

Relative Navigation in Asteroid Missions

Dual Quaternion Approach

B. Razgus

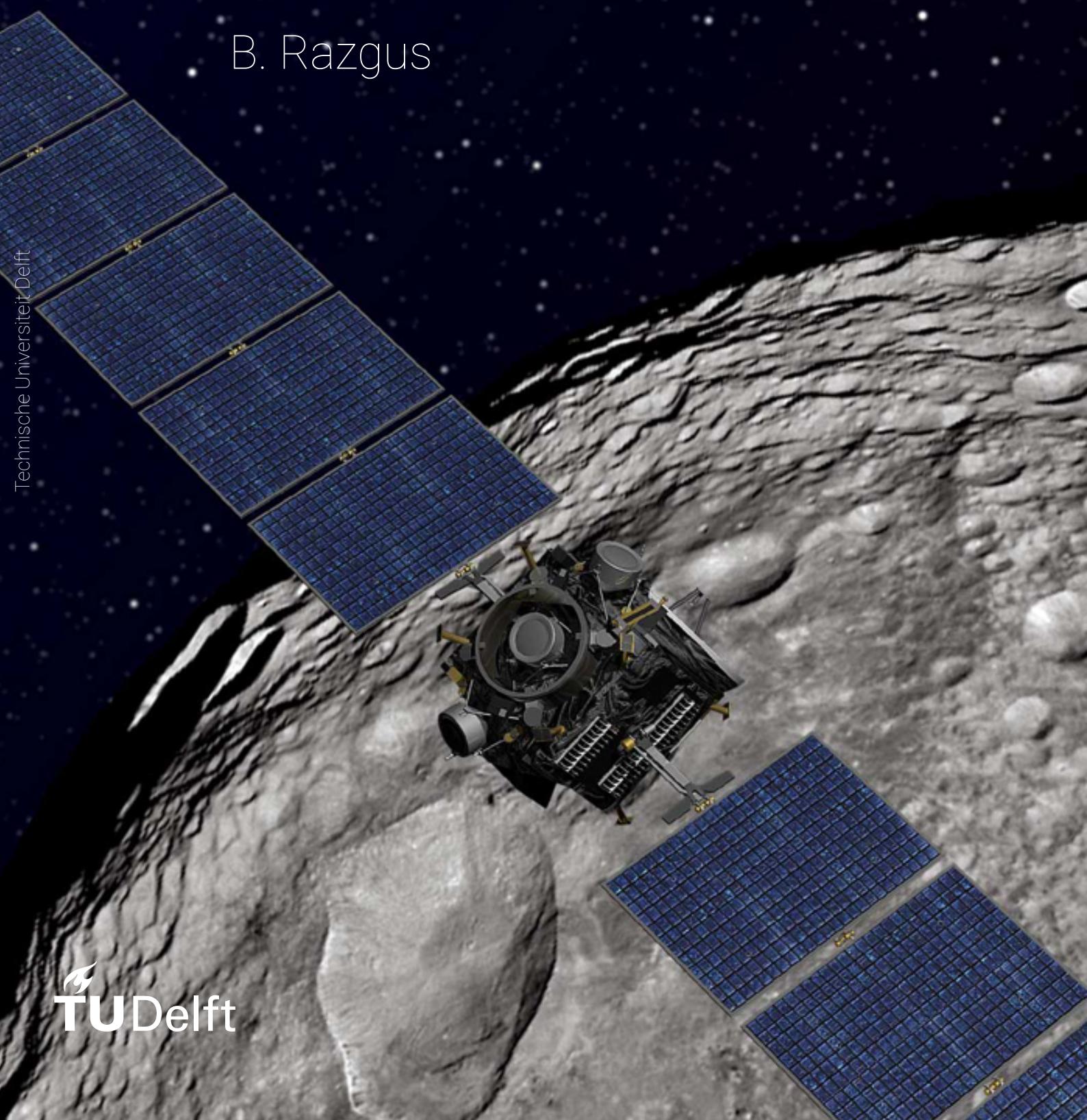


Image credit: NASA

Relative Navigation in Asteroid Missions

Dual Quaternion Approach

by

B. Razgus

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday August 23, 2016 at 09:30 AM.

Student number:	4321677
Project duration:	October 15, 2015 – August 23, 2016
Thesis committee:	Dr. ir. E. Mooij, TU Delft, Astrodynamics and Space Missions, Supervisor Dr. D. Choukroun, Ben-Gurion University of the Negev, Supervisor Prof. dr. ir. P. N. A. M. Visser, TU Delft, Astrodynamics and Space Missions Dr. ir. Q. P. Chu, TU Delft, Control and Simulation

This thesis is confidential and cannot be made public until August 23, 2017.

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Asteroids are gaining more attention among space agencies, due to their preserved state since the beginning of the Solar System. In addition, they could be a potential hazard to our planet or oppositely, another source of resources in the future. Thus, the interest is expected to grow even further. As of August 2016 three missions dedicated to small bodies were accomplished. NEAR visited and landed on the asteroid Eros in 2000. JAXA's mission Hayabusa retrieved samples from the Itokawa asteroid, and ESA's mission Rosetta successfully landed on a comet in 2014. All of these missions required precise navigation relative to the asteroid/comet. Earth-based measurements provide accurate results, however, they introduce a delay, which could be too large for proximity manoeuvres, thus autonomy in navigation is required.

When position and attitude (pose) are required between two non-inertial frames, e.g., spacecraft docking, formation flying, an asteroid mapping/landing, it is common practice to develop navigation algorithms based on relative pose dynamics modelling, as opposed to inertial pose dynamics modelling. The usual way to describe the pose consists of a vector (Cartesian coordinates) and a quaternion. An alternative modelling approach that has recently spawned growing attention models the pose via a dual quaternion, which represents position and attitude in an eight-dimensional vector. As of August 2016, there have been no attempts to implement dual quaternions for navigation in an asteroid mission, which is where this thesis comes in. The main contribution of this thesis is the investigation of dual quaternions as an efficient representation in relative-navigation algorithms for asteroid missions.

This thesis compares two different ways of representing the pose in the extended Kalman filter: a conventional one (Cartesian coordinates for position and quaternions for attitude) and a novel approach (dual quaternions). Moreover, this thesis presents a 'realistic' modelling for dynamics and hardware simulation for missions around small bodies. It includes a polyhedron gravity field modelling, polyhedron gravity gradient torque, navigation camera and laser ranger measurements.

After the development of the two extended Kalman filters, it turned out that they are very alike, which resulted in an identical performance of the filters in the steady-state. However, the transient phase at the beginning showed a larger discrepancy between the two representations. Dual quaternion filter is moderately more accurate in that phase and converges faster than the quaternion-vector counterpart. Moreover, the gravity-field model in the filter turned out to be very important, especially in situations where no or very few landmarks were visible. It was also shown that it is possible to estimate extra parameters such as gyroscope drift and the angular rate of the asteroid.

Preface

Spacecraft landing on an asteroid, asteroid deflection, manned flights to asteroids, and asteroid mining are or will be in the near future a reality. I am pleased to have had an opportunity to work on this exciting research area, and hopefully my work will contribute to the development of these missions.

This thesis report is the culmination of my research carried out in order to fulfil the requirements of the Master of Science degree program at the faculty of Aerospace Engineering, Delft University of Technology. Moreover, it is expected that a conference paper based on this thesis will be published at the AIAA GNC conference in 2017.

I am very grateful to my daily supervisors Erwin Mooij and Daniel Choukroun who had enough patience with me during this long run. With their invaluable guidance this thesis work has become what it is. Erwin, although being very busy, always found a minute (literally) to answer my questions on the go, and helped me often with the cost of his personal time. Daniel, although being a few thousand kilometres apart, was always available for a talk on Skype with his extensive comments about every single detail I had said or written. Our meetings were not very ordinary, since I used to come to Erwin's office and the we would call Daniel on Skype. I have immensely enjoyed these talks, especially the first few minutes, which were usually devoted for chatting, as if three old friends have met.

I am also thankful to Prof. dr. ir. P. N. A. M. Visser and Dr. ir. Q. P. Chu who agreed to be a part of my thesis defence committee. Many thanks to all of the ninth floor people and especially my fellow students who I had a pleasure to work with in the same room. I would also like to thank to family and friends, who did not lose the faith in me totally, although I was one year late with the graduation; and my house mates for understanding my absence at home and absence during houseworks, I appreciate that. The last thanks goes to Enrico, who has been my daily companion for the last two/three years; please do not envy me, your turn will come soon.

I did have a great pleasure doing the research and writing this thesis, and I hope you, the reader, will find this read interesting and useful.

*B. Razgus
Delft, August 2016*

Contents

Abstract	iii
Nomenclature	xi
1 Introduction	1
1.1 Research Questions	3
1.2 Report Outline	3
2 Asteroids and Mission Heritance	5
2.1 Asteroids	5
2.1.1 Asteroids' Locations	6
2.1.2 Shapes and Sizes	6
2.1.3 Rotational Rates	7
2.1.4 Target Asteroids.	7
2.2 Mission Heritance	8
2.2.1 Asteroid Missions	8
2.2.2 Relative Navigation	10
2.3 Mission Requirements/Assumptions	11
2.4 Conclusions.	12
3 Spacecraft Dynamics	13
3.1 Reference Frames	13
3.2 Translational Motion.	13
3.2.1 Constant Density Polyhedron Gravity field	14
3.2.2 Disturbance Forces.	16
3.3 Rotational Motion	16
3.3.1 Attitude Representation	16
3.3.2 Kinematic Equations	20
3.3.3 Attitude Dynamics	20
3.4 Asteroid Dynamics	21
3.5 Relative States	21
4 Navigation	23
4.1 Hardware	23
4.1.1 Gyroscope.	23
4.1.2 Star Tracker	24
4.1.3 Navigation Camera.	24
4.1.4 Laser Ranger	26
4.2 Extended Kalman Filter	27
4.2.1 Linearisation	27
4.2.2 Time Propagation Stage	28
4.2.3 Measurement Update	29
4.2.4 EKF for Quaternions	29
4.2.5 Full EKF Algorithm	30
4.3 Conclusions.	30

5 Dual Quaternions	31
5.1 Dual Numbers	31
5.2 Dual Quaternion	32
5.2.1 Conjugates	32
5.2.2 Dual Quaternions for Pose Representation	33
5.3 Screw Displacement	34
5.3.1 Plucker line	35
5.3.2 Screw Coordinates	35
5.3.3 Example	35
6 Simulator Design	37
6.1 State Vectors	37
6.2 SPICE	37
6.3 Top-Level Software Architecture	38
6.4 Spacecraft Model	39
6.4.1 Dimensions	39
6.4.2 Mass Properties	39
6.4.3 Reflectivity	41
6.5 Asteroid Model	41
6.5.1 Polyhedron Model	41
6.5.2 Landmark Generation	42
6.6 Dynamics Block	44
6.6.1 Translational motion	44
6.6.2 Polyhedron Gravity Field	44
6.6.3 Solar Radiation Pressure Force	53
6.6.4 Rotational motion	54
6.6.5 Gravity-Gradient Torque	55
6.7 Validation	60
6.8 Integrator	61
6.9 Hardware	62
6.9.1 Gyroscope	62
6.9.2 Star Tracker	62
6.9.3 Navigation Camera	62
6.9.4 Laser Ranger	65
6.10 Conclusions	67
7 Navigation Filters	71
7.1 Quaternion-Vector Extended Kalman Filter	71
7.1.1 QVEKF State Vector	71
7.1.2 Non-linear State Variable Model for \mathbf{X}	72
7.1.3 The Discrete-time Linear Perturbation Model	74
7.1.4 Measurement equations	78
7.1.5 Summary	79
7.2 Dual Quaternion Extended Kalman Filter	79
7.2.1 Relative Pose	79
7.2.2 DQEKF State Vector	79
7.2.3 Dual Quaternion Kinematic Equation	80
7.2.4 Linear Perturbation Model for Dual Quaternion	80
7.2.5 Measurement Equations	83
7.2.6 Summary	84

7.3	Improved Gravity-field Model	84
7.3.1	QVEKF	84
7.3.2	DQEKF	85
7.4	Verification	85
7.4.1	First Attempt	87
7.4.2	Second Attempt.	88
7.4.3	Third Attempt.	88
7.4.4	Conclusions	89
7.5	Conclusions.	90
8	Simulations	91
8.1	Scenarios	91
8.2	Filter Tuning	92
8.3	Results.	92
8.3.1	Scenario 1, Kleopatra.	94
8.3.2	Scenario 1, Itokawa.	97
8.3.3	Scenario 2, Kleopatra.	99
8.3.4	Scenario 2, Itokawa.	101
8.3.5	Scenario 3, Kleopatra.	102
8.3.6	Scenario 3, Itokawa.	104
8.3.7	Scenario 4, Kleopatra.	106
8.3.8	Scenario 4, Itokawa.	108
8.4	Monte-Carlo Simulation	111
8.5	KS4 with Improved Gravity Model	114
8.6	Time Consumption	115
8.7	Conclusions.	116
9	Conclusions and Recommendations	117
9.1	Conclusions.	117
9.2	Recommendations.	119
A	EKF parameters	121
A.1	EKF Design Model	121
A.1.1	QVEKF.	121
A.1.2	DQEKF	123
A.2	EKF Tuning	123
Bibliography		127

Nomenclature

Notation

$\check{\otimes}$	Dual quaternion multiplication
$\check{\cdot}$	Dual number/vector
$\hat{\cdot}$	Estimated value
\odot_{S}	Value related to the Sun
\odot	Quaternion multiplication
\oplus	Generic composition
\otimes	Quaternion multiplication
\sim	Measured value

Latin Symbols

A	Area, m ²
C	Direction Cosines Matrix
d	Distance, m
F	Force, N
F	Jacobian matrix of process function f
$f(\mathbf{X})$	Process function
f	Focal length, m
G	Noise mapping matrix
\mathbf{g}	Gravitational acceleration vector, m/s ²
G	Gravitational constant, m ² / (kg · s ²)
$\mathbf{h}(\mathbf{X})$	Measurement function
I	Inertia tensor, kg · m ²
M	Mass, kg
m	Mass, kg
P	Error covariance matrix
Q	Process noise covariance matrix
q	Quaternion
R	Measurement covariance matrix
\mathbf{R}	Position vector, m
T	Torque, Nm
U	Gravitational potential, J/kg
v	Generic measurement noise
V	Velocity vector, m/s
w	Process noise
X	State vector
z	Measurement vector

Greek Symbols

ϵ	Reflectivity coefficient
η	Noise vector
μ	Gyroscope drift, rad/s

μ	Gravitational parameter, m ³ /s ²
ω	Angular velocity, rad/s
ω_f	Solid angle, rad
Φ	Discrete time state transition matrix
ρ	Density, kg/m ³
σ	Standard deviation

Subscripts

A, B, I	Asteroid, Body and Inertial frame respectively
B/I	Value of B frame with respect to I frame
$k+1/k$	At time step $k+1$, given measurements of time step k

Superscripts

$*$	Conjugate
-1	Inverse
T	Transpose

Abbreviations

AIDA	Asteroid Impact and Deflection Assessment
AU	Astronomical Unit
DCM	Direction Cosine Matrix
DQ	Dual Quaternion
DQEKF	Dual Quaternion Extended Kalman Filter
DQMEKF	Dual Quaternion Multiplicative Extended Kalman Filter
EKF	Extended Kalman Filter
ESA	European Space Agency
FOV	Field-Of-View
GG	Gravity Gradient
GNC	Guidance, Navigation and Control
GPS	Global Positioning System
IMU	Inertial Measurement Unit
JPL	Jet Propulsion Laboratory
LIDAR	Light Imaging, Detection, And Ranging
LR	Laser Ranger
LVLH	Local Vertical Local Horizontal
MC	Monte Carlo
MEKF	Multiplicative Extended Kalman Filter
MEMS	Micro-Electro-Mechanical System
NAIF	Navigation and Ancillary Information Facility
NASA	National Aeronautics and Space Administration
NAVCAM	Navigation Camera
NEA	Near-Earth Asteroid
NEAR	Near Earth Asteroid Rendezvous
QVEKF	Quaternion-Vector Extended Kalman Filter
RK45	Runge-Kutta 45
SC	Spacecraft
SPICE	Spacecraft, Planet, Instrument, C-matrix, Events
SRP	Solar-Radiation Pressure

1

Introduction

Relative navigation in Asteroid missions: Dual Quaternion approach. The importance of this thesis problem lies in every single word of the title, so to understand it even better, the title is separated into few blocks: asteroids, relative navigation and dual quaternions. First, let us see why it is essential to explore asteroids. These are rocky/metallic bodies, mainly situated between Mars' and Jupiter's orbit, with no favourable conditions for life, however, still attracting attention, so why? Asteroids are interesting for three different fields:

- **Science.** It is believed that asteroids did not change much since the Solar System formation era, thus they have preserved the conditions, which could help answering the question regarding how our Solar System was formed. Scientific missions as Hayabusa (to asteroid Itokawa) and its successor Hayabusa-2 (which will reach asteroid Ryugu in 2018) were designed to explore asteroids and return samples back to Earth.
- **Defence.** As of July, 2016, there are 1712 potentially hazardous asteroids discovered¹, meaning that they have orbits such that a close approach or even an impact with Earth is possible. One such object is believed to have been the cause of extinction of dinosaurs. As a result, to save our planet from a disaster, there are missions planned to mitigate asteroid orbits in a way such that in the long term it would pass Earth within a safe distance. Asteroid Impact and Deflection Assessment (AIDA), a joint National Aeronautics and Space Administration (NASA) and European Space Agency (ESA) mission, is one of these. It will impact asteroid Didymos to observe the effects on the asteroid's orbit.
- **Resources.** Not in a very far future resources from celestial bodies, such as asteroids, could be exploited. In fact, there are already many companies investigating possibilities to retrieve resources from asteroids. One good example is Planetary Resources, Inc.², which has plans to mine Near-Earth Asteroids (NEAs) rich of metals. Moreover, on 18-11-2015, the US congress passed a law³, which allows US space companies to own and sell natural resources from space bodies, including asteroids. This law is expected to boost private-sector investments into space industry, and, of course, attention to asteroids will increase significantly.

Furthermore, the significance of the navigation system has to be discussed. A Guidance, Navigation and Control (GNC) system, which tells the spacecraft (SC), where it is (Navigation), where

¹<http://neo.jpl.nasa.gov/orbits/>; date accessed: 25-07-16

²<http://www.planetaryresources.com/>; date accessed: 20-01-16

³<https://www.gpo.gov/fdsys/pkg/BILLS-114hr2262enr/pdf/BILLS-114hr2262enr.pdf>; date accessed: 20-01-16

it wants to go (Guidance) and how to get there (Control), is one of the most vital systems on-board a spacecraft. Moreover, the environment in the vicinity of asteroids is extremely perturbed, due to weak and highly irregular gravity fields, the fast rotation of asteroids, and the significant effect of Solar-Radiation Pressure (SRP). This results in the challenging task of keeping the spacecraft on the desired orbit or, moreover, to land it on the surface. GNC is a very vast topic by itself, far beyond the scope of this thesis, so the focus is only put on the navigation part.

The eternal question of every human being is 'Where am I?'. Not going very deep into philosophy, it can be said that this is by far not a trivial question even for a state-of-the-art spacecraft. Navigation concerns about estimating the position and attitude (together called pose) of a spacecraft. In Earth-based missions, the Global Positioning System (GPS), sensors related to Earth's albedo and the magnetic field, and inertial measurement units (gyroscopes and accelerometers) serve for this purpose. In inter-planetary missions, the attitude estimates rely on star trackers and gyroscopes, and the position is tracked by ground stations. However, the tracking signal introduces a delay of up to tens of minutes, which could result in significant position errors, thus autonomy of the navigation system is desired. Moreover, the position and attitude are estimated relative to an inertial reference, which is not always the preferred representation. In an asteroid mission it is useful to know your position and orientation relative to the rotating asteroid frame. For example, when mapping the surface, trying to land or observing the properties of the asteroid, one does not want to know the absolute pose, but the relative one. Then we care about a specific landing point, landmarks, or gravity anomalies at a specific point of the asteroid, but not the inertial position. As a result, the name of 'relative navigation' infers a spacecraft's position and attitude with respect to the asteroid itself.

Since all the aforementioned sensors are not perfect (they have noise, scaling and misalignment, and/or other errors), and they do not necessarily measure position and attitude directly, a technique (an estimator) for filtering the noise and fusing the measurement data is needed. The major breakthrough in developing estimators was done in 1960's, (Zarchan and Musoff, 2009), with the invention of Kalman filter. Kalman filter is a linear quadratic estimator that includes process and measurement models, and thus combines the data from various sensors with the dynamics and noise statistics. The filter works in two steps: first, the state is propagated, and second, when the measurement is available, it is updated. If the process model is accurate enough, it can estimate the state even though the measurements would be lost for a short time. Kalman filter works only with linear systems (which are hard to find in the real world), so new techniques had to be invented to work with non-linear systems as well.

One of the most straight-forward approaches was an extended Kalman filter that simply linearises the process equations around the current estimate. However, the extended Kalman filter is sensitive to tuning and initial conditions. It also introduces linearisation errors, and no off-line analysis can be done to evaluate its performance. Despite all of these drawbacks, it is still one of the most widely used estimators, because it is almost the same as the simple Kalman filter, it is computational efficient, and it works well in practical problems. Moreover, during all of the years of experience, a vast knowledge about the filter and its tuning was gained, therefore, there are many literature sources available which make the development and tuning much easier. In this thesis the history of extended Kalman filter is continued, and it is chosen as an estimator for the navigation system.

Finally, dual quaternions and quaternions themselves have to be explained. Quaternions are four-dimensional vectors comprising attitude in an elegant way. They were first introduced by Sir William Rowan Hamilton more than a century ago, (Fraiture, 2008), then forgotten, and with the space era reborn again.

They gained their popularity due to their simple kinematics equation, compact form (saving memory on board) and being singularity-free. Although it is not easy to visualize a quaternion, however, it is closely related to Euler's theorem, stating that any attitude transformation can be ac-

accomplished by a single rotation around a specific axis, so-called Euler axis. Or in other words, it does not matter how you rotate one frame relative to another, there is always one axis, around which a rotation is performed that brings one frame to another.

Furthermore, dual numbers are an extension to the real numbers, however, it is not that straightforward to explain further. A vector represented with real numbers gives a direction, meaning that you can place it wherever you like, it will still be the same vector. When we construct a vector with dual numbers, it also records its position (starting point) in addition to the direction. If we build a quaternion with dual numbers, we expect to get something that represents not only the attitude, but also the position, (Fischer, 1999). As a result, a dual quaternion is an eight-dimensional vector, having a compact form for the whole pose.

Although it sounds very complicated, a dual quaternion might be seen as the motion of a screw. Imagine two displaced and rotated reference frames. Then, there is always one axis, around/along which a simultaneous rotation and translation performed (screw motion) will bring one frame to another, (Jia, 2013). We realize this may be confusing at this point, so the reader is referred to Chapter 5 for further details.

Now, one could ask, why to confuse ourselves with this complex algebra, and not use simple Cartesian coordinates and quaternions? Well, the author believes that dual quaternions could be in the same situation quaternions were several decades ago - undervalued. Dual quaternions inherited all the nice properties of regular quaternions - a linear kinematics equation (one equation for both translation and rotation) and a compact form. Moreover, dual quaternions have a coupled representation, which means that when having coupled dynamics or sensors measuring attitude and position at the same time (e.g., a navigation camera), one could represent the pose at once with only one vector. The belief is that it could be more computationally efficient and might be an even more accurate representation. That is basically, what this thesis work tries to show.

1.1. Research Questions

The thesis focuses on relative navigation algorithms using dual quaternions in an asteroid mission, which is a novel approach in relative navigation topics, not widely used by scientific community, and which, to the current knowledge of the author, has never been implemented in space missions. The key research question of this thesis is:

How to make relative navigation techniques more robust, reliable and autonomous in asteroid missions?

With the following sub-questions:

1. Would implementing dual quaternions give better accuracy compared to conventional methods?
2. Which mission scenarios are most benefited from using dual quaternions?
3. Are dual quaternions bearing advantages in real-time implementation? What is the time consumption of dual quaternion filters?
4. What is the effect of different target asteroids on filter performance?
5. What extra parameters could be estimated and how?

1.2. Report Outline

The report consists of nine chapters with the first one being the current Introduction chapter. The list of remaining chapters is presented below.

Chapter 2 discusses the nature of asteroids, their locations and properties. It also includes previous missions to asteroids and/or relative navigation techniques used. The chapter results in the selection of two target asteroids for simulations and a set of sensors/navigation techniques to be used in this thesis.

Chapter 3 covers the dynamics of a spacecraft in the vicinity of an asteroid. The governing equations for translational and rotational motion are presented, which are later used for the development of the simulator.

Chapter 4 introduces the concept of the navigation system. It describes the sensors and the models to be used in the simulation process, and presents the work-flow of the Extended Kalman Filter (EKF) for estimating the state.

Chapter 5 explains the nature of dual quaternions. It starts with showing the algebra of dual numbers and proceeds to the implementation of dual quaternions for pose description.

Chapter 6 is where the practical work of thesis starts. It discusses the architecture of the simulator and introduces novel approaches for simulating dynamics and sensors in asteroid missions. The validation of the dynamics is also presented, where the data from Rosetta mission is used to check whether the simulator is working properly.

Chapter 7 presents the development of the two navigation filters: Quaternion-Vector Extended Kalman Filter (QVEKF) and Dual Quaternion Extended Kalman Filter (DQEKF). The development for the both filters starts with defining the state vector, then proceeds with deriving linear perturbation equations and deriving the state transition and measurement matrices.

Chapter 8 is where the results are generated and discussed. It starts with defining orbit scenarios to be used and continues with results and analysis of the two filters. Also, a Monte Carlo (MC) simulation is run to check the statistical behaviour of the filters.

Chapter 9 concludes the report with the outcome of the simulations and presents the recommendations for the future work.

2

Asteroids and Mission Heritage

In the recent two decades the interest in asteroids has been increasing. New asteroids have been discovered by observatories around the world, and the number of space missions to these small bodies has increased as well. This chapter gives a short overview of the properties of asteroids, their locations and missions that observed them.

2.1. Asteroids

In the past, the classification of celestial bodies was simpler. There were the large objects (Sun, planets), the small objects (asteroids and comets) and very small objects (dust, meteorites, etc.). However, with the discovery of a large object in the Kuiper belt in 2005, named Eris (informally called Xena), the debates started about how planets and asteroids should be categorized. This matter was solved in the 26th General Assembly of the International Astronomical Union, in August 2006. The outcome of this was a resolution that Solar System is made of planets, dwarf planets and small objects (comets and asteroids). The formal definitions are as follows, (Dymock, 2010):

A planet is a celestial body that:

- Is in orbit around the Sun.
- Has sufficient mass for its self-gravity to overcome rigid body forces so that it assumes a hydrostatic equilibrium (nearly round) shape.
- Has cleared the neighbourhood around its orbit.

A dwarf planet is a celestial body that:

- Is in orbit around the Sun.
- Has sufficient mass for its self-gravity to overcome rigid body forces so that it assumes a hydrostatic equilibrium (nearly round) shape.
- Has not cleared the neighbourhood around its orbit.
- Is not a satellite.

If a body primarily orbits another Solar System object (not the Sun), it is called a moon or a natural satellite, and, if it is very small, it is often referred to a moonlet. All the other objects that do not fall into these definitions and are not natural satellites, are referred to small Solar System

objects. This change made the former planet Pluto and the former asteroid Ceres to become dwarf planets. Furthermore, an asteroid is defined as a rocky, carbonaceous or metallic body, smaller than a planet and orbiting the Sun. The distinction between a comet and an asteroid is fuzzy. If an object is observed to have a coma, then it is called a comet and when not an asteroid. However, sometimes objects classified as asteroids appear to be comets.

2.1.1. Asteroids' Locations

Asteroids are not distributed evenly in the Solar System, hence according to their location, they are divided into groups (Dymock, 2010):

Vulcanoids Hypothetical asteroids believed to have a circular orbit between 0.07 to 0.21 Astronomical Units (AU) from the Sun. However, after searching for the last 150 years, astronomers have not found a single vulcanoid yet. The name Vulcan was given to a hypothetical planetary object which was believed to be in an orbit closer to the Sun than Mercury.

Near-Earth Asteroid (NEA) NEAs have orbits with perihelion distance less than 1.3 AU, hence close to the orbit of Earth. There are three types of NEAs defined by their orbit: Athene, Apollos and Amors. Athene are Earth-crossing NEAs with semi-major axes smaller than Earth's, therefore they spend most of their time within Earth's orbit. The Apollos have semi-major axes larger than Earth's, hence spend most of their time further from the Sun than Earth, and the Amors always stay outside Earth's (with orbits exterior to Earth's but interior to Mars'). There is a subset of the Athene known as Inner-Earth objects, informally called Apohele, which have orbits within that of Earth's. This makes them hard to detect, because they are always located in the direction of Sun.

Main Belt Asteroids The main belt of asteroids lies between Mars' and Jupiter's orbits, approximately between 2.1 and 3.3 AU from the Sun. The majority of all discovered asteroids are in this belt. Within this belt there are gaps, which correspond to orbital resonances with Jupiter.

Trojans Trojan asteroids have the same orbit as a planet has, but they are located around Lagrangian points (L4 and L5), which are 60 degrees ahead and behind a planet. The majority of discovered Trojans lie in Jupiter's orbit; it is estimated that Jupiter could have about 600,000 Trojans larger than 1 km in size, which is approximately the number of asteroids in the asteroid belt. Mars has four discovered Trojans, Neptune has six Trojans and recent discoveries show that Earth has also a Trojan, called 2010 TK7, located in the L4 point (Connors et al, 2011).

Centaurs Centaurs occupy the region between Jupiter and Neptune. They have orbits with a semi-major axis between 5.5 and 29 AU. Sometimes these asteroids behave like comets, having a small coma of gas around them. Their orbits are significantly influenced by the giant planets due to their large size, hence they can be thrown into inner Solar System or out of it.

2.1.2. Shapes and Sizes

Large celestial bodies with diameters of several thousands of kilometres have their internal strength dominated by their gravity field, hence they result in an equilibrium shape, which in most cases is a sphere or an ellipsoid. A different situation is that of asteroids having sizes from tens of meters to several hundreds of kilometres. It has been shown that asteroids could sustain shapes different from hydrostatic equilibrium (Holsapple, 2001) that are influenced by their spin rates. The spin-rate statistics, which were derived from light-curve data, show that for bodies above 0.15 km an upper limit of the spin rate exists, at which the centrifugal force would overcome self-gravity (Tanga et al, 2009). This results in various shapes of asteroids as can be seen in Figure 2.1¹.

¹http://www.astronomind.com/wp-content/uploads/2011/11/shapes_of_asteroids.jpg, date accessed: 01-07-16

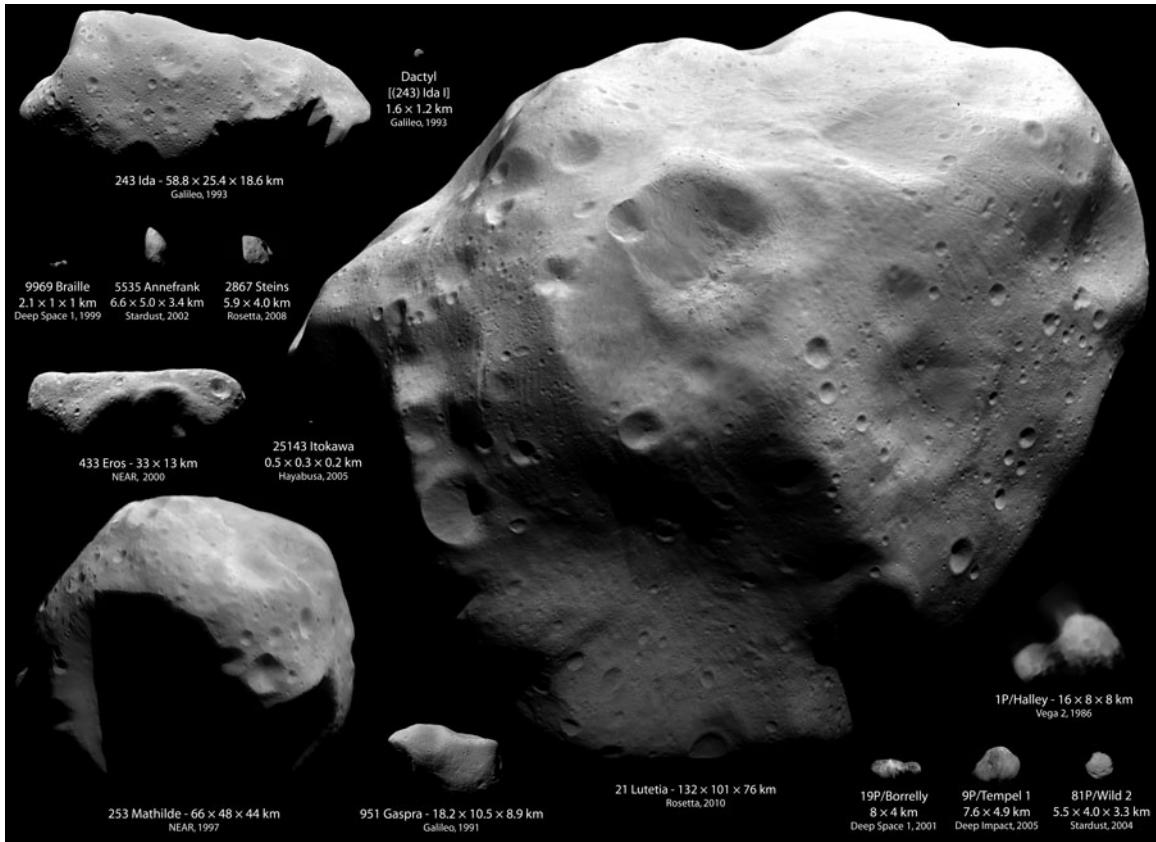


Figure 2.1: Different shapes and sizes of asteroids¹.

Irregular shapes of asteroids result in a perturbed gravity field, which cannot be represented by a central-gravity field model any more. The way to model it is discussed in Subsection 3.2.1. Furthermore, the small mass of the asteroid makes the gravity field weak, thus other non-gravitational perturbations' effects become more significant.

2.1.3. Rotational Rates

Rotation periods of asteroids vary from tens of hours to a fraction of an hour, depending on the size and the structure of the asteroid. As Harris and Pravec (2006) show, there is a lower limit of asteroid periods - a rubble-pile spin barrier, with about two hour period. If an asteroid is spinning faster than this limit, and is made of rubble pile, it would simply disintegrate. On the other hand, there are small asteroids (less than several hundreds of meters diameter), which can spin much faster, suggesting they must have higher tensile strength and are not made of loose regolith.

2.1.4. Target Asteroids

In general, the selection of target asteroids for missions is governed by scientific goals and technical feasibility to reach it. However, for testing navigation algorithms' performance, one would like to choose the most perturbed environment. This would include irregular shapes, fast spinning and/or small size of the asteroid.

For this report, the availability of data also has to be taken into account. Asteroid Itokawa had a dedicated mission to it, thus there exists a significant amount of data about the asteroid properties and navigation algorithms. As a result, it could serve as a valuable target asteroid for this report. However, it is a small asteroid, being less than a kilometre in size; its gravity perturbations might be overcome by non-gravitational ones, thus a larger asteroid would be preferable for com-

parison. In that sense Kleopatra 216 asteroid is a good candidate as well. It is a large (200 km size), dog-bone-shape asteroid, which is often encountered in research papers for gravity-field modelling and orbit design. Although it has not been visited by a spacecraft, there is sufficient data available for using this body. So in the end, two asteroids are selected - Itokawa and Kleopatra for a comparison of a small and a large asteroid mission. The key parameters of the asteroids are listed in Table 2.1².

Table 2.1: Properties of asteroids Kleopatra and Itokawa²

	Kleopatra	Itokawa
Size (km)	$217 \times 94 \times 81$	$0.535 \times 0.295 \times 0.209$
Mass (kg)	4.64×10^{18}	3.51×10^{10}
Rotation period (h)	5.385	12.13
Semi-major axis	2.79 AU (main belt)	1.32 AU (NEA)

2.2. Mission Heritage

In this section, a few of the most relevant asteroid missions are discussed, including the Rosetta mission, which is a mission to a comet, but, due to its similar design, it fits well in this section. Also, a few studies on relative navigation and dual quaternions are discussed.

2.2.1. Asteroid Missions

As of July 2016 three missions dedicated to small bodies were accomplished. NASA's mission Near Earth Asteroid Rendezvous (NEAR) visited and landed on the asteroid Eros in 2000 (Cheng, 2002). JAXA's mission Hayabusa retrieved samples from the Itokawa asteroid, (Kawaguchi et al, 2008), and ESA's mission Rosetta successfully landed on a comet in 2014 (Zuiani et al, 2015). A closer look, how these missions were accomplished is presented below.

NEAR was a NASA mission designed to explore near-Earth asteroid Eros. On its way to the target, the spacecraft passed within 1212 kilometres of the main belt asteroid Mathilde at a speed of 9.93 km/s on 27 June 1997 (Badescu, 2013). In 2000 the spacecraft successfully entered an orbit around asteroid Eros. It was the first attempt of an orbit insertion around an asteroid, which required precise orbit determination, because of the low gravity field of the asteroid. The main problems of navigating NEAR around Eros were the irregular shape of the asteroid and uncertainty of physical properties of it before arrival, which could result in unexpected perturbations of the orbit. To solve these problems the Deep Space Network radiometric Doppler and range tracking was used. In addition to that, new navigation techniques of optical landmark tracking and laser ranging from the spacecraft were implemented (Williams, 2002).

Hayabusa was a mission designed by Japanese Space Agency JAXA to explore asteroid Itokawa, Figure 2.2. It was the first spacecraft to land on an asteroid and return samples back to Earth. In November 2005 two touchdowns were performed, staying approximately 30 min on the surface. There was also a micro-rover designed, which was released from the main spacecraft; however, landing was not successful (Badescu, 2013). The GNC system of Hayabusa had typical absolute attitude sensors, a two-axis Sun-aspect sensor, a star tracker and inertial measurement units, which included an accelerometer that was used for thrust measurements. However, having the need of precise relative position and attitude knowledge, these sensors were not enough. Thus, Hayabusa

²<http://ssd.jpl.nasa.gov/sbdb.cgi>; date accessed: 10-05-16

also had 2 optical cameras for scientific purposes and navigation. Also, it had a Light Imaging, Detection, And Ranging (LIDAR) device for measuring the altitude above the surface. In addition to that, the spacecraft was equipped with a laser-range finder, which served as an altimeter at lower altitudes, besides it could also measure the attitude relative to the surface.

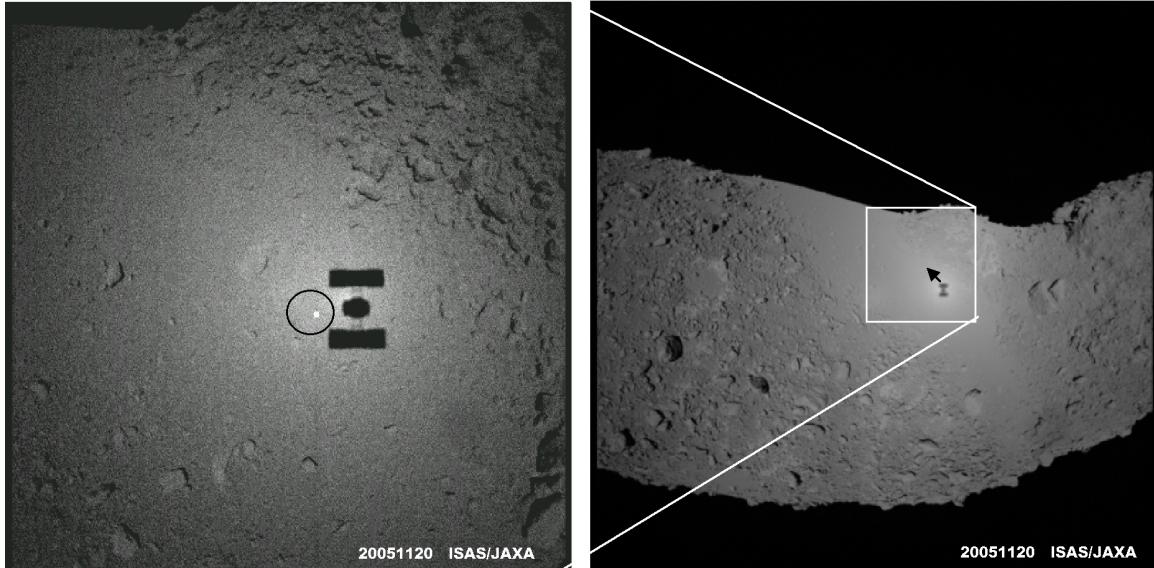


Figure 2.2: Hayabusa's shadow on the surface of Itokawa. Image courtesy: JAXA.

Rosetta was a spacecraft built and launched by European Space Agency (ESA) to study comet 67P/Churyumov-Gerasimenko, Figure 2.3. On 5 September 2008 Rosetta passed by the main belt asteroid Steins at a distance of 800 km and a relative speed of 8.6 km/s. Furthermore, on 10 July 2010, Rosetta flew-by asteroid Lutetia at a distance of 3170 km with a relative velocity of 15 km/s (Badescu, 2013). Philae, the lander module of Rosetta, successfully landed on the comet on November 12 in 2014.

The navigation of Rosetta around the comet was a challenging task due to the uncertainty of the kinematics and dynamics of the comet. All this data had to be gathered during the navigation process itself. Radiometric tracking, although being very accurate, does not provide enough information to know the trajectory and the attitude of the orbiter relative the comet itself. Therefore, Rosetta was equipped with optical cameras to take images of the comet and estimate the directions from the spacecraft to centre of the comet or to a recognizable feature (a landmark) on the surface if it. The first technique (estimating the direction to the centre of the comet) had already been used for Steins and Lutetia fly-bys, while the second one (estimating directions to landmarks) was used for the first time in ESA missions (Munoz et al, 2014).

Rosetta mission had a few phases with different orbits designed. After the comet was reached, the spacecraft entered a comet initial characterization phase. There it flew with a distance of 60-100 km from the comet, and identified first landmarks on the surface of the comet, determined its shape and rotational state, and obtained initial estimated of the gravitational field (Munoz et al, 2014).

Afterwards, the distance to the comet was reduced to 20 km, and the global mapping phase started. During this phase, the objective was to map at least 80% of the surface of the comet and improve the estimates of the navigation parameters to allow entering even closer orbits. Finally, the close observation orbit was initialized with the distances from 5 to 10 km. The candidate landing sites for Philae were selected during this phase (Munoz et al, 2014).

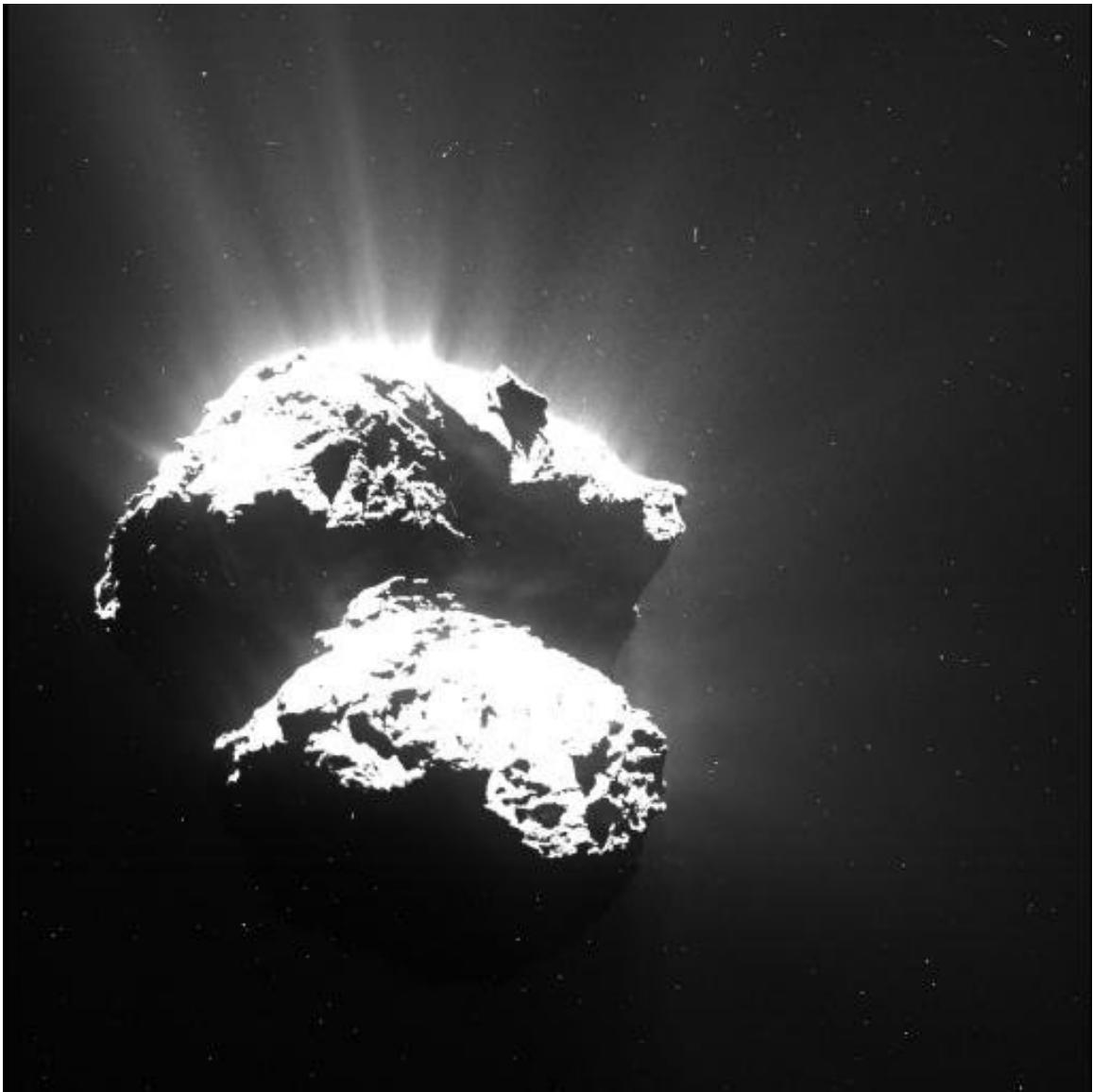


Figure 2.3: 67P/Churyumov-Gerasimenko comet as seen by Rosetta. Image courtesy: ESA.

Hayabusa 2 is a JAXA mission that was launched on 3 December 2014. It will arrive to its target asteroid Ryugu in July 2018. After approximately two years of operations, the spacecraft will return the samples of the asteroid back to Earth. During these two years, Hayabusa 2 will make three sampling attempts in different positions on the asteroid. The spacecraft will come close to the surface of the asteroid, fire a projectile and collect the resulting dust³. All of the sampling attempts will require precise position and attitude knowledge with respect to the asteroid, therefore the spacecraft is equipped with three optical navigation cameras, LIDAR, laser-range finder, and, furthermore, the spacecraft will drop artificial targets close to the sampling zones for navigation purposes.

2.2.2. Relative Navigation

Relative navigation techniques are widely applied in rendezvous, docking or formation-flying missions. In these cases the position and attitude are estimated with respect to another spacecraft or space station. Usually, the concept of target and chaser is used, where the chaser spacecraft navi-

³<http://nssdc.gsfc.nasa.gov/nmc/spacecraftDisplay.do?id=2014-076A>

gates with respect to the target reference frame.

Kim et al (2007) investigated this problem with developing a Multiplicative Extended Kalman Filter (MEKF) for the relative attitude estimation. The research used vision-based measurements to obtain relative pose between the two vehicles. It was shown that it is possible to estimate not only the target states (position, attitude, biases), but also the chaser ones. This is a useful result, which would also be applicable to an asteroid mission, e.g. estimating the rate of the asteroid.

An interesting research was done by Dionne (2009), where the navigation for small-body explorations was discussed. The navigation-camera measurements were augmented with laser-ranger data, which proved to have a significant diminution of the navigation error. Position error, for example, was reduced from two to 5 times with the inclusion of the distance measurement. Also, the laser-ranger errors, due to pointing errors, were introduced, which in certain situations can become large. This type of errors is hard to model statistically, since they are depending on the state of the spacecraft and the irregular shape of the asteroid. In the paper of Dionne (2009), however, a perfect ellipsoid was chosen to represent the asteroid, thus analytical solutions were possible to obtain.

One of the first attempts to adopt dual-quaternions for pose estimation was accomplished by Goddard (1997). In his PhD dissertation, he developed a measurement model for vision-based navigation using the concept of dual lines. He proposed using lines as landmarks instead of points, because lines (dual lines) can be transformed with dual quaternions, and thus a measurement model for both position and attitude is obtained. However, the author did not use the full dual-quaternion vector in the extended Kalman filter.

The Dual Quaternion Multiplicative Extended Kalman Filter (DQMEKF) was first developed by Filipe (2014), where the dual quaternion multiplicative properties were exploited in developing the process equations of the Kalman filter; however, no relative pose measurements were included.

2.3. Mission Requirements/Assumptions

Having the mission heritage discussed, we can draw some requirements and assumptions, upon which this thesis is based. These requirements are:

- The mission shall include high altitude orbits, close observation orbits and asteroid synchronous orbits to explore different dynamics around asteroids.
- The spacecraft shall have mass and size applicable to an asteroid mission as far as 3 AU distance from the Sun.
- The spacecraft shall be equipped with a star-tracker and a gyroscope for the inertial attitude measurements, and with a navigation camera and a laser-ranger for measurements relative to the asteroid.
- The spacecraft shall orbit two different asteroids (Kleopatra and Itokawa) to investigate the performance of the navigation system in different environments
- The spacecraft shall run two navigation filters for each orbit and compare their performance.
- The navigation system shall estimate the relative states and/or auxiliary parameters.

The assumptions read as follows:

- The asteroid's orbit around the Sun is not simulated, since the simulation time is only 10,000 s.
- The asteroid has a constant angular velocity, which is known with an uncertainty for the spacecraft.

- The spacecraft has a perfect knowledge of landmarks and their positions on the asteroid, assuming an a-priori phase that did the global mapping.
- Position and attitude are not controlled during the simulation.
- All sensors have no misalignment or scaling errors.
- No image processing is done, it is a black-box model that outputs noisy pixel coordinates of landmarks.

2.4. Conclusions

In this chapter the properties of asteroids and a few asteroid missions were discussed. There were two asteroids selected for simulations later on. These were Kleopatra and Itokawa. From the missions to the small bodies, we saw that all of them were equipped with navigation cameras. Furthermore, according the work of Dionne (2009), it is advantageous to have laser-ranger measurements as well. As a result, the spacecraft for this thesis will be equipped with a navigation camera and a laser-ranger, in addition to a star-tracker and a gyroscope, which are the must for all of the interplanetary missions.

3

Spacecraft Dynamics

This chapter presents the dynamics of a spacecraft in the vicinity of an asteroid. Section 3.1 introduces reference frames, which are the essential elements in discussing translational and rotational motion. Since the gravity force is the main force acting on the spacecraft, it is discussed in a greater detail afterwards, in Section 3.2. The polyhedron gravity model is used, because it is the most accurate model one can get for an arbitrarily shaped asteroid. Eventually, the attitude representations, kinematic and dynamic equations are given in Section 3.3. Moreover, a short look to the dynamics of the asteroid is given in Section 3.4, and the way to calculate the relative states is presented in Section 3.5.

3.1. Reference Frames

Three reference frames are used throughout this thesis:

Inertial reference frame I , which, for simplicity, is assumed to have the origin in the centre of mass of the asteroid, and has the axes inertially fixed. It is therefore a non-rotating and non-accelerating frame.

Asteroid reference frame A , this is a rotating, asteroid-fixed, frame, which has the axes coinciding with the principal moments of inertia of the asteroid, and, for a flat-spinner asteroid, the Z-axis coincides with the rotation axis.

Body reference frame B , which is a spacecraft-fixed frame that has axes coinciding with the principal moments of inertia of the spacecraft (SC). Its Y-axis is along the direction of the solar panels, the positive Z-axis nominally points nadir, because the navigation camera and laser ranger are situated there, and the X-axis completes the right-handed triad.

All the three frames are depicted in Figure 3.1.

3.2. Translational Motion

The dynamics is simulated in the inertial frame, because, essentially, the equations of motion, based on Newton laws, hold only for inertial reference frames. Moreover, it is more trivial to use this reference frame for debugging purposes, for example, if one simulates a Keplerian orbit in the central gravity field, one expects to get a perfect conic shape. However, if it was simulated in a rotating frame (e.g., the asteroid frame), then it would become a more complex trajectory, e.g. a spiral. Therefore, the inertial reference frame is used, and the equations of motion in this frame are written in the following form.

$$\dot{\mathbf{R}}_I = \mathbf{V}_I \quad (3.1)$$

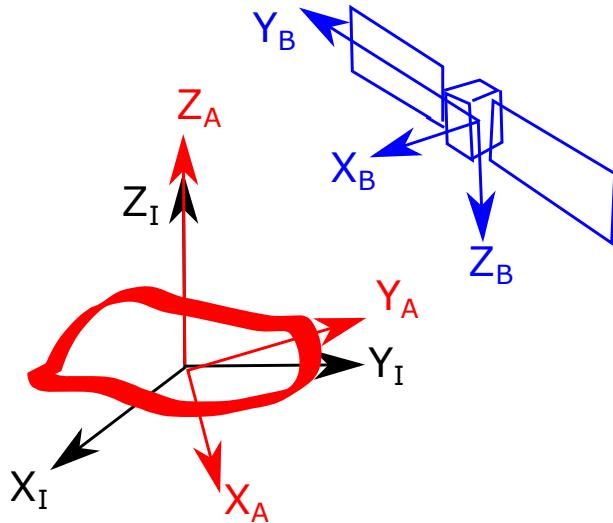


Figure 3.1: Reference frames.

$$\dot{\mathbf{V}}_I = \mathbf{g}_I + \mathbf{a}_{dist} \quad (3.2)$$

where \mathbf{R} , \mathbf{V} and \mathbf{g} are position, velocity and gravity field vectors in the I frame respectively. \mathbf{a}_{dist} is a perturbing acceleration, due to non-gravitational disturbances.

3.2.1. Constant Density Polyhedron Gravity field

Since the gravity field of an asteroid is highly irregular, a central gravity-field approximation would be inaccurate. Having a three-dimensional surface model (polyhedron) of an asteroid and assuming constant density, the most accurate way to simulate it is a constant-density polyhedron gravity model, that was first introduced by Werner and Scheeres (1997).

A polyhedron is a three-dimensional solid body whose surface consists of planar faces meeting along straight edges or at isolated points called vertices. Exactly two faces meet at each edge. Three or more edges and a like number of faces meet at each vertex. The accuracy of gravitational model depends on the number of faces in the polyhedron. As an example, the polyhedron models of Kleopatra (top) and Itokawa (bottom) are given in Figure 3.2 with 4092 and 3688 faces, respectively.

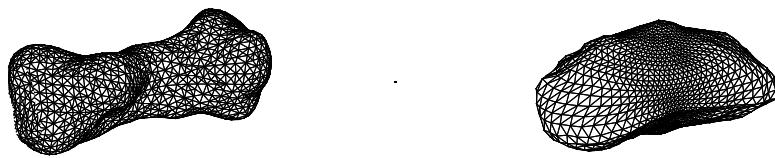


Figure 3.2: Polyhedron models of Kleopatra (left), Itokawa in scale(a dot in the center) and Itokawa not in scale (right).

