

Asteroid Mission Guidance and Control using Dual Quaternions

Master Thesis

Marc Trullas Ballester

Technische Universiteit Delft



Image credit: ESA public domain

ASTEROID MISSION GUIDANCE AND CONTROL USING DUAL QUATERNIONS

MASTER THESIS

by

Marc Trullas Ballester

in partial fulfillment of the requirements for the degree of

Master of Science

in Aerospace Engineering - Space Flight

at Delft University of Technology,

to be defended publicly on Tuesday April 17, 2018 at 4 PM.

Student number: 4453166
Project duration: April 1, 2017 – April 17, 2018
Supervisors: Prof. dr. ir. J. McMahon, CU Boulder
Dr. ir. E. Mooij, TU Delft
Thesis committee: Dr. ir. P. Chu, TU Delft
Prof. L. L. A. Vermeersen, TU Delft (Chair)

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

PREFACE

Asteroids, comets and space exploration is such an exciting topic and I am glad I had the opportunity to do some research about it and learn so much. In this thesis, some advantages and disadvantages have been found for dual quaternions applied to problems in asteroid close-proximity operations. Hopefully, this work will contribute to the development of new space exploration missions and inspire future research.

This report is the culmination of my research carried out in order to fulfill 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 and presented in 2018.

This thesis research has been carried out in Boulder (CO), under the supervision of Prof. dr. ir. J. McMahon from the University of Colorado Boulder and Dr. ir. E. Mooij from Delft University of Technology. I would like to express my special and sincere gratitude to both of them for their valuable advice, patience and comprehension during all this work. This thesis would have not been the same without you.

Last, I would also like to thank my parents and Lena for all their unconditional support. Thank you very much.

Marc Trullas Ballester

Barcelona, April 2018

ABSTRACT

The investigation of small bodies in the Solar System such as asteroids and comets is of increasing interest to the space exploration community for scientific and security reasons. A key enabling technology is the availability to control position and attitude of spacecraft for asteroid close-proximity operations. Dynamics and control issues related to these operations have been investigated addressing the orbital and attitude motion of the spacecraft with no coupling between them assumed. In reality however, the translational and rotational dynamics of spacecraft are generally coupled. It is therefore desirable to simultaneously take into account the full six degrees of freedom problem when developing pose (position and orientation) control laws for practical applications. One way of doing that is by expressing the state variables with dual quaternions, a mathematical formulation that naturally couples both motions in an efficient and compact way.

The goal of this thesis is to determine what advantages and disadvantages can a six degrees of freedom controller based on dual quaternions bring over classical methods regarding asteroid close-proximity operations. To do so, a simulator was developed in Matlab R2016b with the two different representations: the dual quaternion one and the classical one using vectors for position, velocity and angular velocity, and quaternions for the attitude. Then, a controller was designed and optimized for each representation to follow a reference trajectory in three different scenarios, such as orbiting, hovering and landing on the asteroid Bennu. The dual quaternion representation has only two equations of motion instead of the four required for the classical one because of its compactness and natural way of coupling rotational and translational motions. However, results show that the dual quaternion controller takes 15% more computational time to achieve the same level of accuracy and energy consumption as the classical controller for the nominal scenarios. The reason for that is because the total number of operations involved in the integration of the equations of motions is higher. Moreover, a stability analysis was carried out and showed that controllers based on dual quaternions can handle much larger perturbations than the classical ones. Specifically, between 23 and 227 times larger perturbations depending on the scenario. Apart from that, results also proved that dual quaternion controllers can correct trajectory perturbations in a much faster and efficient way than the classical ones. Specifically, they have a settling time between 29% and 92% shorter, accumulate between 2.6 and 19 times less error and consume between 3.2 and 9.3 times less energy, depending on the scenario.

CONTENTS

Acronyms	xi
List of symbols	xiii
1 Introduction	1
1.1 Research objective	3
1.2 Report outline.	4
2 Mission heritage	5
2.1 Asteroid missions	5
2.2 Attitude and position representations.	8
2.3 Mission and vehicle description	10
2.3.1 Mission definition	11
2.3.2 Spacecraft model	13
2.4 Mission and systems requirements	15
3 Quaternions and dual quaternions	17
3.1 Quaternions	17
3.1.1 Attitude representation	18
3.2 Dual quaternions	20
3.2.1 Attitude and position representation.	23
3.2.2 Coordinate transformations	23
4 Flight dynamics and space environment	25
4.1 Reference frames	25
4.1.1 Asteroid-centered inertial	25
4.1.2 Spacecraft body-fixed	26
4.1.3 Asteroid-centered asteroid-fixed.	26
4.2 Equations of motion	26
4.2.1 Classical formulation	27
4.2.2 Dual quaternion formulation	28

4.3	Force models	28
4.3.1	Gravity force	29
4.3.2	Solar radiation force	31
4.3.3	Solar tides (3rd body force).	32
4.4	Gravity gradient torque model	33
5	Spacecraft control	35
5.1	Classic control	36
5.1.1	Attitude control	36
5.1.2	Position control	37
5.2	Dual quaternion control	38
5.3	Gains optimization	39
5.3.1	Cost function	40
5.3.2	Selected gains	43
6	Numerical methods	47
6.1	Integration method	47
6.2	Root finding.	49
6.3	Interpolation	49
7	Simulator design	51
7.1	Architecture design	51
7.1.1	Guidance	52
7.1.2	Control.	53
7.1.3	Actuators.	53
7.1.4	Sensors	53
7.1.5	Navigation	53
7.2	Verification and pseudo-validation	54
7.2.1	Navigation	55
7.2.2	Guidance	67
7.2.3	Control, actuators and sensors.	69
7.2.4	System test.	70
8	Results	75
8.1	Kinematic propagation	75
8.2	Two body problem	76
8.3	Nominal scenarios	79
8.3.1	Orbiting	79
8.3.2	Hovering.	83

8.3.3	Landing	86
8.4	Frequency analysis	90
8.5	Stability analysis	91
8.5.1	Energy	92
8.5.2	Errors	94
8.5.3	Settling time	95
8.5.4	Computational time	97
9	Conclusions and recommendations	99
9.1	Conclusions.	99
9.2	Recommendations and future work.	100
	Appendix A: Stability theory	103
	Appendix B: Other system test results	107
	Bibliography	119

ACRONYMS

ACAF	Asteroid-Centered Asteroid-Fixed
ACI	Asteroid-Centered Inertial
AU	Astronomical Unit
DCM	Direction Cosine Matrix
DOF	Degrees of Freedom
DQ-EKF	Dual Quaternion Extended Kalman Filter
DQ-MEKF	Dual Quaternion Multiplicative Extended Kalman Filter
ESA	European Space Agency
FTC	Finite-Time Control
GNC	Guidance, Navigation and Control
ICRF	International Celestial Reference Frame
IMU	Inertial Measurement Unit
JAXA	Japanese Space Agency
JPL	Jet Propulsion Laboratory
LEO	Low Earth Orbit
Lidar	Light Detection and Ranging
LRF	Laser Range Finder
MRP	Modified Rodrigues Parameters
NASA	National Aeronautics and Space Administration
NEAR	Near Earth Asteroid Rendezvous

NEO	Near Earth Object
OSIRIS-REx	Origins, Spectral Interpretation, Resource Identification, Security, Regolith Explorer
PD	Proportional and Derivative
PID	Proportional, Integral and Derivative
RDCM	Reduced Direction Cosine Matrix
RP	Rodrigues Parameters
SRP	Solar Radiation Pressure

LIST OF SYMBOLS

CALLIGRAPHIC SYMBOLS

Symbol	Description	Unit
\mathcal{H}	Set of quaternions	-
\mathcal{H}_d	Set of dual quaternions	-
\mathcal{H}^r	Set of scalar quaternions	-
\mathcal{H}^v	Set of vector quaternions	-
\mathcal{H}_d^r	Set of dual quaternions with zero dual part	-
\mathcal{H}_d^v	Set of dual vector quaternions	-
\mathbb{R}	Real domain	-

GREEK SYMBOLS

Symbol	Description	Unit
α_e	Reference equatorial radius	m
Δ	Increment	-
ϵ	Dual unit	-
λ	Longitude in spherical coordinates	rad
μ	Gravitational parameter	m^3/s^2
σ	Standard deviation	-
σ_{SB}	Stefan-Boltzmann constant	$\text{W}/\text{m}^2/\text{K}^4$
$\boldsymbol{\tau}$	Torque vector	N·m
Φ	Angle of the quaternion; Euler parameter	rad
Φ	Increment function for integration	-
ω	Angular velocity	rad/s
$\hat{\omega}$	Dual velocity	m/s, rad/s
Ω	Angular velocity	rad/s
∇	Gradient operator	-
∇^2	Laplace operator	-

OPERATORS

Symbol	Description	Unit
\circ	Dual quaternion circle operation	-
$*$	Conjugation operation	-
s	Swap operation	-
\star	Matrix dual quaternion multiplication	-
$*$	Matrix quaternion multiplication	-

ROMAN SYMBOLS

Symbol	Description	Unit
\mathbf{a}	Acceleration vector	m/s ²
$\hat{\mathbf{a}}, \hat{\mathbf{b}}$	Example dual quaternion	-
A_{sat}	Cross-sectional area of satellite	m ²
C_R	Reflectivity coefficient	-
$\bar{C}_{n,m}, \bar{S}_{n,m}$	Spherical harmonic normalized coefficients	-
$C_{n,m}, S_{n,m}$	Spherical harmonic unnormalized coefficient	-
\mathbf{C}	Direction cosine matrix	-
\mathbf{d}	Position vector from the Sun to the asteroid	m
$\hat{\mathbf{d}}$	Dual disturbance force	N, N·m
\mathbf{e}	Rotation vector of a quaternion; Euler vector parameter	-
$\hat{\mathbf{f}}$	Dual force	N, N·m
\mathbf{F}	Force vector	N
\mathbf{G}	Quaternion rotation matrix	-
h	Step size	s
\mathbf{I}	Inertia tensor	kg·m ²
\mathbf{L}	External torque vector	N·m
M	Total mass of a body	kg
\mathbf{M}	Dual inertia matrix	kg, kg·m ²
$\bar{P}_{n,m}$	Normalized Legendre polynomial of order n and degree m	-
\mathbf{q}	Quaternion	-
$\hat{\mathbf{q}}$	Dual quaternion	-
\mathbf{q}_r	Real part of a dual quaternion	-
\mathbf{q}_d	Dual part of a dual quaternion	-
\mathbf{r}	Position vector	m

Symbol	Description	Unit
\mathbf{u}	Attitude control input	N·m
$\hat{\mathbf{u}}$	Dual control input	N, N·m
U	Gravitational potential	J
\mathbf{v}	Velocity vector	m/s
X, Y, Z	Reference frame X, Y, Z Axes	-
$\hat{\mathbf{1}}$	Dual quaternion $\mathbf{1} + \epsilon\mathbf{0}$	-
$\hat{\mathbf{0}}$	Dual quaternion $\mathbf{0} + \epsilon\mathbf{0}$	-
$\mathbf{0}$	Quaternion (0, 0, 0, 0)	-
$\mathbf{1}$	Quaternion (1, 0, 0, 0)	-

1

INTRODUCTION

The investigation of small bodies in the Solar System, such as asteroids and comets, is of increasing interest to the space-exploration community for scientific and possibly safety reasons [Vukovich and Gui, 2017]. They are remnants from the formation of the Solar System and may contain the molecular precursors to the origin of life and the Earth's oceans. Some of them may also become hazardous objects with a probability of impacting the Earth one day [Lissauer and de Pater, 2013]. For these reasons, several missions have been dedicated to study asteroids and comets; such as [Near Earth Asteroid Rendezvous \(NEAR\)](#), Hayabusa, Roseta or Origins, Spectral Interpretation, Resource Identification, Security, Regolith Explorer ([OSIRIS-REx](#)).

One of the key challenges and enabling technology is the availability to control position and attitude of spacecraft for close proximity operations, such as orbiting, hovering or landing on the surface of asteroids. During the past two decades, dynamics and control issues related to these operations have been investigated [Broschart and Scheeres, 2005; Scheeres, 2012; Lee et al., 2015; Kumar, 2008]. The majority of these studies however, addressed the orbital and attitude motion of the spacecraft about the asteroid with no coupling between them assumed; and this is the key for high precision modeling and control. In reality, the translational and rotational dynamics of spacecraft are generally coupled. Since the direction of the thrust force is dependent on the spacecraft orientation, attitude motion can deflect an imposed control force and affect translational motion in this way. In addition, some external torques, such as gravity gradient, are also functions of the spacecraft position [Wang and Xu, 2014]. It is therefore desirable to simultaneously take into account the full six [Degrees of Freedom \(DOF\)](#) coupling when developing pose (position and orientation) control laws for practical applications.

The coupled full six [DOF](#) problem has been extensively studied for Earth-orbiting spacecraft in applications, such as: formation flying, spacecraft rendezvous and docking, space debris removal and many others. However, the corresponding problem for small asteroids is still open. The dynamics environment is quite

different from that of the Earth and poses special challenges for spacecraft-motion control. The gravity fields of asteroids are weak and can be time varying due to high speed rotations and their small size, irregular shape and mass distribution. When the dimensions of the spacecraft is comparable to the asteroid, the gravitational coupling between translational and rotational motion plays an important role and therefore, it has to be taken into account for close proximity operations [Wang et al., 2012; Misra and Sanyal, 2015].

One way of taking the full six DOF coupling into account is by using dual quaternions. A mathematical formulation that naturally couples the translational and rotational motion. They provide an efficient representation of the two motions simultaneously and perform more compact and effectively than other formulations [Wang et al., 2010]. Dual quaternions are an extension of classical quaternions. Therefore, they allow to write the combined translational and rotational equations of motion in the same form as rotational-only equations written in terms of quaternions. This appealing property can be used to design controllers by extending already existing attitude-only controllers with some desirable properties by often simply substituting quaternions with dual quaternions. In addition, they also allow for the use of a single error function to represent the combined position and attitude error. As a result, the asymptotic stability of the combined rotational and translational motion can be proven in a single step by using a Lyapunov function with the same form as the Lyapunov function normally used to prove the asymptotic stability of a rotational-only controller [Filipe and Tsiotras, 2014].

Dual quaternions have been successfully applied to various areas, such as computer vision [Daniilidis, 1999], animation [Ge and Ravani, 1994] and robotics [Aspragathos and Dimitros, 1998]. First, Funda and Paul [1990] proved that dual quaternions are more computationally efficient for robotic applications such as representing screw displacements of multi-linked spatial mechanisms. Then, Aspragathos and Dimitros [1998] showed that dual quaternions perform better than 4x4 homogeneous matrix transformations and Lie algebra in robotic kinematics. However, the first utilization of dual quaternions for spacecraft was proposed in spacial navigation missions [Yuanxin et al., 2005]. Since then, several studies have been carried out using dual quaternions for tracking problems. Among the most relevant ones: Wang et al. [2010] proposed an incremental Proportional, Integral and Derivative (PID) control law for a rendezvous problem on Earth. Zhang and Duan [2011] designed a robust finite-time control strategy for tracking a circular orbit nadir pointing. Filipe and Tsiotras [2014] used dual quaternions for developing an adaptive non-linear position and attitude tracking controller applied for docking two spacecraft orbiting Earth. Finally, Filipe and Ricardo [2014] developed the first Dual Quaternion Multiplicative Extended Kalman Filter (DQ-MEKF) and proved pose-tracking stability without velocity measurements for rendezvous and docking of two spacecraft.

However, non of the mentioned studies focused on asteroid close-proximity operations. One of the first utilizations of dual quaternions for asteroid operations was carried out by Razgus et al. [2017b]. They developed a new Dual Quaternion Extended Kalman Filter (DQ-EKF) and showed significant gains in the transients of the estimation errors compared to conventional filters using classical representations. Apart from that, Vukovich and Gui [2017] also proposed an adaptive robust control scheme based on dual quaternions for asteroid close-proximity operations, such as hovering and landing on 433 Eros. Despite the fact that Razgus et al. [2017b] compared the dual quaternion representation with conventional ones, they focused on filters

and not on the control part. And even though [Vukovich and Gui \[2017\]](#) proposed a controller based on dual quaternion for asteroid operations, they only compared it with a [Finite-Time Control \(FTC\)](#) with different characteristics from [\[Lee et al., 2015\]](#), and proved better performance only for the hovering scenario. The aim of this master thesis is then to take a step further and fairly compare the dual quaternion representation with respect to the classical one, focusing on the control part, for different asteroid close-proximity operations. It will try to determine whether or not, the believed advantages of dual quaternions also apply for asteroid close-proximity operations and quantify them. This thesis will not only look at the precision of the solution and magnitude of the errors, but it will also analyze the energy consumption, stability ranges of the controllers and computational times. The comparisons will be carried out for three very different scenarios, such as orbiting, hovering and landing on the asteroid Bennu.

1.1. RESEARCH OBJECTIVE

This master thesis will compare two different ways of expressing state variables of a full six [DOF](#) problem in asteroid close-proximity operations. The goal is to determine the advantages and disadvantages of each representation and quantify them.

The first representation method will be the classical one in which position, velocity and angular velocity are expressed by three-dimensional vectors, and quaternions are used for the attitude. The second method will be the dual quaternion one, in which both the pose (position and attitude) and dual velocity (linear and angular velocities) of the spacecraft are represented by unit dual quaternions.

All the comparisons will be carried out in three different asteroid close-proximity operations corresponding to orbiting, hovering and landing. The target asteroid will be Bennu, a B-type [Near Earth Object \(NEO\)](#) object that is relatively close to Earth.

This thesis will focus on the guidance and control part. Therefore, no navigation filters will be taken into account. The trajectory of the spacecraft will be assumed to be the one coming from numerically integrating the equations of motion. The space environment will be assumed to be perfectly known and modeled.

Many different aspects will be considered for determining the advantages and disadvantages of each representation. The first one will be the efficiency in which each method can express the equations of motion and solve them. A second one will be the effort taken for designing a controller and proving its stability. Third, the simplicity and time consumption of each method when selecting the control gains will also be assessed. The fourth one will be the performance in terms of errors, energy and time that each method takes when trying to follow a predefined reference trajectory about the asteroid. Fifth, a control frequency sensitivity analysis will be carried out. And finally, a perturbation analysis will be developed to determine which method can deal with larger perturbations and assess the performance of each controller when correcting them.

Bearing all these aspects in mind, the research question and sub-questions to be answered in this thesis can be written as follows.

- **Research question:** *What advantages and disadvantages can a six [DOF](#) controller based on dual quater-*

nions bring over classical methods regarding asteroid close proximity operations?

1.2. REPORT OUTLINE

This report consists of nine chapters and two appendixes. The first chapter is this current introduction and the list of the remaining chapters is presented below.

- **Chapter 2** explains the mission heritage and what have been learned from past missions to asteroids and dual quaternion representations. It describes the way in which attitude and position is usually represented and the mission and vehicle that will be used for this work. Finally, it concludes with the mission and system requirements.
- **Chapter 3** is an introduction to quaternions and dual quaternions and how they can represent attitude and position. Dual quaternions are defined and the most common operations are introduced. It is meant to be a familiarizing chapter for the reader.
- **Chapter 4** describes the flight dynamics and space environment. It first defines the reference frames that will be used in this work. Then, it derives the equations of motion both for the dual quaternion approach and the classical one. Finally, all the environment that will affect the spacecraft during the simulations is modeled.
- **Chapter 5** presents the controllers that will be used in each method for bringing the spacecraft down to the reference trajectory. Proof of Lyapunov stability is provided for all of them. It concludes with the gain selection and the explanation of how each gain will be optimized.
- **Chapter 6** covers the numerical methods that will be used for integrating the equations of motion.
- **Chapter 7** explains how the simulator will be implemented with a top level architectural design of all the unit blocks. Then, the process of verification and validation is presented for all previous blocks.
- **Chapter 8** presents and discusses the results of all simulations. It is where the dual quaternion approach is really compared with the classical one and where the research questions can be answered. A fair comparison is carried out for each of the three asteroid close proximity operations scenarios.
- **Chapter 9** concludes the report with the conclusions coming from the output of the comparisons and presents the recommendations for the future work.

This report ends with Appendixes [A](#) and [B](#). The first one covers theoretical explanations of some lemmas and theorems regarding Lyapunov stability of non-linear systems, which are relevant for the understanding of this work. Appendix [B](#) contains the remaining figures of the system tests that could not be shown in Chapter [7](#) for the sake of readability.

2

MISSION HERITAGE

This chapter aims at providing the mission heritage for this master thesis. Several past mission have been dedicated to study asteroids, therefore, a lot can be learned and used from them. Section 2.1 summarizes these missions and provides useful methodologies that can be applied for this thesis. In Section 2.2, the classical ways of representing the state variables of the full six **DOF** problem are presented. They represent the way in which all previous mission addressed the translational and rotational motion. Apart from that, the mission heritage of dual quaternions is also presented. Several authors have already studied and applied dual quaternions successfully for space applications, providing a solid ground-base to start this thesis. This allows to define several mission scenarios and spacecraft models in order to answer the research question, which is carried out in Section 2.3. Finally, the mission and systems requirements are stated in Section 2.4.

2.1. ASTEROID MISSIONS

The investigation of asteroids is now of increasing interest. They are remnants from the formation of the Solar System 4.6 billion years ago, and may contain the molecular precursors to the origin of life and the Earth's oceans. Some of them may also become hazardous objects, which can potentially impact Earth. For these reasons, several missions have been dedicated to study asteroids [Lissauer and de Pater, 2013]. This section summarizes the most important ones.

- **NEAR mission** Shoemaker was the first scientific mission from [National Aeronautics and Space Administration \(NASA\)](#), dedicated to the comprehensive study of an asteroid (433 Eros). It was launched in 1996. In 2000, the spacecraft successfully entered an orbit around the asteroid and carried out a one year orbital phase of operations at progressively lower altitudes. The orbital mission phase culminated in 2001 with a controlled descent and soft landing on the surface of Eros. Although landing on Eros was

Table 2.1: Hayabusa components for attitude determination and control [Masashi et al., 2006]

Component	Qty.	Specifications
Two Axis Sun Aspect Sensor (TSAS)	1	Field of view: +/- 50 x +/- 50° Accuracy better than 0.05°
Inertial Reference Unit (IRU)	2	Range: 1432°/s Max Resolution: 1.093x10 ⁻⁴ ° Bias stability: 3°/h (3σ) Random walk: 0.07 °/hr ^{1/2}
Star Tracker (STT)	1	Field of view 30° x 40° Star dynamic range: 3 ≈ -1 (CCD Magnitude) Accuracy (random, bias): 3 arcmin (3σ), 1 arcmin (3σ) Update rate: 1 Hz
Reaction Wheel (RW)	3	Maximum momentum: 4 Nms (@5000 rpm) Torque: 12 mNm minimum
Reaction Control System (RCS)	12 jets	Thrust: 20 Nominal Minimum impulse: 0.2 Ns

defined as the end of mission, the phenomenal success of this activity resulted in a two-week mission extension. The controlled descent and soft landing activity was conceived to satisfy two goals. The primary goal was to acquire high resolution images (10 cm) of the asteroid's surface down to a 500 m altitude. The second goal was a flight demonstration of a controlled descent to a small body. To satisfy this goal, an impact velocity less than 3 m/s was specified. Both goals were achieved and even a post touchdown communications beacon was acquired to confirm landing survival. Mission successfully ended in 2001 [Nelson, 2001].

- **Hayabusa mission** was designed by [Japanese Space Agency \(JAXA\)](#) to explore asteroid Itokawa. It was launched in 2003 and it was the first spacecraft to land on an asteroid and return samples back to Earth. Two months were spent for remote observations orbiting at an altitude range from 3 to 20 km. Two touchdowns were performed to collect a total of 1 g of surface material. Due to the small size of the asteroid, its gravitational effects were small and landing was more like docking [Masashi et al., 2006]. For attitude determination and control, Hayabusa used the components summarized in Table 2.1.
- **Rosetta mission** was a spacecraft built by [European Space Agency \(ESA\)](#) to study comet 67P/Churyumov-Gerasimenko launched in 2004. It orbited the comet for more than 2 years at altitudes from 5 to 20 km and successfully deployed and landed its Philae probe in 2014. The spacecraft accompanied the comet through perihelion until the end of mission in 2016 [Regnier et al., 2000]. Propellant mass was one of the main dynamic constraints because it represented about 50% of the satellite mass, thus generated non negligible sloshing effects and included sensitive mass and inertia variations along the mission. Another dynamic constraint was the presence of large solar arrays to ensure sufficient power at a distance of several AU from the Sun. The flexible appendages had rather low-frequency and represented

a significant amount of the spacecraft inertia. Also, during nominal operations, the spacecraft had to operate completely autonomously for up to a maximum period of time of about 2 days. The pointing requirement during the observation phases and the lander ejection was about 0.05° and the antenna pointing requirements in order to ensure Earth/satellite link was about 0.15° . Rosetta **GNC** equipment consisted of: two pairs of coarse Sun sensors to determine the Sun direction, two star trackers, three **Inertial Measurement Unit (IMU)** and two navigation cameras for autonomous asteroid direction measurements and to provide images to the ground. The propulsion system consisted of two sets of 12 thrusters with a nominal force of 10 N for attitude control and velocity increments. Finally, Rosetta was also equipped with four momentum wheels of 40 Nms capacity and a maximum torque of 19 mNm, mainly for precise attitude control and to absorb environment torques [Regnier et al., 2000]. The attitude estimation during the normal mode was performed by means of a gyro-stellar estimator with a 6-state Kalman filter and a constant covariance matrix. The initialization of the estimator was carried out with the attitude quaternion provided by the star tracker using the J2000 reference frame. The interest of that attitude estimator was that it led to a reduced computation load thanks to the constant covariance, without performance degradation. Moreover, the use of quaternions (instead of Euler angles) made it possible to operate at large angles. This was a requirement because the slew maneuvers were envisaged during the normal mode. The attitude control was performed by means of the four momentum wheels (with a maximum torque of 0.4 Nm). The controllers were adapted to each phase of the normal mode in order to ensure an adequate response time. During the asteroid fly-by mode, the two navigation cameras tracked the asteroid optical center at a frequency of 0.4 Hz, which enabled the generation of the attitude guidance profile through the Kalman filter. This was carried out by rotating the satellite about the minimum inertia axis with a maximum angular rate of $0.8^\circ/\text{s}$.

- **Hayabusa 2 mission** is the second asteroid sampling and return mission of **JAXA** followed by its successful predecessor mission. It was launched in 2014 and it is expected to study asteroid Ryugu (1999 JU3) in July 2018. The spacecraft is based on the technological heritage of Hayabusa incorporating some major improvements in terms of navigation, propulsion, attitude control and sampling methods [Tsuda et al., 2013]. It will perform some asteroid close proximity operations for 1.5 years, including 3 touchdowns for sampling and a 2-m-class crater generation by a high-speed impact operation. The sample is to be brought back to Earth by a re-entry capsule. For attitude and orbit control Hayabusa 2 uses the systems summarized in Table 2.2. The sampling operation sequence will be divided into three phases. An initial descent at 0.1 m/s controlled by ground operators; a fully autonomous second descent where a target marker is deployed and followed using on-board navigation cameras and the flash **Lidar**; and a final landing phase starting at an altitude of 30 m using the **LRF** to determine the local surface orientation relative to the spacecraft [Tsuda et al., 2013].
- **OSIRIS-REx mission** which stands for Origins, Spectral Interpretation, Resource Identification, Security Regolith Explorer (**OSIRIS-REx**) is a **NASA** mission launched in 2016 to characterize and map the surface of asteroid Bennu and return a sample to Earth (60 g approx). The spacecraft is expected to

Table 2.2: Specifications of the Hayabusa 2 spacecraft [Tsuda et al., 2013]

System	Specifications
Attitude and orbit control	HR5000S based processor, double redundant
	4 Reaction wheels, 2 IRUs, 2 Star trackers
	4 Coarse sun aspect sensors, 4 accelerometers
	Sensor for proximity operation Lidar, LRF, 5 Target makers, Flash lamp Optical navigation cameras
Propulsion system	Reaction control system
	- Bi-propellant hydrazine system
	- 20 N thruster x 12
	Ion engine system
	- Xe microwave discharge ion engine system
	- Maximum thrust 28 mN, Isp = 2800 s - 4 thruster heads on gimballed stage - 3 operative at once (4/3 redundant)

arrive in November 2018 and orbit Bennu at 1 and 1.5 km altitudes for one year. A touch and go phase is planned for getting the sample [Lorenz et al., 2017]. With a one-way light time of up to 18 minutes, spacecraft operations during this maneuver will have to be performed completely autonomously with a [Light Detection and Ranging \(Lidar\)](#) system combined with an optical system for [Guidance, Navigation and Control \(GNC\)](#). Unlike a planetary orbital environment where the planet's gravity is by far the dominant force acting on the spacecraft, the "small forces" acting on the [OSIRIS-REx](#) spacecraft will be of comparable magnitude to the asteroid's gravity, so that the accuracy with which the accelerations from these forces can be modeled will determine how accurately the spacecraft's future orbital state can be predicted. Figure 2.1 shows the magnitude of these small forces. It is clear that the most relevant forces will be the ones coming from Bennu's gravity field, the [SRP](#) and the Sun third body forces. The requirements in position for the touch-and-go sequence are set to be an ellipse of only 25 m in diameter. To do so, [OSIRIS-REx](#) is equipped with 4 reaction wheels and 28 engines divided into four groups: a bank of four high-thrust main engines, six medium-thrust engines, 16 attitude control thrusters, and a pair of specialized low-thrust engines. It is also equipped with 2 [IMU](#), 2 star trackers, 4 headed Sun sensors, a [Lidar](#) system and a navigation optical camera [May et al., 2014]. All the details regarding the touch-and-go sequence are summarized in Figure 2.2.

2.2. ATTITUDE AND POSITION REPRESENTATIONS

Position, velocity and angular velocity are always expressed with a three dimensional vector. That is the way all previous mentioned missions did it. However, there are different ways of expressing the orientation of a rigid body. The most commonly used and well known representations are: [Direction Cosine Matrix](#)

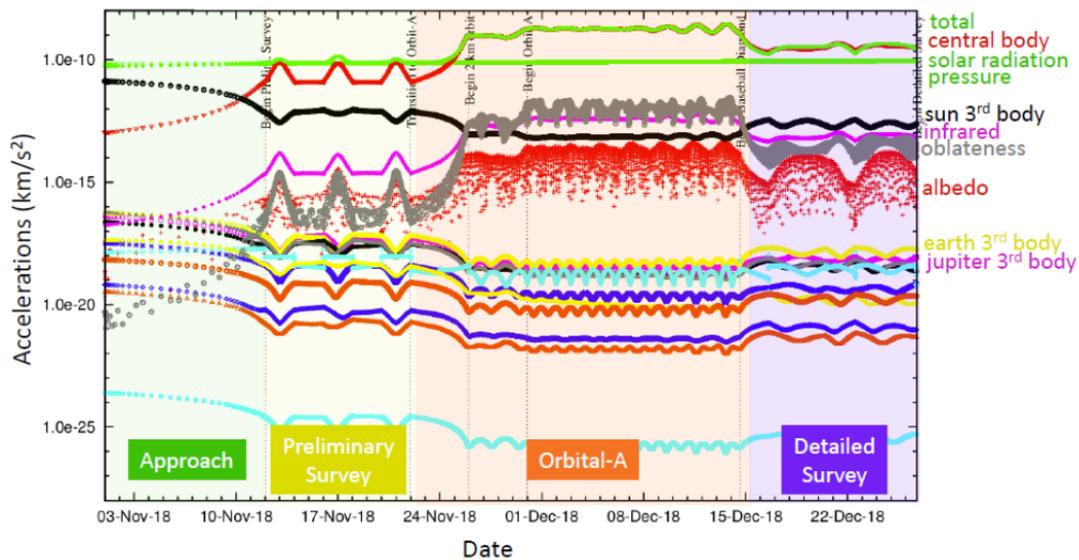


Figure 2.1: Different forces acting on OSIRIS-REx spacecraft on different phases. [Everett et al., 2017]

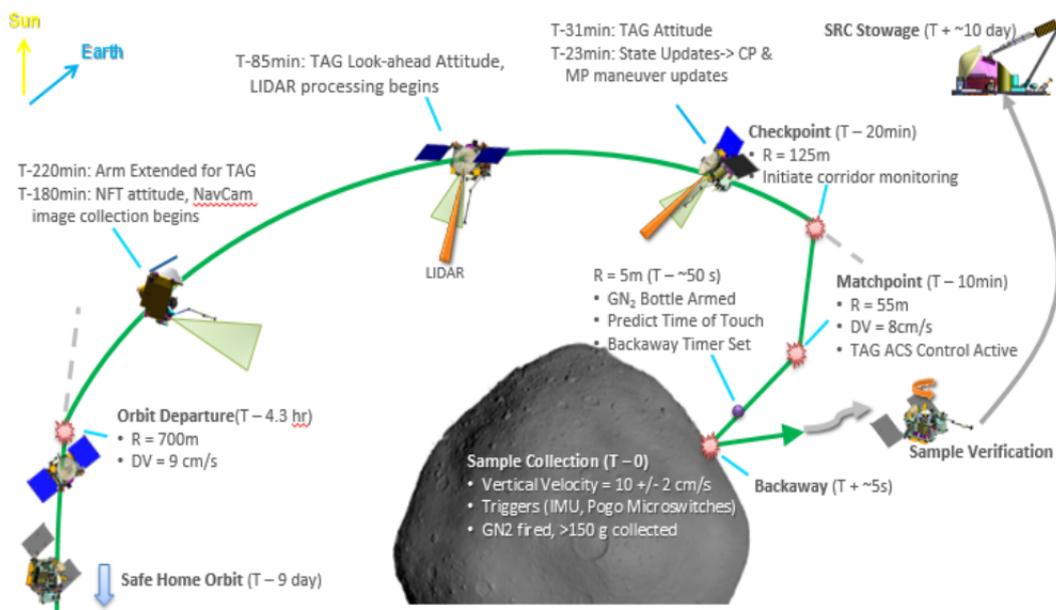


Figure 2.2: OSIRIS-REx touch-and-go phase event overview [Lorenz et al., 2017]

(DCM), Reduced Direction Cosine Matrix (RDCM), euler angles, principal rotation vectors, unit quaternions, Rodrigues Parameters (RP) and Modified Rodrigues Parameters (MRP).

