 s_3 \tag{5.59}$$

For the UKF, these equations are sufficient. For the EKF, the derivatives with respect to the different states need to calculated. The results can be found in equations to .

For the first implementation of the angle measurements, the simplified model is used for the planet and Sun sensor, as explained in section 5.4.1. The attitude and position, as well as their measurements are kept unrelated. This means the Earth, Moon and Sun sensor are mounted in the inertial frame, instead of on the spacecraft. The star tracker is fixed on the spacecraft and measures the attitude. This way, the implementation of the star tracker can be tested.

For the link to the attitude, the derivatives of the azimuth, elevation and angular size are to be calculated according to:

$$\frac{\partial \alpha}{\partial w} = \left(-x_S \frac{\partial z_S}{\partial w} + z_S \frac{\partial x_S}{\partial w}\right) \frac{1}{x_S^2 + z_S^2} \tag{5.60}$$

$$\frac{\partial \epsilon}{\partial w} = \left(-y_S \left(x_S \frac{\partial x_S}{\partial w} + y_S \frac{\partial y_S}{\partial w} + z_S \frac{\partial z_S}{\partial w}\right) + r_s^2 \frac{\partial y_S}{\partial w}\right) \frac{1}{r_S^2 \sqrt{r_S^2 - y_S^2}} \tag{5.61}$$

$$\frac{\partial \theta}{\partial r_1} = -2 \left(\frac{\partial x_P}{\partial w} x_P + \frac{\partial y_P}{\partial w} y_P + \frac{\partial z_P}{\partial w} Z_P\right) \frac{R_P^2}{r_P^2 \sqrt{r_P^2 - R_P^2}} \tag{5.62}$$

In this equation, w is a position or attitude state. The derivatives with respect to the velocity or rates are zero. For the derivation of these equations and the derivatives, see section **??** and **??**.

The measurements of the position vector are given by:

$$\begin{bmatrix} x_S \\ y_S \\ z_S \end{bmatrix} = \boldsymbol{T}_{SI} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{5.63}$$

where $\boldsymbol{T}_{SI}$ is the transformation matrix from inertial to sensor frame. The derivatives are the same as the ones being used in the equations 5.60 until 5.62 for deriving x_S, y_S and z_S.

Filter Implementation and Assumptions

The first implementation keeps a complete separation between attitude and position. This allows to check if the process for the attitude and the star trackers are properly implemented. For the planet and Sun sensors, the simplified implementation is used and they are linked to the inertial frame. To test the implementation of the star tracker and the attitude, two attitudes are tested: $\boldsymbol{q}_1 = (0.5, 0.5, 0.5, 0.5)$ and $\boldsymbol{q}_2 = (0, 0, 0, 1)$. The first one checks the normal behaviour, the second one whether the filter can handle situations where one of the quaternions goes to a limit. In both cases, the rotational rates are kept at zero. As a result, the attitude quaternions need to remain constant. The field-of-view is not limited.

The second implementation includes the link between the attitude and position. This implementation allows to check if the link between attitude and position works. The field-of-view is not limited to be able to check the complete behaviour with all sensor data. If it is not possible to converge with continuous measurements, it is most likely not possible to converge with a limited field-of-view. The attitude is kept constant. The spacecraft initial position is chosen at:

$$\boldsymbol{x} = \left(\frac{\sqrt{2}}{2} r_0, \frac{\sqrt{2}}{2} r_0, 0\right) \tag{5.64}$$

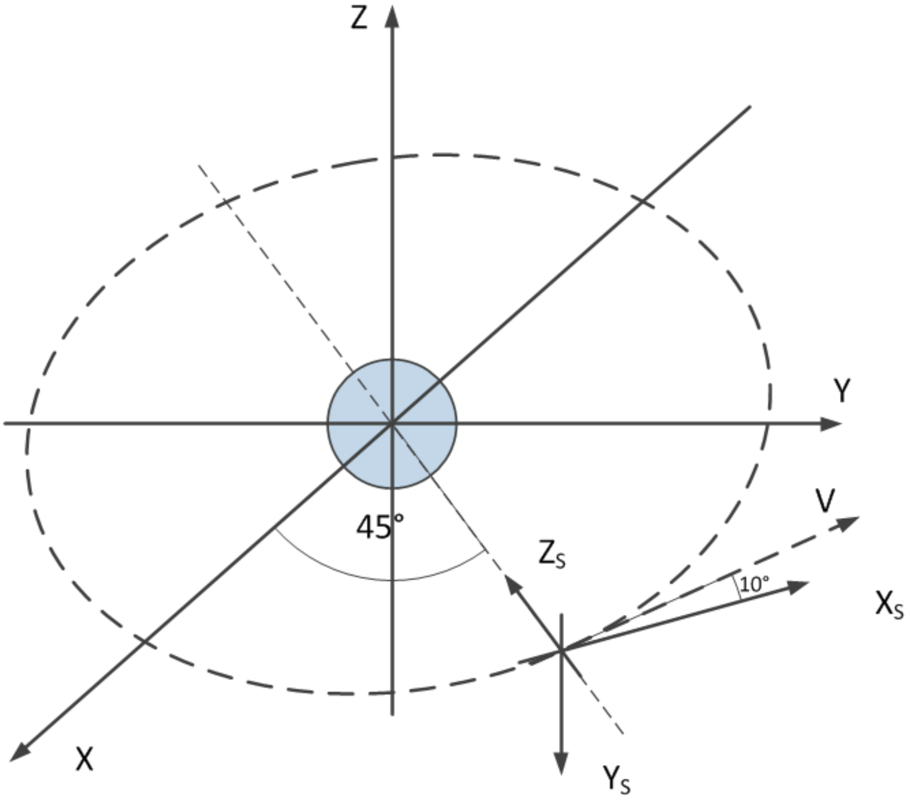

Figure 5.17: Initial position of the spacecraft in Earth orbit

r_0 is taken at 40 000 km. The velocity is computed to form a circular orbit with an inclination of 10°. To have Earth in sight of the sensor, the attitude of the spacecraft is defined by 2 rotations: one about the X-axis with -90° and one about the Y-axis with -135°. This is achieved with the following equations:

$$\boldsymbol{q_{3_1}} = (\sin -90°, 0, 0, \cos -90°) \tag{5.65}$$

$$\boldsymbol{q_{3_2}} = (0, \sin -135°, 0, \cos -135°) \tag{5.66}$$

$$\boldsymbol{q_3} = \boldsymbol{q_1} \cdot \boldsymbol{q_2} \tag{5.67}$$

$$\boldsymbol{q_3} = \left[-0.2706 - 0.65330.6533, 0.2706\right]^T \tag{5.68}$$

This way, the Z-axis of the spacecraft points to the centre of the inertial reference frame, thus to Earth, as shown in figure 5.17. By keeping the Earth sensor in the same direction as the spacecraft-fixed, Earth will initially be at 0° azimuth and elevation. This scenario is tested for the three filter implementations: the EKF with angle measurements, the EKF with position vector measurements and the UKF with angle measurements.

In the final implementation, the field-of-view is restricted, according to the field-of-view of the different sensors. First, the orientation is chosen such that one of the planet sensors has a continuous view on Earth. This is achieved by starting with an initial position in the XY-plane with the ascending node at 45°. The satellite Z-axis points to the centre of Earth, and the Y-axis north. Then the satellite is given a rotational rate which makes the satellite rotate once per orbit about its own axis. The resulting orbit allows the planet sensor to be pointed to Earth continuously. The star tracker is assumed to have continuous observations.

Finally, the same orbit is taken, but with the satellite rotating four times per orbit about its Y-axis in the opposite direction as before. Normally, Earth should pass five time in the field-of-view per orbit during this 'strobing' test. Due to the chosen rotation rate, also the Moon is visible during short passages. The influence of the different objects on the accuracy of the orbits can be analysed and a limited presence of objects in view can be assessed.

5.5.2 Results

Without Link between Position and Attitude

The first step in the implementation of the star tracker is mounting it on the previous system and expand the states. The sensors from the previous system remain fixed in the inertial system to first check the implementation of the propagation and measurements of the star tracker. Since the position is measured independently from the attitude, the results are similar to those obtained in section 5.4.2.

For both $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, the behaviour of the EKF and UKF is similar for measurements and attitude. The measurements and the error on the attitude for the EKF with the initial attitude $\boldsymbol{q}_1$ are given in figures 5.18 and 5.19. For the UKF, only the error on the attitude is given in figure 5.20. The star tracker measurements are not given for the UKF, since these are very similar to the EKF. For both filters, the attitude has a variation between the $3-\sigma$ boundaries, as required. The noise on the measurement for q_4 is higher than the noise on q_1 until q_3, because the rotational noise is twice the directional noise. q_1 to q_3 represent the pointing axis and q_4 the rotation.

The results for $\boldsymbol{q}_2$ (EKF: measurements and attitude; UKF: only attitude) can be seen in figures **??** to **??**. In both cases, the behaviour is similar for the EKF and UKF with q_1 and q_2 having some variation, but with a fairly flat curve for q_3 and q_4 for both attitude and measurements. The reason for the flat curve for q_3 and q_4 is caused by the noise modelling on the measurements. The Z-axis of the sensor frame is the boresight of the sensor. In case of an attitude of $\boldsymbol{q}_2$, the axis on which the error is applicable is the Z-axis, since there is no further rotation between sensor and inertial frame. As a result, the noise results in a rotation close to the Z-axis, so with a small influence on the Z-direction and a large iunfluence on the X- and Y- direction. Since q_1, q_2 and q_3 correspond to the X-, Y- and Z-axis, respectively, the variation on q_1 and q_2 will be larger than on q_3. Finally, q_4 represents the cosine of (half of) the rotation. This means that noise, which causes a small rotation, on a $0°$ angle has almost no influence. Therefore, also this difference results in a flat trajectory. Despite the flat curve, the estimation converges.