A polyhedron gravity model has an advantage when the shape of a body is highly irregular, e.g., Kleopatra asteroid, which looks like a dog's bone. The model is able to estimate the gravity potential everywhere in the space around the body. The full derivation of the potential function is given by Werner and Scheeres (1997) and only necessary equations are presented below. The potential function, U , can be written:

$$U = \frac{1}{2}G\rho \sum_{e \in edges} \mathbf{r}_e \mathbf{E}_e \mathbf{r}_e L_e - \frac{1}{2}G\rho \sum_{f \in faces} \mathbf{r}_f \mathbf{F}_f \mathbf{r}_f \omega_f \quad (3.3)$$

where G is the gravitational constant, ρ is the mean density of the asteroid (assumed to be constant), \mathbf{r}_e and \mathbf{r}_f are the distances from a field point (in this case the spacecraft) to an edge and a face respectively. \mathbf{E}_e , a dyadic matrix of an edge, \mathbf{F}_f , an outer product of a face normal, are calculated as follows:

$$\mathbf{E}_e = \mathbf{n}_A (\mathbf{n}_{12}^A)^T + \mathbf{n}_B (\mathbf{n}_{21}^B)^T \quad (3.4)$$

$$\mathbf{F}_f = \mathbf{n}_f (\mathbf{n}_f)^T \quad (3.5)$$

where \mathbf{n}_A , \mathbf{n}_B and \mathbf{n}_f are unit vectors normal to faces A, B and a general face, respectively, see Figure 3.3 on the right. The unit vectors \mathbf{n}_{12}^A and \mathbf{n}_{21}^B are normal to the edge and the associated face normal. Furthermore, the dimensionless factors L_e and ω_f in Eq. (3.3) are defined as follows:

$$L_e = \ln \frac{r_i + r_j + e_{ij}}{r_i + r_j - e_{ij}} \quad (3.6)$$

$$\omega_f = 2 \arctan \frac{\mathbf{r}_i \cdot \mathbf{r}_j \times \mathbf{r}_k}{r_i r_j r_k + r_i (\mathbf{r}_j \cdot \mathbf{r}_k) + r_j (\mathbf{r}_k \cdot \mathbf{r}_i) + r_k (\mathbf{r}_i \cdot \mathbf{r}_j)} \quad (3.7)$$

Vectors \mathbf{r}_i , \mathbf{r}_j and \mathbf{r}_k are shown in the left of Figure 3.3, and r_i , r_j and r_k are the magnitudes of them, respectively. e_{ij} is the length of the edge.

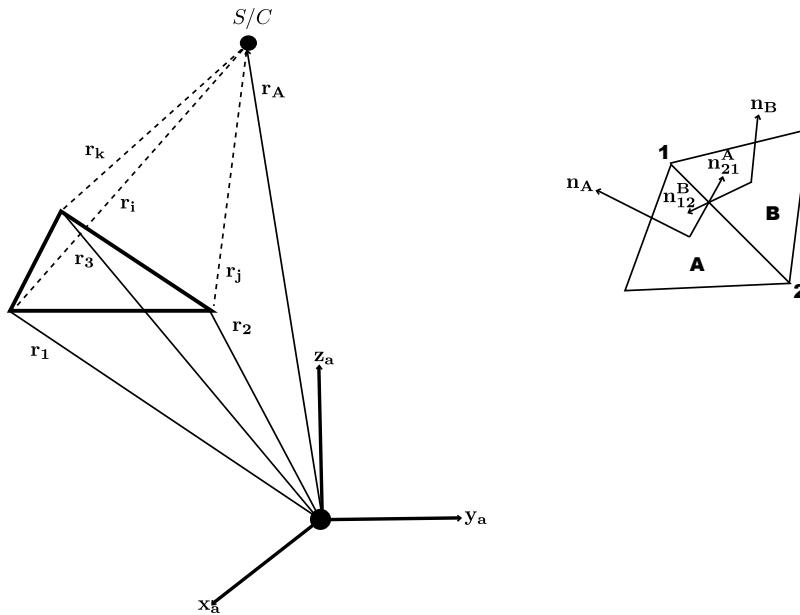


Figure 3.3: Elements of polyhedron gravity model explained.

The gravitational attraction is then given as the gradient of the potential in Eq. (3.3) and is equal to:

$$\mathbf{g}_A = \nabla U = -G\rho \sum_{e \in edges} \mathbf{E}_e \mathbf{r}_e L_e + G\rho \sum_{f \in faces} \mathbf{F}_f \mathbf{r}_f \omega_f \quad (3.8)$$

Furthermore, the gradient of the attraction, or the Laplacian of the potential is as follows:

$$\nabla^2 U = -G\rho \sum_{f \in faces} \omega_f \quad (3.9)$$

The sum through all faces of dimensionless per-face factor ω_f gives a solid angle, which vanishes if the field point is outside the polyhedron, or equals 4π , if it is inside the volume. This is a very useful property of polyhedron models, since it gives a means to know, whether a spacecraft crashes into an asteroid. Moreover, this property is exploited for hardware simulations later on.

Please note that the computed gravitational acceleration value is in the A frame, thus it has to be transformed to the I frame for use in Eq. (3.2).

3.2.2. Disturbance Forces

Third-body perturbations for the orbits simulated in this paper are negligible, therefore only solar-radiation-pressure (SRP) force, \mathbf{F}_{SRP} , is simulated. It can be calculated as a sum over all (N) illuminated planes of the spacecraft, (Markley and Crassidis, 2013), as follows:

$$\mathbf{F}_{SRP} = -P_{\odot} \sum_{i=1}^N \cos \theta_i A_i ((1-\epsilon) \mathbf{e}_{\odot,i} + 2\epsilon \cos \theta_i \mathbf{n}_i) \quad (3.10)$$

where P_{\odot} is the solar radiation pressure (it is a function of the distance from the Sun), θ_i is an angle of incidence of the radiation, A_i is the area of the i^{th} plane, $\mathbf{e}_{\odot,i}$ and \mathbf{n}_i are a unit vector from a plane to the Sun and a normal to the plane respectively, and ϵ is a reflectivity coefficient. Please note that $\mathbf{e}_{\odot,i}$ and \mathbf{n}_i have to be expressed in the same reference frame; and if they are expressed in B frame, then the resulting force will also be in the body frame, so the SRP force has to be transformed accordingly.

3.3. Rotational Motion

3.3.1. Attitude Representation

There exists a variety of attitude description methods. Historically, descriptions of finite rotations started with Cardano in the middle of the 16th century; 200 years later Euler did a significant work, solving most of the problems of rotation at his time. A parallel work of Rodrigues and Hamilton filled the gaps left by Euler by introducing quaternions. The work of orientation parametrization continued until the second half of the 20th century, with modified-Rodrigues parameters being introduced, (Fraiture, 2008). As a result, each of the descriptions have their own advantages and disadvantages, which is discussed further on.

Direction Cosine Matrix Consider two reference frames, I and B , with a right hand set of three orthogonal unit vectors, $(\mathbf{i}_1 \ \mathbf{i}_2 \ \mathbf{i}_3)^T$ and $(\mathbf{b}_1 \ \mathbf{b}_2 \ \mathbf{b}_3)^T$ respectively, Figure 3.4.

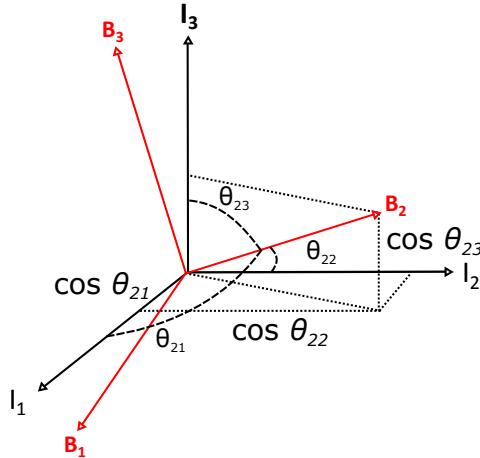


Figure 3.4: Direction cosines visualized.

The basis vectors of frame B can be expressed in frame I as follows:

$$\mathbf{b}_1 = C_{11}\mathbf{i}_1 + C_{12}\mathbf{i}_2 + C_{13}\mathbf{i}_3 \quad (3.11)$$

$$\mathbf{b}_2 = C_{21}\mathbf{i}_1 + C_{22}\mathbf{i}_2 + C_{23}\mathbf{i}_3 \quad (3.12)$$

$$\mathbf{b}_3 = C_{31}\mathbf{i}_1 + C_{32}\mathbf{i}_2 + C_{33}\mathbf{i}_3 \quad (3.13)$$

The constants C_{ij} refer to the projections of vectors \mathbf{b}_1 , \mathbf{b}_2 , \mathbf{b}_3 to the frame I vectors \mathbf{i}_1 , \mathbf{i}_2 , \mathbf{i}_3 . In Figure 3.1, it is shown that the constants C_{21} , C_{22} , C_{23} are equal to the cosines of the angles between corresponding axes. These cosines are called direction cosines. Then the whole frame B can be expressed in unit vectors of the frame I .

$$\begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{pmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{pmatrix} \mathbf{i}_1 \\ \mathbf{i}_2 \\ \mathbf{i}_3 \end{pmatrix} = \begin{bmatrix} \cos\theta_{11} & \cos\theta_{12} & \cos\theta_{13} \\ \cos\theta_{21} & \cos\theta_{22} & \cos\theta_{23} \\ \cos\theta_{31} & \cos\theta_{32} & \cos\theta_{33} \end{bmatrix} \begin{pmatrix} \mathbf{i}_1 \\ \mathbf{i}_2 \\ \mathbf{i}_3 \end{pmatrix} = \mathbf{C}_{B/I} \begin{pmatrix} \mathbf{i}_1 \\ \mathbf{i}_2 \\ \mathbf{i}_3 \end{pmatrix} \quad (3.14)$$

The corresponding direction cosines matrix $\mathbf{C}_{B/I}$ represents the attitude of frame B with respect to frame I and is called a Direction Cosine Matrix (DCM). It maps vectors in the inertial frame to the body frame, thus it can be used to transform vectors from I to B frame. For example, having position vector, \mathbf{R}_I , in the inertial frame I we can write it in the body frame as follows:

$$\mathbf{R}_B = \mathbf{C}_{B/I} \mathbf{R}_I \quad (3.15)$$

The DCM is an orthonormal matrix with the following properties:

$$\det \mathbf{C}_{B/I} = 1 \quad (3.16)$$

$$\mathbf{C}_{B/I}^{-1} = \mathbf{C}_{I/B} = \mathbf{C}_{B/I}^T \quad (3.17)$$

$$\mathbf{C}_{B/I} \mathbf{C}_{B/I}^T = \mathbf{I}_3 \quad (3.18)$$

The DCM serves well in analytical studies and the conversions of vectors from one frame to another; however, it has nine parameters, when only three are needed for a complete attitude/orientation definition, and thus it has six constraints.

Euler angles The Euler-angle representation can be defined as a rotation from the initial to the final frame as a series of successive rotations of the original and intermediate frames. Consider a rotation around the X-axis by an angle θ , Figure 3.5.

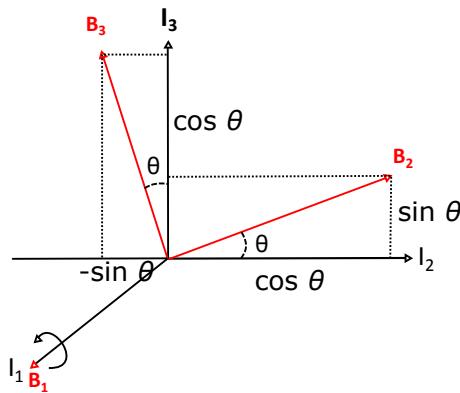


Figure 3.5: Rotation around X axis visualized.

Then the rotation can be written as a rotation matrix \mathbf{R}_X :

$$\mathbf{R}_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \quad (3.19)$$

Similarly, the rotations around Y- and Z-axis are given as:

$$\mathbf{R}_Y = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad (3.20)$$

$$\mathbf{R}_Z = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.21)$$

The resulting DCM, \mathbf{C} , of a rotation sequence XYZ (first rotation around X, then Y, and finally Z) is given by the multiplication of matrices $\mathbf{R}_z\mathbf{R}_y\mathbf{R}_x$. The multiplication sequence is important, since matrix multiplication is not commutative. As a result, we have 12 different combinations (XYZ, XZY, YXZ, YZX, ZXY, ZYX and XYX, XZX, YXY, YZY, ZXZ, ZYZ). The first group of six is often called Tait-Bryan angles and the second one is called proper Euler angles.

Euler angles have the advantage of requiring only three numbers for attitude representation, they are easy to understand and visualize. However, these angles suffer from singularities and they involve many trigonometric functions, which make the computation more demanding.

Unit Quaternions Quaternions are four-dimensional hyper-complex numbers, first introduced by Sir William Rowan Hamilton in the 19th century (Altman, 1986). He wrote them in the form given below:

$$\mathbf{q} = q_1 i + q_2 j + q_3 k + q_4 \quad (3.22)$$

where \mathbf{q} is a quaternion, i , j and k are imaginary numbers that have the following properties:

$$ij = k \quad jk = i \quad ki = j \quad (3.23)$$

and

$$i^2 = j^2 = k^2 = ijk = -1 \quad (3.24)$$

In this report the quaternion is treated as a four dimensional vector with the first three components being a vectorial part and the fourth one being a scalar.

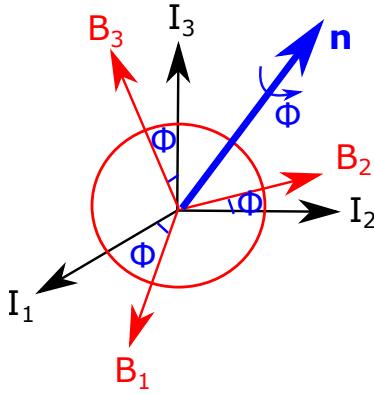
$$\mathbf{q} = \begin{pmatrix} \mathbf{q}_{1:3} \\ q_4 \end{pmatrix} \quad (3.25)$$

A quaternion is closely related to Euler's theorem, stating that any transformation of a reference frame can be accomplished by one rotation around a so-called Euler axis of rotation, Figure 3.6. Then a quaternion is written in the form:

$$\mathbf{q} = \begin{pmatrix} \sin \frac{\Phi}{2} \mathbf{n} \\ \cos \frac{\Phi}{2} \end{pmatrix} \quad (3.26)$$

where \mathbf{n} is the axis of rotation and Φ is the rotation angle. Since the axis of rotation is defined by a unit vector, a quaternion of rotation must have a norm of 1, $\|\mathbf{q}\| = 1$. Furthermore, two quaternion products are defined as:

$$[\mathbf{q} \otimes] = \begin{bmatrix} q_4 \mathbf{I}_3 - [\mathbf{q}_{1:3} \times] & \mathbf{q}_{1:3} \\ -\mathbf{q}_{1:3}^T & q_4 \end{bmatrix}, \quad [\mathbf{q} \odot] = \begin{bmatrix} q_4 \mathbf{I}_3 + [\mathbf{q}_{1:3} \times] & \mathbf{q}_{1:3} \\ -\mathbf{q}_{1:3}^T & q_4 \end{bmatrix} \quad (3.27)$$

Figure 3.6: Rotation around arbitrary axis, \mathbf{n} , visualized

where $\mathbf{q}_{1:3}$ is the vectorial part of a quaternion (first three components), q_4 is the scalar part (fourth component) of a quaternion, and $[\mathbf{q}_{1:3} \times]$ is a cross product matrix, which for a general vector \mathbf{a} is defined as follows:

$$[\mathbf{a} \times] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (3.28)$$

Furthermore, the following holds for the two products:

$$\mathbf{q}_2 \otimes \mathbf{q}_1 = \mathbf{q}_1 \odot \mathbf{q}_2 \quad (3.29)$$

The second product is historically the original one, although the first one is more often used, since it represents the composition of rotations in a similar way as it is done with a direction cosine matrix (DCM). Suppose we have two quaternions $\mathbf{q}_{B/I}$ and $\mathbf{q}_{I/A}$, and we want to know the resulting quaternion. If we use the first product, then $\mathbf{q}_{B/I} \otimes \mathbf{q}_{I/A} = \mathbf{q}_{B/A}$ is an intuitive result, since the resulting quaternion's indexes match the ones of the multiplicands. While if we use the second product, the result $\mathbf{q}_{I/A} \odot \mathbf{q}_{B/I} = \mathbf{q}_{B/A}$ is harder to understand, since the indexes are out of order. Finally, a DCM can be expressed in quaternions as:

$$\begin{aligned} \mathbf{C}(\mathbf{q}) &= \left(q_4^2 - \|\mathbf{q}_{1:3}\|^2 \right) \mathbf{I}_3 + 2\mathbf{q}_{1:3}\mathbf{q}_{1:3}^T - 2q_4 [\mathbf{q}_{1:3} \times] = \\ &= \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1 q_2 + q_3 q_4) & 2(q_1 q_3 - q_2 q_4) \\ 2(q_1 q_2 - q_3 q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2 q_3 + q_1 q_4) \\ 2(q_1 q_3 + q_2 q_4) & 2(q_2 q_3 - q_1 q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \end{aligned} \quad (3.30)$$

Quaternions have the advantage of being non-singular and they require only four parameters to describe the attitude. However, it is hard to comprehend and/or visualize them, and they still have one constraint ($\mathbf{q}\mathbf{q}^* = 1$), which has to be satisfied.

Rodrigues Parameters The Rodrigues parameter, also called as the Gibbs vector, is closely related to quaternions. The Rodrigues parameter, \mathbf{g} , is expressed in a quaternion as:

$$\mathbf{g} = \frac{\mathbf{q}_{1:3}}{q_4} \quad (3.31)$$

It is a three-dimensional vector, thus it has one dimension less than a quaternion itself. By inspecting Eq. (3.26), we would see that in terms of the Euler axis and the angle Φ it can be written as follows:

$$\mathbf{g} = \mathbf{n} \tan \Phi / 2 \quad (3.32)$$

Since a \tan function is undefined when its argument is $\pi/2$, the Gibbs vector is also undefined when $\Phi = \pi$. Therefore, a Rodrigues parameter is bounded within the range of $\Phi = -\pi$ and $\Phi = \pi$.

Modified Rodrigues Parameters The modified Rodrigues parameter is the most recently invented representation, which aims to solve the problem of the Rodrigues parameters of being undefined at $\Phi = \pi$. In terms of a quaternion, the modified Rodrigues parameter, \mathbf{p} is written as follows (Markley and Crassidis, 2013):

$$\mathbf{p} = \frac{\mathbf{q}_{1:3}}{1 + q_4} \quad (3.33)$$

or if it is expressed in terms of the Euler axis and the angle:

$$\mathbf{p} = \mathbf{n} \tan \Phi / 4 \quad (3.34)$$

From the equation above we see that the modified Rodrigues parameter is only undefined at $\Phi = 2\pi$, which means a full revolution, thus the undefined situation can be avoided.

3.3.2. Kinematic Equations

The kinematics equation of a DCM as given by Markley and Crassidis (2013) reads as follows:

$$\dot{\mathbf{C}}_{B/I} = -[\boldsymbol{\omega}_{B/I}^B \times] \mathbf{C}_{B/I} \quad (3.35)$$

where $\mathbf{C}_{B/I}$ is an attitude matrix of a body frame with respect to the inertial frame and $\boldsymbol{\omega}_{B/I}^B$ is the angular velocity of frame B with respect to I frame, expressed in frame B .

The Euler-angles kinematics equation depends on the sequence used. For example, the kinematics for a 3-2-1 sequence is given by (Wie, 2006):

$$\begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix} = \frac{1}{\cos \theta_2} \begin{bmatrix} \cos \theta_2 & \sin \theta_1 \sin \theta_2 & \cos \theta_1 \sin \theta_2 \\ 0 & \cos \theta_1 \cos \theta_2 & -\sin \theta_1 \cos \theta_2 \\ 0 & \sin \theta_1 & \cos \theta_1 \end{bmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \quad (3.36)$$

The equation becomes singular when $\theta_2 = \pi/2$. The quaternion kinematics can be written as follows::

$$\dot{\mathbf{q}}_{B/I} = \frac{1}{2} \boldsymbol{\omega}_{B/I}^B \otimes \mathbf{q}_{B/I} = \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}_{B/I}^B) \mathbf{q}_{B/I} \quad (3.37)$$

where $\boldsymbol{\Omega}$ is defined accordingly to Eq. (3.27), and is expressed below.

$$\boldsymbol{\Omega} = \begin{bmatrix} -[\boldsymbol{\omega} \times] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix} = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \quad (3.38)$$

The kinematics equations for Rodrigues parameters and modified Rodrigues parameters are not presented, because their expressions are bulkier, and thus they are discarded. Out of the three kinematic equations discussed, the quaternion one has the most appealing form. It is linear with respect to the quaternion itself, has no trigonometric functions and involves less multiplications than the DCM kinematics equation. As a result, quaternions will be used to represent the attitude in this work.

3.3.3. Attitude Dynamics

The attitude dynamics is represented by Euler's dynamic equation (Wie, 2006):

$$\dot{\mathbf{H}}_B + \boldsymbol{\omega}_{B/I}^B \times \mathbf{H}_B = \mathbf{T}_B \quad (3.39)$$

where \mathbf{H}_B is the angular momentum of the spacecraft and \mathbf{T}_B is the sum of all torques acting on the body (e.g. disturbance or control torques). For a rigid body with constant mass properties, the dynamics equation is rewritten in the form presented below.

$$\mathbf{I}\dot{\boldsymbol{\omega}}_{B/I}^B + \boldsymbol{\omega}_{B/I}^B \times \mathbf{I}\boldsymbol{\omega}_{B/I}^B = \mathbf{T}_B \quad (3.40)$$

The substitution $\mathbf{H}_B = \mathbf{I}\boldsymbol{\omega}_{B/I}^B$ was made, where \mathbf{I} is the inertia tensor of the spacecraft around its centre of mass, expressed in the B frame.

Disturbance torques Disturbance torques can either be internal or external. Internal torques are caused by fuel sloshing, mass imbalances in reaction wheels, etc., however, these are assumed to be absent. External torques are caused by sources, such as solar-radiation pressure, a gravity gradient, magnetic fields and an atmosphere. Since an asteroid does not have a significant magnetic field, nor an atmosphere, these can be discarded. Moreover, the simulated spacecraft is axi-symmetric, both in geometry and surface properties, so the SRP torque is neglected. Thus, only the gravity-gradient torque is simulated. Many sources give the gravity-gradient torque for a central gravity field as (Markley and Crassidis, 2013; Wie, 2006):

$$\mathbf{T}_{gg} = 3 \frac{\mu}{R^3} \mathbf{a}_3 \times \mathbf{I} \mathbf{a}_3 \quad (3.41)$$

where μ is the gravitational parameter, R is the distance to the attracting body and \mathbf{a}_3 is the third component of Local Vertical Local Horizontal reference frame. Equation (3.41) only holds for central gravity fields, which is not true for an asteroid, so a new way to simulate this torque is needed. This is discussed in Chapter 6.

3.4. Asteroid Dynamics

The orbit of an asteroid around the Sun is not simulated, since the simulation time is up to (only) three hours; therefore the movement around the Sun is negligible. Furthermore, the angular velocity of the asteroid is assumed to be constant in direction and magnitude, ignoring any nutation and/or precession:

$$\boldsymbol{\omega}_{A/I}^A = \text{const} \quad (3.42)$$

This leaves only the attitude of the asteroid to be simulated, which, if expressed in quaternions, is given similarly to the SC kinematics, Eq. (3.37):

$$\dot{\mathbf{q}}_{A/I} = \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}_{A/I}^A) \mathbf{q}_{A/I} \quad (3.43)$$

In other words, if one knows the initial attitude of the asteroid and its angular velocity, one can also know it at any time in the future.

3.5. Relative States

Until now, the dynamics was expressed with respect to an inertial reference frame, however, the navigation filter will deal with relative states, so these have to be expressed. The relative spacecraft states are position, \mathbf{R}_A , and velocity, \mathbf{V}_A , expressed in frame A , attitude, $\mathbf{q}_{B/A}$, and angular velocity $\boldsymbol{\omega}_{B/A}^B$. Let us start with the position and velocity. Relative position is simply obtained by frame transformation:

$$\mathbf{R}_A = \mathbf{C}_{A/I} \mathbf{R}_I \quad (3.44)$$

where $\mathbf{C}_{A/I}$ is defined by Eq. (3.30) with $\mathbf{q}_{A/I}$. Differentiating Eq. (3.44), gives the relative velocity, with the help of Eq. (3.35):

$$\mathbf{V}_A = \mathbf{C}_{A/I} \mathbf{V}_I - \boldsymbol{\omega}_{A/I}^A \times \mathbf{R}_A \quad (3.45)$$

The relative attitude, expressed in quaternions is:

$$\mathbf{q}_{B/A} = \mathbf{q}_{B/I} \otimes \mathbf{q}_{A/I}^{-1} \quad (3.46)$$

and the angular rate:

$$\boldsymbol{\omega}_{B/A}^B = \boldsymbol{\omega}_{B/I}^B - \mathbf{C}_{B/A} \boldsymbol{\omega}_{A/I}^A \quad (3.47)$$

where $\mathbf{C}_{B/A}$ is obtained by substituting the quaternion defined by Eq. (3.46) into Eq. (3.30).

4

Navigation

This chapter presents the main concepts of a spacecraft navigation system. This comprises of navigation sensors and, due to the noisy nature of them, an estimator/filter to fuse and filter the measurements and provide accurate results. Section 4.1 gives an overview of the hardware that will be used. This includes working principles and equations of the measurements. Section 4.2 continues the chapter with the most popular estimation technique - the extended Kalman filter. As it was discussed in the introduction, despite the linearisation errors, difficult tuning and instabilities, it is still widely used in space missions, because it is computationally efficient, it has the same structure as the linear Kalman filter, and there is a vast knowledge existing on how to develop and tune this filter. As a result, the extended Kalman filter is implemented in this thesis as well. The concept and governing equations of the filter are presented in Section 4.2.

4.1. Hardware

The simulated spacecraft is equipped with a star tracker and a gyroscope for inertial state measurements, a navigation camera and a laser-ranger for relative state measurements.

4.1.1. Gyroscope

A gyroscope is a part of Inertial Measurement Unit (IMU) that measures the inertial angular velocity directly. By working principles they can be separated into three different types: rotating mass, optical and vibrating structure gyroscopes.

The rotating-mass type is the oldest type. It consists of a rotating disc on a gimbal, which, due to conservation of momentum, tends to remain inertially fixed. When a spacecraft rotates, the rotating mass remains inertially fixed and, by measuring the change in the gimbal angle or the torque produced to keep the gimbal angle at zero, one can get the angular rate.

Optical gyroscopes can be laser-ring or optical-fibre gyroscopes. Both are based on the interference of a split beam of light travelling in opposite directions of an optical loop. This interference is also known as Sagnac effect. If the base of the loop is rotating, then the beam travelling along the direction of rotation will travel longer distance than the one travelling in the opposite direction. From the interference of these beams, one can extract the difference in time that it took for them to travel. This time is then directly proportional to the angular velocity.

The last and the most recent type is the vibrating structure gyroscope, which has a vibrating part that tends to keep the plane of oscillations constant. When the support of this structure rotates, Coriolis force is applied to it and by measuring the force, one can determine the angular rate. The system can be integrated on a Micro-Electro-Mechanical System (MEMS) and thus be very small. MEMS gyroscopes are widely spread in consumer electronics, e.g., smartphones.

For this thesis, we are not concerned about the type of gyroscope, because mathematically they will output the same result, which, as all sensors, is susceptible to noise. Moreover, a gyroscope experiences a drift, which is an accumulated noise over time. Other errors, such as misalignment and scale errors are discarded for now, assuming that the knowledge of these parameters is absolute. In continuous time, the measured rate, $\tilde{\boldsymbol{\omega}}_{B/I}^B$ (the tilde symbol denotes measured values), of a spacecraft can be written as:

$$\tilde{\boldsymbol{\omega}}_{B/I}^B(t) = \boldsymbol{\omega}_{B/I}^B(t) + \boldsymbol{\mu}(t) + \boldsymbol{\eta}_v(t) \quad (4.1)$$

$$\dot{\boldsymbol{\mu}}(t) = \boldsymbol{\eta}_u(t) \quad (4.2)$$

where $\boldsymbol{\mu}$ is the drift, and $\boldsymbol{\eta}_v$, $\boldsymbol{\eta}_u$ are zero-mean, white noise vectors, with known variance. The discrete time implementation will be provided in Chapter 5.

4.1.2. Star Tracker

A star tracker measures the position of the stars in its field-of-view and compares this to the known positions on the celestial sphere in the star map, stored on-board. Basically, it would output a single unit vector for each star, and when there are more than one star, one could estimate the attitude. Modern star trackers output a quaternion $\tilde{\boldsymbol{q}}_{B/I}$ directly, having a simple estimator inside the tracker itself. This will be the way the star tracker is simulated. The real quaternion $\boldsymbol{q}_{B/I}$ will be affected by a noise quaternion $\delta\boldsymbol{q}$.

$$\tilde{\boldsymbol{q}}_{B/I} = \delta\boldsymbol{q}^{-1} \otimes \boldsymbol{q}_{B/I} \quad (4.3)$$

where $\delta\boldsymbol{q}$ is approximated by a small-angle rotation vector (Markley and Crassidis, 2013):

$$\delta\boldsymbol{q} = \begin{pmatrix} \phi/2 \\ \theta/2 \\ \psi/2 \\ 1 \end{pmatrix} \quad (4.4)$$

The noise angles ϕ, θ, ψ are simulated as zero-mean, white noise vectors, with known variance. The noise, corresponding to the boresight axis usually, has a higher variance. Note that the noise quaternion is no longer a unit quaternion, thus the measured quaternion in Eq. (4.3) has to be normalized. Also, no scale-factor or misalignment errors are simulated, since perfect knowledge of them is assumed. The inclusion of this kind of instrument errors remains to be done as future work.

4.1.3. Navigation Camera

A Navigation Camera (NAVCAM) is one of the most common type of sensors in interplanetary missions, because it is possible to achieve precise results with relatively simple hardware. There are two modes, in which the camera can operate: relative (optical flow) and absolute. The first one is used when a relative motion needs to be estimated between a spacecraft and a planet/asteroid. In this mode, the camera tracks changes in the picture and estimates how much it moved since the last picture. The working principle resembles the one of an optical mouse, which takes tens or hundreds of images per second and outputs how much it moved. This method can be used, for example, in a lander operation, where a horizontal movement relative to the surface has to be determined. The latter mode requires more sophisticated algorithms for image processing. Landmark (e.g., craters, ridges, boulders) coordinates have to be extracted from the images and then compared to the a-priori generated landmark map. Another mission, or mission phase, is required to generate this map, although there are techniques, which could do simultaneous localization (navigation) and mapping (Conway et al, 2014), which will not be discussed in this thesis. The image processing will also be left as a future work, because the topic is vast by itself, so we will assume a navigation camera, which outputs the pixel coordinates of landmarks in its field-of-view.

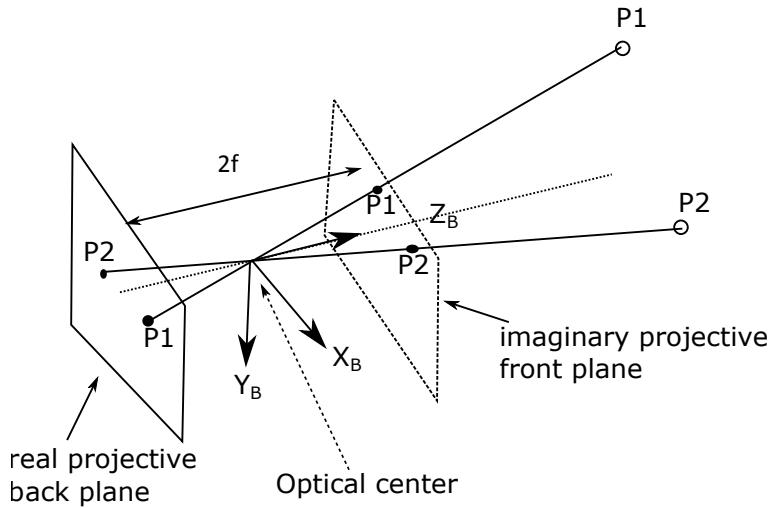


Figure 4.1: Pinhole camera model.

A simple, yet effective, pinhole camera model will be used, as shown in Figure 4.1. The camera reference frame is assumed to be aligned with the body frame and is situated at the optical centre of the camera. The real projective plane is positioned at $-f$ distance along the Z -axis, where f is the focal length of the optics used in the camera.

Suppose there is a point P_1 with coordinates $(X_B, Y_B, Z_B)^T$, then the coordinates of this point on the projective plane are:

$$X_p = -f \frac{X_B}{Z_B} \quad (4.5)$$

$$Y_p = -f \frac{Y_B}{Z_B} \quad (4.6)$$

To avoid having the minuses, a concept of an imaginary front projective plane is used, which then results in projection coordinates having the same signs as the real coordinates:

$$X_p = f \frac{X_B}{Z_B} \quad (4.7)$$

$$Y_p = f \frac{Y_B}{Z_B} \quad (4.8)$$

Furthermore, the projective plane is a discretized matrix, usually CCD or CMOS, which has a limited number of pixels. If a pixel size is p , then the pixel coordinates u, v of the point P_1 will be:

$$u = \frac{f}{p} \frac{X_B}{Z_B} \quad (4.9)$$

$$v = \frac{f}{p} \frac{Y_B}{Z_B} \quad (4.10)$$

where u and v are expressed in non-dimensional coordinates of pixels. Furthermore, since the actual landmark-recognition and centre-finding algorithm is not simulated, errors/noise are added to the coordinates:

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} \delta u \\ \delta v \end{pmatrix} \quad (4.11)$$

where $\delta u, \delta v$ are zero-mean, white noises with known variance, expressed in (sub-)pixels. Sometimes it is more convenient to use unit vectors to the landmarks and write the measurements as follows:

$$\tilde{\mathbf{b}} = \frac{1}{\sqrt{(p\tilde{u})^2 + (p\tilde{v})^2 + f^2}} \begin{pmatrix} p\tilde{u} \\ p\tilde{v} \\ f \end{pmatrix} \quad (4.12)$$

To implement this model, we first need to generate the landmark map on the surface of an asteroid, then determine which landmarks the camera sees, transform their coordinates from frame A to B and then project them on the plane with Eqs. (4.9) and (4.10). The actual implementation will be discussed in Chapter 6.

4.1.4. Laser Ranger

A Laser Ranger (LR) is a device that measures the time that takes the laser light to travel forward, reflect from a surface and travel back. Since the speed of light is a constant, one can easily calculate the distance from the reflection point. The physics behind this process is much more complicated, since the signal can be attenuated or dispersed by the surface properties or by the environment it propagates through, which would result in a weaker return signal and thus errors in distance. The errors are also a function of the wavelength used, power of the laser beam, laser pulse width and other parameters; however, these are out of the scope of the thesis and will be left as a future work.

A simplified model of the laser-ranger is used, which outputs the distance directly. Measuring random distances to the surface of an asteroid is not useful, since an asteroid can have a significantly irregular shape, so the distance of a random point would not give useful information for the navigation. Points on the surface with known coordinates are needed. As the landmark map is assumed to be present, the same points will be used to measure the distance to them. However, this requires the ability to point the laser in the direction of a landmark. Since, the spacecraft cannot be torqued (no control assumption), it is assumed that the laser has a gimbal/mirror system that allows it to point the beam in desired direction.

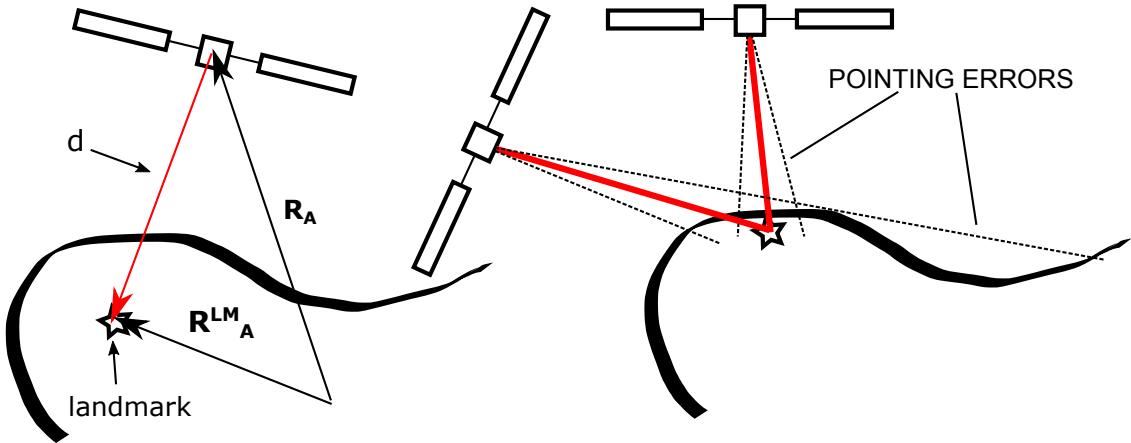


Figure 4.2: Laser ranger measurements and errors.

The distance, d , from the spacecraft to the landmark, Figure 4.2, can be expressed as:

$$d = \| \mathbf{R}_A^{LM} - \mathbf{R}_A \| \quad (4.13)$$

where \mathbf{R}_A^{LM} is the coordinate of a landmark. However, the pointing of the laser, as any real life system, is not perfect and small errors are introduced, which can change the measured distance signifi-

cantly. Figure 4.2 (right) shows that when the angle between the laser beam and the local normal increases, the same angular deviation causes larger overshoots. This effect is increased even more for irregular shapes, which is mostly the case for asteroids. This is known as foreshortening effect and has a critical impact on mapping accuracy in missions around regular bodies. It will be accounted for in this work, though not for mapping performances but for laser ranging error modelling. As a result, Eq. (4.13) is not valid any-more for getting the measured distance, since the pointing-angle error changes the distance in an unpredictable way. There is no analytical function to solve this problem for an irregular body, thus a numerical solution must be found. This will be done in Chapter 6.

4.2. Extended Kalman Filter

Imagine a fictional spacecraft, whose pose dynamics is a linear function of the pose itself. It also has a 'positionmeter' and 'attitudemeter', thus measure the pose directly with some noise. Then one would need a state estimator, which would fuse the measurements and estimate the true state. In this case the system dynamics and the measurement equations are linear with respect to the state itself, so a linear quadratic estimator, or a Kalman filter would be used. However, the real world tends to be governed by non-linear processes, and neither a sensor such as 'positionmeter' nor as an 'attitudemeter' exists. To deal with this problem, there are non-linear state estimators, such as Unscented Kalman filter, developed by Julier and Uhlman (1997). Nevertheless, the most widely used estimator for non-linear systems is the Extended Kalman Filter (EKF): the filter is applied to non-linear systems simply by linearising them around the current best estimate and it proved to give good results. Since the objective of the thesis is to compare two different pose representations for navigation purposes, we would rather choose a filter, for which one can find many literature sources. This saves time of understanding and deriving new equations ourselves. Therefore, the EKF is used in this thesis.

4.2.1. Linearisation

Consider having a non-linear system:

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}) + \mathbf{G}\mathbf{w} \quad (4.14)$$

where \mathbf{X} is the state, \mathbf{w} is a zero-mean, Gaussian process noise, with a known covariance \mathbf{Q} , and \mathbf{G} is a noise mapping matrix. Furthermore, sensors are usually read with a digital processor, therefore they provide discrete-time measurements. Therefore, a non-linear measurement equation in a discrete time is as follows:

$$\mathbf{z}_{k+1} = \mathbf{h}(\mathbf{X}) + \mathbf{v} \quad (4.15)$$

where \mathbf{v} is a zero-mean, Gaussian measurement noise, with a known covariance \mathbf{R} . Both functions $\mathbf{f}(\mathbf{X})$ and $\mathbf{h}(\mathbf{X})$ are non-linear and continuously differentiable. The goal is to obtain linear perturbation equations and then apply the simple Kalman filter. For an extensive Kalman filter explanation and development the reader is referred to the books of Zarchan and Musoff (2009) and Markley and Crassidis (2013). Let \mathbf{X}^* and \mathbf{z}_k^* be a nominal behaviour of the system, then the perturbations can be written as follows:

$$\begin{cases} \delta\mathbf{X} = \mathbf{X} - \mathbf{X}^* \\ \delta\mathbf{z}_k = \mathbf{z}_k - \mathbf{z}_k^* \end{cases} \quad (4.16)$$

Now we can approximate the functions $\mathbf{f}()$ and $\mathbf{h}()$ with the first order Taylor series expansion in $\delta\mathbf{X}$ around the point \mathbf{X}^* :

$$\mathbf{f}(\mathbf{X}) = \mathbf{f}(\mathbf{X}^*) + \mathbf{F}(\mathbf{X}^*)\delta\mathbf{X} \quad (4.17)$$

Then, the linear perturbation can be expressed as:

$$\delta\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X}^*)\delta\mathbf{X} + \mathbf{G}\mathbf{w} \quad (4.18)$$

where $\mathbf{F}(\mathbf{X}^*)$ is a Jacobian of the function $\mathbf{f}()$, evaluated at \mathbf{X}^* :

$$\mathbf{F} = \left[\begin{array}{ccc} \frac{\partial f_1}{\partial X_1} & \frac{\partial f_1}{\partial X_2} & \dots \\ \frac{\partial f_2}{\partial X_1} & \frac{\partial f_2}{\partial X_2} & \dots \\ \dots & \dots & \dots \end{array} \right]_{X=X^*} \quad (4.19)$$

Similar expressions can be obtained for the measurement equation:

$$\mathbf{h}(\mathbf{X}_k) = \mathbf{h}(\mathbf{X}_k^*) + \mathbf{H}(\mathbf{X}_k^*) \delta \mathbf{X}_k \quad (4.20)$$

$$\delta \mathbf{z}_{k+1} = \mathbf{H}(\mathbf{X}^*) \delta \mathbf{X}_k + \mathbf{v} \quad (4.21)$$

where the measurement matrix $\mathbf{H}(\mathbf{X})$ is defined similarly to the \mathbf{F} matrix and given as follows:

$$\mathbf{H} = \left[\begin{array}{ccc} \frac{\partial h_1}{\partial X_1} & \frac{\partial h_1}{\partial X_2} & \dots \\ \frac{\partial h_2}{\partial X_1} & \frac{\partial h_2}{\partial X_2} & \dots \\ \dots & \dots & \dots \end{array} \right]_{X=X^*} \quad (4.22)$$

At this moment, we already have a linear state space model for perturbations, consisting of Eqs. (4.18) and (4.21). At this point the ordinary Kalman filter algorithm can be applied to the perturbation, which is then used to correct the nominal state \mathbf{X}^* :

$$\hat{\mathbf{X}} = \mathbf{X}^* + \delta \hat{\mathbf{X}} \quad (4.23)$$

The nominal state \mathbf{X}^* is chosen as the best available estimate of \mathbf{X} at the current time, thus the system will be linearised around the already estimated trajectory.

4.2.2. Time Propagation Stage

The linear perturbation $\delta \mathbf{X}$ is propagated at time step $k+1$, given the estimate $\delta \mathbf{X}_{k/k}$. Note that the subscript a/b should be read as the time step a , given the measurements until time step b . In general, the propagation of the perturbation would look like:

$$\delta \mathbf{X}_{k+1/k} = \Phi \delta \mathbf{X}_{k/k} = \Phi (\hat{\mathbf{X}}_{k/k} - \mathbf{X}^*) \quad (4.24)$$

where $\Phi = \exp(\mathbf{F}(\hat{\mathbf{X}}_{k/k}) \Delta t) \approx \mathbf{I} + \mathbf{F}(\hat{\mathbf{X}}_{k/k}) \Delta t$ is the discrete version of the state transition matrix (the Jacobian \mathbf{F}). By definition it was decided that the nominal trajectory \mathbf{X}^* would be the best current estimate, so $\mathbf{X}^* = \hat{\mathbf{X}}_{k/k}$, thus the perturbation is propagated as follows:

$$\delta \mathbf{X}_{k+1/k} = \Phi (\hat{\mathbf{X}}_{k/k} - \hat{\mathbf{X}}_{k/k}) = 0 \quad (4.25)$$

which basically means that the linear perturbation is not propagated in time and just set to zero. This is called a full reset. Then the nominal state is propagated (the dynamics equation \mathbf{f} is integrated). In this thesis an Euler step method is used:

$$\hat{\mathbf{X}}_{k+1/k} = \hat{\mathbf{X}}_{k/k} + \mathbf{f}(\hat{\mathbf{X}}_{k/k}) \Delta t \quad (4.26)$$

And finally the error covariance matrix, \mathbf{P} , is propagated in time:

$$\mathbf{P}_{k+1/k} = \Phi \mathbf{P}_{k/k} \Phi^T + \mathbf{G} \mathbf{Q} \mathbf{G}^T \Delta t \quad (4.27)$$

4.2.3. Measurement Update

When a measurement is available, the measurement update stage takes place. First of all, the perturbation for the measurement, as in Eq. (4.16), reads as follows:

$$\delta z_{k+1} = z_{k+1} - h(\hat{X}_{k+1/k}) \quad (4.28)$$

Then, using the best current estimate, $\hat{X}_{k+1/k}$, the measurement gradient matrix H is computed and the Kalman gain evaluated:

$$K_{k+1} = P_{k+1/k} H^T (H P_{k+1/k} H^T + R)^{-1} \quad (4.29)$$

Afterwards, the system can then be updated:

$$\delta \hat{X}_{k+1/k+1} = \delta \hat{X}_{k/k+1} + K_{k+1} (\delta z_{k+1} - H \delta \hat{X}_{k/k+1}) \quad (4.30)$$

Taking to account the full reset ($\delta \hat{X}_{k/k+1} = 0$), the update becomes:

$$\delta \hat{X}_{k+1/k+1} = K_{k+1} \delta z_{k+1} = K_{k+1} (z_{k+1} - h(\hat{X}_{k+1/k})) \quad (4.31)$$

And the state itself can be updated:

$$\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + \delta \hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_{k+1} (z_{k+1} - h(\hat{X}_{k+1/k})) \quad (4.32)$$

Finally the covariance matrix is updated with:

$$P_{k+1/k+1} = (I - KH) P_{k+1/k} (I - KH)^T + KRK^T \quad (4.33)$$

4.2.4. EKF for Quaternions

When a quaternion is a part of the state vector, its properties have to be taken into account, when developing a filter. A quaternion represents the attitude only when its norm is one. Multiplying or integrating the kinematic equation (3.37) does not change its norm. If it were initialized as a unit quaternion, it will stay unit, neglecting small numerical errors. However, for the EKF we need to define a quaternion error. By a direct analogy of Eq. (4.16), we can write:

$$\delta q = q - \hat{q} \quad (4.34)$$

In this case the so-called additive quaternion error is defined. In general, the additive error δq is not unit any more, thus it is not a quaternion of rotation. The normalization process has to be done at each step to keep the estimated quaternion with a norm of one. Another approach is to use a multiplicative quaternion error, which is written as follows:

$$\delta q = q \otimes \hat{q}^{-1} \quad (4.35)$$

The error quaternion defined in the equation above preserves the norm and thus stays a quaternion of rotation. Suppose the inertial attitude of a spacecraft is $q_{B/I}$ and its estimate is $\hat{q}_{\hat{B}/I}$; then, the error quaternion is:

$$\delta q = q_{B/I} \otimes \hat{q}_{\hat{B}/I}^{-1} = q_{B/I} \otimes \hat{q}_{I/\hat{B}} = q_{B/\hat{B}} \quad (4.36)$$

Therefore, the multiplicative quaternion error has a meaning of a small rotation from the estimated frame \hat{B} to the real one B . Similarly, the measurement error (if the measurement is a quaternion itself, e.g. a star tracker output) in multiplicative error terms is as follows:

$$\delta z = \tilde{q} \otimes \hat{q}^{-1} \quad (4.37)$$

where \tilde{q} is the measured quaternion. Since the multiplicative error has a clear physical meaning, it is further used for developing the filter.

4.2.5. Full EKF Algorithm

This subsection summarizes the work-flow of the extended Kalman filter. The work-flow can be divided into three steps: development, initialization and operation.

Development

1. Write down the dynamic and measurement functions $f()$ and $h()$
2. Derive the matrices F and H , Eqs. (4.19) and (4.22)

Initialization

1. Initialize the state $\hat{X}_{0/0}$
2. Initialize/set the covariance matrices $P_{0/0}$, Q and R
3. Tune the matrices to achieve the desired performance

Operation

1. Propagate the state, Eq. (4.26)
2. Evaluate the matrix Φ with $\hat{X}_{k+1/k}$
3. Propagate the covariance matrix, Eq. (4.27)
4. If a measurement is available go to 5., if not go back to 1.
5. Calculate the measurement residual, Eq. (4.28) or Eq. (4.37)
6. Evaluate the measurement matrix H with $\hat{X}_{k+1/k}$
7. Compute the Kalman gain, Eq. (4.29)
8. Evaluate the state error, Eq. (4.31)
9. Update the state with $\hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} \oplus \delta \hat{X}_{k+1/k+1}$
10. Update the error covariance matrix, Eq. (4.33)

The symbol ' \oplus ' in the ninth step of operations denotes a generic composition, which can either be a simple sum or a quaternion product. The development of the filter will be presented in Chapter 7 and the initialization/operation in Chapter 8.

4.3. Conclusions

This chapter presented the main elements of a navigation system. These were the sensors that are going to be used in this work, and the most widely used estimation technique, the extended Kalman filter. Out of the four sensors, the navigation camera and the laser ranger appeared to be more difficult to simulate, therefore a more extensive description of implementations are provided in Chapter 6. The introduction to the EKF showed that a special treatment has to be taken into account when using quaternions in the EKF.

5

Dual Quaternions

Up to this point, the position and attitude representation were treated separately with, usually, Cartesian coordinates (a vector) representing position and a DCM (or a quaternion) for the attitude. The first attempt to represent the pose by one eight-dimensional vector was done by William Kingdom Clifford in his paper about bi-quaternions, or, as we know them now, dual quaternions, (Clifford, 1873). A dual quaternion is an extension of a regular quaternion to dual algebra, which is discussed in Section 5.1. Section 5.2 introduces a dual quaternion as an 8-tuple vector for the pose representation. Finally, Section 5.3 provides a means how a dual quaternion can be visualized by a screw axis.

5.1. Dual Numbers

To understand dual quaternions, it is first needed to understand dual numbers and dual algebra. Dual numbers (also called duplexes) are an extension to the real numbers. In the form they are written, dual numbers resemble complex numbers, (Fischer, 1999):

$$\check{d} = a + \epsilon b \quad (5.1)$$

where \check{d} is a dual number (the symbol ‘‘’ denotes a dual quantity), a and b are real numbers, and ϵ has the following properties:

$$\epsilon \neq 0, \quad \epsilon^2 = 0 \quad (5.2)$$

There is no trivial explanation of these properties of ϵ , thus it has to be taken as it is given, and accepted as truth. The first part in Eq. (5.1) is called the primary (or real) part of the dual number and the second one represents the dual component. Furthermore, the addition and multiplication of dual numbers are defined as:

$$\check{d}_1 + \check{d}_2 = (a_1 + a_2) + \epsilon(b_1 + b_2) \quad (5.3)$$

$$\check{d}_1 \cdot \check{d}_2 = (a_1 + \epsilon b_1)(a_2 + \epsilon b_2) = a_1 a_2 + \epsilon(a_1 b_2 + b_1 a_2) \quad (5.4)$$

Similarly to complex numbers, the division of two dual numbers can be written as:

$$\frac{\check{d}_1}{\check{d}_2} = \frac{a_1 + \epsilon b_1}{a_2 + \epsilon b_2} \cdot \frac{a_2 - \epsilon b_2}{a_2 - \epsilon b_2} = \frac{a_1 a_2 + \epsilon(b_1 a_2 - a_1 b_2)}{a_2^2} \quad (5.5)$$

Another useful property of dual numbers is their expansion by Taylor series. Using Eq. (5.2), it follows:

$$f(a + \epsilon b) = f(a) + \epsilon b \frac{df(a)}{da} \quad (5.6)$$

This equation allows to express trigonometric and other functions for dual numbers. For further details about dual numbers and dual algebra the reader is referred to the book of Fischer (1999).

5.2. Dual Quaternion

Similarly as a dual number is constructed, a Dual Quaternion (DQ) $\check{\mathbf{q}}$ can be written as, (Jia, 2013):

$$\check{\mathbf{q}} = \mathbf{q}_r + \epsilon \mathbf{q}_d \quad (5.7)$$

where \mathbf{q}_r is the real and \mathbf{q}_d is the dual part of the dual quaternion, and both of them are quaternions (not necessarily unit quaternions). Furthermore, a dual quaternion can be seen as a dual-hyper-complex vector and can be written in the following form:

$$\check{\mathbf{q}} = q_{r1}i + q_{r2}j + q_{r3}k + q_{r4} + \epsilon(q_{d1}i + q_{d2}j + q_{d3}k + q_{d4}) \quad (5.8)$$

where i , j and k are imaginary numbers defined in Eqs. (3.23) and (3.24). It is cumbersome to manipulate a dual quaternion algebraically, as in Eq. (5.8). Thus, in this thesis a dual quaternion is seen as an 8-tuple vector, defined as follows:

$$\check{\mathbf{q}} = \begin{pmatrix} \mathbf{q}_r \\ \mathbf{q}_d \end{pmatrix} = \begin{pmatrix} q_{r1} \\ q_{r2} \\ q_{r3} \\ q_{r4} \\ q_{d1} \\ q_{d2} \\ q_{d3} \\ q_{d4} \end{pmatrix} \quad (5.9)$$

The multiplication ($\check{\otimes}$) of two dual quaternions follows the rules of dual numbers and quaternions. It reads as follows:

$$\check{\mathbf{q}}_1 \check{\otimes} \check{\mathbf{q}}_2 = \mathbf{q}_{r1} \otimes \mathbf{q}_{r2} + \epsilon (\mathbf{q}_{r1} \otimes \mathbf{q}_{d2} + \mathbf{q}_{d1} \otimes \mathbf{q}_{r2}) \quad (5.10)$$

where $\check{\otimes}$ is the dual quaternion multiplication, which in matrix form is written as:

$$[\check{\mathbf{q}}_1 \check{\otimes}] \check{\mathbf{q}}_2 = \begin{bmatrix} [\mathbf{q}_{r1} \otimes] & \mathbf{0}_{4 \times 4} \\ [\mathbf{q}_{d1} \otimes] & [\mathbf{q}_{r1} \otimes] \end{bmatrix} \begin{pmatrix} \mathbf{q}_{r2} \\ \mathbf{q}_{d2} \end{pmatrix} \quad (5.11)$$

5.2.1. Conjugates

A dual quaternion can have three different conjugates (Jia, 2013):

$$\check{\mathbf{q}}^\diamond = \mathbf{q}_r - \epsilon \mathbf{q}_d \quad (5.12)$$

$$\check{\mathbf{q}}^* = \mathbf{q}_r^* + \epsilon \mathbf{q}_d^* \quad (5.13)$$

$$\check{\mathbf{q}}^\circ = \mathbf{q}_r^* - \epsilon \mathbf{q}_d^* \quad (5.14)$$

The first one has the dual part with the opposite sign than the real one. A multiplication of a DQ with its first conjugate yields:

$$\check{\mathbf{q}} \check{\otimes} \check{\mathbf{q}}^\diamond = (\mathbf{q}_r + \epsilon \mathbf{q}_d)(\mathbf{q}_r - \epsilon \mathbf{q}_d) = \mathbf{q}_r \otimes \mathbf{q}_r + \epsilon(\mathbf{q}_d \otimes \mathbf{q}_r - \mathbf{q}_r \otimes \mathbf{q}_d) \quad (5.15)$$

This is not very useful, since the quaternion product $\mathbf{q}_r \otimes \mathbf{q}_r$ is not generally scalar, and $\mathbf{q}_d \otimes \mathbf{q}_r \neq \mathbf{q}_r \otimes \mathbf{q}_d$. As a result, this conjugate is rarely used.