For this thesis work, unit quaternions have been selected to be the classical way of representing the attitude of the spacecraft. They have no singularities, only four parameters are required, and dual quaternions are much similar to them than to any other representation. Furthermore, all previous mentioned past mission most probably also used unit quaternions to express their orientation.

However, as has been already introduced in Chapter 1, there is another way of representing the rotational and translational motion of a rigid body. Position and attitude can be represented simultaneously with a single variable using dual quaternions. The same can be said about the linear and angular velocities. Dual quaternions are a mathematical formulation based on dual numbers with its own algebra associated with them [Filipe and Tsiotras, 2013]. They are an extension of classical quaternions and therefore, they allow to write the combined translational and rotational equations of motion in the same form as rotational-only equations written in terms of quaternions. This appealing property can be used to design controllers by extending already existing attitude-only controllers with some desirable properties by often simply substituting quaternions with dual quaternions. Fundamentals of dual quaternions and how they can represent position and attitude at the same time is explained in detail in Chapter 3.

Dual quaternions have been successfully applied to different fields, such as computer vision [Daniilidis, 1999], animation [Ge and Ravani, 1994] and robotics [Aspragathos and Dimitros, 1998]. However, the first utilization of dual quaternions for spacecraft was proposed in spacial navigation missions [Yuanxin et al., 2005]. Since then, several studies have been carried out using dual quaternions and they are becoming more popular in the literature [Wang et al., 2010; Zhang and Duan, 2011; Filipe and Tsiotras, 2014].

Relevant heritage can be found in the work of Filipe and Ricardo [2014] and Razgus et al. [2017b], who developed new filters using dual quaternions and showed their benefits; such as proving significant gains in the transients of the estimation errors compared to conventional filters using classical representations. Apart from that, relevant heritage can also be found in the work of Wang and Yu [2010]; Filipe and Tsiotras [2013]; Gui and Vukovich [2015]; Vukovich and Gui [2017], who proposed several controllers using dual quaternions. Wang and Yu [2010] designed an incremental PID control law taking into account the coupled dynamics and control for a rendezvous problem during final approach on an Earth orbit. Filipe and Tsiotras [2014] designed a nonlinear adaptive controller for the same problem, but without requiring information about the mass and inertial matrix of the chaser satellite. They also showed almost global asymptotic stability of the tracking errors. Then, Gui and Vukovich [2015] developed another controller using dual quaternions combining a simple PD controller with an adaptive algorithm which provided estimations of unknown parameters and disturbances. They ensured almost global asymptotic convergence of the relative pose tracking error as well, and showed a reduced control energy consumption and computationally more efficiency than [Filipe and Tsiotras, 2014]; due to the gyroscopic terms involved in their control law. Finally, Vukovich and Gui [2017] improved their own previous control law [Gui and Vukovich, 2015] by introducing another adaptive algorithm to dynamically adjust the compensation of perturbations due to parametric uncertainties and external disturbances.

2.3. MISSION AND VEHICLE DESCRIPTION

This section describes the mission scenarios and vehicle model that will be used for comparing the dual quaternion approach with the classical one.



Figure 2.3: A rendering of Benu asteroid. Image credit NASA public domain

2.3.1. MISSION DEFINITION

To compare the dual quaternion representation with the classical ones, a realistic mission scenario involving close proximity operations around an asteroid is required. Thus, a target asteroid is first needed.

It has to be an asteroid small enough so that coupled dynamics can play an important role. Also, it has to be a previously visited asteroid or otherwise, a sufficiently characterized asteroid with on-ground measurements, so that dynamical models can be developed for its close-proximity environment. Finally, it is always desired to select an asteroid feasible to reach with current technology. For these reasons, asteroid Benu has been selected to be the main body of all simulations that will be carried out in this thesis. Benu is a B-type [Near Earth Object \(NEO\)](#) object that is relatively close to Earth (it orbits within 1.3 [Astronomical Unit \(AU\)](#) of the Sun) and has an Earth-like orbit with low eccentricity and inclination. Apart from that, it is the best characterized asteroid with on-ground measurements; and there is sufficient information to create accurate enough models. Benu is also rich in terms of chemical composition because it contains carbon, organic molecules, volatiles and amino acids that may have been the precursors to life on Earth [[Scheeres et al., 2016](#)]. Apart from that, it has not significantly changed since it was formed 4 billion years ago. Figure 2.3 shows a rendered image of the current Benu model.

The main physical properties of Benu that will be used in this thesis are summarized in Table 2.3. It is important to mention that Benu is still "unknown", because it has not been visited yet and all information available at the moment might change since it is Earth-based measured. Table 2.3 presents the current assumptions that are thought to be accurate within certain accuracy.

Benu orbital elements at Epoch 2455562.5 (2011-Jan-01.0) are stated in Table 2.4 taken from the [Jet Propulsion Laboratory \(JPL\)](#) database for [NEO](#)¹.

Then, three different asteroid close-proximity operations can be defined. They are meant to represent realistic scenarios for asteroid characterization missions. These are: orbiting, hovering and landing on Benu. These will be the test scenarios where the dual quaternion approach will be compared with the classical one to come up with different conclusions regarding computational times, efficiency, energy consumption and precision.

ORBITING

The first scenario will be to orbit Benu at 1500 m. This orbiting scenario tries to simulate real mission phases where the spacecraft studies the asteroid while orbiting. Normally, if the asteroid allows for stable orbits, they

¹ Retrieved from "<https://ssd.jpl.nasa.gov/sbdb.cgi?sstr=Benu>" on 8 August 2017

Table 2.3: Bennu assumed properties (Earth-based) [Scheeres et al., 2016]

Mean radius	246 ± 10 m
Volume	$6.23 \pm 0.06 \cdot 10^7$ m ³
Density	1260 ± 70 kg/m ³
Nominal Total mass	$7.8 \cdot 10^{10}$ kg
Nominal Gravitational parameter	5.2 m ³ /s ²
Inertia principal moments	
I_{xx}	$1.3749 \cdot 10^{15}$ kg/m ²
I_{yy}	$1.4285 \cdot 10^{15}$ kg/m ²
I_{zz}	$1.5421 \cdot 10^{15}$ kg/m ²
Spin rate	$4.0613 \cdot 10^{-4} \pm 1.9 \cdot 10^{-7}$ rad/s
Spin pole	Retrograde - obliquity $\approx 180^\circ$

Table 2.4: JPL database for Bennu ephemeris ¹

Eccentricity	$0.20374510 \pm 2.2237e-08$ (1 σ)
Semi Major Axis	$1.12639102602 \pm 4.0315e-11$ au (1 σ)
Inclination	$6.034939 \pm 2.8409e-06^\circ$ (1 σ)
Perihelion distance	$0.89689436 \pm 2.5034e-08$ au (1 σ)
Aphelion distance	$1.35588768855 \pm 4.8529e-11$ au (1 σ)
Longitude of the ascending node	$2.060867 \pm 3.9515e-06^\circ$ (1 σ)
Argument of perihelion	$66.223067 \pm 6.096e-06^\circ$ (1 σ)
Mean anomaly	$101.703947 \pm 2.6659e-06^\circ$ (1 σ)
Time of perihelion passage	$2455439.141945 \pm 3.2307e-06$ JED (1 σ)
Orbital period	$436.64872818 \pm 2.3442e-08$ days (1 σ)
Mean motion	$0.82446135019 \pm 4.4263e-11^\circ$ /d (1 σ)

do not perform any type of orbit control (or barely any) so that the gravity field can be characterized, or many other scientific experiments can be carried out. In the case of Bennu, it allows for very stable orbits [Hergenrother, 2014], therefore, no position control will be carried out. The selected orbit for all the simulations in this thesis will be a Sun-synchronous terminator orbit. This is an orbit at 1500 m following the plane perpendicular to the Sun vector from the center of the asteroid. It is a very scientifically interesting orbit because apart from being a polar orbit (covering all latitudes and longitudes), the illumination is always the same; which is desired for observational purposes. Regarding the orientation of the satellite, it will try to point to the asteroid (simulating an observing camera) at all times. The orbital period for such an orbit is about 45.46 hours with a velocity of about 5.89 cm/s. The simulation time has been selected to be of 1 orbital period.

HOVERING

The second scenario will consist in hovering at 500 m from the surface of Bennu. It has been found [Hergenrother, 2014], that one of the best spots for hovering is in the equator. It allows to observe the asteroid

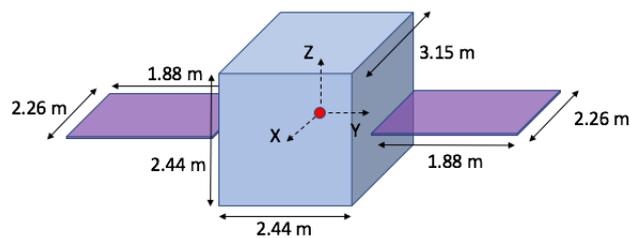


Figure 2.4: Spacecraft model (based on OSIRIS-REx parameters)

while it rotates and capture many interesting scientific features. The perfect illumination for this is when the line from the satellite to the center of the asteroid forms a 30° angle with the line going from the Sun to the center of Bennu. The shadows created from the craters and features on the surface are easier to detect and understand. For this reason, this will be the selected spot for hovering in all these simulations. Regarding the orientation of the satellite, it will always try to point to the center of the asteroid as well (simulating a camera or Lidar for scientific purposes). It is important to remark here that this is an inertial hovering. This means that the satellite will be fixed in an inertial spot while the asteroid will be rotating. This allows the satellite to observe all the faces of the asteroid, which is scientifically very interesting.

LANDING

The last scenario will be the landing on the surface of Bennu. On a real mission, the spacecraft always first studies the surface of the asteroid and then a landing spot is selected taking into account many different aspects such as scientific interests, safety, data links, etc. As has been already explained, Bennu has never been visited before, therefore, the real surface is not really known. For this reason, any point of the surface can be a candidate spot for landing. For the sake of simplicity, the reference landing trajectory has been selected to follow the x body-fixed axis of Bennu from an altitude of 60 m above the surface to actually land on it. This axis corresponds to the principal moment of inertia. The nominal landing velocity will be 10 cm/s; this means that the maneuver will take 10 minutes to be completed. Regarding the orientation of the satellite, it will have to point to the center of the asteroid at all times.

2.3.2. SPACECRAFT MODEL

The spacecraft model selected for this thesis work is a simplification of the real OSIRIS-REx spacecraft. Almost all parameters are published and will be used.

PHYSICAL PROPERTIES

Regarding physical properties, everything has been made public except for the inertia tensor (Lockheed Martin does not publish this type of information). Table 2.5 states all the known information. Figure 2.4 shows the spacecraft model with its dimensions and body frame coordinate system definition.

To compute the inertia tensor, the spacecraft is assumed to be a box with two panels on each side, as shown in Figure 2.4. The panels are located in the XY plane of the body frame of the satellite and are both

Table 2.5: Spacecraft model properties

Name	Magnitude	Comments
Mass	1193.5 kg	Mass at Bennu arrival ²
Main body mass	1138.5 kg	Mass of the main body (cuboid)
Solar panels mass	22.5 kg	Mass of the solar panels (each)
Height	2.44 m	Height of main body ³
Width	2.44 m (6.2 m)	Width (with solar panels deployed) ³
Length	3.15 m	Length of main body ³
Solar array dim.	8.5 m ²	Total area of solar arrays
Reflectivity coef.	1.28	Reflectivity coefficient
Mass to Area ration	≈100 kg/m ²	Mass to area ratio for SRP computations

equal. They have a total area of 8.5 m², with assumed dimensions of 2.26 m and 1.88 m each. Their masses are assumed to be 22.5 kg each. It is important to remark that the real **OSIRIS-REx** spacecraft has different dimensions and inertia tensor. Since the goal of this master thesis is not to reproduce very accurate results compared to the **OSIRIS-REx** mission, but to study possible advantages of dual quaternion, there is no reason why the inertia tensor should be exactly the same.

The inertia tensor of a filled box (I_b) with respect to its center of mass can be computed as follows:

$$\mathbf{I}_b = \begin{bmatrix} \frac{1}{12} m_b (l_y^2 + l_z^2) & 0 & 0 \\ 0 & \frac{1}{12} m_b (l_x^2 + l_z^2) & 0 \\ 0 & 0 & \frac{1}{12} m_b (l_x^2 + l_y^2) \end{bmatrix} = \begin{bmatrix} 1129.70 & 0 & 0 \\ 0 & 1506.24 & 0 \\ 0 & 0 & 1506.24 \end{bmatrix} \text{ kg} \cdot \text{m}^2 \quad (2.1)$$

where m_b is the mass of the box (1138.5 kg) and l_x, l_y and l_z are its dimensions (depth = 3.15 m, width = 2.44 m and height = 2.44 m). Then, applying the parallel axis theorem (2.2), the inertia of each panel (I_w) with respect to the center of mass of the complete body (which does not change when adding the two solar panels) can be computed as follows. It is important to remark that a negligible thickness is considered.

$$\mathbf{I}_w = \mathbf{I}_{w_{cm}} + \mathbf{M} \cdot \mathbf{d}^2 \quad (2.2)$$

$$\mathbf{I}_w = \begin{bmatrix} \frac{1}{12} m_w (s_y^2) + m_w (d_x^2) & 0 & 0 \\ 0 & \frac{1}{12} m_w (s_x^2) + m_w (d_y^2) & 0 \\ 0 & 0 & \frac{1}{12} m_w (s_x^2 + s_y^2) + m_w (d_z^2) \end{bmatrix} \quad (2.3)$$

$$\mathbf{I}_w = \begin{bmatrix} 111.60 & 0 & 0 \\ 0 & 9.58 & 0 \\ 0 & 0 & 121.18 \end{bmatrix} \text{ kg} \cdot \text{m}^2 \quad (2.4)$$

²Taken from Beckman et al. [2013]

³"OSIRIS-REx: Asteroid Sample Return Mission" (PDF) (Press Kit). NASA. August 2016. Retrieved 7 August 2017

where m_w is the mass of one solar panel (22.5 kg) and s_x and s_y are the dimensions (depth = 2.26 m and width = 1.88 m) of the panel respectively. d_x, d_y and d_z are the distances between the main body principal axis and the solar panel principal axis of inertia. The total inertia tensor of the spacecraft is just then the summation of the inertia tensors of all components with respect to the center of mass.

$$\mathbf{I}_{Total} = \mathbf{I}_b + 2\mathbf{I}_w = \begin{bmatrix} 1352.9 & 0 & 0 \\ 0 & 1525.4 & 0 \\ 0 & 0 & 1748.6 \end{bmatrix} \text{ kg} \cdot \text{m}^2 \quad (2.5)$$

ACTUATORS

For attitude and orbit control, the spacecraft should be able to provide torques and forces. Since the goal of this master thesis is to study the possible advantages or disadvantages when expressing the state variables with dual quaternions, it does not really matter how these torques or forces are provided. OSIRIS-REx spacecraft uses thrusters or reactions wheels for attitude control depending on the phase of the mission and the requirements. For the sake of simplicity, it will be assumed that the attitude control in all the simulations will be provided with reaction wheels. These reaction wheels will be able to create torques in all three body axis of the satellite and will have a maximum torque of 0.2 Nm per axis. There will be no limitation on time providing maximum torque.

For orbit control, it will be assumed that the spacecraft has six thrusters (one in each face) aligned with the center of mass of the satellite to make things easier. In that way, it should be able to create a force in any direction without creating any torque. The maximum thrust force allowed by these engines will be 20 N in all six directions without any limitation on how much time they can be firing.

This configuration will allow the controllers to create torques and forces so that the spacecraft can follow the desired trajectories. The reason why there is a maximum torque and force in each direction is because in this way, fair comparisons can be made between controllers. If there were no maximum limitations, the controllers would simply command almost infinite torque/force so that the settling time would be as small as possible. Allowing for maximums not only brings more reality but it also solves this problem. These maximums have been selected by looking into actuators performances of OSIRIS-REx reaction wheels and engine thrusters and taking into account other past missions heritage.

2.4. MISSION AND SYSTEMS REQUIREMENTS

In this master thesis, controllers should be able to bring the spacecraft down to the reference trajectory meeting some requirements. If these requirements are met, the reference trajectory can be considered to be successfully achieved. Having said that, some mission and systems requirements can be defined.

MISSION REQUIREMENTS

- **MR01** The spacecraft shall land on the surface of Bennu following the defined reference trajectory.

- **MR01.1** The position error with respect to the landing trajectory shall not exceed 0.01 m in absolute value for each 3 axes.
- **MR01.2** The velocity error with respect to the landing trajectory shall not exceed 0.01 cm/s in absolute value for each 3 axes.
- **MR01.3** The pointing attitude error with respect to the landing trajectory shall not exceed 0.01° in absolute value.
- **MR01.4** The angular velocity error with respect to the landing trajectory shall not exceed $0.0001^\circ/\text{s}$ in absolute value for each 3 axes.
- **MR01.5** The previous stability conditions shall be met before the settling time of 1 minute.
- **MR02** The spacecraft shall hover about Bennu in the defined spot.
 - **MR02.1** The position error with respect to the hovering reference spot shall not exceed 0.01 m in absolute value for each 3 axes.
 - **MR02.2** The velocity error with respect to the hovering reference velocity shall not exceed 0.01 cm/s in absolute value for each 3 axes.
 - **MR02.3** The pointing attitude error with respect to the hovering reference attitude shall not exceed 0.01° in absolute value.
 - **MR02.4** The angular velocity error with respect to the hovering reference angular velocity shall not exceed $0.0001^\circ/\text{s}$ in absolute value for each 3 axes.
 - **MR02.5** The previous stability conditions shall be met before the settling time of 1 minute.
- **MR03** The spacecraft shall orbit Bennu following the defined reference trajectory.
 - **MR03.1** The pointing attitude error with respect to the orbiting trajectory shall not exceed 0.01° in absolute value.
 - **MR03.2** The angular velocity error with respect to the orbiting trajectory shall not exceed $0.0001^\circ/\text{s}$ in absolute value for each 3 axes.
 - **MR03.5** The previous stability conditions shall be met before the settling time of 10 minutes.

SYSTEMS REQUIREMENTS

- **SR01** Position actuators shall be able to provide forces at all times with a maximum nominal force of 20 N in each direction.
- **SR02** Position actuators shall not generate a net torque when producing forces.
- **SR02** Attitude actuators shall be able to provide torques at all times with a maximum nominal torque of 0.2 N·m in each direction.
- **SR01** Attitude actuator shall not generate a net force when executing torque commands.
- **SR02** The spacecraft shall be able to withstand all conditions suffered during close-proximity operations.

3

QUATERNIONS AND DUAL QUATERNIONS

The aim of this chapter is to introduce and define the fundamentals of quaternions and dual quaternions. Apart from that, it also explains how quaternions can represent the attitude of spacecraft and how dual quaternions can simultaneously represent both the attitude and the position of them. First, some fundamentals of quaternions are provided in Section 3.1 and then, the way in which they can be used to represent the orientation of a rigid body is explained in Subsection 3.1.1. Second, dual quaternions are presented. They are not easy to understand and it can represent a challenge for the reader to follow all the derivations in this master thesis. For this reason, a proper introduction to them and some fundamentals are presented in Section 3.2, with some visual interpretations. The way in which dual quaternions can be used to represent both the position and orientation of spacecraft is defined in Subsection 3.2.1. Finally, some coordinate transformations with dual quaternions are defined in Subsection 3.2.2.

3.1. QUATERNIONS

As explained in [Han et al., 2008], quaternions can be understood as an extension of complex numbers. A quaternion can be defined as

$$\mathbf{q} = q_0 + q_1i + q_2j + q_3k \quad (3.1)$$

where $q_0, q_1, q_2, q_3 \in \mathbb{R}$. i, j, k satisfy $i^2 = j^2 = k^2 = -1$ and $i = jk = -kj, j = ki = -ik$ and $k = ij = -ji$. However, they can also be represented as an ordered pair $\mathbf{q} = (q_0, \bar{\mathbf{q}})$; where q_0 is called the scalar part and $\bar{\mathbf{q}} = [q_1; q_2; q_3] \in \mathbb{R}^3$ is called the vector part. The set of quaternions are denoted by \mathcal{H} and some of their basic operations are summarized here.

Addition:

$$\mathbf{a} + \mathbf{b} = (a_0 + b_0, \bar{\mathbf{a}} + \bar{\mathbf{b}}) \in \mathcal{H} \quad (3.2)$$

Multiplication by a scalar:

$$\lambda \mathbf{a} = (\lambda a_0, \lambda \bar{\mathbf{a}}) \in \mathcal{H} \quad (3.3)$$

Multiplication:

$$\mathbf{a}\mathbf{b} = (a_0 b_0 - \bar{\mathbf{a}} \cdot \bar{\mathbf{b}}, a_0 \bar{\mathbf{b}} + b_0 \bar{\mathbf{a}} + \bar{\mathbf{a}} \times \bar{\mathbf{b}}) \in \mathcal{H} \quad (3.4)$$

Conjugation:

$$\mathbf{a}^* = (a_0, -\bar{\mathbf{a}}) \in \mathcal{H} \quad (3.5)$$

Dot product:

$$\mathbf{a} \cdot \mathbf{b} = \frac{1}{2}(\mathbf{a}^* \mathbf{b} + \mathbf{b}^* \mathbf{a}) = (a_0 b_0 + \bar{\mathbf{a}} \cdot \bar{\mathbf{b}}, \mathbf{0}) \in \mathcal{H} \quad (3.6)$$

Cross product:

$$\mathbf{a} \times \mathbf{b} = \frac{1}{2}(\mathbf{a}\mathbf{b} - \mathbf{b}^* \mathbf{a}^*) = (0, b_0 \bar{\mathbf{a}} + a_0 \bar{\mathbf{b}} + \bar{\mathbf{a}} \times \bar{\mathbf{b}}) \in \mathcal{H} \quad (3.7)$$

Norm:

$$\|\mathbf{a}\|^2 = \mathbf{a}\mathbf{a}^* = (a_0^2 + \bar{\mathbf{a}} \cdot \bar{\mathbf{a}}, \mathbf{0}) \in \mathcal{H}_d \quad (3.8)$$

It is important to remark that the quaternion multiplication is not commutative.

3.1.1. ATTITUDE REPRESENTATION

Quaternions are sometimes also referred to as the four Euler parameters, and they can provide a redundant, nonsingular attitude description. They are well suited to describe arbitrary, larger rotations. A unit quaternion \mathbf{q} can be defined in terms of principal rotation elements, to represent the attitude of a solid rigid, as described by [Schaub and Junkins \[2009\]](#).

$$q_0 = \cos(\Phi/2) \quad (3.9)$$

$$q_1 = e_1 \sin(\Phi/2) \quad (3.10)$$

$$q_2 = e_2 \sin(\Phi/2) \quad (3.11)$$

$$q_3 = e_3 \sin(\Phi/2) \quad (3.12)$$

where Φ is the rotated angle and e_1 , e_2 and e_3 are the component of the unit vector along which the rotated angle is applied. It is evident since $e_1^2 + e_2^2 + e_3^2 = 1$, that the q_i satisfy the holonomic constraint

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \quad (3.13)$$

It is important to mention that this constraint geometrically describes a four dimensional unit sphere. Any rotation described through a quaternion has a trajectory on the surface of this constraint sphere. Given a certain attitude, there are actually two sets of Euler parameters that will describe the same orientation. This is due to the non-uniqueness of the principal rotation elements themselves. Since any point on the unit constraint sphere surface represents a specific orientation, the anti-pole to that point represents the exact same orientation. The difference between the two attitude descriptions is that one specifies the orientation through the shortest single axis rotation, and the other through the longest.

Using the trigonometric identities

$$\sin \Phi = 2 \sin(\Phi/2) \cos(\Phi/2)$$

$$\cos \Phi = 2 \cos^2(\Phi/2) - 1$$

The DCM can be written in terms of quaternions as

$$\mathbf{C} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (3.14)$$

It is evident that most general angular motion of a reference frame generates two arcs on the four dimensional unit sphere (the geodesic arcs generated by $\mathbf{q}(t)$ and $-\mathbf{q}(t)$). This elegant description is universally non-singular and is unique to within the sign $\pm \mathbf{q}(t)$. In this thesis the positive sign will always be selected. The inverse transformation from \mathbf{C} to quaternions can be found following the algorithm developed by Sheppard [Schaub and Junkins, 2009]. First, the four q_i^2 terms are computed:

$$q_0^2 = \frac{1}{4}(1 + \text{trace} \mathbf{C}) \quad (3.15)$$

$$q_1^2 = \frac{1}{4}(1 + 2C_{11} - \text{trace} \mathbf{C}) \quad (3.16)$$

$$q_2^2 = \frac{1}{4}(1 + 2C_{22} - \text{trace} \mathbf{C}) \quad (3.17)$$

$$q_3^2 = \frac{1}{4}(1 + 2C_{33} - \text{trace} \mathbf{C}) \quad (3.18)$$

Then, the square root of the largest q_i^2 found from Equation (3.15) to (3.18) has to be taken and the sign of q_i is chosen to be positive to avoid singularities. The other q_i are found by dividing the appropriate three of the following six in Equation (3.19) by the chosen largest q_i coordinate:

$$\begin{aligned} q_0 q_1 &= (C_{23} - C_{32})/4 \\ q_0 q_2 &= (C_{31} - C_{13})/4 \\ q_0 q_3 &= (C_{12} - C_{21})/4 \\ q_2 q_3 &= (C_{23} - C_{32})/4 \\ q_3 q_1 &= (C_{31} - C_{13})/4 \\ q_1 q_2 &= (C_{12} - C_{21})/4 \end{aligned} \quad (3.19)$$

A very important property of the quaternions is the manner in which they allow two sequential rotations to be combined into one overall composite rotation. Let the quaternion \mathbf{q}' describe the first, \mathbf{q}'' the second, and \mathbf{q} the composite rotation. It is then clear that

$$\mathbf{FN}(\mathbf{q}) = \mathbf{FB}(\mathbf{q}'')\mathbf{BN}(\mathbf{q}') \quad (3.20)$$

Introducing Equation (3.14) into Equation (3.20) and equating corresponding elements leads to the following elegant transformation that bilinearly combines \mathbf{q}' and \mathbf{q}'' into \mathbf{q} :

$$\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{bmatrix} q''_0 & -q''_1 & -q''_2 & -q''_3 \\ q''_1 & q''_0 & q''_3 & -q''_2 \\ q''_2 & -q''_3 & q''_0 & -q''_1 \\ q''_3 & q''_2 & -q''_1 & q''_0 \end{bmatrix} \begin{pmatrix} q'_0 \\ q'_1 \\ q'_2 \\ q'_3 \end{pmatrix} \quad (3.21)$$

By transmutation of Equation (3.21), an alternate expression $\mathbf{q} = \mathbf{G}(\mathbf{q}')\mathbf{q}''$ is found

$$\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{bmatrix} q'_0 & -q'_1 & -q'_2 & -q'_3 \\ q'_1 & q'_0 & q'_3 & -q'_2 \\ q'_2 & -q'_3 & q'_0 & -q'_1 \\ q'_3 & q'_2 & -q'_1 & q'_0 \end{bmatrix} \begin{pmatrix} q''_0 \\ q''_1 \\ q''_2 \\ q''_3 \end{pmatrix} \quad (3.22)$$

where components of the matrix $\mathbf{G}(\mathbf{q}')$ are given in Equation (3.22). Note the useful identity

$$[\mathbf{G}(\mathbf{q}')]^T \mathbf{q} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.23)$$

By inspection, it is evident that the 4x4 matrices in Equation (3.21) and (3.22) are orthogonal. These transformations provide a simple, non singular, and bi-linear method to combine two successive rotations described through quaternions. For other attitude parameters such as the Euler angles, this same composite transformation would yield a very complicated, transcendental expression.

3.2. DUAL QUATERNIONS

Dual quaternions were first introduced by Clifford [1871] and they are built on and are an extension of classical quaternions. Dual quaternions are closely related to Chasles Theorem, which states that the general displacement of a rigid body can be represented by a rotation about an axis (called the screw axis) and a translation along that axis, creating a screw-like motion [Murray et al., 1994]. This allows to visualize dual quaternions in a similar way a quaternion can be visualized (Euler theorem). The geometry of this process can be depicted in Figure 3.1. The initial pose of a certain spacecraft could be the one on the right of Figure 3.1, and its final position could be the one on the left. 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 a distance d . The final frame is displaced by a vector \mathbf{R} (expressed in the initial frame) with respect to the initial frame.

However, to better understand dual quaternions, it is also needed to understand dual numbers and dual algebra. In linear algebra, the dual numbers extend the real numbers by adjoining one new element ϵ with the property $\epsilon^2 = 0$ (ϵ is nilpotent). The collection of dual numbers forms a particular two-dimensional commutative unital associative algebra over the real numbers. The dual numbers can also be thought of as the exterior algebra of a one-dimensional vector space. Since some operations and properties of dual numbers

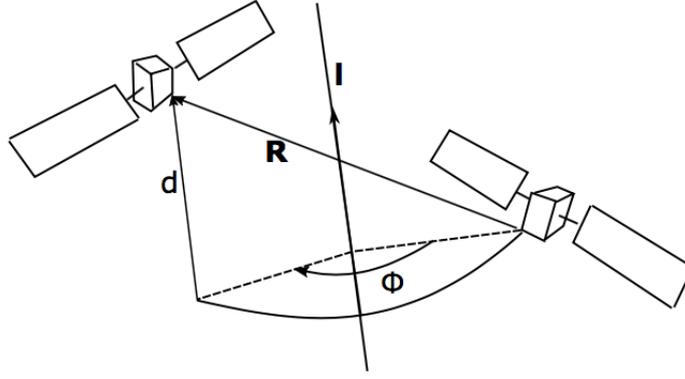


Figure 3.1: Visualization of a screw transformation [Razgus et al., 2017a]

(and specifically dual quaternions) might not be familiar to the reader, this chapter defines the basis and some operations and properties that will be used in this thesis. It is meant to be a familiarizing chapter.

As explained in [Filipe and Tsiotras, 2014], a dual quaternion is a dual number defined as

$$\hat{\mathbf{q}} = \mathbf{q}_r + \epsilon \mathbf{q}_d \quad (3.24)$$

where ϵ is the dual unit defined by

$$\epsilon^2 = 0 \quad \epsilon \neq 0 \quad (3.25)$$

The quaternions $\mathbf{q}_r, \mathbf{q}_d \in \mathcal{H}$ are called the "real part" and the "dual part" of the dual quaternion, respectively. It is important to notice that dual quaternions are represented in bold and with a hat. This distinguishes them from normal quaternions and vectors.

Dual vector quaternions and dual scalar quaternions are dual quaternions formed from vector quaternions (i.e., $\mathbf{q}_r, \mathbf{q}_d \in \mathcal{H}^v$) and scalar quaternions (i.e., $\mathbf{q}_r, \mathbf{q}_d \in \mathcal{H}^s$), respectively. The set of dual quaternions, dual scalar quaternion, dual vector quaternions, and dual scalar quaternions with zero dual part will be respectively denoted by

$$\begin{aligned} \mathcal{H}_d &= \{\hat{\mathbf{q}} : \hat{\mathbf{q}} = \mathbf{q}_r + \epsilon \mathbf{q}_d, \mathbf{q}_r, \mathbf{q}_d \in \mathcal{H}\} \\ \mathcal{H}_d^s &= \{\hat{\mathbf{q}} : \hat{\mathbf{q}} = \mathbf{q}_r + \epsilon \mathbf{q}_d, \mathbf{q}_r, \mathbf{q}_d \in \mathcal{H}^s\} \\ \mathcal{H}_d^v &= \{\hat{\mathbf{q}} : \hat{\mathbf{q}} = \mathbf{q}_r + \epsilon \mathbf{q}_d, \mathbf{q}_r, \mathbf{q}_d \in \mathcal{H}^v\} \\ \mathcal{H}_d^r &= \{\hat{\mathbf{q}} : \hat{\mathbf{q}} = \mathbf{q}_r + \epsilon 0, \mathbf{q}_r \in \mathcal{H}^s\} \end{aligned}$$

Some basic operations with dual quaternions, which will be used in this research, are defined as follows [Filipe and Tsiotras, 2014]. $\hat{\mathbf{a}}, \hat{\mathbf{b}} \in \mathcal{H}_d$ and $\lambda \in \mathfrak{R}$. It is important to familiarize with these operations because they will be often used in this thesis work.