Linking Position and Attitude

As a first test to check the implementation of the equations, the measurement matrix is calculated using the complete model, but instead of using the attitude as an input, the

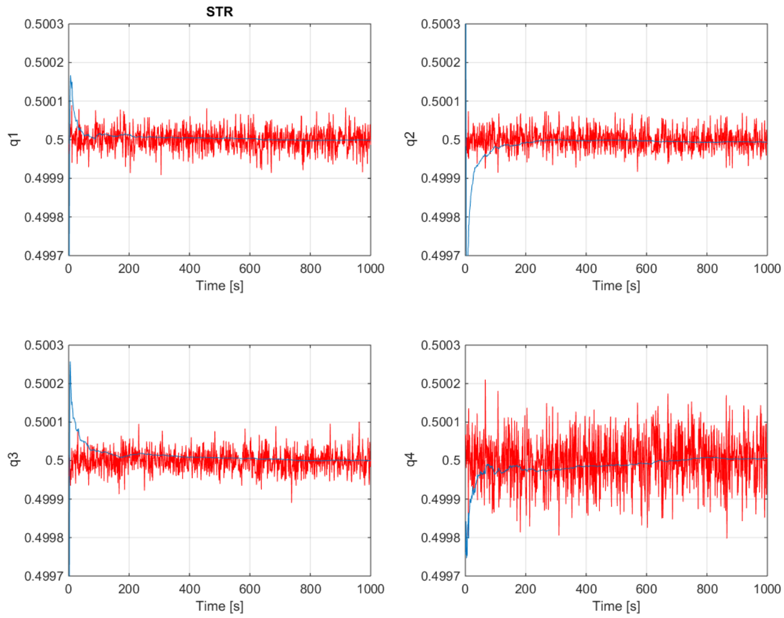

Figure 5.18: Star tracker measurements (sensor (red) and in the filter (blue)) for star tracker with the planet/Sun sensors in the inertial frame and attitude $\boldsymbol{q}_1$.

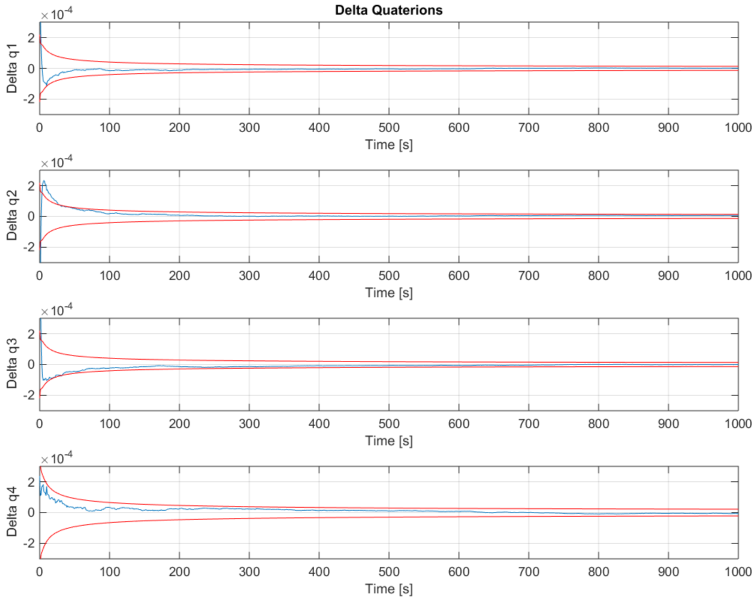

Figure 5.19: Error on the attitude quaternion for the EKF with the planet/Sun sensors in the inertial frame and attitude $\boldsymbol{q}_1$.

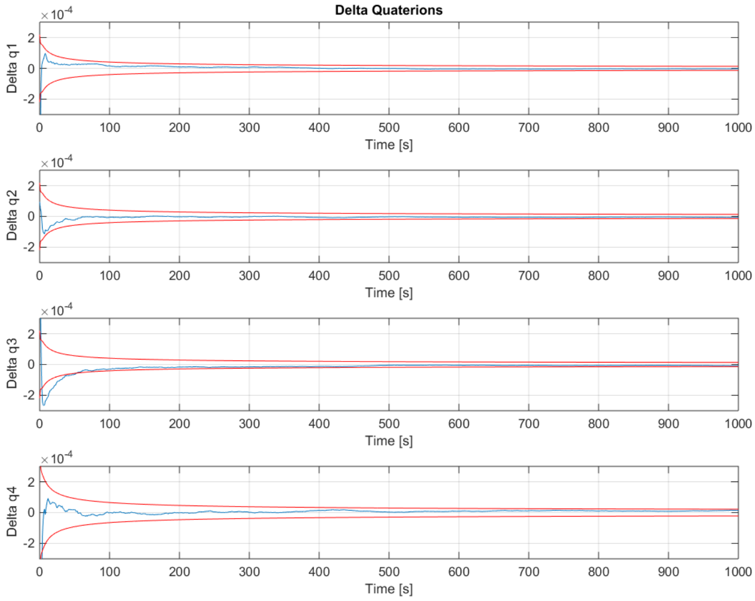

Figure 5.20: Error on the attitude quaternion for the UKF with the planet/Sun sensors in the inertial frame and attitude $\boldsymbol{q}_1$.

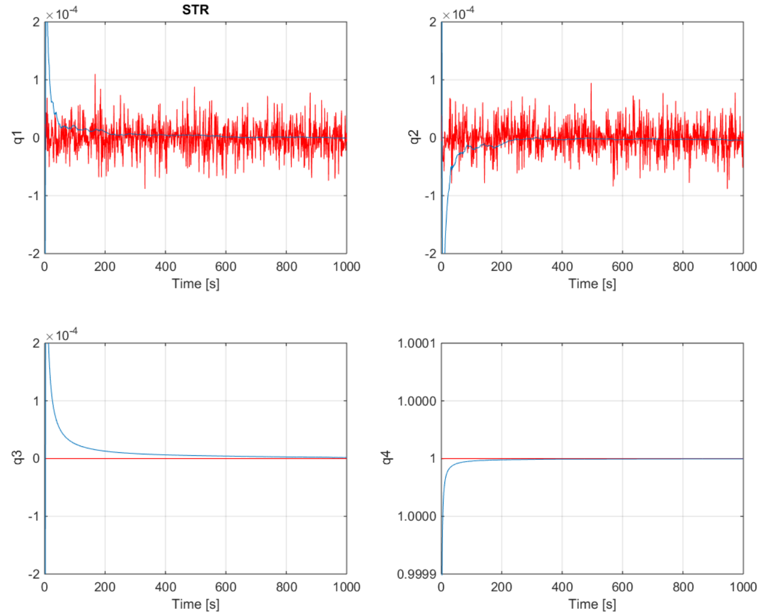

Figure 5.21: Star tracker measurements (sensor (red) and in the filter (blue)) for star tracker with the planet/Sun sensors in the inertial frame and attitude $\boldsymbol{q}_2$. Note the different scale for q_3.

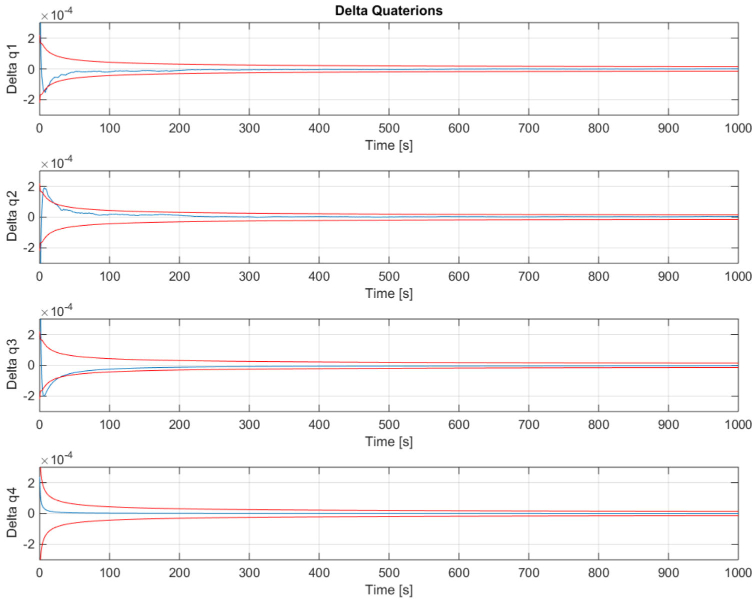

Figure 5.22: Error on the attitude quaternion for the EKF with the planet/Sun sensors in the inertial frame and attitude $\boldsymbol{q}_2$.

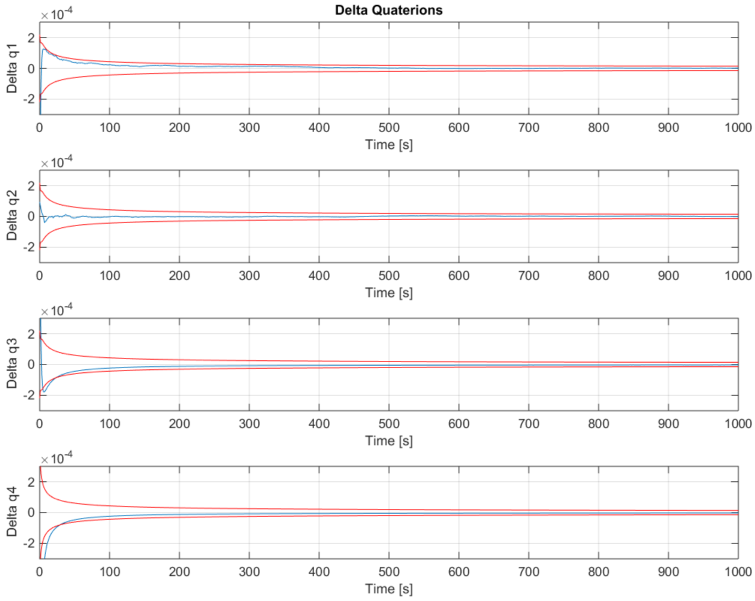

Figure 5.23: Error on the attitude quaternion for the UKF with the planet/Sun sensors in the inertial frame and attitude $\boldsymbol{q}_2$.