The second conjugate is a quaternion conjugate of the real and dual parts. A multiplication of DQ with this conjugate gives:

$$\check{\mathbf{q}} \check{\otimes} \check{\mathbf{q}}^* = (\mathbf{q}_r + \epsilon \mathbf{q}_d)(\mathbf{q}_r^* + \epsilon \mathbf{q}_d^*) = \mathbf{q}_r \otimes \mathbf{q}_r^* + \epsilon(\mathbf{q}_r \otimes \mathbf{q}_d^* + \mathbf{q}_d \otimes \mathbf{q}_r^*) \quad (5.16)$$

A quaternion multiplication with its conjugate ($\mathbf{q}_r \otimes \mathbf{q}_r^*$) gives a scalar, and the dual part, after some mathematical treatment, turns out to be also a scalar. Thus, the result of the second conjugate product is given as follows:

$$\check{\mathbf{q}} \check{\otimes} \check{\mathbf{q}}^* = \|\mathbf{q}_r\|^2 + 2\epsilon(q_{r1}q_{d1} + q_{r2}q_{d2} + q_{r3}q_{d3} + q_{r4}q_{d4}) \quad (5.17)$$

The result in a general case is a dual number ($a + \epsilon b$), but if the real part is orthogonal to the dual one, the dual part vanishes, and the product becomes a real number.

The third conjugate is the combination of the first and the second conjugates and is given in the equation below:

$$\check{\mathbf{q}} \check{\otimes} \check{\mathbf{q}}^\diamond = \mathbf{q}_r \otimes \mathbf{q}_r^* + \epsilon(\mathbf{q}_d \otimes \mathbf{q}_r^* - \mathbf{q}_r \otimes \mathbf{q}_d^*) \quad (5.18)$$

The real part of this product is a real scalar, but the dual part remains to be a quaternion.

5.2.2. Dual Quaternions for Pose Representation

Up to this point, we have discussed the general properties of dual numbers and dual quaternions, but we have not introduced a way to represent both, the position and the attitude (pose of a Cartesian coordinates frame with respect to another frame) with a dual quaternion. To begin with, the real part of a dual quaternion will be a unit quaternion, thus representing the attitude. Furthermore, the position has to be incorporated as well. Jia (2013) gives the dual part of the dual quaternion as a multiplication of the real part with a position vector from the origin frame to the frame the pose is described for (defined as $(\mathbf{R} \ 0)^T$):

$$\mathbf{q}_d = \frac{1}{2}\mathbf{q}_r \otimes \mathbf{R} \quad (5.19)$$

Then we calculate the dual quaternion product with its second conjugate:

$$\begin{aligned} \check{\mathbf{q}} \check{\otimes} \check{\mathbf{q}}^* &= \left(\mathbf{q}_r + \frac{1}{2}\epsilon\mathbf{q}_r \otimes \mathbf{R} \right) \check{\otimes} \left(\mathbf{q}_r + \frac{1}{2}\epsilon\mathbf{q}_r \otimes \mathbf{R} \right)^* = \left(\mathbf{q}_r + \frac{1}{2}\epsilon\mathbf{q}_r \otimes \mathbf{R} \right) \check{\otimes} \left(\mathbf{q}_r^* + \frac{1}{2}\epsilon\mathbf{R}^* \otimes \mathbf{q}_r^* \right) = \\ &= \mathbf{q}_r \otimes \mathbf{q}_r^* + \frac{1}{2}\epsilon(\mathbf{q}_r \otimes \mathbf{R} \otimes \mathbf{q}_r^* + \mathbf{q}_r \otimes \mathbf{R}^* \otimes \mathbf{q}_r^*) \end{aligned} \quad (5.20)$$

The quaternion conjugate of a pure vector is simply $\mathbf{R}^* = -\mathbf{R}$, so Eq. (5.20) reduces to a scalar 1, which, means that the dual quaternion has the norm of one. Thus, in a similar way the quaternion of rotation must have a norm of one, a dual quaternion to represent a pose must also have a norm (real number) of one:

$$\check{\mathbf{q}} \check{\otimes} \check{\mathbf{q}}^* = 1 \quad (5.21)$$

The equation above requires two constraints to be met. The real part of a dual quaternion has to have a norm of one, which is equivalent to the quaternion being a quaternion of rotation, and it is written as follows:

$$\|\mathbf{q}_r\|^2 = 1 \quad (5.22)$$

Furthermore, a second constraint is introduced, which is derived from the dual part of the dual quaternion product in Eq. (5.17).

$$(q_{r1}q_{d1} + q_{r2}q_{d2} + q_{r3}q_{d3} + q_{r4}q_{d4}) = \mathbf{q}_r \cdot \mathbf{q}_d = 0 \quad (5.23)$$

which requires that the dot product between the real and the dual parts is equal to zero. In other words, the two quaternions have to be orthogonal.

A dual-quaternion is an eight-dimensional vector, so these constraints remove two parameters, and thus only six are left, which is the number required to represent the pose.

Now, it is essential to define which reference frame the position vector is expressed in. Suppose having two reference frames A and B . The attitude of the frame B with respect to the frame A is $\mathbf{q}_{B/A}$. Then we would write the position vector in frame A and the resulting dual quaternion, representing the pose of frame B with respect to frame A , would be given as follows:

$$\check{\mathbf{q}}_{B/A} = \mathbf{q}_{B/A} + \frac{\epsilon}{2} \mathbf{q}_{B/A} \otimes \mathbf{R}_A \quad (5.24)$$

If one wants to use the vector in frame B , then a simple right multiplication of the dual part with a quaternion unity $\mathbf{I}_q = \mathbf{q}^* \otimes \mathbf{q}$ has to be done.

$$\check{\mathbf{q}}_{B/A} = \mathbf{q}_{B/A} + \frac{\epsilon}{2} \mathbf{q}_{B/A} \otimes \mathbf{R}_A \otimes \mathbf{q}_{B/A}^* \otimes \mathbf{q}_{B/A} \quad (5.25)$$

where $\mathbf{q}_{B/A} \otimes \mathbf{R}_A \otimes \mathbf{q}_{B/A}^*$ is a quaternion frame transformation for vectors in the frame A to be expressed in frame B . As a result, Eq. (5.25) becomes:

$$\check{\mathbf{q}}_{B/A} = \mathbf{q}_{B/A} + \frac{\epsilon}{2} \mathbf{R}_B \otimes \mathbf{q}_{B/A} \quad (5.26)$$

Comparing Eqs. (5.24) and (5.26), we can see that the same dual parts can be expressed in two different frames by just switching the places of the multiplicands.

To retrieve the attitude quaternion and the position vector from a dual quaternion, one has to do the following steps:

$$\mathbf{q}_{B/A} = \mathbf{q}_r \quad (5.27)$$

$$\mathbf{R}_A = 2\mathbf{q}_r^* \otimes \mathbf{q}_d \quad (5.28)$$

5.3. Screw Displacement

Chasle's theorem states that any rigid displacement is equivalent to a rotation around a line, called the screw axis, followed by a translation in the direction of the line (Jia, 2013). This allows us to visualize a dual quaternion in a similar way a quaternion can be visualized (Euler theorem). The geometry of this process is depicted in Figure 5.1.

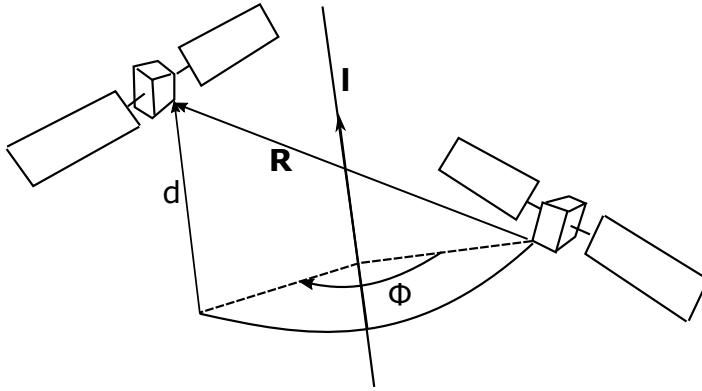


Figure 5.1: Screw transformation visualized.

Suppose an initial pose of a spacecraft as depicted on the right in Figure 5.1, and a final pose as shown on the left of the same figure. Then, the transformation from the initial to the final pose is described as a simultaneous translation and rotation along the line \mathbf{l} . The spacecraft is rotated by an angle Φ and translated by a distance d . The final frame is displaced by a vector \mathbf{R} (expressed in the initial frame) with respect to the initial frame. The screw axis is expressed in Plucker line coordinates, which will be detailed now.

5.3.1. Plucker line

A line in space can be described by a direction vector, \mathbf{l} , and a point, \mathbf{p} , that lies on the line, expressed in a general reference frame, as shown in Figure 5.2. The Plucker line coordinates are defined to be (\mathbf{l}, \mathbf{m}) , where \mathbf{m} is the momentum of a line, $\mathbf{m} = \mathbf{p} \times \mathbf{l}$. These coordinates make the choice of point \mathbf{p} independent, since the norm of the cross product always gives the shortest distance from the origin to the line.

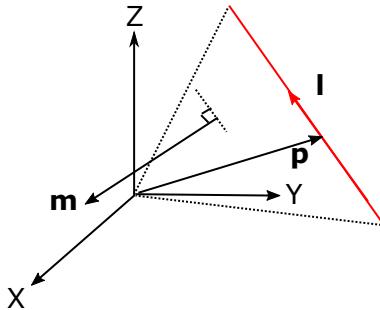


Figure 5.2: Plucker coordinates.

If one uses a unit vector with respect to the origin of a coordinates system for the direction vector, \mathbf{l} , then the moment of the line will have the same length as the shortest distance to the line. Furthermore, vectors \mathbf{m} and \mathbf{l} are orthogonal, therefore the Plucker coordinates have two constrains: $\|\mathbf{l}\| = 1$ and $\mathbf{l} \cdot \mathbf{m} = 0$. As a result, there are only four degrees of freedom in representing a line in space.

5.3.2. Screw Coordinates

Having the Plucker coordinates defined, the screw parameters for a pose transformation can be written down. As shown in Figure 5.1, a transformation between two arbitrary positions and orientations can be described by the screw axis, around which the frame is rotated by the angle, Φ , also referred as a pitch angle (do not confuse it with the pitch Euler angle), and, along which it is translated by the distance, d . Then the screw coordinates are the Plucker line coordinates \mathbf{l} and \mathbf{m} , plus the angle Φ and distance d . These (eight) parameters can be extracted from a dual quaternion.

Line vector \mathbf{l} is essentially the rotation axis \mathbf{n} that is a part of a quaternion of rotation in Eq. (3.26). Similarly, the pitch angle is calculated from the fourth component of the unit quaternion. The distance is calculated as follows:

$$d = \mathbf{R} \cdot \mathbf{l} \quad (5.29)$$

The line momentum (expressed in the initial frame) requires more effort to derive. As given by Jia (2013), it can be calculated as:

$$\mathbf{m} = \frac{1}{2} \left(\mathbf{R} \times \mathbf{l} + \mathbf{l} \times (\mathbf{R} \times \mathbf{l}) \cot \frac{\Phi}{2} \right) \quad (5.30)$$

5.3.3. Example

A simple example is presented to understand how the dual quaternion representation can be visualized. Suppose a reference frame (blue) that is $(2 \ 2 \ 2)^T$ distance units away from the origin of another reference frame (red), as shown in Figure 5.3. The first one is rotated 180° around X-axis, thus its attitude is $(1 \ 0 \ 0 \ 0)^T$.

Now, we can construct the dual quaternion representing the pose of the blue frame with respect to the red one. The real part of the DQ is the attitude quaternion itself, and the dual part, with the

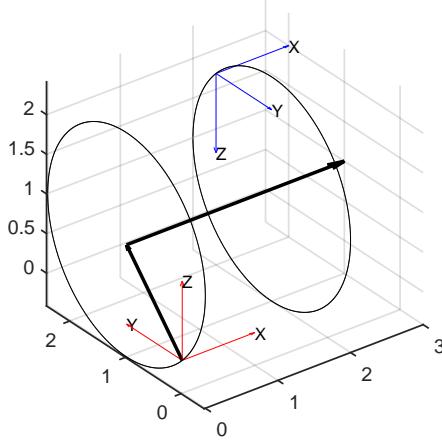


Figure 5.3: Dual quaternion representation visualized.

help of Eq. (5.24), is as follows:

$$\begin{aligned} \mathbf{q}_d &= \frac{1}{2} \mathbf{q}_{B/A} \otimes \mathbf{R}_A = \frac{1}{2} \begin{bmatrix} q_4 & q_3 & -q_2 & q_1 \\ -q_3 & q_4 & q_1 & q_2 \\ q_2 & -q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{pmatrix} R_x \\ R_y \\ R_z \\ 0 \end{pmatrix} = \\ &= \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} 2 \\ 2 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ -1 \\ -1 \end{pmatrix} \end{aligned} \quad (5.31)$$

The dual quaternion is then given as:

$$\check{\mathbf{q}} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \\ -1 \end{pmatrix} \quad (5.32)$$

The screw parameters can be recovered from this dual quaternion. The pitch angle and the distance are $\Phi = 180^\circ$ and $d = 2$, respectively. The line vector is $\mathbf{l} = (100)^T$ and the point on the line (calculate from the momentum) is $\mathbf{p} = (011)^T$. The screw axis is depicted in Figure 5.3 with the thick black arrow. To conclude, the red frame would be translated by 2 units and rotated by 180° around the black axis and then the two frames would coincide.

This example sums up the introduction of dual quaternions. The more in depth derivation of kinematics equations and other properties are discussed in Chapter 7.

6

Simulator Design

The starting point of practical work of this thesis is to develop a simulator, which would represent the actual dynamics of a spacecraft in the vicinity of an asteroid. Since the work does not include any hardware in the loop, as it is out of the scope of this thesis, the necessary attitude and position sensors are modelled as well. The structure of the software is given in different levels, and the actual implementation is discussed, following the verification of each block. The architecture is given in diagrams, where quadrangles with oppositely inclined sides denote inputs (also in a light grey colour), rectangles with double sides mean the subroutines and parallelograms denote outputs. The chapter starts with the state vector being defined for inertial and relative states. Section 6.2 introduces a toolkit that is used to retrieve data from various space missions, and Section 6.3 defines the top level architecture of the software. The chapter continues with a model of the asteroids and the spacecraft that are used in this thesis, Sections 6.4 and 6.5 respectively. The implementation of dynamics equations is presented in Section 6.6, which is followed by validation in Section 6.7. Section 6.8 introduces the integrator to be used, and, finally, the hardware modelling is discussed in Section 6.9.

6.1. State Vectors

The simulator outputs the inertial state \mathbf{X}_I and the relative state \mathbf{X}_A (state relative to an asteroid frame, A).

$$\mathbf{X}_I = \begin{pmatrix} \mathbf{R}_I \\ \mathbf{V}_I \\ \mathbf{q}_{B/I} \\ \boldsymbol{\omega}_{B/I}^B \\ \mathbf{q}_{A/I} \end{pmatrix}, \quad \mathbf{X}_A = \begin{pmatrix} \mathbf{R}_A \\ \mathbf{V}_A \\ \mathbf{q}_{B/A} \\ \boldsymbol{\omega}_{B/A}^B \end{pmatrix} \quad (6.1)$$

The dynamic equations are written in the inertial frame, thus the inertial states are the direct output of the simulations. The relative states are computed as a transformation of the inertial states, as it was discussed in Section 3.5.

6.2. SPICE

Spacecraft, Planet, Instrument, C-matrix, Events (SPICE) is a software toolkit, developed by Navigation and Ancillary Information Facility (NAIF), that can be downloaded from their website¹. This toolkit, which is also available for Matlab, allows to read data from missions, so-called kernels. Kernels contain general data of ephemerides of the planets, spacecraft, coordinate frame definitions

¹<https://naif.jpl.nasa.gov/naif/toolkit.html>; date accessed: 10-11-15

and attitude. For a specific mission, each instrument frame is defined and its corresponding orientation is available.

For this thesis, SPICE will be used to retrieve Rosetta mission data for validation purposes and initial data for simulations, e.g., the Sun vector in the Asteroid frame.

6.3. Top-Level Software Architecture

The whole software, which is developed, can be separated into three major blocks, as seen in Figure 6.1. These blocks are:

- **Dynamics** - the block that calculates the translational and rotational motion of a satellite, governed by gravity and Solar-radiation pressure forces, rotation of the asteroid and gravity gradient torque. The output of the Dynamics block is the inertial state X_I and the relative state X_R .
- **Hardware** - simulation of the physical sensors: a gyroscope, star tracker, laser ranger and navigation camera. The output of this block is separated into an inertial measurement vector z_I and relative vector z_R
- **Navigation** - contains the actual navigation algorithms that would run 'on-board' the space-craft. This block takes the measurements from the simulated hardware and outputs an estimated relative state, \hat{X}_A .

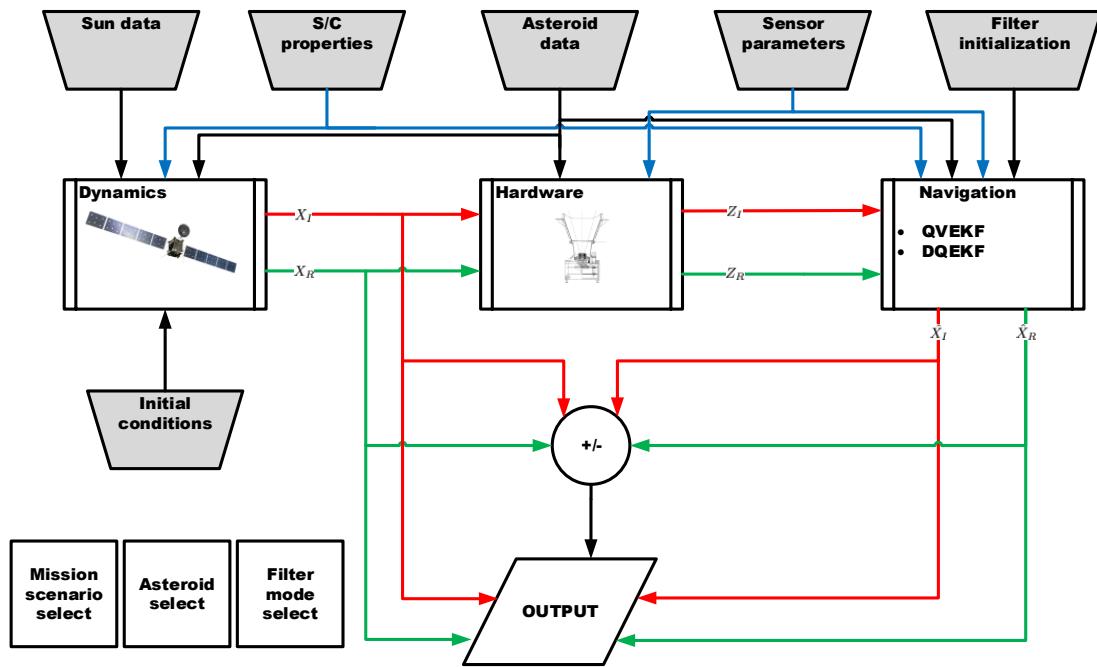


Figure 6.1: Top level software architecture.

The red and green colours of the lines in Figure 6.1 denote inertial and relative vectors, respectively. The blue line is for a more clear view of the lines. The input blocks are user inputs of different

parameters, which makes the simulator versatile and adaptable to different kinds of missions. These blocks are:

- **Sun Data** - solar flux Φ in the vicinity of an asteroid; Sun vector \mathbf{R}_I^S expressed in the I frame.
- **Spacecraft properties** - the physical parameters of the spacecraft: mass m_{SC} , inertia tensor \mathbf{I}_{SC} and a simplified 3D model of the SC with the reflectivity coefficients of each plane.
- **Asteroid Data** - contains the asteroid mass m_A , mean density ρ , angular velocity ω_{AI}^A , a 3D model given by a polyhedron model, expressed in frame A , and a landmark map.
- **Sensor Properties** - parameters of sensors: noise standard deviations σ_i , focal length of the camera.
- **Filter Initialization** - initial values for filters: states $\hat{\mathbf{X}}_{0/0}$ and covariance matrix $\mathbf{P}_{0/0}$
- **Initial Conditions** - initial state vector $\mathbf{X}_{I,0}$

The three square blocks: Mission scenario select, Asteroid select and Filter mode select are switches for changing the inputs and initial conditions for different cases.

From this top-level diagram, one can already see that the hardware block does not affect the dynamics, and the navigation does not affect either the hardware, or the dynamics blocks, because no control is simulated. This suggests that each block can be simulated separately, starting with the dynamics, then hardware and finally navigation. As a result, the navigation block could be run independently, with previously generated measurement data. This enables running the two different filter modes with the same set of measurements, thus saving the time of running the simulation of dynamics and hardware.

The following sections discuss the spacecraft model, the dynamics and hardware blocks, leaving the navigation for the following chapter.

6.4. Spacecraft Model

This section deals with the physical properties of a spacecraft modelled in the simulator. The physical dimensions, mass, inertia moments and reflectivity properties of surfaces are provided.

As a baseline for the spacecraft, Rosetta spacecraft is taken, because it is the most recently accomplished mission to a small body.

6.4.1. Dimensions

Rosetta comprises of a cuboid-shaped body and two solar panels attached on both sides. The body has dimensions of 2.1 m, 2.0 m and 2.8 m along the X-, Y- and Z-axes, respectively, see Figure 6.2. Each solar panel, which is 14 m long, has an area of 32 square meters, thus having a width of 2.3 m. They are attached to the SC at points (0 1 0) and (0 -1 0) in the B reference frame (Richner et al, 2003). The solar array has an offset of one meter from the body, so the total 'wingspan' is 32 m.

6.4.2. Mass Properties

Rosetta has a dry mass of 1300 kg and a total mass of 3000 kg². The total ΔV the spacecraft can produce is around 2200 m/s. For the orbit-correction manoeuvres, until the rendezvous with the comet, roughly 775 m/s was used ³, thus the spacecraft mass at the comet can be estimated using Tsiolkovski equation:

²<http://nssdc.gsfc.nasa.gov/nmc/spacecraftDisplay.do?id=2004-006A>; date accessed: 01-12-15

³<http://blogs.esa.int/rosetta/2014/07/01/four-complete-six-to-go-burning-down-to-comet-rendezvous/>; date accessed: 01-12-2015

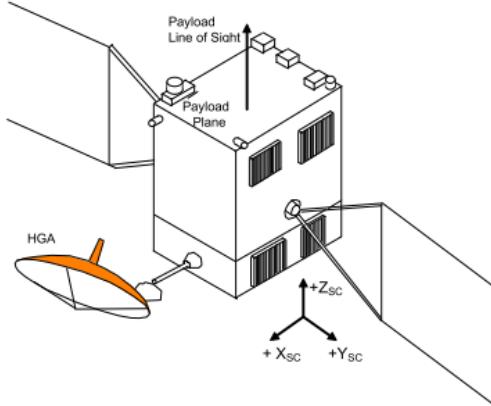


Figure 6.2: Body reference frame (Richner et al, 2003).

$$M_0 = M_t e^{-\frac{\Delta v}{v_0}} \quad (6.2)$$

where M_t is the total mass, M_0 is the remaining mass at the spacecraft, v_0 is the exhaust velocity of the engine (≈ 2500 m/s), and Δv is the planned velocity change. This gives the mass of the spacecraft $M_{SC} \approx 2100$ kg.

The moments of inertia are calculated separately for the body (cuboid) and the solar panels. Then, using the parallel axis theorem, they are added together. However, the mass of the solar panels must be known. The arrays are capable of producing 10 kW at 1 Astronomical Unit (AU), and according to Wertz et al (2011), the high-end solar arrays have a specific performance of 66 W/kg at 1 AU. This gives a rough estimate that the panels should have a total mass of ≈ 150 kg or 75 kg each. Then the body (cuboid) mass is $M_b = 2100 - 150 = 1950$ kg. The inertia tensor for a cuboid, assuming constant density, can be calculated as:

$$\mathbf{I}_b = \begin{bmatrix} \frac{1}{12} M_b (l_y^2 + l_z^2) & 0 & 0 \\ 0 & \frac{1}{12} M_b (l_x^2 + l_z^2) & 0 \\ 0 & 0 & \frac{1}{12} M_b (l_x^2 + l_y^2) \end{bmatrix} = \begin{bmatrix} 1924 & 0 & 0 \\ 0 & 1991 & 0 \\ 0 & 0 & 1364 \end{bmatrix} \text{kgm}^2 \quad (6.3)$$

where l_x , l_y , l_z are dimensions along X-, Y-, Z-axis respectively (2.1 m, 2.0 m and 2.8 m). The solar panels have normal vectors parallel to the X axis of the body frame (lying on Y-Z plane). They have dimensions $s_y = 14$ m and $s_z = 2.3$ m (s_x , the thickness, is assumed to be zero). Their principal axis of rotation are displaced by $d_x = 9$ m, $d_y = 0$ m and $d_z = 9$ m. Then, according to the parallel axis theorem, the moments of inertia of one solar-panel around B frame axes are:

$$\mathbf{I}_s = \begin{bmatrix} \frac{1}{12} M_s (s_y^2 + s_z^2) + M_s d_x^2 & 0 & 0 \\ 0 & \frac{1}{12} M_s s_z^2 & 0 \\ 0 & 0 & \frac{1}{12} M_s s_y^2 + M_s d_z^2 \end{bmatrix} = \begin{bmatrix} 7333 & 0 & 0 \\ 0 & 33 & 0 \\ 0 & 0 & 7300 \end{bmatrix} \text{kgm}^2 \quad (6.4)$$

where M_s is the mass of one solar-panel and equal to 75 kg. Finally, the total moment of inertia of the spacecraft is:

$$\mathbf{I}_{SC} = \mathbf{I}_b + 2\mathbf{I}_s = \begin{bmatrix} 16590 & 0 & 0 \\ 0 & 2057 & 0 \\ 0 & 0 & 15964 \end{bmatrix} \text{kgm}^2 \quad (6.5)$$

The moments might look large, however, as a comparison, Envisat's (ESA's veteran satellite) inertia moments are given by Virgili et al (2014):

$$\mathbf{I}_{Envisat} = \begin{bmatrix} 17023 & 0 & 0 \\ 0 & 124825 & 0 \\ 0 & 0 & 129112 \end{bmatrix} \text{kgm}^2 \quad (6.6)$$

It is clearly seen that Envisat has much larger moments of inertia, thus it is concluded that the calculated moments of inertia are within the limits of reality.

6.4.3. Reflectivity

To define the reflection properties of the spacecraft, a simplified model of it is made. This model consists of ten rectangular faces - six of the body and four of the two solar-panels (thickness of panels assumed to be zero), as shown in Figure 6.3.

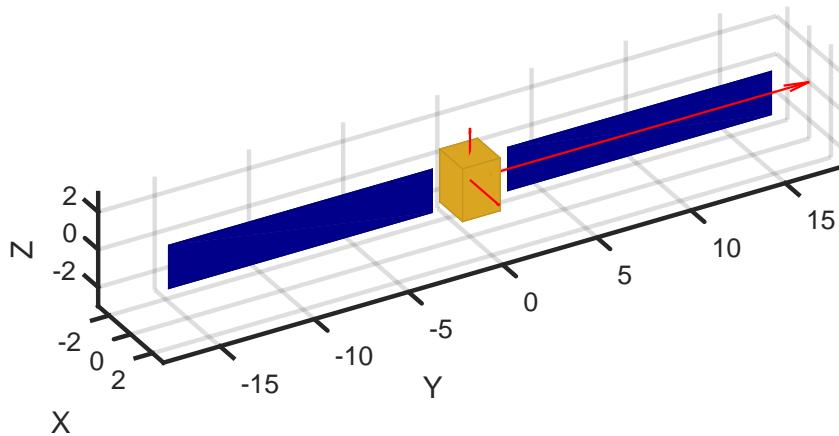


Figure 6.3: Simplified SC model.

Each of the faces has four vertices. Moreover, each of the face has a normal vector and a vector from the origin of the B frame to the centre of the face associated with them. Also, reflectivity/absorption coefficients are defined for the surface. Montenbruck and Gill (2001) give the value of reflectivity for solar panels $\epsilon = 0.21$, thus the absorption is $1 - \epsilon = 0.79$. The body is assumed to be more reflective, having $\epsilon = 0.5$.

6.5. Asteroid Model

The asteroid model includes the basic properties of the body: mass, angular velocity, mean density, the shape of the asteroid and a generated landmark map.

6.5.1. Polyhedron Model

A polyhedron model is usually given as a .txt file. It incorporates the coordinates of all the vertices in the asteroid-fixed frame. Then each face, which is triangular, is defined by the three numbers of vertices, in counter-clockwise order, if looked from the outside. Polyhedron models have different resolutions (number of vertices/faces), depending on whether the model was generated by a visiting spacecraft (Itokawa, Eros) or by ranging measurements from Earth (Kleopatra), with the latter being less accurate. The models can be obtained from the Jet Propulsion Laboratory (JPL) website⁴;

⁴<http://echo.jpl.nasa.gov/>; date accessed: 17-11-15

however, for easier search, one should go to another database⁵. In Figure 6.4, models for Kleopatra and Itokawa are given.

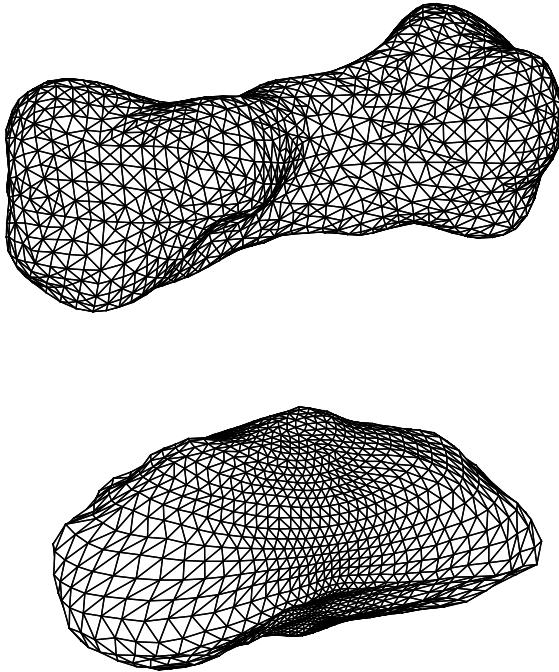


Figure 6.4: Polyhedron models of Kleopatra and Itokawa, top and bottom respectively (not in scale).

6.5.2. Landmark Generation

In general, a landmark is a distinguishable surface feature - a crater, ridge, rock, or something else, that a vision based navigation system could recognize. Taking real images and doing image processing is out of the scope of this thesis, thus landmarks are assumed to be points on the surface of the asteroid and the landmark map (LM) is assumed to be known a-priori.

Doing a visual inspection of asteroid images, one could notice that the distinct features on the surface are not evenly spread. The distribution is influenced by impacts, rotational rates, surface properties, and etc.; however it is beyond the scope of this thesis to take into account these factors. Therefore, the landmarks are distributed randomly.

Since the polyhedron asteroid model used in this thesis is made of triangular faces, the landmark generation problem changes to how to place a random point on a triangle. Fortunately, Osada et al (2002) investigated this problem in their shape-recognition paper. The equation is given as:

$$\mathbf{P} = (1 - \sqrt{r_1})\mathbf{A} + \sqrt{r_1}(1 - r_2)\mathbf{B} + \sqrt{r_1}r_2\mathbf{C} \quad (6.7)$$

where \mathbf{A} , \mathbf{B} and \mathbf{C} are the vertices of a triangle and r_1, r_2 are uniformly distributed random numbers, $r_1, r_2 \in [0, 1]$. Using this equation, one can generate random points that lie on a triangle. As an example, Figure 6.5 shows 1000 points spread on an arbitrary triangle.

Now, the total number of landmarks has to be decided. The current Eros model, according to Gaskell et al (2006), has 4900 landmarks. Having a surface area of 1125 km^2 , it would result in 4.35 LM/km^2 . Takeishi (2013) used 213 detected landmarks for Itokawa, which has a surface area of

⁵<http://space.frieger.com/asteroids/>; date accessed: 17-11-15

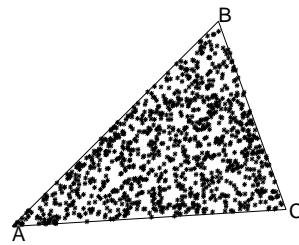


Figure 6.5: Random points placed on a triangle.

0.404 km^2 , thus a density of 527 LM/km^2 , while Dionne (2009) uses a value of 0.04 LM/km^2 for an arbitrary ellipsoid shaped asteroid, resulting in 124 landmarks. It is clear that the density could vary significantly with the size of the asteroid, thus it might not be the most useful parameter. However, the total number of LM varies only from 100 to 5000. As a result, the total number of LM will be taken as an input parameter, but keeping the possibility to change the density as well.

Once the total number of landmarks is decided, then the same number of random faces are taken and a point is placed randomly on it. Note that some faces can have a few landmarks, while others none. The landmark map on Kleopatra asteroid was generated with number of landmarks $N_{LM} = 500, 1000, 2000, 4000$, as shown in Figure 6.6.

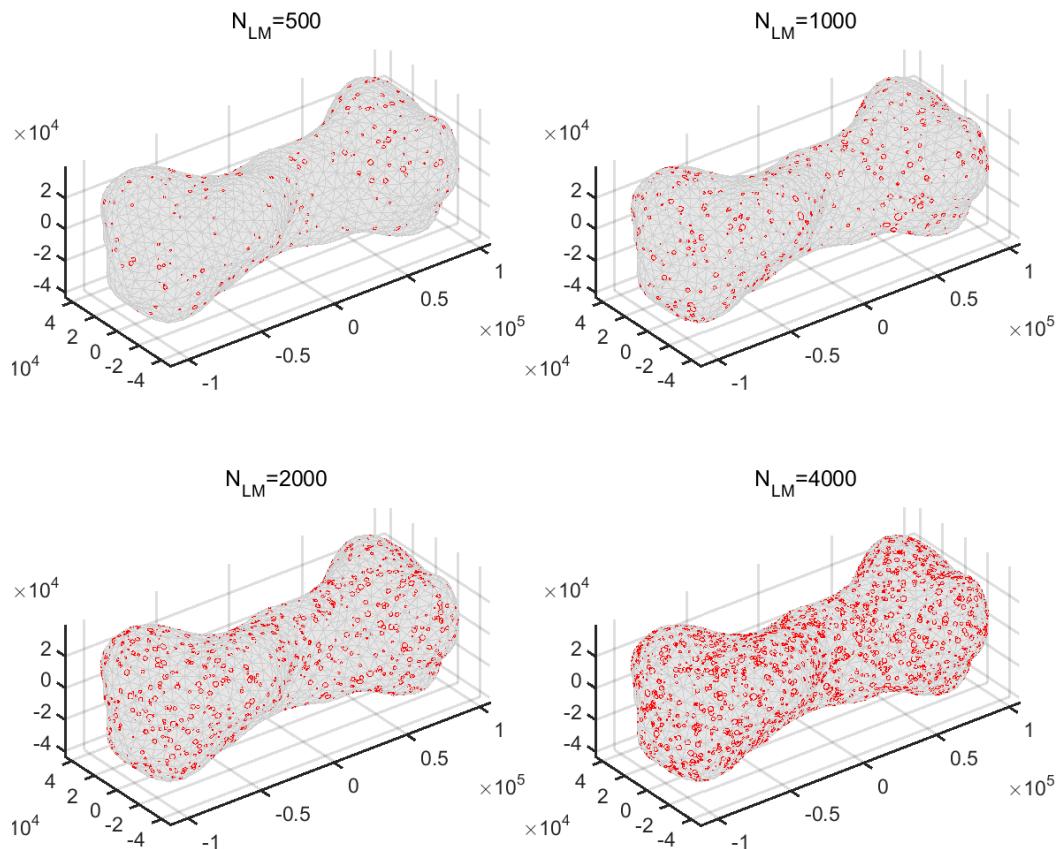


Figure 6.6: Generated landmarks for Kleopatra.

6.6. Dynamics Block

The dynamics block simulates translational and rotational motion of the spacecraft in the vicinity of an asteroid. It also propagates the attitude of an asteroid. The software diagram can be seen in Figure 6.7. The upper part, with the red surrounding box, denotes the simulated inertial state, and the lower green one is just the computed relative state. In this section, each piece of this diagram is examined and verified.

6.6.1. Translational motion

Let us start with the translational motion (in the upper right side of Figure 6.7). The block consists of an integration of dynamics equations, discussed in Section 3.2. The input of this part is the sum of all acting forces divided by the mass of the spacecraft, and the outputs are the position, \mathbf{R}_I , and velocity, \mathbf{V}_I , vectors. To verify that this part is working properly, let us assume a point-mass gravity field acting on a spacecraft:

$$\mathbf{F}_I = G \frac{Mm_{SC}}{R^3} \mathbf{R}_I \quad (6.8)$$

For verification a sphere having 50 km radius is taken with the mean density of $\rho = 5000 \text{ kg/m}^3$, thus having mass $M = 2.618 \times 10^{18} \text{ kg}$. Having only the point-mass gravity and no other forces acting on the spacecraft, one would expect to get a precise Keplerian orbit with constant energy. The circular orbit velocity at an altitude of 10 km is:

$$V_{circ} = \sqrt{\frac{GM}{R_{orb}}} = 53.964 \text{ m/s} \quad (6.9)$$

Thus, initial conditions were set for an equatorial orbit:

$$\mathbf{R}_0 = \begin{pmatrix} 0 \\ 60 \\ 0 \end{pmatrix} \text{ km}, \quad \mathbf{V}_0 = \begin{pmatrix} 53.964 \\ 0 \\ 0 \end{pmatrix} \text{ m/s} \quad (6.10)$$

and the simulation was run for 10,000 seconds to capture at least one period of the orbit, which can be calculated as:

$$T = 2\pi \sqrt{\frac{R_{orb}^3}{GM}} \approx 6,986 \text{ s} \quad (6.11)$$

Figure 6.8 presents the results obtained by simulation. The graph shows the position of an orbiting body around the point-mass gravity field. Since the initial conditions were set to lie in the equatorial (X-Y) plane, the Z component stays zero for all of the simulation time. The points, where the X component crosses zero a second time, gives the period, which, if zoomed in, would turn out to be exactly the same as in Eq. (6.11). The total energy and the gravitational acceleration in this orbit remain constant, -1456.058 J and 0.048535 m/s^2 respectively. Furthermore, the radius of the orbit does not change in time, so the orbit stays circular, as it was initialized.

These results prove that the orbit does not change its shape nor orientation, has a constant energy, and the calculated value of period matches the simulated one; thus the translation block works perfectly.

6.6.2. Polyhedron Gravity Field

Asteroids are highly irregular heavenly bodies. Thus, assuming their gravity as a point-mass, would be highly inaccurate. As already discussed in Section 3.2, the most accurate gravity model is a constant density polyhedron model (if a constant density is assumed). The gradient of potential is then

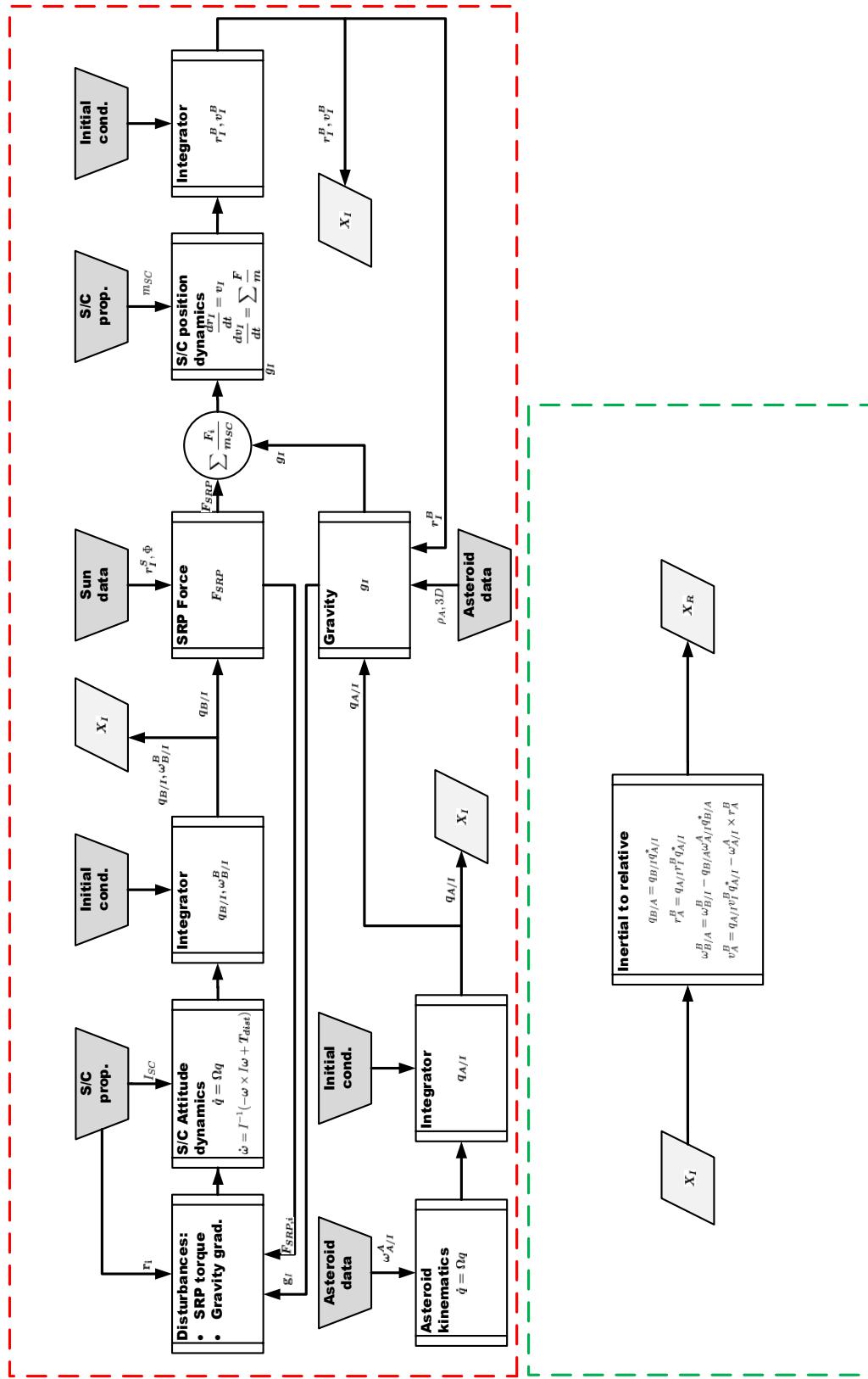


Figure 6.7: Simulator dynamics block architecture.

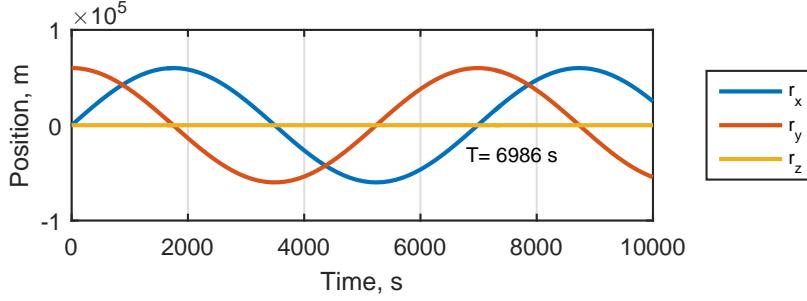


Figure 6.8: Simulated motion with point-mass gravity.

given as:

$$\nabla U = -G\sigma \sum_{e \in \text{edges}} \mathbf{E}_e \cdot \mathbf{r}_e L_e + G\sigma \sum_{f \in \text{faces}} \mathbf{F}_f \cdot \mathbf{r}_f \cdot \omega_f \quad (6.12)$$

To implement this gravity-field expression, the polyhedron model (Section 6.5) of Kleopatra and Itokawa asteroids are used. The first term in Equation (6.12) denotes the summation over all the edges, and the second one - over faces. The dyadic matrices \mathbf{E}_e and \mathbf{F}_f are defined as:

$$\mathbf{E}_e = \mathbf{n}_A \mathbf{n}_{12}^A + \mathbf{n}_B \mathbf{n}_{21}^B \quad (6.13)$$

and:

$$\mathbf{F}_f = \mathbf{n}_f \mathbf{n}_f \quad (6.14)$$

where \mathbf{n}_f , \mathbf{n}_A and \mathbf{n}_B are unit normal vectors to the facet, Figure 6.9. Knowing the vectors to the vertices (these are defined in the polyhedron file), the normal vectors can be found by a cross product:

$$\mathbf{n}_f = \frac{(\mathbf{r}_2 - \mathbf{r}_1) \times (\mathbf{r}_3 - \mathbf{r}_2)}{\|(\mathbf{r}_2 - \mathbf{r}_1) \times (\mathbf{r}_3 - \mathbf{r}_2)\|} \quad (6.15)$$

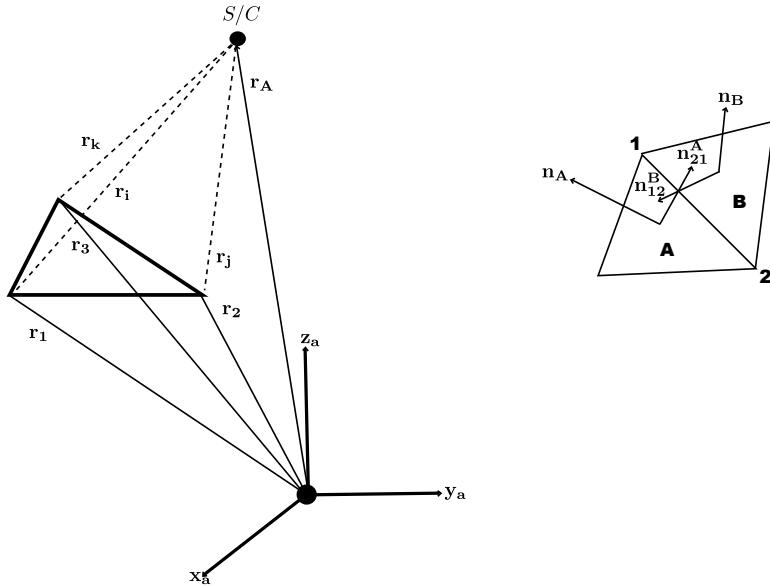


Figure 6.9: An edge at two facets.

where \mathbf{n}_{12} and \mathbf{n}_{21} are unit vectors normal to the face and the edge, see the right side of Figure 6.9. They are calculated in such a way that the cross product of this vector with the edge vector (counter-clockwise direction) would give the normal to the face:

$$\mathbf{n}_{21}^A = \frac{(\mathbf{r}_1 - \mathbf{r}_2) \times \mathbf{n}_A}{\|(\mathbf{r}_1 - \mathbf{r}_2) \times \mathbf{n}_A\|} \quad (6.16)$$

and similarly:

$$\mathbf{n}_{12}^B = \frac{(\mathbf{r}_2 - \mathbf{r}_1) \times \mathbf{n}_B}{\|(\mathbf{r}_2 - \mathbf{r}_1) \times \mathbf{n}_B\|} \quad (6.17)$$

It is seen from Eqs. (6.16) and (6.17) that the two matrices E_e and F_f do not depend on the field point, in which the gravity field is calculated. Thus they can be precomputed for each edge and face in advance, thus decreasing time for simulations.

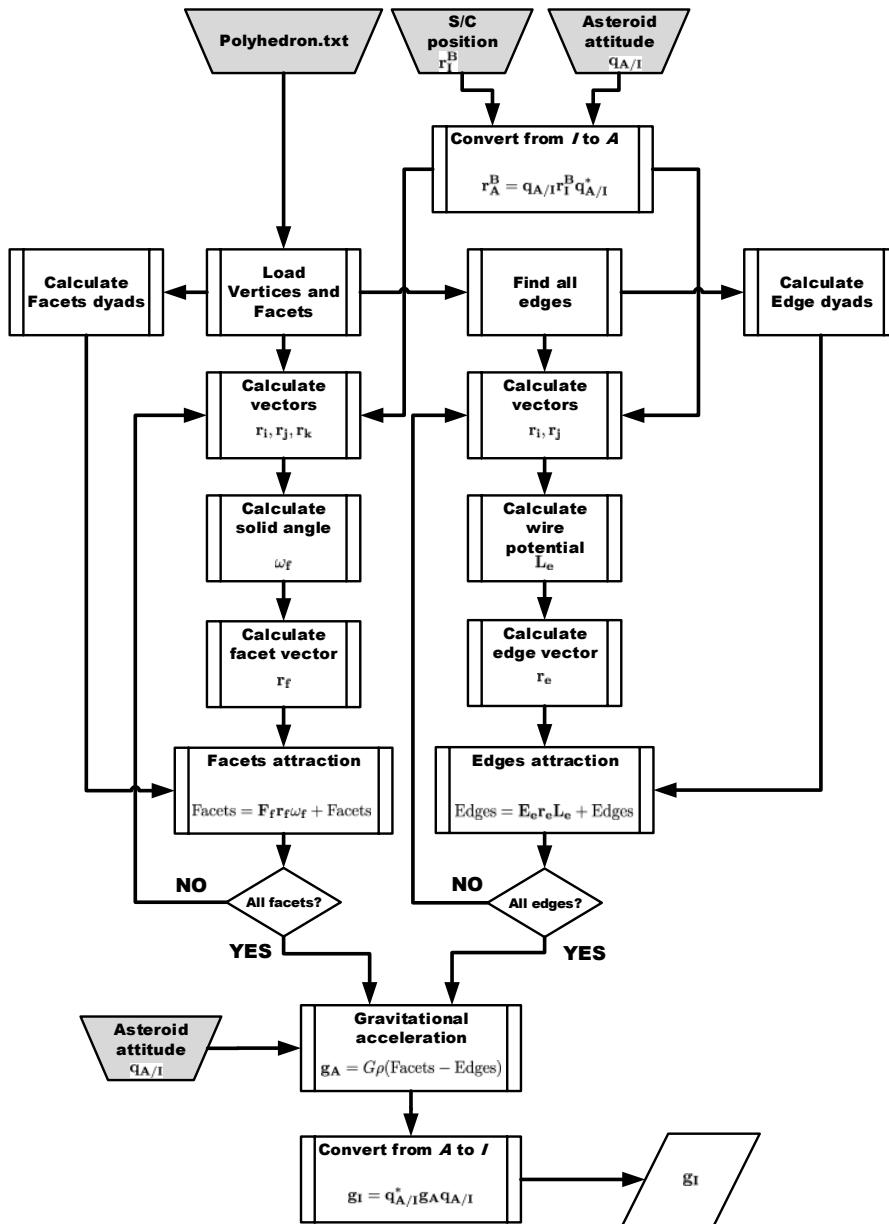


Figure 6.10: Architecture of polyhedron gravity field computation.

Let us define the vectors \mathbf{r}_i , \mathbf{r}_j and \mathbf{r}_k . These are the vectors from the field point (spacecraft), outside the main body (asteroid), to the vertices, see Figure 6.9. They can be found from vertices vectors \mathbf{r}_1 , \mathbf{r}_2 , \mathbf{r}_3 and the spacecraft position in the A frame, \mathbf{r}_A :

$$\mathbf{r}_i = \mathbf{r}_1 - \mathbf{r}_A \quad (6.18)$$

$$\mathbf{r}_j = \mathbf{r}_2 - \mathbf{r}_A \quad (6.19)$$

$$\mathbf{r}_k = \mathbf{r}_3 - \mathbf{r}_A \quad (6.20)$$

These vectors have magnitudes r_i , r_j and r_k respectively. Then, the potential of a wire L_e and the dimensionless per-face factor ω_f can be computed using Eqs. (3.6) and (3.7). Finally, vectors \mathbf{r}_e and \mathbf{r}_f are arbitrary vectors from a spacecraft to an edge and a face respectively. They are defined to lie in the middle of an edge and a triangular face.

$$\mathbf{r}_e = \frac{\mathbf{r}_i + \mathbf{r}_j}{2} \quad (6.21)$$

$$\mathbf{r}_f = \frac{\mathbf{r}_i + \mathbf{r}_j + \mathbf{r}_k}{3} \quad (6.22)$$

The structure of the polyhedron gravity-algorithm is given in Figure 6.10.

The verification of the polyhedron gravity-field is divided into several steps. First of all, the gravity is calculated for a perfectly shaped body, a cuboid, and compared to the values given by other authors. This verifies that the algorithm is working well. Then, a polyhedron model of a sphere is taken and the computed results are compared to analytical solutions of point-mass gravity. This also includes dynamic simulations of translational motion.

Gravity of a cuboid A cuboid is chosen, due to the fact that it can be represented perfectly by a polyhedron model. There are no approximations of the shape.

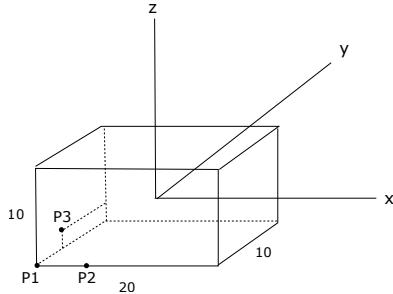


Figure 6.11: Points on a box

In the papers of Tsoulis (2012) and D'Urso (2013) the gravity field of a cuboid ($20 \times 10 \times 10$ m) is evaluated at specific points, $P1 = (-10 - 5 - 5)$, $P2 = (-5 - 5 - 5)$ and $P3 = (-10 - 2 - 3)$, Figure 6.11. The authors use a similar gravity calculation method for polyhedron bodies. The difference is that the volume integral is changed to line integrals, so the summation is done only through the lines of the body, but this does not matter as far as the values match. Special care has to be taken with the line potential value L_e . Since two of the proposed points lie on the edge of the cuboid, the logarithm in Eq. (3.6) goes to infinity. On the other hand, the edge vector \mathbf{r}_e goes to zero, thus there is a singularity. However, it can be proved that the limit of this function approaching zero is zero. As a result, an *if* statement is included into the code, and if L_e becomes infinity, it is set to zero. Moreover, Tsoulis (2012) and D'Urso (2013) use the gravitational constant value $G^* = 6.67259 \times 10^{-11} \text{m}^3 \text{kg}^{-1} \text{s}^{-2}$ instead of $G = 6.67408 \times 10^{-11} \text{m}^3 \text{kg}^{-1} \text{s}^{-2}$ used in this thesis. For the sake of comparison, the gravity of cuboid will be computed with the G^* value. The results are given in Table 6.1

Table 6.1: Comparision of cuboid gravity results.

Point	Units	Calculated value	Error, Tsoulis (2012)	Error, D'Urso (2013)
P1	$U, \text{m}^2/\text{s}^2$	$3.19403761604211 \times 10^{-5}$	7.3911389×10^{-8}	1.060×10^{-15}
	$U_x, \text{m/s}^2$	$2.31329148957265 \times 10^{-6}$	7.3911389×10^{-8}	0
	$U_y, \text{m/s}^2$	$1.91973919943187 \times 10^{-6}$	7.3911388×10^{-8}	0
	$U_z, \text{m/s}^2$	$1.91973919943187 \times 10^{-6}$	7.3911388×10^{-8}	0
P2	$U, \text{m}^2/\text{s}^2$	$3.99993558939122 \times 10^{-5}$	7.3911390×10^{-8}	1.524×10^{-15}
	$U_x, \text{m/s}^2$	$9.9011553489007 \times 10^{-7}$	7.3911391×10^{-8}	4.063×10^{-15}
	$U_y, \text{m/s}^2$	$3.24128042248715 \times 10^{-6}$	7.3911390×10^{-8}	0
	$U_z, \text{m/s}^2$	$3.24128042248715 \times 10^{-6}$	7.3911390×10^{-8}	0
P3	$U, \text{m}^2/\text{s}^2$	$4.03528375471853 \times 10^{-5}$	7.3911390×10^{-8}	1.008×10^{-15}
	$U_x, \text{m/s}^2$	$4.73368592565013 \times 10^{-6}$	7.3911389×10^{-8}	0
	$U_y, \text{m/s}^2$	$9.6816436289255 \times 10^{-7}$	7.3911391×10^{-8}	4.155×10^{-15}
	$U_z, \text{m/s}^2$	$1.59674500375495 \times 10^{-6}$	7.3911389×10^{-8}	0

Table 6.1 provides values of the potential U and the gradients of potential, U_x , U_y and U_z , evaluated at points P1, P2 and P3. The errors are given as $\frac{|A-B|}{B}$, where A is the calculated value and B is the value given by the authors. The calculated values match the ones of D'Urso (2013) up to 14 significant digits. The values of Tsoulis (2012) match 6-7 significant digits. As a result, we conclude that the algorithm works as it should.