Addition:

$$\hat{\mathbf{a}} + \hat{\mathbf{b}} = (\mathbf{a}_r + \mathbf{b}_r) + \epsilon(\mathbf{a}_d + \mathbf{b}_d) \in \mathcal{H}_d \quad (3.26)$$

Multiplication by a scalar:

$$\lambda \hat{\mathbf{a}} = (\lambda \mathbf{a}_r) + \epsilon(\lambda \mathbf{a}_d) \in \mathcal{H}_d \quad (3.27)$$

Multiplication:

$$\hat{\mathbf{a}}\hat{\mathbf{b}} = (\mathbf{a}_r\mathbf{b}_r) + \epsilon(\mathbf{a}_r\mathbf{b}_d + \mathbf{a}_d\mathbf{b}_r) \in \mathcal{H}_d \quad (3.28)$$

Conjugation:

$$\hat{\mathbf{a}}^* = \mathbf{a}_r^* + \epsilon\mathbf{a}_d^* \in \mathcal{H}_d \quad (3.29)$$

Swap:

$$\hat{\mathbf{a}}^s = \mathbf{a}_d + \epsilon\mathbf{a}_r \in \mathcal{H}_d \quad (3.30)$$

Dot product:

$$\hat{\mathbf{a}} \cdot \hat{\mathbf{b}} = \frac{1}{2}(\hat{\mathbf{a}}^* \hat{\mathbf{b}} + \hat{\mathbf{b}}^* \hat{\mathbf{a}}) = \frac{1}{2}(\hat{\mathbf{a}}\hat{\mathbf{b}}^* + \hat{\mathbf{b}}\hat{\mathbf{a}}^*) = \mathbf{a}_r \cdot \mathbf{b}_r + \epsilon(\mathbf{a}_d \cdot \mathbf{b}_r + \mathbf{a}_r \cdot \mathbf{b}_d) \in \mathcal{H}_d^s \quad (3.31)$$

Cross product:

$$\hat{\mathbf{a}} \times \hat{\mathbf{b}} = \frac{1}{2}(\hat{\mathbf{a}}\hat{\mathbf{b}} - \hat{\mathbf{b}}^* \hat{\mathbf{a}}^*) = \mathbf{a}_r \times \mathbf{b}_r + \epsilon(\mathbf{a}_d \times \mathbf{b}_r + \mathbf{a}_r \times \mathbf{b}_d) \in \mathcal{H}_d^v \quad (3.32)$$

Circle product:

$$\hat{\mathbf{a}} \circ \hat{\mathbf{b}} = \hat{\mathbf{a}}_r \hat{\mathbf{b}}_r + \hat{\mathbf{a}}_d \hat{\mathbf{b}}_d \in \mathcal{H}_d^v \quad (3.33)$$

Dual norm:

$$\|\hat{\mathbf{a}}\|_d^2 = \hat{\mathbf{a}}\hat{\mathbf{a}}^* = \hat{\mathbf{a}}^* \hat{\mathbf{a}} = \hat{\mathbf{a}} \cdot \hat{\mathbf{a}} = (\mathbf{a}_r \cdot \mathbf{a}_r) + \epsilon(2\mathbf{a}_r \cdot \mathbf{a}_d) \in \mathcal{H}_d^s \quad (3.34)$$

Scalar norm:

$$\text{sc}(\hat{\mathbf{a}}) = \text{sc}(\mathbf{a}_r) + \epsilon \text{sc}(\mathbf{a}_d) \in \mathcal{H}_d^s \quad (3.35)$$

Vector part:

$$\text{vec}(\hat{\mathbf{a}}) = \text{vec}(\mathbf{a}_r) + \epsilon \text{vec}(\mathbf{a}_d) \in \mathcal{H}_d^v \quad (3.36)$$

It is important to notice that the dual quaternion multiplication is not commutative. Another definition commonly used is that the dual quaternions $\mathbf{1} + \epsilon\mathbf{0}$ and $\mathbf{0} + \epsilon\mathbf{0}$ are denoted by $\hat{\mathbf{1}}$ and $\hat{\mathbf{0}}$, respectively.

Because the dot product and dual norm of dual quaternions yield, in general, a dual number, the norm of a dual quaternion will be defined as

$$\|\hat{\mathbf{a}}\|^2 = \hat{\mathbf{a}} \circ \hat{\mathbf{a}} \in \mathcal{H}_d^r \quad (3.37)$$

It might be important to notice that the dual quaternion circle product is commutative. The multiplication of a matrix $M \in \mathfrak{R}^{8 \times 8}$ with a dual quaternion $\hat{\mathbf{q}} \in \mathcal{H}_d$ is defined as

$$\mathbf{M} \star \hat{\mathbf{q}} = (\mathbf{M}_{11} \star \mathbf{q}_r + \mathbf{M}_{12} \star \mathbf{q}_d) + \epsilon(\mathbf{M}_{21} \star \mathbf{q}_r + \mathbf{M}_{22} \star \mathbf{q}_d) \in \mathcal{H}_d \quad (3.38)$$

where

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{pmatrix}, \quad \mathbf{M}_{11}, \mathbf{M}_{12}, \mathbf{M}_{21}, \mathbf{M}_{22} \in \mathfrak{R}^{4 \times 4}$$

This definition is analogous to the multiplication of an 8x8 matrix with an eight-dimensional vector. It can be shown that the following properties follow from the previous definitions [Gui and Vukovich, 2015]

$$\hat{\mathbf{a}} \circ (\hat{\mathbf{b}}\hat{\mathbf{c}}) = \hat{\mathbf{b}}^s \circ (\hat{\mathbf{a}}^s \hat{\mathbf{c}}^*) = \hat{\mathbf{c}}^s \circ (\hat{\mathbf{b}}^* \hat{\mathbf{a}}^s) \in \mathfrak{R} \quad \hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}} \in \mathcal{H}_d \quad (3.39)$$

$$\hat{\mathbf{a}} \circ (\hat{\mathbf{b}} \times \hat{\mathbf{c}}) = \hat{\mathbf{b}}^s \circ (\hat{\mathbf{c}} \times \hat{\mathbf{a}}^s) = \hat{\mathbf{c}}^s \circ (\hat{\mathbf{a}}^s \times \hat{\mathbf{b}}) \in \mathfrak{R}, \quad \hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}} \in \mathcal{H}_d^v \quad (3.40)$$

$$\hat{\mathbf{a}} \times \hat{\mathbf{a}} = \hat{\mathbf{0}}, \quad \hat{\mathbf{a}} \in \mathcal{H}_d^v \quad (3.41)$$

$$\hat{\mathbf{a}} \times \hat{\mathbf{b}} = -\hat{\mathbf{b}} \times \hat{\mathbf{a}}, \quad \hat{\mathbf{a}}, \hat{\mathbf{b}} \in \mathcal{H}_d^v \quad (3.42)$$

$$\hat{\mathbf{a}} \circ \hat{\mathbf{b}}^s = \hat{\mathbf{a}} \circ \hat{\mathbf{b}}, \quad \hat{\mathbf{a}}, \hat{\mathbf{b}} \in \mathcal{H}_d \quad (3.43)$$

$$\|\hat{\mathbf{a}}^s\| = \|\hat{\mathbf{a}}\|, \quad \hat{\mathbf{a}} \in \mathcal{H}_d \quad (3.44)$$

$$\|\hat{\mathbf{a}}^*\| = \|\hat{\mathbf{a}}\|, \quad \hat{\mathbf{a}} \in \mathcal{H}_d \quad (3.45)$$

$$(\mathbf{M} \star \hat{\mathbf{a}}) \circ \hat{\mathbf{b}} = \hat{\mathbf{a}} \circ (\mathbf{M}^T \star \hat{\mathbf{b}}), \quad \hat{\mathbf{a}}, \hat{\mathbf{b}} \in \mathcal{H}_d, \quad \mathbf{M} \in \mathbb{R}^{8 \times 8} \quad (3.46)$$

$$|\hat{\mathbf{a}} \circ \hat{\mathbf{b}}| \leq \|\hat{\mathbf{a}}\| \|\hat{\mathbf{b}}\|, \quad \hat{\mathbf{a}}, \hat{\mathbf{b}} \in \mathcal{H}_d \quad (3.47)$$

$$\|\hat{\mathbf{a}}\hat{\mathbf{b}}\| \leq \sqrt{3/2} \|\hat{\mathbf{a}}\| \|\hat{\mathbf{b}}\|, \quad \hat{\mathbf{a}}, \hat{\mathbf{b}} \in \mathcal{H}_d \quad (3.48)$$

Finally, the \mathcal{L}_∞ norm of a function $\hat{u}: [0, \infty) \rightarrow \mathcal{H}_d$ is defined as

$$\|\hat{u}\|_\infty = \sup_{t \geq 0} \|\hat{u}(t)\| \quad (3.49)$$

Moreover, $\hat{u} \in \mathcal{L}_\infty$ if and only if $\|\hat{u}\|_\infty < \infty$.

3.2.1. ATTITUDE AND POSITION REPRESENTATION

The orientation and position of a body frame with respect to the inertial frame can be represented by a unit quaternion $\mathbf{q}_{B/I} \in \mathcal{H}^u$ and by a translation vector $\bar{\mathbf{r}}_{B/I} \in \mathbb{R}^3$ respectively. Alternatively, it can also be represented more compactly by the unit dual quaternion

$$\hat{\mathbf{q}}_{B/I} = \mathbf{q}_{B/I} + \epsilon \frac{1}{2} \mathbf{r}_{B/I}^I \mathbf{q}_{B/I} = \mathbf{q}_{B/I} + \epsilon \frac{1}{2} \mathbf{q}_{B/I} \mathbf{r}_{B/I}^B \quad (3.50)$$

where $\mathbf{r}_{Y/Z} = (\bar{\mathbf{r}}_{Y/Z}^X, 0)$ and $\bar{\mathbf{r}}_{Y/Z}^X = [x_{Y/Z}^X \ y_{Y/Z}^X \ z_{Y/Z}^X]^T$ is the translation vector from the origin of the Z frame to the origin of the Y frame expressed in the X frame. It is important to notice that $\hat{\mathbf{q}}_{B/I}$ is a unit dual quaternion, because it belongs to the set $\mathcal{H}_d^u = \{\hat{\mathbf{q}} \in \mathcal{H}_d : \hat{\mathbf{q}} \cdot \hat{\mathbf{q}} = \hat{\mathbf{q}} \hat{\mathbf{q}}^* = \hat{\mathbf{q}}^* \hat{\mathbf{q}} = \|\hat{\mathbf{q}}\|_d = \hat{\mathbf{1}}\}$.

Moreover, the velocity and the angular velocity (i.e., dual velocity) of a body frame Y with respect to a Z frame expressed in a X frame can also be represented by a unit dual quaternion in a very compact way.

$$\hat{\boldsymbol{\omega}}_{Y/Z}^X = \boldsymbol{\omega}_{Y/Z}^X + \epsilon (\mathbf{v}_{Y/Z}^X + \boldsymbol{\omega}_{Y/Z}^X \times \mathbf{r}_{X/Y}^X), \quad \mathbf{v}_{Y/Z}^X = (\bar{\mathbf{v}}_{Y/Z}^X, 0)$$

where $\boldsymbol{\omega}_{Y/Z}^X$ is the angular velocity of the body frame Y with respect to Z expressed in the body frame X and $\bar{\mathbf{v}}_{Y/Z}^X$ is the linear velocity of the Y frame with respect to the Z frame expressed in the X frame [Filipe and Tsiotras, 2014].

3.2.2. COORDINATE TRANSFORMATIONS

As the reader will see, there are several coordinate transformations in this master thesis, necessary to understand all the derivations. In the case of dual quaternions, coordinate transformations are carried out in a different way than normal vectors. For this reason, if one desired to express the pose of the spacecraft in any other different reference frame, he or she will have to apply the following transformations.

Dual quaternion pose coordinate transformation:

$$\hat{\mathbf{q}}_{B/D} = \hat{\mathbf{q}}_{D/I}^* \hat{\mathbf{q}}_{B/I} \quad (3.51)$$

Dual quaternion velocity coordinate transformation:

$$\hat{\boldsymbol{\omega}}_{B/D}^B = \hat{\boldsymbol{\omega}}_{B/I}^B - \hat{\boldsymbol{\omega}}_{D/I}^B \quad (3.52)$$

where $\hat{\boldsymbol{\omega}}_{D/I}^B$ is the dual quaternion velocity of D with respect to I expressed in the B frame. To expressed in another frame, one has to apply the following transformation

$$\hat{\boldsymbol{\omega}}_{D/I}^B = \hat{\mathbf{q}}_{B/D}^* \hat{\boldsymbol{\omega}}_{D/I}^B \hat{\mathbf{q}}_{B/D} \quad (3.53)$$

4

FLIGHT DYNAMICS AND SPACE ENVIRONMENT

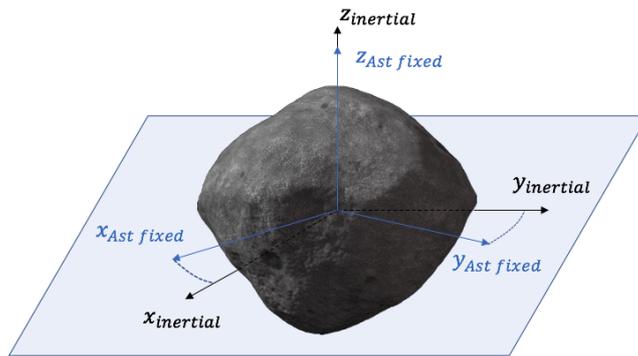
This chapter aims at defining the flight dynamics involved in the simulations that will be carried out in this thesis work and how the space environment will be modeled. First of all however, the three reference frames that will be used are described in Section 4.1. Then, equations of motions for the dual quaternion approach and the classical one are derived in Section 4.2. Finally, all different forces that will determine the trajectory of the spacecraft are modeled in Section 4.3. The torques are modeled in Section 4.4.

4.1. REFERENCE FRAMES

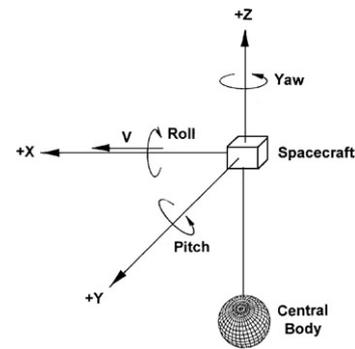
In this thesis work, three different reference frames will be used: the [Asteroid-Centered Inertial \(ACI\)](#) reference frame, the [Asteroid-Centered Asteroid-Fixed \(ACAF\)](#) reference frame and the spacecraft body fixed frame. The first one will be used to express all variables for integration of the equations of motion. It is always easier to work in an inertial reference frame. The second one (ACAF) will be used for representing the gravity field of Bennu. In this case, the gravity field will move with the asteroid itself; therefore, it is desirable to express it in a reference frame that moves with it. Finally, the last one (body-fixed frame) will be used to express all the relative errors and attitude orientations of the spacecraft.

4.1.1. ASTEROID-CENTERED INERTIAL

The [ACI](#) reference frame has its origin in the center of mass of the asteroid and its set of Cartesian axes point in the same direction as the [International Celestial Reference Frame \(ICRF\)](#). The [ACI](#) frame is not a 'true' inertial reference frame because its origin has an acceleration about the Sun. However, this acceleration is negligible compared to the accelerations acting on the satellite problem. For that reason, it is reasonable to accept it as



(a) ACI reference frame (black) and ACAF reference frame
(blue)



(b) Representation of body-fixed frame [Markley and
Crassidis, 2014]

an inertial reference frame. Figure 4.1a shows a sketch in black. As can be seen, this inertial reference frame does not rotate with the asteroid and remains fixed in space.

4.1.2. SPACECRAFT BODY-FIXED

Another important reference frame for this research is the spacecraft body-fixed frame, defined by an origin at the center of mass of the spacecraft and three Cartesian axes that move together with it. For this work, the set of axes are chosen to be the three principal axes of inertia of the spacecraft. Figure 4.1b shows a sketch of this reference frame. As can be seen, rotations about the x axis will be called rolling; rotations about the y axis will be called pitching; and rotation about the z axis will be called yawing.

4.1.3. ASTEROID-CENTERED ASTEROID-FIXED

The ACAF reference frame is similar to the body frame but instead of having the origin at the center of mass of the spacecraft it has its origin at the center of mass of the target asteroid. Then a set of three Cartesian axes rotate together with the asteroid. They are the three principal axes of inertia of the asteroid. It is important to mention that this reference frame is completely non inertial due to the rotation of the asteroid and its accelerated center of mass. Figure 4.1a shows a sketch in blue. The discontinuous blue lines represent the rotation of these axes with respect to the inertial reference axes (represented in black). As can be seen, they rotate together with the principal axes of inertia of the asteroid.

4.2. EQUATIONS OF MOTION

The equations of motion to be solved in this work are stated in this section. First, the classical equations of motions will be defined. There are four classical equations of motion. Two for the translational motion and two for the rotational motion expressing the rate of change of every state variable. Second, the equations of motion expressed in the dual quaternion representation will be presented. In this case however, there are

only two equations of motion. One for the rate of change of the dual quaternion pose and another one for the dual velocity. All the equations are expressed with respect to the **ACI** reference frame, because it is an inertial reference frame, thus, the equations of motion are simpler, easier and faster to integrate numerically.

4.2.1. CLASSICAL FORMULATION

In the classical formulation, position, velocity and angular velocity are represented by three-dimensional vectors. The attitude is represented with a unit quaternion. The derivative of the position in an inertial frame can be expressed as in Equation (4.1). As can be seen, it is simply the velocity of the spacecraft with respect to the inertial frame.

$$\dot{\mathbf{r}}_{BI}^I = \mathbf{v}_{BI}^I \quad (4.1)$$

where \mathbf{r}_{BI}^I is the position of the spacecraft with respect to the inertial frame, expressed in the inertial frame and \mathbf{v}_{BI}^I is the same but with the velocity. Applying the second Newton's law, the derivative of the velocity with respect to time can be defined as

$$\dot{\mathbf{v}}_{BI}^I = \mathbf{a}_{BI}^I = \frac{\mathbf{F}_B^I}{M} \quad (4.2)$$

where \mathbf{a}_{BI}^I is the acceleration of the spacecraft with respect to the inertial reference frame expressed in this same inertial reference frame. This acceleration is dictated by the forces acting on the spacecraft over time. For this reason, F_B^I are the external forces acting on the spacecraft expressed in the **ACI** frame. M is the mass of the spacecraft. It is very important that all variables are expressed in the same reference frame. Next, the derivative of the attitude expressed with quaternions with respect to time can be written

$$\dot{\mathbf{q}}_{BI} = \frac{1}{2} \mathbf{B}(\mathbf{q}_{BI}) \boldsymbol{\omega}_{BI}^B \quad (4.3)$$

where \mathbf{q}_{BI} represents the quaternion rotation of the spacecraft with respect to the inertial axes and $\boldsymbol{\omega}_{BI}^B$ is the angular velocity of the spacecraft with respect to **ACI** expressed in body frame. Finally, $\mathbf{B}(\mathbf{q}_{BI})$ is just

$$\mathbf{B}(\mathbf{q}) = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \quad (4.4)$$

Last but not least, the time derivative of the angular velocity is expressed in Equation (4.5). This equation is also known as the rotational Euler equation of motion, which dictates how torques affect the angular velocity of bodies, and thus, their orientation.

$$\dot{\boldsymbol{\omega}}_{BI}^B = (\mathbf{I}^B)^{-1} [\boldsymbol{\tau}^B - \boldsymbol{\omega}_{BI}^B \times (\mathbf{I}^B \boldsymbol{\omega}_{BI}^B)] \quad (4.5)$$

where \mathbf{I}^B is the inertia tensor of the spacecraft expressed in the body-fixed frame, $\boldsymbol{\tau}^B$ are the external torques acting on the spacecraft over time expressed in the body-fixed frame and $\boldsymbol{\omega}_{BI}^B$ is the angular velocity of the spacecraft with respect to the inertial frame expressed in the body frame [Markley and Crassidis, 2014]. It is important to mention that this equation of motion is only valid for rigid bodies. This means that the satellite has to be assumed with a constant shape and without allowing deformations.

4.2.2. DUAL QUATERNION FORMULATION

In contrast with the classical formulation, the position and the attitude of the spacecraft in the dual quaternion formulation are expressed with the same variable; an eight-dimensional dual quaternion. The same happens with the linear and angular velocities. From now on, they will be addressed as dual pose and dual velocity of the spacecraft. The time derivative of the dual pose is

$$\dot{\hat{\mathbf{q}}}_{B/I} = \frac{1}{2} \hat{\mathbf{q}}_{B/I} \hat{\boldsymbol{\omega}}_{B/I}^B \quad (4.6)$$

where $\hat{\mathbf{q}}_{B/I}$ is the dual pose of the spacecraft and represents both the attitude and the position at the same time. It is expressed with respect to the **ACI** and expressed in the body-fixed frame. $\hat{\boldsymbol{\omega}}_{B/I}^B$ is the so called dual velocity and it represents the linear and angular velocities of the spacecraft with respect to an inertial frame and expressed in the body-fixed frame. The derivative of this dual velocity with respect to time is

$$(\dot{\hat{\boldsymbol{\omega}}}_{B/I}^B)^s = (\mathbf{M}^B)^{-1} \star \left[\hat{\mathbf{f}}^B - \hat{\boldsymbol{\omega}}_{B/I}^B \times \{ \mathbf{M}^B \star (\hat{\boldsymbol{\omega}}_{B/I}^B)^s \} \right] \quad (4.7)$$

where \mathbf{M}^B is the dual inertia matrix representing both the mass and the inertia of the spacecraft, $\hat{\mathbf{f}}^B$ is the external dual force expressed in body frame, which represents the torques and forces acting on the spacecraft, expressed in the body fixed-frame. As can be seen, the equations of motion in the dual quaternion form (4.6) (4.7) are identical to the rotational ones in the classical approach expressed in quaternions (4.3) (4.5). It is very important to bear in mind all the dual quaternion operations defined in Chapter 3, such as the swap operator or the matrix multiplication with a dual quaternion. Also, it might be worth mentioning that only two equations of motion are needed with dual quaternions to express exactly the same that the classical method with four. Everything is more compact with dual quaternions.

4.3. FORCE MODELS

As can be seen from the equations of motion, the forces acting on the spacecraft will determine the motion of the spacecraft. This section describes the forces that will interact with the spacecraft and how they can be modeled. In real scenarios, many different forces have to be taken into account for precise orbit determination, however, for this thesis work, only the most relevant ones have been modeled. Figure 4.2 shows the different forces (in km/s^2) acting on the real **OSIRIS-REx** mission during close proximity operation phases. As can be seen, the most relevant ones during proximity operations are the main body gravity and the **SRP**. Then, depending on the phase of the mission, the third body effects from the Sun and the infrared forces come into play. The first ones are two orders of magnitude, or sometimes even up to four, larger than the second ones. The grey line represents the gravity forces due to the oblateness of Bennu, and the pink one represents the infrared forces generated due to the different temperature parts of the spacecraft. Finally, the red line represents the albedo forces due to the radiation reflected by Bennu and impacting on the spacecraft. In this thesis, only the gravitational, the **SRP** and the Sun third body forces will be modeled. Since no thermal analysis will be carried out, it is out of the scope to model the infrared radiation. The same applies with the albedo forces, they are out of the scope of this master thesis. However, this is still a reasonable modeling because these forces are from 4 to 6 orders of magnitude smaller than the modeled ones.

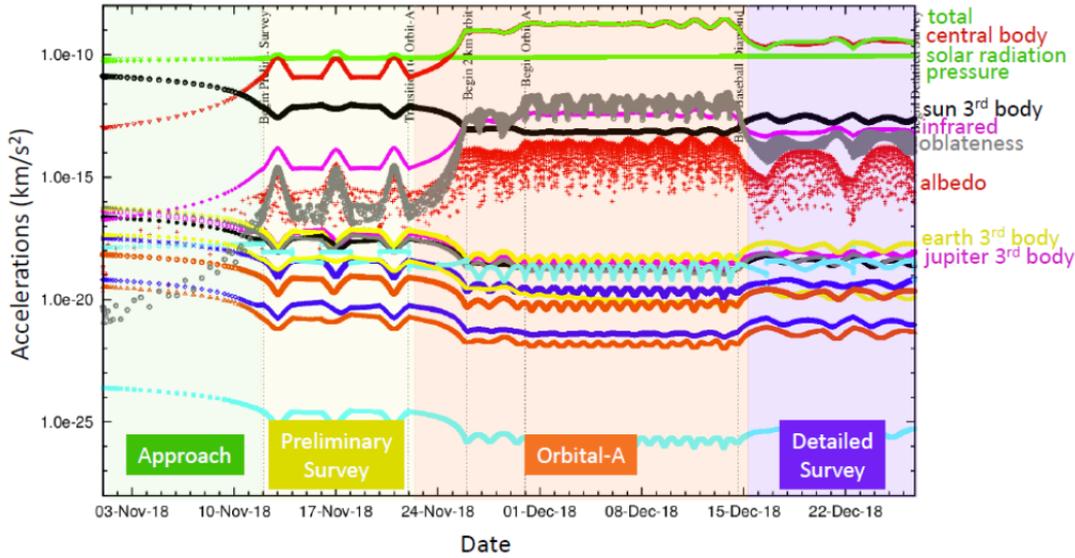


Figure 4.2: Different forces acting on OSIRIS-REx spacecraft on different phases. [Everett et al., 2017]

4.3.1. GRAVITY FORCE

Gravity representations may be split into four categories [Jones, 2010]: a series expansion of orthogonal functions, a linear combination of potential functions, linear combinations of functions defined via splines, kernels or finite elements, and collocation methods using minimum norm or least squares. The latter one most accurately represent local variations in the gravity field, but the model is not optimal for global variations.

The combination of nodes define the gravity field, or some derived quantity, as a collection of nodes with interpolation performed between them for evaluation of the gravity field. Typically, these models provide fast evaluation or estimation capabilities [Jones, 2010]. Other options are to use B-splines defined on the surface of a sphere, polynomials on subdivisions of the surface of the cube or B-splines on the surface of a cube. The last one is the cubed-sphere model, which combines the benefits of the first two. Their killing drawbacks are the computational effort required and the amount of memory necessary to store all the coefficients.

Models using a linear combinations of potentials usually assume a collection of point masses, i.e. the sum of individual gravity potentials yields the total potential of a system. However, other forms exists. For example, [Werner and Scheeres 1997] present a model for evaluating the gravity field of any general polyhedron. An irregularly shaped body, such as an asteroid, may be represented by a polyhedron to some precision. Formulating the gravity potential as a volume integral of differential masses, the gravity field may be represented by a linear combination of surface integrals on each face and edge of the polyhedron. This model satisfies Laplace's equation at any point outside of the body, with accuracy limited by an assumption of constant density and the accuracy of the shape representation. Instead of a single polyhedron, [Scheeres et al. 2000] defined the body as a collection of polyhedra, allowing for each polyhedron to have a different, but individually uniform, density. Unfortunately, the polyhedron model is computational intensive [Jones, 2010]. Its major drawback is the computational effort needed to add up all these potentials functions.

Finally, the spherical harmonic model represents the gravity field as a linear combination of an infinite number of orthogonal functions on the sphere. These functions use the associated Legendre functions of the first kind for a given degree and order. The frequency of each function increases with degree, which implies a smaller wavelength. These shorter wavelengths correspond to smaller areas on the sphere. Thus, high-degree models imply increased spatial resolution of the model. This method can guarantee to converge to the correct gravity field outside of the circumscribing sphere. They can be truncated at finite order to size the accuracy in representation with the accuracy needed in modeling, and a rich literature exists for the computation and evaluation of such expansions [Jones, 2010]. They allow to compute an accurate enough gravity field with a very cheap computational price and time. For these reasons and for the sake of simplicity, the gravity force in this master thesis will be modeled using spherical harmonics.

However, the spherical harmonics have some drawbacks and limitations to bear in mind. The major one is that the same form of the exterior harmonic expansion is no longer guaranteed to converge inside the circumscribing sphere, and indeed often diverges [Werner and Scheeres, 1997]. In the case of Bennu, since it is a very rounded body, the circumscribing sphere is very close to the surface and has no major effects on the solution. However, for other asteroids, this would not be the case. Takahashi [2013] compared the gravity field computed with a polyhedron model and with spherical harmonics for the case of Bennu in his PhD thesis. One of his results was that the surface accelerations errors (in %) of the spherical harmonic method compared to the polyhedron model are 0.74 in average, with a maximum of 10.95 using a 4 degree model. This level of accuracy is good enough for this master thesis. Another drawback is that the harmonic expansion yields no information about whether a field point is outside or inside the body. In a spacecraft simulation, a separate algorithm must be used to detect this important geometric condition.

The gravity force can then be modeled with an spherical harmonic approximation using Kaula normalized coefficients. The gravitational potential function can be written as [Gottlieb, 1993]

$$U = \frac{\mu}{r} + \sum_{n=0}^{\infty} \sum_{m=0}^n \frac{\mu}{r} \left(\frac{\alpha_e}{r}\right)^n P_{nm}(\epsilon) (C_{n,m} \cos(m\lambda) + S_{nm} \sin(m\lambda)) \quad (4.8)$$

where μ is the gravitational parameter, α_e is the reference equatorial radius and r is the magnitude of the position vector \mathbf{x} . P_{nm} are the associated Legendre functions with its Legendre polynomials. Also, ϵ is the latitude of the position vector and λ its longitude. The C_{nm} and S_{nm} are the unnormalized cosine and sine gravity coefficients that results from the mass distribution of the asteroid. The use of normalized coefficients is justified because they do not require the computation of terms in the order of magnitude $(n + m)!$.

Then, the gravitational acceleration vector \mathbf{g} can be calculated as minus the first partial derivative of U with respect to the asteroid-fixed position vector \mathbf{x} (Equation 4.9). Thus, the rotation of Bennu will have to be taken into account.

$$\mathbf{g} = -\frac{\partial U}{\partial \mathbf{x}} \quad (4.9)$$

The normalized spherical harmonics coefficients of constant density for Bennu that will be used in the simulations are listed in Table 4.1. The frame is centered at the center of the asteroid and aligned with its

Table 4.1: Normalized spherical harmonic coefficients of constant density (1260 kg/m³) for Bennu [McMahon et al., 2018]

Degree (n)	Order (m)	C Coeff (C _{nm})	S Coeff (S _{nm})
0	0	1	-
1	0	0	-
1	1	0	0
2	0	-0.0175112	-
2	1	-0.0000023	0
2	2	0.0058194	-0.0000197
3	0	0.0056102	-
3	1	0.0015471	0.0015368
3	2	0.0001115	0.0000635
3	3	0.0026660	-0.0009332
4	0	0.0102498	-
4	1	0.0004360	0.0018562
4	2	-0.0021919	0.0007749
4	3	-0.0010761	0.0001024
4	4	0.0021356	0.0030684
5	0	-0.0013767	-
5	1	0.0005161	-0.0000786
5	2	0.0005464	-0.0012414
5	3	0.0004250	0.0002269
5	4	0.0013801	-0.0005038
5	5	0.0004869	0.0005241
6	0	-0.0024871	-
6	1	0.0011514	-0.0000084
6	2	0.0007281	-0.0002936
6	3	0.0013361	-0.0009706
6	4	-0.0005090	-0.0006485
6	5	0.0001236	-0.0005993
6	6	0.0007195	-0.0000500

principal moments of inertia. The coefficients are Kaula normalized and are computed for a reference radius of 245.887 m, which is the radius of the sphere with equivalent volume as the shape model.

It is important to always bear in mind the limitations and consequences of each model. In this case, the reason for stopping at degree and order 6 is because no more data was available. As has been explained, all the information from Bennu comes from ground-based measurements. Bennu has not been visited yet. And these measurements cannot give more resolution of the gravity field. The consequences of this fact are a more central-body gravity field without abrupt changes in gravity. Finally, as can be seen in Table 4.1, the assumption of constant density, combined with the rounded shape model of Bennu, result in high order coefficient of the table very small. This as well, makes the gravity field more uniform.

4.3.2. SOLAR RADIATION FORCE

A variety of models has been used to study solar radiation for various applications. The simplest model, which has been used for most analytical studies to date, states that the force caused by solar radiation pressure acts along the object-Sun line, directly away from the Sun. In the spacecraft community, this model is often re-

ferred to as the cannonball model, because this is how solar radiation would affect a cannonball-like spherical object with a uniform optical properties in space. This model is used as the baseline for many missions and has also been applied to the debris problem [McMahon, 2011]. One of the advantages for the cannonball model is that its simplicity allows it to be expressed as a potential disturbing function, which allows easy analytical investigation of the perturbative effects through the use of Lagrange's planetary equations.