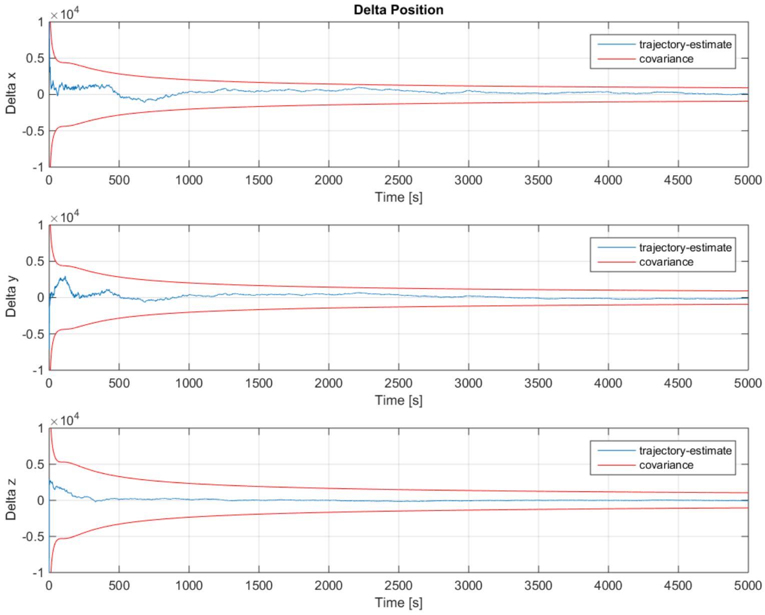

Figure 5.24: Error on the position for the EKF with position vector measurements with attitude $\boldsymbol{q}_3$.

orientation is fixed in the inertial frame. If implemented correctly, the result should be the same as in the previous implementation, since the equations of the simplified model are in principle a special case of the realistic one. The results are the same as in figures 5.21 to 5.23.

The position and attitude errors or the position vector measurements and the UKF with attitude $\boldsymbol{q_3}$ are shown in figures 5.24 to 5.27. In both cases, the differences between simulated and estimated follow the expected behaviour and converge quickly within the $3 - \sigma$ boundaries. For both implementations, a RK4 integrator has been used, since the Euler integration was not sufficiently accurate for convergence. The results of the UKF converge faster than those of the EKF and result in a better accuracy after 5000s. However, the UKF requires about 57s, where the EKF is finshed after 27s using the same computer. In the UKF, a propagation of the 26 sigma points, the generation of the measurements from these and the decomposition of the covariance matrix using the Cholesky decomposition is required. In the EKF, only one set of states needs to be propagated and one set of measurements needs to be generated. This difference can potentially be reduced by modelling the dynamics better.

Another observation on the performance of both filter is the better estimation of the attitude when including the planet sensor and sun sensor measurements. For both the EKF (with position vector measurements) and the UKF, the attitude estimations are about 4 times more accurate when comparing figures **??** and **??** for the EKF and figures **??** and **??** for the UKF.

The results for the position and attitude estimation are shown in figures 5.28 and 5.29. These results are generated using the same integrator as used for integrating the 'real'

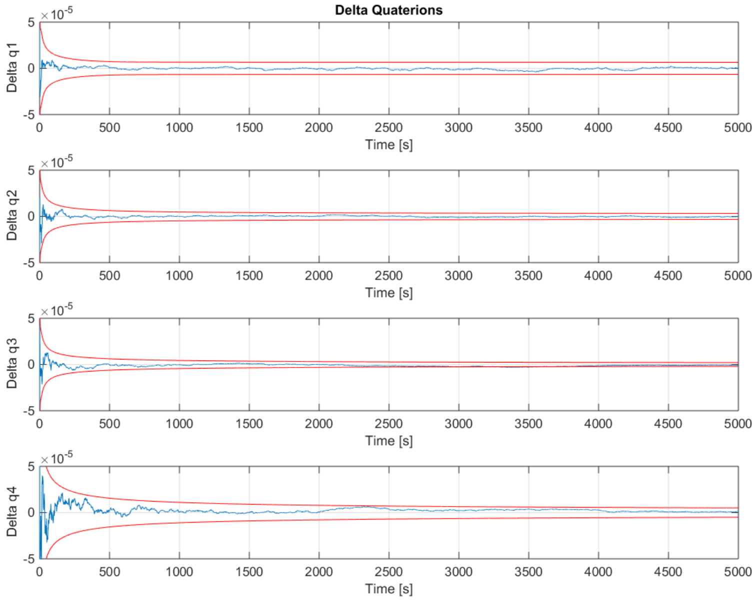

Figure 5.25: Error on the attitude for the EKF with position vector measurements with attitude $\boldsymbol{q}_3$.

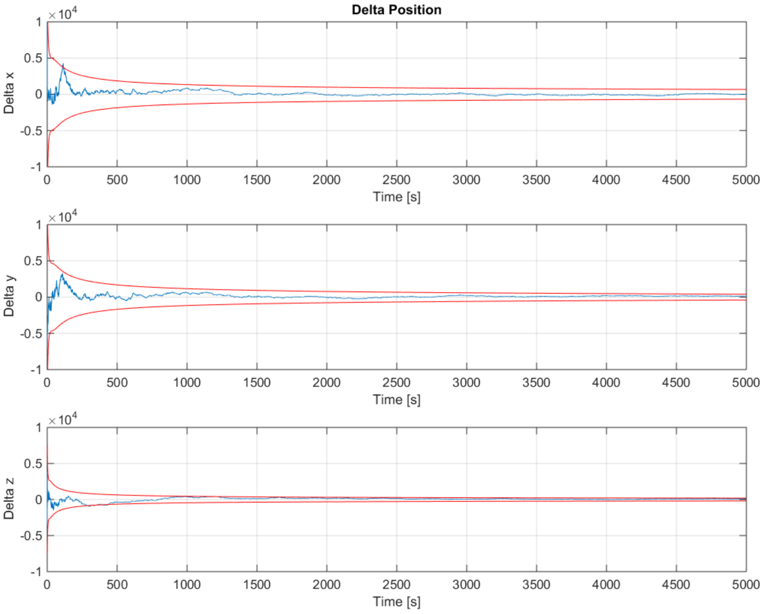

Figure 5.26: Error on the position for the UKF with attitude $\boldsymbol{q}_3$.

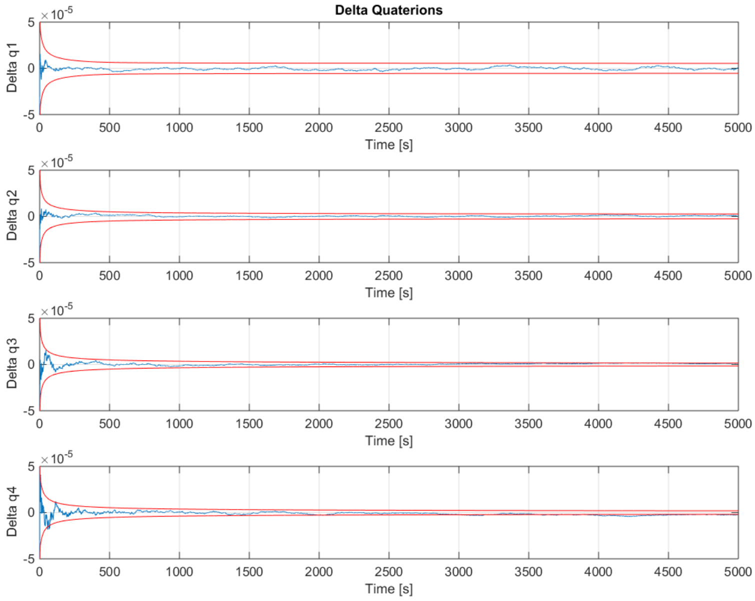

Figure 5.27: Error on the attitude for the UKF with attitude $\boldsymbol{q}_3$.

orbit, the ode45 integrator from Matlab. Despite using this integrator, compared to the RK4 used for the EKF with the position vector measurements, the results are not converging. The derivatives of x_S, y_S and z_S in the equations 5.60 to 5.62 are also used in the position vector measurement matrix. The use of this measurement matrix in the position vector measurements implementation appeared be good, as shown in figures 5.24 and 5.25. Therefore, the cause of the non-divergence can be, either the implementation or the derivation of the azimuth, elevation and/or angular size, or the link between the attitude quaternion and the angular measurements in a linearised measurement matrix. Since the derivation and implementation has been performed multiple times independently from each other, ending with similar results, the conclusion of the author is that the cause is, most likely, the incapability of the EKF to handle the link between attitude and position as it is implemented here.