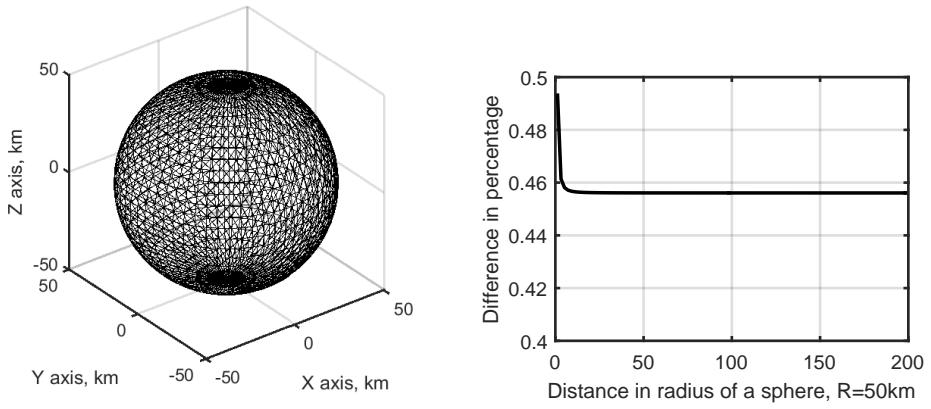
Polyhedron Sphere To check how well a polyhedron model represents a sphere, we will take a same size, 50 km radius sphere, as we used in translational motion verification. The sphere is drawn by Autodesk Inventor software and saved as an .stl file. The file is defined basically the same as the regular polyhedron file and, using a Matlab function, it can be imported as a list of vertices and facets. The sphere was drawn with a relatively low resolution, i.e., 3480 facets, Figure 6.12a. It will be called **Sphere 1**.

The gravity field of this sphere, having a mean density $\rho = 5000 \text{ kg/m}^3$, is computed with the polyhedron gravity model and compared to the analytical point-mass gravity solution. First of all, the gravitational acceleration is computed along the X-axis ($Y=Z=0$). At far distances the polyhedron and analytical solutions should converge. The difference of the theoretical gravitational acceleration, g_t , and the polyhedron one, g_p , is given as a percentage:

$$g_{diff} = \frac{g_t - g_p}{g_t} \cdot 100\% \quad (6.23)$$

where $g_t = \frac{GM}{R^2}$. The results are given in Figure 6.12b

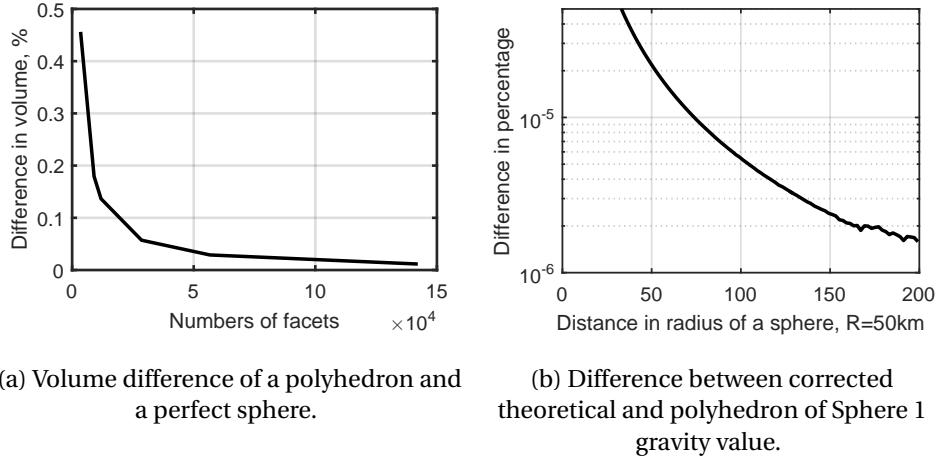
The results show a steady state error of about 0.46 %, which should not be there. However looking at how the polyhedron model is made, one would understand that the polyhedron is only an approximation of a sphere with triangular faces, which results in lower volume than that of a perfect sphere. This turns into a different mass, which results in a constant error in the gravity field. Fortunately, there are ways to calculate the volume of the polyhedron model, thus the difference can be taken into account. The volume of Sphere 1 is $V_1 = 5.2121 \times 10^{14} \text{ m}^3$, while the volume of the perfect sphere is $V_t = 5.2360 \times 10^{14} \text{ m}^3$; this gives a difference of 0.4561 %.



(a) Polyhedron model of a sphere.

(b) Difference between the theoretical and polyhedron of Sphere 1 gravity value.

Figure 6.12: Representation of a sphere with a polyhedron model



(a) Volume difference of a polyhedron and a perfect sphere.

(b) Difference between corrected theoretical and polyhedron of Sphere 1 gravity value.

Figure 6.13: Volume and gravity differences between a perfect and a polyhedron sphere.

If we take more faces, the volume difference decreases, Figure 6.13a, which shows that the more facets one takes, the better representation of the real shape it is. Now, if this volume difference is taken into account when calculating the theoretical value, better results are obtained, Figure 6.13b.

Comparing Figures 6.12b and 6.13b, a large improvement is seen in the latter. There is an obvious convergence of the two gravity models, although, at far distances, approximately 170 radii, the error starts having a noise-like behaviour, Figure 6.13b. Since the gravity field at this distance is very weak, in the order of 10^{-6} m/s^2 , and the error in percentages between the models is close to $10^{-6}\%$, then the absolute error is of order 10^{-14} m/s^2 . This is most likely caused by numerical errors, since the gravity is calculated by summations over edges and facets. In the Sphere-1 case, there were 5220 edges⁶ and 3,480 facets, which results in about 8000 summations of small numbers, thus the error can build up.

Furthermore, three more spheres are defined. Sphere 2 has 9,024 facets, Sphere 3 has 28,560 and Sphere 4 has 56,640. The same comparison of the gravity field along X-axis is done, Figure 6.14.

All four spheres with a different number of faces converge to the theoretical value, but at a dif-

⁶In triangular facet polyhedron, there are always 1.5 times more edges than facets

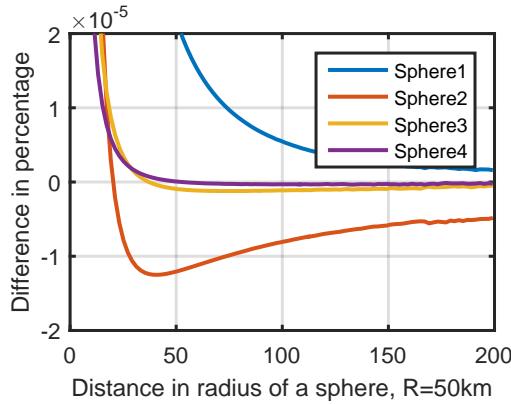
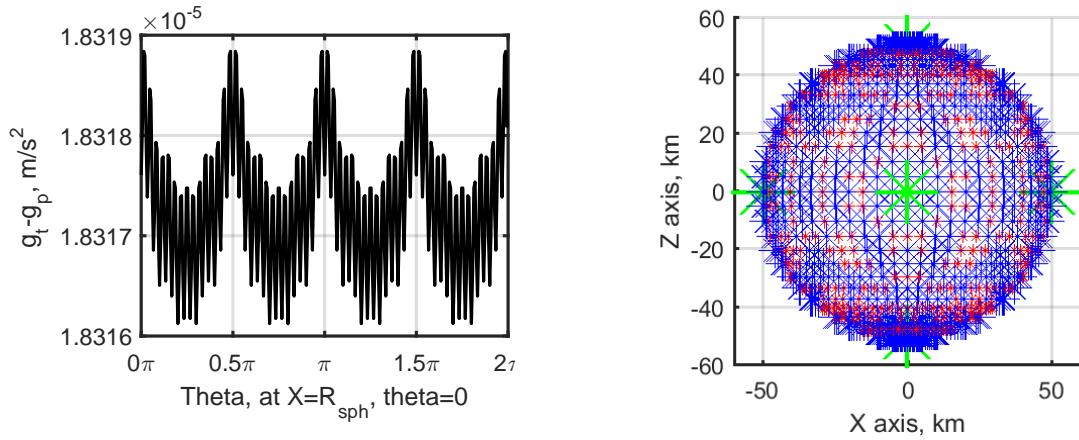


Figure 6.14: Comparison of different sphere resolution with theoretical value.

ferent rate. Sphere 2 crosses zero the fastest, however it overshoots and only then starts converging to zero, while Sphere 3 and Sphere 4 converge to zero directly, with the latter being the most precise.

The presented simulations prove that the polyhedron gravity model works, since the results converge to theoretical values at large distances, and the convergence is more rapid for spheres with larger number of facets, which is a logical outcome, as the polyhedron with more faces represents the real shape better.



(a) Gravity field comparison in a 60 km radius circle around Sphere 1.

(b) Vertices of Sphere 1.

Figure 6.15: Small variations of the gravity field around the sphere caused by different lengths of the vertices.

Next, the gravitational acceleration is computed in a circle (orbit) of $R=60$ km in X-Y plane for Sphere 1. In theory, the magnitude of g should stay constant, as the distance does not change. However, the results in Figure 6.15a are different. Although the error is very small, of the order 10^{-5} m/s^2 , which makes around 0.04% relative difference, it produces a strange pattern. The high frequency oscillations are most likely caused by the 'edgy' nature of the sphere. If one calculated the number of periods in these oscillations, it would exactly match the number of vertices around the equator ($Z=0$ plane), which is 50 for Sphere 1. Now the low frequency peaks at $0, 0.5\pi, \pi$ and 1.5π require a closer look to the way the sphere is constructed. One would expect that all vertices of the polyhedron are lying on the real sphere, thus having a magnitude of 50 km (theoretically the norm of vertices positions should be equal to the radius of the sphere). However, the actual length of these vertices vary. Figure 6.15b shows the vertices as points around the equator. The blue colour

denotes a vertex that is smaller than 50 km, red larger, and green precisely 50 km.

At the points of $X=0$ km, $Y=+/-50$ km, $Z=0$ km and $X=+/-50$ km, $Y=0$ km, $Z=0$ km, which represent 0 , 0.5π , π and 1.5π , there are green dots, meaning these vertices are exactly 50 km. However, in the area around these points, there is a noticeable blue dot area (right-left and above-below), meaning that this part is smaller than 50 km. The standard deviation of the vertices lengths is $\sigma \approx 3$ mm = 0.003 m. To get a rough estimate of how this error would affect the gravitational acceleration, the point-mass gravity field error is calculated as follows:

$$\Delta g = \frac{GM}{R_{orb}^2} - \frac{GM}{(R_{orb} + \sigma)^2} \approx 5 \times 10^{-9} \text{ m/s}^2 \quad (6.24)$$

If we take another look at Figure 6.15b, we see a peak-to-peak error of about 3×10^{-9} m/s², which is of the same order as the one calculated in Eq. (6.24).

As it was done for the point-mass gravity, the same dynamic simulation of translational motion is done with the polyhedron model.

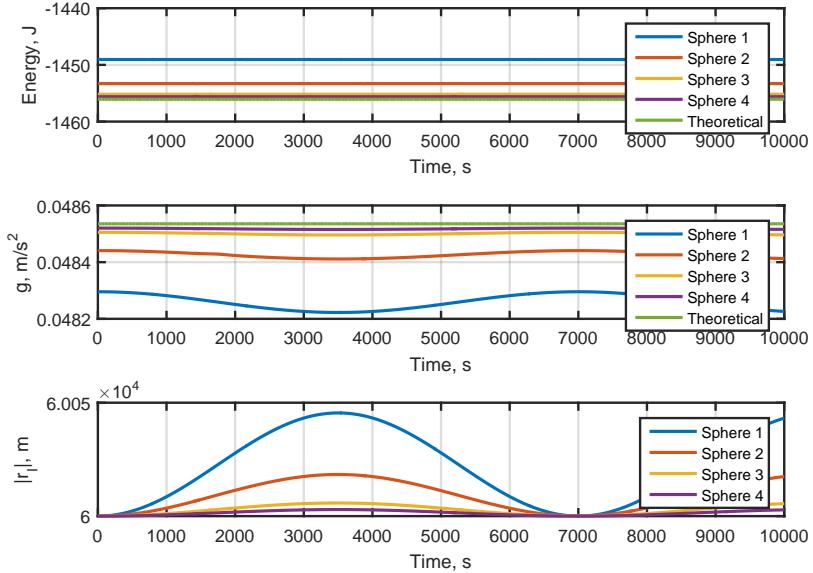


Figure 6.16: Dynamic simulation results, $R_{orb}=60$ km.

Figure 6.16 shows the orbital energy, gravitational acceleration and distance evolution of a 60 km orbit using the four sphere models. All the values show a rapid convergence to the theoretical values with increasing number of faces. For example, the orbital energy value for Sphere-1 is around 7 J larger than the theoretical one, while for Sphere-4 it is only 0.5 J larger, thus the error is 14 times smaller. We recall that Sphere-1 has 3,480 and Sphere-4 has 56,640 faces, so the latter has around 16 times more faces than the first one. This number is very similar to the number of times the error in energy value has decreased, thus there is a strong correlation between the accuracy of the polyhedron model and the accuracy of values plotted in Figure 6.16.

Finally, the gravity fields are computed for Kleopatra and Itokawa in X-Y plane, see Figure 6.17. In a central gravity field, the values of g would appear as concentric circles, while for Itokawa and especially Kleopatra the polyhedron gravity field follows the irregular shapes of the asteroids. Kleopatra and Itokawa have the maximum acceleration of about 0.05 m/s² and about 5×10^{-5} m/s², respectively, thus the gravity field is around 1000 times larger around Kleopatra than it is around Itokawa.

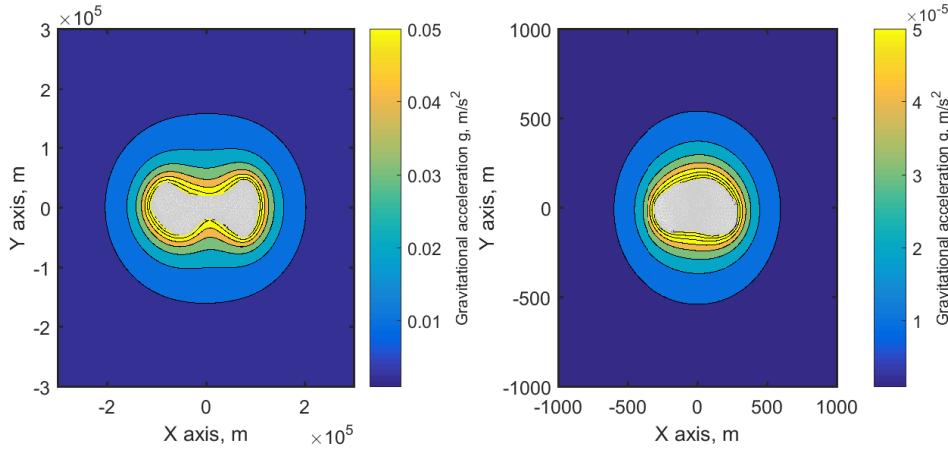


Figure 6.17: Gravity fields of Kleopatra (left) and Itokawa (right) at X-Y plane

This discussion sums up that a polyhedron gravity model may come very close to the real (analytical) solutions with more accurate models, but will never be exactly the same, since numerical errors and approximations are introduced.

6.6.3. Solar Radiation Pressure Force

The solar radiation pressure (SRP) force is modelled assuming that the spacecraft is made of N plates, each having an area A_i and reflectivity ϵ_i . Equation (3.10) shows that the total SRP force can be calculated as a sum over the individual plates. For the implementation strategy, see Figure 6.18.

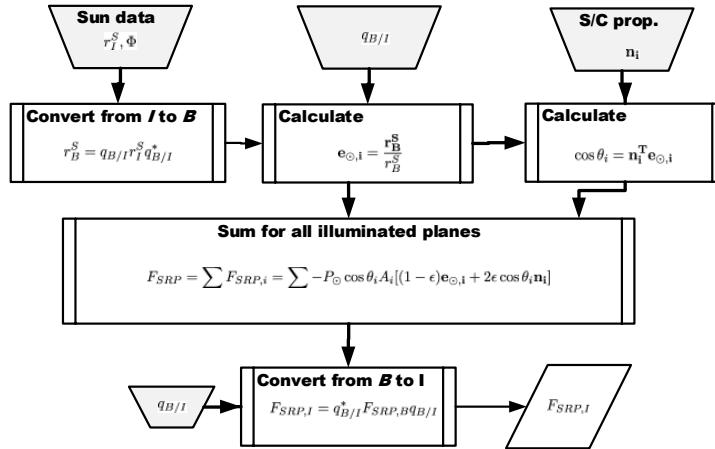


Figure 6.18: SRP force functional diagram.

To verify that the SRP model works as it should, variation equations, given by Wakker (2010), are used:

$$|\Delta a|_{max} = 2 \frac{f}{n_0^2} \cos \alpha_0 \quad (6.25)$$

$$|\Delta i|_{max} = \frac{f}{n_0^2 r_0} \sin \alpha_0 \quad (6.26)$$

$$|\Delta \Omega|_{max} = 2 \frac{f}{n_0^2 r_0} \sin \alpha_0 \quad (6.27)$$

where a , i and Ω are the semi-major axis, inclination and right ascension of ascending node respectively, f is the SRP acceleration, n_0 is the mean motion in orbit and α_0 is the angle between orbital plane and sun direction. Equations (6.25) to (6.27) give the maximum variations of orbital elements over one period of orbit.

A circular orbit of 100 km in a central gravity field with gravitational parameter equal to the one of Kleopatra's is simulated. A polar orbit with its plane perpendicular to the Sun vector is initialized and spacecraft attitude is kept constant, so that the solar panels would always point towards the Sun.

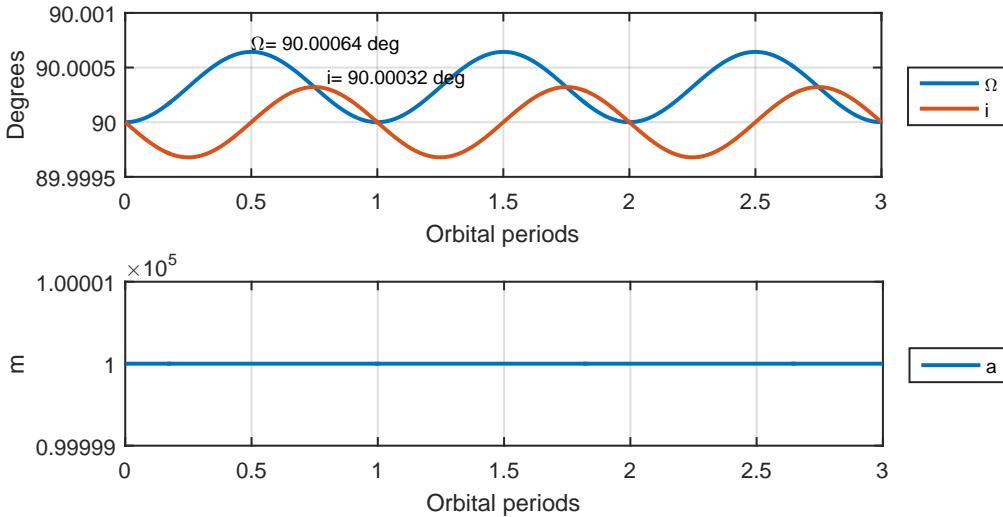


Figure 6.19: Orbital element variations due to SRP force, obtained by simulations.

The SRP acceleration for this configuration is $f = 1.739 \times 10^{-7} \text{ m/s}^2$, the mean motion $n = 5.565 \times 10^{-4} \text{ rad/s}$, and $\alpha_0 = 90^\circ$. This gives the following maximum variations:

$$|\Delta a|_{max} = 0 \quad (6.28)$$

$$|\Delta i|_{max} = 5.62 \times 10^{-6} = 3.22 \times 10^{-4} \text{ deg} \quad (6.29)$$

$$|\Delta \Omega|_{max} = 1.12 \times 10^{-5} = 6.44 \times 10^{-4} \text{ deg} \quad (6.30)$$

The analytical results, Eqs. (6.28) to (6.30), give the same value as the numerical solution, presented in Figure 6.19. This concludes that the SRP force model was implemented correctly.

6.6.4. Rotational motion

The rotational motion of a satellite, left-side of Figure 6.7, is a straight-forward implementation of quaternion kinematics equation and Euler's dynamics equation, see Section 3.3. To check whether it is working properly, a torque-free motion is simulated with the initial angular rate:

$$\boldsymbol{\omega}_0 = \begin{pmatrix} 6.28318 \times 10^{-4} \\ 0 \\ 0 \end{pmatrix} \text{ rad/s} \quad (6.31)$$

which corresponds to a rotation around X-axis with a period of ten thousand seconds.

From Figure 6.20 it is clear that when no torque is applied, angular velocities stay constant and the first and the fourth components of a quaternion oscillate with a period twice larger than that of the rotation of the spacecraft. This is due to the fact that a quaternion is a non-unique attitude representation, meaning that there are two quaternions for the same attitude. This paragraph concludes that the integration of rotational dynamics works well.

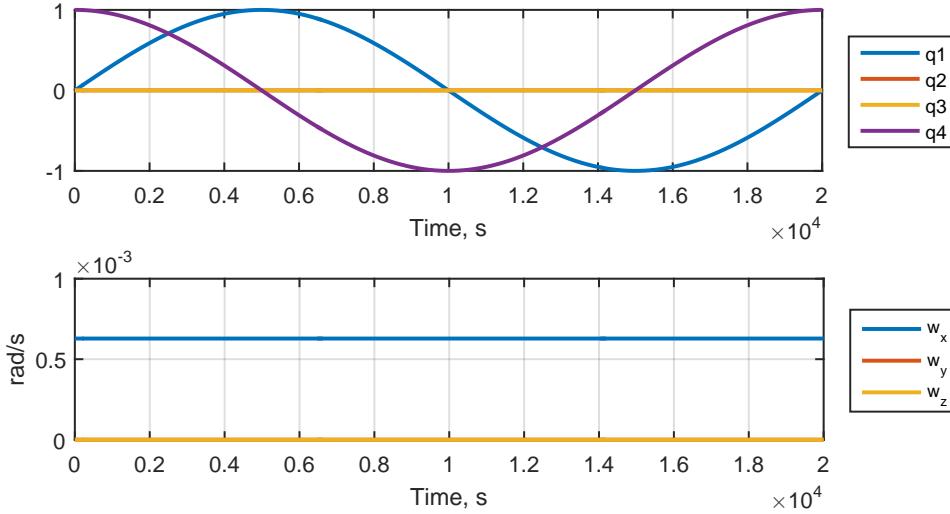


Figure 6.20: Quaternions and angular velocities when no torque applied.

6.6.5. Gravity-Gradient Torque

In general, the Gravity Gradient (GG) torque arises from different gravity field strengths at different parts of the spacecraft body. Books by Wie (2006) or by Markley and Crassidis (2013) give the gravity-gradient torque as:

$$\mathbf{T}_{gg} = 3 \frac{\mu}{R_c^3} \mathbf{a}_3 \times \mathbf{I} \mathbf{a}_3 \quad (6.32)$$

where R_c is the distance from the centre of the attracting body (asteroid), μ is its gravitational parameter, \mathbf{a}_3 is the third component of the Local Vertical Local Horizontal (LVLH) frame, expressed in body frame, and \mathbf{I} is the inertia moment tensor of the SC. However, Eq. (6.32) was derived assuming some simplifications and approximations. First of all, a central gravity field was assumed, meaning that the gravity vector always points towards the centre of the attracting body. Figure 6.21b shows that in the asteroid gravity field it is not the case.

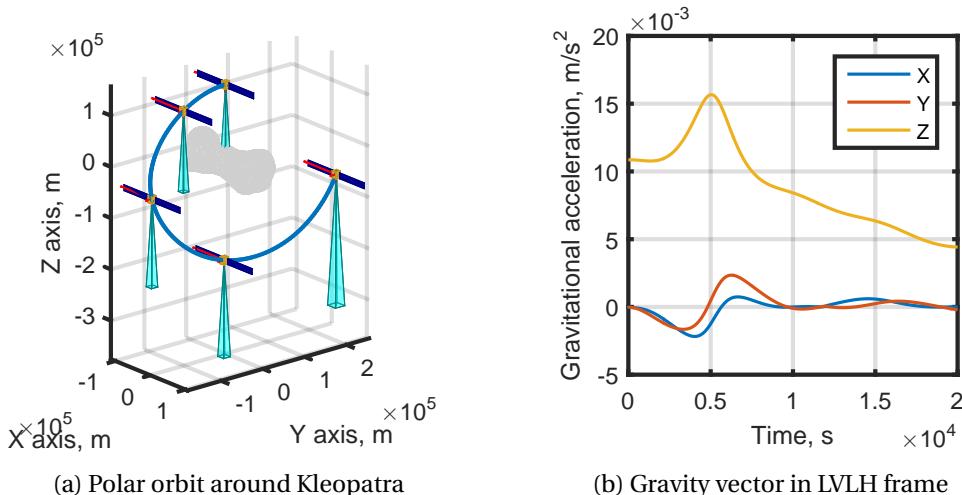


Figure 6.21: Gravity vector in LVLH frame in a polar orbit around Kleopatra.

We recall that in the LVLH frame the Z-axis points nadir, X- to the local north and Y- to the local east. Figures 6.21a and 6.21b present a 'polar' orbit around Kleopatra and the gravitational

acceleration values in LVLH frame respectively. The values show not only the Z-axis component of the gravity vector (which would be in a central gravity field), but also components in X- and Y-axis directions, which, in this 'polar' orbit case, can be up to 20% of the Z component. As a result, Eq. (6.32), by assuming central gravity field, would introduce relatively large GG torque errors in simulations around an asteroid, so a new way to simulate this torque is needed.

Numerical Method If one assumes the SC to be made of N point-masses, the torque is then given by:

$$\mathbf{T}_{gg,B} = \sum_{i=1}^N \mathbf{R}_{B,i} \times m_i \mathbf{g}_{B,i} \quad (6.33)$$

where $\mathbf{R}_{B,i}$ is the distance of the point-mass from the centre of mass of the SC, $\mathbf{g}_{B,i}$ is the gravity field strength at the point-mass (calculated with polyhedron model), and m_i is its mass. Obviously, the more point-masses are taken into account, the more accurate the torque will be. To show how well a discretized spacecraft resembles the real one, the inertia tensor of the SC will be calculated as a sum of individual point masses:

$$I_x = \sum_{i=1}^n m_i (r_{i,y}^2 + r_{i,z}^2) \quad (6.34)$$

$$I_y = \sum_{i=1}^n m_i (r_{i,x}^2 + r_{i,z}^2) \quad (6.35)$$

$$I_z = \sum_{i=1}^n m_i (r_{i,x}^2 + r_{i,y}^2) \quad (6.36)$$

The numerical solutions will be compared to the analytically computed ones, provided in Eq. (6.5). The error is calculated as $\frac{I^* - I}{I} \cdot 100\%$, where I^* is the calculated moment of inertia and I is the real one.

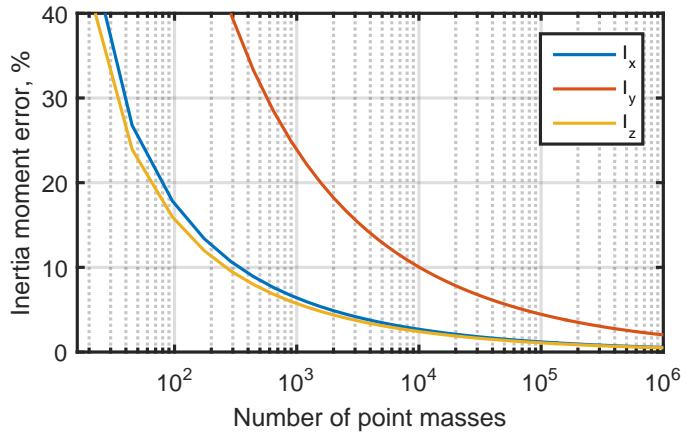


Figure 6.22: Numerical solution of inertia moments as a function of number of masses taken.

Figure 6.22 shows a large discrepancy between numerical and analytical computations. For example, one needs to take more than ten thousand point masses to have an accuracy better than 10%. Due to these discrepancies, the GG torque cannot be calculated numerically and compared

to Eq. (6.32), because, unless an unreasonable number of point masses is used, it cannot represent the actual spacecraft. To compare how well the simplified GG torque in Eq. (6.32) matches the 'real' one, a simplified SC model is made, Figure 6.23. The fictional SC is made of eight point masses, thus Eq. (6.33) represents the real GG torque.

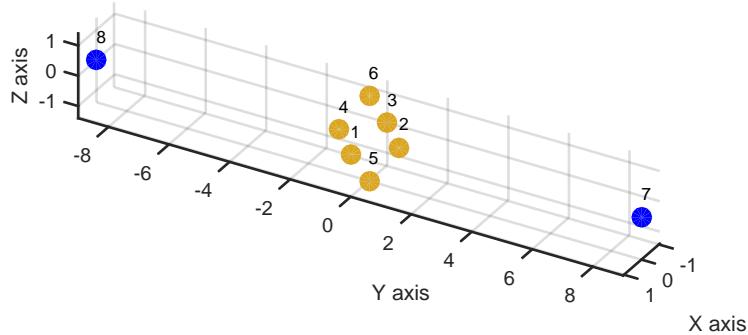


Figure 6.23: Simplified SC model with 8 point masses.

Masses 1 and 3 lie on the X-axis, 2 and 4 on the Y-axis, 5 and 6 on the Z-axis, and 7-8, representing the solar panels, lie on the Y-axis. Masses 1 to 6 are 325 kg, 7,8 are 75 kg, thus the total mass is 2100 kg, the same as the real SC mass. The inertia moments are calculated using Eqs. (6.34) to (6.36) and they are $I_x^* = 14074 \text{ kgm}^2$, $I_y^* = 1991 \text{ kgm}^2$, $I_z^* = 13517 \text{ kgm}^2$. Now, having these properties, the simplified GG torque, Eq. (6.32), and the real one, Eq. (6.33), can be compared. First, this comparison is done for a central gravity field, to see if the two methods match.

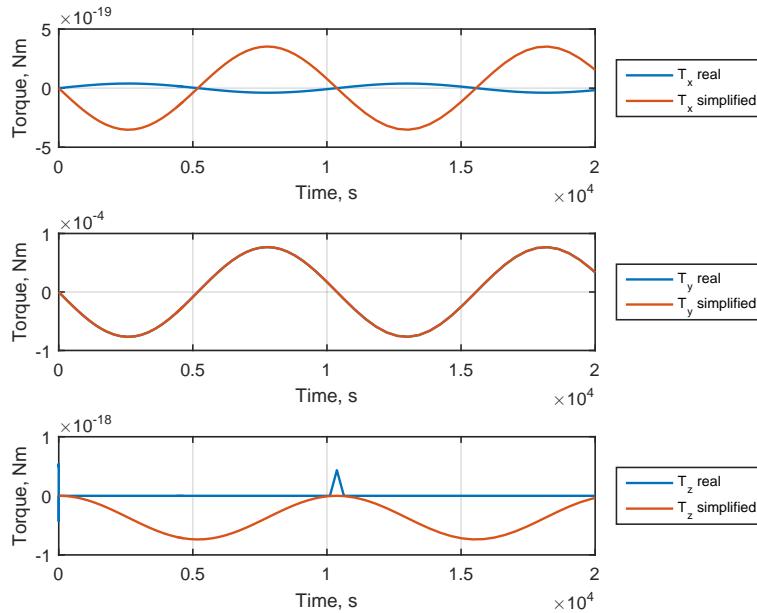


Figure 6.24: Gravity gradient torques compared in a point mass gravity field.

Point mass gravity field A polar orbit in point mass gravity field with constant attitude, Figure 6.21a, is simulated. Since the attitude is inertially fixed, the only torque that should be present is around Y-axis (axis along the solar panel direction). This is proved in Figure 6.24, where the Y-axis torques

match precisely and X- and Z-axis torques are essentially zero; any visible oscillation is due to rounding errors.

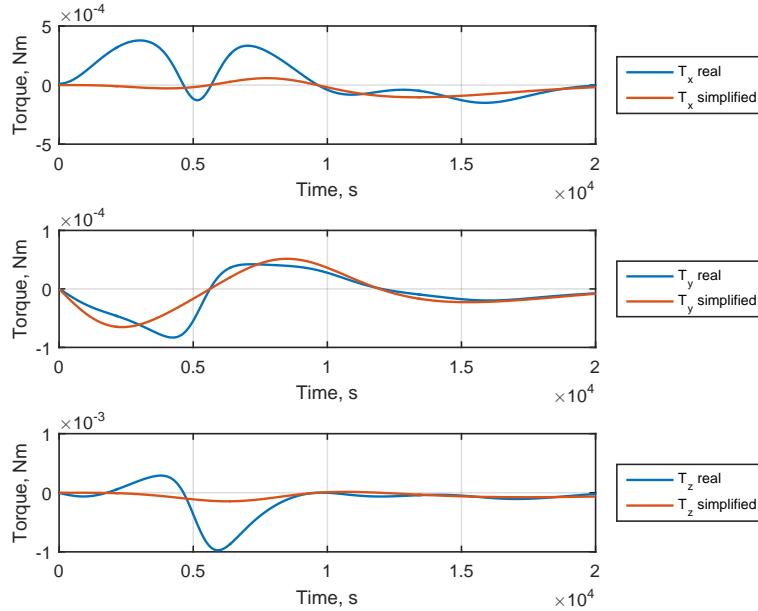


Figure 6.25: Gravity gradient torques compared in an asteroid gravity field.

Asteroid gravity field A polar orbit around Kleopatra is then simulated to compare the two gravity-gradient torque models (the real one and the simplified one). Figure 6.25 clearly shows that the simplified torque model does not capture higher order couplings and/or non central gravity field terms. A similar trend in both models is noticed with, however, significantly different magnitudes. This section concludes that Eq. (6.32) is not suitable for GG torques around asteroids. Hence, the numerical way has to be checked. As it was shown in Figure 6.22, the accuracy of the numerical model strictly depends on the number of masses taken; thus, the torque has to be evaluated with a different number of point-masses as well.

Real spacecraft The spacecraft simulated in this thesis is not made of a few point-masses as in Figure 6.23, therefore, Eq. (6.33) is a numerical approximation. Figure 6.26 gives the comparison of GG torques computed with different number of point-masses. There is an obvious convergence trend seen, when the number of point-masses increases, whereas the simplified torque model is instead inaccurate.

To understand the errors better, it is assumed that the model with 175 point masses is the 'real' one, since it is the most accurate, and the relative errors are computed as $T = \frac{|T_{real} - T(N)|}{T_{real}} \times 100\%$. The results, plotted in Figure 6.27, show remarkable results, which were unexpected. All the relative errors stay constant. The constant is clearly a function of the number of point masses taken, N . As a result, it suggests a different way of calculating GG torque accurately. Since:

$$\frac{T_{real} - T(N)}{T_{real}} = \text{const}(N) \quad (6.37)$$

Then the real gravity gradient torque can be computed as:

$$T_{real} = \frac{T(N)}{1 + \text{const}(N)} \quad (6.38)$$

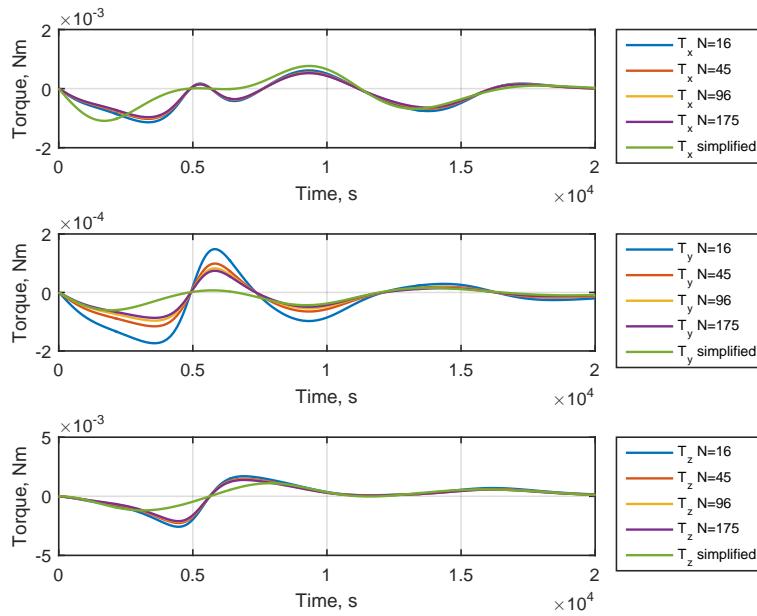


Figure 6.26: Gravity Gradient torques with different number of masses compared.

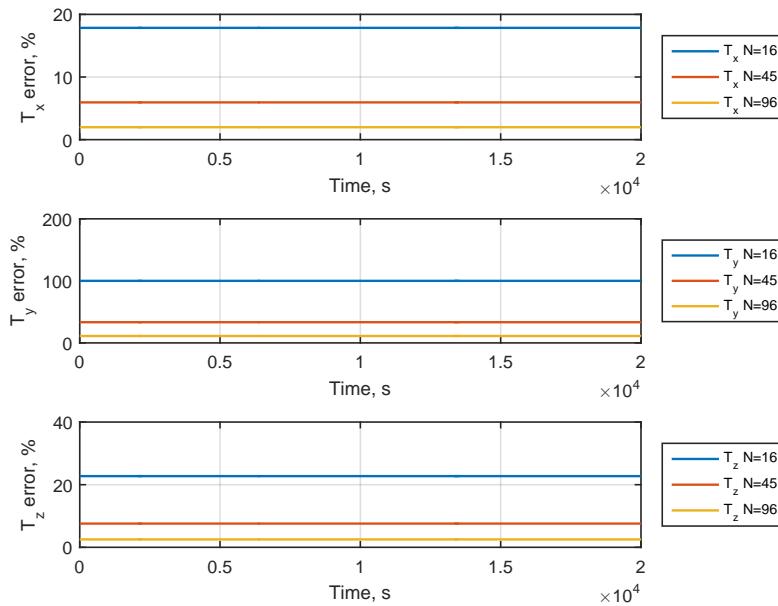


Figure 6.27: Gravity Gradient torques with various number of point-masses compared to N=175 GG torque.

So we need to evaluate the constant for a large number of point-masses taken, then use the least number of point masses, e.g., 16 to simulate the GG torque, which is eventually corrected, using Eq. (6.38). However, this method is based on empirical data and lacks some background theory to explain why the error is constant.

6.7. Validation

In this section the validation process of the simulator is presented. The output of the simulator is compared to the real data from the Rosetta mission. Rosetta is an ESA mission that has explored the 67P/Churyumov–Gerasimenko comet. Although the latter is a comet and not an asteroid, it still represents the dynamics around small bodies. The main difference between an asteroid and a comet is the density, and thus the mass, of the body, for example, the aforementioned comet has the density of 533 kg/m^3 , while asteroids have much higher ones ($2000 - 5000 \text{ kg/m}^3$).

Data from the mission is retrieved using the SPICE toolkit, described in Section 6.2, which allows us to read ephemeris and attitude information from mission files, so-called kernels. The time of simulation is chosen to be between October 15 and 18, 2014, when Rosetta entered a close-observation orbit at 10km, Figure 6.28.

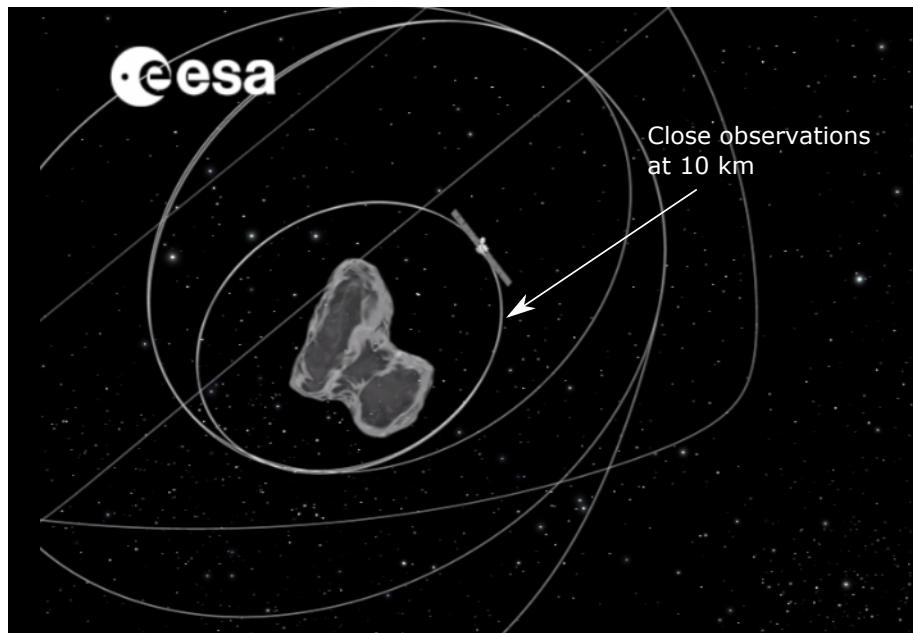


Figure 6.28: Rosetta at 10 km orbit. Image credit: ESA.

This time period is chosen, because no large orbital manoeuvres were performed during it and thus the spacecraft dynamics was mainly governed by the gravitational attraction of the comet. The angular velocity of the spacecraft was approximately the same as the angular motion in the orbit, such that the SC would always point nadir. However, there were small attitude manoeuvres done, which were not included in the SPICE database. As a result the simulated attitude is expected not to match the real one precisely. Moreover, the simulation time was $200,000 \text{ s} \approx 55 \text{ h}$, which is around 80% of the 10 km orbital period (67 h). The simulation time could not be extended longer, because there were some discontinuities detected in the real velocity that most likely imply small orbital corrections, which would corrupt the validation. Position and velocity errors are expressed in percentages, e.g., $\delta R = \frac{\|R_{\text{real}} - R_{\text{sim}}\|}{\|R_{\text{real}}\|} * 100\%$ and attitude errors are presented in the rotation angle $\Phi = 2 \arccos(\mathbf{q}_{\text{real}} \otimes \mathbf{q}_{\text{sim}}^*)$. Also, two simulations were run, one with and another without the SRP force, to see the effect of the inclusion of the SRP force.

Figure 6.29 presents the results of the validation. One could see a large improvement of the position and velocity errors with the inclusion of the SRP force, as an example, the maximum position error is 0.55% or $\approx 55 \text{ m}$ for the case without SRP and only 0.2% or $\approx 20 \text{ m}$ when the SRP is included. The SC attitude errors, although relatively large (up to 20°), but show a steady-state error with a number of discontinuities (control manoeuvres). This means that the simulated angular speed might not match the real one, and all the control manoeuvres make the comparison hard.

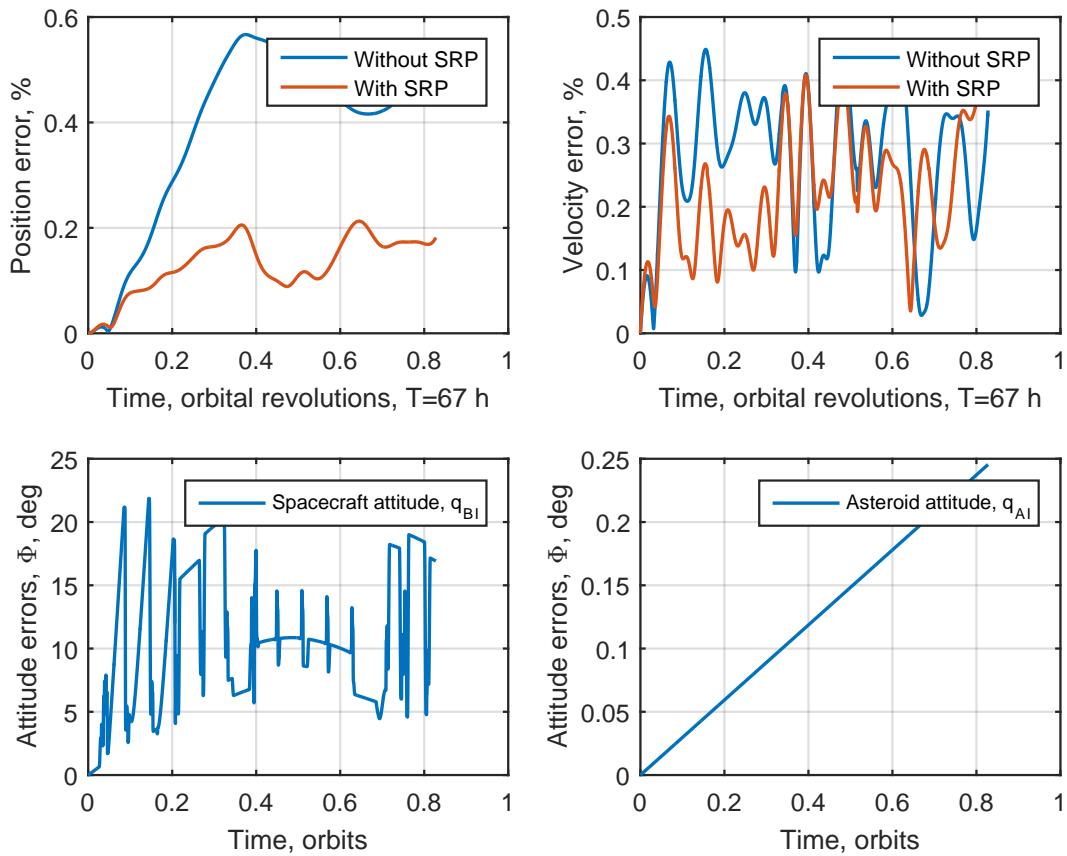


Figure 6.29: Validation results

However, the error is at least not diverging from zero. Oppositely to that, the asteroid attitude error is increasing unboundedly in time. By the end of the simulation there was a 0.25° error, which for the duration time of the simulation (200,000 s) is actually very small, because it would translate into $\approx 1.25 \times 10^{-6}$ deg/s $\approx 2.18 \times 10^{-8}$ rad/s error in asteroid rate. In practice, the asteroid rate is not known with this precision, so the error is acceptable.

These results conclude the validation and all the development of the simulator. The main blocks were verified and the complete simulator was validated, so it is ready to be used.

6.8. Integrator

If we take a look back to Figure 6.2, we see that the three main blocks of the software (dynamics simulator, measurements and estimation) can be simulated separately, since the spacecraft is uncontrollable. For the dynamics we want a precise integration of the equations and since we need to run it once for each orbit scenario, we can choose the most accurate integrator available. In Simulink environment this is the Runge-Kutta 45 (RK45) solver.

The step size is chosen according to the smallest time step needed to generate measurements, which is 0.1 s for a gyroscope. The measurements are discrete and do not involve any dynamics equations, so an integrator does not play a role there. The navigation filters have a propagation of the states, which is usually done with a simple Euler-step method.

6.9. Hardware

The Hardware block is a stand-alone software, meaning that it can generate measurements off-line. The spacecraft is equipped with attitude sensors for inertial state, which are a star tracker and gyroscope, and, sensors for relative navigation, which consist of a navigation camera (NAVCAM) and a laser ranger (LR).

6.9.1. Gyroscope

The mathematical model of a rate gyroscope is given in continuous time, Equation (4.1), however, in practice, the measurements are done in discrete-time. Markley and Crassidis (2013) provide discrete time equations for the measured velocity and drift:

$$\tilde{\boldsymbol{\omega}}_{k+1} = \boldsymbol{\omega}_{k+1} + \frac{1}{2} (\boldsymbol{\mu}_{k+1} + \boldsymbol{\mu}_k) + \left(\frac{\sigma_v^2}{\Delta t} + \frac{1}{12} \sigma_u^2 \Delta t \right)^{1/2} \boldsymbol{\eta}_v \quad (6.39)$$

$$\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k + \sigma_u \Delta t^{1/2} \boldsymbol{\eta}_u \quad (6.40)$$

State-of-the-art gyroscopes, such as the one from Airbus⁷, have noise parameters of $\sigma_v = 5.8 \times 10^{-7}$ rad/s^{1/2} and random walk $\sigma_u = 5.8 \times 10^{-8}$ rad/s^{3/2}, where σ denotes the standard deviation of the noise.

6.9.2. Star Tracker

A quaternion-output star tracker is simulated. In general, a star-tracker gives the true attitude quaternion affected by a small noise:

$$\tilde{\boldsymbol{q}}_{B/I} = \delta \boldsymbol{q}_n^{-1} \otimes \boldsymbol{q}_{B/I} \quad (6.41)$$

where \sim denotes a measured value and $\delta \boldsymbol{q}_n$ is the measurement noise quaternion. Since the accuracy of a star tracker is usually given in arcseconds for roll, pitch and yaw angles, the noise quaternion has to be constructed accordingly. The star tracker is assumed to coincide with the SC reference frame, thus having its boresight pointing along the X-axis in the B frame. Then, according to Markley and Crassidis (2013), for small rotations, a quaternion in Euler angles can be written as:

$$\delta \boldsymbol{q}_n \approx \begin{bmatrix} \phi/2 \\ \theta/2 \\ \psi/2 \\ 1 \end{bmatrix} \quad (6.42)$$

where ϕ, θ, ψ are Euler angles around the X-, Y- and Z-axis respectively. Please note that $\delta \boldsymbol{q}_n$ is not a unit quaternion any more, thus it has to be normalized before being used in Eq. (6.41). The angles ϕ, θ, ψ will be simulated as zero-mean, white noise with known standard deviations $\sigma_\phi, \sigma_\theta$ and σ_ψ respectively. The values are taken from the Rosetta spacecraft star tracker with $\sigma_\phi = 45$ arsec, $\sigma_\theta = 5$ arsec and $\sigma_\psi = 5$ arsec (Buemi et al, 1999), thus with the roll axis being the least accurate.

6.9.3. Navigation Camera

To simulate the output of the navigation camera, the landmark map (LM) has to be generated first, in the way that was explained in Subsection 6.5.2. Then, according to the SC position and attitude relative to the asteroid, the camera Field-Of-View (FOV) and the focal length, f , the landmarks in the FOV have to be projected on the sensor plane. Camera properties are taken from the Rosetta mission. The FOV is assumed to be squared ($5^\circ \times 5^\circ$), the focal length is $f = 152.5$ mm, and the pixel count is 1024×1024 . The implementation sounds straight-forward using the equations provided in Subsection 4.1.3; however, the difficult part is to know which landmarks are seen by the NAVCAM.

⁷<http://www.space-airbusds.com/fr/equipements/astrix-200-fiber-optic-gyro-1j8.html>; date accessed: 03-02-2016

Landmarks in FOV The FOV forms an imaginary pyramid with a square base, Figure 6.30a. If one extends this pyramid up to a point such that the base is inside the asteroid (if possible), then the landmarks within the FOV, will be in this pyramid, Figure 6.30b. We now recall the useful property of polyhedron models that the sum of the solid angle ω_f through all faces indicates, whether a point lies inside or outside the body, Eqs. (3.7) and (3.9) (when $\sum \omega_f = 0$, the point is outside, when $\sum \omega_f = 4\pi$, the point is inside). As a result, we form a simple polyhedron model for the pyramid with five vertices and six faces (in general a rectangular pyramid has five faces, however, the algorithm deals with triangular faces, thus the square base is split into two triangles). Running through all the landmarks and checking the sum, gives those landmarks, which are in the FOV of the camera.

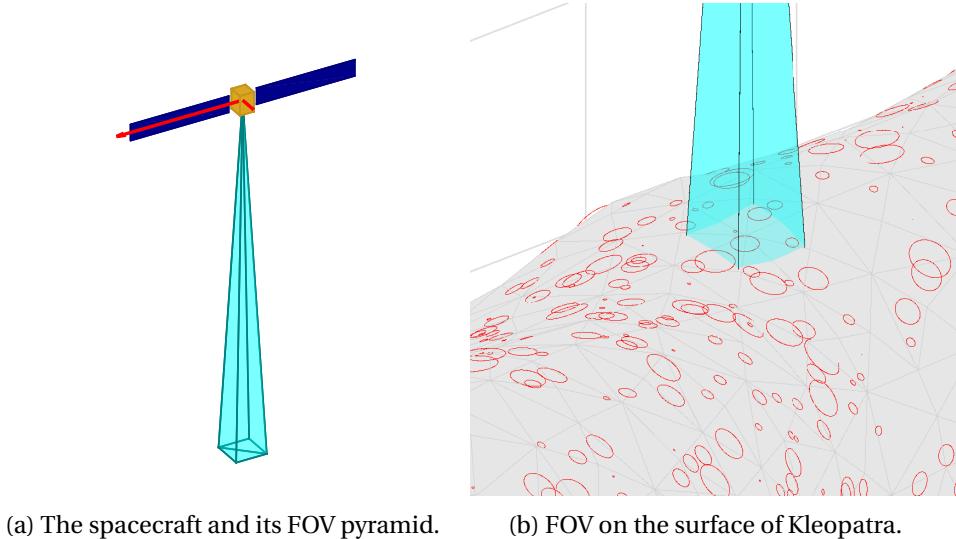


Figure 6.30: Spacecraft FOV with landmarks.

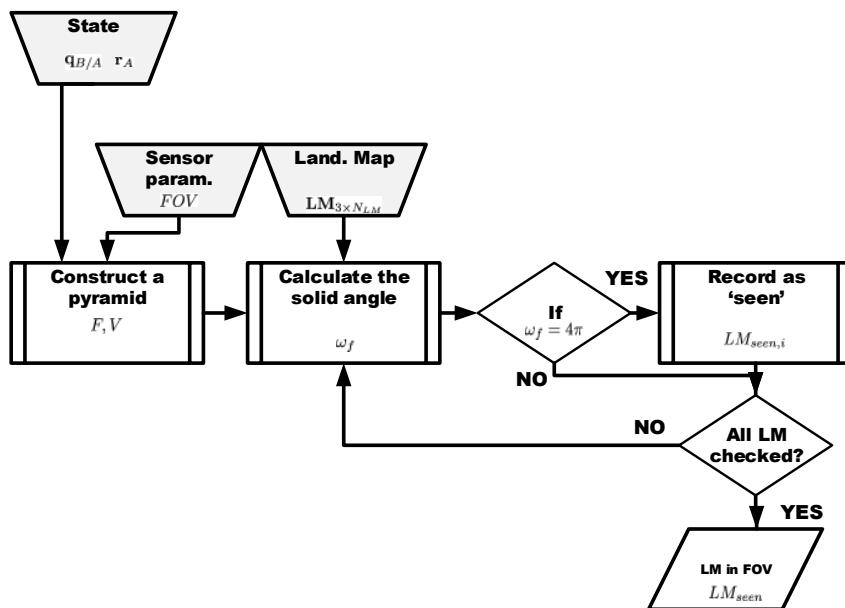


Figure 6.31: Landmarks in FOV algorithm defined.

Furthermore, since every landmark has an associated surface normal to it, it has to be checked, whether it is lit by the Sun or not.

$$\cos \theta = (\mathbf{C}_{A/I} \mathbf{R}_I^{sun}) \cdot \mathbf{n}_{LM}^A \quad (6.43)$$

If $\cos \theta \leq 0$, then it is not seen by the camera, since it is in the shadow. The algorithm for finding landmarks in the FOV is presented in Figure 6.31.

NAVCAM Measurements When the landmarks in FOV are identified, they have to be projected on the sensor plane. Suppose one LM is seen by the camera, Figure 6.32.

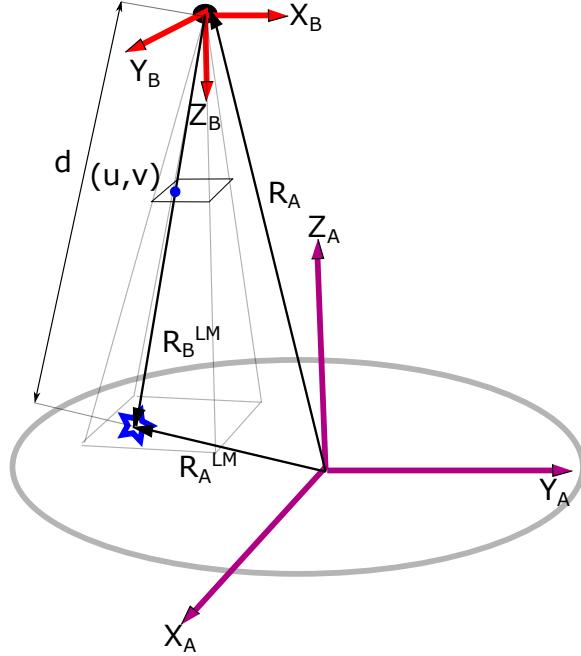


Figure 6.32: A landmark projected on the sensor plane.