Spacecraft missions often desire much higher fidelity in their solar radiation pressure model than the cannonball model. The next step is still a simple model characterized by a small number of facets to capture the general geometry of the spacecraft. Many spacecraft can be modeled as a "box-wing" model. With a small number of facets, it is still tractable to approach the problem analytically. However, the solar radiation pressure force is computed per fact for each Sun position as needed, which requires a numerical implementation. As the models of the spacecraft become more and more detailed, they are more and more reliant on numerical methods and are referred to as micro models. These micro models can be extremely detailed, approaching full CAD models with the shape of the spacecraft reproduced and each component being assigned its own optical properties. The effects of solar radiation pressure for such high fidelity models is determined through extremely detailed ray-tracing methods which can include all self-shadowing and secondary incidence effects [McMahon, 2011].

Since the purpose of this study is to focus on the control part using dual quaternions and not to highly represent the Solar Radiation Pressure, a simple "box-wing" model will be use to compute the SRP forces. This means that the SRP forces will depend on the actual orientation of the spacecraft, making even stronger coupling between translational and rotational motions. The acceleration due to the SRP can then be modeled as

$$\mathbf{a}_{SRP} = C_r \frac{A(\mathbf{q})}{M} \frac{P_\Phi}{R^2} \hat{\mathbf{D}} \quad (4.10)$$

where C_r is the coefficient of reflectivity, M is the mass of the spacecraft, $A(\mathbf{q})$ is the wet area, which depends on the orientation of the spacecraft and needs to be computed every time step. P_Φ is the solar-radiation pressure at the Sun's surface ($1 \cdot 10^{14}$ kg·km/s²) and R here is the distance from the spacecraft to the Sun. Finally, $\hat{\mathbf{D}}$ is the unit vector pointing from the Sun to the spacecraft [McMahon et al., 2018].

It is important to mention that no eclipses or radiation coming from the asteroid (albedo) will be modeled. The eclipse assumption is totally valid and accurate because the terminator orbit will not enter any eclipse. Moreover, the hovering and landing will always be in a fully illuminated part of the asteroid.

4.3.3. SOLAR TIDES (3RD BODY FORCE)

In this thesis work, 3rd body effects refer to the acceleration produced by the Sun, which will be modeled as

$$\mathbf{a}_{ST} = -\frac{\mu_{Sun}}{|\mathbf{r}_{sat} + \mathbf{d}|^3} (\mathbf{r}_{sat} + \mathbf{d}) + \frac{\mu_{Sun}}{|\mathbf{d}|^3} \mathbf{d} \quad (4.11)$$

μ_{Sun} is the gravitational parameter of the Sun, \mathbf{r}_{sat} is the position vector in the ACI frame and \mathbf{d} is the position vector pointing from the Sun to the center of the asteroid. It is important to remark here that the second term

is due to the fact that this acceleration is expressed in the asteroid centered inertial frame.

4.4. GRAVITY GRADIENT TORQUE MODEL

Apart from the previous mentioned forces acting on the spacecraft, the simulations will also take into account the gravity gradient torque. The vast majority of studies compute the gravity gradient torque with the following equation, as explain [Schaub and Junkins \[2009\]](#) in their book.

$$\boldsymbol{\tau} = 3 \frac{\mu}{R_c^3} \mathbf{a}_3 \times \mathbf{I} \mathbf{a}_3 \quad (4.12)$$

where R_c is the distance from the center of the attracting body (asteroid), μ is its gravitational parameter, \mathbf{a}_3 is the third component of the Local Vertical Local Horizontal frame, expressed in body frame, and \mathbf{I} is the inertia moment tensor of the spacecraft. However, Equation (4.12) 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 center of the attracting body. This assumption is not valid at all in the case of small asteroids with irregular shapes and mass distributions [[Wang and Xu, 2013](#)]. For this reason, a numerical approach is necessary. In this thesis work, it will be obtained through the spherical harmonics gravity field. It is not a central-body gravity field and it could be as accurate as the spherical harmonics can be. As explained by [Gottlieb \[1993\]](#), the assumption of this method is that the gravity gradient torque can be computed assuming that the gravity varies linearly about the center of mass of the satellite. This means that the gravity at the vicinity of a point is given by the expression

$$\mathbf{g} = \mathbf{g}_{cm} + \frac{\partial \mathbf{g}}{\partial \mathbf{r}} \delta \mathbf{r} \quad (4.13)$$

where \mathbf{g}_{cm} is the gravity acceleration at the center of mass of the satellite and \mathbf{r} is any arbitrary vector with its origin at the center of mass of the satellite. Then, the gravity gradient torque can be modeled as

$$\boldsymbol{\tau} = \int \boldsymbol{\rho} \times \mathbf{G} \boldsymbol{\rho} dm = \begin{bmatrix} g_{23}(I_{zz} - I_{yy}) + g_{13}I_{xy} - g_{12}I_{xz} + I_{yz}(g_{33} - g_{22}) \\ g_{13}(I_{xx} - I_{zz}) - g_{23}I_{xy} + g_{12}I_{yz} + I_{xz}(g_{11} - g_{33}) \\ g_{12}(I_{yy} - I_{xx}) + g_{23}I_{xz} - g_{13}I_{yz} + I_{xy}(g_{22} - g_{11}) \end{bmatrix} \quad (4.14)$$

where I_{ij} are the different components of the inertia tensor (in body frame) and g_{ij} are just the coefficients of the matrix \mathbf{G} defined by

$$\mathbf{G} = \mathbf{B}^T \frac{\partial \mathbf{g}}{\partial \mathbf{r}} \mathbf{B} \quad (4.15)$$

where \mathbf{g} is the gravity vector at the center of mass of the satellite (so its derivative is just the second derivative of the gravity potential mentioned in the previous section with respect to the position vector). And the \mathbf{B} matrix is just the rotation matrix that transforms from vector from body fixed coordinate frame to the asteroid fixed reference frame.

This method has been compared with a full numerical simulation of a box-shaped spacecraft orbiting about Bennu in close proximity. The full numerical simulation computes the torques assuming that the

spacecraft is a combination of point masses (up to 262144). Furthermore, it takes into account the full resolution of the gravity field without assuming linear variations from the center of mass of the satellite to all of its points masses. The results show that the maximum relative errors between the methods are between 3 and 5% for the worst cases depending on the body axis. This errors are totally affordable taking into account the amount of time saved by applying the [Gottlieb \[1993\]](#) method.

5

SPACECRAFT CONTROL

To follow the desired trajectories for each scenario of this thesis, attitude and orbit control will clearly be required. There are many different ways and possibilities on how to control spacecraft. Depending on the selected control laws or even depending on the selected gains, the spacecraft will behave differently to compensate for disturbances and bring it back to its reference trajectory. Finding the best controller for each case has occupied the minds of many engineers for hundred years, and it is still a problem nowadays.

One of the key points when selecting a control law is stability; specially in the field of autonomous navigation around asteroids. It is always desirable to guarantee that your controller will always bring your spacecraft down to reference trajectory and succeed in your mission. However, still nowadays, global stability cannot be guaranteed without making some assumptions. For instance, a very common assumption is to assume perfect knowledge of your spacecraft (mass and inertia). For most of the cases, they are well known with sufficient precision to guarantee stability. However, sometime, spacecraft have liquid propellant tanks with fuel slushing and changing the inertia of the spacecraft all the time. This is very difficult and costly to model. If the stability requirements are very restrictive, they cannot always be guaranteed.

Bearing these limitations in mind, this chapter defines the controllers that will bring the spacecraft to follow the desired trajectories for this master thesis. Section 5.1 defines the control approach that will be used for the classical representation, whereas Section 5.2, defines the one for the dual quaternion representation. For both of them, proof of Lyapunov stability is derived. Then, a process for gain selection and optimization is explained in Section 5.3. The final selected gains for each scenario are stated at the end of this chapter.

5.1. CLASSIC CONTROL

This section first defines the attitude control scheme that will be applied to control the spacecraft in the simulations by following the development explained in [Schaub and Junkins, 2009]. Then the orbit control is defined with a similar approach.

5.1.1. ATTITUDE CONTROL

The selected attitude controller will be a feedback control very well suited for tracking reference trajectories. It will guarantee the stability of the rotational motion. Defining $\delta\boldsymbol{\omega}$ as the error in angular velocity (Equation (5.1)) with respect to a desired reference angular rate ($\boldsymbol{\omega}_r$), and \boldsymbol{q} as the attitude error to some reference orientation; it is possible to create a Lyapunov function.

$$\delta\boldsymbol{\omega} = \boldsymbol{\omega} - \boldsymbol{\omega}_r \quad (5.1)$$

The Euler's rotational equations of motion for this problem is given by

$$\mathbf{I}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} + \mathbf{u} + \mathbf{L} \quad (5.2)$$

Taking into account the Lyapunov theory explained in Appendix A, a positive definite function, radially unbounded and that represents some how the relative state error to the reference trajectory has to be defined. A possible candidate can be the function defined by

$$V(\delta\boldsymbol{\omega}, \boldsymbol{q}) = \frac{1}{2}\delta\boldsymbol{\omega}^T \mathbf{I}\delta\boldsymbol{\omega} + K(\boldsymbol{q} - \mathbf{1})^T (\boldsymbol{q} - \mathbf{1}) \quad (5.3)$$

The parameter K is a positive scalar attitude feedback gain and $\mathbf{1}$ represents the zero attitude quaternion (1, 0, 0, 0). The derivative of the Lyapunov function can be expressed as

$$\dot{V} = \delta\boldsymbol{\omega}^T \left(\mathbf{I} \frac{d}{dt} \Big|_B (\delta\boldsymbol{\omega}) + K\boldsymbol{\epsilon} \right) \quad (5.4)$$

where $\boldsymbol{\epsilon}$ is just the vector part of the error quaternion defined as $\boldsymbol{\epsilon} = (q_1, q_2, q_3)^T$. It is important to see that the derivative has to be evaluated in a body-fixed frame. Now, forcing this derivative to be negative semi-definite by setting it equal to

$$\dot{V} = -\delta\boldsymbol{\omega}^T \mathbf{P}\delta\boldsymbol{\omega} \quad (5.5)$$

where P is the positive definite angular velocity feedback gain matrix, this leads to the following stability constraint:

$$\mathbf{I} \frac{d}{dt} \Big|_B (\delta\boldsymbol{\omega}) + K\boldsymbol{\epsilon} + \mathbf{P}\delta\boldsymbol{\omega} = 0 \quad (5.6)$$

Now, making use of the local derivative, applying the transport theorem, and introducing the dynamic equation of motion (5.2), this stability condition can be expressed as

$$K\boldsymbol{\epsilon} + \mathbf{P}\delta\boldsymbol{\omega} - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} + \mathbf{L} + \mathbf{u} - \mathbf{I}(\dot{\boldsymbol{\omega}}_r - \boldsymbol{\omega} \times \boldsymbol{\omega}_r) = 0 \quad (5.7)$$

\mathbf{L} represents the total external torques acting on the spacecraft except for the control torque \mathbf{u} . The nonlinear feedback control is then given by

$$\mathbf{u} = -K\boldsymbol{\epsilon} - P\delta\boldsymbol{\omega} + \boldsymbol{\omega} \times I\boldsymbol{\omega} + I(\dot{\boldsymbol{\omega}}_r - \boldsymbol{\omega} \times \boldsymbol{\omega}_r) - \mathbf{L} \quad (5.8)$$

On basis of the Lyapunov theory explained in Appendix A, it can be guaranteed that this feedback control will stabilize the rotational motion to the reference trajectory. It is important to bear in mind that it has been assumed perfectly known forces acting on the spacecraft, perfectly known inertia and mass, and perfectly known orientation and angular rates.

5.1.2. POSITION CONTROL

The selected position controller will be very similar to the attitude one. It will also guarantee stability and it is very well suited for tracking reference trajectories. To start with, a good candidate Lyapunov function, positive definite, radially unbounded and representing the state error relative to a reference trajectory is expressed as

$$V(\mathbf{r}, \mathbf{v}) = \frac{1}{2}\mathbf{v}^T\mathbf{v} + \frac{1}{2}k_p\mathbf{r}^T\mathbf{r} \quad (5.9)$$

where \mathbf{r} represents here the position of the spacecraft with respect to the reference position and \mathbf{v} represents the relative velocity. The parameter K_p is a positive scalar position feedback gain. The derivative of this function with respect to time can be expressed as

$$\dot{V} = \mathbf{v}^T\dot{\mathbf{v}} + k_p\mathbf{r}^T\mathbf{v} \quad (5.10)$$

As was done in the attitude section, this function derivative has to be negative semi-definite so that stability can be guaranteed. Setting this function equal to

$$\dot{V} = -k_v\mathbf{v}^T\mathbf{v} \quad (5.11)$$

which is always negative and where k_v is a positive scalar velocity feedback gain. Forcing both function derivatives to be equal leads to the stability condition

$$\mathbf{v}^T\dot{\mathbf{v}} + k_p\mathbf{r}^T\mathbf{v} + k_v\mathbf{v}^T\mathbf{v} = 0 \quad (5.12)$$

Now, substituting the equation of motion (for $\dot{\mathbf{v}}$), set by Newton's second law taking into account a control force, the stability condition can be re-written as

$$\mathbf{v}^T\frac{\mathbf{F}}{M} + \mathbf{v}^T\frac{\mathbf{F}_c}{M} - \mathbf{v}^T\mathbf{a}_d + k_p\mathbf{r}^T\mathbf{v} + k_v\mathbf{v}^T\mathbf{v} = 0 \quad (5.13)$$

where \mathbf{F} are the external forces acting on the spacecraft, \mathbf{F}_c are the control forces, M is the mass of the spacecraft and \mathbf{a}_d is the desired acceleration. This finally leads to a control force given by

$$\mathbf{F}_c = -K_p\mathbf{r} - K_v\mathbf{v} + M\mathbf{a}_d - \mathbf{F} \quad (5.14)$$

And therefore, making use of Lyapunov theory explained in Appendix A, it can be proven that this controller can guarantee position and velocity Lyapunov stability.

5.2. DUAL QUATERNION CONTROL

Assuming that the systems dynamics are perfectly known, this sections defines a PD controller based on the work done by Gui and Vukovich [2015] for controlling both the position and orientation of the spacecraft simultaneously with dual quaternions. The proposed controller is defined by

$$\hat{\mathbf{f}}_c^B = -(\hat{k}_1 \odot \hat{\mathbf{p}}_{B/D}^B)^s - (\hat{k}_2 \odot \hat{\boldsymbol{\omega}}_{B/D}^B)^s + \hat{\boldsymbol{\omega}}_{D/I}^B \times \mathbf{M}^B \hat{\boldsymbol{\omega}}_{D/I}^B + \mathbf{M}^B \left(\hat{\mathbf{q}}_{B/D}^* \hat{\boldsymbol{\omega}}_{D/I}^D \hat{\mathbf{q}}_{B/D} \right) - \hat{\mathbf{f}}_{ext}^B \quad (5.15)$$

where $\hat{k}_1 = k_q + \epsilon 2k_r$ and $\hat{k}_2 = k_\omega + \epsilon k_v$ are dual numbers gains ($k_q, k_r, k_\omega, k_v \in \mathbb{R}^+$). The super-indexes always stand for the reference frame in which each variable is expressed, whereas the sub-indexes tell with respect to which frame each variable is expressed. The \odot operation stands for the multiplication between a dual number and a dual quaternion:

$$\hat{c} \odot \hat{\mathbf{a}} = (c_r + \epsilon c_d) \odot (\mathbf{a}_r + \epsilon \mathbf{a}_d) = c_r \mathbf{a}_r + \epsilon c_d \mathbf{a}_d \quad (5.16)$$

$\hat{\mathbf{f}}_c^B$ is just the dual quaternion control force. $\hat{\mathbf{p}}_{B/D}^B$ is the dual quaternion pose error of the spacecraft with respect to the reference trajectory, expressed in body frame.

$$\hat{\mathbf{p}}_{B/D}^B = \mathbf{q}_{B/D} + \epsilon \frac{1}{2} \mathbf{r}_{B/D}^B \quad (5.17)$$

$\mathbf{q}_{B/D}$ is the orientation quaternion error with respect to the reference orientation, $\mathbf{r}_{B/D}^B$ is the position error relative to the reference position expressed in body frame, $\hat{\boldsymbol{\omega}}_{B/D}^B$ is the same but with angular velocity, \mathbf{M}^B is the so called dual inertial matrix combining information of the spacecraft's mass and inertia; and $\hat{\mathbf{f}}_{ext}^B$ is the total dual quaternion forces acting on the spacecraft.

It is important to note that the $\hat{\mathbf{p}}_{B/D}^B$ and $\hat{\boldsymbol{\omega}}_{B/D}^B$ feedbacks ensure convergence of the tracking error, while the remaining terms involved are used to compensate for the dual forces acting on the spacecraft.

As can be seen, controller (5.15) has exactly the same form as classical attitude controller (5.8), with all the equivalent terms. The reason behind the selection of these controllers is to make them comparable. It is important to bear in mind that the goal of this master thesis is to compare the dual quaternion representation with the classical one. Therefore, the controllers have to be similar. Otherwise, certain advantages could be associated with the control approach and not with the state variable representations.

The stability of this controller can be proven with Lyapunov theory. Equation (5.18) is a positive definite Lyapunov function, radially unbound and expressing the state dual quaternion error with respect to the reference trajectory.

$$V(\hat{\mathbf{q}}_{B/D}, \hat{\boldsymbol{\omega}}_{B/D}^B) = (\hat{k}_1 \odot (\hat{\mathbf{q}}_{B/D} - \hat{\mathbf{1}})) \circ (\hat{\mathbf{q}}_{B/D} - \hat{\mathbf{1}}) + \frac{1}{2} (\hat{\boldsymbol{\omega}}_{B/D}^B)^s \circ (\hat{\mathbf{M}} \hat{\boldsymbol{\omega}}_{B/D}^B) \quad (5.18)$$

which satisfies that is always positive and $V(\hat{\mathbf{1}}, \hat{\mathbf{0}}) = 0$. Noting that $(\hat{k}_1 \odot (\hat{\mathbf{q}}_{B/D} - \hat{\mathbf{1}})) \circ \hat{\mathbf{q}}_{B/D} + (\hat{k}_1 \odot \hat{\mathbf{q}}_{B/D}) \circ (\hat{\mathbf{q}}_{B/D} - \hat{\mathbf{1}}) = 2(\hat{k}_1 \odot (\hat{\mathbf{q}}_{B/D} - \hat{\mathbf{1}})) \circ \hat{\mathbf{q}}_{B/D}$, and taking into account that $\hat{\mathbf{a}} \circ \hat{\mathbf{b}} = \mathbf{a}_r \mathbf{b}_r + \mathbf{a}_d \mathbf{b}_d$; the derivative of this Lyapunov function can be computed along the close-loop trajectory as

$$\begin{aligned} \dot{V} &= 2(\hat{k}_1 \odot (\hat{\mathbf{q}}_{B/D} - \hat{\mathbf{1}})) \circ \hat{\mathbf{q}}_{B/D} + (\hat{\boldsymbol{\omega}}_{B/D}^B)^s \circ (\hat{\mathbf{M}} \hat{\boldsymbol{\omega}}_{B/D}^B) \\ &= -(\hat{\boldsymbol{\omega}}_{B/D}^B)^s \circ (\hat{k}_2 \odot \hat{\boldsymbol{\omega}}_{B/D}^B)^s - (\hat{\boldsymbol{\omega}}_{B/D}^B)^s \circ (\hat{k}_1 \odot \hat{\mathbf{p}}_{B/D}^B)^s + (\hat{k}_1 \odot (\hat{\mathbf{q}}_{B/D} - \hat{\mathbf{1}})) \circ (\hat{\mathbf{q}}_{B/D} \hat{\boldsymbol{\omega}}_{B/D}^B) \end{aligned} \quad (5.19)$$

Noting that $\hat{\omega}_{B/D}^B$ is a vector dual quaternion, using the expression $\hat{\mathbf{a}} \circ (\hat{\mathbf{b}} \hat{\mathbf{c}}) = \hat{\mathbf{c}}^s \circ (\hat{\mathbf{b}}^* \hat{\mathbf{a}}^s)$ it is possible to obtain

$$\begin{aligned} ((\hat{k}_1 \circ (\hat{\mathbf{q}}_{B/D} - \hat{\mathbf{1}})) \circ (\hat{\mathbf{q}}_{B/D} \hat{\omega}_{B/D}^B)) &= (\hat{\omega}_{B/D}^B)^s \circ (\hat{\mathbf{q}}_{B/D}^* (\hat{k}_1 \circ (\hat{\mathbf{q}}_{B/D} - \hat{\mathbf{1}}))) \\ &= (\hat{\omega}_{B/D}^B)^s \circ \text{vec} \left(\hat{\mathbf{q}}_{B/D}^* (\hat{k}_1 \circ (\hat{\mathbf{q}}_{B/D} - \hat{\mathbf{1}}))^s \right) \end{aligned} \quad (5.20)$$

and simple algebra manipulations shows that

$$\begin{aligned} \text{vec} \left(\hat{\mathbf{q}}_{B/D}^* \times (\hat{k}_1 \circ (\hat{\mathbf{q}}_{B/D} - \hat{\mathbf{1}}))^s \right) &= \text{vec} \left(\left(\mathbf{q}_{B/D}^* + \epsilon \frac{1}{2} (\mathbf{r}_{B/D}^B)^* \mathbf{q}_{B/D}^* \right) \left(\frac{1}{2} k_r \mathbf{q}_{B/D} \mathbf{r}_{B/D}^B + \epsilon k_q (\mathbf{q}_{B/D} - \mathbf{1}) \right) \right) \\ &= \text{vec} \left(\frac{1}{2} k_r \mathbf{r}_{B/D}^B + \epsilon k_q (\mathbf{1} - \mathbf{q}_{B/D}^*) \right) \\ &= (\hat{k}_1 \circ \hat{\mathbf{p}}_{B/D})^s \end{aligned} \quad (5.21)$$

Now, Equation (5.20) can be re-written as

$$((\hat{k}_1 \circ (\hat{\mathbf{q}}_{B/D} - \hat{\mathbf{1}})) \circ (\hat{\mathbf{q}}_{B/D} \hat{\omega}_{B/D}^B)) = (\hat{\omega}_{B/D}^B)^s \circ (\hat{k}_1 \circ \hat{\mathbf{p}}_{B/D})^s \quad (5.22)$$

Then, introducing this expression into Equation (5.19), the derivative of the Lyapunov function yields

$$\dot{V} = -(\hat{\omega}_{B/D}^B)^s \circ (\hat{k}_2 \circ \hat{\omega}_{B/D}^B)^s \leq 0 \quad (5.23)$$

Therefore, $\lim_{t \rightarrow \infty} V(t)$ exists and is finite and the states $\hat{\mathbf{q}}_{B/D}$ and $\hat{\omega}_{B/D}^B$ are uniformly bounded, which means that $\hat{\mathbf{f}}_c^B$ and thus $\hat{\omega}_{B/D}^B$ are also bounded. Hence, $\hat{\omega}_{B/D}^B$ and \dot{V} are uniformly continuous. Making use of Barbalat's lemma (explained in Appendix A), $\lim_{t \rightarrow \infty} V(t) = 0$, and thus $\lim_{t \rightarrow \infty} \hat{\omega}_{B/D}^B(t) = \hat{\mathbf{0}}$. Then, since $\ddot{\omega}_{B/D}^B(t)$ is bounded; $\dot{\omega}_{B/D}^B(t)$ is uniformly continuous. Making use again of Barbalat's lemma, $\lim_{t \rightarrow \infty} \dot{\omega}_{B/D}^B(t) = \hat{\mathbf{0}}$; and thus, $\lim_{t \rightarrow \infty} \hat{\mathbf{p}}_{B/D}^B(t) = \hat{\mathbf{0}}$. This implies that $\lim_{t \rightarrow \infty} \hat{\mathbf{q}}_{B/D}(t) = \pm \hat{\mathbf{1}}$ and $\lim_{t \rightarrow \infty} \hat{\omega}_{B/D}^B(t) = \hat{\mathbf{0}}$. Which then means that the spacecraft will be stabilized within the reference trajectory.

5.3. GAINS OPTIMIZATION

As can be seen in the controllers given by Equations (5.8), (5.14) and (5.15), there are different scalars or dual-number gains. Depending on the value of these gains, controllers will behave in different ways and its performance will depend on them. That is why they have to be tuned for each specific case to obtain the best desired performance.

To optimize and tune each controller, a desired goal or requirement is needed. Then, the criteria for assessing what is optimal can be defined. In this thesis, a cost function has been developed for this reason, taking into many different aspects described below.

First of all, the selected gain values have to meet the mission requirements specified in Section 2.4. For instance, the attitude pointing error cannot exceed 0.01° during the hovering scenario. All the combination of gains that do not meet these requirements will not be considered. Second of all, the selected gains have to bring the spacecraft to the reference trajectory before a given settling time, also specified in Section 2.4. The spacecraft will have to remain within the reference trajectory for the rest of the simulation. Again, all the combinations of gains that do not meet this requirement will not be considered. As an example, the settling time for the landing scenario has been defined to be one minute. After one minute, the controls should

have steered the spacecraft to its reference trajectory. Finally, the selected gain values also have to meet some overshoot requirements. Since it is not desired to have abrupt control responses, the initial overshoots cannot exceed ten times the maximum control errors. For instance, the spacecraft error velocity cannot exceed the value of 1 cm/s during the hovering scenario. All combination of gains that do not meet these requirements will not be considered.

All gains meeting these requirements can be used to achieve the mission goals. However, some of them have better performances than others. Once the mission requirements are guaranteed, the second most relevant aspect is to analyze how much energy does it take to achieve this goal. Some combination of gains can meet the requirements but consume a lot of energy. For this reason, the energy spent by each combination of gains has to be taken into account. The same concept also applies for the steady state error. Some combinations of gains can meet the requirements but have a large constant steady state error.

5.3.1. COST FUNCTION

Bearing all these aspects in mind, it is now possible to define an optimization method and a cost function that will determine which combination of gains is the optimal for each scenario. It is important to mention that all combination of gains will be optimized for the nominal trajectory. This means, that the spacecraft is always supposed to start with the correct states and no initial errors.

The optimization method chosen for this master thesis is a trial and error method. It consists in generating all possible different combinations of gains and assessing their performance with a cost function. It is a time consuming method but it works; specially when the behavior of the cost function is clearly non-linear (as will be seen below in this section). It might be important to notice that there are always four gains to be optimized. However, since the classical representation has different control approaches for the rotational motion than for the translational motion, the gains for each motion are completely independent. This is not the case for the dual quaternion representation, where the four gains are totally linked. This has an important impact on the optimization method. It means that for the classical representation, the method can be divided into two two-dimensional optimization processes. Whereas for the dual quaternion representation, a full four-dimensional optimization process is required. The first one is clearly much faster and also has the advantage of being able to graphically visualize the results. This is a drawback of the dual quaternion representation: the gain selection is much difficult and time consuming.

The optimization process defined for this master thesis is as follows. First, a grid of possible combinations of gain values is generated. For each combination, the three mentioned requirements are checked to be met (mission requirements, settling time requirements and overshoot requirements). If the combination of gains meet the requirements, a cost function is evaluated. This process can be represented by:

$$\text{COMBINATION OF GAINS} \Rightarrow \text{MEET REQUIREMENTS?} \Rightarrow \begin{cases} \text{NO} & \rightarrow \text{NOT CONSIDERED} \\ \text{YES} & \rightarrow \text{COST FUNCTION EVALUATION} \end{cases}$$

The cost function is a combination of two weighted factors. The first and most important is the energy consumption. The second is the steady state error. The priority of the optimization process is to first meet the

requirements, then spend as less energy as possible, and finally, have the minimum error while meeting the control requirements. For this reason, the selected weights have been chosen to be 80% for the energy and 20% for the steady state errors. A sensitivity analysis has been carried out to prove that results are only slightly affected by these weights. It is important to bear in mind that energy has been always treated as more important than the steady state errors because it means less fuel. And fuel is very expensive in space. If mission requirements are met, the steady state errors should only be considered for choosing between two very similar energy consumption performances of different combinations of gains. This is the reason for selecting these weights. The cost function is then given by

$$\text{COST FUNCTION} = 0.8 \cdot \text{ENERGY CONSUMPTION} + 0.2 \cdot \text{STEADY STATE ERROR} \quad (5.24)$$

To compute the energy consumption of each combination of gains, torques and forces have to be taken into account. However, the order of magnitude of the forces required to control the spacecraft is much higher than the order of magnitude of the torques. For this reason, they have been normalized with the maximum available torque and force, defined in Section 2.3. In this way, they can be added up and get a performance parameter indicating the energy spent for a specific combination of gains. This parameter is defined by

$$\text{ENERGY CONSUMPTION} = \int_{t_0}^t \frac{\sqrt{F_x^2 + F_y^2 + F_z^2}}{F_{max}} dt + \int_{t_0}^t \frac{\sqrt{\tau_x^2 + \tau_y^2 + \tau_z^2}}{\tau_{max}} dt \quad (5.25)$$

where F_i are each of the control force components and τ_i each of the control torque components. F_{max} and τ_{max} are their maximum values defined by actuator limitations. In this way, the total energy spent for each combination of gains can be evaluated. To compute the steady state error, a similar evaluation has to be carried out. Again, all different errors have to be integrated and normalized. In this case, the maximum control requirement values, defined in Section 2.4, have been used for normalization. For instance, the position error has to be divided by its maximum permitted error. In this way, all different errors with different units and orders of magnitudes can be added up. The steady state error parameter is defined by

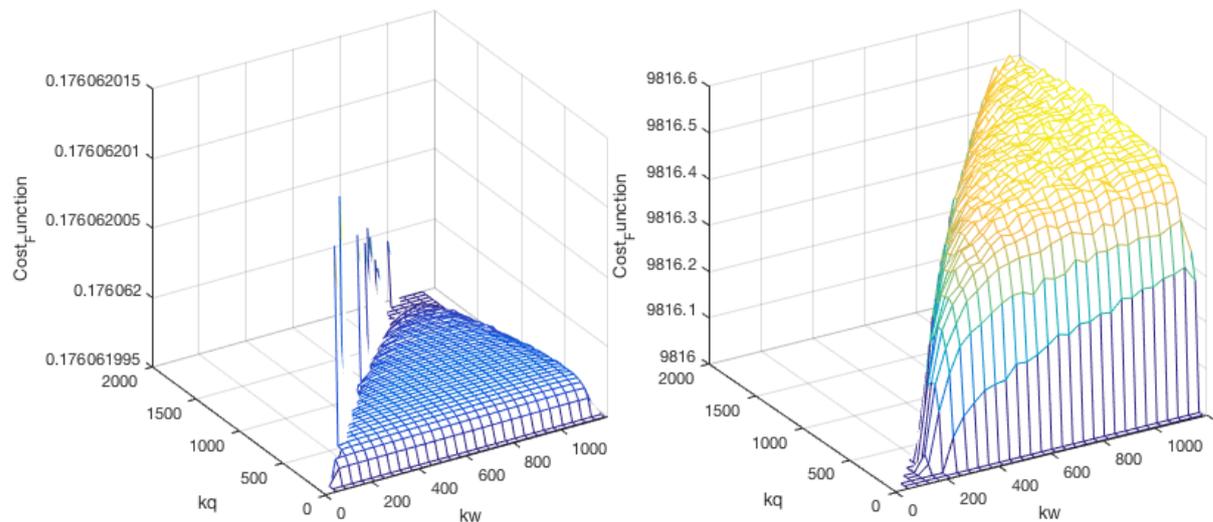
$$\text{STEADY STATE ER.} = \text{POSITION ER.} + \text{VELOCITY ER.} + \text{ATTITUDE ER.} + \text{ANGULAR VELOCITY ER.}$$

where

$$\begin{aligned} \text{POSITION ERROR} &= \int_{t_0}^t \frac{\sqrt{r_x^2 + r_y^2 + r_z^2}}{\rho_{\text{position}}} dt \\ \text{VELOCITY ERROR} &= \int_{t_0}^t \frac{\sqrt{v_x^2 + v_y^2 + v_z^2}}{\rho_{\text{velocity}}} dt \\ \text{ATTITUDE ERROR} &= \int_{t_0}^t \frac{|2 \cdot \arccos(q_0)|}{\rho_{\text{attitude}}} dt \\ \text{ANGULAR VELOCITY ERROR} &= \int_{t_0}^t \frac{\sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}}{\rho_{\omega}} dt \end{aligned} \quad (5.26)$$

where ρ is the maximum allowed error for each variable defined in Section 2.4.

Following this optimization method, all gains for all different scenarios have been optimized. It is important to notice two consequences derived from choosing this method. The first one is that, as one can imagine,



(a) Energy consumption for the rotational hovering motion (b) Steady state error for the rotational hovering motion

Figure 5.1: Energy and steady state error cost functions for the hovering scenario

it is impossible to generate all possible combination of gains because there are infinite numbers between two values. For this reason, a dense grid has to be chosen so that it can handle as much combinations as possible. With a dense grid, the gain values can be found with more precision and therefore, a better performance can be achieved. The second one is that this takes a lot of time. For instance, if each simulation takes 2 seconds to run, a grid of 25 values for each gain takes four and a half days. This is of course unacceptable. To avoid such long computing times, the Matlab parallel toolbox has been used. Since the evaluation of each simulation is completely independent from the next one, the computer can divide its processor and create a pool of different workers. The more powerful the computer is, the more workers it can have. Each worker can compute in parallel different simulations with a certain combination of gains. Combining all independent results, the optimum combination of gains can be founded faster.