For identifying the potential causes, the correct behaviour of the UKF and EKF with the position vector measurements is very useful due to commonly used parts. The fundamental matrix and process covariance matrix are the same as being used in the other EKF implementation. The measurements generated by the sensors and the measurement update inside the filter are the same as the ones used in the UKF. The filter equations have been tested in previous simulations. Therefore, the cause must be in what is purely specific to this implementation: the link between attitude and angle measurements in the measurement matrix and/or the use of a diagonal measurement covariance, $\boldsymbol{R}$. The other two filter implementations also use a diagonal covariance matrix. The UKF overlays this with the cross-covariance calculated inside the filter, making it easier to implement with large cross-correlations. The EKF with the position vector reduces the problem to simple vector inputs with a coloured noise. A final reason could be that the derivatives in the

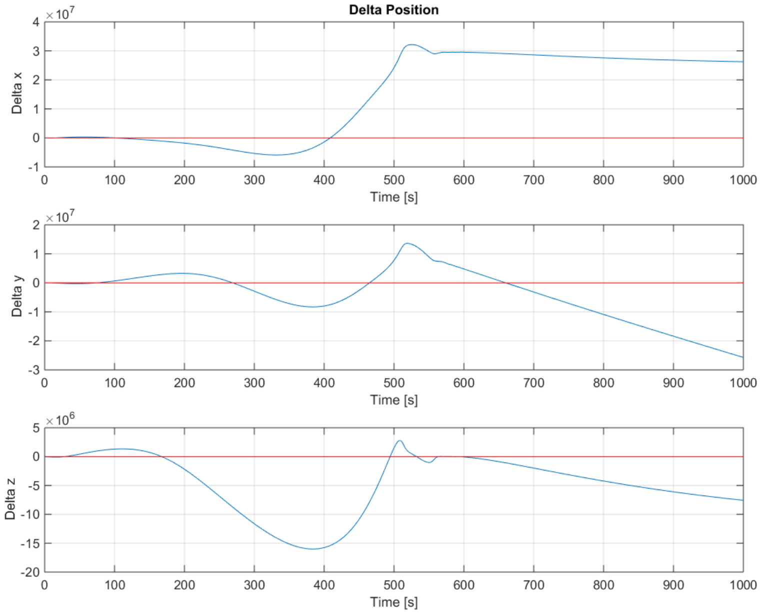

Figure 5.28: Error on the position for the EKF with angle measurements with attitude $\boldsymbol{q}_3$. The red line is the covariance.

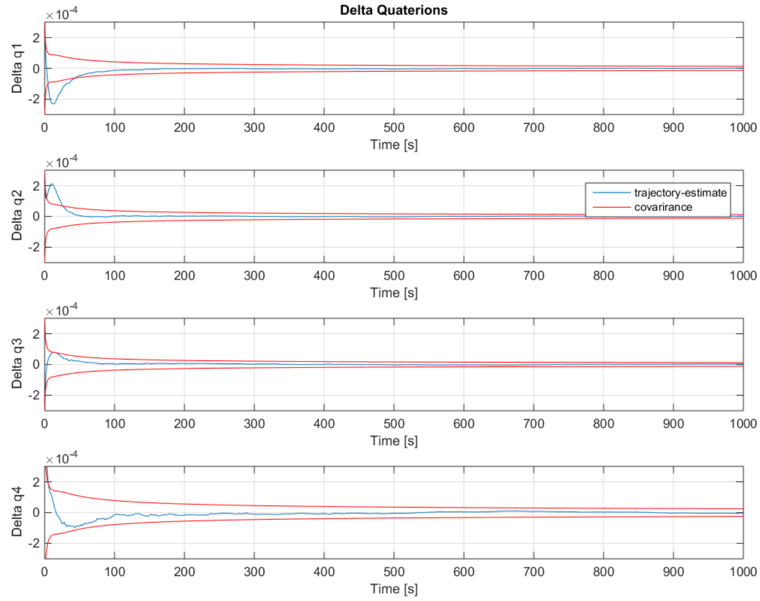

Figure 5.29: Error on the attitude for the EKF with angle measurements with attitude $\boldsymbol{q}_3$. The red line is the covariance.

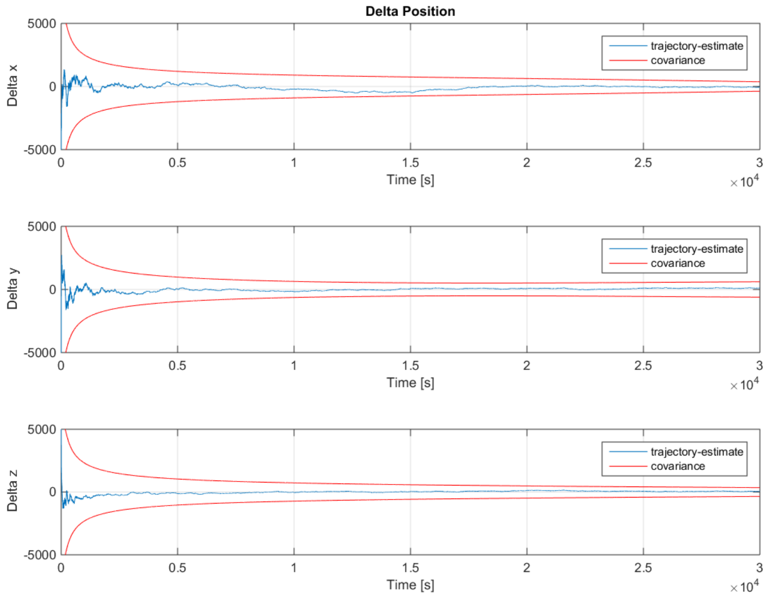

Figure 5.30: Error on the position for the EKF with position vector measurements in Earth pointing configuration.

measurement matrix are too much of a linearisation of the problem and that the EKF is therefore not capable of solving it.

Limited Field-of-View, Continuous Earth Visibility

This test is not performed for the EKF with angle measurements, due to the failed test before. For the other two implementations, the implementation was successful. In figures 5.30 to 5.46, the position and attitude are given for both filters. Since the star tracker and Earth sensor have continuous observations, it is expected both position and attitude converge.

In all previous tests, the attitude was kept constant. The changing attitude can be observed from the star tracker measurements, see figure 5.37. The difference between the filter measurements and the sensor outputs is hard to see and is shown separately in figure 5.38. Due to the nice fit of both the star tracker and attitude measurements, it can be assumed the propagation of the rotational rates is properly implemented. Furthermore, due to the rotational rates and the limited field-of-view, not all measurements are available all the time, as can be observed in the Moon and Sun sensor measurements, see figures 5.35 and 5.36, respectively. When there are no measurements, they become zero. For completeness, the Earth sensor measurements for the EKF are also given in figure 5.34. Due to the missing measurements, the accuracy of bith implementations is slightly worse than before. This is expected, since more information should result in more accurate estimations.

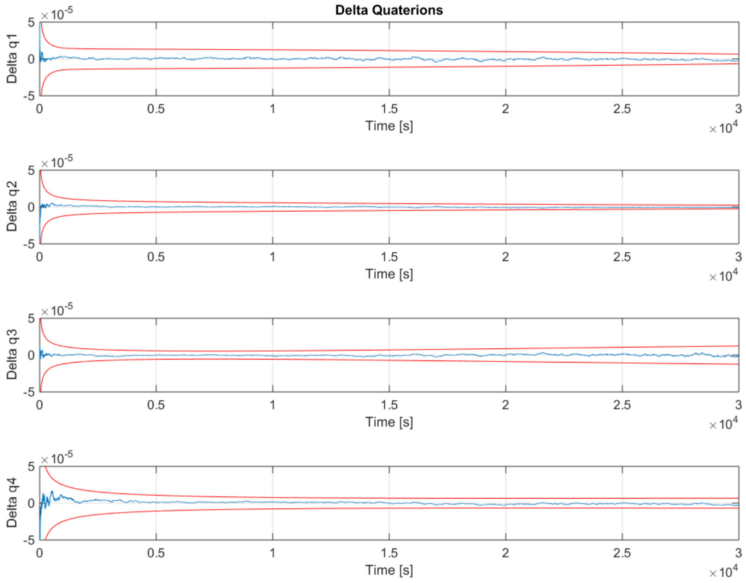

Figure 5.31: Error on the attitude for the EKF with position vector measurements in Earth pointing configuration.

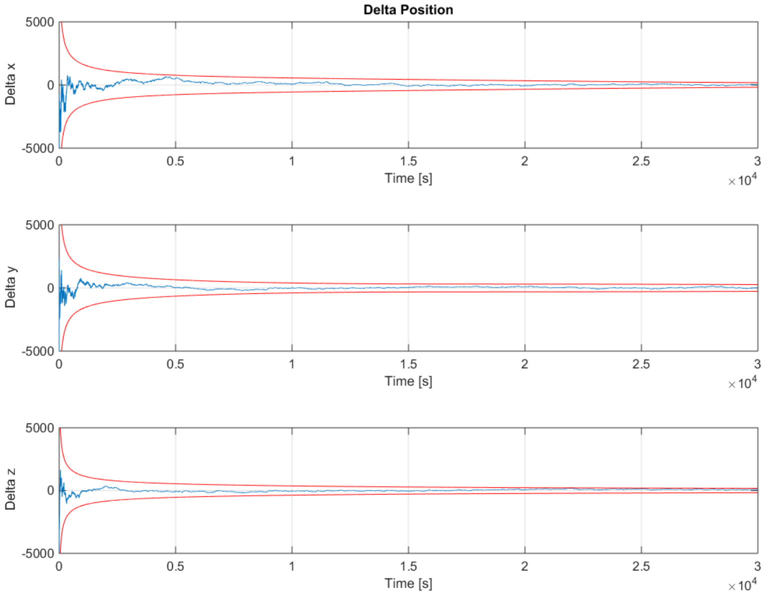

Figure 5.32: Error on the position for the UKF in Earth pointing configuration.

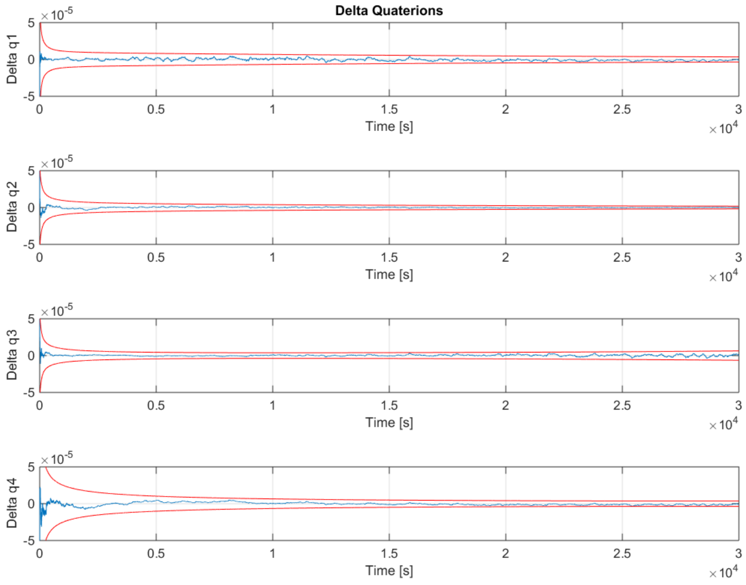

Figure 5.33: Error on the attitude for the UKF in Earth pointing configuration.