The coordinates of the landmarks in the Asteroid frame are assumed to be known, so the following relation holds:

$$\mathbf{R}_B^{LM} = \mathbf{C}_{B/A} (\mathbf{R}_A^{LM} - \mathbf{R}_A) \quad (6.44)$$

where \mathbf{R}_B^{LM} is a vector from the SC to the landmark, expressed in B frame, \mathbf{R}_A^{LM} is the landmark position in the Asteroid frame (assumed to be known), \mathbf{R}_A is the SC position in the Asteroid frame (part of the relative state) and $\mathbf{C}_{B/A}$ is the relative attitude, which is also a part of the relative state. Then, the coordinates of the landmarks, as seen from the B frame, are projected onto the sensor plane:

$$\mathbf{y} = (u \ v)^T = \left(\frac{f}{p} \frac{X_B^{LM}}{Z_B^{LM}} \quad \frac{f}{p} \frac{Y_B^{LM}}{Z_B^{LM}} \right)^T \quad (6.45)$$

where f is the focal length and p is the pixel size, which is equal to $13 \mu\text{m}$ (taken from Rosetta mission), and X_B^{LM} , Y_B^{LM} and Z_B^{LM} are the components of the vector \mathbf{R}_B^{LM} . In the end, the NAVCAM measurements are a function of the relative attitude $\mathbf{q}_{B/A}$ and of the position \mathbf{R}_A . Then, using Eqs. (4.11) and (4.12), the noise is added to the pixels and the unit measurement vector $\tilde{\mathbf{b}}$ is generated. The full software diagram for these measurements is presented in Figure 6.33.

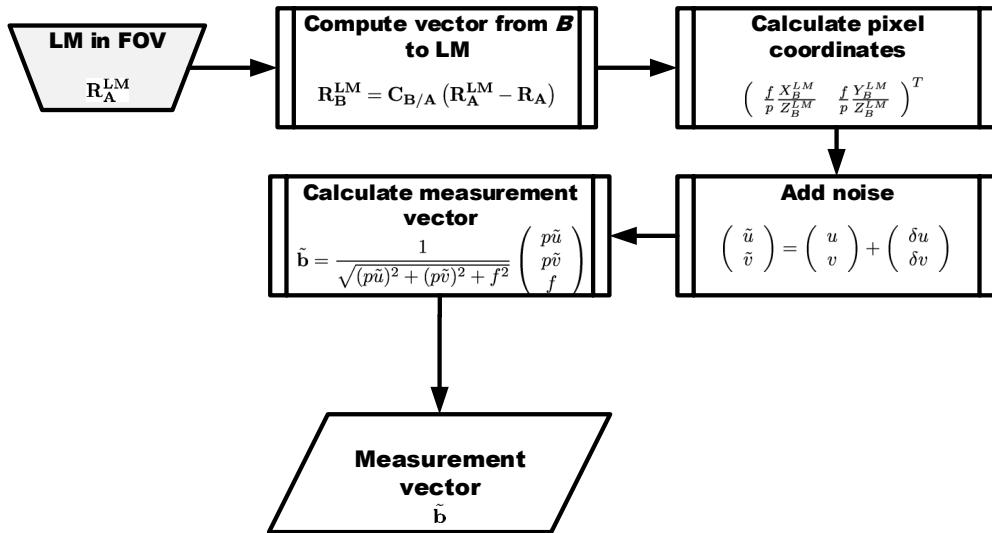


Figure 6.33: Software architecture for NAVCAM measurements.

Verification The model of the NAVCAM needs to be verified to check, whether it sees all the landmarks in the FOV and projects them on the sensor plane correctly. We create a $50 \times 50 \times 50$ m cube and place the spacecraft 230 meters above it, as shown in Figure 6.34a. Then, we place 80 landmarks on the cube's top face, forming a cross, and separated from each other by one meter, Figure 6.34b.

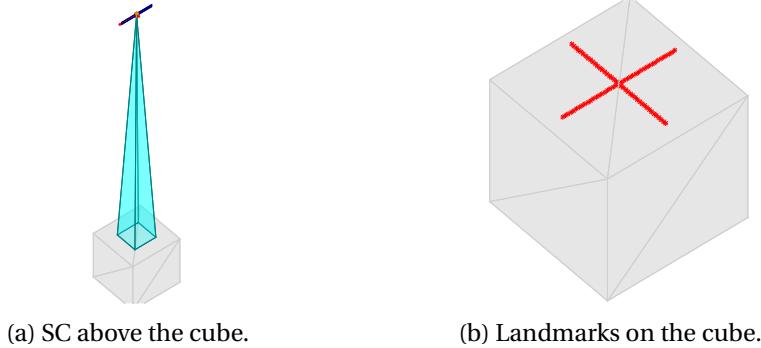


Figure 6.34: Landmarks and FOV.

The FOV of the spacecraft is 5° , and the SC is 230 meters above the surface, so the dimension of the FOV at the surface can be calculated as follows:

$$a = 2R_{SC} \tan \frac{FOV}{2} = 2 \cdot 230 \cdot 0.0437 = 20.084 \text{ m} \quad (6.46)$$

which means that in each axis the NAVCAM should see 20 landmarks, so in total 40 of them. Now we run the algorithm that detects the landmarks in the FOV and projects them on the sensor plane.

Figure 6.35 shows 40 landmarks in total or 20 on each axis projected on the sensor plane. This concludes the NAVCAM part as it was proved that the model works properly.

6.9.4. Laser Ranger

The laser-ranger (LR) measures the distance between the SC and a selected point, which we chose to be a landmark seen by the NAVCAM. In Subsection 4.1.4 it was shown that measurements suffer

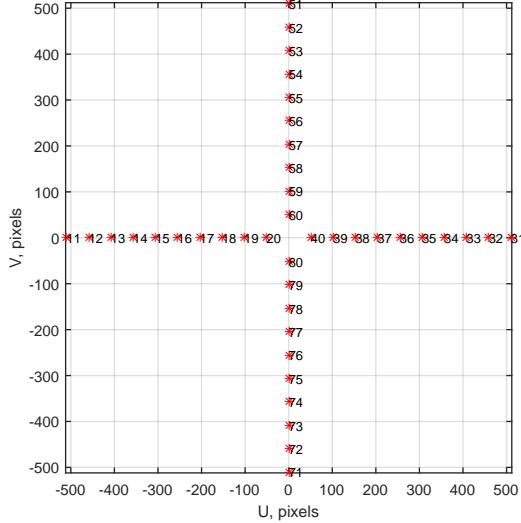


Figure 6.35: Detected landmarks on the sensor plane.

from the pointing error introduced by the gimbal/mirror system, therefore a numerical way to get the measurement data has to be found. For the implementation the very same property of polyhedron models is exploited, which is checking whether a point lies inside or outside of a polyhedron body.

Pointing of the LR Suppose the nominal pointing of the laser is $\mathbf{b}_B^{nom} = (0 \ 0 \ 1)^T$, then this vector is rotated around the X- and Y-axes in the B frame in an active (alibi) manner to point to the landmark:

$$\mathbf{b}_B^{LM} = \mathbf{R}_y(\tilde{\theta})\mathbf{R}_x(\tilde{\phi})\mathbf{b}_B^{nom} \quad (6.47)$$

where $\tilde{\phi}$ and $\tilde{\theta}$ are control angles, extracted from the landmark pixel-coordinates, with introduced pointing errors:

$$\tilde{\phi} = \phi + \delta\phi \quad \tilde{\theta} = \theta + \delta\theta \quad (6.48)$$

where $\delta\phi, \delta\theta$ are zero-mean, white noise representing the pointing error. This can be seen as the actual gimbal/mirror system, rotating the laser beam to the selected landmark. The actual geometry is depicted in Figure 6.36.

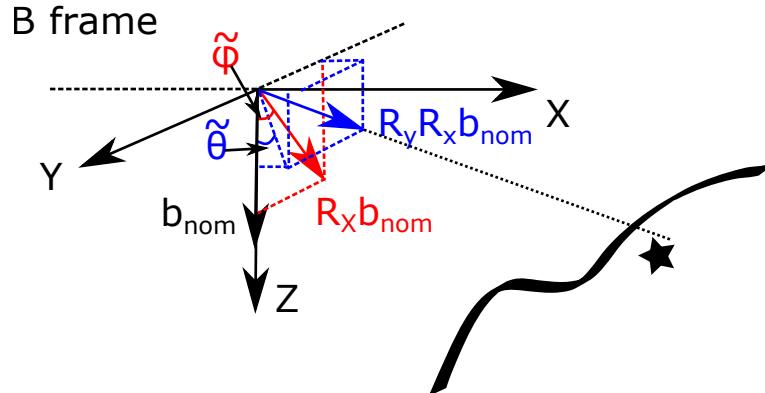


Figure 6.36: Laser pointing geometry.

Getting the distance If the pointing vector \mathbf{b}_A^{LM} (expressed in frame A) is extended in the same direction by \tilde{d} , Figure 6.37, at some point the vector $\mathbf{R}_A + \tilde{d}\mathbf{b}_A^{LM}$ will be inside the asteroid (the sum of the solid angle ω_f will be 4π). Then, in a similar way as the bisection method works, the boundary between inside and outside can be found, which will represent the measured distance \tilde{d} . For example, suppose we extend the vector by \tilde{d}_1 , Figure 6.37. The vector $\mathbf{R}_A + \tilde{d}_1\mathbf{b}_A^{LM}$ is still outside the asteroid, so we extend it again by the same distance, which results in $\tilde{d}_2\mathbf{b}_A^{LM}$. Now the vector $\mathbf{R}_A + \tilde{d}_2\mathbf{b}_A^{LM}$ is inside the asteroid, so we reduce the distance by half of \tilde{d}_1 value, which represents the vector $\tilde{d}_3\mathbf{b}_A^{LM}$. At this point, the vector $\mathbf{R}_A + \tilde{d}_3\mathbf{b}_A^{LM}$ is outside the asteroid, so now we would extend it by a quarter of \tilde{d}_1 value and continue the process until a desired accuracy is reached. The measured distance \tilde{d} will be the value from the last iteration, \tilde{d}_i .

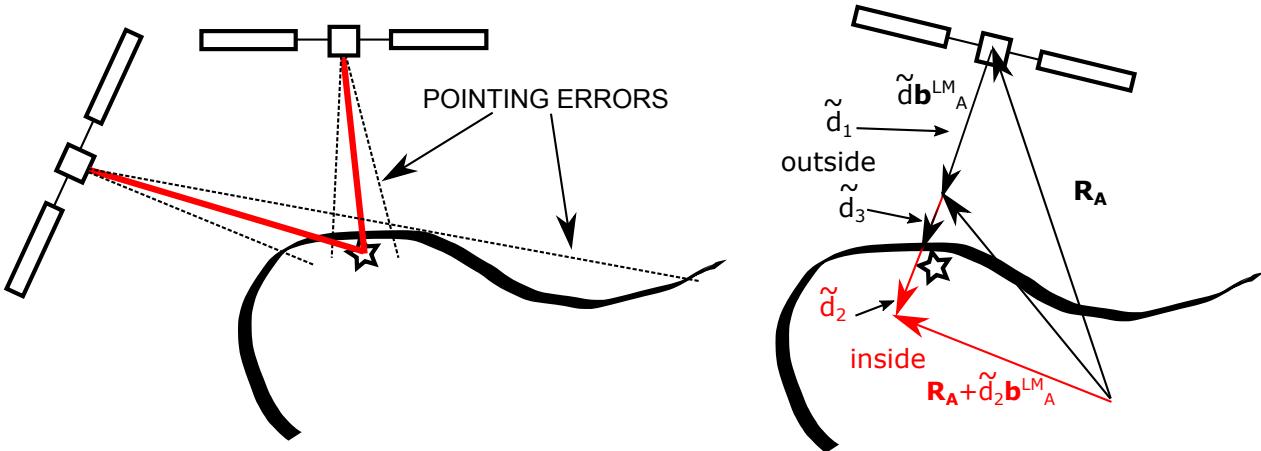


Figure 6.37: Laser ranger errors (left) and measurement simulation process (right)

The proposed method is computationally hard, because at each step a summation over all the faces of the asteroid has to be done, and to get one distance with desired accuracy we might need ten or more of these evaluations. However, the measurements will be recorded only once for each orbital scenario chosen, so it will not need to be run many times. The software diagram for laser ranger model is presented in Figure 6.38.

Verification For the verification we will use a similar set up as in the NAVCAM case. The SC is placed 75 m above the cube, Figure 6.39a. The pointing of the laser is limited within the FOV of the NAVCAM, since it gets commands from the seen landmarks, so we will rotate the laser from -2.5° to 2.5° with 0.5° resolution around the X-axis (angle ϕ).

The true distance can then be calculated as follows:

$$d_{real} = \frac{R_{SC}}{\cos \phi} \quad (6.49)$$

The results are presented as $|d_{real} - d_{meas}|$ in Figure 6.39b. The computations were done with accuracies 0.001, 0.01 and 0.1. The plot shows that the errors for each accuracy value are always smaller than the accuracy itself, which proves that the algorithm works properly.

6.10. Conclusions

This chapter presented the design and implementation of the dynamics simulator and the sensors. A more in-depth verification/investigation was done for the polyhedron gravity field and the gravity-gradient torque. The first one was verified with other papers' data, while for the latter, it was shown that using the ordinary model for the gravity-gradient torque would be too inaccurate, thus a new,

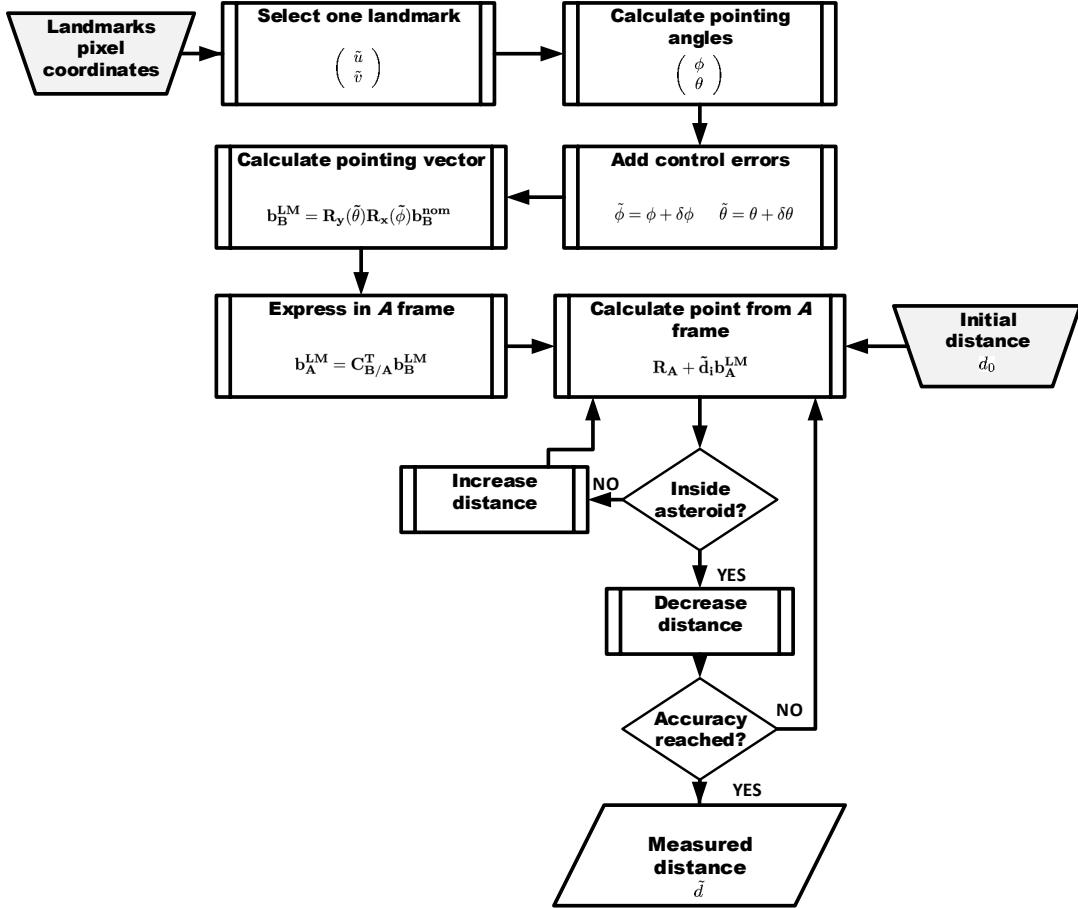


Figure 6.38: Software architecture for laser ranger measurements.

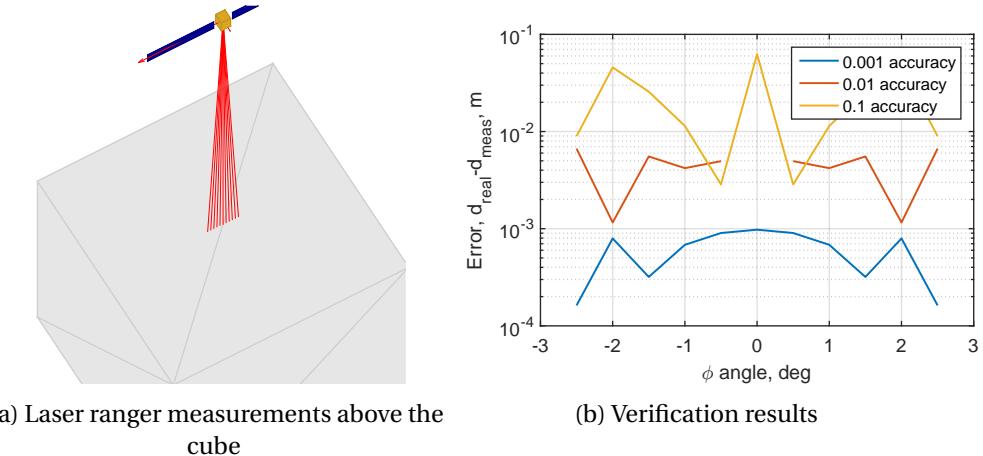


Figure 6.39: Laser ranger verification

numerical model was introduced. The new model gave surprising results that the difference between the torques calculated with different accuracies was actually a constant. This allowed us to reduce the computational time, while keeping the same accuracy. The dynamics simulator was validated by the Rosetta mission data, provided by the SPICE tool.

Furthermore, two novel approaches for simulating a navigation camera and a laser-ranger were

introduced. The difficulty in simulating the NAVCAM was to know which landmarks are seen by the camera, so a new method (creating an imaginary FOV pyramid and checking whether a landmark is inside or outside it) was developed for that. The laser ranger, suffering from a pointing error, was simulated in a similar way a bisection method works. The beam from the spacecraft is extended until it is inside the asteroid, then the length is reduced and the boundary (the surface) distance is found.

7

Navigation Filters

Chapter 7 presents the development of the navigation filters: the Quaternion-Vector Extended Kalman Filter (QVEKF) and the Dual Quaternion Extended Kalman Filter (DQEKF). Section 7.1 starts the chapter with the development of QVEKF. First, the state vector is defined and the dynamics functions are written. Then, a linear perturbation model is derived from those equations, which results in the discrete time state transition matrix Φ . Finally, the measurement functions are defined and their linearised versions are obtained, which results in the measurement matrices H . Section 7.2 follows the same steps for DQEKF filter. In Section 7.3 an improved gravity-field model is introduced, which supposedly will improve the performance of the filters. The chapter ends with a verification of a general extended Kalman filter.

7.1. Quaternion-Vector Extended Kalman Filter

Quaternion-vector Kalman filter is our benchmark for comparison with the dual quaternion counterpart. The relative attitude in this filter is represented by a unit quaternion and the position by a vector (Cartesian coordinates).

7.1.1. QVEKF State Vector

The goal of the navigation system is to estimate the position, attitude, translational and angular velocities of the spacecraft with respect to the asteroid. Since the relative angular rate is a function of inertial and asteroid rates Eq. (3.47), and the inertial rate is measured by the gyroscope, which is susceptible to drift, the common practice is to estimate the gyroscope drift instead of the angular rate itself. However, it is also assumed that the asteroid rate is known with some errors, thus it also has to be estimated. Furthermore, the estimation of the gyro drift requires frequent inertial attitude measurements, e.g., star-tracker, so the inertial attitude is also included in the state vector. As a result, the state vector for the QVEKF filter is defined as follows:

$$\mathbf{X}_{20} = \begin{pmatrix} \mathbf{R}_A \\ \mathbf{V}_A \\ \mathbf{q}_{B/A} \\ \mathbf{q}_{B/I} \\ \boldsymbol{\mu} \\ \boldsymbol{\omega}_{A/I} \end{pmatrix} \quad (7.1)$$

where the state variables are relative position and velocity of the spacecraft in the asteroid reference frame; relative attitude quaternion of frame B with respect to A ; inertial attitude quaternion of frame B with respect to I ; gyroscope drift, expressed in the body reference frame; and the asteroid angular rate, reading from top to bottom, respectively.

7.1.2. Non-linear State Variable Model for \mathbf{X}

In general, the non-linear state variable model is given as follows:

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}) + \mathbf{G}\boldsymbol{\omega} \quad (7.2)$$

where \mathbf{f} is a non-linear function, $\boldsymbol{\omega}$ is a zero-mean, Gaussian process noise, with a known covariance \mathbf{Q} and \mathbf{G} is a noise-mapping matrix, so now we need to find the function \mathbf{f} .

Position and Velocity Let us start with the relative position and velocity. The position derivative is simply expressed as:

$$\dot{\mathbf{R}}_A = \mathbf{V}_A \quad (7.3)$$

For the velocity derivative, or acceleration, we need to decide what gravity-field model to use. The polyhedron model, although being the most accurate, suffers from high computation time needed, thus not suitable for real-time applications. A spherical harmonic model could be selected, but it requires more time to implement, and the spherical harmonics coefficients are not known, so as a starting point the central field is used, because it is easy to implement, while the spherical harmonic model is left as a future work. Moreover, it is a sound choice, because the gravity field is not well known before the arrival to the asteroid, so the central-field model is the initial approximation. The point-mass gravity field will introduce errors, thus a process noise $\boldsymbol{\eta}_g$ must be included. Furthermore, since the acceleration is expressed in the rotating asteroid frame, Coriolis and centrifugal terms have to be added and the dynamics equation reads as follows:

$$\dot{\mathbf{V}}_A = -\frac{GM}{R_A^3} \mathbf{R}_A - 2\boldsymbol{\omega}_{A/I} \times \mathbf{V}_A - \boldsymbol{\omega}_{A/I} \times \boldsymbol{\omega}_{A/I} \times \mathbf{R}_A + \boldsymbol{\eta}_g \quad (7.4)$$

The standard deviation σ_g of the noise $\boldsymbol{\eta}_g$ will be a tuning parameter for the filter.

Inertial Attitude and Gyroscope Drift Let us start with the inertial quaternion, $\mathbf{q}_{B/I}$. The true kinematics equation can be written as follows:

$$\dot{\mathbf{q}}_{B/I} = \frac{1}{2} \boldsymbol{\Omega}(\boldsymbol{\omega}_{B/I}^B) \mathbf{q}_{B/I} = \frac{1}{2} \boldsymbol{\Xi}(\mathbf{q}_{B/I}) \boldsymbol{\omega}_{B/I}^B \quad (7.5)$$

We recall that the measured angular rate is expressed as:

$$\tilde{\boldsymbol{\omega}}_{B/I}^B = \boldsymbol{\omega}_{B/I}^B + \boldsymbol{\mu} + \boldsymbol{\eta}_v \quad (7.6)$$

So the real rate, expressed with the measured rate is:

$$\boldsymbol{\omega}_{B/I}^B = \tilde{\boldsymbol{\omega}}_{B/I}^B - \boldsymbol{\mu} - \boldsymbol{\eta}_v \quad (7.7)$$

The time derivative of the drift is simply:

$$\dot{\boldsymbol{\mu}} = \boldsymbol{\eta}_u \quad (7.8)$$

where $\boldsymbol{\eta}_v$ and $\boldsymbol{\eta}_u$ are the noises of the gyroscope with standard deviations σ_v and σ_u , respectively. Substituting Eq. (7.7) into Eq. (7.5) and after some mathematical manipulation, the following kinematic equation for the inertial quaternion is obtained:

$$\dot{\mathbf{q}}_{B/I} = \frac{1}{2} \boldsymbol{\Omega}(\tilde{\boldsymbol{\omega}}_{B/I}^B - \boldsymbol{\mu}_t) \mathbf{q}_{B/I} - \frac{1}{2} \boldsymbol{\Xi}(\mathbf{q}_{B/I}) \boldsymbol{\eta}_v \quad (7.9)$$

Relative Attitude The relative attitude quaternion can be written in terms of the spacecraft and asteroid inertial attitudes:

$$\mathbf{q}_{B/A} = \mathbf{q}_{B/I} \otimes \mathbf{q}_{A/I}^{-1} \quad (7.10)$$

We recall that $\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{|\mathbf{q}|^2}$, so in case of a unit quaternion $\mathbf{q}^{-1} = \mathbf{q}^*$. Then, differentiating Eq. (7.10) with respect to time yields:

$$\dot{\mathbf{q}}_{B/A} = \dot{\mathbf{q}}_{B/I} \otimes \mathbf{q}_{A/I}^* + \mathbf{q}_{B/I} \otimes \dot{\mathbf{q}}_{A/I}^* = \frac{1}{2} \boldsymbol{\omega}_{B/I}^B \otimes \mathbf{q}_{B/A} + \mathbf{q}_{B/I} \otimes \dot{\mathbf{q}}_{A/I}^* \quad (7.11)$$

The derivative of the inverse quaternion $\dot{\mathbf{q}}_{A/I}^*$ has to be expressed. By definition we have:

$$\mathbf{I}_q = \mathbf{q}_{A/I} \otimes \mathbf{q}_{A/I}^* \quad (7.12)$$

where \mathbf{I}_q is the identity quaternion. Differentiating the above equation with respect to time, the following is obtained:

$$\mathbf{0} = \frac{1}{2} \boldsymbol{\omega}_{A/I}^A \otimes \mathbf{q}_{A/I} \otimes \mathbf{q}_{A/I}^* + \mathbf{q}_{A/I} \otimes \dot{\mathbf{q}}_{A/I}^* \quad (7.13)$$

$$\mathbf{q}_{A/I} \otimes \dot{\mathbf{q}}_{A/I}^* = -\frac{1}{2} \boldsymbol{\omega}_{A/I}^A \quad (7.14)$$

$$\dot{\mathbf{q}}_{A/I}^* = -\frac{1}{2} \mathbf{q}_{A/I}^* \otimes \boldsymbol{\omega}_{A/I}^A \quad (7.15)$$

Now substituting Eq. (7.15) into (7.11) yields:

$$\dot{\mathbf{q}}_{B/A} = \frac{1}{2} \boldsymbol{\omega}_{B/I}^B \otimes \mathbf{q}_{B/A} - \frac{1}{2} \mathbf{q}_{B/I} \otimes \mathbf{q}_{A/I}^* \otimes \boldsymbol{\omega}_{A/I}^A \quad (7.16)$$

$$\dot{\mathbf{q}}_{B/A} = \frac{1}{2} \boldsymbol{\omega}_{B/I}^B \otimes \mathbf{q}_{B/A} - \frac{1}{2} \mathbf{q}_{B/A} \otimes \boldsymbol{\omega}_{A/I}^A \quad (7.17)$$

The relative angular velocity $\boldsymbol{\omega}_{B/A}^B$, according to Eq. (3.47) is:

$$\boldsymbol{\omega}_{B/A}^B = \boldsymbol{\omega}_{B/I}^B - \mathbf{C}_{BA}(\mathbf{q}) \boldsymbol{\omega}_{A/I}^A \quad (7.18)$$

or written as a quaternion transformation:

$$\boldsymbol{\omega}_{B/A}^B = \boldsymbol{\omega}_{B/I}^B - \mathbf{q}_{B/A} \otimes \boldsymbol{\omega}_{A/I}^A \otimes \mathbf{q}_{B/A}^* \quad (7.19)$$

Thus:

$$\boldsymbol{\omega}_{A/I}^A = \mathbf{q}_{B/A}^* \otimes (\boldsymbol{\omega}_{B/I}^B - \boldsymbol{\omega}_{B/A}^B) \otimes \mathbf{q}_{B/A} \quad (7.20)$$

Substituting Eq. (7.20) into Eq. (7.17), the following relation is obtained:

$$\dot{\mathbf{q}}_{B/A} = \frac{1}{2} \boldsymbol{\omega}_{B/I}^B \otimes \mathbf{q}_{B/A} - \frac{1}{2} \mathbf{q}_{B/A} \otimes \mathbf{q}_{B/A}^{-1} \otimes (\boldsymbol{\omega}_{B/I}^B - \boldsymbol{\omega}_{B/A}^B) \otimes \mathbf{q}_{B/A} = \frac{1}{2} \boldsymbol{\omega}_{B/A}^B \otimes \mathbf{q}_{B/A} \quad (7.21)$$

Basically, this result is the usual quaternion kinematics equation, just expressed in the relative state quaternion, $\mathbf{q}_{B/A}$, and relative angular velocity, $\boldsymbol{\omega}_{B/A}^B$. Now we can write the relative kinematics equation in terms of the measured angular velocity:

$$\dot{\mathbf{q}}_{B/A} = \frac{1}{2} \boldsymbol{\Omega} (\boldsymbol{\omega}_{B/I}^B - \boldsymbol{\mu} - \mathbf{C}_{BA}(\mathbf{q}_{B/A}) \boldsymbol{\omega}_{A/I}^A) \mathbf{q}_{B/A} - \frac{1}{2} \boldsymbol{\Xi}(\mathbf{q}_{B/A}) \boldsymbol{\eta}_v \quad (7.22)$$

Asteroid Rate We have already assumed that the asteroid rate is constant, thus:

$$\dot{\omega}_{A/I}^A = \mathbf{0} \quad (7.23)$$

However, for the estimation of the rate to work better, a small process noise is added:

$$\dot{\omega}_{A/I}^A = \boldsymbol{\eta}_\omega \quad (7.24)$$

where $\boldsymbol{\eta}_\omega$ is zero-mean, Gaussian noise with standard deviation σ_ω . This noise is not actually added to the rate, but rather its variance appears in the process covariance matrix \mathbf{Q} , so the filter would 'think' that the noise is present.

Full non-linear model Finally, we obtained the full non-linear state model:

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{X}) + \mathbf{G}(\mathbf{X}) \mathbf{w} \quad (7.25)$$

where

$$\mathbf{f}(\mathbf{X}) = \begin{pmatrix} \mathbf{V}_A \\ -\frac{GM}{R_A^3} \mathbf{R}_A - 2\omega_{A/I} \times \mathbf{V}_A - \omega_{A/I} \times \omega_{A/I} \times \mathbf{R}_A \\ \frac{1}{2} \boldsymbol{\Omega} (\tilde{\omega}_{B/I}^B - \boldsymbol{\mu} - \mathbf{C}_{B/A}(\mathbf{q}_{B/A}) \omega_{A/I}^A) \mathbf{q}_{B/A} \\ \frac{1}{2} \boldsymbol{\Omega} (\tilde{\omega}_{B/I}^B - \boldsymbol{\mu}) \mathbf{q}_{B/I} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \quad (7.26)$$

$$\mathbf{G}(\mathbf{X}) = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{4 \times 3} & -\frac{1}{2} \boldsymbol{\Xi}(\mathbf{q}_{B/A}) & \mathbf{0}_{4 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{4 \times 3} & -\frac{1}{2} \boldsymbol{\Xi}(\mathbf{q}_{B/I}) & \mathbf{0}_{4 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (7.27)$$

$$\mathbf{w} = \begin{pmatrix} \boldsymbol{\eta}_g \\ \boldsymbol{\eta}_v \\ \boldsymbol{\eta}_u \\ \boldsymbol{\eta}_\omega \end{pmatrix} \quad (7.28)$$

7.1.3. The Discrete-time Linear Perturbation Model

Once the non-linear model is obtained, we need to derive the linear perturbation model, as it was discussed in Subsection 4.2.1. It has to be written in the following form:

$$\delta \mathbf{X}_{k+1} = \Phi(\mathbf{X}_k) \delta \mathbf{X}_k + \mathbf{w}_k \quad (7.29)$$

$$\delta \mathbf{z}_{k+1} = \mathbf{H}(\mathbf{X}_{k+1}) \delta \mathbf{X}_{k+1} + \mathbf{v}_{k+1} \quad (7.30)$$

where $\Phi(\mathbf{X}_k)$ is the state transition matrix, derived from $f(\mathbf{X})$, and $\mathbf{H}(\mathbf{X}_{k+1})$ is the gradient matrix of $\mathbf{h}(\mathbf{X}_{k+1})$. In general, we would only need to find the Jacobians for the function \mathbf{f} and \mathbf{h} , Eq. (7.26), however quaternion errors require a special manipulation when defining the errors.

Inertial attitude Let us start with the inertial state again. An error between an estimated and a real quaternion is defined in a multiplicative way and reads as follows:

$$\delta \mathbf{q}_{B/\hat{B}} = \mathbf{q}_{B/I} \otimes \hat{\mathbf{q}}_{\hat{B}/I}^* \quad (7.31)$$

where the $\hat{\cdot}$ symbol denotes an estimated value. The perturbation is then differentiated with respect to time:

$$\delta \dot{\mathbf{q}}_{B/\hat{B}} = \dot{\mathbf{q}}_{B/I} \otimes \hat{\mathbf{q}}_{\hat{B}/I}^* + \mathbf{q}_{B/I} \otimes \dot{\hat{\mathbf{q}}}_{\hat{B}/I}^* \quad (7.32)$$

We recall the derivative of an inverse/conjugate quaternion from Eq. (7.15) and the kinematic equation of a quaternion. Then, the quaternion error kinematics can be written as:

$$\delta \dot{\mathbf{q}}_{B/\hat{B}} = \frac{1}{2} \boldsymbol{\omega}_{B/I}^B \otimes \mathbf{q}_{B/I} \otimes \hat{\mathbf{q}}_{\hat{B}/I}^* - \mathbf{q}_{B/I} \otimes \frac{1}{2} \hat{\mathbf{q}}_{\hat{B}/I}^* \otimes \hat{\boldsymbol{\omega}}_{B/I}^B = \frac{1}{2} \boldsymbol{\omega}_{B/I}^B \otimes \delta \mathbf{q}_{B/\hat{B}} - \frac{1}{2} \delta \mathbf{q}_{B/\hat{B}} \otimes \hat{\boldsymbol{\omega}}_{B/I}^B \quad (7.33)$$

The real angular rate is:

$$\boldsymbol{\omega}_{B/I}^B = \tilde{\boldsymbol{\omega}}_{B/I}^B - \boldsymbol{\mu} - \boldsymbol{\eta}_v \quad (7.34)$$

where $\tilde{\boldsymbol{\omega}}_{B/I}^B$ is the measured rate from the gyro. The estimated rate is defined as:

$$\hat{\boldsymbol{\omega}}_{B/I}^B = \tilde{\boldsymbol{\omega}}_{B/I}^B - \hat{\boldsymbol{\mu}} \quad (7.35)$$

Combining Eqs. (7.34) and (7.35) gives:

$$\boldsymbol{\omega}_{B/I}^B = \hat{\boldsymbol{\omega}}_{B/I}^B + \hat{\boldsymbol{\mu}} - \boldsymbol{\mu} - \boldsymbol{\eta}_v = \hat{\boldsymbol{\omega}}_{B/I}^B + \delta \boldsymbol{\omega} \quad (7.36)$$

where $\delta \boldsymbol{\omega} = \hat{\boldsymbol{\mu}} - \boldsymbol{\mu} - \boldsymbol{\eta}_v$. Substituting the expression above into Eq. (7.33), yields:

$$\begin{aligned} \delta \dot{\mathbf{q}}_{B/\hat{B}} &= \frac{1}{2} (\hat{\boldsymbol{\omega}}_{B/I}^B + \delta \boldsymbol{\omega}) \otimes \delta \mathbf{q}_{B/\hat{B}} - \frac{1}{2} \delta \mathbf{q}_{B/\hat{B}} \otimes \hat{\boldsymbol{\omega}}_{B/I}^B = \\ &= \frac{1}{2} \hat{\boldsymbol{\omega}}_{B/I}^B \otimes \delta \mathbf{q}_{B/\hat{B}} - \frac{1}{2} \delta \mathbf{q}_{B/\hat{B}} \otimes \hat{\boldsymbol{\omega}}_{B/I}^B + \frac{1}{2} \delta \boldsymbol{\omega} \otimes \delta \mathbf{q}_{B/\hat{B}} \end{aligned} \quad (7.37)$$

We recall the two quaternion products and their relation (Markley and Crassidis, 2013):

$$[\mathbf{q} \otimes] = \left[\begin{array}{cc} q_4 \mathbf{I}_3 - [\mathbf{q}_{1:3} \times] & \mathbf{q}_{1:3} \\ -\mathbf{q}_{1:3}^T & q_4 \end{array} \right] \quad (7.38)$$

$$[\mathbf{q} \odot] = \left[\begin{array}{cc} q_4 \mathbf{I}_3 + [\mathbf{q}_{1:3} \times] & \mathbf{q}_{1:3} \\ -\mathbf{q}_{1:3}^T & q_4 \end{array} \right] \quad (7.39)$$

$$\mathbf{q} \otimes \mathbf{q}' = \mathbf{q}' \odot \mathbf{q} \quad (7.40)$$

Then, since the angular rate, when expressed as a quaternion, has the fourth component equal to zero, Eq. (7.37) simplifies to:

$$\begin{aligned} \delta \dot{\mathbf{q}}_{B/\hat{B}} &= \frac{1}{2} \hat{\boldsymbol{\omega}}_{B/I}^B \otimes \delta \mathbf{q}_{B/\hat{B}} - \frac{1}{2} \hat{\boldsymbol{\omega}}_{B/I}^B \odot \delta \mathbf{q}_{B/\hat{B}} + \frac{1}{2} \delta \boldsymbol{\omega} \otimes \delta \mathbf{q}_{B/\hat{B}} \\ &= \frac{1}{2} \left[\begin{array}{cc} -2 [\hat{\boldsymbol{\omega}}_{B/I}^B \times] & 0 \\ 0 & 0 \end{array} \right] \delta \mathbf{q}_{B/\hat{B}} + \frac{1}{2} \delta \boldsymbol{\omega} \otimes \delta \mathbf{q}_{B/\hat{B}} = \end{aligned} \quad (7.41)$$

By definition, a small rotation expressed in a quaternion can be approximated as follows:

$$\delta \mathbf{q} = \left(\begin{array}{c} \frac{1}{2} \delta \boldsymbol{\vartheta} \\ 1 \end{array} \right) \quad (7.42)$$

where $\boldsymbol{\vartheta}$ is the Euler vector associated with the estimation error rotation, so, considering this and neglecting second order effects with respect to the components of the $\boldsymbol{\vartheta}$, Eq. (7.41) becomes:

$$\delta \dot{\mathbf{q}}_{B/\hat{B}} = \begin{bmatrix} -[\hat{\omega}_{B/I}^B \times] & 0 \\ 0 & 0 \end{bmatrix} \delta \mathbf{q}_{B/\hat{B}} + \frac{1}{2} \delta \boldsymbol{\omega} \quad (7.43)$$

Or if we write the error in the rotation vector terms:

$$\delta \dot{\boldsymbol{\vartheta}}_I = [-\hat{\omega}_{B/I}^B \times] \delta \boldsymbol{\vartheta}_I + \delta \boldsymbol{\omega} = [-\hat{\omega}_{B/I}^B \times] \delta \boldsymbol{\vartheta}_I - \delta \boldsymbol{\mu} - \boldsymbol{\eta}_v \quad (7.44)$$

where index I denotes the inertial attitude error and $\delta \boldsymbol{\mu} = \boldsymbol{\mu} - \hat{\boldsymbol{\mu}}$. With Eq. (7.44) we obtained a linear perturbation model for the inertial attitude, expressed in a rotation vector $\boldsymbol{\vartheta}_I$. This also means that the error of the attitude quaternion can be expressed as a three-dimensional vector, instead of four dimensions used to represent the quaternion itself.

Relative Attitude For the relative attitude quaternion, the same process is repeated. The error quaternion, $\delta \mathbf{q}_R$ (R denotes the relative state) is defined as:

$$\delta \mathbf{q}_R = \mathbf{q}_{B/A} \otimes \hat{\mathbf{q}}_{B/A}^* \quad (7.45)$$

Then the time derivative is expressed:

$$\begin{aligned} \delta \dot{\mathbf{q}}_R &= \dot{\mathbf{q}}_{B/A} \otimes \hat{\mathbf{q}}_{B/A}^* + \mathbf{q}_{B/A} \otimes \dot{\hat{\mathbf{q}}}_{B/A}^* = \frac{1}{2} \boldsymbol{\omega}_{B/A}^B \otimes \mathbf{q}_{B/A} \otimes \hat{\mathbf{q}}_{B/A}^* - \frac{1}{2} \mathbf{q}_{B/A} \otimes \hat{\mathbf{q}}_{B/A}^* \otimes \hat{\boldsymbol{\omega}}_{B/A}^B = \\ &= \frac{1}{2} \boldsymbol{\omega}_{B/A}^B \otimes \delta \mathbf{q}_R - \frac{1}{2} \delta \mathbf{q}_R \otimes \hat{\boldsymbol{\omega}}_{B/A}^B \end{aligned} \quad (7.46)$$

The real and estimated relative angular rates are:

$$\boldsymbol{\omega}_{B/A}^B = \boldsymbol{\omega}_{B/I}^B - \mathbf{C}(\mathbf{q}_{B/A}) \boldsymbol{\omega}_{A/I}^A \quad (7.47)$$

$$\hat{\boldsymbol{\omega}}_{B/A}^B = \hat{\boldsymbol{\omega}}_{B/I}^B - \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \hat{\boldsymbol{\omega}}_{A/I}^A \quad (7.48)$$

The asteroid rate estimation error, $\delta \boldsymbol{\omega}_A$, is defined as follows:

$$\delta \boldsymbol{\omega}_A = \boldsymbol{\omega}_{A/I}^A - \hat{\boldsymbol{\omega}}_{A/I}^A \quad (7.49)$$

The true attitude matrix $\mathbf{C}(\mathbf{q}_{B/A})$ can be approximated to first order in $\delta \boldsymbol{\vartheta}_R$ as (Kim et al, 2007):

$$\mathbf{C}(\mathbf{q}_{B/A}) = (\mathbf{I}_3 - [\delta \boldsymbol{\vartheta}_R \times]) \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \quad (7.50)$$

where $\delta \boldsymbol{\vartheta}_R$ is a small rotation vector of the relative attitude. Then, substituting Eqs. (7.36), (7.48), (7.49) and (7.50) into Eq. (7.47), the real relative rate is expressed.

$$\begin{aligned} \boldsymbol{\omega}_{B/A}^B &= \hat{\boldsymbol{\omega}}_{B/I}^B + \delta \boldsymbol{\omega} - (\mathbf{I}_3 - [\delta \boldsymbol{\vartheta}_R \times]) \mathbf{C}(\hat{\mathbf{q}}_{B/A}) (\hat{\boldsymbol{\omega}}_{A/I}^A + \delta \boldsymbol{\omega}_A) = \\ &= \hat{\boldsymbol{\omega}}_{B/I}^B - \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \hat{\boldsymbol{\omega}}_{A/I}^A + \delta \boldsymbol{\omega} + [\delta \boldsymbol{\vartheta}_R \times] \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \hat{\boldsymbol{\omega}}_{A/I}^A - \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \delta \boldsymbol{\omega}_A + [\delta \boldsymbol{\vartheta}_R \times] \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \delta \boldsymbol{\omega}_A \end{aligned} \quad (7.51)$$

The first two terms represent the estimated relative rate $\hat{\boldsymbol{\omega}}_{B/A}^B$, the last term, if second-order terms are neglected, is essentially zero, so after some mathematical treatment, the real rate is expressed as follows:

$$\boldsymbol{\omega}_{B/A}^B = \hat{\boldsymbol{\omega}}_{B/A}^B + \delta \boldsymbol{\omega} - \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \hat{\boldsymbol{\omega}}_{A/I}^A \times \delta \boldsymbol{\vartheta}_R - \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \delta \boldsymbol{\omega}_A \quad (7.52)$$

Substituting this expression to Eq. (7.46):

$$\delta \dot{\mathbf{q}}_R = \frac{1}{2} \hat{\boldsymbol{\omega}}_{B/A}^B \otimes \delta \mathbf{q}_R - \frac{1}{2} \hat{\boldsymbol{\omega}}_{B/A}^B \odot \delta \mathbf{q}_R + \frac{1}{2} \delta \boldsymbol{\omega} \otimes \delta \mathbf{q}_R - \frac{1}{2} (\mathbf{C}(\hat{\mathbf{q}}_{B/A}) \hat{\boldsymbol{\omega}}_{A/I}^A \times \delta \boldsymbol{\vartheta}_R) \otimes \delta \mathbf{q}_R - \quad (7.53)$$

$$\begin{aligned} & -\frac{1}{2}(\mathbf{C}(\hat{\mathbf{q}}_{B/A})\delta\omega_A)\otimes\delta\mathbf{q}_R = \\ & = \begin{bmatrix} -[\hat{\omega}_{B/A}^B \times] & 0 \\ 0 & 0 \end{bmatrix}\delta\mathbf{q}_R + \frac{1}{2}\delta\omega - \frac{1}{2}\mathbf{C}(\hat{\mathbf{q}}_{B/A})\delta\omega_A - \frac{1}{2}\mathbf{C}(\hat{\mathbf{q}}_{B/A})\hat{\omega}_{A/I}^A \times \delta\vartheta_R \end{aligned}$$

We represent the quaternion error with a rotation-vector error and write the error kinematics:

$$\delta\mathbf{q}_R = \begin{pmatrix} \frac{1}{2}\delta\vartheta_R \\ 1 \end{pmatrix} \quad (7.54)$$

$$\delta\dot{\vartheta}_R = -[\hat{\omega}_{B/A}^B \times]\delta\vartheta_R + \delta\omega - \mathbf{C}(\hat{\mathbf{q}}_{B/A})\delta\omega_A - \mathbf{C}(\hat{\mathbf{q}}_{B/A})\hat{\omega}_{A/I}^A \times \delta\vartheta_R \quad (7.55)$$

The first and the last terms can be combined: $-[\hat{\omega}_{B/A}^B \times]\delta\vartheta_R - \mathbf{C}(\hat{\mathbf{q}}_{B/A})\hat{\omega}_{A/I}^A \times \delta\vartheta_R = -[\hat{\omega}_{B/I}^B \times]\delta\vartheta_R$ and after $\delta\omega = -\delta\mu - \eta_v$ is substituted, the following is obtained:

$$\delta\dot{\vartheta}_R = -[\hat{\omega}_{B/I}^B \times]\delta\vartheta_R - \delta\mu - \eta_v - \mathbf{C}(\hat{\mathbf{q}}_{B/A})\delta\omega_A \quad (7.56)$$

The result is interesting itself, because the relative state error is mainly driven by the inertial rate. Moreover, a major difference with $\delta\dot{\vartheta}_I$ is the term with $\delta\omega_A$, which creates observability of the variable $\omega_{A/I}^A$.

Position and Velocity The position-error kinematics is simply:

$$\delta\dot{\mathbf{R}}_A = \delta\mathbf{V}_A \quad (7.57)$$

Velocity error kinematics has to be obtained by differentiating the expression in Eq. (7.26) with respect to the state itself.

$$\frac{df(\mathbf{X})_{4:6}}{d\mathbf{X}} = \left[\begin{array}{ccccc} \frac{df(\mathbf{X})_{4:6}}{d\mathbf{R}_A} & \frac{df(\mathbf{X})_{4:6}}{d\mathbf{V}_A} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ & & \frac{df(\mathbf{X})_{4:6}}{d\omega_{A/I}^A} & & \end{array} \right] \quad (7.58)$$

Here as well the observability of the variable $\omega_{A/I}^A$ is created thanks to the gradient above.

Full State-error Vector Since we have changed the attitude error representation, the error state vector $\delta\mathbf{X}$ changes accordingly and reads as follows:

$$\delta\mathbf{X}_{18} = \begin{pmatrix} \delta\mathbf{R}_A \\ \delta\mathbf{V}_A \\ \delta\vartheta_R \\ \delta\vartheta_I \\ \delta\mu \\ \delta\omega_A \end{pmatrix} \quad (7.59)$$

It has two dimension less than the state vector itself, since the two quaternion errors are three-dimensional.

Full Linear-perturbation Model Combinig Eqs. (7.44), (7.56), (7.57) and (7.59), the linear perturbation model is obtained:

$$\delta\mathbf{X}_{k+1} = \Phi(\mathbf{X}_k)\delta\mathbf{X}_k + \mathbf{G}\mathbf{w}_k \quad (7.60)$$

where Φ is the discrete time version of the Jacobian, \mathbf{F} :

$$\Phi = \exp(\mathbf{F}\Delta t) \approx \mathbf{I}_{18 \times 18} + \mathbf{F}\Delta t \quad (7.61)$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \frac{df(\mathbf{X})_{4:6}}{d\mathbf{R}_A} \Big|_{\mathbf{X}=\hat{\mathbf{X}}} & \frac{df(\mathbf{X})_{4:6}}{d\mathbf{V}_A} \Big|_{\mathbf{X}=\hat{\mathbf{X}}} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \frac{df(\mathbf{X})_{4:6}}{d\omega_{A/I}^A} \Big|_{\mathbf{X}=\hat{\mathbf{X}}} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -[\hat{\omega}_{B/I}^B \times] & \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} & -\mathbf{C}(\hat{\mathbf{q}}_{B/A}) \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -[\hat{\omega}_{B/I}^B \times] & -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (7.62)$$

The noise mapping matrix changes accordingly:

$$\mathbf{G} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (7.63)$$

The explicit formulations of the Jacobians are given in Appendix A.1.

7.1.4. Measurement equations

In this filter three types of measurements are used: star tracker, which outputs an inertial quaternion, navigation camera, which gives the relative attitude and distances to the landmarks in the FOV, and the laser ranger measurements, which gives the distance to a selected landmark. Not all the measurement equations are linear, thus they have to be linearised. Note, that gyroscope measurements are not considered as external measurements in the filter, since they are a part of the process equation.

Star Tracker Essentially, the star tracker outputs a noisy inertial quaternion directly, so the measurement error expressed in a multiplicative way is as follows:

$$\delta \mathbf{z}_{ST} = \tilde{\mathbf{q}}_{B/I} \otimes \hat{\mathbf{q}}_{B/I}^* \quad (7.64)$$

which gives the error quaternion directly or, equivalently, the rotation vector error. As a result, the measurement matrix is simply an identity matrix for the inertial attitude error $\delta \boldsymbol{\vartheta}_I$:

$$\mathbf{H}_{ST} = [\mathbf{0}_{3 \times 9} \quad \mathbf{I}_{3 \times 3} \quad \mathbf{0}_{3 \times 6}] \quad (7.65)$$

Navigation Camera The navigation camera outputs unit vectors to the landmarks, so its measurement to the i^{th} landmark is:

$$\mathbf{z}_{NAVCAM,i} = \mathbf{h}(\mathbf{X}) = \mathbf{C}(\mathbf{q}_{B/A}) \frac{\mathbf{R}_{LM,i} - \mathbf{R}_A}{\|\mathbf{R}_{LM,i} - \mathbf{R}_A\|} \quad (7.66)$$

To adapt Eq. (7.66) for the filter, we need to express the error between the real and predicted measurement. First, we write the attitude matrix as follows (Kim et al, 2007):

$$\mathbf{C}(\mathbf{q}_{B/A}) = (\mathbf{I}_3 - [\delta \boldsymbol{\vartheta}_R \times]) \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \quad (7.67)$$

Substituting Eq. (7.67) into Eq. (7.66), and defining $\delta \mathbf{z} = \mathbf{z} - \hat{\mathbf{z}}$ yields:

$$\delta \mathbf{z} = \mathbf{C}_{B/A}(\hat{\mathbf{q}}_{B/A}) \frac{\mathbf{R}_{LM,i} - \mathbf{R}_A}{\|\mathbf{R}_{LM,i} - \mathbf{R}_A\|} \times \delta \boldsymbol{\vartheta}_R \quad (7.68)$$

The measurement error becomes linear with respect to the relative attitude error, but not to the position. As a result, the measurement matrix for a single landmark is:

$$\mathbf{H}_{NAVCAM} = \left[\frac{d\mathbf{h}(\mathbf{X})}{d\mathbf{R}_A} \Big|_{\mathbf{X}=\hat{\mathbf{X}}} \quad \mathbf{0}_{3 \times 3} \quad \left[\mathbf{C}_{B/A}(\hat{\mathbf{q}}_{B/A}) \frac{\mathbf{R}_{LM,i} - \hat{\mathbf{R}}_A}{\|\mathbf{R}_{LM,i} - \hat{\mathbf{R}}_A\|} \times \right] \quad \mathbf{0}_{3 \times 9} \right] \quad (7.69)$$

Laser Ranger The laser ranger measures distance to a selected landmark:

$$z_{LR} = h(\mathbf{X}) = \|\mathbf{R}_{LM,i} - \mathbf{R}_A\| \quad (7.70)$$

To obtain the measurement matrix, we only need to find the Jacobian of Eq. (7.70).

$$\mathbf{H}_{LR} = \left[\frac{d\mathbf{h}(\mathbf{X})}{d\mathbf{R}_A} \Big|_{\mathbf{X}=\hat{\mathbf{X}}} \quad \mathbf{0}_{1 \times 15} \right] \quad (7.71)$$

The explicit formulations of the measurement matrices are given in Appendix A.1.

7.1.5. Summary

In this section the functions f and h and the matrices Φ , \mathbf{H}_{ST} , \mathbf{H}_{NAVCAM} and \mathbf{H}_{LR} were derived, therefore, the quaternion-vector filter is ready to be used.

7.2. Dual Quaternion Extended Kalman Filter

This section explains the development of dual quaternion extended Kalman filter (DQEKF). The main difference between DQEKF and QVEKF is that the relative pose in the latter is expressed by a quaternion-vector pair and in DQEKF by a dual quaternion. All the other states remain the same. Also, the relative attitude is represented identically in both filters.

7.2.1. Relative Pose

Suppose having reference frames A and B defined relative to the I frame. We recall that the inertial frame has the same origin as the asteroid frame, then we express their pose in dual quaternions as:

$$\check{\mathbf{q}}_{A/I} = \mathbf{q}_{A/I} + \epsilon \mathbf{0} \quad (7.72)$$

$$\check{\mathbf{q}}_{B/I} = \mathbf{q}_{B/I} + \frac{\epsilon}{2} \mathbf{q}_{B/I} \otimes \mathbf{R}_I \quad (7.73)$$

where \mathbf{R}_I is the position vector of the spacecraft, expressed in the inertial frame. The relative pose is then:

$$\check{\mathbf{q}}_{B/A} = \check{\mathbf{q}}_{B/I} \check{\otimes} \check{\mathbf{q}}_{A/I}^* = \left(\mathbf{q}_{B/I} + \frac{\epsilon}{2} \mathbf{q}_{B/I} \otimes \mathbf{R}_I \right) \left(\mathbf{q}_{A/I}^* + \epsilon \mathbf{0} \right) = \mathbf{q}_{B/I} \otimes \mathbf{q}_{A/I}^* + \frac{\epsilon}{2} \mathbf{q}_{B/I} \otimes \mathbf{R}_I \otimes \mathbf{q}_{A/I}^* \quad (7.74)$$

The position quaternion can be expressed in the asteroid frame using the quaternion frame transformation:

$$\mathbf{R}_I = \mathbf{q}_{A/I}^* \otimes \mathbf{R}_A \otimes \mathbf{q}_{A/I} \quad (7.75)$$

Substituting Eq. (7.75) into Eq. (7.74) yields:

$$\check{\mathbf{q}}_{B/A} = \mathbf{q}_{B/A} + \frac{\epsilon}{2} \mathbf{q}_{B/I} \otimes \mathbf{q}_{A/I}^* \otimes \mathbf{R}_A \otimes \mathbf{q}_{A/I} \otimes \mathbf{q}_{A/I}^* = \mathbf{q}_{B/A} + \frac{\epsilon}{2} \mathbf{q}_{B/A} \otimes \mathbf{R}_A \quad (7.76)$$

which is a logical outcome, showing a translation, \mathbf{R}_A , in the asteroid frame, followed by a rotation.