Last but not least, results from the classical rotating hovering case are presented here as an example. As has been mentioned before, the optimization process for the classical representation can be divided into two two-dimensional optimization processes. The following results correspond to the rotational control gains. Since it is a two dimensional problem, it can be represented in a 3d-plot. Figure 5.1a shows the energy consumption and Figure 5.1b the steady state error.

The first thing one has to realize is that, both functions have been inverted, for representation purposes. The sign has been changed in both of them and they have been brought to positive numbers by adding the lower minimum value. A second important thing to mention is that, in both cases, the maximum performance is very close to their neighbors. For instance, the difference in energy spent by all combination of gains showed in Figure 5.1a is very small. It is important to mention that values on the lower plane have been changed so that the difference can be appreciated. The combination of gains that do not have any value correspond to the ones not meeting the mission requirements. This means, that almost all the gains that converge have very similar energy performances. This happened to all the results from this optimization process

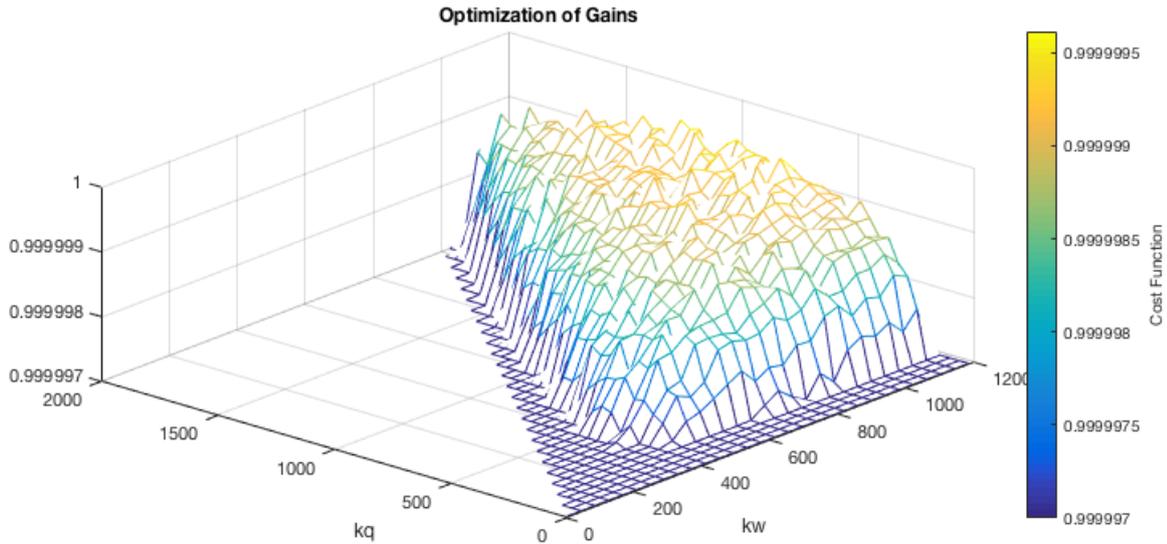


Figure 5.2: Cost function for the rotational hovering motion

in the different three scenarios. Optimal performances are reached with a large combination of different gain values. However, there was always a combination of gains with a maximum performance. The spikes on the left part of Figure 5.1a correspond to local phenomena, derived from the combination of frequencies with the integration time step. They are however compensated when doing the average with the steady state error function.

Doing the previous mentioned weighted average with the energy consumption and the steady state error, the cost function for this specific example is shown in Figure 5.2. As can be seen, values have been normalized from one to zero. A value of one in the cost function would mean that the same combination of gains had both the best energy performance and the best steady state error performance.

As happened before, despite the combination of factors, the optimal gains still have a very similar performance index than their neighbors. This means that many different options have similar performances. For this specific case, the maximum was found for $K_q = 1065.9 \text{ N}\cdot\text{m}$ and $K_\omega = 1106.5 \text{ N}\cdot\text{m}\cdot\text{s}/\text{rad}$.

5.3.2. SELECTED GAINS

All gains have been optimized following the described optimization process. Table 5.1 shows the selected gains for the orbiting scenario and the simulation settings used for the optimization. The first thing to notice is that the initial conditions correspond to the nominal ones. The initial quaternion attitude corresponds to the one pointing at the center of the asteroid from the initial position. As can be seen, the integration frequency has been set to be 1 Hz. This corresponds to an integration time-step of one second, which provides an accuracy of 10^{-9} in order of magnitude to all state variables. The control frequency is 0.5 Hz. This corresponds to a control update every two seconds. With this control frequency, all mission requirements can be met easily. The last but not least important thing to notice is that the time span used to optimize these gains has been selected to be a tenth of the orbital period. This selection is very important because it comes

with some consequences. On one hand, the time span has to be as short as possible to reduce the computational time of the optimization process. On the other hand, it has to be long enough to allow the controller to stabilize the motion of the spacecraft. It is important to take into account the time evolution of all the state variables. Tiny increasing errors could be present and do not have an important impact on the results but become a problem for longer periods of time. After trying several time span options, a tenth of an orbit has been found to be long enough to stabilize all motions and do not take excessive computational time. The value of the selected gains is very similar for K_ω but much different for K_q . It is worth mentioning that in the orbiting scenario, only the rotational motion is controlled, since the spacecraft is already in a very stable terminator orbit. For this reason, only K_q and K_ω need to be optimized in both cases.

Table 5.1: Selected gains and settings for orbiting scenario

Parameter	Value
Initial position (Inertial frame)	[0; 0 ; 1500] m
Initial velocity (Inertial frame)	[0.03563 ; -0.04687 ; 0] m/s
Initial attitude	[0 ; -0.3193 ; 0.9476 ; 0]
Initial angular velocity (Body frame)	[-1.9626 ; -0.000002106 ; -0.1074] $\cdot 10^{-5}$ rad/s
Integration and control freq.	IF = 1 Hz ; CF = 0.5 Hz
Time span	10% orbital period ($1.6 \cdot 10^4$ s)
Classic control gains	$K_q = 1165.0, K_\omega = 1129.0$
Dual quaternion control gains	$K_q = 208.3, K_\omega = 1108.0$

For the hovering scenario, the selected gains and settings for the optimization process are shown in Table 5.2. As can be noticed, the initial conditions also correspond to the nominal ones, with no initial deviations. In this case, the integration frequency has been set to be 2 Hz to provide an accuracy of the state errors of 10^{-12} in order of magnitude. The control frequency has also been selected to be 0.5 Hz, because it allows the controllers to meet with all the mission requirements. All flight proven actuators and sensors on-board current missions to asteroids can deal with these control frequencies. For the same reasons mentioned in the orbiting scenario, and after some trial an error, the time span for this optimization process has been chosen to be one hour. This time span allows the tiny motions with low frequencies to stabilize and guarantee that both controllers meet the mission requirements. It is important to notice that, in this case, both the rotational and translational motions have to be controlled. In contrast with the orbiting scenario, four gains have to be selected for each control approach. As can be seen, the gain values are completely different for the classical representation than for the dual quaternion one. The reason for that is the natural coupling of the motions in the dual quaternion representation.

Finally, Table 5.3 shows the selected gain values for the landing scenario, together with the optimization settings. The initial conditions are again the ones from the nominal trajectory, with no initial errors. However, in this case, the integration frequency has been selected to be 4 Hz. The reason behind this selection can be found in the accuracy of the errors. Since the landing trajectory involves higher control commands to meet

Table 5.2: Selected gains and settings for hovering scenario

Parameter	Value
Initial position (Inertial frame)	[-288.563 ; -687.929 ; 0] m
Initial velocity (Inertial frame)	[0. ; 0. ; 0] m/s
Initial attitude	[0.1812 ; 0.0973 ; 0.7004 ; 0.6835]
Initial angular velocity (Body frame)	[0 ; 0 ; 0] · 10 ⁻³ rad/s
Integration and control freq.	IF = 2 Hz ; CF = 0.5 Hz
Time span	0 - 1 h
Classic control gains	$K_q = 1065.9, K_r = 862.1, K_\omega = 1106.5, K_v = 1075.9$
Dual quaternion control gains	$K_q = 512.7, K_r = 1.82, K_\omega = 581.9, K_v = 1.82$

Table 5.3: Selected gains and settings for landing scenario

Parameter	Value
Initial position (ACI frame)	[345.46 ; 0 ; 0] m
Initial velocity (ACI frame)	[-0.1 ; -0.1403 ; 0] m/s
Initial attitude	[0.4743 ; 0.5244 ; -0.4743 ; -0.5244]
Initial angular velocity (Body frame)	[0.0932 ; 0.6336 ; -0.0191] · 10 ⁻³ rad/s
Integration and control freq.	IF = 4 Hz ; CF = 1 Hz
Time span	0 - 600 s
Classic control gains	$K_q = 3271.4, K_r = 3974.0, K_\omega = 1978.6, K_v = 2295.0$
Dual quaternion control gains	$K_q = 950.0, K_r = 0.267, K_\omega = 1050.0, K_v = 1.213$

the mission requirements, the integration time-steps had to be smaller to achieve an accuracy of 10^{-12} in order of magnitude. For the landing scenario, a control frequency of 1 Hz was needed to achieve the mission requirements. In this case, there was no other option than having a time span of 600 s because of the duration of the defined landing trajectory. After 600 s the spacecraft has to be landed on the asteroid. Again, the value of the gains for the classical representation are very different than for the dual quaternion representation, because of the same reasons than in the hovering scenario.

6

NUMERICAL METHODS

This chapter aims at describing the numerical methods used in this master thesis. First of all, the integration method used for solving the differential equations of motions is explained in Section 6.1. Apart from integration algorithms, this thesis also needs a root-finding method for solving algebraic equations. It is defined in Section 6.2. Finally, for verification purposes, some particular solutions need to be interpolated to meet a certain output points. The interpolation numerical method used in this master thesis is defined in Section 6.3.

6.1. INTEGRATION METHOD

Many methods have been developed for the numerical integration and each of them has its own inherent advantages and drawbacks. It is in general not possible to simply select one method as best suited for the prediction of a satellite motion. The selection of the best numerical integrator for solving the equations of motions involved in asteroid close-proximity operations could be a single thesis alone. There are single and multiple-step methods, fixed and variable step-sizes algorithms, explicit and implicit approaches, one and multiple-stage integrators, different order methods, energy conservative algorithms, and many other characteristics affecting the performance of each integrator. However the goal of this master thesis is to compare the dual quaternion representation with the classical one. For this reason, only two aspects have been taken into account when selecting the numerical integration technique.

The first one is that it should be a fixed step-size algorithm. The reason is because the spacecraft has to be controlled so that it can follow the desired trajectories. All controllers have a working frequency at which they operate. The same happens with the navigation sensors. For this reason, it is more convenient to have a fixed integration step-size method, so that these operating frequencies can be handled in a much easier way.

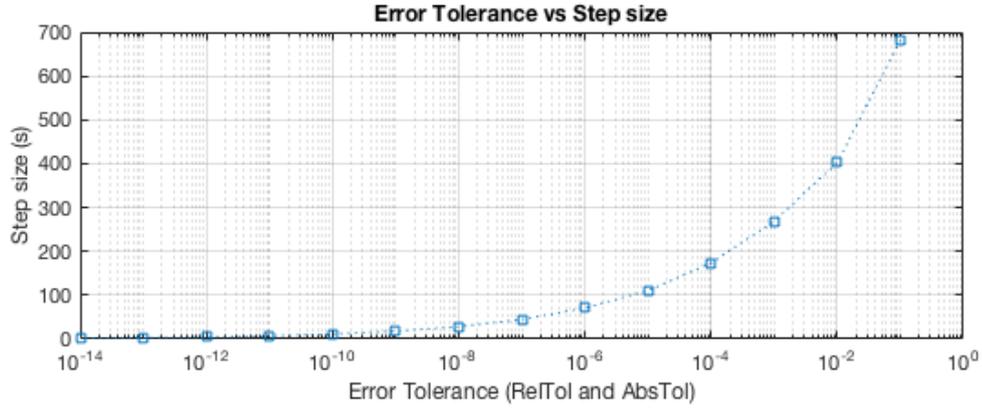


Figure 6.1: Step size dependency with tolerance of the solution for close-proximity operations in Bennu

The second one is that the integration technique should provide sufficient precision of the solution so that the dual quaternion representation can be fairly compared with the classical one.

Bearing these requirements in mind, a Runge-Kutta 4 algorithm is perfectly suited for this work. It is a fixed step-size method easy to implement and totally capable of providing the required precision. To prove that, the required time-steps needed to achieve certain levels of accuracy has been computed in the vicinity of asteroid Bennu. Figure 6.1 shows this. As can be seen, the larger the step-sizes are, the larger the errors of the solution are. More specifically, it shows that the required step-size to achieve an error of the solution in the order of magnitude of 10^{-14} , is two seconds. The mission requirements defined in Section 2.4 have an order of magnitude much larger than 10^{-14} . Maybe another algorithm could provide the same level of accuracy in a faster way, but it is not the important aspect to take into account in this thesis.

The Runge-Kutta 4 is an explicit, multi-stage and single-step integration method. It is a fourth-order method, meaning that the local truncation error is on the order of $O(h^5)$, while the total accumulated error is on the order of $O(h^4)$. It uses the following expression to approximate the solution of \mathbf{y} at some later time $t_0 + h$:

$$\mathbf{y}(t_0 + h) \approx \mathbf{y}_0 + h \cdot \Phi = \boldsymbol{\eta}(t_0 + h) \quad (6.1)$$

It is evident that Φ , the increment function, should closely approximate the slope of the secant through (t_0, \mathbf{y}_0) and $(t_0 + h, \mathbf{y}(t_0 + h))$. Carl Runge and Wilhelm Kutta developed different expressions for Φ based on the slopes at various points within the integration step. In the classical Runge-Kutta 4 method, the increment function Φ is calculated as the weighted mean [Montenbruck and Gill, 2000]

$$\Phi_{RK4} = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (6.2)$$

of four slopes

$$\begin{aligned} k_1 &= \mathbf{f}(t_0, \mathbf{y}_0) \\ k_2 &= \mathbf{f}(t_0 + h/2, \mathbf{y}_0 + h \cdot k_1/2) \\ k_3 &= \mathbf{f}(t_0 + h/2, \mathbf{y}_0 + h \cdot k_2/2) \\ k_4 &= \mathbf{f}(t_0 + h, \mathbf{y}_0 + h \cdot k_3) \end{aligned} \quad (6.3)$$

This formula approximates the exact solution up to terms of order h^4 , provided that $\mathbf{y}(t)$ is sufficiently smooth and differentiable, and it is therefore called a 4th-order method. Its local truncation error

$$e_{RK4} = |\mathbf{y}(t_0 + h) - \boldsymbol{\eta}(t_0 + h)| \leq \text{const} \cdot h^5 \quad (6.4)$$

is bound by a term of order h^5 .

6.2. ROOT FINDING

There are also many root finding methods: bisection methods, false position, Newton's methods, secant methods, interpolation methods, inverse interpolation methods... All of them have their own advantages and disadvantages and are better suited for different equations. However, as has been done with the numerical integrator, the important aspect is that it has to work properly. For this reason, a simple secant method has been selected for solving the algebraic equations of this master thesis. It is defined by the following recurrence relation:

$$x_n = x_{n-1} - f(x_{n-1}) \frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})}$$

As can be seen, it requires two initial values, which should ideally be chosen close to the root. Its order is approximately 1.6.

6.3. INTERPOLATION

Sometimes, a given solution is required to be known at a point where it has not been defined yet. For these cases, an interpolation technique is needed. There are also many possibilities with different levels of accuracy and difficulties, however, for this master thesis, only linear interpolations will be carried out. To carry out a linear interpolation between two known points with coordinates (x_0, y_0) and (x_1, y_1) , the solution is very straight forward. The linear interpolant will be the straight line between these points. For a value x in the interval (x_0, x_1) , the value y along the straight line is given by:

$$y = \frac{y_0(x_1 - x) + y_1(x - x_0)}{x_1 - x_0}$$

7

SIMULATOR DESIGN

All the elements involved in the simulations have been defined and characterized in previous chapters. Therefore, it is now possible to build a simulator in Matlab and assemble all different parts together. This chapter describes the way in which this simulator has been implemented in Section 7.1, and provides a top-level architectural design of it. To make sure that everything works as expected and that the results obtained are reliable, a detailed process of verification and validation has been carried out. Section 7.2 describes this process. This is a very important procedure, because it gives credibility to the research performed in this thesis and shows that results can be trusted.

7.1. ARCHITECTURE DESIGN

To build a simulator that represents the dynamical environment of Bennu and the spacecraft, many different aspects come into play. It is therefore desirable to divide the simulator into several blocks. This section provides a top level architectural design of the simulator implemented in Matlab, showing these divisions. The simulator can be divided into five different top level blocks: guidance, control, actuators, sensors and navigation. Figure 7.1 shows them and how they interact with each other.

For the sake of clarification, 4 different colors have been used. Blue boxes in Figure 7.1 represent the main blocks already mentioned. Dark blue boxes represent the outputs of the simulator. As can be seen, the outputs of the simulations are the desired trajectories to follow, force and torque commands and the actual trajectory followed by the spacecraft. Grey boxes are the inputs of the simulation. As can be seen, there are only 4 inputs: the mission goal, initial conditions, integration settings and control settings. Finally, purple boxes represent the different databases where information about the spacecraft or the asteroid trajectory is stored.

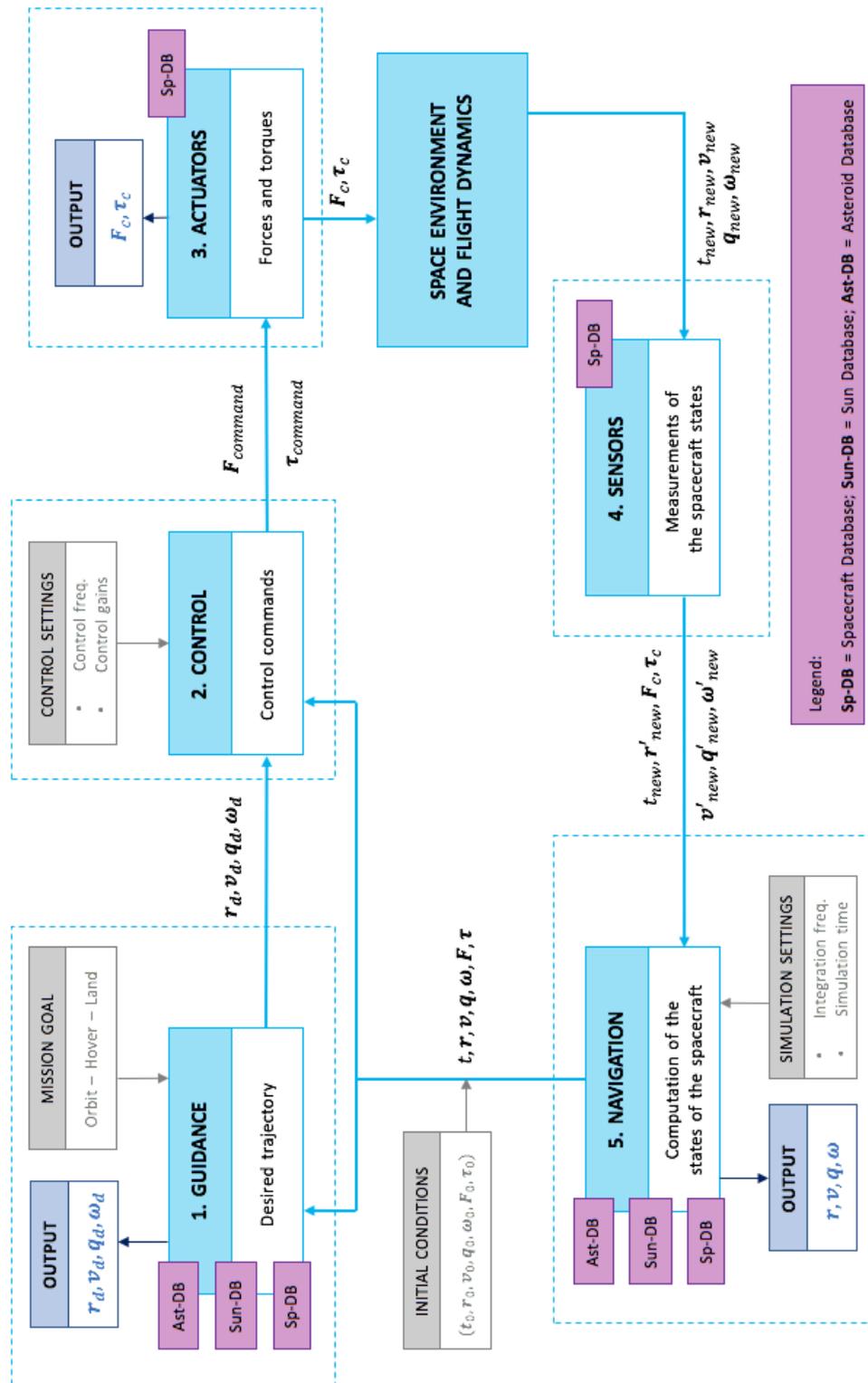


Figure 7.1: Top level architecture design of the simulator

7.1.1.1. GUIDANCE

The guidance block is responsible for determining the reference trajectory to follow. It tells the spacecraft where to go. As inputs, it takes the mission goal and the state of the spacecraft. In real missions, it should

be able to recompute the desired trajectory depending on different factors. For instance, if something goes wrong, it should command a safe maneuver. Another example could be to readjust the trajectory depending on the fuel level. In this master thesis however, the reference trajectory will not change. As explained in Chapter 2, these reference trajectories will be the orbiting, hovering and landing on Bennu.

7.1.2. CONTROL

The control block is the one responsible for computing the control forces and torques necessary to follow the desired trajectory. Taking as inputs the reference trajectory and the states of the spacecraft, it computes the required forces and torques following the control laws defined in Chapter 5. It is important to mention that as inputs, it also receives the control frequency and the control gains. The control frequency is always lower than the integration frequency due to hardware limitations (on-board CPU).

7.1.3. ACTUATORS

The actuators block represents all the real actuators on the spacecraft meant to execute the forces and torques commands computed by the control block. All real actuators have a certain efficiency and a certain precision, thus, the computed commands are not always the real ones acting on the spacecraft. Apart from that, real actuators also have an working range. For instance, there is a maximum force one can achieve with a given thruster. These limitations have been defined in Chapter 2. Bearing this in mind, the actuator block takes these limitations into account and computes the real control forces and torques that will steer the spacecraft.

7.1.4. SENSORS

The sensors block represents the spacecraft sensors measuring different types of information. For example, gyroscopes measuring angular velocities, star trackers giving the orientation of the spacecraft, accelerometers measuring accelerations or navigation cameras measuring features on the surface of the asteroid. Each sensor has a working frequency which limits the integration of the different state variables. In this master thesis however, no sensors have been defined or simulated.

7.1.5. NAVIGATION

The navigation block is responsible for determining the position and orientation of the spacecraft. It usually contains a filter or a combination of filters to statistically determine all the state variables over time. As inputs, it takes the integration settings and the measured states from the sensors block. In this master thesis however, no sensors data is available. For this reason, the position and orientation of the spacecraft will be assumed to be exactly the ones coming from numerically integrating the equations of motion with the models described in Chapter 4. Perfect knowledge of the environment is assumed. The navigation block can be sub-divided into smaller sub-blocks, easier to understand, verify and validate. This is shown in Figure 7.2, where the architectural design of the navigation block is represented in detail.

As can be seen, five different sub-blocks can be identified. They correspond to the different models de-

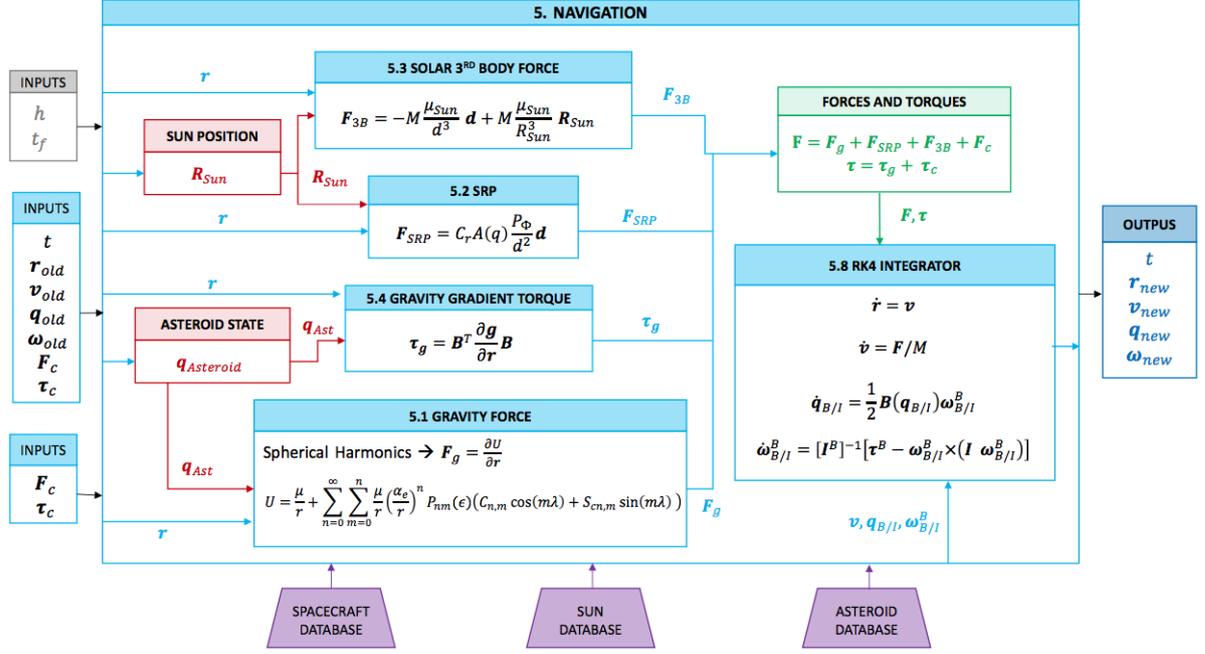


Figure 7.2: Architectural design of the navigation block

scribed in Sections 4.3 and 4.4, together with the integrator sub-block. The first one is the gravity force model, which computes the gravity forces coming from the asteroid using the spherical harmonics approximation proposed in [Gottlieb, 1993]. The second one is the SRP model followed by the 3rd body force model. The fourth one is the gravity gradient torque sub-block. And finally, the last one is the Runge-Kutta 4 integrator, which numerically integrates the position and orientation of the spacecraft over time. It is important to see the relationship between each sub-block and what are their inputs and outputs. For instance, the gravity force sub-block only needs the current orientation of the asteroid and the inertial position of the spacecraft. Moreover, the SRP sub-block requires the current position of the Sun at each time step and the position of the spacecraft as well.

Figures 7.1 and 7.2 represent the top level architectural design for the classical representation. For the dual quaternion representation, the diagrams are almost identical. Only the equations of motion are different and the way the variables are expressed and taken as inputs or outputs. As has been explained in Chapter 3, dual quaternions are more compact and only require two equations of motion. However, all five sub-blocks are still necessary and the architectural design remains the same.

7.2. VERIFICATION AND PSEUDO-VALIDATION

To make sure that all written code in Matlab has been properly implemented and works as expected, each block and sub-block defined in the architectural design has to be verified and validated. The previous subdivisions have been made so that different unit tests can be defined. In this way, the process of verification and validation becomes much clearer and easier. All different blocks and sub-blocks will be verified and validated in this section through unit tests and then, a final system test will be carried out to check if the

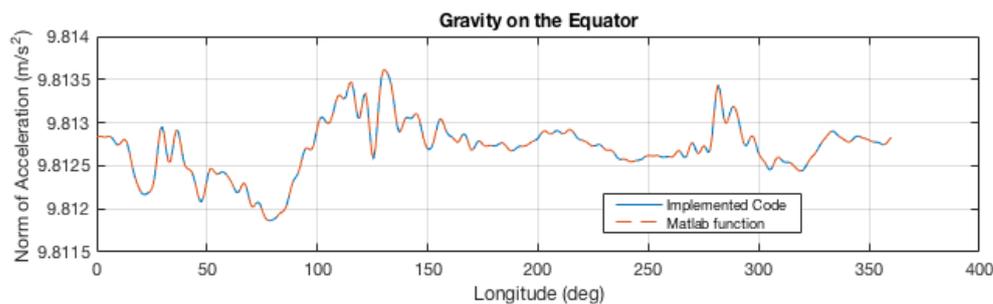


Figure 7.3: Acceleration on the equator comparison

whole simulator works properly. The system test will determine whether or not the developed simulator can be used as a tool for the purpose of this thesis or not.

It is important to bear in mind the difference between verification and validation. On one hand, the process of verification checks whether the implemented code works as expected. It has nothing to do with the physical meaning. On the other hand, the process of validation proves that the implemented code provides results that represent the real expected physical values with physical meaning. Since Bennu has never been visited, the only way of validating in this master thesis, is to compare results with other previous independent studies. That is why it is called pseudo-validation in this case.

Despite the fact that the simulation cycle starts with the guidance block, it makes more sense to first validate the smaller sub-blocks and their models associated with them in the navigation block. The reason is because this is the chronological order followed when developing the simulator.

7.2.1. NAVIGATION

As already depicted in Figure 7.2, the navigation block can be sub-divided into smaller sub-blocks that will define different unit tests. This section describes them all.

GRAVITY FORCE

The first sub-block to be verified and pseudo-validated is the implemented gravity force model. To verify that the spherical harmonics gravity force model code has been implemented correctly, an auxiliary Matlab function has been used. Matlab has its already build-in function which provides the user with the acceleration at any point around the Earth given a spherical harmonic model and its order and degree. If the code is well written, both functions should give the same acceleration at the same point given the same spherical harmonic model.

The Matlab function used is *gravitysphericalharmonic* and the gravity model selected is the EGM96. By default, Matlab uses only 70 of the 360 spherical harmonic coefficients of the model. Figure 7.3 shows the gravity acceleration on the equator for each function and Figure 7.4 shows the same but on the Greenwich meridian. Figures 7.5 and 7.6 show their differences.