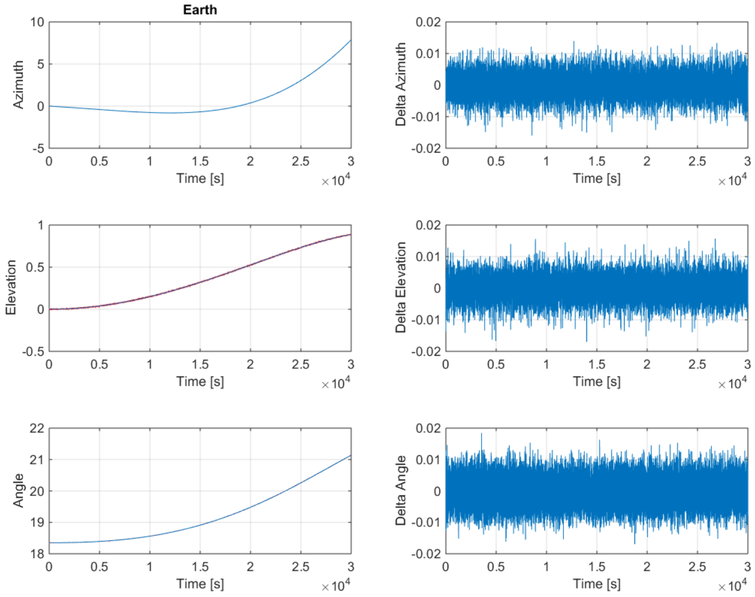

Figure 5.34: Earth sensor measurements for the UKF in Earth pointing configuration. Left: Red is the sensor output and blue the filter measurement. Red: the difference between the sensor output and the filter measurement.

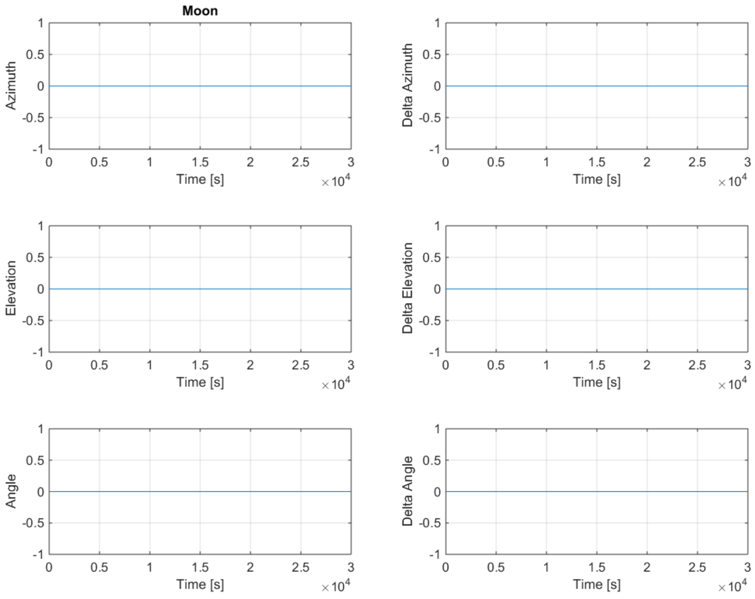

Figure 5.35: Moon sensor measurements for the UKF in Earth pointing configuration. Left: Red is the sensor output and blue the internal measurement. Red: the difference between the sensor output and the filter measurement.

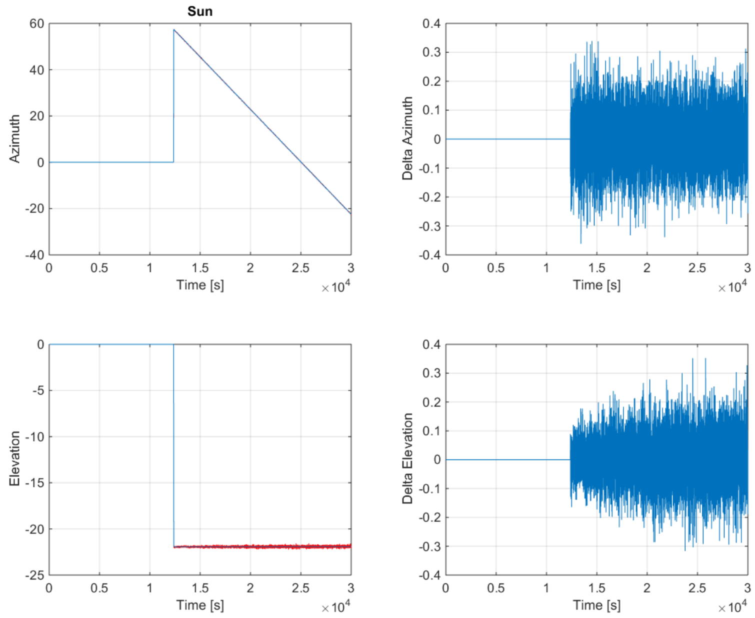

Figure 5.36: Sun sensor measurements for the UKF in Earth pointing configuration. Left: Red is the sensor output and blue the internal measurement. Red: the difference between the sensor output and the filter measurement.

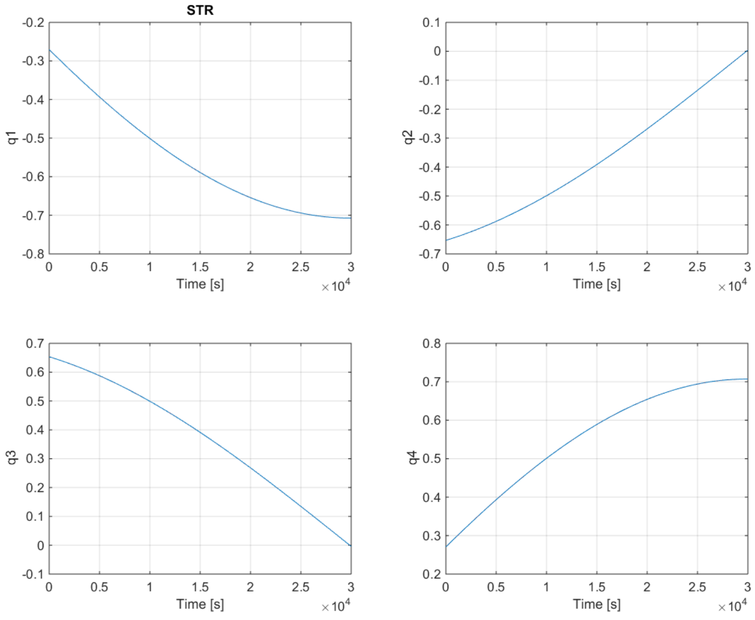

Figure 5.37: Star tracker measurements for the UKF in Earth pointing configuration. Red is the sensor output and blue the internal measurement. The difference is shown in figure 5.38

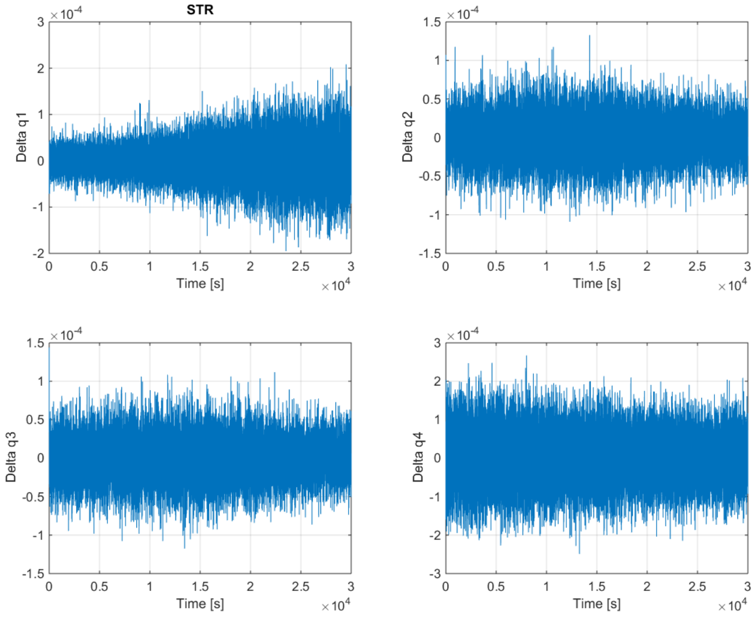

Figure 5.38: The difference between the filter measurements and sensor outputs of the star tracker for the UKF in Earth pointing configuration.