7.2.2. DQEKF State Vector

In DQEKF the relative pose is expressed in a dual-quaternion form. We change the state representation accordingly, and the full state vector reads as follows:

$$\mathbf{X}_{21} = \begin{pmatrix} \check{\mathbf{q}}_{B/A} \\ V_A \\ \mathbf{q}_{B/I} \\ \boldsymbol{\mu} \\ \boldsymbol{\omega}_{A/I}^A \end{pmatrix} \quad (7.77)$$

We can see that the state representation increased by one dimension because the position is now expressed in a four-dimensional quaternion form.

7.2.3. Dual Quaternion Kinematic Equation

By taking Eq.(7.76) and differentiating it with respect to time, the following is obtained:

$$\dot{\tilde{\mathbf{q}}}_{B/A} = \dot{\mathbf{q}}_{B/A} + \frac{\epsilon}{2} (\dot{\mathbf{q}}_{B/A} \otimes \mathbf{R}_A + \mathbf{q}_{B/A} \otimes \dot{\mathbf{R}}_A) = \quad (7.78)$$

$$= \frac{1}{2} \boldsymbol{\omega}_{B/A}^B \otimes \mathbf{q}_{B/A} + \frac{\epsilon}{2} \left(\frac{1}{2} \boldsymbol{\omega}_{B/A}^B \otimes \mathbf{q}_{B/A} \otimes \mathbf{R}_A + \mathbf{q}_{B/A} \otimes \mathbf{V}_A \right)$$

This result, after some rearrangement, can be written in a matrix form:

$$\dot{\tilde{\mathbf{q}}}_{B/A} = \frac{1}{2} \begin{bmatrix} [\boldsymbol{\omega}_{B/A}^B \otimes] & \mathbf{0}_{4 \times 4} \\ [\mathbf{V}_A \odot] & [\boldsymbol{\omega}_{B/A}^B \otimes] \end{bmatrix} \begin{pmatrix} \mathbf{q}_r \\ \mathbf{q}_d \end{pmatrix} \quad (7.79)$$

where $\mathbf{q}_r = \mathbf{q}_{B/A}$ and $\mathbf{q}_d = \frac{\epsilon}{2} \mathbf{q}_{B/A} \otimes \mathbf{R}_A$ are the real and the dual parts of the dual quaternion respectively. Furthermore the term $\mathbf{q}_{B/A} \otimes \mathbf{V}_A$ in Eq. (7.78) can be rewritten as:

$$\mathbf{q}_{B/A} \otimes \mathbf{V}_A = \mathbf{q}_{B/A} \otimes \mathbf{V}_A \otimes \mathbf{q}_{B/A}^* \mathbf{q}_{B/A} = \mathbf{V}_B \otimes \mathbf{q}_{B/A} \quad (7.80)$$

Then, substituting Eq. (7.80) into Eq. (7.78) and writing it in matrix form, yields:

$$\dot{\tilde{\mathbf{q}}}_{B/A} = \frac{1}{2} \begin{bmatrix} [\boldsymbol{\omega}_{B/A}^B \otimes] & \mathbf{0}_{4 \times 4} \\ [\mathbf{V}_B \otimes] & [\boldsymbol{\omega}_{B/A}^B \otimes] \end{bmatrix} \begin{pmatrix} \mathbf{q}_r \\ \mathbf{q}_d \end{pmatrix} \quad (7.81)$$

Here, we introduce a dual velocity, which is defined in the *B* frame as:

$$\check{\boldsymbol{\omega}}_B = \boldsymbol{\omega}_{B/A}^B + \epsilon \mathbf{V}_B \quad (7.82)$$

The kinematics equation is then:

$$\dot{\tilde{\mathbf{q}}}_{B/A} = \frac{1}{2} \check{\boldsymbol{\omega}}_B \check{\otimes} \check{\mathbf{q}}_{B/A} \quad (7.83)$$

where $\check{\otimes}$ is a dual quaternion product and, according to Eq. (7.81), it is defined as follows:

$$[\check{\boldsymbol{\omega}}_{B/I}^B \check{\otimes}] = \begin{bmatrix} [\boldsymbol{\omega}_{B/A}^B \otimes] & \mathbf{0}_{4 \times 4} \\ [\mathbf{V}_B \otimes] & [\boldsymbol{\omega}_{B/A}^B \otimes] \end{bmatrix} \quad (7.84)$$

Full process model Only the relative position representation has changed, so the process function \mathbf{f} changes accordingly:

$$\mathbf{f}(X) = \begin{pmatrix} \frac{1}{2} \boldsymbol{\Omega} (\tilde{\boldsymbol{\omega}}_{B/I}^B - \boldsymbol{\mu} - \mathbf{C}(\mathbf{q}_r) \boldsymbol{\omega}_{A/I}^A) \mathbf{q}_r \\ \frac{1}{2} \boldsymbol{\omega}_{B/A}^B \otimes \mathbf{q}_d + \frac{1}{2} \mathbf{q}_r \otimes \mathbf{V}_A \\ -\frac{GM}{R_A^3} \mathbf{R}_A - 2 \boldsymbol{\omega}_{A/I} \times \mathbf{V}_A - \boldsymbol{\omega}_{A/I} \times \boldsymbol{\omega}_{A/I} \times \mathbf{R}_A \\ \frac{1}{2} \boldsymbol{\Omega} (\tilde{\boldsymbol{\omega}}_{B/I}^B - \boldsymbol{\mu}) \mathbf{q}_{B/I} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \quad (7.85)$$

7.2.4. Linear Perturbation Model for Dual Quaternion

As was done for the QVEKF, a linear perturbation model for dual quaternions is developed first.

Dual quaternion error The dual quaternion error is defined as:

$$\delta \check{\mathbf{q}} = \check{\mathbf{q}}_{B/A} \otimes \hat{\check{\mathbf{q}}}_{B/A}^* \quad (7.86)$$

where $\check{\mathbf{q}}_{B/A}$ is the real dual quaternion and $\hat{\check{\mathbf{q}}}_{B/A}$ is the estimated one. They are expressed as follows:

$$\check{\mathbf{q}}_{B/A} = \mathbf{q}_{B/A} + \frac{\epsilon}{2} \mathbf{q}_{B/A} \otimes \mathbf{R}_A \quad (7.87)$$

$$\hat{\check{\mathbf{q}}}_{B/A} = \hat{\mathbf{q}}_{B/A} + \frac{\epsilon}{2} \hat{\mathbf{q}}_{B/A} \otimes \hat{\mathbf{R}}_A \quad (7.88)$$

Substituting Eqs. (7.87) and (7.88) to (7.86) and using $(\mathbf{q}_1 \otimes \mathbf{q}_2)^* = \mathbf{q}_2^* \otimes \mathbf{q}_1^*$ yields:

$$\begin{aligned} \delta \check{\mathbf{q}} &= \left(\mathbf{q}_{B/A} + \frac{\epsilon}{2} \mathbf{q}_{B/A} \otimes \mathbf{R}_A \right) \left(\hat{\mathbf{q}}_{B/A}^* + \frac{\epsilon}{2} \hat{\mathbf{R}}_A^* \otimes \hat{\mathbf{q}}_{B/A}^* \right) = \\ &= \mathbf{q}_{B/A} \otimes \hat{\mathbf{q}}_{B/A}^* + \frac{\epsilon}{2} \mathbf{q}_{B/A} \otimes \mathbf{R}_A \otimes \hat{\mathbf{q}}_{B/A}^* + \frac{\epsilon}{2} \mathbf{q}_{B/A} \hat{\mathbf{R}}_A^* \otimes \hat{\mathbf{q}}_{B/A}^* \end{aligned} \quad (7.89)$$

The quaternion error $\mathbf{q}_{B/A} \otimes \hat{\mathbf{q}}_{B/A}^*$ is $\delta \mathbf{q}_R$, Eq. (7.45), and $\hat{\mathbf{R}}_A^* = -\hat{\mathbf{R}}_A$, then the error expression is simplified:

$$\begin{aligned} \delta \check{\mathbf{q}} &= \delta \mathbf{q}_R + \frac{\epsilon}{2} \mathbf{q}_{B/A} \otimes (\mathbf{R}_A - \hat{\mathbf{R}}_A) \otimes \hat{\mathbf{q}}_{B/A}^* = \\ &= \delta \mathbf{q}_R + \frac{\epsilon}{2} \mathbf{q}_{B/A} \otimes \hat{\mathbf{q}}_{B/A}^* \otimes \hat{\mathbf{q}}_{B/A} \otimes (\mathbf{R}_A - \hat{\mathbf{R}}_A) \otimes \hat{\mathbf{q}}_{B/A}^* = \\ &= \delta \mathbf{q}_R + \frac{\epsilon}{2} \delta \mathbf{q}_R \otimes \delta \mathbf{R}_B \end{aligned} \quad (7.90)$$

With a first order approximation $\delta \mathbf{q}_R \otimes \delta \mathbf{R}_B \approx \delta \mathbf{R}_B$, so the dual quaternion error becomes:

$$\delta \check{\mathbf{q}} \approx \delta \mathbf{q}_R + \frac{\epsilon}{2} \delta \mathbf{R}_B \quad (7.91)$$

which shows a useful result, since the dual part of the error is the position error itself, however, expressed in frame B .

Error kinematics We differentiate the error, Eq. (7.90), with respect to time:

$$\begin{aligned} \delta \dot{\check{\mathbf{q}}}_{B/A} &= \dot{\check{\mathbf{q}}}_{B/A} \otimes \hat{\check{\mathbf{q}}}_{B/A}^{-1} + \check{\mathbf{q}}_{B/A} \otimes \dot{\hat{\check{\mathbf{q}}}}_{B/A}^{-1} = \frac{1}{2} \check{\boldsymbol{\omega}}_B \check{\otimes} \check{\mathbf{q}}_{B/A} \otimes \hat{\check{\mathbf{q}}}_{B/A}^{-1} - \frac{1}{2} \check{\mathbf{q}}_{B/A} \otimes \hat{\check{\mathbf{q}}}_{B/A}^{-1} \otimes \hat{\boldsymbol{\omega}}_B = \\ &= \frac{1}{2} \check{\boldsymbol{\omega}}_B \otimes \delta \check{\mathbf{q}} - \frac{1}{2} \delta \check{\mathbf{q}} \otimes \hat{\boldsymbol{\omega}}_B = \frac{1}{2} \check{\boldsymbol{\omega}}_B \otimes \delta \check{\mathbf{q}} - \frac{1}{2} \hat{\boldsymbol{\omega}}_B \odot \delta \check{\mathbf{q}} = \frac{1}{2} ([\check{\boldsymbol{\omega}}_B \otimes] - [\hat{\boldsymbol{\omega}}_B \odot]) \delta \check{\mathbf{q}} \end{aligned} \quad (7.92)$$

Now we express the real dual velocity, using Eqs. (7.52)

$$\check{\boldsymbol{\omega}}_B = \hat{\boldsymbol{\omega}}_{B/A}^B + \delta \boldsymbol{\omega} - \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \hat{\boldsymbol{\omega}}_{A/I}^A \times \delta \boldsymbol{\vartheta}_R - \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \delta \boldsymbol{\omega}_A + \epsilon (\hat{\mathbf{V}}_B + \delta \mathbf{V}_B) \quad (7.93)$$

Substituting Eq. (7.93) into (7.92) and using (7.91), the following is obtained:

$$\begin{aligned} \delta \dot{\check{\mathbf{q}}} &= \frac{1}{2} (\hat{\boldsymbol{\omega}}_{B/A}^B + \delta \boldsymbol{\omega} - \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \hat{\boldsymbol{\omega}}_{A/I}^A \times \delta \boldsymbol{\vartheta}_R - \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \delta \boldsymbol{\omega}_A + \epsilon (\hat{\mathbf{V}}_B + \delta \mathbf{V}_B)) \check{\otimes} \left(\delta \mathbf{q}_R + \frac{\epsilon}{2} \delta \mathbf{R}_B \right) + \\ &\quad - \frac{1}{2} \left(\delta \mathbf{q}_R + \frac{\epsilon}{2} \delta \mathbf{R}_B \right) \check{\otimes} (\hat{\boldsymbol{\omega}}_{B/A}^B + \epsilon \hat{\mathbf{V}}_B) = \\ &= \frac{1}{2} (\hat{\boldsymbol{\omega}}_{B/A}^B + \delta \boldsymbol{\omega} - \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \hat{\boldsymbol{\omega}}_{A/I}^A \times \delta \boldsymbol{\vartheta}_R - \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \delta \boldsymbol{\omega}_A) \otimes \delta \mathbf{q}_R + (\epsilon (\hat{\mathbf{V}}_B + \delta \mathbf{V}_B)) \otimes \delta \mathbf{q}_R + \\ &\quad + \frac{\epsilon}{4} (\hat{\boldsymbol{\omega}}_{B/A}^B + \delta \boldsymbol{\omega} - \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \hat{\boldsymbol{\omega}}_{A/I}^A \times \delta \boldsymbol{\vartheta}_R - \mathbf{C}(\hat{\mathbf{q}}_{B/A}) \delta \boldsymbol{\omega}_A) \otimes \delta \mathbf{R}_B + \end{aligned} \quad (7.94)$$

$$-\frac{1}{2}\delta\mathbf{q}_R \otimes \hat{\omega}_{B/A}^B - \frac{\epsilon}{2}\delta\mathbf{q}_R \otimes \hat{V}_B - \frac{\epsilon}{4}\delta\mathbf{R}_B \otimes \hat{\omega}_{B/A}^B$$

The real part of the dual quaternion error derivative is essentially the same as was derived for the relative attitude quaternion, Eq. (7.53), therefore it is not repeated again. The dual part (all terms with ϵ) derivative is as follows:

$$\delta\dot{\mathbf{q}}_d = (\hat{V}_B + \delta V_B) \otimes \delta\mathbf{q}_R + \frac{1}{4}(\hat{\omega}_{B/A}^B + \delta\boldsymbol{\omega} - \mathbf{C}(\hat{\mathbf{q}}_{B/A})\hat{\omega}_{A/I}^A \times \delta\boldsymbol{\vartheta}_R - \mathbf{C}(\hat{\mathbf{q}}_{B/A})\delta\boldsymbol{\omega}_A) \otimes \delta\mathbf{R}_B + \quad (7.95)$$

$$-\frac{1}{2}\delta\mathbf{q}_R \otimes \hat{V}_B - \frac{1}{4}\delta\mathbf{R}_B \otimes \hat{\omega}_{B/A}^B$$

Neglecting the second order terms yields:

$$\begin{aligned} \delta\dot{\mathbf{q}}_d &= \frac{1}{2}\hat{V}_B \otimes \delta\mathbf{q}_R + \frac{1}{2}\delta V_B + \frac{1}{4}\hat{\omega}_{B/A}^B \otimes \delta\mathbf{R}_B - \frac{1}{2}\delta\mathbf{q}_R \otimes \hat{V}_B - \frac{1}{4}\delta\mathbf{R}_B \otimes \hat{\omega}_{B/A}^B = \\ &= -[\hat{V}_B \times] \delta\mathbf{q}_R + \frac{1}{2}\delta V_B - \frac{1}{2}[\hat{\omega}_{B/A}^B \times] \delta\mathbf{R}_B = \\ &= -[\mathbf{C}(\hat{\mathbf{q}}_{B/A})\hat{V}_A \times] \delta\mathbf{q}_R + \frac{1}{2}\mathbf{C}(\hat{\mathbf{q}}_{B/A})\delta V_A - [\hat{\omega}_{B/A}^B \times] \delta\mathbf{q}_d \end{aligned} \quad (7.96)$$

Since the dual part error can be represented by $\delta\mathbf{R}_B$, Eq. (7.91), and the real part by a small rotation vector, $\boldsymbol{\vartheta}_R$, Eq. (7.54), the whole dual quaternion error is written as:

$$\delta\check{\mathbf{q}} = \begin{pmatrix} \frac{1}{2}\delta\boldsymbol{\vartheta}_R \\ 1 \\ \frac{1}{2}\delta\mathbf{R}_B \\ 0 \end{pmatrix} \quad (7.97)$$

Therefore, we reduce the pose vector dimension from eight to six:

$$\delta\check{\mathbf{q}}_{1:6} = \begin{pmatrix} \delta\boldsymbol{\vartheta}_R \\ \delta\mathbf{R}_B \end{pmatrix} \quad (7.98)$$

Then, Eq. (7.96) becomes:

$$\delta\dot{\mathbf{R}}_B = -[\mathbf{C}(\hat{\mathbf{q}}_{B/A})\hat{V}_A \times] \delta\boldsymbol{\vartheta}_R + \mathbf{C}(\hat{\mathbf{q}}_{B/A})\delta V_A - [\hat{\omega}_{B/A}^B \times] \delta\mathbf{R}_B \quad (7.99)$$

As opposed to the QV filter the derivative of $\delta\mathbf{R}_A$ is not δV_A .

Full State-error Vector Since the dual quaternion error is now expressed as a six-dimensional number, the full state-error vector has the same size as in QVEKF case:

$$\delta\mathbf{X}_{18} = \begin{pmatrix} \delta\boldsymbol{\vartheta}_R \\ \delta\mathbf{R}_B \\ \delta V_A \\ \delta\boldsymbol{\vartheta}_I \\ \delta\boldsymbol{\mu} \\ \delta\boldsymbol{\omega}_A \end{pmatrix} \quad (7.100)$$

Full Linear-perturbation Model The linear perturbation model would basically be different only by the position error kinematics, expressed in Eq. (7.99). However, since the error, $\delta \mathbf{R}_B$ is now expressed in B instead of A , as it was in QVEKF, the Jacobian for the velocity has to be modified accordingly. Without loss of generality, we can write:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{R}_B} = \frac{\partial \mathbf{f}}{\partial \mathbf{R}_A} \frac{\partial \mathbf{R}_A}{\partial \mathbf{R}_B} \quad (7.101)$$

where $\frac{\partial \mathbf{R}_A}{\partial \mathbf{R}_B}$ is simply an attitude matrix:

$$\frac{\partial \mathbf{R}_A}{\partial \mathbf{R}_B} = \mathbf{C}_{A/B} = \mathbf{C} (\mathbf{q}_{B/A})^T \quad (7.102)$$

As a result, the velocity Jacobian is expressed as follows:

$$\frac{\partial \mathbf{V}_A}{\partial \mathbf{R}_B} = \frac{\partial \mathbf{V}_A}{\partial \mathbf{R}_A} \mathbf{C} (\mathbf{q}_{B/A})^T \quad (7.103)$$

$$\mathbf{F} = \left[\begin{array}{cccccc} -[\hat{\omega}_{B/I}^B \times] & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} & -\mathbf{C}(\mathbf{q}_{B/A}) \\ -[\mathbf{C}(\mathbf{q}_{B/A}) \hat{\mathbf{V}}_A \times] & -[\hat{\omega}_{B/A}^B \times] & \mathbf{C}(\mathbf{q}_{B/A}) & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \frac{d\mathbf{f}(X)_{7:9}}{d\mathbf{R}_A} \mathbf{C}(\mathbf{q}_{B/A})^T \Big|_{X=\hat{X}} & \frac{d\mathbf{f}(X)_{7:9}}{d\mathbf{V}_A} \Big|_{X=\hat{X}} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \frac{d\mathbf{f}(X)_{7:9}}{d\omega_{A/I}^A} \Big|_{X=\hat{X}} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -[\hat{\omega}_{B/I}^B \times] & -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{array} \right] \quad (7.104)$$

Since the state vector has changed, the noise mapping matrix has changed accordingly:

$$\mathbf{G} = \left[\begin{array}{cccc} \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{array} \right] \quad (7.105)$$

Full expressions of the gradients are provided in Appendix A.1.

7.2.5. Measurement Equations

Since the position error in DQEKF is represented by $\delta \mathbf{R}_B$, and it was $\delta \mathbf{R}_A$ for QVEKF, then the measurement matrices are essentially the same.

Star tracker The star tracker matrix is very alike to the QVEKF one, and it reads as follows:

$$\mathbf{H}_{ST} = [\mathbf{0}_{3 \times 9} \quad \mathbf{I}_{3 \times 3} \quad \mathbf{0}_{3 \times 6}] \quad (7.106)$$

Note that the order of the state variables has changed, so the matrix has changed accordingly, see Eq. (7.100).

Navigation Camera For the Jacobians with respect to the position vector, the same method is applied as in Eq. (7.102):

$$\frac{\partial \mathbf{h}}{\partial \mathbf{R}_B} = \frac{\partial \mathbf{h}}{\partial \mathbf{R}_A} \mathbf{C} (\mathbf{q}_{B/A})^T \quad (7.107)$$

This allows us to use the same measurement matrices as for QVEKF by multiplying them with $\mathbf{C}_{B/A}^T$. The measurement matrix for the navigation camera is then:

$$\mathbf{H}_{NAVCAM} = \left[\begin{array}{c|c} \left[\mathbf{C}_{B/A}(\hat{\mathbf{q}}_{B/A}) \frac{\mathbf{R}_{LM,i} - \hat{\mathbf{R}}_A}{\|\mathbf{R}_{LM,i} - \hat{\mathbf{R}}_A\|} \times \right] & \frac{d\mathbf{h}(X)}{d\mathbf{R}_A} \mathbf{C} (\mathbf{q}_{B/A})^T \quad \mathbf{0}_{3 \times 12} \\ \hline \end{array} \right] \Big|_{X=\hat{X}} \quad (7.108)$$

A different approach for dual quaternions is possible, where the landmarks are treated not as points, but rather as lines connecting two landmarks. This method was presented in the PhD thesis of Goddard (1997), however, for the sake of equal comparison the DQEKF will process the landmarks as points. The dual-line representation is left as a future work.

Laser Ranger Similarly to the NAVCAM measurement matrix, the laser ranger one is obtained:

$$\mathbf{H}_{LR} = \left[\begin{array}{c|c} \mathbf{0}_{1 \times 3} & \frac{d\mathbf{h}(X)}{d\mathbf{R}_A} \mathbf{C} (\mathbf{q}_{B/A})^T \quad \mathbf{0}_{1 \times 12} \\ \hline \end{array} \right] \Big|_{X=\hat{X}} \quad (7.109)$$

7.2.6. Summary

In this section the dual quaternion Kalman filter was developed. It turned out that the dual part error can be represented by a position error expressed in body frame, which simplified the derivation of the transition and measurement matrices. This makes the two filters, QVEKF and DQEKF very alike, and could result in a similar performance. The simulations and results of the two filters are provided in Chapter 8.

7.3. Improved Gravity-field Model

It has already been discussed in Subsection 3.2.1 and shown in Subsection 6.6.2 that gravity fields around asteroids are highly irregular, and the central-field approximation is inaccurate. To demonstrate, how much the performance of the filters can be improved, the polyhedron model is used on-board the spacecraft as well. It is not a realistic case, because it means that the actual gravity field is known perfectly; however, it is rather a demonstrator, what can be achieved with more accurate models. Although, the spherical harmonics model could be used, it requires lengthy derivations of the Jacobians, therefore, it is left as a future work.

The main difference between the aforementioned QVEKF and DQEKF and the improved gravity filters, are the dynamics function \mathbf{f} and the Jacobian matrix \mathbf{F} , concerning the time derivative of velocity.

7.3.1. QVEKF

For the quaternion-vector extended Kalman filter, the dynamics function for velocity reads as follows:

$$\mathbf{f}(X)_{4:6} = \mathbf{g}_A - 2\boldsymbol{\omega}_{A/I} \times \mathbf{V}_A - \boldsymbol{\omega}_{A/I} \times \boldsymbol{\omega}_{A/I} \times \mathbf{R}_A \quad (7.110)$$

where \mathbf{g}_A is the gravity-field value computed with the polyhedron model. The Jacobian with respect to the velocity and the asteroid rate does not change; only the derivative with respect to the position changes. Werner and Scheeres (1997) give the gradient for the gravitational acceleration as follows:

$$\nabla \mathbf{g} = G\rho \sum_{e \in edges} \mathbf{E}_e L_e - G\rho \sum_{f \in faces} \mathbf{F}_f \boldsymbol{\omega}_f \quad (7.111)$$

Then, the full Jacobian for the velocity with respect to the position is obtained:

$$\frac{\partial \mathbf{f}(X)_{4:6}}{\partial \mathbf{R}_A} = G\rho \sum_{e \in edges} \mathbf{E}_e L_e - G\rho \sum_{f \in faces} \mathbf{F}_f \boldsymbol{\omega}_f + \begin{bmatrix} \omega_y^2 + \omega_z^2 & -\omega_x \omega_y & -\omega_x \omega_z \\ -\omega_x \omega_y & \omega_x^2 + \omega_z^2 & -\omega_y \omega_z \\ -\omega_x \omega_z & -\omega_y \omega_z & \omega_x^2 + \omega_y^2 \end{bmatrix} \quad (7.112)$$

7.3.2. DQEKF

For the dual quaternion extended Kalman filter the dynamics function is exactly the same as in Eq. (7.110), and the Jacobian is a modified version of Eq. (7.112):

$$\left(\frac{\partial \mathbf{f}(\mathbf{X})_{4:6}}{\partial \mathbf{R}_A} \right)_{DQ} = \frac{\partial \mathbf{f}(\mathbf{X})_{4:6}}{\partial \mathbf{R}_A} \cdot \mathbf{C}(\mathbf{q}_{B/A})^T \quad (7.113)$$

It can be seen from Eq. (7.112) that the computation of the Jacobian requires the summation over all edges and faces of the polyhedron model, therefore it requires large computational power. As a result, it will not be possible to run extensive simulations with this model, but, as it was mentioned earlier, it is rather a demonstrator for one case.

7.4. Verification

In this section the extended Kalman filter algorithm is verified with the data provided by Zarchan and Musoff (2009). The problem, investigated in the book, is formulated as follows. A high-velocity object is falling on a tracking radar, as shown in Figure 7.1. It has an initial position and an initial velocity of 200,000 feet and 6000 ft/s (please note that imperial units are used in this example). Gravity and drag are the only forces acting on the body, and its position is measured with a 0.1 s sample time. We want to estimate its position and velocity, given the measurement data. First of all, let us define the state vector:

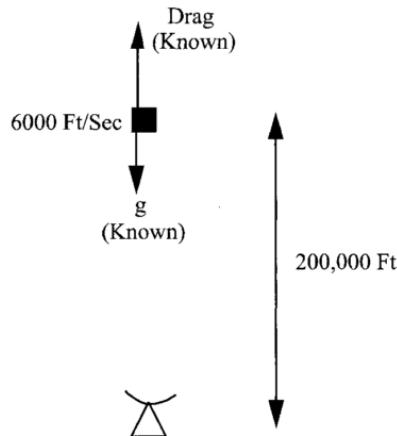


Figure 7.1: Tracking a falling body problem, (Zarchan and Musoff, 2009).

$$\mathbf{X} = \begin{pmatrix} X \\ \dot{X} \end{pmatrix} \quad (7.114)$$

The time derivative of the position is simply the velocity and the derivative of the velocity is as follows:

$$\ddot{X} = \frac{0.0034g \exp(-X/22000) \dot{X}^2}{2\beta} - g \quad (7.115)$$

where β is a ballistic coefficient and g is the gravitational acceleration (assumed to be constant and equal to 32.2ft/s²). So the dynamics function, \mathbf{f} is written as:

$$\mathbf{f} = \begin{pmatrix} \dot{X} \\ \frac{0.0034g \exp(-X/22000) \dot{X}^2}{2\beta} - g \end{pmatrix} \quad (7.116)$$

We can linearize Eq. (7.115) by saying that:

$$\Delta \mathbf{X} = \begin{pmatrix} \Delta \dot{X} \\ \Delta \ddot{X} \end{pmatrix} = \begin{bmatrix} \frac{\partial \dot{X}}{\partial X} & \frac{\partial \dot{X}}{\partial \dot{X}} \\ \frac{\partial \ddot{X}}{\partial X} & \frac{\partial \ddot{X}}{\partial \dot{X}} \end{bmatrix} \begin{pmatrix} \Delta X \\ \Delta \dot{X} \end{pmatrix} + \begin{pmatrix} 0 \\ \eta \end{pmatrix} \quad (7.117)$$

Here a process noise η for the acceleration is introduced, which is useful for tuning the filter. The partial derivatives are expressed below:

$$f_{11} = \frac{\partial \dot{X}}{\partial X} = 0 \quad (7.118)$$

$$f_{12} = \frac{\partial \dot{X}}{\partial \dot{X}} = 1 \quad (7.119)$$

$$f_{21} = \frac{\partial \ddot{X}}{\partial X} = -\frac{0.0034g \exp(-X/22000) \dot{X}^2}{44000\beta} \quad (7.120)$$

$$f_{22} = \frac{\partial \ddot{X}}{\partial \dot{X}} = \frac{0.0034g \exp(-X/22000) \dot{X}}{\beta} \quad (7.121)$$

Then the transition matrix is built, which is evaluated at current estimate:

$$\mathbf{F} = \begin{bmatrix} f_{12} & f_{12} \\ f_{21} & f_{22} \end{bmatrix}_{X=\hat{X}} \quad (7.122)$$

and its discrete time counterpart is approximated:

$$\Phi \approx \mathbf{I} + \mathbf{F}\Delta t \quad (7.123)$$

where Δt is the time step for the filter. The discrete time process noise covariance matrix, given by Zarchan and Musoff (2009), reads as follows:

$$\mathbf{Q} = \sigma_\eta^2 \begin{bmatrix} \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} + f_{22} \frac{\Delta t^3}{3} \\ \frac{\Delta t^2}{2} + f_{22} \frac{\Delta t^3}{3} & \Delta t + f_{22} \Delta t^2 + f_{22}^2 \frac{\Delta t^3}{3} \end{bmatrix} \quad (7.124)$$

The radar measures the position each 0.1 s, so its measurement equation is given as:

$$z = \tilde{X} = [1 \ 0] \mathbf{X} + v \quad (7.125)$$

The measurement matrix is simply:

$$\mathbf{H} = [1 \ 0] \quad (7.126)$$

and the measurement covariance is a scalar:

$$R = \sigma_v^2 \quad (7.127)$$

Now we have everything needed to build the filter, which, after initialization, follows the standard Kalman filter steps. First, the state is propagated with an Euler step method:

$$\mathbf{X}_{k+1/k} = \mathbf{X}_{k/k} + \mathbf{f}(\mathbf{X}_{k/k}) \Delta t \quad (7.128)$$

Then the discrete time transition matrix is evaluated and the error covariance matrix is propagated:

$$\mathbf{P}_{k+1/k} = \Phi \mathbf{P}_{k/k} \Phi^T + \mathbf{Q} \quad (7.129)$$

When the measurement is available, the update stage starts. The steps are as follows:

$$\mathbf{K} = \mathbf{P}_{k+1/k} \mathbf{H}^T (\mathbf{H} \mathbf{P}_{k+1/k} \mathbf{H}^T + R)^{-1} \quad (7.130)$$

$$\hat{\mathbf{X}}_{k+1/k+1} = \hat{\mathbf{X}}_{k+1/k} + \mathbf{K}(z_{k+1} - \mathbf{X}_{k+1/k}) \quad (7.131)$$

$$\mathbf{P}_{k+1/k+1} = (\mathbf{I} - \mathbf{K} \mathbf{H}) \mathbf{P}_{k+1/k} (\mathbf{I} - \mathbf{K} \mathbf{H})^T + \mathbf{K} \mathbf{R} \mathbf{K}^T \quad (7.132)$$

The verification is divided into three parts (attempts), which discuss different ways to mitigate convergence/accuracy problems of an extended Kalman filter.

7.4.1. First Attempt

In the first attempt the process noise is set to zero, $\sigma_\eta = 0$, the measurement noise has standard deviation $\sigma_v = 1000$ ft. The initial position and velocity estimates are 200,025 ft and -6150 ft/s. The initial covariance matrix has diagonal elements of σ_v^2 and 20,000. The filter is run for 30 s with a 0.1 s sample time for the filter and for the measurements.

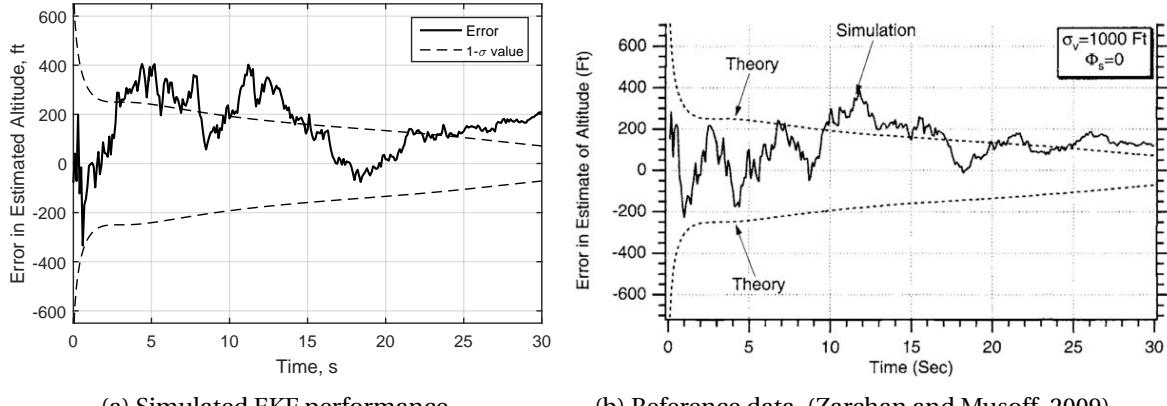


Figure 7.2: Altitude estimates, $\sigma_v = 1000$ ft.

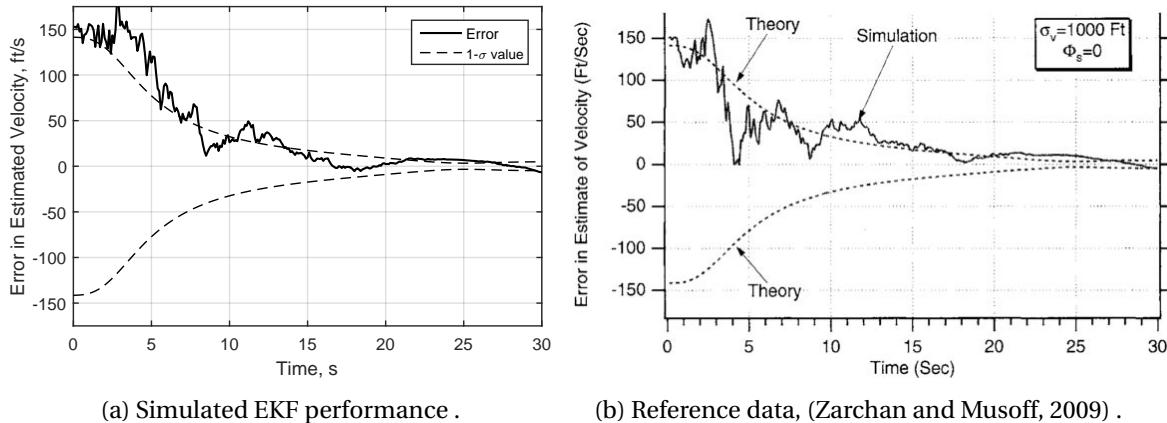
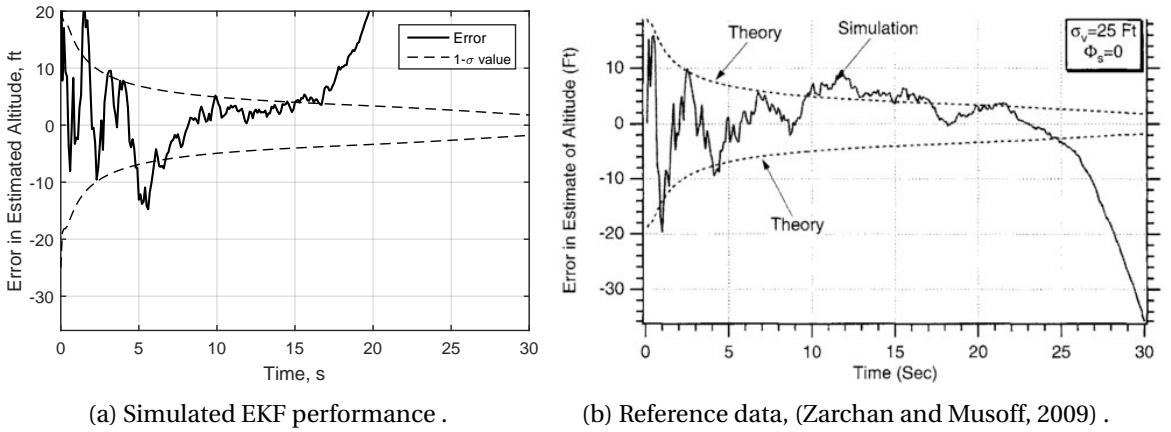
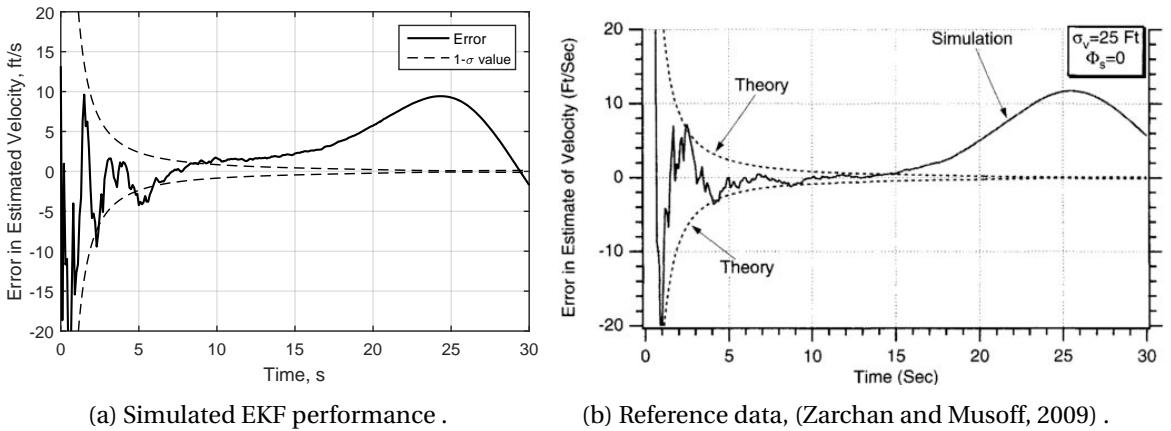


Figure 7.3: Velocity estimates, $\sigma_v = 1000$ ft.

We can see from Figures 7.2 and 7.3 that the simulated EKF and the one provided by Zarchan and Musoff (2009) are very similar. There are some differences, mainly because different random number generators have been used with different seeds, but the trend is the same. From the first glance the performance of the EKF looks correct, the error stays within 1σ boundaries most (1σ corresponds to 68%) of the time. Now, the measurement noise standard deviation is reduced to 25 ft.

Figures 7.4 and 7.5 present the simulation with $\sigma_v = 25$ ft. The differences between the simulated and the reference Kalman filter are larger, but the trend is the same. Now, the altitude and velocity errors start diverging after $\approx 15 - 20$ s, because Kalman gains approach zero and the update stage has less influence.

Figure 7.4: Altitude estimates, $\sigma_v = 25$ ft.Figure 7.5: Velocity estimates, $\sigma_v = 25$ ft.

7.4.2. Second Attempt

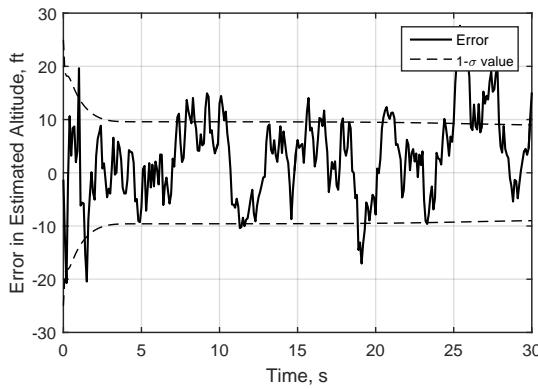
To overcome the problem of Kalman gains going to zero, a small process noise is added, as shown in Eq. (7.118). The variance of the process noise, σ_η^2 is set to 100.

Figures 7.6 and 7.7 present the EKF results for the second attempt with the process noise added. Again, the two filters perform almost identically. Now, with the inclusion of the process noise, the estimates do not diverge any more, because the noise keeps the Kalman gains away from zero. However, this artificial process noise does not represent reality, so we might want a different way to mitigate the problem of divergence.

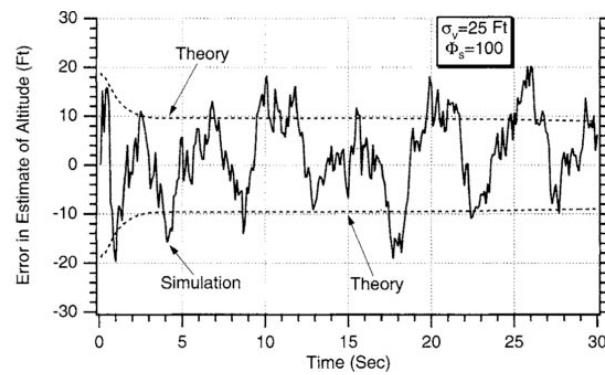
7.4.3. Third Attempt

With the third (and final) attempt, the Kalman filter frequency is changed. Since now, we used 0.1 s for the propagation and update stage, however, we recall using an Euler step method for the integration of the dynamics equations, which is not very accurate, unless small time steps are chosen. So in this third attempt, the time step of the filter is reduced to 0.001 s, but leaving the measurements with 0.1 s sample rate.

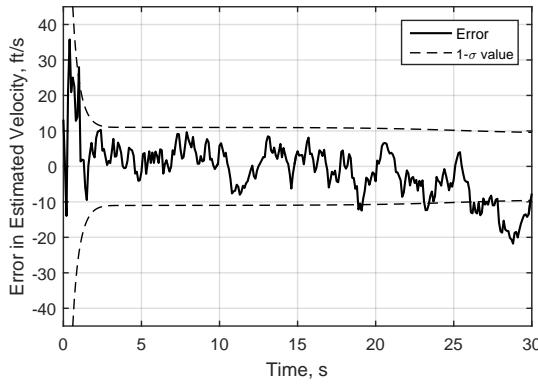
Figures 7.8 and 7.9 presents the EKF results with a reduced time step. The simulated filter proves to match the reference filter again. The divergence of the filter is not present any more, even without the process noise. The filter appears to stay stable and within the boundaries of theoretical standard deviations.



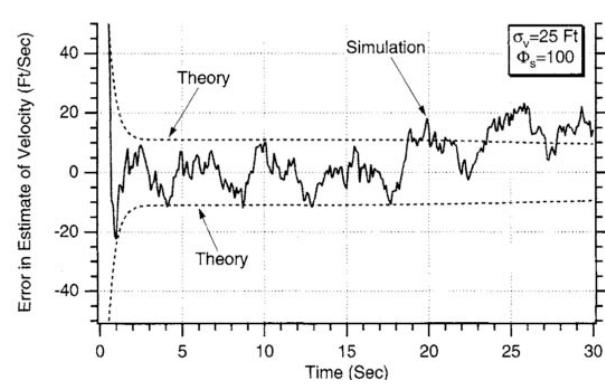
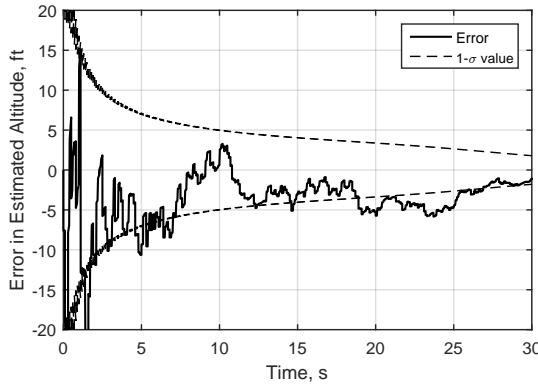
(a) Simulated EKF performance .



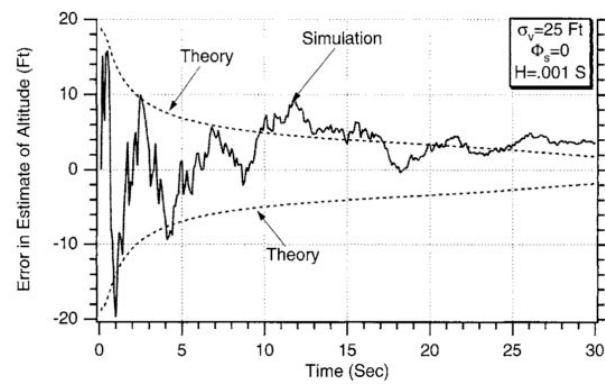
(b) Reference data, (Zarchan and Musoff, 2009) .

Figure 7.6: Altitude estimates, $\sigma_v = 25 \text{ ft}$, $\sigma_\eta^2 = 100$.

(a) Simulated EKF performance .

Figure 7.7: Velocity estimates, $\sigma_v = 25 \text{ ft}$, $\sigma_\eta^2 = 100$.

(a) Simulated EKF performance .



(b) Reference data, (Zarchan and Musoff, 2009) .

Figure 7.8: Altitude estimates, $\sigma_v = 25 \text{ ft}$, $\sigma_\eta^2 = 0$, $\Delta t = 0.001$.

7.4.4. Conclusions

The simulated EKF matched the EKF by Zarchan and Musoff (2009) almost identically in all three attempts, so this proves that the filter algorithm was implemented properly.

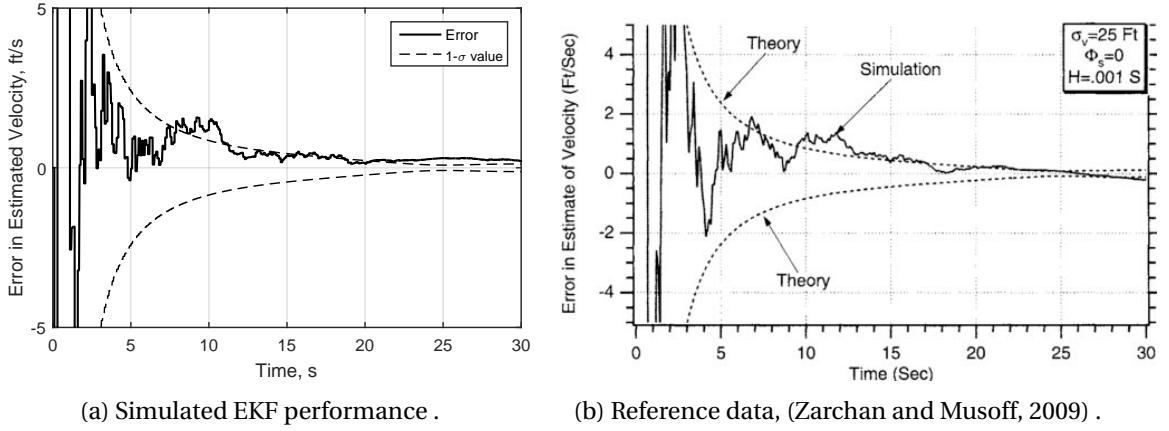
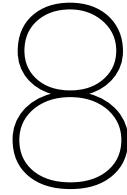


Figure 7.9: Velocity estimates, $\sigma_v = 25$ ft, $\sigma_\eta^2 = 0$, $\Delta t = 0.001$.

7.5. Conclusions

In this chapter the development of a quaternion-vector and a dual quaternion extended Kalman filter was presented. The QVEKF development was following the paper by Kim et al (2007), where the relative attitude filter for two satellites was discussed. First, the general dynamics and measurement equations were written down, afterwards, the linear perturbation model was derived for the dynamics and measurements. The dual quaternion representation appeared to have a very similar error representation, thus the measurement equations from the QVEKF were adapted to the DQEKF. The linear model for dynamics, though, was derived independently. The chapter ended with a verification of a general extended Kalman filter, to prove that the algorithm works properly. The data was compared to the one provided in the book by Zarchan and Musoff (2009).



Simulations

This chapter presents the outcome of all the thesis. The simulations are run for two different asteroids, with four different orbital scenarios and two filters, which give the total of 16 different cases. The four scenarios are introduced in Section 8.1. Section 8.2 continues the chapter with the tuning of the filters discussed. The results for each case are discussed and compared in Section 8.3; and then one of the scenarios is chosen for a Monte-Carlo run in Section 8.4. An additional simulation is done for the improved gravity-field model in Section 8.5 to show what can be achieved with a higher accuracy models. Section 8.6 compares the time consumption of the two filters and the chapter is concluded in Section 8.7.

8.1. Scenarios

In this section mission scenarios are defined that will be used for simulations. We want versatile test cases, which could show different performance of the filters.

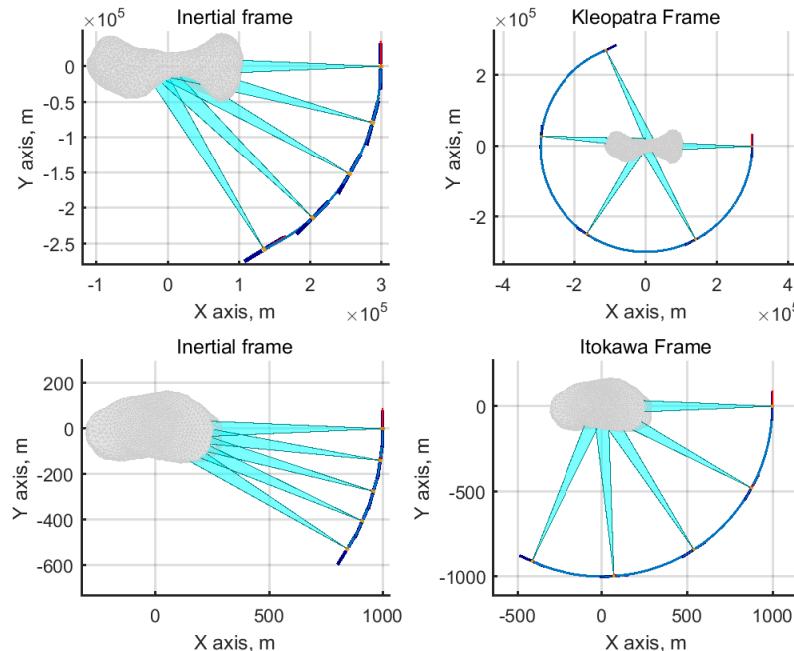


Figure 8.1: **Scenario 1:** retrograde, 'equatorial' orbit, 10,000 s.

Scenario 1 is an 'equatorial' retrograde orbit, Figure 8.1. In the Kleopatra case, the orbit has a 300 km radius, and in the Itokawa case this is 1000 meters. Since it is a retrograde orbit, the position in the asteroid frame changes fast, which gives good observability of various landmarks. The orbit is relatively high, thus the perturbations in the gravity field are accordingly smaller.

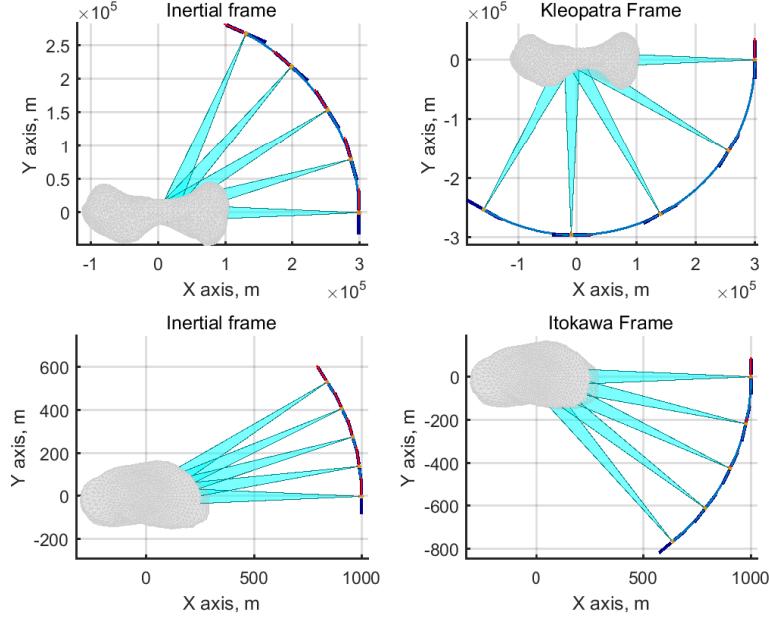


Figure 8.2: **Scenario 2:** prograde, 'equatorial' orbit, 10,000 s.

Scenario 2 is a similar orbit to the Scenario 1. It is an 'equatorial' prograde orbit, Figure 8.2, and all the conditions are the same as in Scenario 1. Since it is a prograde orbit, the relative position changes slower, thus the observability is less favourable than in Scenario 1.

Scenario 3 is an asteroid synchronous orbit, Figure 8.3. The relative position is close to a constant (there are some perturbations) throughout all the simulation. In the Kleopatra case, the spacecraft is situated around 100 km from the center of the asteroid, while in Itokawa it is around 550 m from it. This scenario is interesting due to slow or almost non-existing dynamics of the spacecraft relative to the asteroid. This also decreases the number of visible landmarks.

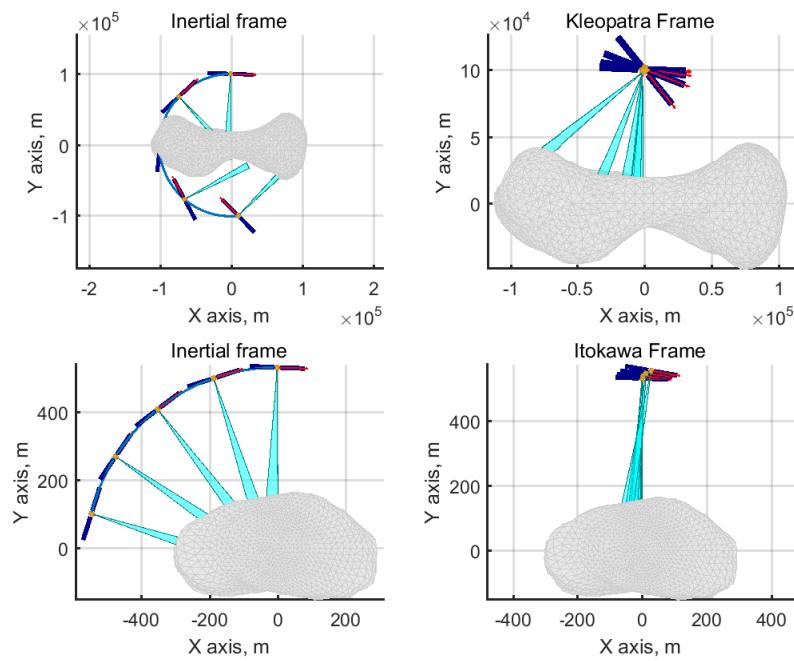
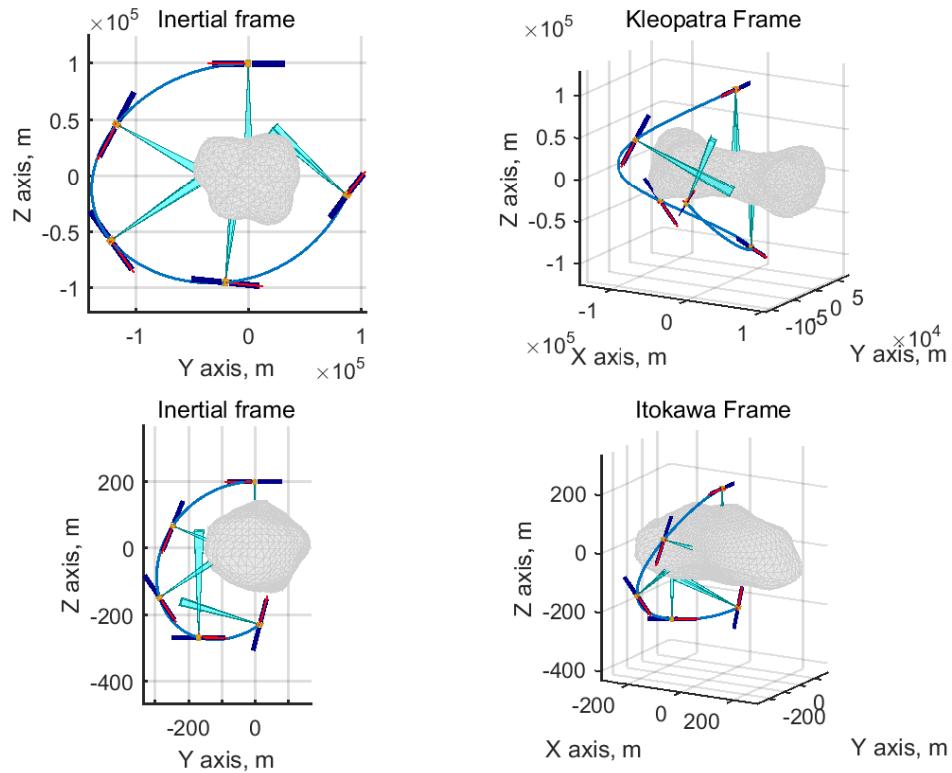
Scenario 4 is a 'polar' orbit with a 100 km and 200 m initial radius for Kleopatra and Itokawa respectively, Figure 8.4. It is the most perturbed scenario, because the distance is relatively small and the dynamics is changing rapidly.