As can be seen, the acceleration given by the two methods is almost the same up to a computer precision. It is interesting to mention that if less degrees are used on the spherical harmonic model, the solution is

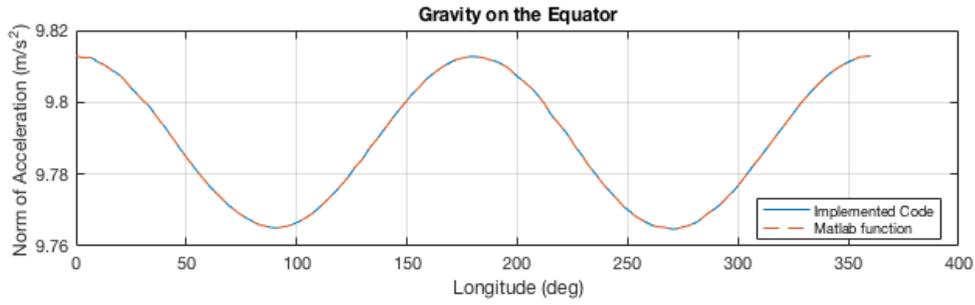


Figure 7.4: Acceleration on the Greenwich meridian comparison

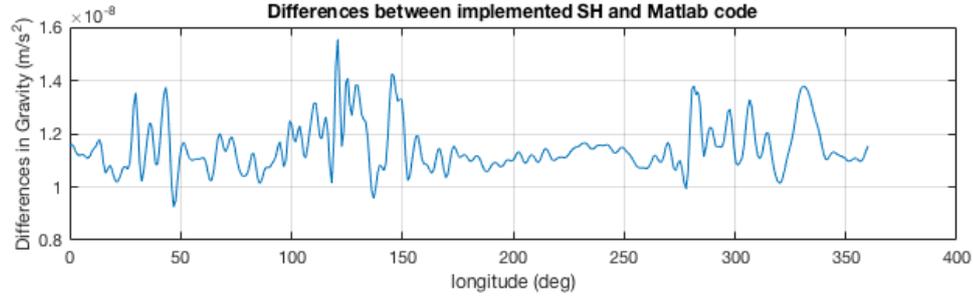


Figure 7.5: Differences in acceleration in Figure 7.3

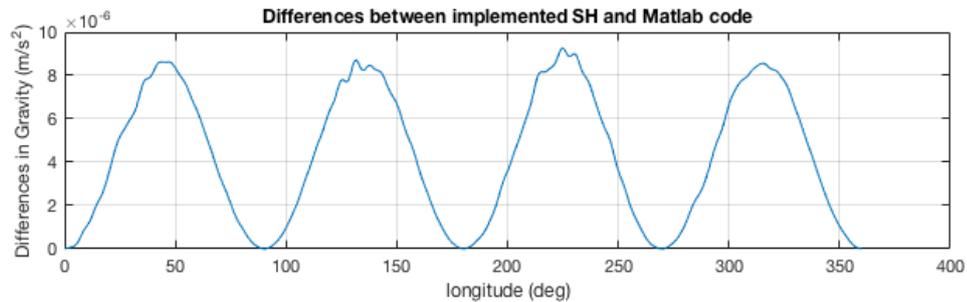
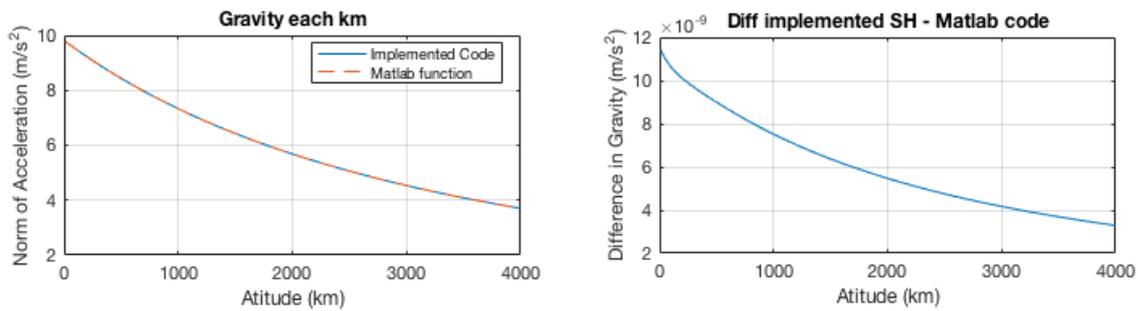


Figure 7.6: Differences in acceleration in Figure 7.4

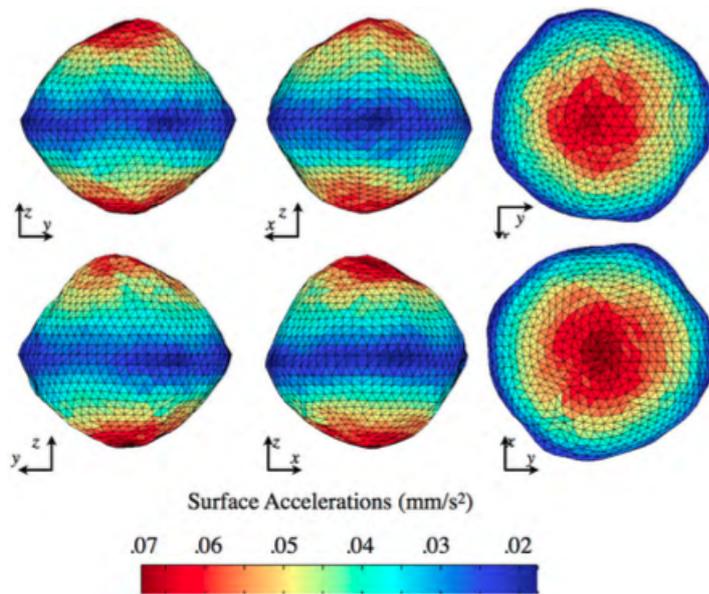
still similar except for the higher frequencies missing. The differences are in the order of magnitude of 10^{-8} m/s^2 in the equator and 10^{-6} m/s^2 in the meridian. Finally, to check how the model behave at different altitudes, Figure 7.7a shows the norm of the acceleration from the surface to up to 4.000 km and Figure 7.7b its differences with the Matlab function. As it can be seen, the further away from the Earth, the more similar the results are and the more it resembles to the central body acceleration. Differences are in the order of magnitude of 10^{-9} m/s^2 . These results verify that at least the spherical harmonics gravity field code is well implemented and that it works on Earth.

If one just changes the coefficients and use the spherical harmonics coefficients for Bennu, the code should still work properly. However, to pseudo-validate the results, data on [Hergenrother, 2014] has been used to check for the order of magnitude of the surface gravity field. Figure 7.8 shows the surface accelerations (in mm/s^2) computed by Hergenrother [2014], whereas Figure 7.9 shows the computed surface acceleration coming from the implemented code. As it can be seen, in both cases the surface acceleration has



(a) Acceleration on different altitudes on Earth

(b) Differences in acceleration in Figure 7.7a

Figure 7.8: Benu's surface acceleration in mm/s^2 from [Hergenrother, 2014]

the same order of magnitude 10^{-5} m/s^2 and follows the same pattern. It is important to mention that these accelerations are not computed using the same method. In [Hergenrother, 2014], they are computed using a polyhedron gravity model and in this thesis they are computed using spherical harmonics. As it was explained in [Takahashi, 2013], the spherical harmonics gravity model is only guaranteed to converge outside the Brillouin sphere, and the surface of Benu is clearly inside of it. That explains why the values are not exactly the same. However, the differences are inside the predicted error values computed by Takahashi [2013] when comparing a polyhedron model with a spherical harmonic model.

The combination of all these tests makes one believe that the gravity field code is well implemented and that it works as required.

SOLAR RADIATION PRESSURE

The second navigation sub-block to be verified and validated is the SRP one. To check if the solar-radiation force has been properly implemented, the order of magnitude of this force and its direction have been compared to data taken from [McMahon et al., 2018]. Figure 7.10 shows different accelerations acting on OSIRIS-

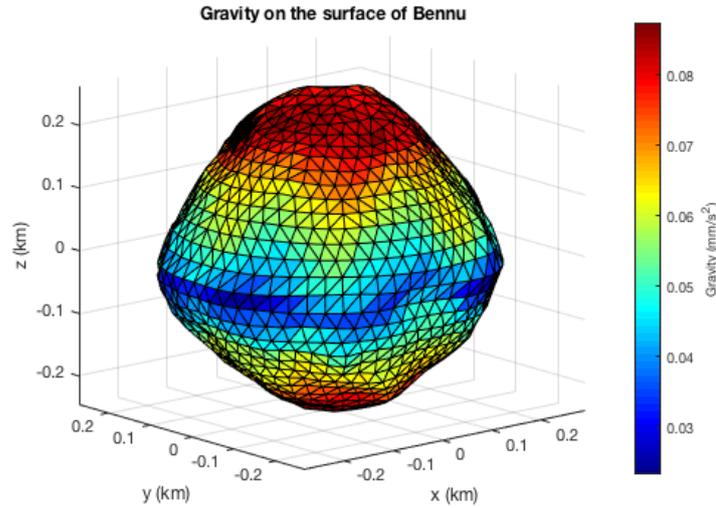
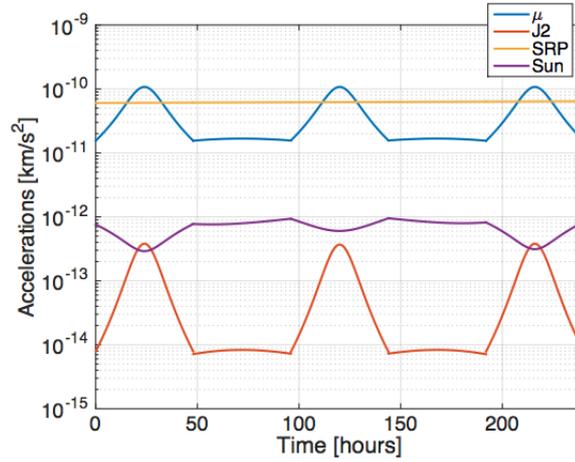
Figure 7.9: Computed Bennu's surface acceleration in mm/s^2 

Figure 7.10: Comparison of acceleration magnitudes during preliminary survey [McMahon et al., 2018]

Table 7.1: Solar radiation pressure acceleration at 7 km in ACI frame

Point (km)	Acc x (m/s^2)	Acc y (m/s^2)	Acc z (m/s^2)	Norm (m/s^2)
(7,0,0)	$2.655 \cdot 10^{-8}$	$2.018 \cdot 10^{-8}$	$0.203 \cdot 10^{-8}$	$3.341 \cdot 10^{-8}$

REx during the preliminary survey phase, which happens to be a bunch of flybys with periapsis of 7 km each.

As can be seen, the SRP acceleration is constant with an order of magnitude of 10^{-8} m/s^2 . The implemented code gives an acceleration that can be shown in Table 7.1 at (7,0,0) km in the ACI reference frame. The order of magnitude is exactly the same and the direction is pointing outwards the Sun. This unit test has been carried out at different other points in the proximity of Bennu and the result is exactly the same due to the fact that the Sun is very far away compared to the changing distances in the proximity of Bennu.

Table 7.2: Solar tide acceleration at 7 km in ACI frame

Point (km)	Acc x (m/s^2)	Acc y (m/s^2)	Acc z (m/s^2)	Norm (m/s^2)
(7, 0, 0)	$1.108 \cdot 10^{-10}$	$1.784 \cdot 10^{-10}$	$0.180 \cdot 10^{-10}$	$2.108 \cdot 10^{-10}$
(-7, 0, 0)	$-1.108 \cdot 10^{-10}$	$-1.784 \cdot 10^{-10}$	$-0.180 \cdot 10^{-10}$	$2.108 \cdot 10^{-10}$
(0, 7, 0)	$1.784 \cdot 10^{-10}$	$0.117 \cdot 10^{-10}$	$0.137 \cdot 10^{-10}$	$1.793 \cdot 10^{-10}$
(0, -7, 0)	$-1.784 \cdot 10^{-10}$	$-0.117 \cdot 10^{-10}$	$-0.137 \cdot 10^{-10}$	$1.793 \cdot 10^{-10}$
(0, 0, 7)	$0.180 \cdot 10^{-10}$	$0.137 \cdot 10^{-10}$	$-1.225 \cdot 10^{-10}$	$1.246 \cdot 10^{-10}$
(0, 0, -7)	$-0.180 \cdot 10^{-10}$	$-0.137 \cdot 10^{-10}$	$1.225 \cdot 10^{-10}$	$1.246 \cdot 10^{-10}$

3RD BODY PERTURBATIONS

The next navigation sub-block to be verified and validated, corresponds to the third body perturbations coming from the Sun. As happened with the SRP, this unit test consisted of verifying and pseudo-validating the order of magnitude and direction of the perturbing force. On one hand, Figure 7.10 shows that the order of magnitude of the acceleration coming from the Sun is $10^{-10} m/s^2$. This is data taken from [McMahon et al., 2018], which is an independent study. On the other hand, the implemented codes gives an acceleration shown in Table 7.2, at different points, in the ACI frame as well.

As can be seen, the order of magnitude is the same. However, how can we verify the direction of this force? Figure 7.11 shows the direction of the Sun perturbations (also known as solar tides), seen by the satellite in an orbit of 1 km in the ACI frame. It can be understood as the force received by 1 kg point mass, at a distance of 1 km. The direction of the acceleration is outwards near the equator and inwards on the poles. This is the same type of acceleration that generates tides on Earth, due to different 3rd bodies like the Sun or the Moon. That is why it is also called solar tide. Equation (4.11) in Chapter 4, overlaps the gravitation attraction from the Sun to the satellite, minus the acceleration from the Sun to the asteroid, to express it in the ACI frame. Comparing the direction of this perturbing field with the one in [Montenbruck and Gill, 2000], it can be verified that the implemented code represents indeed the expected force direction.

GRAVITY GRADIENT TORQUE

The unit test for the gravity gradient torque has two parts. The first one consist in comparing the torque with the analytic Equation 4.12 from [Schaub and Junkins, 2009]. This equation is an approximation of the gravity gradient torque, because it only considers a central gravity field. However, it will serve for validating the order of magnitude and direction of these torque.

The second part of the unit test is based on the fact that the gravity force model has been already verified and pseudo-validated. It consists on computing the torque using two completely independent methods. The first method is the one which will be used during all the simulations on this thesis, based on the Gottlieb algorithm [Gottlieb, 1993], and explained in Section 4.4. The second method is based on point masses. Since the acceleration of any point is known to be correct, and given by the Matlab function *gravitysphericalharmonics*, the gravity gradient torque is computed by cross-multiplying the force that each point mass is receiving per

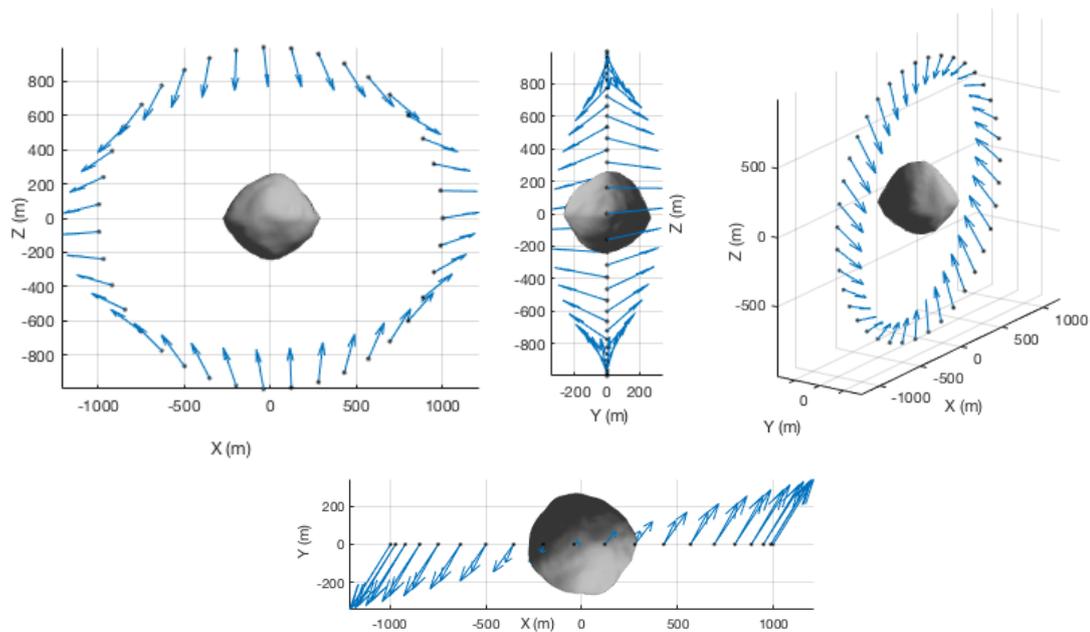


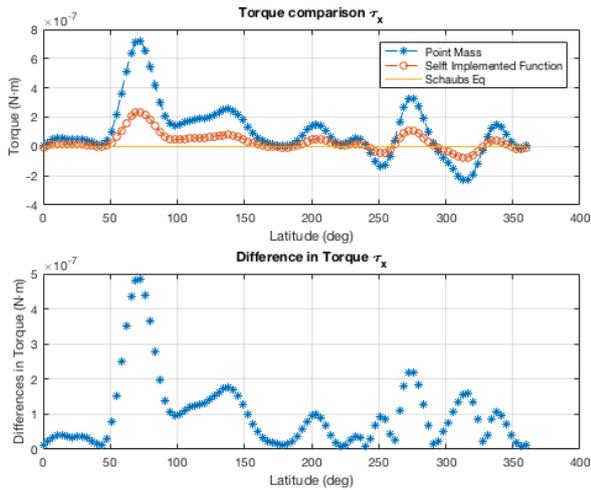
Figure 7.11: Direction of the solar tide acceleration. XZ plane (top left). YZ plane (top center). 3d view (top right). XY plane (bottom)

its distance to the center of mass and then adding up all these torques. In this case the acceleration is not assumed to vary linearly (in contrast with the first method) and it is actually computed at all points of the spacecraft using the spherical harmonics model. If the number of points is high enough, the Inertia should be close to equal and therefore arrive to a very similar gravity gradient torque.

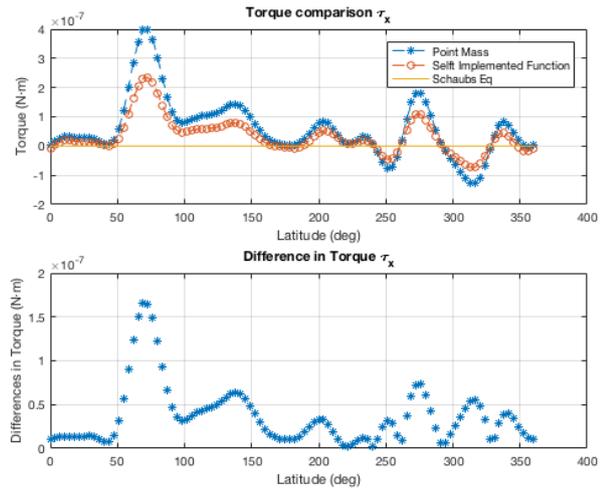
The second method has been compared with the first one for several number of point masses representing the spacecraft (in this case just a cuboid). The number of points selected to represent the spacecraft has been: 8, 64, 512, 8000, 32768 and 262144. The scenario simulated was a **LEO** polar orbit and the cuboid dimensions were 1x2x0.5 m (width, depth, height). The orientation of the satellite was maintained inertially fixed with respect to the Earth during all the orbit. The gravity gradient point was computed 100 times for each orbit in the body frame. The results are shown in Figures 7.12 to 7.14. The blue lines represent the point-mass method, the red lines the Gottlieb method and the orange lines the central body method, from Equation 4.12.

It is important to remark that, due to the approximations carried out in the central body method (orange lines), it only provides torques in the y direction of the body frame. This is one of the reasons why this method was not considered from the very beginning. However, it can be used to validate the order of magnitude and direction of this particular torque. As can be seen in Figure 7.13, the implemented method and this central body approximation are almost the same.

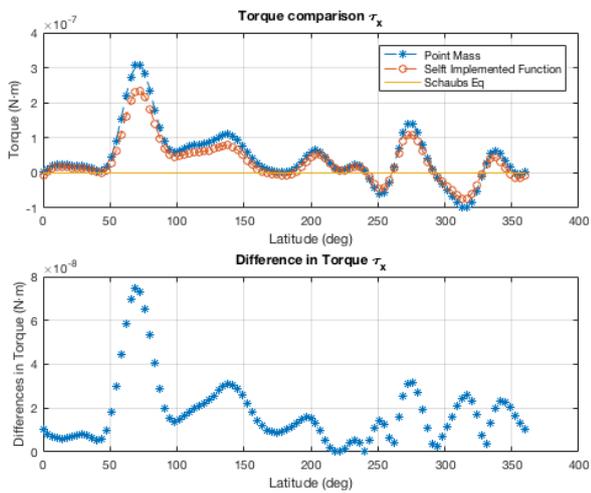
Moreover, comparing the Gottlieb method with the point-mass one, the more point masses used to represent the cuboid, the closer the solutions become. This can be seen from left to right and up to down in Figures 7.12 to 7.14. Each figure increases the number of point masses used to represent the cuboid. In the lower part of each figure, the difference between each method is plotted. As can be seen, these differences become smaller and smaller when increasing the number of point masses. This test has a maximum of 262144 number of point masses for computational reasons. With 262144 point masses, the differences in torques are



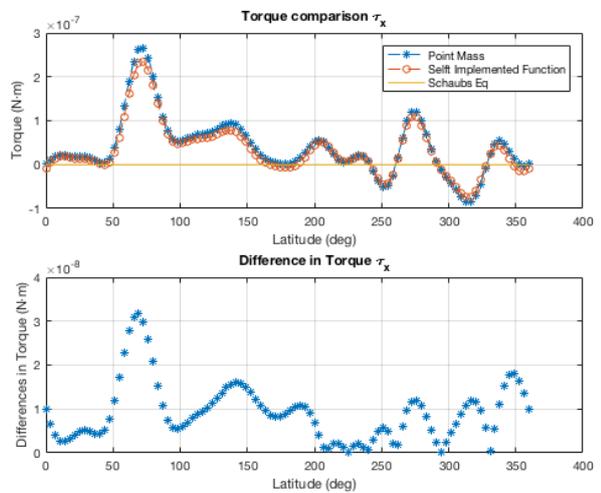
(a) Gravity gradient torque with 8 point masses



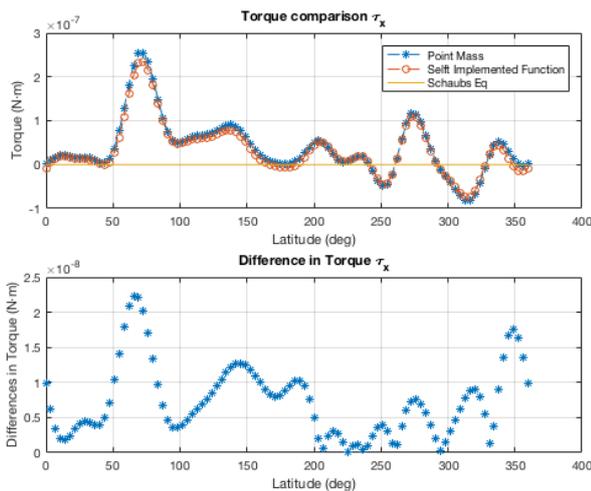
(b) Gravity gradient torque with 64 point masses



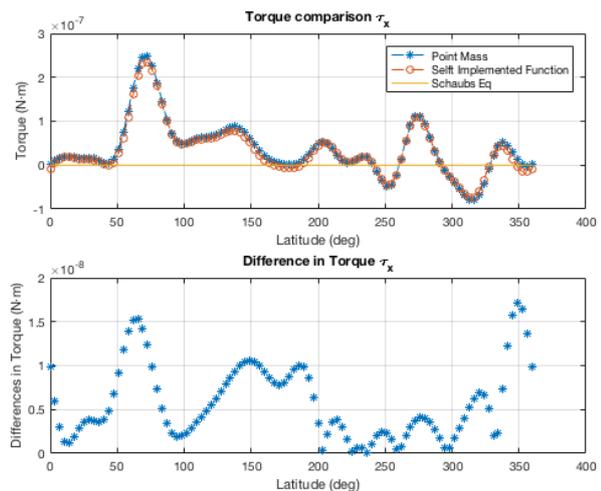
(c) Gravity gradient torque with 512 point masses



(d) Gravity gradient torque with 8000 point masses

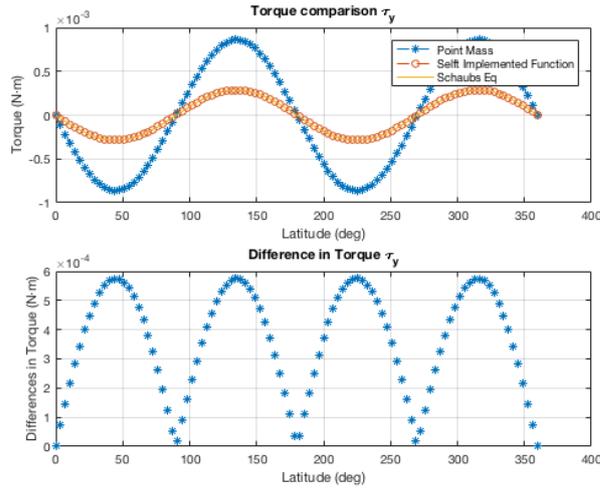


(e) Gravity gradient torque with 32768 point masses

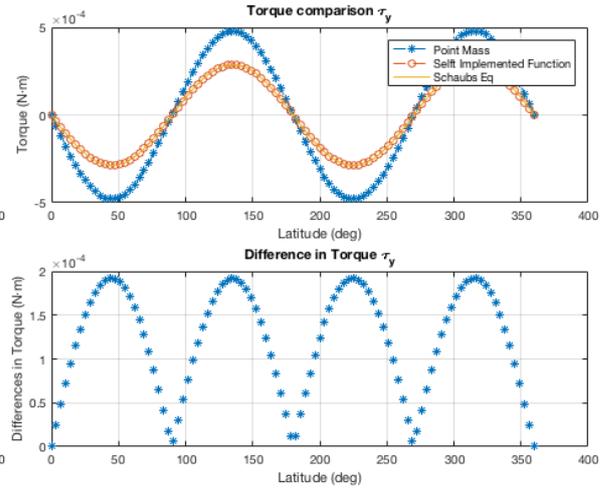


(f) Gravity gradient torque with 262144 point masses

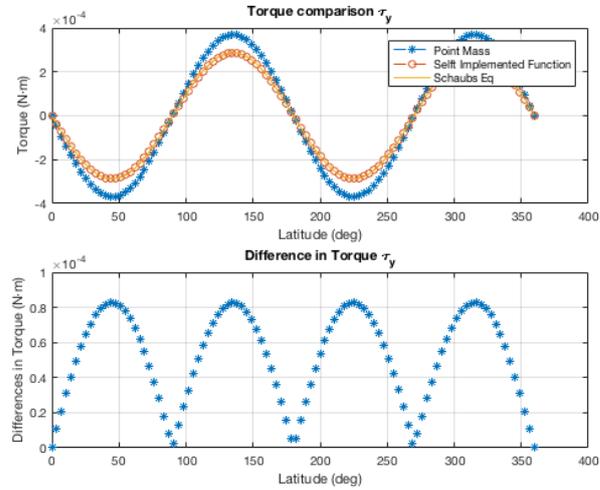
Figure 7.12: Comparison of the gravity gradient torque (τ_x) methods



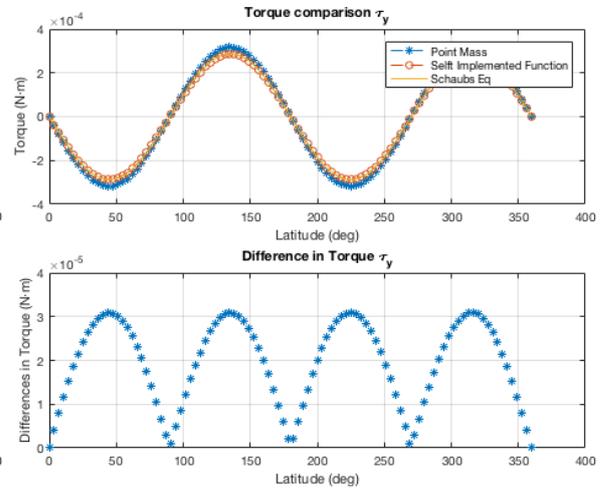
(a) Gravity gradient torque with 8 point masses



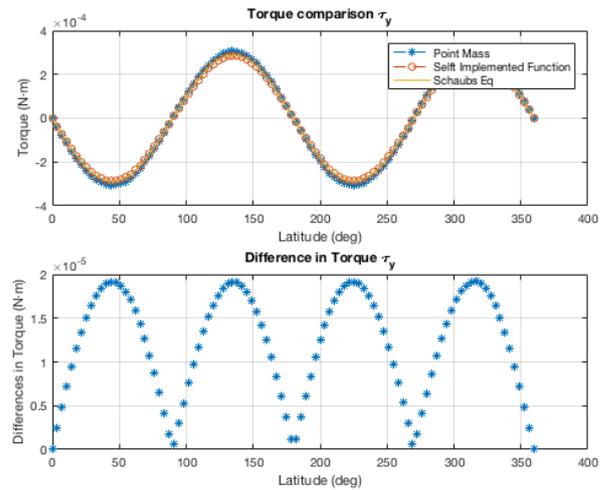
(b) Gravity gradient torque with 64 point masses



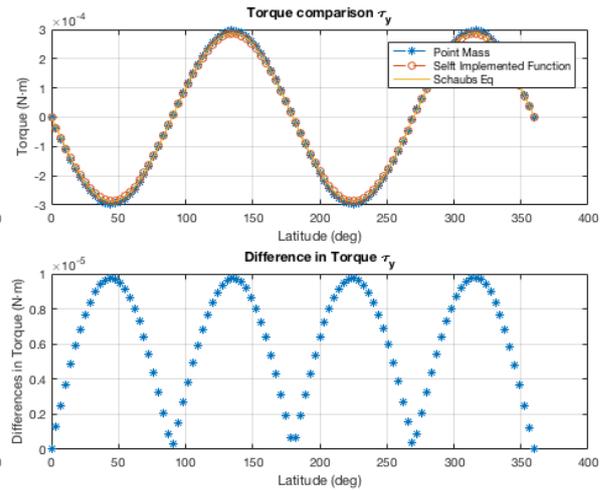
(c) Gravity gradient torque with 512 point masses



(d) Gravity gradient torque with 8000 point masses

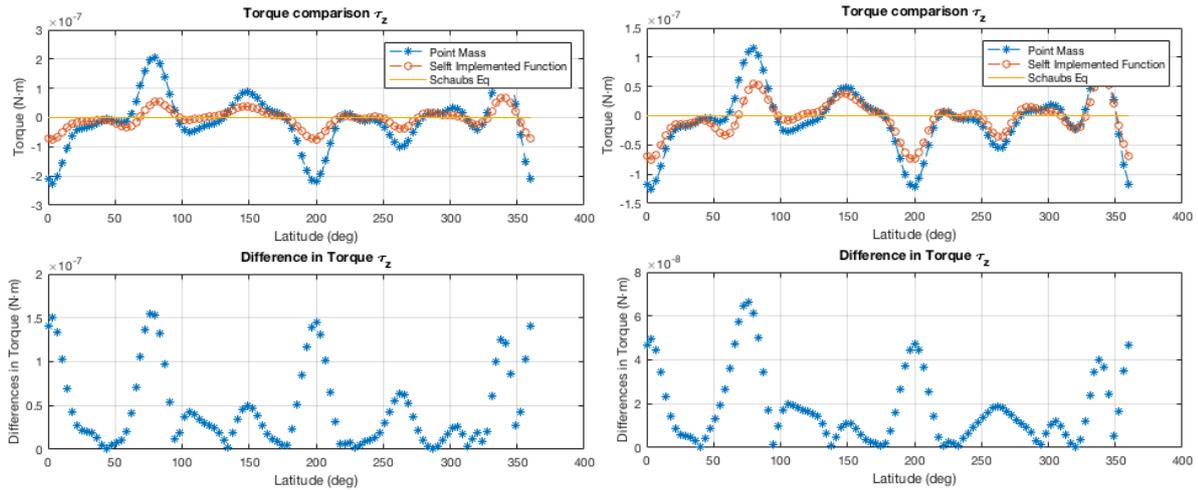


(e) Gravity gradient torque with 32768 point masses



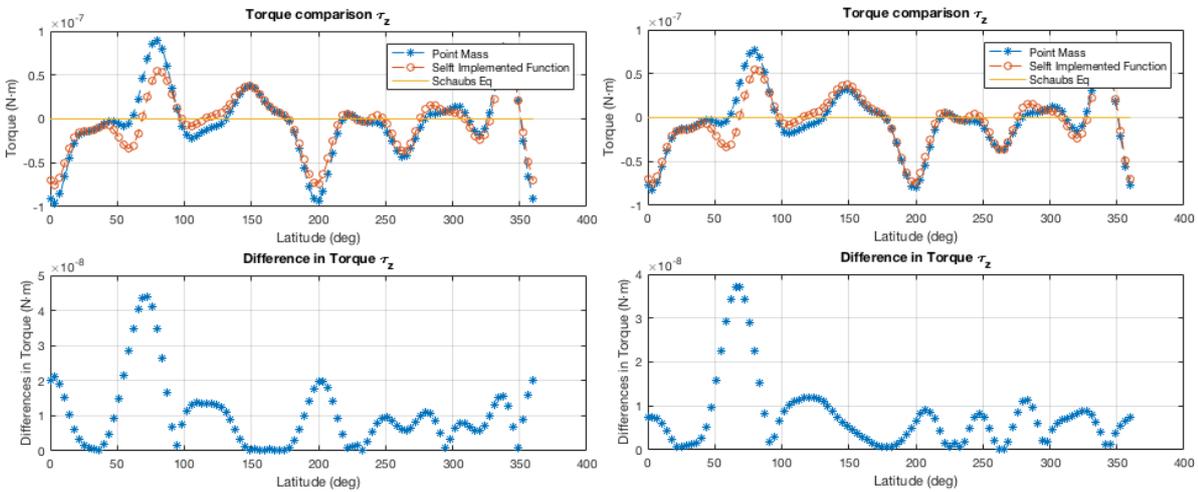
(f) Gravity gradient torque with 262144 point masses

Figure 7.13: Comparison of the gravity gradient torque (τ_y) methods



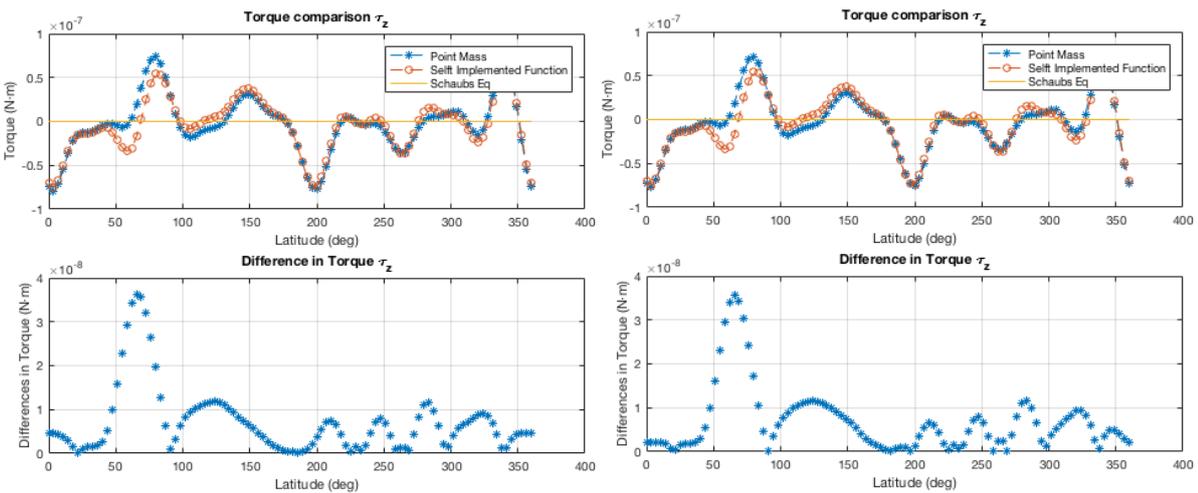
(a) Gravity gradient torque with 8 point masses

(b) Gravity gradient torque with 64 point masses



(c) Gravity gradient torque with 512 point masses

(d) Gravity gradient torque with 8000 point masses



(e) Gravity gradient torque with 32768 point masses

(f) Gravity gradient torque with 262144 point masses

Figure 7.14: Comparison of the gravity gradient torque (τ_z) methods

only one order of magnitude less than the actual torques and all curves follow the same pattern. Since both independent methods produce the same torques, and all them match the analytic equation, it can be verified and validated that the gravity gradient torque sub-block is correct and that it provides the expected physical values indeed.