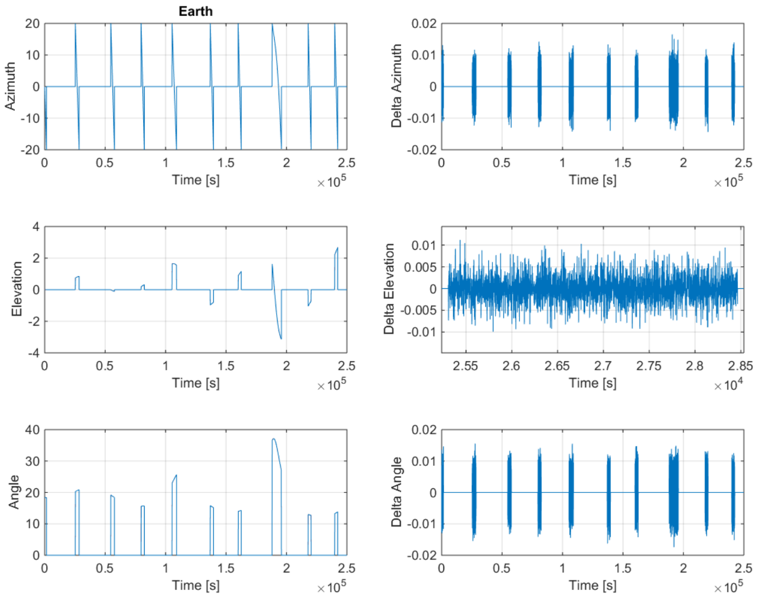

Figure 5.39: Earth sensor measurements for the UKF in Earth strobing configuration. Left: Red is the sensor output and blue the filter measurement. Red: the difference between the sensor output and the filter measurement.

Limited Field-of-View, Earth strobing

This test is performed using the EKF with position vector measurements and the UKF over a simulation time of 250 000s, which is about 3 orbits around Earth. The Earth sensor, Moon sensor, Sun sensor and star tracker measurements for the UKF are given in figures 5.39 to 5.42. The rotation about the Z-axis is clearly visible in the measurements. The observation periods are short, with long parts without measurements. Only the star trackers are assumed to have full view all the time. Here the rotation is clearly visible in the repeating motion of the attitude quaternions. This also results in coloured noise, due to the changing influence of each quaternion, depending on the rotation of the spacecraft.

From the Earth sensor (see figure 5.39), it can be seen that the Earth orbit is being disturbed. At around 190 000s, the angular size of Earth is significatly higher than in the beginning. This reflects itself in longer visibility periods in the azimuth and elevation curves. The plot showing the difference between the filter and the sensor output for the elevation is zoomed in on one of the short periods. The normal noise behavious can clearly be seen. In the elevation and the angular size, the sinusoidal motion of the orbit can be observed, with only short parts of the motion visible due to the limited field-of-view.

A zoom-in on the Moon sensor azimuth (see figure 5.39), shows going through the azimuth from maximum to minimum due to the rotation of the spacecraft along its (initially) north-pointing axis. This same behaviour is also present in the Earth sensor measurements. For the Earth measurements, the azimuth does not go from -20° to +20°, due to the higher elevation and the fact a round field-of-view is assumed. For the Earth sensor, the elevation

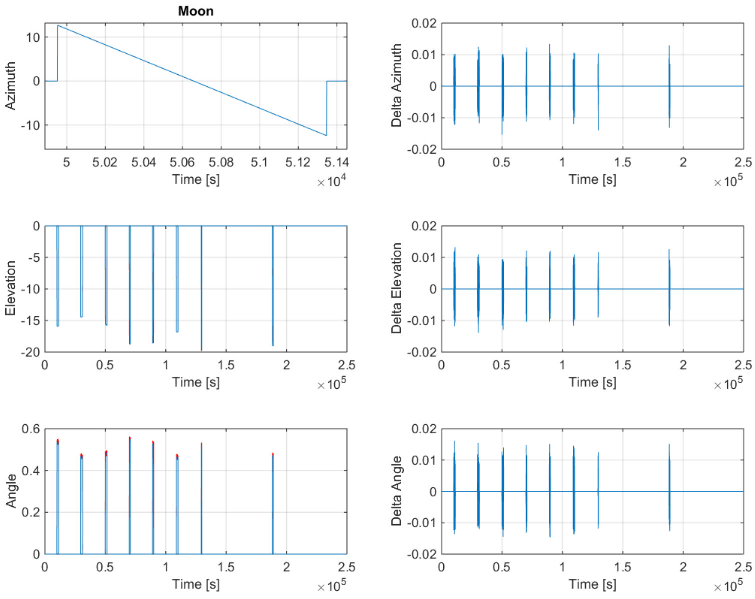

Figure 5.40: Moon sensor measurements for the UKF in Earth strobing configuration. Left: Red is the sensor output and blue the internal measurement. Red: the difference between the sensor output and the filter measurement.

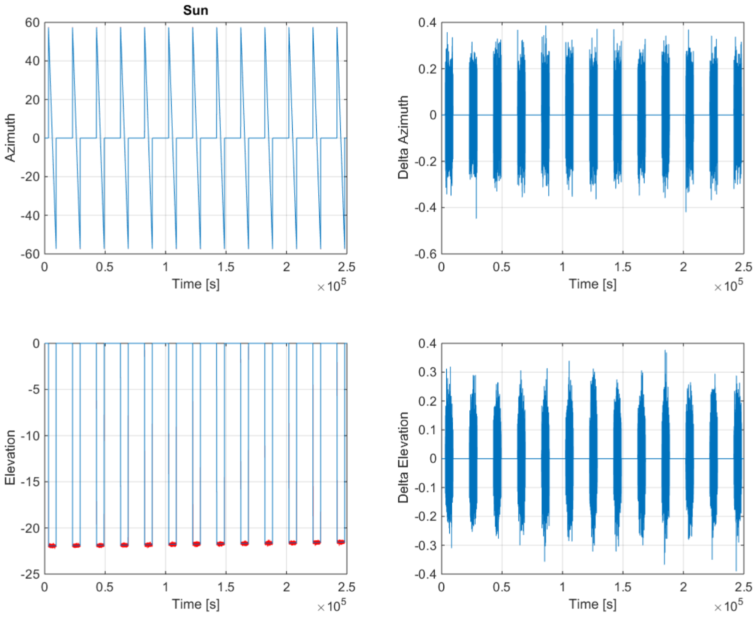

Figure 5.41: Sun sensor measurements for the UKF in Earth strobing configuration. Left: Red is the sensor output and blue the internal measurement. Red: the difference between the sensor output and the filter measurement.

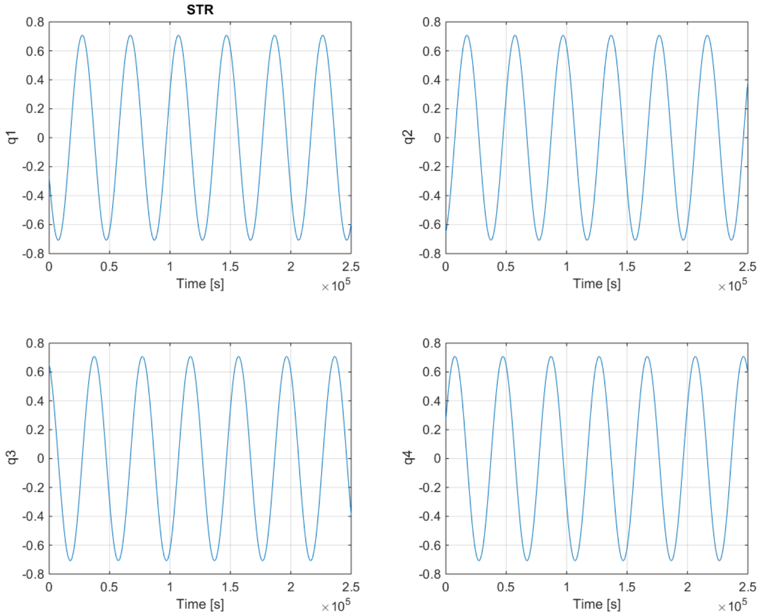

Figure 5.42: Star tracker measurements for the UKF in Earth strobing configuration. Red is the sensor output and blue the internal measurement. The difference is shown in figure 5.43

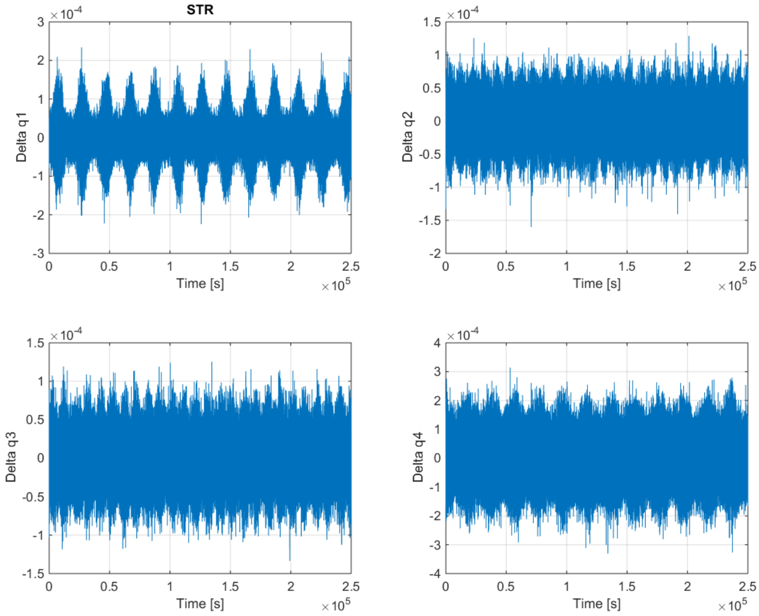

Figure 5.43: The difference between the filter measurements and sensor outputs of the star tracker for the UKF in Earth strobing configuration.