8.2. Filter Tuning

The two Kalman filters are tuned separately for each case and for each asteroid. Mainly, the tuning process involves a proper selection of matrices \mathbf{Q} and $\mathbf{P}_{0/0}$, however the measurement matrices are also tuned to achieve better results. Laser-ranger variance is selected actively during the filter run. The full explanation of this selection process and all the matrices for all the cases are provided in Appendix A.2.

8.3. Results

This section provides the results of the simulations. Since there are four scenarios and two asteroids, we have eight cases in total. These cases are named as KS1, KS2, KS3 and KS4 for Kleopatra (Scenarios 1 to 4), and IS1, IS2, IS3 and IS4 for Itokawa. Each of the case is run on both filters, QVEKF

Figure 8.3: **Scenario 3:** asteroid synchronous orbit, 10,000 s.Figure 8.4: **Scenario 4:** 'polar' orbit, 10,000 s.

and DQEKF for 10,000 s.

To reduce the number of plots needed, scalar errors (norm) of the vectorial variables are presented. For example, the error for position is given as follows:

$$\delta R = \|\mathbf{R} - \hat{\mathbf{R}}\| \quad (8.1)$$

where \mathbf{R} and $\hat{\mathbf{R}}$ are the real and estimated values, respectively. Attitude errors are given as the angle Φ , which is calculated as:

$$\Phi = 2 \arccos \delta q_4 \quad (8.2)$$

where δq_4 is the fourth component of the error quaternion:

$$\delta \mathbf{q} = \mathbf{q} \otimes \hat{\mathbf{q}}^* \quad (8.3)$$

where \mathbf{q} and $\hat{\mathbf{q}}$ are the real and estimated quaternions, respectively. Since the DQEKF does not estimate the position directly, it has to be expressed with:

$$\mathbf{R}_A = 2\mathbf{q}_r^* \otimes \mathbf{q}_d \quad (8.4)$$

Sensor parameters that are common for all the scenarios/cases are presented in Table 8.1.

Table 8.1: Sensor parameters.

Sensor	Value
Star tracker:	
σ_x	45 arcsec = 0.0125 deg
$\sigma_y = \sigma_z$	5 arcsec = 0.0014 deg
Gyroscope:	
σ_v	5.8×10^{-7} rad/s ^{1/2}
σ_u	5.8×10^{-8} rad/s ^{3/2}
μ_0	10^{-5} rad/s (each axis)
NAVCAM:	
Camera pixel count	1024×1024
FOV	$5^\circ \times 5^\circ$
Image processing accuracy	0.1 pixel
Focal length	152.5 mm
LR:	
Lidar pointing error	0.01° (each axis)
Sampling times:	
Gyroscope (Filter)	0.1 s
Star tracker	1 s
NAVCAM/LIDAR	10s

Furthermore, the results will also have the 3σ values, where σ is a standard deviation of the error, and is obtained from the diagonal elements of the error covariance matrix (the square root of them). If a filter is tuned properly, this standard deviation would match the real one of the error. If it is not, the Kalman filter will 'think' that the estimates have a higher/lower variance than the actual one is. This is later checked with a Monte-Carlo simulation.

8.3.1. Scenario 1, Kleopatra

Scenario 1 of Kleopatra (KS1) is an equatorial retrograde orbit around Kleopatra. It is a high altitude orbit, around 300 km distance from the centre of the asteroid. The filter is initialized with the values, provided in Table 8.2 and the tuned covariance matrices are presented in Appendix A.2.

Table 8.2: KS1 initial conditions.

	Real	Estimated
Position and velocity:		
\mathbf{R}_A (km)	$(300 \ 0 \ 0)^T$	$(301 \ 1 \ 1)^T$
\mathbf{V}_A (m/s)	$(0 \ -129.36 \ 0)^T$	$(1 \ -128.36 \ 1)^T$
Attitude:		
$\mathbf{q}_{B/I}$	$(0 \ -0.707 \ 0 \ 0.707)^T$	$(0 \ -0.535 \ 0.2773 \ 0.802)^T$
$\mathbf{q}_{B/A}$	$(0 \ -0.707 \ 0 \ 0.707)^T$	$(0 \ -0.512 \ 0.384 \ 0.768)^T$
Angular rates:		
$\boldsymbol{\omega}_{B/I}^B$ (rad/s)	$(-1.071 \ 0 \ 0)^T \times 10^{-4}$	N/A
$\boldsymbol{\omega}_{A/I}^A$ (rad/s)	$(0 \ 0 \ 3.241)^T \times 10^{-4}$	$(0.02 \ 0.02 \ 3.403)^T \times 10^{-4}$
$\boldsymbol{\mu}$ (rad/s)	$(1 \ 1 \ 1)^T \times 10^{-5}$	$(0 \ 0 \ 0)^T$

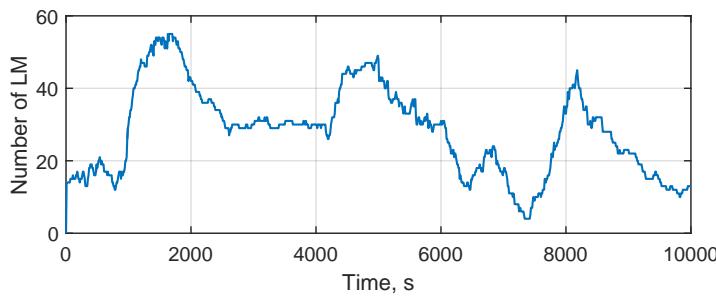


Figure 8.5: KS1: number of landmarks seen by NAVCAM.

Let us start with the number of landmarks seen by the navigation camera. Figure 8.5 shows the variation of this number in time. We can see that the maximum is reached at around 1500 s with 55 landmarks and the minimum is at around 7500 s with only five landmarks. If we look back at Figure 8.1, we would understand this pattern. At the beginning the spacecraft points towards the end of the dog-bone-shaped asteroid almost perpendicularly to the surface, so it records around 18 landmarks. Then it propagates and once it passes the round-shaped end, it immediately covers a large surface area, which results in the maximum of landmarks seen. At around 2500 s the spacecraft is pointing almost perpendicularly to the hollow centre part of the asteroid, therefore, we see a decrease in the number. Later, it reaches the other end of Kleopatra, where the minimum occurs.

Position and velocity errors in Figure 8.6 show surprising results. It turns out that QVEKF and DQEKF perform identically at the steady state, thus no comparison between them can be done. In general, the filters converge in about 1500 s. Velocity estimates stay steady for all the time with a 3σ value of ≈ 0.4 m/s. Position errors, though, appear to have two peaks in the steady state happening at 1500 s and 7500 s, which can be explained by the number of landmarks seen by the NAVCAM. The first peak coincides with the flat region between 2500 s and 4000 s. Although the number of landmarks is still relatively high (≈ 30), however, it appears that the filters react sensitively to the sudden change of this number. The second peak matches the minimum of the landmarks seen. This proves that there is a strong influence of landmarks in the FOV on the position estimates. Furthermore, the filter 3σ values seem to match the estimates in the steady state, but not in the transient phase in the beginning. Fast convergence of the standard deviations happens before the estimates converge. This could mean that the filter is not tuned perfectly, although the estimates show a good performance.

Inertial and relative attitude errors, shown in Figure 8.7, converge to a steady-state in the same

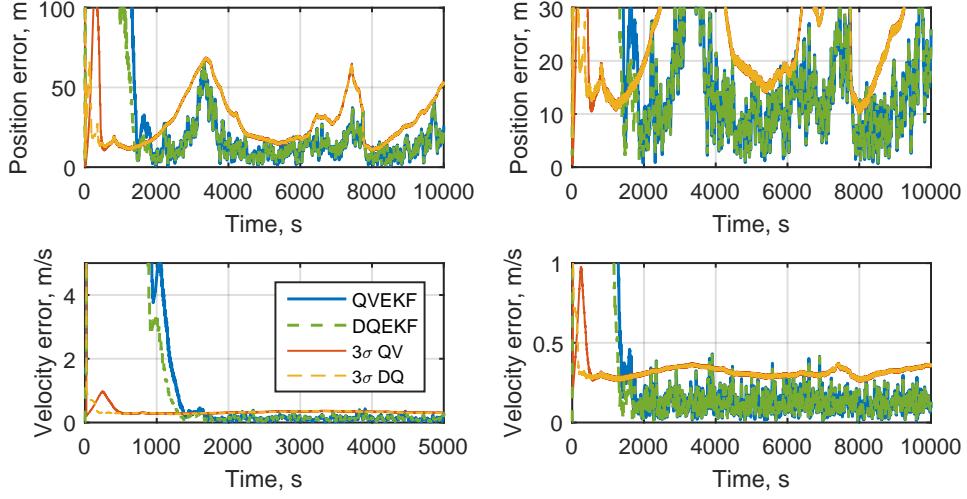


Figure 8.6: KS1: position and velocity errors.

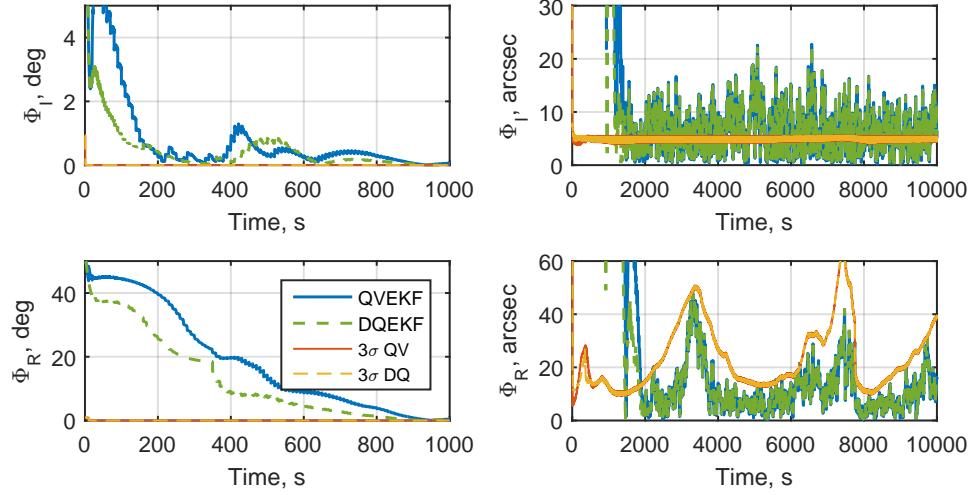


Figure 8.7: KS1: inertial and relative attitude errors.

time position and velocity errors do. The steady state performance of both filters is again exactly the same, but on the large scale some differences occur. DQEKF seem to have a faster convergence and with smaller errors. The relative attitude error has the aforementioned two peaks in the steady state, while they are absent in the inertial estimate. This outcome is logical, since the inertial attitude is measured by the star-tracker, so it is not affected by the number of landmarks seen by the NAVCAM. In general, the inertial estimates fall bellow 15 arcsec (most of the time) and the relative attitude error reaches the minimum of around 15 arcsec. We recall the star tracker having the standard deviation of the noise around the roll axis of 45 arcsec, therefore we see a ninefold reduction of the noise in the inertial attitude errors. Theoretical 3σ values for Φ_I do not represent the actual estimates, since they stay below the estimates most of the time. The transient phase is again not represented well by the theoretical values.

The final results of KS1 case are the gyroscope drift and angular rate estimates, presented in Figure 8.8. Drift errors converge to the steady-state marginally faster, because the estimates rely

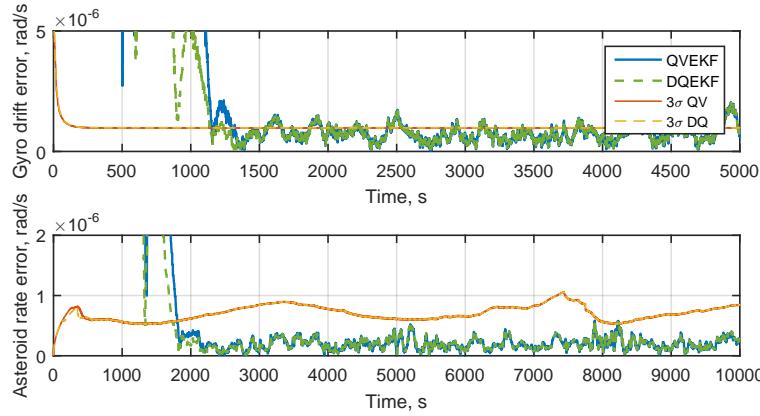


Figure 8.8: KS1: gyroscope drift and asteroid rate errors.

mainly on the star-tracker measurements. The asteroid rate takes almost 2000 s to converge. The two filters, as it was seen in previous results, perform in the same manner. In general, the drift errors are concentrated within 3σ boundary, but not perfectly, and errors of the rate stay within them all the time.

The first test case was run with a retrograde equatorial orbit around Kleopatra. The results showed no significant differences between QVEKF and DQEKF in fact, they performed identically in the steady state. A clear dependency on the landmarks in the FOV can be seen in the estimates, which means the system is rather sensitive to that. Furthermore, the error variances appear not to follow the estimates in the transient phase until convergence.

8.3.2. Scenario 1, Itokawa

Scenario 1 for Itokawa (IS1) is the same retrograde orbit, but with significantly lower altitude (1000 m), since the asteroid itself is relatively small. The gravity field of Itokawa is accordingly much weaker and the spinning rate of the asteroid is around two times lower than the one of Kleopatra, which results in slower orbital dynamics around Itokawa, as seen in Figure 8.1. Initial conditions are listed in Table 8.3. Covariance matrices are tuned accordingly and provided in Appendix A.2.

Table 8.3: IS1 initial conditions.

	Real	Estimated
Position and velocity:		
\mathbf{R}_A (m)	$(1000 \ 0 \ 0)^T$	$(1001 \ 1 \ 1)^T$
\mathbf{V}_A (m/s)	$(0 \ -0.1997 \ 0)^T$	$(0.001 \ -0.1987 \ 0.001)^T$
Attitude:		
$\mathbf{q}_{B/I}$	$(0 \ -0.707 \ 0 \ 0.707)^T$	$(0 \ -0.535 \ 0.2773 \ 0.802)^T$
$\mathbf{q}_{B/A}$	$(0 \ -0.707 \ 0 \ 0.707)^T$	$(0 \ -0.512 \ 0.384 \ 0.768)^T$
Angular rates:		
$\boldsymbol{\omega}_{B/I}^B$ (rad/s)	$(-5.583 \ 0 \ 0)^T \times 10^{-5}$	N/A
$\boldsymbol{\omega}_{A/I}^A$ (rad/s)	$(0 \ 0 \ 1.439)^T \times 10^{-4}$	$(0.02 \ 0.02 \ 1.510)^T \times 10^{-4}$
$\boldsymbol{\mu}$ (rad/s)	$(1 \ 1 \ 1)^T \times 10^{-5}$	$(0 \ 0 \ 0)^T$

Analysis of the results starts again with the landmarks seen by the navigation camera, given in Figure 8.9. At the beginning of the orbit around ten landmarks are seen, which after 3000 s starts

increasing and reaches the maximum of 55 landmarks in the FOV between 6500 and 8500 s. This is explained similarly as in the case of Kleopatra. The rapid increase is caused by a larger angle between the boresight and local surface normal vectors, because then a larger surface area is covered by the FOV.

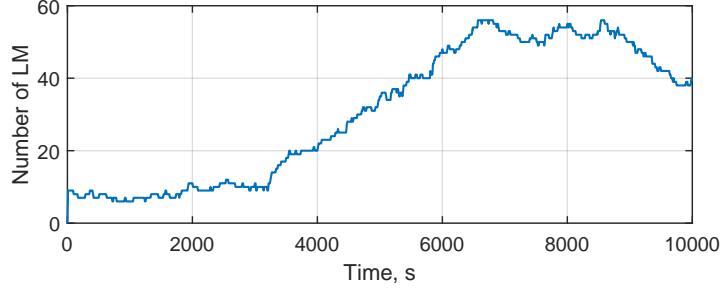


Figure 8.9: IS1: number of landmarks seen by NAVCAM.

Position and velocity estimates' errors are presented in Figure 8.10. The filters are repeatedly indistinguishable, so a different dynamics/asteroid does not affect this similarity. In this case the convergence happens at around 4000 s and the errors stay moderately stable. There are no significant peaks as in Kleopatra case, because there is no rapid decrease in the number of seen landmarks. The errors are roughly below 0.1 m and 0.4 mm/s for position and velocity respectively after the convergence.

Attitude estimates, shown in Figure 8.11, are almost the same as in Kleopatra case. QVEKF and DQEKF perform identically even on the large scale (left of Figure 8.11). Inertial attitude is estimated with up to 15 arcsec accuracy and the relative one goes down to ≈ 15 arcsec, which essentially is the same as in KS1 case.

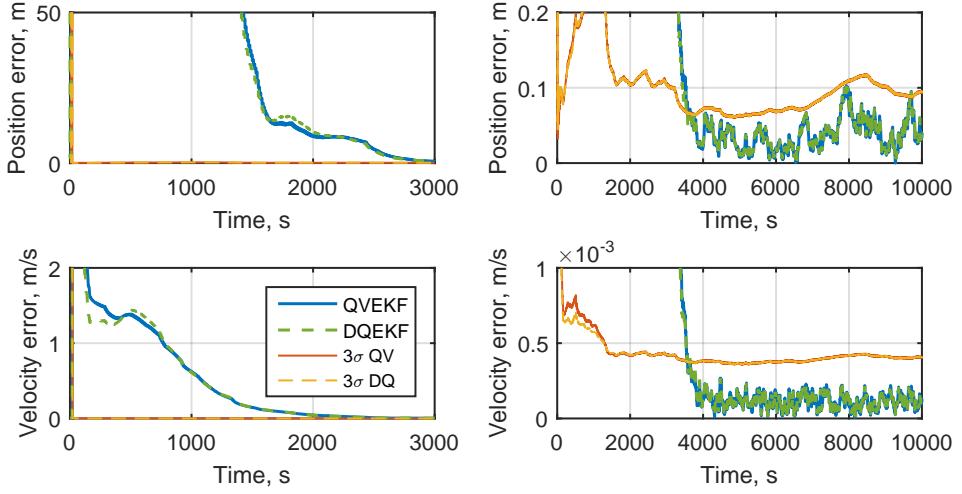


Figure 8.10: IS1: position and velocity errors.

Gyroscope drift and Itokawa angular rate estimates, given in Figure 8.12, are again the same as in KS1 case. The only difference is the convergence time, which is longer for Itokawa, which can be explained by the slower dynamics around the asteroid. The scenery changes less rapidly and thus the increase of the number of landmarks happens later in time, which translates into the longer convergence time.

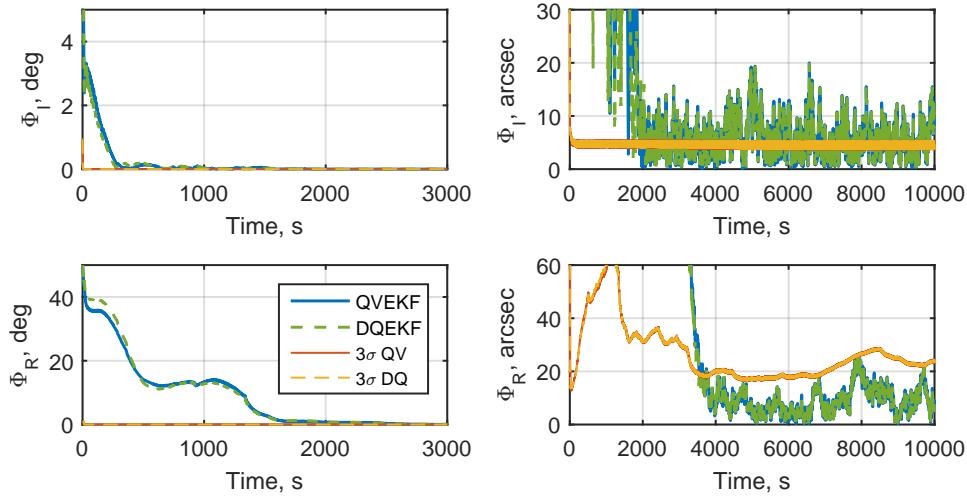


Figure 8.11: IS1: inertial and relative attitude errors.

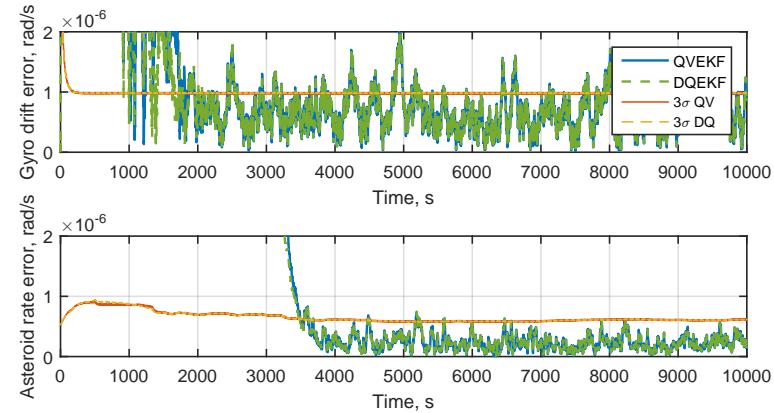


Figure 8.12: IS1: gyroscope drift and asteroid rate errors.

Scenario 1 for Itokawa was a retrograde orbit at a distance of 1000 m from Itokawa. No difference between QVEKF and DQEKF was detected. The steady state performance of the filters was also very similar to KS1 case. The slow dynamics around the asteroid appeared to have some influence on the convergence rate, since the spacecraft spent more time seeing the same landmarks in its FOV, thus the observability was reduced.

8.3.3. Scenario 2, Kleopatra

Scenario 2 for Kleopatra (KS2) is practically the same orbit as KS1, but it is prograde. Please note that it is prograde in the inertial reference frame, but since the orbit is higher than the asteroid synchronous orbit, Kleopatra spins faster than the orbit propagates. This results in an apparent retrograde motion with respect to the asteroid. Initial conditions for this case are provided in Table 8.4, while the covariance matrices are listed in Appendix A.2.

The number of seen landmarks, shown in Figure 8.13, is very similar to the one for KS1, but it is extended in time. This is due to the fact that the relative position changes in the same way it was for a retrograde orbit, however, this time much slower. As a result, we see only one peak followed by the flat region, instead of the three peaks, as seen in Figure 8.5.

Table 8.4: KS2 initial conditions.

	Real	Estimated
Position and velocity:		
R_A (km)	$(300 \ 0 \ 0)^T$	$(301 \ 1 \ 1)^T$
V_A (m/s)	$(0 \ -65.10 \ 0)^T$	$(1 \ -64.10 \ 1)^T$
Attitude:		
$q_{B/I}$	$(0 \ -0.707 \ 0 \ 0.707)^T$	$(0 \ -0.535 \ 0.2773 \ 0.802)^T$
$q_{B/A}$	$(0 \ -0.707 \ 0 \ 0.707)^T$	$(0 \ -0.512 \ 0.384 \ 0.768)^T$
Angular rates:		
$\omega_{B/I}^B$ (rad/s)	$(1.071 \ 0 \ 0)^T \times 10^{-4}$	N/A
$\omega_{A/I}^A$ (rad/s)	$(0 \ 0 \ 3.241)^T \times 10^{-4}$	$(0.02 \ 0.02 \ 3.403)^T \times 10^{-4}$
μ (rad/s)	$(1 \ 1 \ 1)^T \times 10^{-5}$	$(0 \ 0 \ 0)^T$

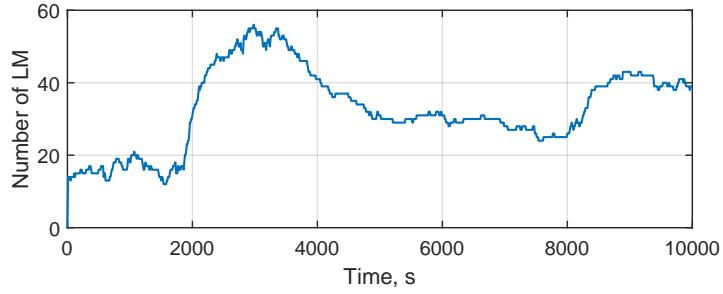


Figure 8.13: KS2: number of landmarks seen by NAVCAM.

Position and velocity estimates, as seen in Figure 8.14, resemble the results from KS1 case, but they are extended in time, because the relative dynamics is much slower in this 'prograde' case. The convergence time, accuracy and the trend appears to be the same as given in Figure 8.6. Furthermore, the two filters have practically the same performance.

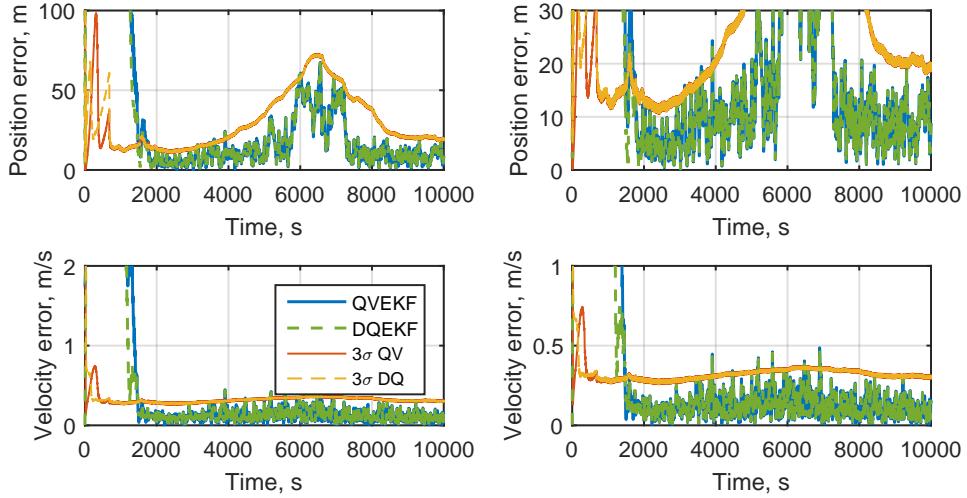


Figure 8.14: KS2: position and velocity errors.

Attitude and drift/angular rate plots are not included, because they show a very similar performance to the KS1 case, but with a slower process.

8.3.4. Scenario 2, Itokawa

Scenario 2 for Itokawa (IS2) is a prograde orbit, but due to the rotation of Itokawa, the relative position of the spacecraft has an apparent retrograde motion relative to the asteroid. Initial conditions are provided in Table 8.5 and the covariance matrices are presented in Appendix A.2.

Table 8.5: IS2 initial conditions.

	Real	Estimated
Position and velocity:		
\mathbf{R}_A (m)	$(1000 \ 0 \ 0)^T$	$(1001 \ 1 \ 1)^T$
\mathbf{V}_A (m/s)	$(0 \ -0.088 \ 0)^T$	$(0.001 \ -0.1987 \ 0.001)^T$
Attitude:		
$\mathbf{q}_{B/I}$	$(0 \ -0.707 \ 0 \ 0.707)^T$	$(0 \ -0.535 \ 0.2773 \ 0.802)^T$
$\mathbf{q}_{B/A}$	$(0 \ -0.707 \ 0 \ 0.707)^T$	$(0 \ -0.512 \ 0.384 \ 0.768)^T$
Angular rates:		
$\boldsymbol{\omega}_{B/I}^B$ (rad/s)	$(5.583 \ 0 \ 0)^T \times 10^{-5}$	N/A
$\boldsymbol{\omega}_{A/I}^A$ (rad/s)	$(0 \ 0 \ 1.439)^T \times 10^{-4}$	$(0.02 \ 0.02 \ 1.510)^T \times 10^{-4}$
$\boldsymbol{\mu}$ (rad/s)	$(1 \ 1 \ 1)^T \times 10^{-5}$	$(0 \ 0 \ 0)^T$

Relative dynamics around Itokawa are slow and with the prograde motion they slow down even more. This effect can be clearly seen in the number of landmarks in the FOV, shown in Figure 8.15. Roughly ten landmarks are seen by the camera for most of the simulation time, and the increase shows as discrete jumps with constant phases.

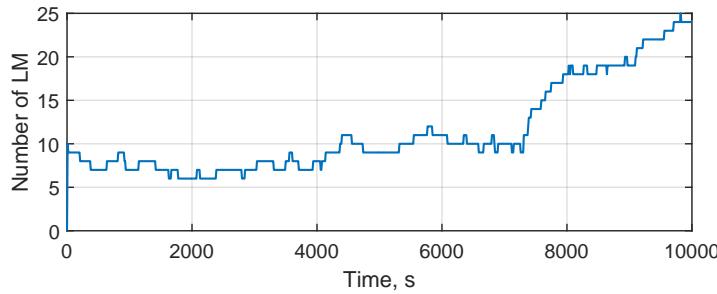


Figure 8.15: IS2: number of landmarks seen by NAVCAM.

Position and velocity estimates, as seen in Figure 8.16, seem to converge to the steady state in roughly 5000 s, which is around 1000 s later than in IS1 case. Since only ten landmarks are seen for most of the time, it takes a longer time to converge, and the estimates reach values as low as for IS1 only by the end of simulation, when the number of landmarks has increased. On the large scale, the difference of QVEKF and DQEKF is more distinct, with the first having larger errors at the beginning.

Inertial attitude estimates, given in Figure 8.17, are not affected by the different scenario and stay within the boundary of 15-20 arcsec, as in all the previously discussed cases. Relative attitude errors reach the 15 arcsec limit only by the end of the simulation time, when the number of visible landmarks has increased. On the large scale DQEKF have lower errors compared to QVEKF but in the steady state they are still the same.

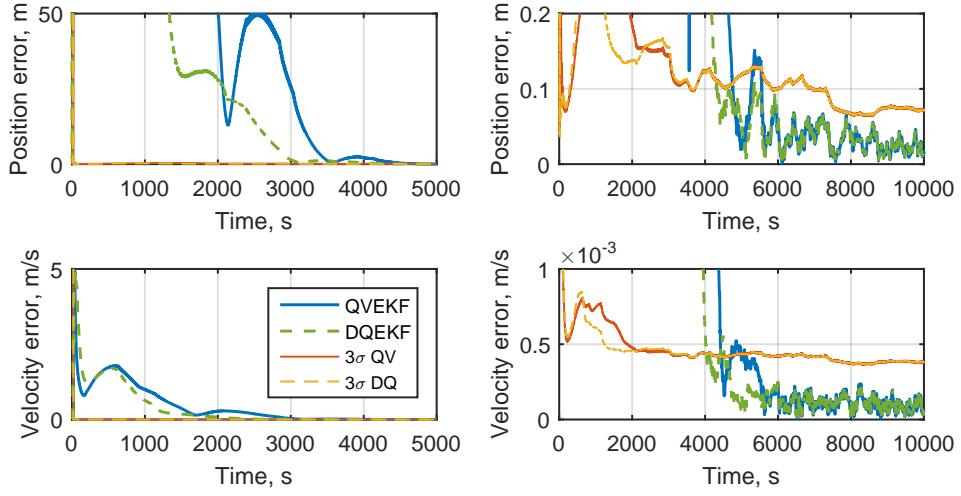


Figure 8.16: IS2: position and velocity errors.

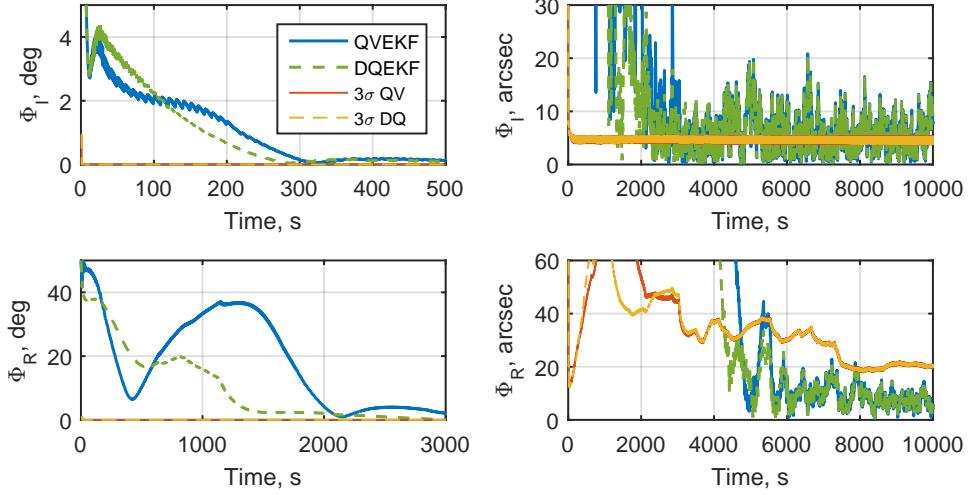


Figure 8.17: IS2: inertial and relative attitude errors.

Gyroscope drift and asteroid rate errors are very similar to the IS1 case, thus not included and further discussed. The main difference is the convergence time, which as for the position, has increased by roughly 1000 s in comparison to the IS1 case.

Scenario 2 around Itokawa (IS2) had very slow dynamics, which resulted in the delay of convergence. Also it made the differences of the QVEKF and DQEKF more distinct on the large scale, although the steady state performance remained identical.

8.3.5. Scenario 3, Kleopatra

Scenario 3 around Kleopatra (KS3) is an asteroid-synchronous orbit, which theoretically means that the spacecraft stays at the same position with respect to the asteroid. Coordinates of equilibrium points for Kleopatra were taken from the paper of Jiang (2015). Furthermore, although the spacecraft stays almost still, its attitude changes, mainly due to the gravity-gradient torque, Figure 8.3. All initial conditions are provided in Table 8.6 and Appendix A.2.

Table 8.6: KS3 initial conditions.

	Real	Estimated
Position and velocity:		
R_A (km)	$(1.164 \quad 100.738 \quad -5.415)^T$	$(2.164 \quad 101.738 \quad -4.415)^T$
V_A (m/s)	$(0 \quad 0 \quad 0)^T$	$(1 \quad 1 \quad 1)^T$
Attitude:		
$q_{B/I}$	$(-0.51 \quad -0.49 \quad 0.51 \quad 0.49)^T$	$(-0.763 \quad -0.3627 \quad 0.385 \quad -0.370)^T$
$q_{B/A}$	$(-0.51 \quad -0.49 \quad 0.51 \quad 0.49)^T$	$(-0.820 \quad -0.255 \quad 0.369 \quad -0.355)^T$
Angular rates:		
$\omega_{B/I}^B$ (rad/s)	$(-3.241 \quad 0 \quad 0)^T \times 10^{-4}$	N/A
$\omega_{A/I}^A$ (rad/s)	$(0 \quad 0 \quad 3.241)^T \times 10^{-4}$	$(0.02 \quad 0.02 \quad 3.403)^T \times 10^{-4}$
μ (rad/s)	$(1 \quad 1 \quad 1)^T \times 10^{-5}$	$(0 \quad 0 \quad 0)^T$

The number of landmarks seen by NAVCAM is provided in Figure 8.18. It is a very unfavourable case, because most of the time there are less than five of these points in the FOV. At roughly 8000 s there are no landmarks detected at all.

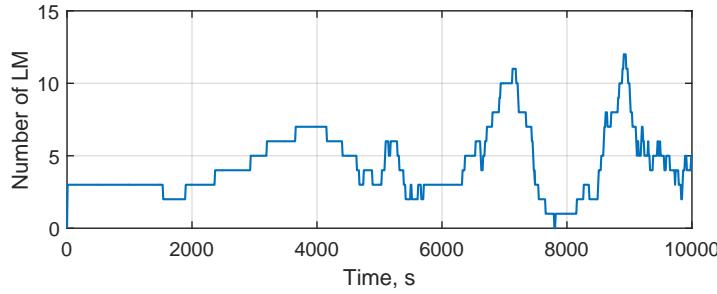


Figure 8.18: KS3: number of landmarks seen by NAVCAM.

Position errors, as shown in Figure 8.19, are much higher when compared to the cases discussed before. The errors barely fall below 20 m value, but rather stay around 100 m. Contrary, the velocity estimates are much better and are actually the same as in the KS1 and KS2 simulations. It is an interesting result, since velocity itself is not measured, but rather estimated via the coupling with the position estimates. Furthermore, QVEKF and DQEKF have the same performances, as it was already expected from the previous results.

Inertial attitude estimates are unaltered, since they are independent from the relative measurements. The relative attitude, however, suffers from a low number of visible landmarks, as seen in Figure 8.20. Most of the time the errors are larger than 200 arcsec and only at around 9000 s they reach roughly 50 arcsec, which corresponds to the time of maximum landmarks in the FOV.

Finally, the drift errors in Figure 8.21 show identical performance to KS1 and KS2 case, but the asteroid rate converges very late and stays with around 2.5×10^{-6} rad/s error. This, of course, complicates the estimation of other variables, which are dependent on the asteroid rate.

Scenario 3 for Kleopatra was a stationary orbit, where the spacecraft had no (or very little) relative velocity with respect to the asteroid. However, lower a distance (≈ 100 km) made fewer landmarks visible for the NAVCAM, which resulted in a worse performance of the filters. No difference between QVEKF and DQEKF was observed in this scenario.

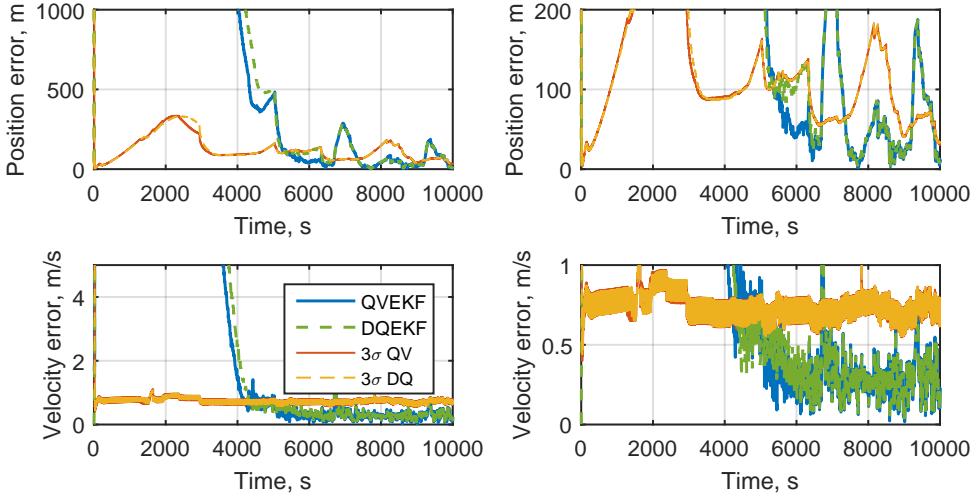


Figure 8.19: KS3: position and velocity errors.

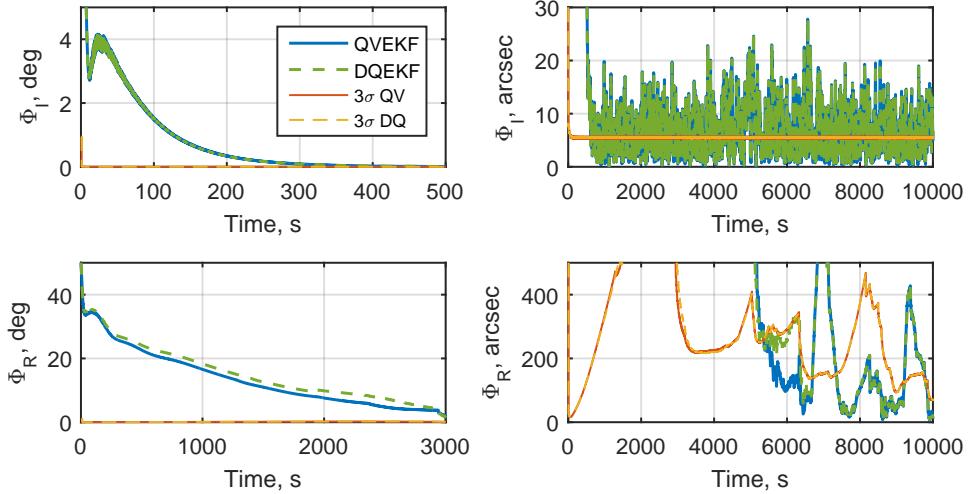


Figure 8.20: KS3: inertial and relative attitude errors.

8.3.6. Scenario 3, Itokawa

Scenario 3 for Itokawa (IS3) is an asteroid synchronous orbit, which was calculated analytically, assuming a central gravity field, therefore, we can see in Figure 8.3 that the spacecraft is slowly drifting from the equilibrium point. Initial conditions for this case are provided in Table 8.7 and Appendix A.2.

Since Itokawa is a relatively small asteroid, its gravity field and, thus, the gravity-gradient torque produced by it is accordingly smaller. As a result, the relative attitude does not change that much during the simulation and the spacecraft keeps pointing at almost the same surface area. This can also be seen in Figure 8.5, where the number of landmarks in the FOV are presented. The number stays almost the same throughout all the simulation, thus it is an interesting case, to see how the navigation filters perform with the same set of data all the time.

Position and velocity errors are comparable to IS1 and IS2 cases. The convergence happens in 4000 s and the errors stay within the theoretical 3 σ values of roughly 0.1 m and 0.4 mm/s. Relative

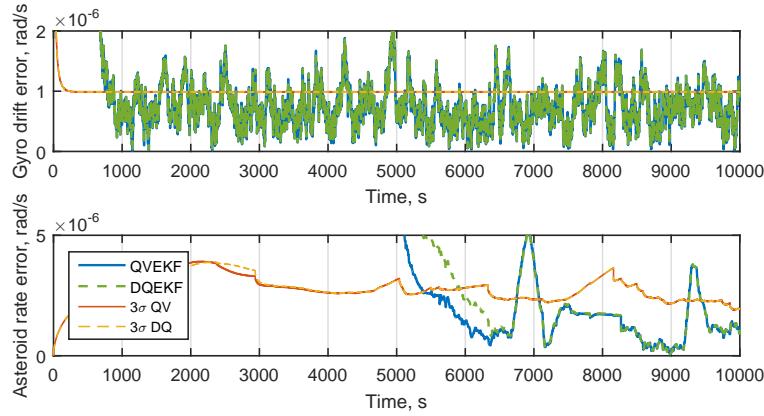


Figure 8.21: KS3: gyroscope drift and asteroid rate errors.

Table 8.7: IS3 initial conditions.

	Real	Estimated
Position and velocity:		
R_A (m)	$(0 \quad 532.07 \quad 0)^T$	$(1 \quad 533.07 \quad -4.415)^T$
V_A (m/s)	$(0 \quad 0 \quad 0)^T$	$(0.001 \quad 0.001 \quad 0.001)^T$
Attitude:		
$q_{B/I}$	$(-0.51 \quad -0.49 \quad 0.51 \quad 0.49)^T$	$(-0.763 \quad -0.3627 \quad 0.385 \quad -0.370)^T$
$q_{B/A}$	$(-0.51 \quad -0.49 \quad 0.51 \quad 0.49)^T$	$(-0.820 \quad -0.255 \quad 0.369 \quad -0.355)^T$
Angular rates:		
$\omega_{B/I}^B$ (rad/s)	$(-1.439 \quad 0 \quad 0)^T \times 10^{-4}$	N/A
$\omega_{A/I}^A$ (rad/s)	$(0 \quad 0 \quad 1.439)^T \times 10^{-4}$	$(0.02 \quad 0.02 \quad 1.510)^T \times 10^{-4}$
μ (rad/s)	$(1 \quad 1 \quad 1)^T \times 10^{-5}$	$(0 \quad 0 \quad 0)^T$

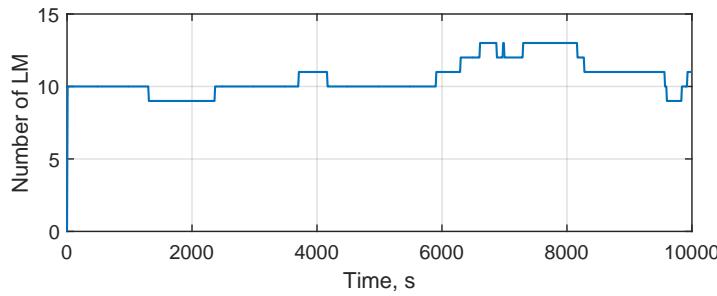


Figure 8.22: IS3: number of landmarks seen by NAVCAM.

attitude errors, given in Figure 8.23, appear to be larger, which are with a 3σ envelope of 40 arsec (IS1 and IS2 had ≈ 15 arsec). One possible explanation for this could be the fact that all NAVCAM measurement vectors in this case are restricted within five degrees of the FOV, so they are close to being collinear, which is an unfavourable condition for attitude estimation. Gyroscope drift and asteroid rate errors are not included, since they are analogous to the ones in IS1 and IS2 case. Finally the two filters perform identically, which has already been expected.

Scenario 3 was a stationary orbit around Itokawa. The spacecraft and its attitude stayed very still, which resulted in the same scenery for the NAVCAM throughout the simulation time with

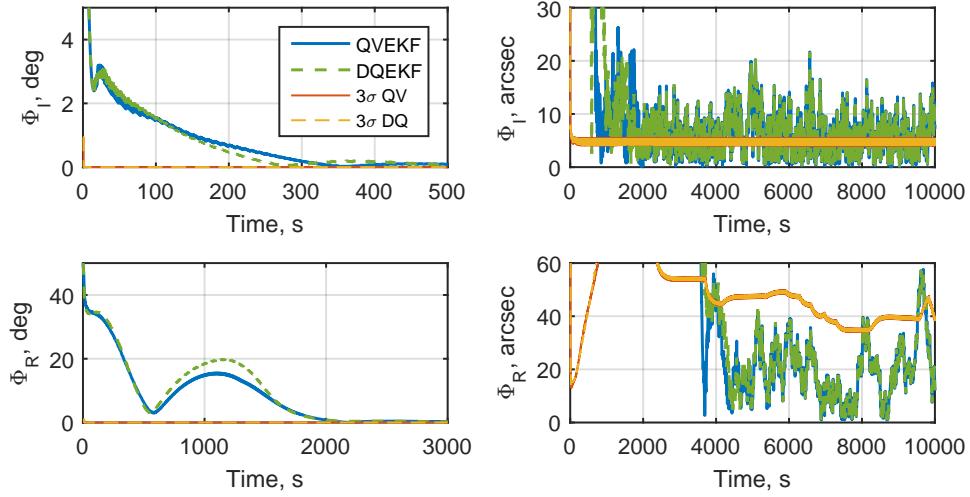


Figure 8.23: IS3: inertial and relative attitude errors.

roughly ten landmarks in the FOV of the camera. This, however, did not have an effect on the position, velocity, drift or the asteroid rate estimates, but there was a moderate increase of the relative attitude errors.

8.3.7. Scenario 4, Kleopatra

Scenario 4 for Kleopatra (KS4) is a polar orbit with a rapidly changing dynamics, as seen in Figure 8.4. Initial conditions are listed in Table 8.8 and the covariance matrices are provided in Appendix A.2.

Table 8.8: KS4 initial conditions.

	Real	Estimated
Position and velocity:		
\mathbf{R}_A (km)	$(0 \ 0 \ 100)^T$	$(1 \ 1 \ 101)^T$
\mathbf{V}_A (m/s)	$(0 \ -55.65 \ 0)^T$	$(1 \ -54.65 \ 1)^T$
Attitude:		
$\mathbf{q}_{B/I}$	$(1 \ 0 \ 0 \ 0)^T$	$(0.945 \ -0.189 \ 0.189 \ -0.189)^T$
$\mathbf{q}_{B/A}$	$(1 \ 0 \ 0 \ 0)^T$	$(0.905 \ -0.272 \ 0.181 \ -0.272)^T$
Angular rates:		
$\boldsymbol{\omega}_{B/I}^B$ (rad/s)	$(4.381 \ 1.854 \ 0)^T \times 10^{-4}$	N/A
$\boldsymbol{\omega}_{A/I}^A$ (rad/s)	$(0 \ 0 \ 3.241)^T \times 10^{-4}$	$(0.02 \ 0.02 \ 3.403)^T \times 10^{-4}$
$\boldsymbol{\mu}$ (rad/s)	$(1 \ 1 \ 1)^T \times 10^{-5}$	$(0 \ 0 \ 0)^T$

Relative position in this orbit changes fast, the distance to the asteroid is rather small, thus the gravitational perturbations are larger. Furthermore the gravity-gradient torque affects the satellite, therefore, at two instances in time it loses the asteroid from its line-of-sight, as seen in Figure 8.24. At ≈ 1500 s and ≈ 6000 s the spacecraft does not detect any landmarks for up to 2000 s. This is a challenging situation, because then the filters have to rely on the process equations used. However, we recall using the central-gravity field approximation in the filter, which at this distance can cause large errors.

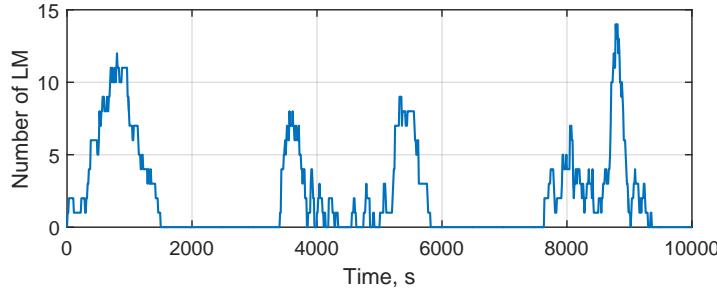


Figure 8.24: KS4: number of landmarks seen by NAVCAM.

Position and velocity estimates, shown in Figure 8.25, seem not to converge properly. During the first 1500 s velocity errors drop below 5 m/s, but then the NAVCAM measurements become unavailable, and the error starts increasing unboundedly. The same effect appears at around 5800 s again. The reason of this is the process model we use. Since the measurements are not present, the filter relies on the dynamics, which for the gravity field were not very accurate. Basically, the filters do not reach a steady state in this case, which results in larger differences between QVEKF and DQEKF. As it has been shown before, the filters are identical only in the steady state. Theoretical 3σ boundaries seem to follow the same trend as the estimates have, but their actual values are much lower, which means that the filters 'think' that the estimates are much more accurate, while they are not. At the very end of the simulation, just after the maximum number of landmarks in the FOV appear, Figure 8.24, the errors fall to the values close to the steady state performance in previous cases; however, the landmarks are lost again from the line-of-sight of the FOV, so the errors increase again.

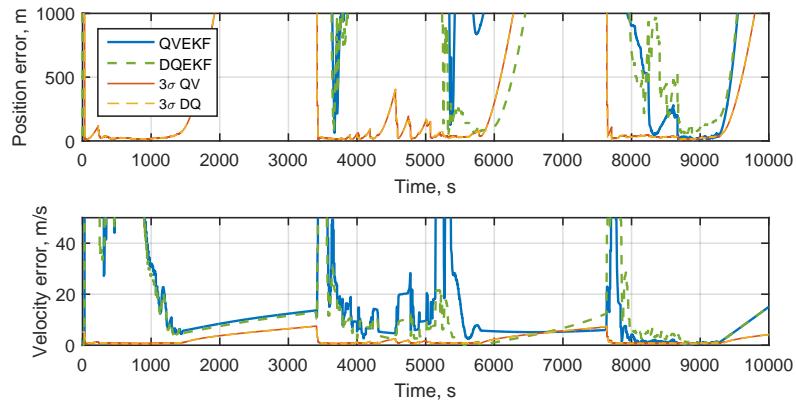


Figure 8.25: KS4: position and velocity errors.

Figure 8.26 presents the results of the inertial and the relative attitude errors. The first one shows the convergence to the steady state with the equal performance for the both filters, while the relative attitude errors do not converge. The errors stay roughly below 5 degrees value and follow a similar pattern as the position and velocity estimates have. The dual quaternion filter has moderately smaller errors during the simulation time, but around 7800 s, when the NAVCAM measurements become available again, the error suddenly increases. The theoretical standard deviations are much lower than the errors, which is a sign that the filters are not tuned properly.

Finally, the drift errors, shown in Figure 8.27, seem to be unaltered. The asteroid rates, however, have significantly higher errors, since the filters do not reach the steady-state. During the periods

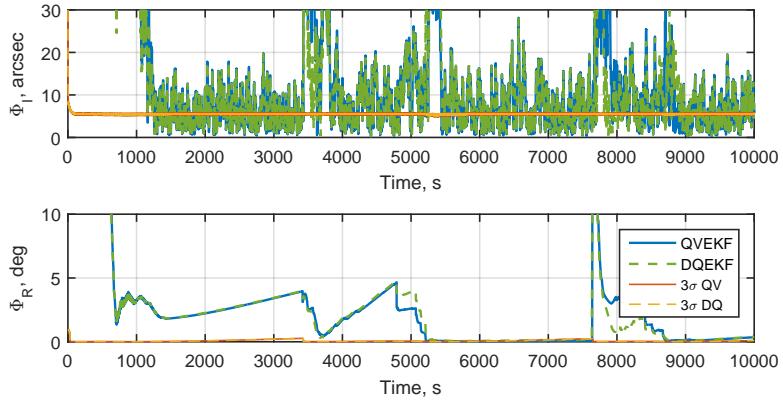


Figure 8.26: KS4: inertial and relative attitude errors.

when relative measurements are unavailable, the rate error stays constant, because its dynamics is invariable with time. It also appears that DQEKF has better accuracy than QVEKF. At ≈ 6000 s the rate error drops below the theoretical 3σ value, however, later the estimate overshoots and becomes less accurate.

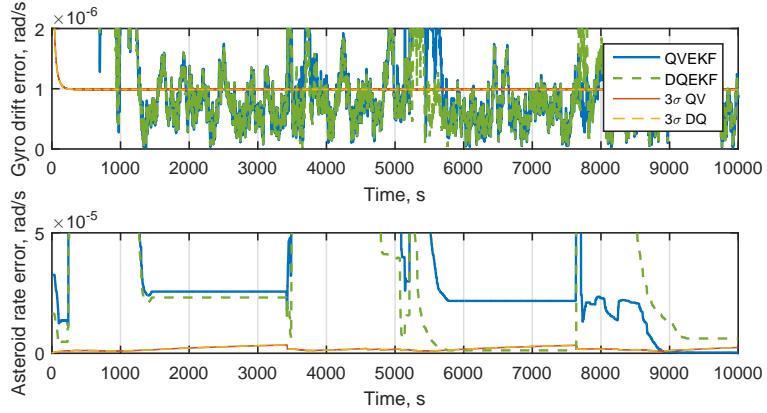


Figure 8.27: KS4: gyroscope drift and asteroid rate errors.

Scenario 4 for Kleopatra (KS4) was a challenging polar orbit. The measurements were lost two times for periods up to 2000 s, which resulted in the absence of convergence of the relative states in both filters. This translated in larger errors between QVEKF and DQEKF with the latter being moderately more accurate in some of the cases. However, the theoretical performance of the filters proved to be very poor, since the 3σ boundaries are greatly offset of the actual errors.