RUNGE-KUTTA 4 INTEGRATOR

Finally, the last navigation sub-block to be verified and validated is the Runge-Kutta 4 integrator. To do so, different tests have been carried out. The first one consists in studying a kinematic propagation. A rigid body without any force or torque acting on it. If the integrator is well implemented, the rigid body should remain unperturbed forever. Results from this test are just straight constant lines and therefore, there is no need to show them. Both position and velocity components remained constant at zero and the energy did not change at all. The testing time was 700 seconds but it should have worked for any period of time.

The second test consists in comparing the numerical two body problem solution with the analytic one from Kepler. The solutions should be the same if the Runge-Kutta 4 integrator is well implemented. The selected orbit was an equatorial orbit about Bennu at 1500 m and with zero eccentricity and inclination. Figure 7.15 shows both trajectories. It might be difficult to distinguish but there is actually a blue line overlapping the red one, corresponding to the integrated solution. Figure 7.16 shows the differences in position and velocity between the two solutions. As can be seen, the maximum difference in position after 3 orbits is 0.04 m and 0.0015 mm/s in velocity. The integration time step in this case was 1 second. If the time step is increased, the errors also increase and if the time step is reduced, so do the errors. This is due to the numerical integration algorithm. The smaller the time step, the better the solution and longer the time it takes to compute it.

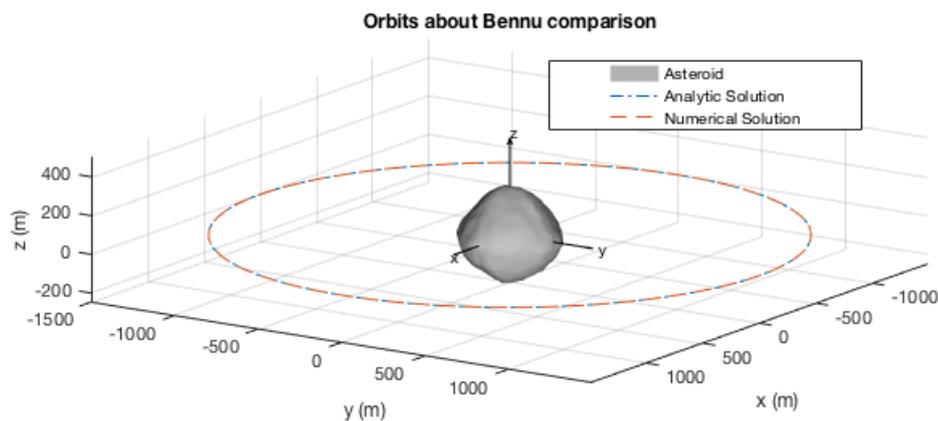


Figure 7.15: Integrated trajectory vs Kepler solution of a two body problem

Another test is the energy check for the same two body problem. In the two body problem, the only force acting on the spacecraft is the gravitational force. It is a central force that comes from a potential function. This means that the total energy should remain constant over time, no matter the number of orbits if there is no other force acting on the spacecraft. The results show that the energy remains constant with a tiny numerical variation shown in Figure 7.17. As can be seen, the energy variations are in the order of magnitude of 10^{-13} J, whereas the total energy is 3.64 J. Therefore, it can be considered constant.

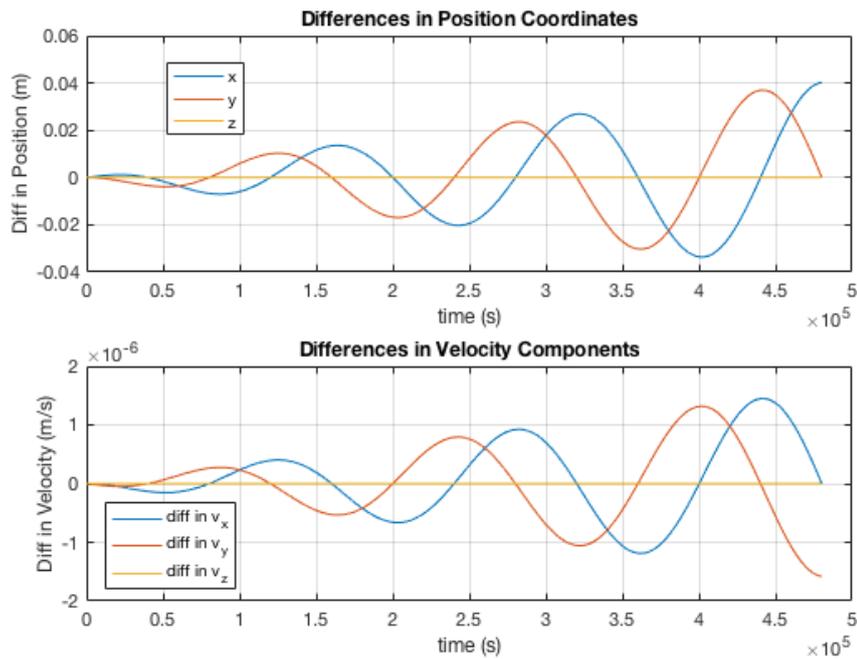


Figure 7.16: Position and velocity comparison of the integrated two body trajectory and the Kepler solution

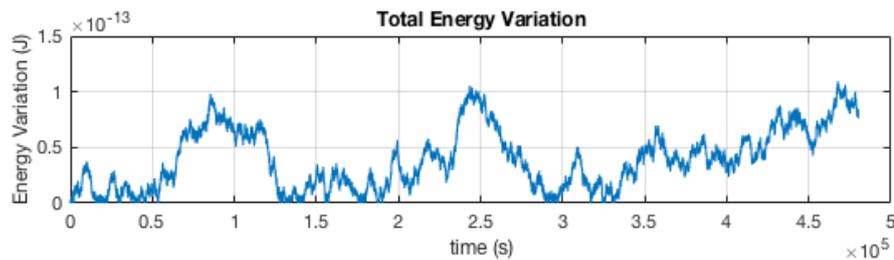


Figure 7.17: Energy variation of the integrated two body trajectory with a time step of 1 second

All these previous tests concerned about the orbital motion. They verify and pseudo-validate that the position and velocity resulting from the numerical integrations are correct and that they represent the desired physical meaning. However, nothing has yet been said about the rotational motion. To check whether or not the attitude integration is correct, the following tests have been carried out.

The first one consists in studying the torque-free motion of a rigid body. For the case of an axis-symmetric body, there is a known analytic solution derived in [Celledoni et al., 2008]. If the Runge-Kutta 4 integrators (classic and dual quaternion) are well implemented, the solution for a given initial condition should be the same as the analytic one. After carrying out these tests, results show that they are indeed the same. The differences depend on the step size of the integration (as it happened with the translational motion and for the exact same reason). With a step size of 10 seconds and an initial angular velocity of $\omega = [0.03, -0.03, 0.015]$ deg/s, the differences in quaternion components were of the order of magnitude of 10^{-5} , and 10^{-8} for the angular velocities components, after 125 hours. The results were exactly the same for both the classic and dual quaternion integrators. For the sake of readability, only the angular velocity from the classical integrator

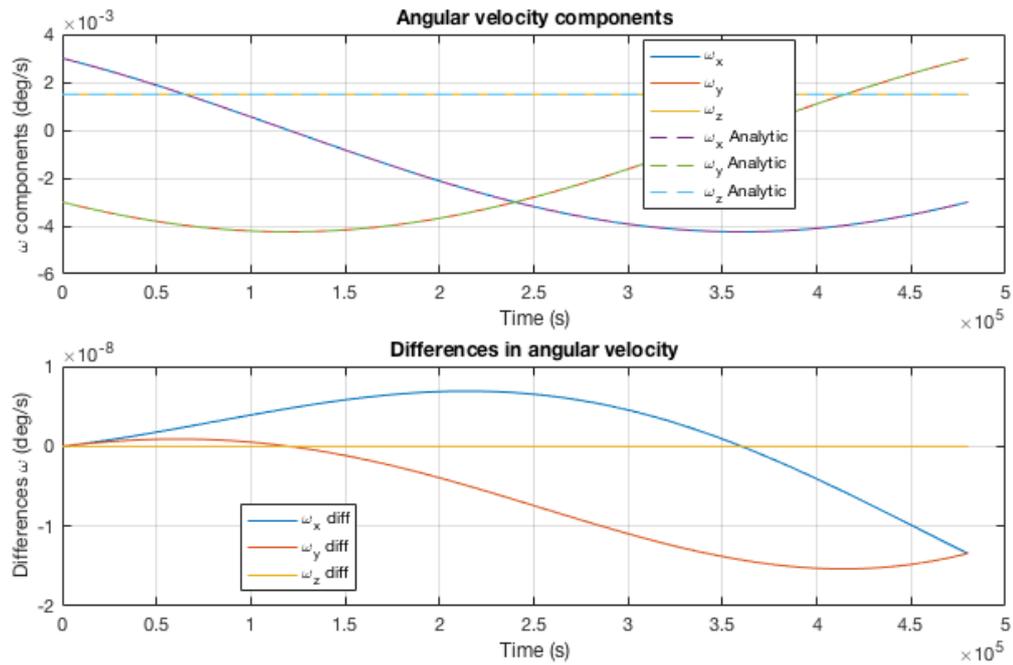


Figure 7.18: Numerical and analytic angular velocities and their differences for a torque-free motion

is shown in Figure 7.18 as an example. The top part shows the angular velocity components and the bottom part, their differences.

The second test consists in checking the rotational energy of the rigid body in the above mentioned motion. Since there are no torques acting on the rigid body, the rotational energy should remain constant. The total angular momentum has to be conserved. As happened with the translational motion, the rotational energy remained constant in all tests, with tinny numerical variations of 10^{-12} times smaller than the mean value. Thus, it can be considered constant.

The third and final test for the rotational motion consists in verifying the implemented integrators, by comparing their solutions with an already verified and trusted solution. With the first and second previous tests, one can only be sure that the numerical integration algorithm works in the absence of torques. Since there is no general analytic solution for the rotational motion in presence of torques, the solutions have to be compared with a previous trusted independent work. In this case, the trusted solution came from a Matlab package called *quaternion*¹, which uses the *ode45* integrator for numerically integrating the rotational equations of motion. This work is totally independent and provides a quaternion solution given the initial conditions and the torques acting on the rigid body, as a function of time.

The rigid body used for this test was the model of the satellite used in all simulations (defined in Section 2.3). A constant torque of $\tau = [0.01, -0.01, 0.005]$ was applied during the 30 minutes simulation. The step size selected was 0.25 seconds. The results of the test showed that both solutions are exactly the same up to some numerical precision. Figure 7.19 shows the quaternion solutions for both cases in the top part. As can be seen

¹Downloaded from MathWorks "https://es.mathworks.com/matlabcentral/fileexchange/33341-quaternion" on 15 January 2018

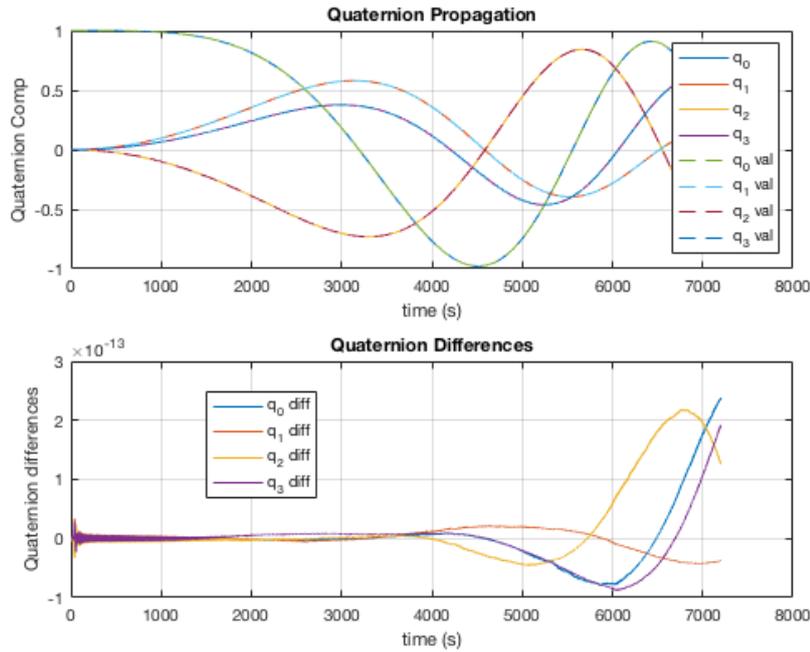


Figure 7.19: Differences between the quaternion solution and a trusted solution for the general rotational motion

in the bottom part, the differences between them are of the order of magnitude of 10^{-13} after 30 minutes. It might be important to note that the differences grow with time. This is because the trusted solution uses the matlab *ode45* integrator. This is a variable step-size algorithm that allows to control the error of your solution. As explained in previous chapters, the numerical algorithm for this thesis is a Runge-Kutta 4, which is a fixed step-size method. Due to the fact that a constant torque is applied, the rotational rates increase over time, and smaller time-steps are required after a period of time to maintain the error constant. Since the Runge-Kutta 4 is maintaining the same time-step, the errors increase over time. The angular velocities are not shown for the sake of readability, but their differences after 30 minutes were of the order of magnitude of 10^{-15} deg/s. These tests serve for verifying and validating that the implemented code works as expected and provide a solution, which corresponds to real physical solution of the rotational motion.

7.2.2. GUIDANCE

The guidance block is responsible for determining the reference trajectory to be followed. Normally, it should allow for changes in the reference trajectory depending on different inputs. As has been explained in the previous section, a possible example could be to abort the desired trajectory and enter a safe mode pointing towards the Sun because something went wrong. In this master thesis however, the reference trajectories will not change at all. Therefore, the unit test t make sure that the guidance block has been properly implemented is to check these reference trajectories.

The first scenario is to orbit about Bennu. The orientation of the satellite should be to point towards the center of the asteroid at all times, as described in Section 2.3. To verify that the guidance block works

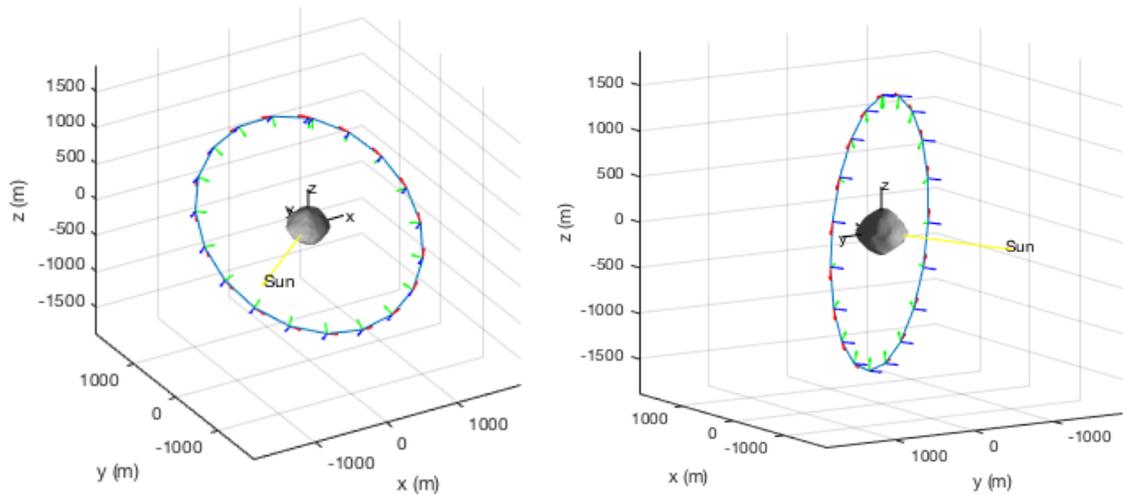


Figure 7.20: Orbiting reference trajectory 3D view

as expected, a graphical representation of the trajectory with the orientation of the spacecraft have been plotted. Figure 7.20 shows a three-dimensional view of the desired motion for the orbiting scenario. The blue-red-green arrows represent the body axes of the spacecraft. As can be seen, the green arrows (z body axis) always point towards the center of the asteroid. A simple test has been carried out to make sure of that. The position vector in the **ACI** frame has been dot-multiplied with this z body axis at all times and the result has always been zero. This proves that their directions are always parallel and that the satellite is indeed pointing towards the center of the asteroid at all times. The blue arrows (x body axis) are always parallel to the Sun direction. A dot-product multiplication between these axes and the Sun vector has been carried out to make sure of that. Finally, both the linear and angular velocity have been numerically differentiated and compared with the values coming from the guidance block. The results have been exactly the same. With these tests and the graphical visualization of the trajectory, one can be sure that the guidance block is indeed working as expected and that its outputs are correct.

The second scenario is to hover around Benu. Similar tests have been carried out for this case. The position and velocity of the spacecraft have been checked to be constant at all times, with the correct values. All zeros in the case of the velocity for example. In the hovering scenario, the spacecraft has to point towards the center of the asteroid again. The same test than for the orbiting scenario has been carried out with successful results. A graphical three-dimensional representation of the motion can also be seen in Figure 7.21. As can be seen, the outputs from the guidance block for the hover case are now proven to be correct.

Finally, the last and third scenario is to land on the surface of Benu. The satellite should point towards the center of the asteroid at all times as well. The exact same tests than for the previous scenarios, have been carried out with successful results. Figure 7.22 shows the graphical three-dimensional representation of the reference trajectory. As can be seen, the spacecraft is pointing towards the center of the asteroid during all the landing procedure. The blue-red-green arrows still represent the orientation of the spacecraft (body axes). As an additional test for the landing scenario, the position vector of the satellite has been dot-multiplied with the x axis of the asteroid in this case. Since the spacecraft is supposed to follow this axis until landing, the

results have been always zeros. This proves that these vectors are always parallel and that the spacecraft is indeed following the x axis of the asteroid. The velocity has been expressed in the asteroid rotating frame and checked that it is indeed 10 cm/s at all times. For these reasons, the guidance block can be said to be verified and that it provides the expected reference trajectories.

7.2.3. CONTROL, ACTUATORS AND SENSORS

The next sub-blocks to be verified and validated are the control and actuators sub-blocks. They have been grouped together because they can be verified and validated at the same time with a single unit test. The purpose of these blocks is to compute a force and torque command to steer the spacecraft back to its reference trajectory. This has to be carried out with some limitations coming from the actuators, defined in Chapter 2. The unit test consist in running several test cases and verifying that the controls commands are actually

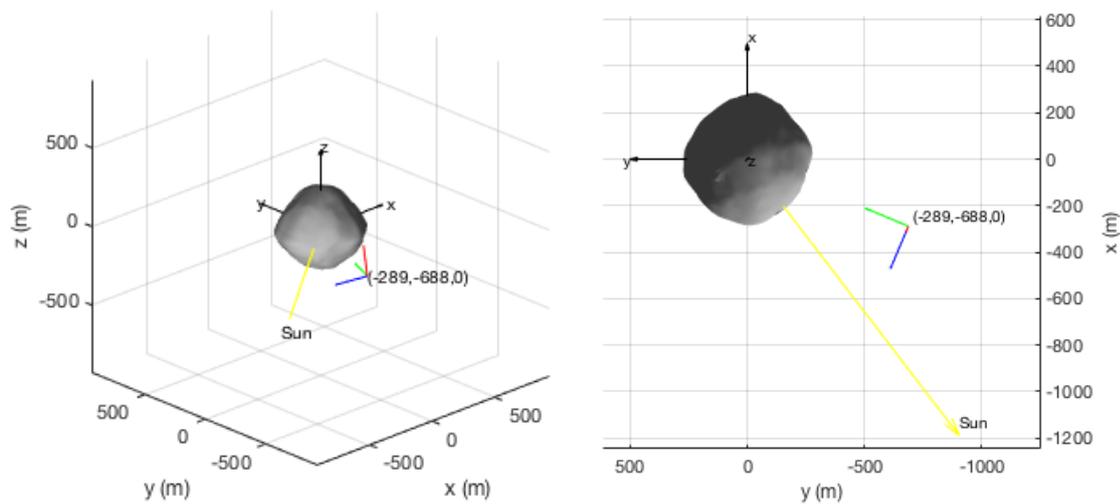


Figure 7.21: Hovering reference trajectory 3D view

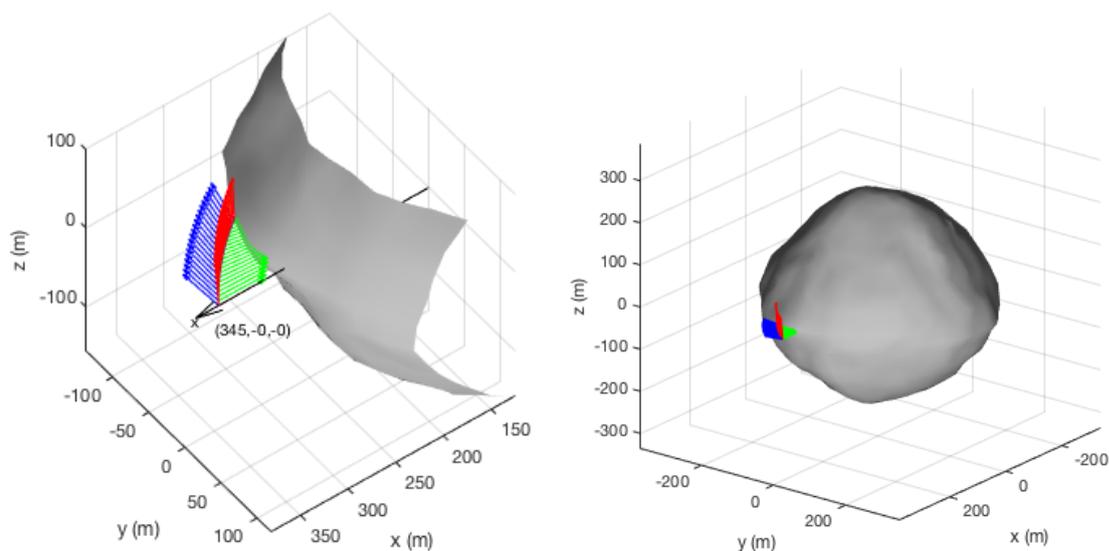


Figure 7.22: Landing reference trajectory 3D view

correct.

The first test case consists in evaluating the control commands when the errors in all state variables are zero. As has been explained in Chapter 5, the control commands when there is no error, should be the reference accelerations indeed. This can be seen in Equations (5.8), (5.14) and (5.15). It is exactly the same for the classical method than for the dual quaternion method. Since the guidance block has been already verified, the outputs from the control sub-block have been compared with the reference accelerations for the case of zero error. Results show that they are indeed the same.

The second test case consists in computing the control commands when the reference accelerations are zero and there is actually a certain given error in the state variables. The computed commands should be these errors multiplied by the control gains. This test has been carried out for both the classic and dual quaternion representations, and the results have been correct.

The third test case consists in running several different simulations and checking that the control commands actually steer the spacecraft through the reference trajectory. For the sake of readability only one example will be shown. However, this test has been carried out for all three mission scenarios and with different initial conditions each. This example consists in hovering on Bennu with certain initial errors in all state variables. The results from this test case show that the controller is actually capable of correcting these initial errors and bring the spacecraft back to the reference trajectory. Figures 7.23a and 7.23b show the control forces and torques of this simulation, for both the classic and dual quaternion representation. As can be seen, the forces and torques are saturated at 20 N and 0.2 Nm at the beginning, and then released. This proves that the actuators sub-block is well implemented and that it does not allow to outreach the actuators capabilities defined in Chapter 2. As can be seen, all control commands converge to almost zero, with final values corresponding to the reference accelerations. This proves that the control sub-block is well implemented and that it is able to steer the spacecraft back to the reference trajectory. The same successful results happened in all the simulated test cases, proving that the control and actuators sub-blocks work as expected.

Finally, since there are actually no sensors, the spacecraft motion is believed to be exactly the same coming out from the numerical integrations. For this reason, it is not possible to verify or validate this sensor block. It has been grouped together with the previous blocks for compactness of the document.

7.2.4. SYSTEM TEST

Last but not least, a system test has been carried out for each of the mission scenarios. The system test consists in verifying and validating that the simulator, as a complete system, works as expected and provides valid results. So far, all different sub-blocks have been separately verified and validated through the unit tests defined in previous sub-sections. However, the combination of all sub-blocks working together to achieve the final goal, has not been tested yet. This is the purpose of the system test; to check whether or not the simulator, as a system, works correctly.

Taking this into account, a different system test has been run for each mission scenario. All the input parameters and specifications of each simulation are summarized in Tables 7.3 to 7.5. As can be seen, all

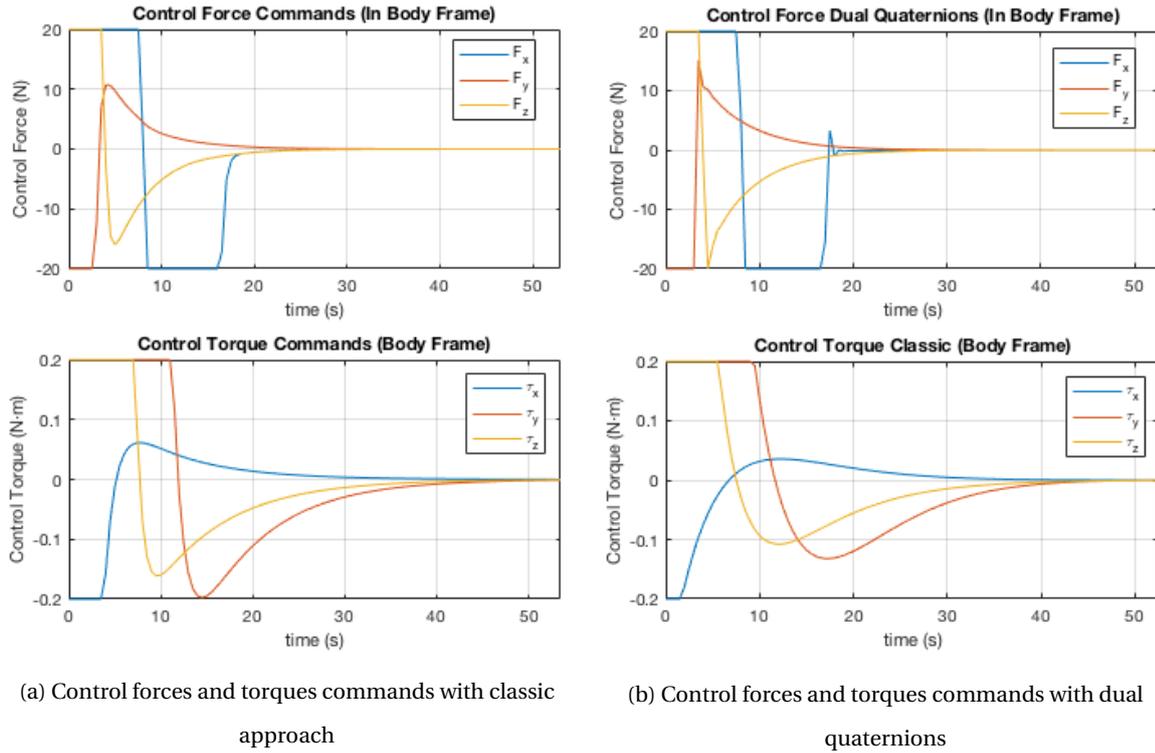


Table 7.3: Parameters for orbiting system test

Parameter	Value
Initial position (ACI frame)	[1500 ; 0 ; 0] m
Initial velocity (ACI frame)	[0.0356 ; -0.0469 ; 0] m/s
Initial attitude	[1 ; 0 ; 0 ; 0]
Initial angular velocity (Body frame)	[0 ; 0 ; 0] · 10 ⁻³ rad/s
Integration and control freq.	IF = 0.1 Hz; CF = 0.1 Hz
Time span	0 - 44.46 h
Classic control gains	$K_q = 4.46, K_\omega = 170$
Dual quaternion control gains	$K_q = 5.57, K_\omega = 214.28$

simulations have a starting initial error in all state variables. In this way, it is possible to see if the controllers are able to steer the spacecraft back to the reference trajectory. The control gains have been selected so that the spacecraft has the minimum settling time and and minimum error. It is important to notice that for these system tests, the procedure described in Section 5.3 for gains optimization, has not been followed. The only goal of these tests was to verify the behavior of the simulators.

After carrying out these system tests for the three mission scenarios, the simulator worked as expected and provided successful results. The spacecraft was able to follow the reference trajectory in all cases. All different sub-blocks worked together as expected, providing a valid solution to each of the mission scenarios.

For the sake of readability, all these system tests results are shown in Appendix B. Only a representative example proving that the spacecraft is indeed following the reference trajectory is shown in this section. This

Table 7.4: Parameters for hovering system test

Parameter	Value
Initial position (ACI frame)	[-287.5639 ; -688.9294 ; 0.5000] m
Initial velocity (ACI frame)	[-0.0100 ; 0.0100 ; 0.0050] m/s
Initial attitude	[0.1937 ; 0.0977 ; 0.7010 ; 0.6794]
Initial angular velocity (Body frame)	[0.1745 ; -0.1745 ; 0.0873] · 10 ⁻³ rad/s
Integration and control freq.	IF = 2 Hz; CF = 2 Hz
Time span	0 - 600 s
Classic control gains	$K_q = 285.17, K_r = 388.89, K_\omega = 1272.70, K_v = 1955.60$
Dual quaternion control gains	$K_q = 97.78, K_r = 6.22, K_\omega = 505.56, K_v = 6$

Table 7.5: Parameters for landing system test

Parameter	Value
Initial position (Inertial frame)	[346.46 ; -1 ; 0.5] m
Initial velocity (Inertial frame)	[-0.11 ; 0.01 ; 0.05] m/s
Initial attitude	[0.4678 ; 0.5218 ; -0.4854 ; -0.5228]
Initial angular velocity (Body frame)	[0.0932 ; 0.6336 ; -0.0191] · 10 ⁻³ rad/s
Integration and control freq.	IF = 2 Hz; CF = 2 Hz
Time span	0 - 600 s
Classic control gains	$K_q = 285.17, K_r = 303.02, K_\omega = 1272.70, K_v = 1997.56$
Dual quaternion control gains	$K_q = 117.14, K_r = 5, K_\omega = 560, K_v = 5$

example corresponds to the landing scenario for the classical representation. Figure 7.24 shows the evolution of all the different reference state variables over time. As can be seen, position and velocity are expressed in the ACI reference frame, showing the reference trajectory to be followed by the spacecraft. The errors in the trajectory after running the system test are shown in Figure 7.25; where the differences between all state variables and their reference are shown. As can be seen, after a settling time, all variables manage to come back to the reference values despite the initial errors.

There is always an overshoot to correct the trajectory, and then, all variables remain within the mission requirements. In this specific system test, the overshoot for the position has a maximum of 1 m. This means that the spacecraft is 1 m away of its nominal trajectory and then goes fast to its nominal position. For the velocities, the maximum overshoot is 0.1 m/s; whereas it is 0.075 deg/s for the angular velocities. The attitude does not have an overshoot. In this case, it is represented as the angle to be rotated so that the spacecraft is pointing to its desired direction. In this system test, it starts with a pointing error of 1.5 deg and it quickly goes to zero. The blue boxes represent a zoom-in of specific regions of the plot. In this way, it is easy to see that all components are indeed converging to the reference trajectory, within the mission requirements.