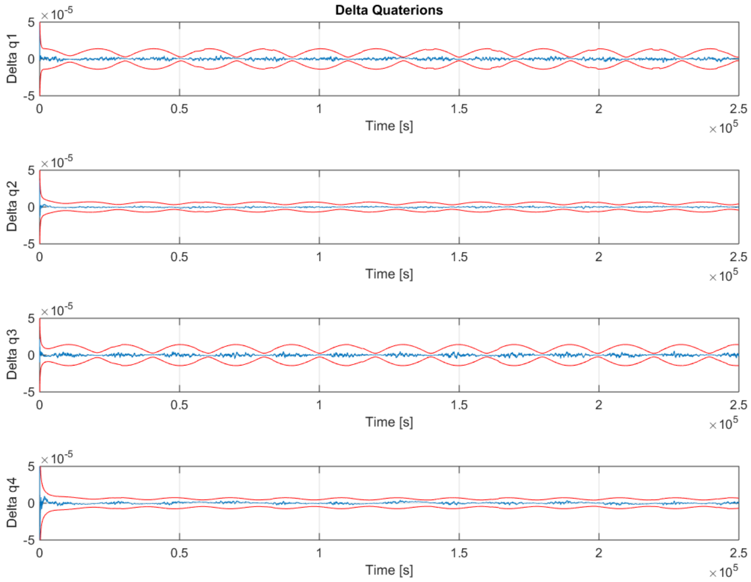

Figure 5.44: Error on the attitude for the EKF with position vector measurements in Earth strobing configuration.

is much lower and, as a result, the elevation uses more of the -20° to +20° range. Due to disturbances on the orbit, the Moon is less visible near the end of the simulation.

The Sun sensor measurements (see figure 5.41) have a much larger error and a larger field-of-view (60° half-cone). Besides the larger field-of-view, the typical saw-tooth behaviour from Earth and Moon sensor is also visible here.

The position and attitude errors of EKF and UKF are given in figures **??** to **??**, with more details on the position errors in figures **??** and **??**. Both configurations show similar behaviour with different performance. In initial part of the position plots, there is a short period of convergence, where Earth observations are available. Then the postion error (and covariance) diverge due to lack of measurements. At about 10 000s, the Moon sensor gets input and a small correction can be observed with the UKF. The EKF is capable of estimating the state slightly better, which can be observed from the discontinuity on the error, but the covariance is not changing. At about 25 000s, the Earth is visible to the Earth sensor again and a big correction is made. After this point, the filter remains in a certain band, where is converges every time observations are available and diverges at the moment non are.

For all quaternion elements, a sine wave motion can be observed in the covariance. This sine wave goes together with the sinusoidal change of the quaternions, as can also be observed in the star tracker measurements (see figure **??**). At certain moments, there are short periods with lower covariance. At these moments, optical measurements are available, which improve the accuracy. This has already been observed in the performance comparison between the inertially placed sensors and the body-mounted sensor comparison.

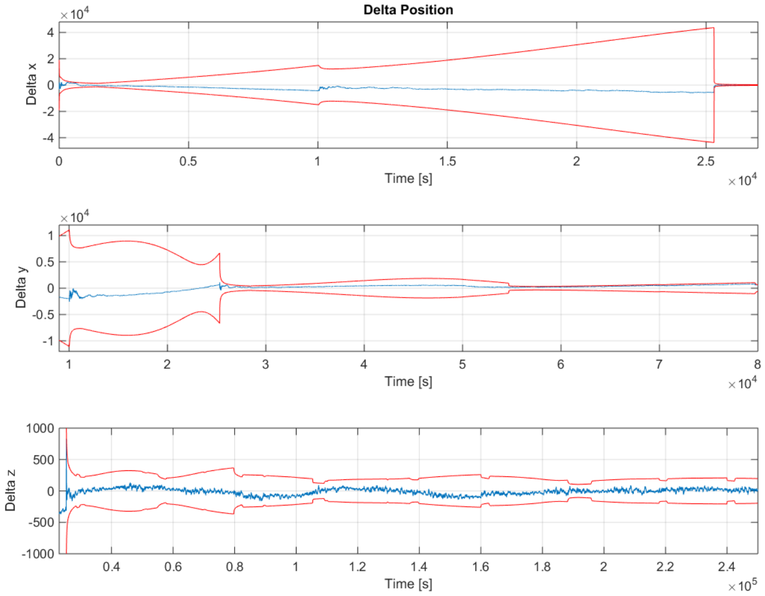

Figure 5.45: Error on the position for the UKF in Earth strobing configuration.

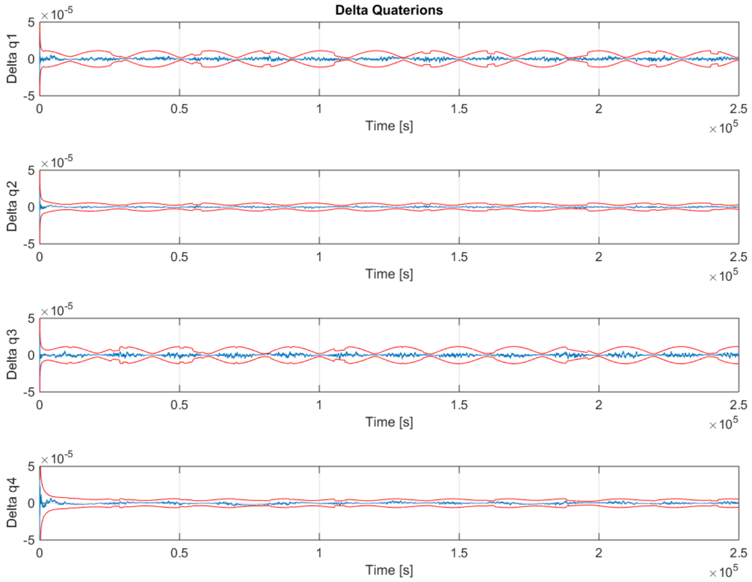

Figure 5.46: Error on the attitude for the UKF in Earth strobing configuration.

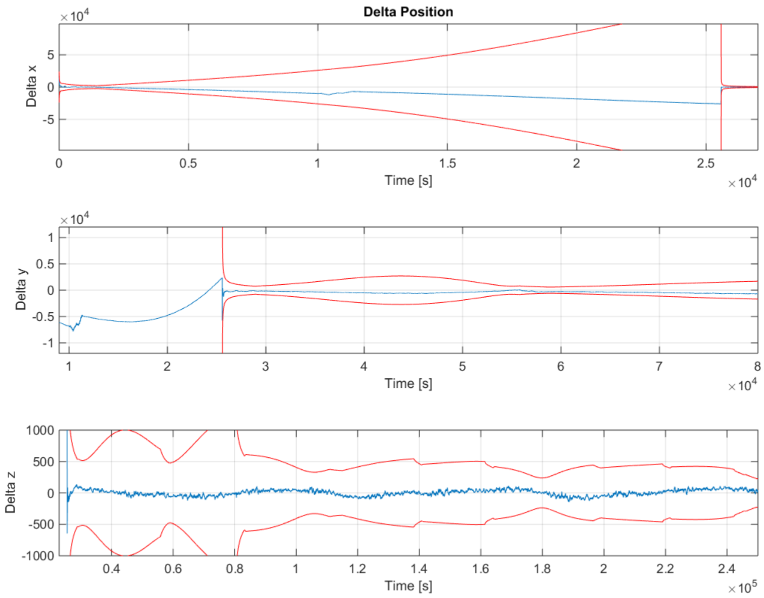

Figure 5.47: Zoomed in on the error on the position for the EKF with position vector measurements in Earth strobing configuration.

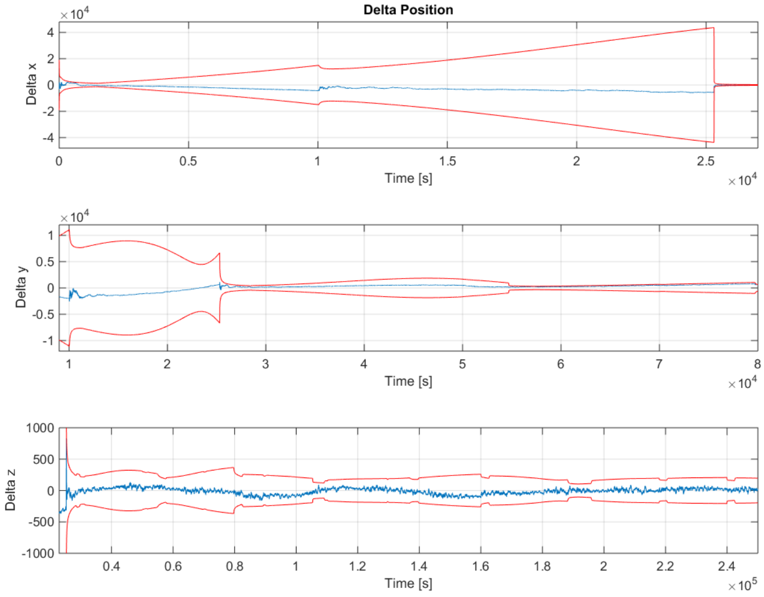

Figure 5.48: Zoomed in on the error on the position for the UKF in Earth strobing configuration.

The performance of the UKF is better than for the EKF. However, the run-time of the EKF is much faster than for the UKF. The state estimation took 469s for the EKF and 1275s for the UKF with a 2.7 GHz processor. Considering that worse processors are generally used on-board, this should leave enough margin for both filters to work. It is probably also possible to optimise both systems for speed by using C++ or better coding. In any case, the UKF will always be slower.

Further tests have been performed using scenarios in the Earth and Moon environment and results remain similar. The UKF provides more accurate results, but the EKF is faster. The UKF is capable of reaching a $3-\sigma$ covariance values of just below 1000m after long term (3-4 days) of stabilisation, whereas the EKF remains below 1500m. If a higher accuracy is required, an active control focused on keeping Earth or Moon inside the field-of-view could be considered. Other solutions can also be considered, like the implementation of knowledge of swarm technology, meaning that calculations of the different spacecraft can be used to improve the position estimation. At that moment, a clear distinction needs to be made on the requirements for relative and absolute position and attitude. Furthermore, the current implementation uses the planet sensors with an accuracy involving two points on opposite sides of the planet. By improving the post-processing of the images, it is possible to improve the quality of the measurements.

Throughout the research, the update time of the filter and the measurements is kept at 1s. By changing the update time to fully use the possible capabilities of a processore, it might be possible to increase the performance of the filter. However, this is all dependent on the available hardware and on the tasks this processing unit needs to perform (*i.e.* purely navigation or also processing of payload measurements and control of the rest of the spacecraft).