8.3.8. Scenario 4, Itokawa

Scenario 4 for Itokawa (IS4) is a polar orbit at ≈ 200 m distance from the centre of the asteroid. Since the gravity field is much weaker, the orbital and attitude motion is less perturbed, when compared to KS4. Initial conditions and filter initialization are provided in Table 8.9 and Appendix A.2, respectively.

The spacecraft is very close to the surface of Itokawa, so its FOV does not cover a large surface area. This can be seen in Figure 8.28, where the number of landmarks is presented. Most of the time, the NAVCAM detects less than five landmarks and at the very end the number goes up to nine, and

Table 8.9: IS4 initial conditions.

	Real	Estimated
Position and velocity:		
\mathbf{R}_A (m)	$(0 \ 0 \ 200)^T$	$(1 \ 1 \ 201)^T$
\mathbf{V}_A (m/s)	$(0 \ -0.125 \ 0)^T$	$(0.001 \ -0.124 \ 0.001)^T$
Attitude:		
$\mathbf{q}_{B/I}$	$(1 \ 0 \ 0 \ 0)^T$	$(0.945 \ -0.189 \ 0.189 \ -0.189)^T$
$\mathbf{q}_{B/A}$	$(1 \ 0 \ 0 \ 0)^T$	$(0.905 \ -0.272 \ 0.181 \ -0.272)^T$
Angular rates:		
$\boldsymbol{\omega}_{B/I}^B$ (rad/s)	$(5.202 \ 0 \ 0)^T \times 10^{-5}$	N/A
$\boldsymbol{\omega}_{A/I}^A$ (rad/s)	$(0 \ 0 \ 1.439)^T \times 10^{-4}$	$(0.02 \ 0.02 \ 1.510)^T \times 10^{-4}$
$\boldsymbol{\mu}$ (rad/s)	$(1 \ 1 \ 1)^T \times 10^{-5}$	$(0 \ 0 \ 0)^T$

then drops to zero.

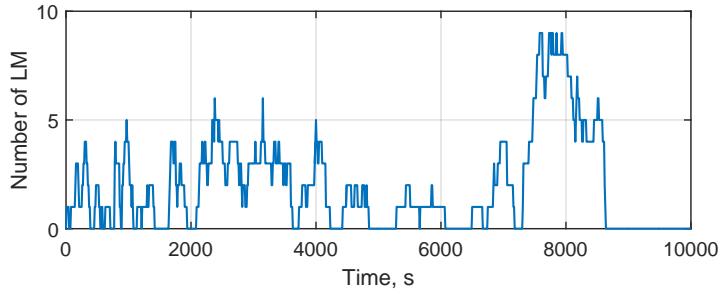


Figure 8.28: IS4: number of landmarks seen by NAVCAM.

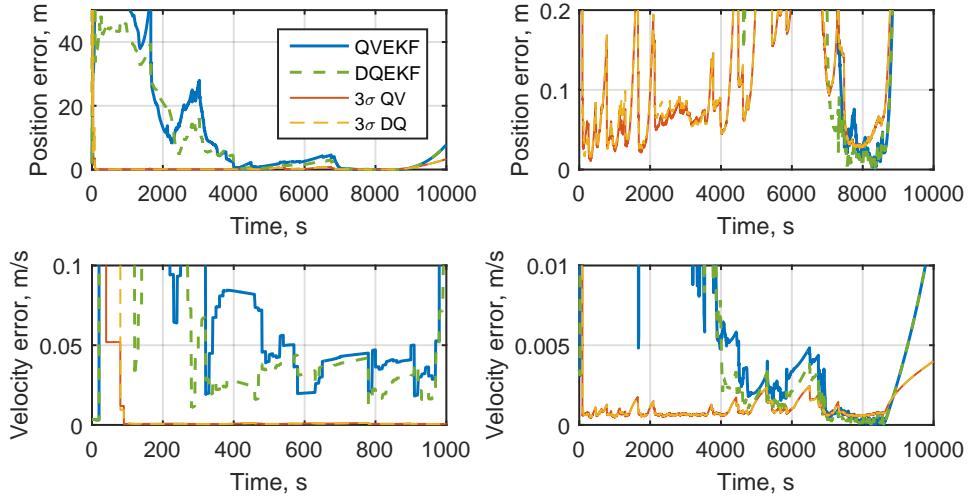


Figure 8.29: IS4: position and velocity errors.

Position and velocity errors, given in Figure 8.29, are in comparison to KS4 case, where the convergence of relative states did not occur. The main difference is that in IS4 case the filters do con-

verge (or come close to the steady-state) at around 8000 s, when there are nine landmarks seen by the NAVCAM. Errors at that 'steady-state' are of the same order as in IS1 and IS2 cases. During the transient phase QVEKF and DQEKF appear to be more distinct, with the latter being marginally more accurate, however, in the steady-state they are identical again. At the very end of simulation the relative measurements become unavailable and thus the errors start diverging unboundedly, because of the inaccuracies in filters' process equations.

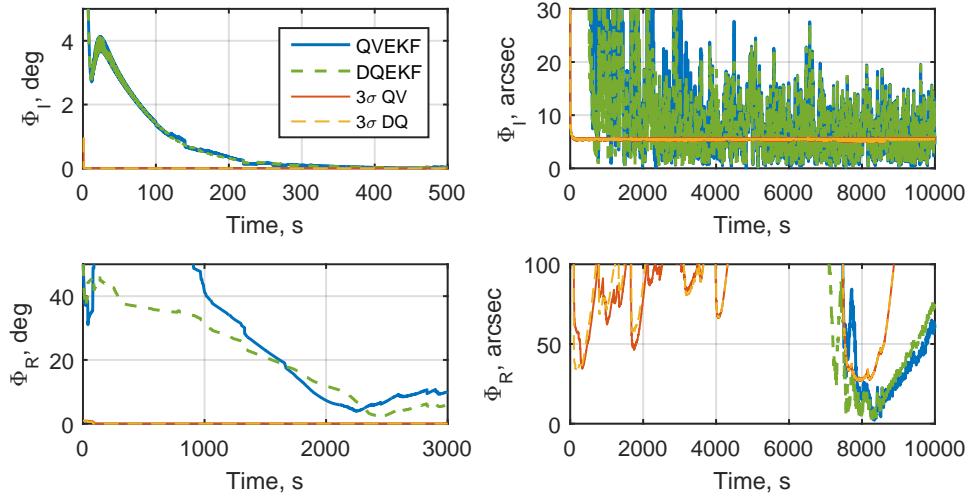


Figure 8.30: IS4: inertial and relative attitude errors.

Relative attitude errors, shown in Figure 8.30, are in accordance with the position and velocity results. The steady-state is reached only at 8000 s with an error of $\approx 20 - 25$ arcsec. The inertial state estimates perform analogously to all of the previous scenarios discussed. Furthermore, QVEKF and DQEKF are rather different for most of the simulation time until the convergence occurs.

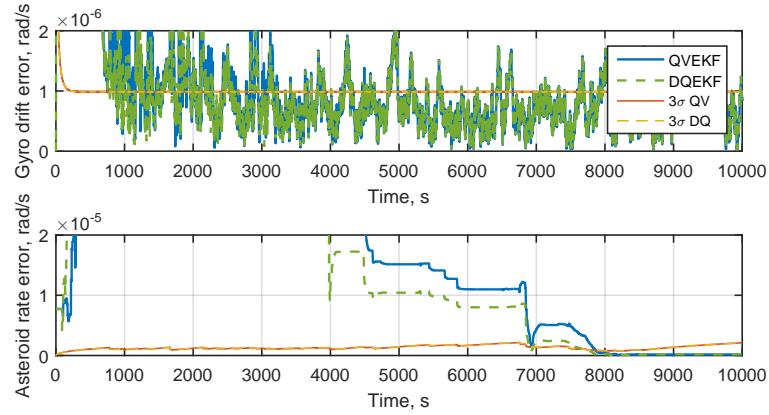


Figure 8.31: IS4: gyroscope drift and asteroid rate errors.

Finally, we see that the asteroid rate in Figure 8.31 also converges and stays constant until the end of simulation, because they are not updated any more (no measurements). DQEKF seems to have lower errors in the transient phase, but the two filters are identical in the steady-state.

Scenario 4 for Itokawa was a polar orbit with a distance of 200 m. Since most of the time there were less than five landmarks seen by the camera, the observability was greatly reduced and only

when there were nine of them, the filters converged. The difference between QVEKF and DQEKF was observable, but mainly due to the lack of convergence. DQEKF seems to have lower errors in the transient phase for all of the estimates. One possible explanation for this is that DQEKF equations of motions have more cross-correlations elements than QVEKF.

8.4. Monte-Carlo Simulation

To compare QVEKF and DQEKF filters further and check the performance of them, a Monte-Carlo (MC) run is done. Due to time constraints and different performance, we cannot run all the scenarios, so one of them is chosen. Basically, one would like to check the steady-state performance of the filters. Since Scenario 3 and Scenario 4 had difficulties achieving it, they were discarded. Furthermore, Scenario 2 had a delay in convergence, thus Scenario 1 is selected. Basically, Scenario 1 was very similar for Kleopatra and Itokawa case, so only the latter one's results are presented, because the steady-state errors were smoother.

A Monte-Carlo analysis is done by running the same filter multiple times with a different seed for the random number generator. Then, for each time step and for each variable the MC mean and variance is calculated. Suppose we run the filter N times, then for each time step k , each state variable will have N different values. Afterwards, the Monte-Carlo mean is calculated, for example, the Monte-Carlo mean for position is as follows:

$$\delta \bar{R}_k = \sum_{n=1}^N \frac{\delta R_{k,n}}{N} \quad (8.5)$$

and its variance:

$$\sigma_{Rk}^2 = \sum_{n=1}^N \frac{(\delta R_{k,n} - \delta \bar{R}_k)^2}{N} \quad (8.6)$$

Having the time history of MC means and standard deviations, one can evaluate the differences between the filters further and check the statistical performance of them. A Monte-Carlo run with $N = 200$ samples is done for both of the filters, and the results are presented below.

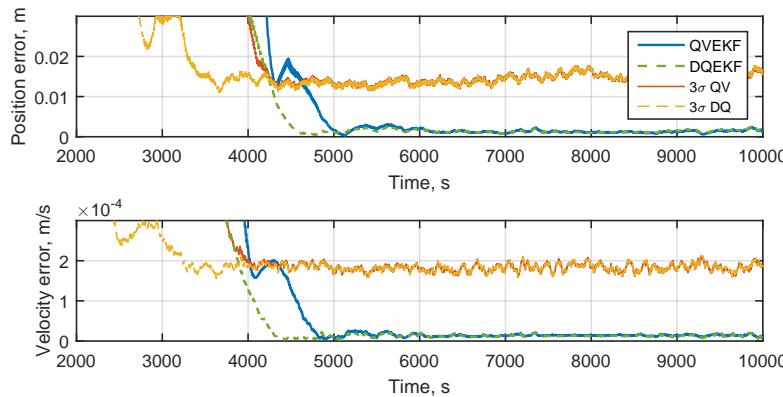


Figure 8.32: MC: position and velocity errors.

Figures 8.32 to 8.34 present the position/velocity, attitude and angular rates errors after the MC run, respectively. In general, all the errors show a great reduction in absolute sense, because theoretically a mean of an error affected by a white-noise is zero. However, there were only 200 runs done, so the errors dropped down only by an order of magnitude or more. Furthermore, in the MC run, QVEKF and DQEKF are performing identically in the steady state. However, the convergence

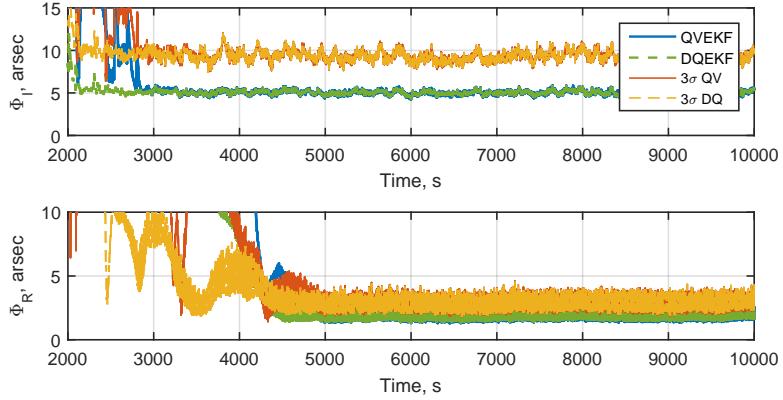


Figure 8.33: MC: Attitude errors.

of DQEKF is ≈ 500 s faster than QVEKF for the relative states, which might be explained by the observability being enhanced due to more cross-correlation elements in the dynamics equations in DQEKF.

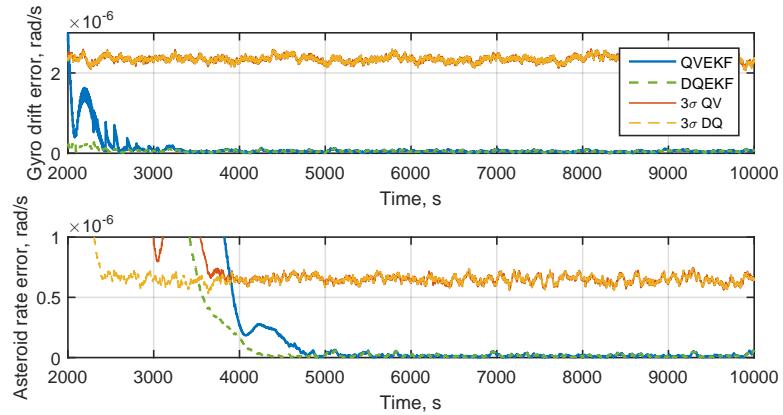


Figure 8.34: MC: Angular rate errors.

After the Monte-Carlo run, it can be further concluded that the quaternion-vector filter and dual quaternion filters are interchangeable in the steady-state. Finally, to check statistical performance of the EKF, Monte-Carlo means and variances are plotted together with the results of a sample run. In this way the performance of the filter can be checked. Ideally, the MC variances should match the ones a Kalman filter calculates. Since, the steady-state performance of the two filters is identical, only one (DQEKF) results are plotted.

Figure 8.35 provides the position and velocity errors after the MC run and a sample run. We can clearly see that for the position the MC 3σ value is much lower than the filter 'thinks' it is. It means that there is still a lot to be improved by tuning the filter or modelling differently the dynamics . The values for the velocity are closer to each other, but there is still an improvement that could be done.

Attitude errors and 3σ values are shown in Figure 8.36. The standard deviation of the inertial attitude is actually larger than the filter estimates. This is mainly influenced by the measurement covariance matrix, R_{ST} , which, thus, should be increased. Relative attitude errors have the MC 3σ value greatly lower than what the filter produces. In fact, the value is as low as the navigation camera noise is. We can see the 3σ value of around 3-4 arcsec, while the NAVCAM image processing errors

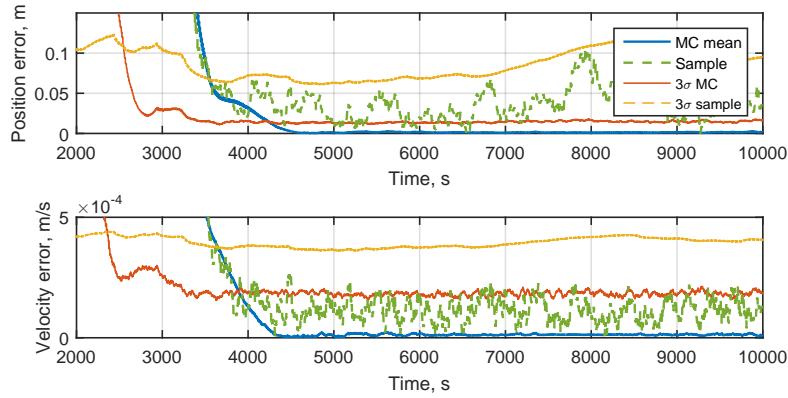


Figure 8.35: MC comparison with a sample run for DQEKF: position and velocity errors.

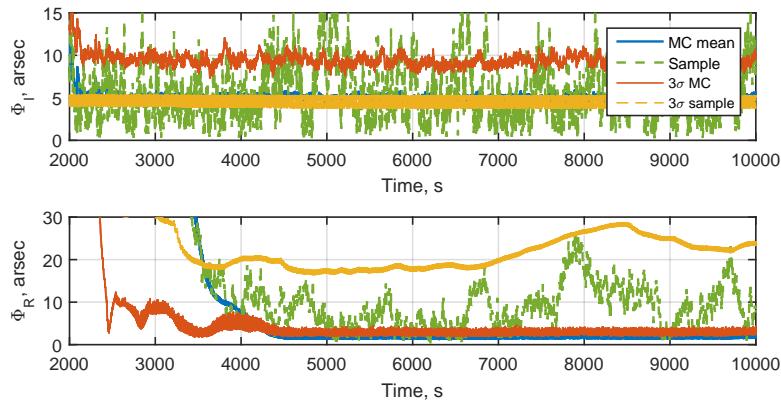


Figure 8.36: MC comparison with a sample run for DQEKF: attitude errors.

have 0.1 pixel accuracy, which corresponds to $5^\circ/1024px \cdot 0.1px \cdot 3600 \approx 1.75$ arcsec. It means that the relative attitude errors can be reduced down to the values of the same order as the noise is.

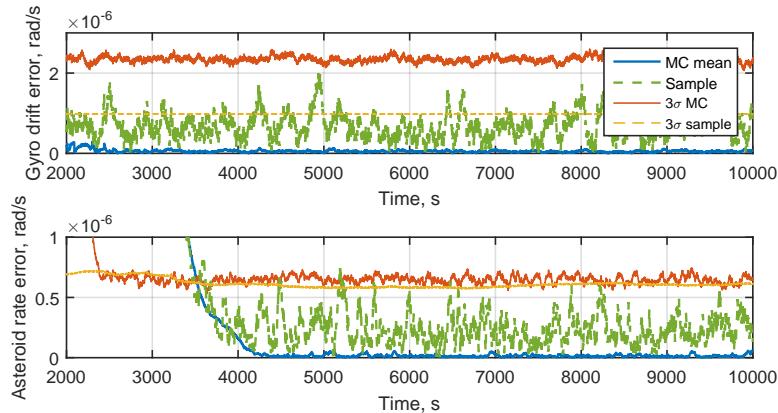


Figure 8.37: MC comparison with a sample run for DQEKF: angular rate errors.

Finally, the drift and the asteroid rate results are provided in Figure 8.37. The drift 3σ value

is actually larger than the one in the filter, so it is a sign of a not perfectly tuned and/or perfectly modelled filter. The asteroid rate errors, though, have a perfect match for the 3σ values, therefore, regarding the asteroid rate, the filter is tuned properly.

8.5. KS4 with Improved Gravity Model

In Section 7.3 an improved gravity model for the filters was developed. In this section the results for KS4 with this model are presented. KS4 is chosen due to the low-altitude orbit, and thus very perturbed gravity field. The filters were run with the same parameters as for KS4 case with central-field model.

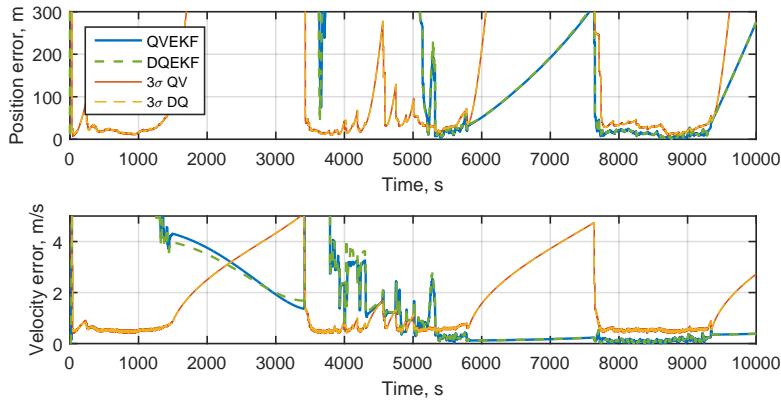


Figure 8.38: KS4 with improved gravity model: position and velocity errors.

Figure 8.38 presents the position and velocity errors. We can clearly see a significantly improved performance when compared to the nominal KS4 case. Both filters do converge to the steady-state at about 8000 s, when the landmark number increases, see Figure 8.24. The errors during that phase are around 15 m and 0.5 m/s for position and velocity, respectively, which are very alike to the KS1 case with a large number of landmarks in the FOV. When there are no landmarks seen by the NAV-CAM, e.g., between 6000 s and 7500 s, the velocity errors do not increase, due to inaccuracies in the design model, but rather stay constant. Also, the difference between DQEKF and QVEKF seem to be smaller, even when the filters have not converged, their performance seems alike.

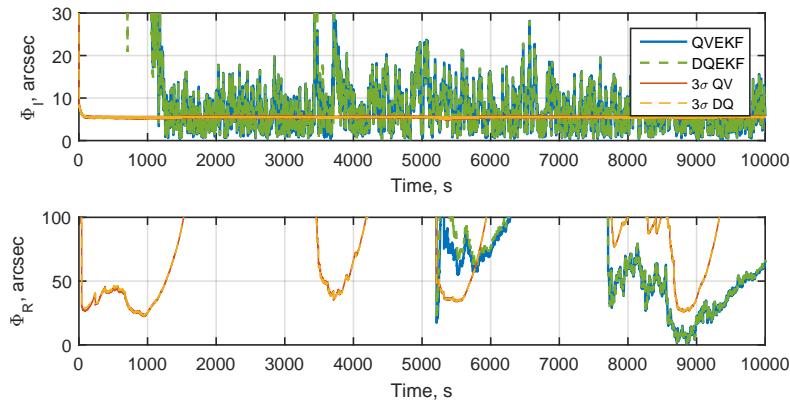


Figure 8.39: KS4 with improved gravity model: inertial and relative attitude errors.

Inertial and relative attitude errors for the improved gravity model are shown in Figure 8.39.

Inertial attitude errors seem to be unaltered, since they are independent on the gravity model used. Relative attitude errors come close to the values as in KS1 case (≈ 20 arcsec) by the very end of the simulation when the number of landmarks increases. The difference between the two filters performance seems to be diminished.

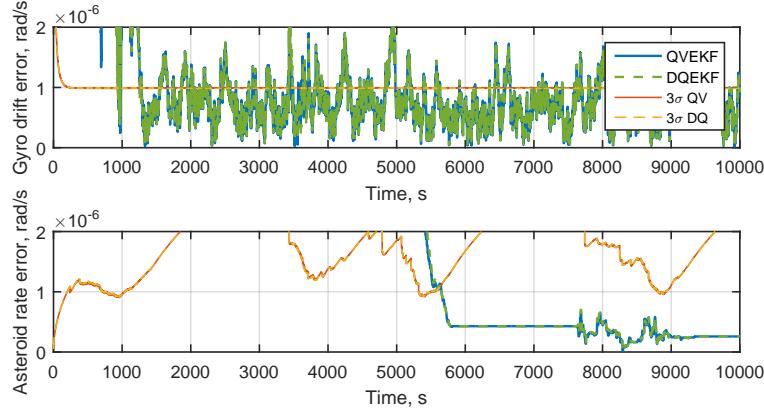


Figure 8.40: KS4 with improved gravity model: gyroscope drift and asteroid rate errors.

Gyroscope drift errors, presented in Figure 8.40, have the same magnitudes as in previous cases. While the asteroid angular rate is estimated with the same accuracy as in the more favourable for navigation KS1 case.

8.6. Time Consumption

The filters in this section are evaluated for the average time it takes them to run 10,000 s simulation. The computational time is also a function of the number of landmarks in the FOV, because the filter then needs to process more data. Therefore, the results are plotted for each scenario and for each asteroid. The simulations were run on an Intel Core i7-2620M processor working at a 2.7 GHz frequency. The results are presented in Figure 8.41.

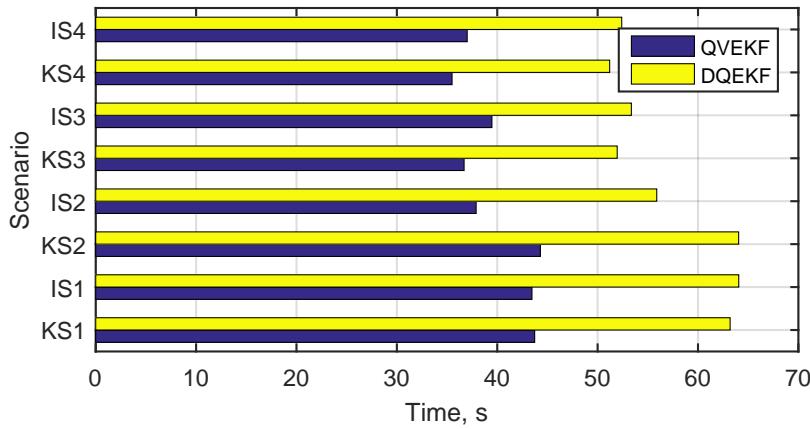


Figure 8.41: Time consumption comparison between the filters.

By looking at Figure 8.41 it is obvious that QVEKF outperforms DQEKF. We see on average a 40 – 50% difference between the two filters. One possible reason for this could be that DQEKF full state vector has more dimensions than QVEKF. Also, it involves more operations in propagation stage and in keeping the orthogonality constrain.

8.7. Conclusions

In this chapter, the results of this thesis work were presented. Eight different cases were simulated and the performance of the dual quaternion and the quaternion-vector filters were analysed. Surprisingly, no large differences in accuracy were detected (identical performance in the steady-state). DQEKF tends to be more accurate in the transient phase, also, tends to converge faster, but suffers from the increased time needed for computations.

9

Conclusions and Recommendations

The thesis was set to answer the research question and its sub-questions, which read as follows:

How to make relative navigation techniques more robust, reliable and autonomous in asteroid missions?

1. Would implementing dual quaternions give better accuracy compared to conventional methods?
2. Which mission scenarios are most benefited from using dual quaternions?
3. Is it advantageous implementing dual quaternions in real-time systems? What is the time consumption of dual quaternion filters?
4. What is the effect of different target asteroids on filter performance?
5. What extra parameters could be estimated and how?

Based on these questions, the main conclusion can be drawn. In the way the research was done, there are no obvious benefits of using dual-quaternions for navigation algorithms around asteroids. The filter provides moderately shorter convergence time and lower errors in the transient phase, while having the same performance in the steady-state, but with a cost of the computational time.

Several other conclusions and recommendations for the future work are presented in the upcoming sections.

9.1. Conclusions

Conclusions of the extensive work done can be separated into the ones regarding the simulator that was developed, and the navigation filters.

There were three novel techniques/methods developed regarding the simulation the dynamics and sensors for missions around asteroids. These are:

- The most widely used equation for the gravity-gradient torque is not accurate in irregular gravity fields, because it takes into account only the central gravity term. The proposed numerical method uses precise gravity field evaluations, thus it can be used universally. Besides, a constant relative difference was discovered, which allows the user to save computational time by evaluating the torque with a lower accuracy and then applying a correction.
- A novel approach for simulating a navigation camera was invented, which gives the means to use more realistic ways to simulate which landmarks are seen by the camera.

- Another novelty is the way the laser-ranger measurements were implemented. Using the property of polyhedron models, it allows one to know the distance from a point in space to any body described as a polyhedron. The body can be any shape and size. The accuracy is achieved with the cost of computational time.

After comparing the performance of the filters that were developed, the following conclusions are drawn:

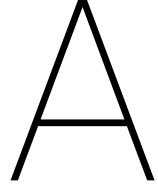
- The dual quaternion filter and the quaternion-vector filter have identical performances in the steady-state, which can be explained by the similarity of the linear-perturbation models and measurement equations between the two filters.
- Another reason for the identical performance in the steady-state could be a high process noise needed for the velocity derivative. Having a high noise makes the propagation part less relevant and the filter relies more on the measurements, which have almost identical measurement equations for both filters.
- The high process noise needed for the filter means that the design model (the gravity field) is very inaccurate, which, indeed, is the case for asteroids. For example, the difference between the polyhedron and central gravity fields at 100 km from Kleopatra can be up to 80%.
- The transient phase performance of the filters is different, with DQEKF being moderately more accurate. The dual quaternion process model has more correlations between the state variables involved, which in an asteroid case, might be more beneficial, since all the states are rather coupled.
- Neither a different asteroid, nor a different orbital scenario have made the steady-state performance of the filters to differ from each other.
- The orbits with high altitude were more beneficial for the navigation system in general, which is a logical outcome. The higher the altitude, the more landmarks in the FOV are detected. Furthermore, the effect of gravitational perturbations decreases with the altitude, thus the gravity field comes closer to the central gravity approximation.
- The convergence time of the dual quaternion filter is lower by ≈ 500 s than the quaternion-vector one in Monte-Carlo simulations.
- The filters are not tuned/designed perfectly, since the variances that the filters estimate, do not match the real ones obtained from the Monte-Carlo simulations.
- The improved gravity field model 'on-board' the spacecraft significantly decreased the errors and made the convergence of the filters possible even in an unfavourable scenario for the navigation system.
- It was also shown that it is possible to estimate some extra parameters (the gyroscope drift and the rate of the asteroid) with the model design used, however, the estimation comes with a cost of the convergence time, since, for example, the asteroid rotational rate is involved in the majority of the dynamics equations in the filters. Therefore, if the filter does not estimate the rate properly, it immediately affects all the other states.
- The computational time for DQEKF is on average 40 – 50% higher than the one of QVEKF partially because of the full state vector being larger by one dimension and also involving more constraints to be followed during the run.

- An explanation for the better performances of DQEKF in transient and in particular for the angular rate of the asteroid might be explained via the differences in the perturbations equations. There are more cross-correlation terms, Observability is enhanced via dynamics for non-measured states.

9.2. Recommendations

Even though the thesis covered an extensive work on modelling the navigation system around an asteroid, there are still many points, where things can be improved or be done in a different way. Therefore, some recommendations are provided as a future work:

- The sensor modelling should include all the misalignment and scaling errors to represent the actual hardware better. However, estimation of these parameters could be impossible, due to the slow dynamics of the spacecraft.
- Since three-dimensional models of asteroids exist that are based on the real images, the image processing could be implemented together with the EKF filters and the dynamics simulator.
- The navigation could be extended to simultaneous localization and mapping, which is a technique for detecting new landmarks and estimating the position of the spacecraft at the same time.
- The laser-ranger model could be further developed to include all the realistic parameters of the surface and the hardware.
- The filters should be tested further with a more precise gravity field model, e.g. spherical harmonic model, to decrease the required process noise.
- It would also be interesting to estimate the gravity-field components in 'real-time' and thus build the gravity model.
- Control forces and torques should be added to increase the stability of the filters (the spacecraft would not loose the asteroid from its line-of-sight).
- The processing of the navigation camera measurements could be more suitable for dual quaternions to exploit their benefits. The technique could be to treat the landmarks not as points, but rather as lines (Plucker lines).
- A full attitude dynamics model could be added to the filter design model, which might be beneficial for DQEKF.



EKF parameters

In this appendix auxiliary data of the extended Kalman filter is presented. Section A.1 gives the gradient matrices that were developed for QVEKF and DQEKF. Section A.2 provides the tuning parameters (covariance matrices) for each case discussed in Results chapter.

A.1. EKF Design Model

A.1.1. QVEKF

The gradient matrix \mathbf{F} is given as:

$$\mathbf{F} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \frac{d\mathbf{f}(X)_{4:6}}{d\mathbf{R}_A} \Big|_{\mathbf{X}=\hat{\mathbf{X}}} & \frac{d\mathbf{f}(X)_{4:6}}{d\mathbf{V}_A} \Big|_{\mathbf{X}=\hat{\mathbf{X}}} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \frac{d\mathbf{f}(X)_{4:6}}{d\omega_{A/I}^A} \Big|_{\mathbf{X}=\hat{\mathbf{X}}} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -[\hat{\boldsymbol{\omega}}_{B/I}^B \times] & \mathbf{0}_{3 \times 3} & -\mathbf{I}_{3 \times 3} & -\mathbf{C}(\hat{\mathbf{q}}_{B/A}) \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -[\hat{\boldsymbol{\omega}}_{B/I}^B \times] & -\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (\text{A.1})$$

Only partial derivatives for the acceleration with respect to the state are presented, since all the other expressions are known. The dynamics function for velocity (acceleration) is given as:

$$\dot{\mathbf{V}}_A = -\frac{GM}{R_A^3} \mathbf{R}_A - 2\boldsymbol{\omega}_{A/I} \times \mathbf{V}_A - \boldsymbol{\omega}_{A/I} \times \boldsymbol{\omega}_{A/I} \times \mathbf{R}_A \quad (\text{A.2})$$

The gradient matrix for the acceleration with respect to the velocity can be expressed as:

$$\frac{\partial \dot{\mathbf{V}}_A}{\partial \mathbf{R}_A} = \begin{bmatrix} \frac{\partial \dot{V}_x}{\partial R_x} & \frac{\partial \dot{V}_x}{\partial R_y} & \frac{\partial \dot{V}_x}{\partial R_z} \\ \frac{\partial \dot{V}_y}{\partial R_x} & \frac{\partial \dot{V}_y}{\partial R_y} & \frac{\partial \dot{V}_y}{\partial R_z} \\ \frac{\partial \dot{V}_z}{\partial R_x} & \frac{\partial \dot{V}_z}{\partial R_y} & \frac{\partial \dot{V}_z}{\partial R_z} \end{bmatrix} \quad (\text{A.3})$$

All the components are presented below.

$$\frac{\partial \dot{V}_x}{\partial R_x} = \omega_y^2 - \frac{\mu}{(X^2 + Y^2 + Z^2)^{3/2}} + \omega_z^2 + \frac{3\mu X^2}{(X^2 + Y^2 + Z^2)^{5/2}} \quad (\text{A.4})$$

$$\frac{\partial \dot{V}_x}{\partial R_y} = \frac{3\mu XY}{(X^2 + Y^2 + Z^2)^{5/2}} - \omega_x \omega_y \quad (\text{A.5})$$

$$\frac{\partial \dot{V}_x}{\partial R_z} = \frac{3\mu XZ}{(X^2 + Y^2 + Z^2)^{5/2}} - \omega_x \omega_z \quad (\text{A.6})$$

$$\frac{\partial \dot{V}_y}{\partial R_x} = \frac{3\mu XY}{(X^2 + Y^2 + Z^2)^{5/2}} - \omega_x \omega_z \quad (\text{A.7})$$

$$\frac{\partial \dot{V}_y}{\partial R_y} = \omega_x^2 - \frac{\mu}{(X^2 + Y^2 + Z^2)^{3/2}} + \omega_z^2 + \frac{3\mu Y^2}{(X^2 + Y^2 + Z^2)^{5/2}} \quad (\text{A.8})$$

$$\frac{\partial \dot{V}_y}{\partial R_z} = \frac{3\mu YZ}{(X^2 + Y^2 + Z^2)^{5/2}} - \omega_y \omega_z \quad (\text{A.9})$$

$$\frac{\partial \dot{V}_z}{\partial R_x} = \frac{3\mu XZ}{(X^2 + Y^2 + Z^2)^{5/2}} - \omega_x \omega_z \quad (\text{A.10})$$

$$\frac{\partial \dot{V}_z}{\partial R_y} = \frac{3\mu YZ}{(X^2 + Y^2 + Z^2)^{5/2}} - \omega_y \omega_z \quad (\text{A.11})$$

$$\frac{\partial \dot{V}_z}{\partial R_z} = \omega_x^2 - \frac{\mu}{(X^2 + Y^2 + Z^2)^{3/2}} + \omega_y^2 + \frac{3\mu Z^2}{(X^2 + Y^2 + Z^2)^{5/2}} \quad (\text{A.12})$$

where X , Y and Z are the components of position vector \mathbf{R}_A . The Jacobian of acceleration with respect to the velocity and the asteroid angular rate are as follows:

$$\frac{\partial \dot{\mathbf{V}}_A}{\partial \mathbf{V}_A} = \begin{bmatrix} 0 & 2\omega_z & -2\omega_y \\ -2\omega_z & 0 & 2\omega_x \\ 2\omega_y & -2\omega_x & 0 \end{bmatrix} \quad (\text{A.13})$$

$$\frac{\partial \dot{\mathbf{V}}_A}{\partial \boldsymbol{\omega}_{A/I}^A} = \begin{bmatrix} -\omega_y Y - \omega_z Z & 2\omega_y X - 2V_z - \omega_x Y & 2V_y + 2\omega_z X - \omega_x Z \\ 2V_z - \omega_y X + 2\omega_x Y & -\omega_x X - \omega_z Z & 2\omega_z Y - 2V_x - \omega_y Z \\ 2\omega_x Z - \omega_z X - 2V_y & 2V_x - \omega_z Y + 2\omega_y Z & -\omega_x X - \omega_y Y \end{bmatrix} \quad (\text{A.14})$$

The measurement function of NAVCAM was given as:

$$\mathbf{z}_{\text{NAVCAM}} = \mathbf{h}(\mathbf{X}) = \mathbf{C}(\mathbf{q}_{B/A}) \frac{\mathbf{R}_{LM,i} - \mathbf{R}_A}{\|\mathbf{R}_{LM,i} - \mathbf{R}_A\|} \quad (\text{A.15})$$

It was shown that it is linear with respect to the attitude error, so only the Jacobian for the position vector has to be found. First of all, the attitude matrix is neglected and the gradient matrix is calculated for the function $\mathbf{h}_A = \frac{\mathbf{R}_{LM,i} - \mathbf{R}_A}{\|\mathbf{R}_{LM,i} - \mathbf{R}_A\|}$.

$$\frac{\partial \mathbf{h}_A}{\partial \mathbf{R}_A} = \begin{bmatrix} \frac{\partial h_x}{\partial R_x} & \frac{\partial h_x}{\partial R_y} & \frac{\partial h_x}{\partial R_z} \\ \frac{\partial h_y}{\partial R_x} & \frac{\partial h_y}{\partial R_y} & \frac{\partial h_y}{\partial R_z} \\ \frac{\partial h_z}{\partial R_x} & \frac{\partial h_z}{\partial R_y} & \frac{\partial h_z}{\partial R_z} \end{bmatrix} \quad (\text{A.16})$$

where:

$$\frac{\partial h_x}{\partial R_x} = -\frac{LM_y^2 - 2LM_y Y + LM_z^2 - 2LM_z Z + Y^2 + Z^2}{((LM_x - X)^2 + (LM_y - Y)^2 + (LM_z - Z)^2)^{3/2}} \quad (\text{A.17})$$

$$\frac{\partial h_x}{\partial R_y} = \frac{(LM_x - X)(LM_y - Y)}{((LM_x - X)^2 + (LM_y - Y)^2 + (LM_z - Z)^2)^{3/2}} \quad (\text{A.18})$$

$$\frac{\partial h_x}{\partial R_z} = \frac{(LM_x - X)(LM_z - Z)}{((LM_x - X)^2 + (LM_y - Y)^2 + (LM_z - Z)^2)^{3/2}} \quad (\text{A.19})$$

$$\frac{\partial h_y}{\partial R_x} = \frac{(LM_x - X)(LM_y - Y)}{((LM_x - X)^2 + (LM_y - Y)^2 + (LM_z - Z)^2)^{3/2}} \quad (\text{A.20})$$

$$\frac{\partial h_y}{\partial R_y} = -\frac{LM_x^2 - 2LM_x X + LM_z^2 - 2LM_z Z + X^2 + Z^2}{((LM_x - X)^2 + (LM_y - Y)^2 + (LM_z - Z)^2)^{3/2}} \quad (\text{A.21})$$

$$\frac{\partial h_y}{\partial R_z} = \frac{(LM_y - Y)(LM_z - Z)}{((LM_x - X)^2 + (LM_y - Y)^2 + (LM_z - Z)^2)^{3/2}} \quad (\text{A.22})$$

$$\frac{\partial h_z}{\partial R_x} = \frac{(LM_y - Y)(LM_z - Z)}{((LM_x - X)^2 + (LM_y - Y)^2 + (LM_z - Z)^2)^{3/2}} \quad (\text{A.23})$$

$$\frac{\partial h_z}{\partial R_y} = \frac{(LM_x - X)(LM_z - Z)}{((LM_x - X)^2 + (LM_y - Y)^2 + (LM_z - Z)^2)^{3/2}} \quad (\text{A.24})$$

$$\frac{\partial h_z}{\partial R_z} = -\frac{LM_x^2 - 2LM_x X + LM_y^2 - 2LM_y Y + Y^2 + Z^2}{((LM_x - X)^2 + (LM_y - Y)^2 + (LM_z - Z)^2)^{3/2}} \quad (\text{A.25})$$

where LM_x , LM_y and LM_z are the coordinates of a landmark. The full Jacobian matrix for the NAV-CAM measurements with respect to the position vector is then:

$$\mathbf{H}_{NAV\text{CAM},R} = \mathbf{C}(\mathbf{q}_{B/A}) \frac{\partial \mathbf{h}_A}{\partial \mathbf{R}_A} \quad (\text{A.26})$$

Finally, the laser ranger measurement function was given as:

$$z_{LR} = h(\mathbf{X}) = \|\mathbf{R}_{LM,i} - \mathbf{R}_A\| \quad (\text{A.27})$$

Then, the measurement matrix (gradient with respect to the position vector) is as follows:

$$\mathbf{H}_{LR,R} = \left[\begin{array}{ccc} -\frac{LM_x - X}{d} & -\frac{LM_y - Y}{d} & -\frac{LM_z - Z}{d} \end{array} \right] \quad (\text{A.28})$$

where d is the distance to the landmark:

$$d = \sqrt{(LM_x - X)^2 + (LM_y - Y)^2 + (LM_z - Z)^2} \quad (\text{A.29})$$

A.1.2. DQEKF

For DQEKF all the matrices are exactly the same, only the ones with respect to the position vector have to be post multiplied with a DCM, since the position error in DQEKF is expressed in B frame.

A.2. EKF Tuning

In this section the tuning parameters for the filters are presented. QVEKF and DQEKF were run with identical set-up, thus only the matrices for QVEKF are presented. The variables in the matrices for DQEKF have to be changed accordingly to the state vector of DQEKF.

KS1

State-error covariance matrix is initialized as follows:

$$\mathbf{P}_{0/0} = \left[\begin{array}{cccccc} 10^6 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 10^{-5} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 10^{-5} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 10^{-12} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & 10^{-18} \mathbf{I}_{3 \times 3} \end{array} \right] \quad (\text{A.30})$$

The process noise covariance matrix after tuning was set to:

$$\mathbf{Q} = \begin{bmatrix} \sigma_g^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \sigma_v^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \sigma_u^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 10^{-16} \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (\text{A.31})$$

where $\sigma_g = 0.008 \text{ m/s}^2$ is the uncertainty in the gravity field, $\sigma_v = 5.8 \times 10^{-7} \text{ rad/s}^{1/2}$ and $\sigma_u = 5.8 \times 10^{-8} \text{ rad/s}^{3/2}$ are the standard deviations of gyroscope noises. Some of the values in the error and process noise covariance matrices are set really small (close to zero, e.g., 10^{-18}), because it proved to give the best performance. Increasing them, could result in a divergence of the filter. The star tracker has the measurement covariance matrix:

$$\mathbf{R}_{ST} = 0.1 \times \begin{bmatrix} \frac{\sigma_x^2}{8} & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} \quad (\text{A.32})$$

where $\sigma_x = 45 \text{ arcsec}$ and $\sigma_y = \sigma_z = 5 \text{ arcsec}$ are standard deviations of the star tracker noise. The X-axis is the boresight axis. The covariance matrix of the navigation camera is:

$$\mathbf{R}_{NAVC} = 40 \times \begin{bmatrix} (p\sigma_n)^2 & 0 & 0 \\ 0 & (p\sigma_n)^2 & 0 \\ 0 & 0 & (p\sigma_n)^2 / 900 \end{bmatrix} \quad (\text{A.33})$$

where p is the pixel size and $\sigma_n = 0.1$ is the standard deviation of navigation camera errors (expressed in pixels). The Z-axis has a far smaller variance set, because the focal length is constant and the noise comes only from the norm of the vector. Finally, the laser-ranger covariance has to be defined. The laser ranger error is hard to model statistically, because it depends on various parameters and the state itself. However, by inspecting how the error depends on the angle between the laser beam and the incident surface normal, see Figure A.1, we could derive a few phases.

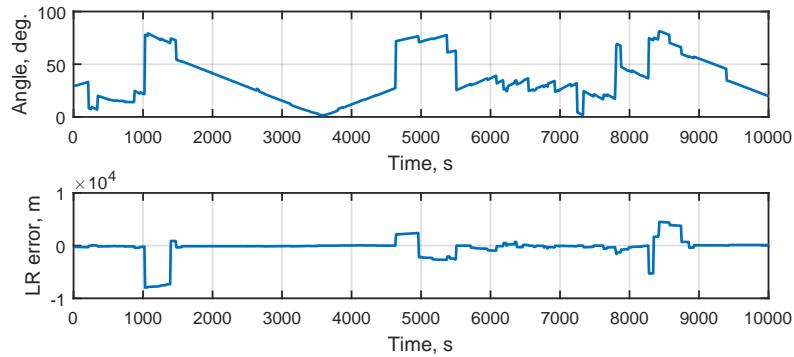


Figure A.1: Laser ranger errors and the angle between the surface normal and the laser beam

In Figure A.1 we see that the error increases up to a few kilometres when the angle is higher than 50 deg. Furthermore, when the angle is below 20 deg the errors are much smaller. According these observations, we define three phases. The first one is when the angle is below 20 deg, the second one is when the angle is between 20 and 50 deg, and the third one is when the angle is above 50. We will change the LR variance accordingly these phases. The values are presented below:

$$R_{LR1} = 2500 \text{ m}^2, \quad R_{LR2} = 500,000 \text{ m}^2 \text{ and} \quad R_{LR3} = 25,000,000 \text{ m}^2 \quad (\text{A.34})$$

IS1

The main difference for IS1 case is the process noise σ_g value, which is set to $0.5 \times 10^{-5} \text{ m/s}^2$, since the asteroid is much smaller. The error covariance matrix is also changed:

$$\mathbf{P}_{0/0} = \begin{bmatrix} 10^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & 10^{-2} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 10^{-5} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 10^{-5} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 10^{-12} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & 10^{-14} \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (\text{A.35})$$

All the other matrices are the same as in KS1 case, only the LR covariance matrix for the three phases is changed:

$$R_{LR1} = 20 \text{ m}^2, R_{LR2} = 100 \text{ m}^2 \text{ and } R_{LR3} = 1000 \text{ m}^2 \quad (\text{A.36})$$

KS2

All the parameters are the same as in KS1 case.

IS2

All the parameters are the same as in IS1 case.

KS3

For KS3 only the process covariance matrix and the laser ranger covariance is changed. The process covariance matrix is set to:

$$\mathbf{Q} = \begin{bmatrix} \sigma_g^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \sigma_v^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \sigma_u^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 5 \cdot 10^{-16} \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (\text{A.37})$$

where $\sigma_g = 0.03 \text{ m/s}^2$, thus it is much bigger than in KS1 case, because the asteroid synchronous orbit does not match the one calculated by the central-field model, and thus the error is much larger. The laser ranger errors are also much smaller, because the spacecraft is closer to the surface.

$$R_{LR1} = 50 \text{ m}^2, R_{LR2} = 100 \text{ m}^2 \text{ and } R_{LR3} = 500 \text{ m}^2 \quad (\text{A.38})$$

IS3

IS3 parameters are the same as in IS1 case.

KS4

KS4 has the process noise covariance matrix set to:

$$\mathbf{Q} = \begin{bmatrix} \sigma_g^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \sigma_v^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \sigma_u^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 2 \cdot 10^{-16} \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (\text{A.39})$$

with $\sigma_g = 0.02 \text{ m/s}^2$. The laser ranger error variances are set to:

$$R_{LR1} = 200 \text{ m}^2, R_{LR2} = 10,000 \text{ m}^2 \text{ and } R_{LR3} = 1,000,000 \text{ m}^2 \quad (\text{A.40})$$

Other parameters are the same as in KS1 case.

IS4

For IS4 case, the process covariance matrix is:

$$\mathbf{Q} = \begin{bmatrix} \sigma_g^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \sigma_v^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \sigma_u^2 \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 0.9 \cdot 10^{-16} \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (\text{A.41})$$

where $\sigma_g = 2 \times 10^{-5}$ m/s². The laser ranger variances are:

$$R_{LR1} = 10 \text{ m}^2, \quad R_{LR2} = 40 \text{ m}^2 \text{ and } R_{LR3} = 250 \text{ m}^2 \quad (\text{A.42})$$

Bibliography

- Altman, S. L., *Rotations, Quaternions and Double Groups*, Clarendon Press, Oxford, 1986.
- Badescu, V., *Asteroids: Prospective Energy and Material Resources*, Springer, Heidelberg, 2013.
- Buemi, M., Landi A., and Procopio, D., "Autonomous Star tracker for Rosetta", 4th International ESA Conference on Guidance, Navigation and Control Systems, ESTEC, Noordwijk, The Netherlands, 1999.
- Cheng, A. F., "Near Earth Asteroid Rendezvous: Mission Summary", *Asteroid III*, University of Arizona Press, Tucson, 2002, pp. 351-366.
- Clifford, W. K., "Preliminary sketch of biquaternions", *Proceedings of the London Mathematical Society*, Vol. 4, 1873, pp. 381-195.
- Connors, M., Wiegert, P., and Veillet C., "Earth's Trojan asteroid", *Nature*, Vol. 475, Issue 7357, 2011, pp. 481-483.
- Conway, D., Macomber, B., Cavalieri, K. A. and Junkins, J. L., "Vision-Based Relative Navigation Filter for Asteroid Rendezvous", AAS 14-015, *AAS Guidance, Navigation, and Control Conference*, Breckenridge, CO, February 2014.
- Dionne, K., "Improving Autonomous Navigation for Small Body Exploration Using Range Measurements", AIAA 2009-6106, *AIAA Guidance, Navigation, and Control Conference*, 2009.
- D'Urso, M. G., "Analytical computation of gravity effects for polyhedral bodies", *Journal of Geodesy*, Vol. 88, Issue 1, 2013, pp. 13-29.
- Dymock, R. *Asteroids and Dwarf Planets and How to Observe Them*, Vol. 1, Springer-Verlag, New York, 2010.
- Filipe, N., *Nonlinear Pose Control and Estimation for Space Proximity Operations: An Approach Based on Dual Quaternions*, PhD thesis, Georgia Institute of Technology, 2014.
- Fischer, I. S., *Dual-number methods in kinematics, statics, and dynamics*, Boca Raton, CRC Press, 1999.
- Fraiture, L., "A history of the description of the three-dimensional finite rotation", *Advances in the Astronautical Sciences*, Vol. 132 (June), 2008, pp. 835-857.
- Gaskell, R. et al, "Landmark navigation studies and target characterization in the Hayabusa encounter with Itokawa", 2006-6660, *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Lake Tahoe, California, 2006.
- Goddard, J., *Pose and Motion Estimation From Vision Using Dual Quaternion-Based Extended Kalman Filtering*, PhD thesis, Knoxville, 1997.
- Harris, A. W., and Pravec, P., "Rotational properties of asteroids , comets and TNOs", *Proc. IAU Symp. Asteroids, Comets, Meteors*, Vol. 229, Cambridge University Press, Cambridge, 2006, pp. 439-447.

- Holsapple, K., "Equilibrium Configurations of Solid Cohesionless Bodies", *Icarus*, Vol. 52, Issue 2, 2001, pp. 432-448.
- Jia, Y. B., Dual quaternion, Com S 477/577, Fall 2015, Course handouts, 2013. URL <http://web.cs.iastate.edu/~cs577/>.
- Jiang, Y., "Equilibrium points and periodic orbits in the vicinity of asteroids with an application to 216 Kleopatra", *Earth Moon Planets*, Vol. 115(1–4), 2015, pp. 31–44.
- Julier, S., and Uhlmann, J., "A New Extension of the Kalman Filter to Nonlinear Systems," *Proc. SPIE*, Vol. 3068, 1997, pp. 182-193.
- Kawaguchi, J., Fujiwara, A., Uesugi, T., "Hayabusa - its technology and science accomplishment summary and Hayabusa-2", *Acta Astronautica*, Vol. 62, Issue 10-11, 2008, pp. 639-647.
- Kim, S. G., Crassidis, J. L., Cheng, Y. and Fosbury, A. M. , "Kalman Filtering for Relative Spacecraft Attitude and Position Estimation", *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 1, 2007, pp. 133-143.
- Kubota, T., Hashimoto, T., Kawaguchi, J., Uo, M., and Shirakawa, K. "Guidance and navigation of Hayabusa spacecraft for asteroid exploration and sample return mission", *SICE-ICASE, 2006. International Joint Conference*, 2007, pp. 2793–2796.
- Markley, F. L., "Attitude Error Representations for Kalman filtering", *Journal of Guidance, Control and Dynamics*, Vol. 26, No. 2, 2003.
- Markley, F.L. and Crassidis, J.L., *Fundamentals of Spacecraft Attitude Determination and Control*, Vol 2, Springer, 2013.
- Montenbruck, O., Gill, E., *Satellite Orbits: Models, Methods and Applications*, Springer Verlag, Heidelberg, 2001.
- Munoz P. et al, "Preparations and Strategy for Navigation during Rosetta Comet Phase", *Proceedings 23rd International Symposium on Space Flight Dynamics - 23rd ISSFD*, Pasadena, USA, 2012.
- Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D., "Shape distributions", *ACM Transactions on Graphics*, Vol. 21, No. 4, 2002, pp. 807-832.
- Pardo, R. de Santayana and Lauer, M., "Optical measurements for Rosetta navigation near the comet", *Proceedings 25th International Symposium on Space Flight Dynamics – 25th ISSFD*, 2015.
- Richner, M., Gebauer, G., and Leissle, T., "Coordinate Systems for Rosetta", RO-DSS-TN-1081, Technical report, Issue 6d, ESA, 2003.
- Takeishi, N., "Automatic Landmark Recognition for Asteroid by Image Features", *29th International Symposium on Space Technology and Science (ISTS)*, Nagoya, Japan, 2013.
- Tanga, P., Hestroffer, D., Delbo, M., Richardson, D. C., "Asteroid rotation and shapes from numerical simulations of gravitational re-accumulation", *Planetary and Space Science*, 57(2), 2009, pp. 193-200.
- Tsoulis, D., "Analytical computation of the full gravity tensor of a homogeneous arbitrarily shaped polyhedral source using line integrals", *Geophysics*, 77(2), 2012.

- Virgili, B., Lemmens, S., and Krag, H., Investigation on Envisat Attitude Motion, *e.Deorbit Workshop*, European Space Agency, The Netherlands, 2014.
- Wakker, K. F., *Astrodynamic I*, AE4874 I course, TU Delft, The Netherlands, 2010.
- Werner, R. and Scheeres, D.J., "Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 Castalia", *Celestial Mechanics and Dynamical Astronomy*, Vol. 65, 1997, pp. 313-344.
- Wertz, J. R., Everett, D. F., and Puschell, J. J., *Space mission engineering: The new SMAD*, Hawthorne, CA: Microcosm Press, 2011.
- Wie, B., *Space Vehicle Dynamics and Control*, Second Edition, American Institute of Aeronautics and Astronautics, Inc., 2006.
- Williams, B. G., "Technical challenges and results for navigation of NEAR Shoemaker", *Johns Hopkins APL Technical Digest*, Vol. 23(1), 2002, pp. 34-45.
- Yang, H., Y., Vetrisano, M., Vasile, M. and Zhang, W., "Autonomous navigation of spacecraft formation in the proximity of minor bodies", *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 230, No. 1, 2015, pp. 189-204.
- Zarchan, P., and Musoff, H., *Fundamentals of Kalman Filtering — A Practical Approach*, Second Edition, American Institute of Aeronautics and Astronautics, 2009.
- Zuiani, F., Castellini, F., Kielbassa, S., Bielsa, C. and Garcia, J.M., "Rosetta command and monitoring operations for Philae landing", *Proceedings 25th International Symposium on Space Flight Dynamics – 25th ISSFD*, 2015.