Chapter 6

Conclusions and Recommendations

First, general conclusions are being drawn from the results in the implementation section (see chapter 5) in section **??**. Then recommendations for future work are given in section 6.2.

6.1 Conclusions

The general requirements for nano-satellites in a swarm are cooperation, autonomy and flexibility. The main requirements for the attitude and orbit determination is autonomy. Autonomy requires the spacecraft to have the instruments on board to provide sufficient input for an algorithm to determine the attitude and orbit of a spacecraft without the support of ground operations. The case of the spacecraft cooperating in the swarm is not taken into account in this research. Throughout this research, it has been shown that there are sensors available, which can be used to determine the attitude and orbit of a spacecraft in an Earth-Moon transfer orbit.

The selected sensors include a star tracker, a sun sensor and an optical planet sensor. The measurement capabilities of all sensors are sufficient for estimating the orbit, but the lifetime of the star tracker and planet sensor (1 year) are not sufficient for long missions. The shielding of the sensors (including the sun sensor) is been developed for a low-Earth orbit. Therefore, it is possible that these sensors might not be able perform optimally after passing through the Van Allen belts on longer missions.

The algorithms for processing the images from the optical camera to obtain useful outputs for the planet sensors are also not available yet. However, it is not unreasonable to argue that it is possible to do this, with currently available techniques. It can be assumed that the computation power of the star tracker algorithm should be sufficient to perform this calculation.

The sensor outputs are used as inputs for the EKF and UKF. For the EKF, problems arise at the moment the sensors are mounted on the spacecraft and the errors on the attitude become an influence on the performance of the optical instruments. Two potential causes have been proposed: the link between the attitude and measurement matrix and/or the use of a diagonal measurement covariance, $\boldsymbol{R}$. Finding a solution to this convergence could potentially increase the accuracy of the EKF, while keeping the computational advantage of the EKF. For error tracking, the common use of scripts and functions between the UKF and EKF, which are both Kalman filters, allows for error tracking. Using the different implementations of the EKF further improves possiblities of cross-checking the results.

With both EKF and UKF a solution is found. The UKF can use the angle measurements approach, which was unsuitable for the EKF. By post-processing the measurements to position vectors, the noise remains coloured, but the EKF becomes capable of handling the problem. However, there is a difference in accuracy between the filters, which only appears at the moment the connection is made between the attitude and the planet measurements. The accuracy of the UKF is better than 1000m $(3-\sigma)$ for the complete transfer and better than 1500m $(3-\sigma)$ for the EKF after stabilisation. The order of accuracy is good enough to allow for control to go from Earth to Moon. If a higher accuracy is required, an active attitude control system can be used to keep Earth or Moon in the field-of-view of the sensors. Potentially, other inputs can be used, like relative positioning with other spacecraft in the swarm or object scanning on the surfaces of Earth or Moon when approaching.

With the EKF, it is relatively easy to set up a filter with a low complexity in the process and measurements. As soon as there are many dependencies between the different states, the Jacobi matrices need to be derived, costing a lot of time and being very sensitive to errors. The UKF is a bit more work to set up initially, but once a well functioning filter is set up in a more or less generic way, extending it becomes very easy by simply extending a measurement model or the integrator. Adding the robustness to higher-order non-linearities and the higher accuracy, it can be more suited for rapidly testing a system than the EKF. It might lose some time when running, but this is often made up by the ease of implementation.

6.2 Recommendations for Future Work

A first step to improve the navigation system consists of improving and further developing the sensors. The planet sensors have a large potential, especially with the advances in processors (weight, size and power consumption). However, the improvement (or re-evaluation) of the radiation shielding and lifetime improvement of the star tracker, with respect to the one used during this research, is required to ensure the sufficient performance over the longer term. Further improvments in computational power can also allow for improved measurements on the planet in view and perhaps feature scanning on the (lunar) surface.

For the simulation of the system, better models of the sensors and the errors can provide a more accurate view on the behaviour of the filter. Currently, no biases are included in the simulations, so also these could have an influence. The inclusion of the control loop and actuators can also improve the truthfulness of the estimation. The same is true for more accurate simulations of the dynamics, by including higher order gravitational influences, sloshing, disturbance torques, and so on.

The general implementation of the sensors in the filters can also be changes by using other inputs. For example, the Sun sensor is assumed to give an azimuth and elevation as an output to the filter. In reality the outputs are four voltages or currents from which the position of the Sun can be derived. The same is true for the other sensors. By using more realistic models, more realistic results come out of the filters, which could be both better or worse.

For the state estimation in general, there are several possibilities for further research. The use of different state variables has an influence on the performance of the filter [17]. The same is true for the choice of the filter. The use of the unified state model (with 6 or 7 states) or the use of particle or other Kalman filters can increase the accuracy or computational efficiency of the state estimation.

In the EKF, a thourough investigation of the measurements matrix or the measurement covariance matrix could be helpful ot investigate if it is possible for the EKF to converge to the correct position with the angle measurements. The measurements covariance matrix can be derived from the noise of the measurements. However, this noise is included in the angles itself. First a linearisation is required and then a derivation can be made to obtain the covariance. The requirement for doing this is obviously linked to the output that will come out of the planet sensors. If these are provided as a position vector, a further investigation is not required.

To be able to reach a final implementation, a large trade-off needs to be made. Which sensors and measurement techniques can be used in the end (or which ones are developed far enough)? Is there going to be one state estimation algorithm been used throughout the complete mission? When looking at the mission of OLFAR, it might be necessary to have different algorithms for different stages of the mission. The presence of relative positioning in a rapidly changing lunra orbit might be better suited for one estimator compared to another. Another one might be more suited for the more predictable trajectory between Earth and Moon, perhaps even with different state variables. This research might point out certain possibilities, but there are still open issues on the overall mission design.

Appendix A

Unit Testing

A.1 Force and Moment Models

Central Gravity

Third Bodies

Solar Pressure

A.2 Propagators

A.2.1 Position: Cartesian State

A.2.2 Attitude: Quaternions

A.3 Sensor Models

A.3.1 Star Tracker

A.3.2 Sun Sensor

A.3.3 Planet Sensor

Appendix B

Derivations for System Testing and Simulations

B.1 3-D Falling Body

Bibliography

[1] F.A.R. Belien, S. Engelen, A. Noroozi, and C.J.M. Verhoeven. Autonomous navigation for trans-lunar nano-satellite missions. In *Proceedings of the 40th Small Satellites Symposium, Cape Town, South Africa, 3-7 October 2011 (IAC-11,B4,3,3,x10526)*, 2011.

[2] Q.P. Chu. Spacecraft attitude dynamics and control, ae4305: Lecture notes. Blackboard: Delft University of Technology, 2009.

[3] C.W. de Boom, J.A.P. Leijtens, and N. van der Heiden. Micro digital sun sensor: "a matchbox miracle". In *Proceedings of the 6th International ESA Conference on Guidancec, Navigation and Control Systems, Loutraki, Greece, 17-20 October 2005 (ESA SP-606, January 2006).*

[4] Analog Devices. *Tri-Axis Inertial Sensor with Magnetometer*, 2009.

[5] S. Engelen, C.J.M. Verhoeven, and M.J. Bentum. Olfar, a radio telescope based on nano-satellites in moon orbit. In *24th Annual Conference on Small Satellites, 9-12 Aug 2010, Utah, USA*, 2010.

[6] R. Fitzpatrick. Runge-kutta methods, 2006. URL `http://farside.ph.utexas.edu/teaching/329/lectures/node35.html`.

[7] Von Karman Institute. Qb50, an fp7 project, April 2015. URL `http://www.qb50.eu/`.

[8] ISIS. Olfar, April 2015. URL `http://www.isispace.nl/cms/index.php/projects/olfar`.

[9] S.J. Julier and J.K. Uhlmann. New extension of the kalman filter to nonlinear systems. In *Proceedings of AeroSense: the 11th international symposium on Aerospace/Defence Sensing, Simulation and Control*, pages 182–193, 1997.

[10] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASMEJournal of Basic Engineering*, pages 35–45, March 1960.

[11] E. Kraft. A quaternion-based unscencted kalman filter for orientation tracking. In *Proceedings of the Sixth International Conference of Information Fusion, 2003. (Volume 1)*, pages 47–54, 2003.

[12] M. Manzano-Jurado, J. Alegre-Rubio, A. Pellacani, G. Seco-Granados, J.A. Lopez-Salcedo, E. Guerrero, and A. Garcia-Rodriguez. Use of weak gnss signals in a mission to the moon. In *7th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC), 2014*, 2014.

[13] T. Segert, S. Engelen, M. Buhl, and B. Monna. Development of the pico star tracker st-200 design challenges and road ahead. In *25th Annual AIAA/USU Conference on Small Satellites*, 2011.

[14] SSE-group. E-moth project, April 2015. URL `http://www.lr.tudelft.nl/en/en/organisation/departments/space-engineering/space-systems-engineering/projects/e-moth-project/`.

[15] TUDelft. Delfi a nanosatellite development line, April 2015. URL `http://www.delfispace.nl/`.

[16] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. Technical report, Cambridge University Engineering Department, 2000.

[17] V. Vittaldev. Unified state model - derivation and applications in astrodynamics and navigation. Master's thesis, Delft University of Technology, 2010.

[18] K. Wakker. *Astrodynamics-I, ae4874: Lecture notes.* Delft University of Technology, 2007.

[19] K. Wakker. *Astrodynamics-II, ae4874: Lecture notes.* Delft University of Technology, 2007.

[20] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[21] P. Zarchan, H. Musoff, and F.K. Lu. *Fundamentals of Kalman Filtering: Practical Approach (Progress in Astronautics and Aeronautics.* American Institute of Aeronautics & Astronautics, 2005.