Finally, Figure 7.26 shows the different control commands generated by the control block over time. As can be seen, both the forces and torques are saturated at the beginning. Then, they all go to almost zero values to steer the spacecraft so that it follows the reference trajectory. For this specific system test, the translational

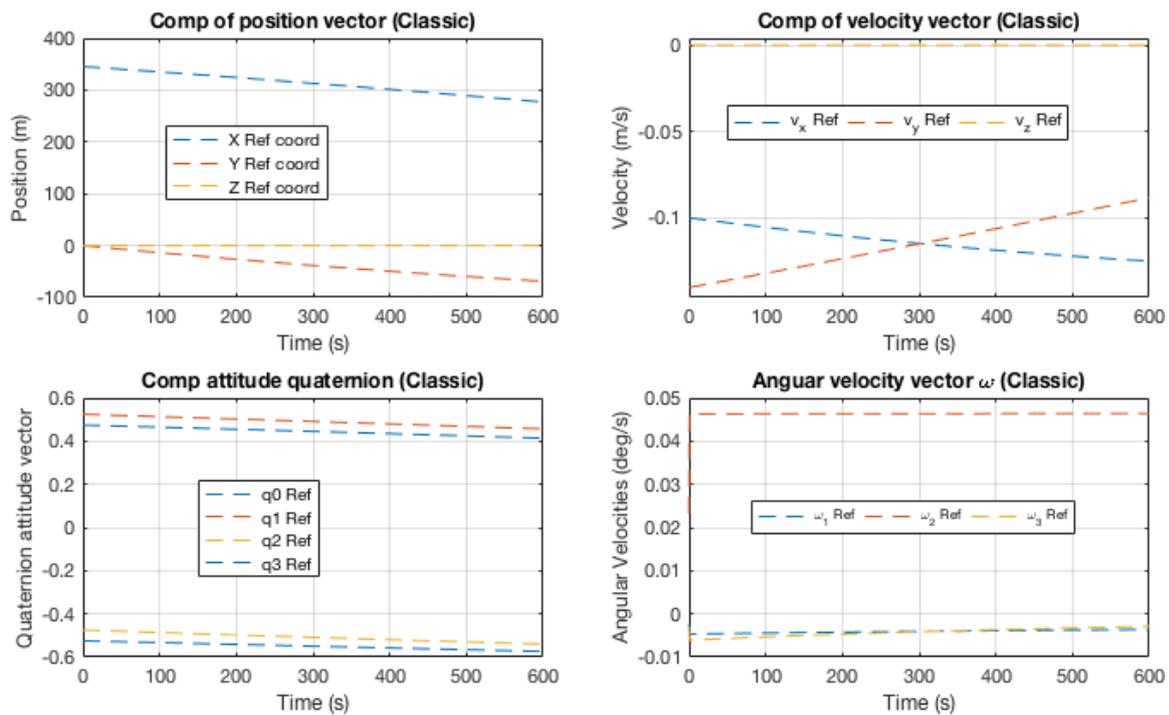


Figure 7.24: Reference state variables values for the landing scenario (obtained by the classical method)

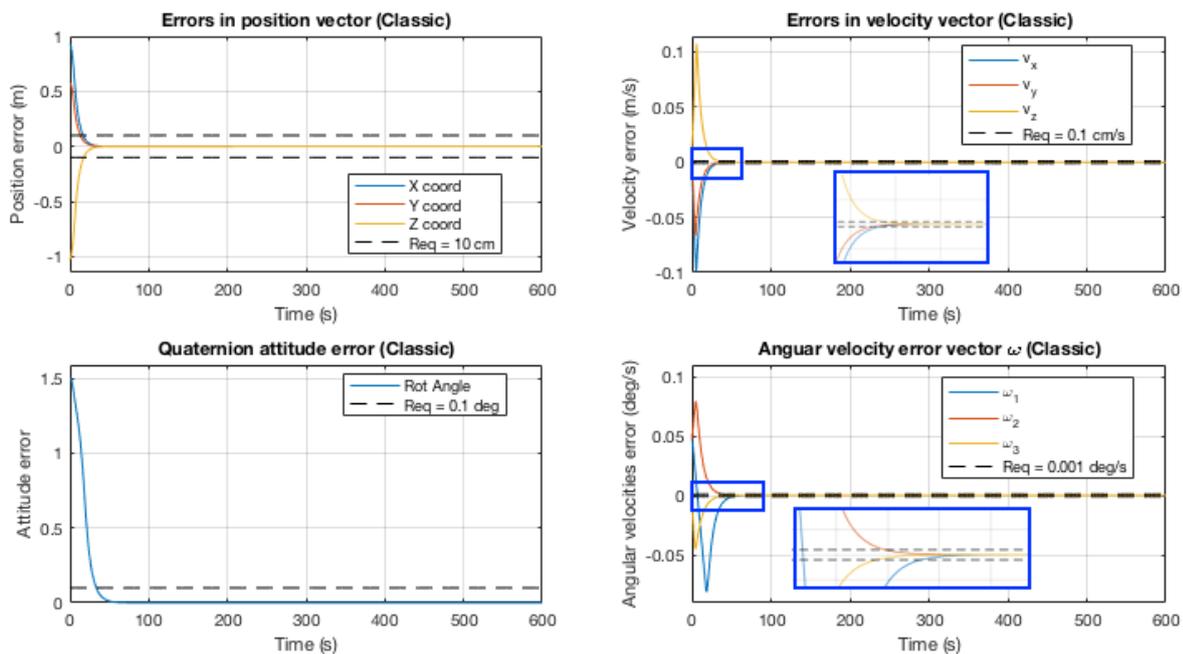


Figure 7.25: State variable errors with respect to reference trajectory with classical approach

motion reaches stability faster than the rotation motion.

Bearing all these results in mind, it is now possible to arrive to the conclusion that the implemented simulator works as expected. Now, it has been verified and validated, and it is ready to provide useful results for comparing the dual quaternion approach with the classical representation.

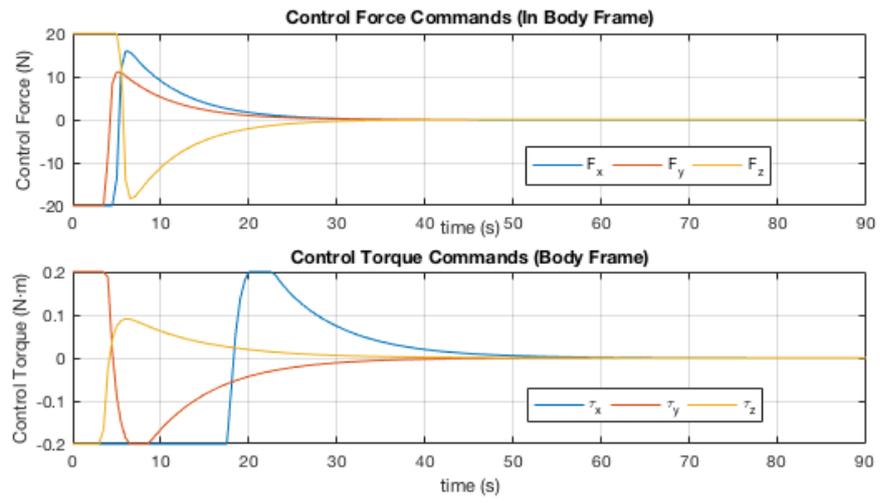


Figure 7.26: Control forces and torques commands with classical approach

8

RESULTS

This chapter presents the results of this master thesis. After having developed a simulator following the steps described in Chapter 7, it is possible to compare both methods and answer the research questions. First of all, in Section 8.1, a simple kinematic propagation test is carried out for determining the computational time of each representation to solve the equations of motion. Second, a more complex scenario is presented in Section 8.2 where the comparison is carried out for a two body problem. The accuracy of the solutions is compared versus the time spent. Third, in Section 8.3, both control approaches are compared in the nominal scenarios of orbiting, hovering and landing. Finally, in Sections 8.4 and 8.5, a sensitivity analysis of the control frequency is carried out, together with a stability comparison. With all these comparison tests, some very interesting conclusion are drawn.

8.1. KINEMATIC PROPAGATION

The first candidate advantage of dual quaternions was that they could be faster than classical methods because there are only two equations of motion to be solved, instead of four. Since the rotational and translational motions are coupled, the integration can be carried out at the same time (as explained in Chapter 4) and could take less computational time. To verify this hypothesis, a very simple test has been performed. It consists in doing a kinematic propagation of a rigid body without the influence of any force or torque. The goal is to determine the computational time each method takes to integrate the equations of motion, given some equal initial conditions.

The test is defined as a rigid body spinning about the y body axis with an initial angular velocity of $\omega_x = 0$, $\omega_y = 0.02$ and $\omega_z = 0$ rad/s and a translational initial velocity of $v_x = 1$, $v_y = 2$ and $v_z = -1$ m/s. Its initial position is the origin of the inertial reference frame and its initial orientation is defined with the quaternion

Table 8.1: Time to run the kinematic propagation with step-size of 1 s

	Classical Method	Dual quaternion method
Time	0.6855 s \pm 0.004 σ	1.3603 s \pm 0.0174 σ

components $q_0 = 1$, $q_1 = 0$, $q_2 = 0$ and $q_3 = 0$. The integration has been carried out for the duration of 3000 s, with a range of different time-steps (from 10 to 0.01 s).

Since the computer used for all the simulations in this master thesis is a Macintosh, the time it takes to run a matlab routine is not constant. Because of the multi-tasking nature of its processor, depending on the number of other applications running at the same time, more or less computational power is dedicated to run the routine. This makes the computational time to change a little bit. To cancel this drawback, all simulations have been carried out several times and only the mean value with its standard deviation has been considered. In all cases, only Matlab was running and all other applications were closed. For this specific test, all simulations were carried out 1000 times for each time step. The results from this test show that dual quaternions always take more time to propagate the equations of motion. For the sake of readability, only results from the time-step of 1 s are presented in Table 8.1.

As can be observed, with a time step of 1 s, the dual quaternion representation takes almost twice as time as the classical representation. This ratio between methods is independent of the integration time-step. This clearly shows that the initial hypothesis of dual quaternions being faster, is not true. Despite the fact of only having two equations of motion, dual quaternions take more time to propagate the six degrees of freedom of a solid rigid. The reason for this is hidden in the dual quaternion operations. Even though there are only two equations of motion with less dual quaternion operations, the total number of basic operations that has to be computed is much larger. As defined in Section 3.2, the dual quaternion multiplication, for instance, involves three separate quaternion multiplications and several vector summations. For this reason, the total number of basic operations involved in each kinematic propagation has been computed. It turns out that, for the dual quaternion representation, it is almost four times larger than for the classical representation. Basic operations have to be understood as additions, subtractions, multiplications and divisions. And this is the reason why the dual quaternion representation takes more time.

8.2. TWO BODY PROBLEM

The fact that dual quaternions take more time for a simple kinematic propagation, does not necessarily mean that they have to be always slower for any application. One of the main advantages of dual quaternions is that they naturally couple the rotational and translational motions. They integrate both motions at the same time, whereas in the classical method, they have to be integrated separately. Due to this natural coupling, despite the fact of taking more time in each integration, they could allow for longer time-steps, resulting in a reduction of the total time. Another possibility is that for a given integration frequency, dual quaternions could be more precise than classical representations. To test these hypothesis, a more complex scenario have

been considered. Since these hypothesis are based on the natural coupling between motions, these coupling effects have to be present. For this reason, a two body problem with the spacecraft orbiting the asteroid has been selected. The forces involved are: the gravitational force approximated by spherical harmonics, the SRP force and third body perturbations. The only torques affecting the spacecraft are the gravity gradient ones.

The goal of this test is to determine if dual quaternions allow for larger step-sizes and if they can be more precise than classical representations. To achieve this goal, one has to integrate the equations of motion with a variable step size method. Then, looking at the relationship between computational total time and precision, some conclusions can be drawn. For this reason, several orbits with different initial conditions have been simulated and studied. The result of each simulation is a plot of the time that it takes, by each method, to integrate the solution, given a certain tolerance error. In this way, it is possible to determine which method is more precise given a certain computational time. Or in other words, which method takes more time given a certain tolerance error.

To do so, the matlab *ode45* function has been used. It is an already implemented numerical integrator, based on the Fehlberg method (or commonly known as Runge-Kutta 4(5)). The error in the solution can be estimated and controlled by using the higher-order embedded method that allows for an adaptive step-size to be determined automatically. To generate the plots, the total time spent to compute the solution has been calculated for a range of different error tolerances. For each error tolerance, the simulation has been carried out 100 times, and only the average time spent has been taken into account with its standard deviation. It is important to mention that, to control the error of the solution, the parameters *RelTol* and *AbsTol* have been used. They are *odeset* input parameters and it is crucial to understand them.

RelTol is the relative error tolerance. It measures the error relative to the order of magnitude of each solution component. Roughly speaking, it controls the number of correct digits in all solution components. *AbsTol* is the absolute error tolerance. It is a threshold below which the value of the solution becomes unimportant. If the solution is smaller than the *AbsTol*, then the solver does not need to obtain any correct digits. For this reason, the value of this tolerance should take into account the scale of the solution components. In this thesis, the order of magnitude of some state variables is 10^{-3} , for this reason, it has been established that, the *AbsTol* should never be smaller than 10^{-6} . For the sake of simplicity, from now on, the expression 'error tolerance' will refer to both *RelTol* and *AbsTol* with the same value, if not indicated otherwise.

The first orbit scenario is a circular orbit ($e = 0$) about Bennu, with a semi-major axis of 1500 m and without inclination ($i = 0$). The initial orientation is defined to be $\mathbf{q} = (1, 0, 0, 0)$ and no initial angular velocity is considered. The simulation time is one orbital period, which is about 44.5 hours for this specific orbit. The results from this scenario are shown in Figure 8.1. As can be seen, the dual quaternion approach always takes more time to compute the solution given a certain error tolerance. The smaller the error tolerance, the bigger the difference between methods. With an error tolerance of 10^{-14} , the dual quaternion method takes 42.26 s, whereas the classical one only takes 12.46 s. This is almost 3.4 times slower. For this specific orbit, it is clear that the previous hypothesis was wrong. Despite the fact of naturally coupling the rotational motion with the translational one, the dual quaternion representation does not allow for longer steps sizes, and it still takes more time to compute the solution given a certain error tolerance.

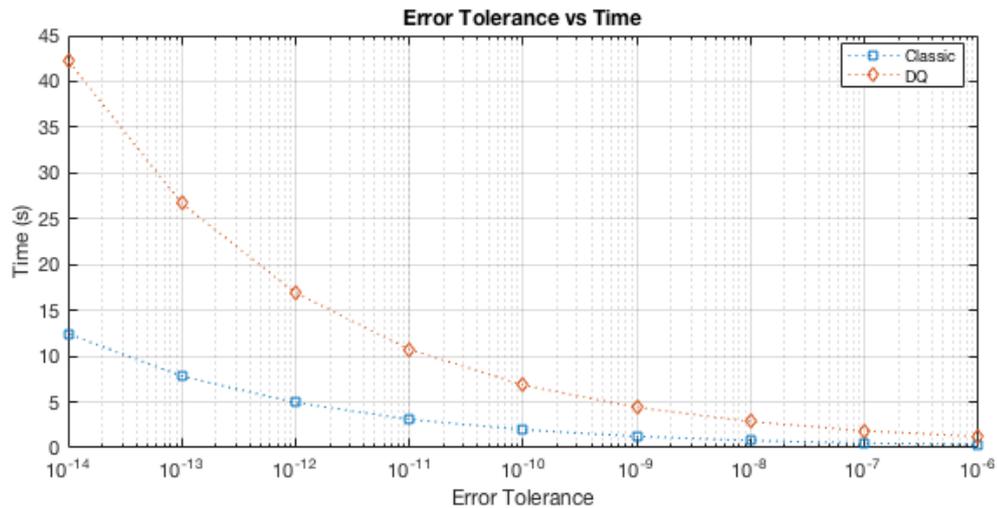


Figure 8.1: Simulation time versus error tolerances for circular orbit at 1500 m

To make these results more general, four other different orbit scenarios with different initial conditions have been simulated as well. In the first scenario, the orbit did not have eccentricity or inclination. Also, the angular speeds were very small. If the goal is to analyze the benefits of dual quaternions, extreme scenarios with stronger couplings between motions have to be tested. That is why the next following tests try to go to the limits of orbiting scenarios. A second scenario, for instance, consists in performing another circular orbit ($e = 0$) but closer to Bennu. This time, the semi-major axis has been set to only 500 meters. The inclination has been kept at zero ($i = 0$). The initial orientation has been defined to be $\mathbf{q} = (1, 0, 0, 0)$ as well, but now, some initial angular velocity is taken into account: $\omega_x = 0.001$, $\omega_y = -0.001$ and $\omega_z = 0.0005$ rad/s.

A third orbit scenario is an equatorial elliptical orbit ($e = 0.44$) with periapsis at 500 m. The initial position in the **ACI** frame has been defined to be $\mathbf{r}_0 = (0, 500, 0)$ m with initial velocity of $\mathbf{v}_0 = v_c(1.2, 0, 0)$ m/s. v_c is the circular orbital velocity at this initial point. The initial orientation has been defined to be $\mathbf{q} = (1, 0, 0, 0)$, with an initial angular velocity of $\omega_x = 0.001$, $\omega_y = -0.001$ and $\omega_z = 0.0005$ rad/s.

A fourth orbit scenario consists in the same elliptical orbit ($e = 0.44$), but now some inclination is taken into account ($i = 45^\circ$). The initial position in the **ACI** frame has been defined to be $\mathbf{r}_0 = (0, 500, 0)$ m with a velocity of $\mathbf{v}_0 = v_c(\sqrt{2}/2, 0, \sqrt{2}/2)$ m/s. The initial orientation was $\mathbf{q} = (1, 0, 0, 0)$, with initial angular velocity components: $\omega_x = 0.001$, $\omega_y = -0.001$ and $\omega_z = 0.0005$ rad/s.

Finally, to go to even more extreme situations, a fifth and last scenario has been defined as an elliptical orbit ($e = 0.7$) with an inclination of 45° and periapsis at 416 m. Moreover, the angular velocities have been increased to $\omega_x = 0.01$, $\omega_y = -0.01$ and $\omega_z = 0.005$ rad/s, which means ten times faster than previous scenarios.

In all these scenarios, the results have been exactly the same than the first scenario. For this reason and for the sake of readability, they are not shown in this document as figures. Dual quaternions always take more time to compute the solution given a certain fixed error tolerance. However, one relevant aspect has been discovered. The more extreme the scenario is and the more coupled the rotational and translational

Table 8.2: Times dual quaternions are slower than classical method

	1st scenario	2nd scenario	3rd scenario	4th scenario	5th scenario
Times DQ slower	3.3	2.2	1.9	1.6	1.4

motions are, the smaller the differences between methods become. Taking the error tolerance of 10^{-14} , the ratio between the time spent for the dual quaternion method and the classical one is summarized in Table 8.2. As can be seen, the ratio is always higher than one but gets reduced in each scenario.

These results state that, for the tested orbiting scenarios, dual quaternions take more time to compute a solution with a given error tolerance than the classical representation. They also show that, for these scenarios, dual quaternions do not allow for larger step sizes. However, these results also suggest that, maybe in a highly coupled rotational and translational dynamics situation, dual quaternions might still have some advantages, or at least be equal than classical representations, in terms of precision and computational speed. This suggestion, of course, has to be tested. And there is no better scenario than the ones described in Section 2.3 of this thesis for testing them.

8.3. NOMINAL SCENARIOS

So far, it has been proven that for the tested scenarios, dual quaternions are slower than classical representations to achieve the same level of accuracy in the solutions. However, no control has been applied yet. In the previous scenarios, the coupling between the translational and the rotational motion was present because of the dependency of some forces with the attitude of the spacecraft (such as the SRP force); and the dependency of the gravity gradient torque with the position of the spacecraft. However, these couplings are weak and do not have a clear impact for the simulated periods of times. When they do become important is when introducing controls laws. Since the direction of the control thrust force is dependent on the spacecraft orientation, attitude motion can deflect an imposed control force and affect translational motion in this way [Wang and Xu, 2014]. For this reason, comparing the performance of each method in the mentioned scenarios, with the spacecraft trying to follow a given reference trajectory, will allow to test the hypothesis mentioned in the previous section. This will also determine which advantages or disadvantages, the dual quaternion representation has in terms of controllability, energy consumption and stability.

It is important to mention that, in the three close-proximity scenarios defined in Section 2.3, the gains have been optimized following the methodology described in Section 5.3. The spacecraft always starts in the reference trajectory and therefore, the initial errors are always zero.

8.3.1. ORBITING

The first scenario defined in Section 2.3 is the orbiting scenario. It consists in orbiting Bennu in a Sun-synchronous terminator orbit at 1500 m. The initial conditions and all the parameters for the simulations are summarized in Table 8.3. The selected gains are the ones found in Section 5.3.

Table 8.3: Initial conditions and settings for orbiting simulation

Parameter	Value
Initial position (Inertial frame)	[0; 0; 1500] m
Initial velocity (Inertial frame)	[0.03563; -0.04687; 0] m/s
Initial attitude	[0; -0.3193; 0.9476; 0]
Initial angular velocity (Body frame)	[-1.9626; -0.000002106; -0.1074] $\cdot 10^{-5}$ rad/s
Integration and control freq.	IF = 1 Hz; CF = 0.5 Hz
Time span	1 orbital period ($1.6 \cdot 10^5$ s)
Classic control gains	$K_q = 1165.0, K_\omega = 1129.0$
Dual quaternion control gains	$K_q = 208.3, K_\omega = 1108.0$

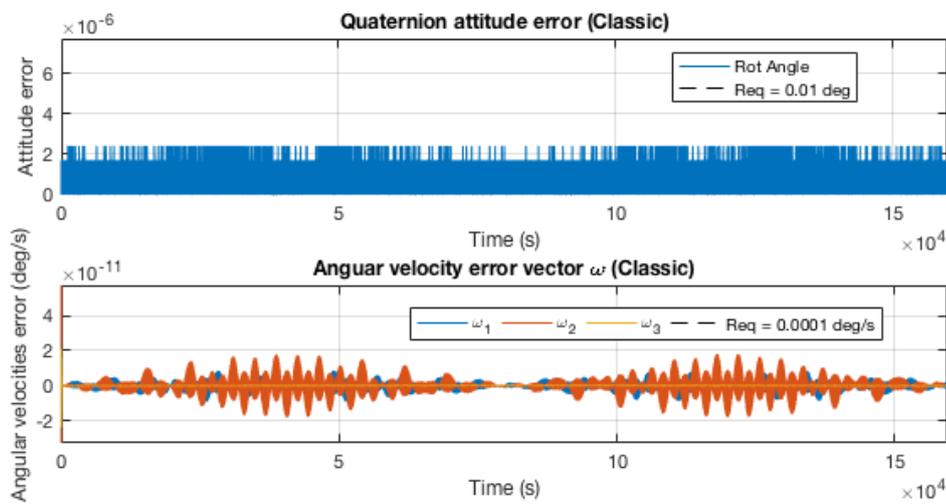


Figure 8.2: State variable errors with respect to reference trajectory with classical approach

After performing the same simulation with the classical method and the dual quaternion one, the results obtained are almost identical. For this reason and for the sake of readability, only the classical results will be shown in this case; and the tiny differences will be explained.

As a first result, both control approaches succeeded in guiding the spacecraft through the reference trajectory, and within the requirements defined in Section 2.2. Figure 8.2 shows the errors between the classical solution and the reference trajectory. It is important to mention that, for the orbiting scenario, only the rotational motion had to be controlled. This scenario tries to represent real missions, where the spacecraft is orbiting in a stable orbit around the asteroid, and only attitude control is required for scientific purposes. For this reason, only errors in attitude and angular velocities are shown.

As can be seen, the order of magnitude of the pointing errors is 10^{-6} degrees, whereas the requirement was 0.01 degrees. The order of magnitude of the angular velocity errors is 10^{-11} degrees per second, whereas the requirement was $1 \cdot 10^{-4}$ degrees per second. Both of them are well below the requirements. The errors obtained with the dual quaternion approach are exactly the same. No differences can be appreciated comparing the plots.

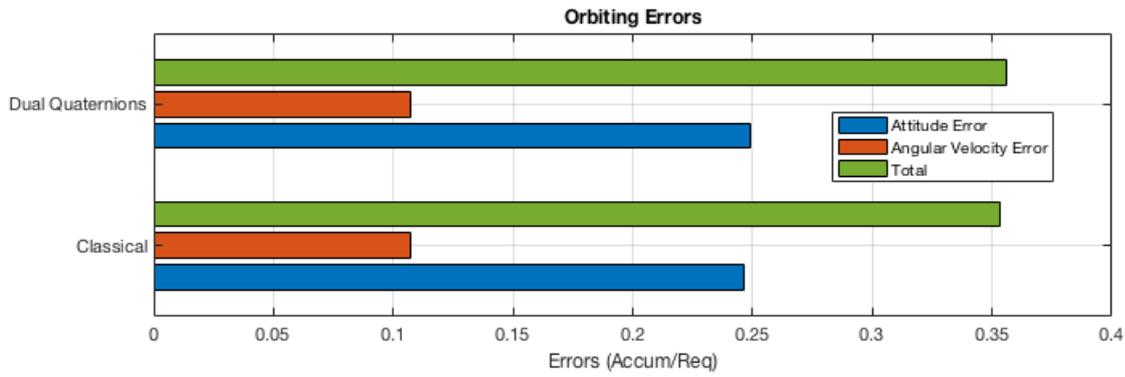


Figure 8.3: Rotational motion errors for the orbiting scenario

Several things are worth mentioning. First, the pointing errors only have discrete values. This is because the pointing error has been computed as two times the arc-cosine of the first component of the attitude quaternion. It represents the angle to be rotated so that the orientation of the spacecraft is the same as the reference one. Since all state variables have been treated as double floating point precision numbers, the arc-cosine function has a maximum precision of 10^{-7} degrees, when close to zero. This is the reason why the values are discrete. Second, the angular velocity errors have some oscillations in the order of magnitude of 10^{-11} , but they are stable. The simulation time has been increased up to ten orbits and these oscillations have remained constant. This also happens in the exact same way for the dual quaternion representation. Third and finally, despite the fact that it cannot be observed in Figure 8.2, the angular velocity has an overshoot of almost $3 \cdot 10^{-9}$ degrees per second in the y axis, before stabilizing. The overshoot for the dual quaternion representation is in the same order of magnitude.

To numerically compare the errors of both methods, each error state component has been integrated over time. In this way, the total accumulated error can be compared. Figure 8.3 shows this comparison in a histogram. The blue bars represent the total accumulated pointing error. It is the summation of the absolute values of the angle to be rotated over the time of simulation. The red bars represent the total accumulated angular velocity errors. It is the summation of the errors in all three body axes over time. The green bars represent the summation of the blue and red bars. It represents the total rotational error of each method. It is important to mention that, all errors have been normalized with their requirements. In this way, it is possible to add pointing errors with angular velocity errors. Therefore, the physical meaning is the number of times the accumulated error exceeds each requirement.

As can be seen, the accumulated errors are almost the same for both methods. To be more precise, the total accumulated rotational error (green bar) for the dual quaternion representation is 0.356, whereas for the classical representation it is 0.354. This difference is insignificant. These results state that both methods have the same errors and behave in very similar ways for this orbiting scenario.

A second aspect to be compared of both methods is the energy spent. It could happen that for achieving the same errors, one of them spends much more energy. Figure 8.4 shows the control torques commands for each methods. As can be seen, they are identical again. This probably explains why the error performance is

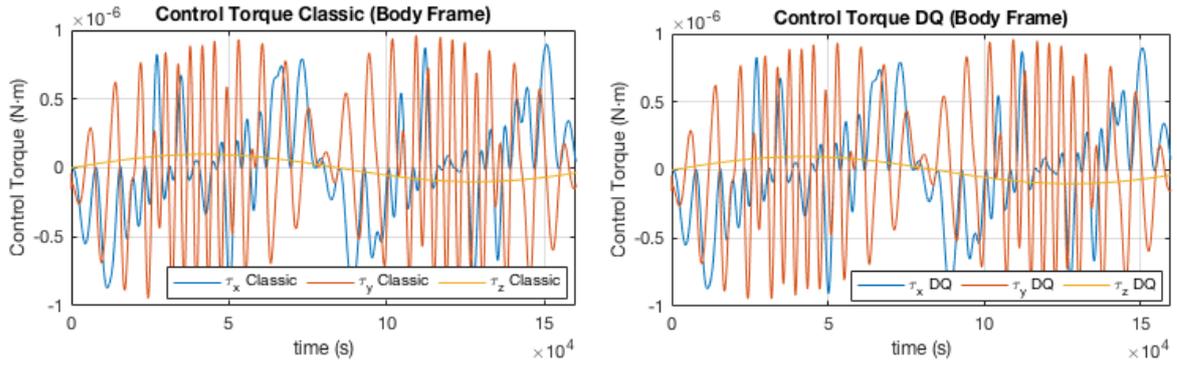


Figure 8.4: Control torque commands for the orbiting scenario (classical and dual quaternion)

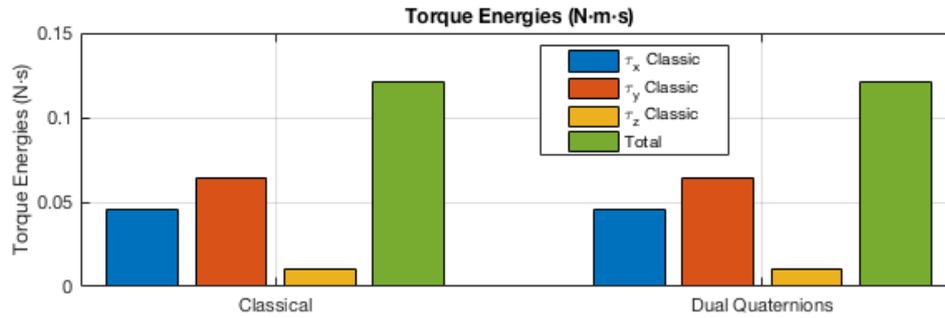


Figure 8.5: Rotational energy spent for the orbiting scenario

also identical. If the control torques are the same, the spacecraft will follow the same trajectory. It is worth mentioning that, the the order of magnitude of the torques is 10^{-6} N·m, which is the order of magnitude of the gravity gradient torques affecting the spacecraft.

As has been carried out with the errors, to numerically compare both representations, the total energy spent has been accounted. This is shown in Figure 8.5. The blue-red-yellow bars represent the integration over time of each torque in the three body axes. As can be seen, the total energy spent by each method is again almost identical. To be more precise, the difference in energy spent is only $4 \cdot 10^{-11}$ N·m·s, whereas the order of magnitude of the values is 10^{-1} N·m·s. This difference is completely insignificant. For this reason, it can be affirmed that both methods used the same amount of energy to achieve the same level of precision.

Finally, after proving that both methods are completely equivalent in terms of errors and energy, the time spent for each method has to be compared. Maybe, to achieve the same performance, one method takes more computational time than the other. And this is clearly the case. After running the same simulation 1000 times, the computational times for each method are presented in Table 8.4. The dual quaternion method takes clearly more time to achieve the same results than the classical method. The hypothesis of a coupled rotational and translational dynamics problem with dual quaternions being faster, has been proven to be false for the orbiting scenario.

To conclude, this section has proved that the dual quaternion representation takes more time to achieve the same level of precision and energy consumption than the classical representation.

Table 8.4: Time to run orbiting scenario

	Classical Method	Dual quaternion method
Time	24.267 ± 0.034 s	28.045 ± 0.039 s

Table 8.5: Initial conditions and settings for hovering simulation

Parameter	Value
Initial position (Inertial frame)	[-288.563 ; -687.929 ; 0] m
Initial velocity (Inertial frame)	[0 ; 0 ; 0] m/s
Initial attitude	[0.1812 ; 0.0973 ; 0.7004 ; 0.6835]
Initial angular velocity (Body frame)	[0 ; 0 ; 0] · 10 ⁻³ rad/s
Integration and control freq.	IF = 2 Hz ; CF = 0.5 Hz
Time span	0 - 1 h
Classic control gains	$K_q = 1065.918, K_r = 862.069, K_\omega = 1106.531, K_v = 1075.862$
Dual quaternion control gains	$K_q = 512.727, K_r = 1.818, K_\omega = 581.818, K_v = 1.818$

8.3.2. HOVERING

The second scenario defined in Section 2.3 is the hovering scenario. It consists in hovering at 500 m from the surface of Bennu. The initial conditions and all the parameters for the simulations are summarized in Table 8.5. The selected gains are the ones found in Section 5.3.

As happened with the orbiting simulations, the results from the hovering scenario of the dual quaternion method and the classical one are almost identical again. For this reason and for the sake of readability, this time, only the dual quaternion results will be shown in this document; and the tiny differences will be explained.

As a first result, both control approaches succeeded in maintaining the spacecraft on the hovering spot with the right orientation, within the requirements defined in Sections 2.2. Figure 8.6 shows the errors between the dual quaternion solution and the reference one. The main difference between the hovering scenario and the orbiting one is that the former requires both translational and rotational control. This is the reason why four different sub-plots are shown in this figure.

The first sub-figure on the upper left side corresponds to the position errors. As can be seen, the orders of magnitude of the errors are of 10⁻¹¹ m, whereas the control requirement is 1 cm. These errors are way below the requirement. In this case, a first transitional region can be observed in the very beginning (specially in the y direction). After that, the error oscillates but does not increase over time. To prove that, the hover simulation has been prolonged for ten hours and the results showed that the errors do not grow. This errors are completely dependent on the gains selected during the optimization process. Similar results are obtained for the velocities. The order of magnitude is the same but the requirement is 0.01 cm/s, therefore, the requirement is completely met.

The sub-figure on the bottom left part of Figure 8.6 corresponds the pointing errors. As can be observed,

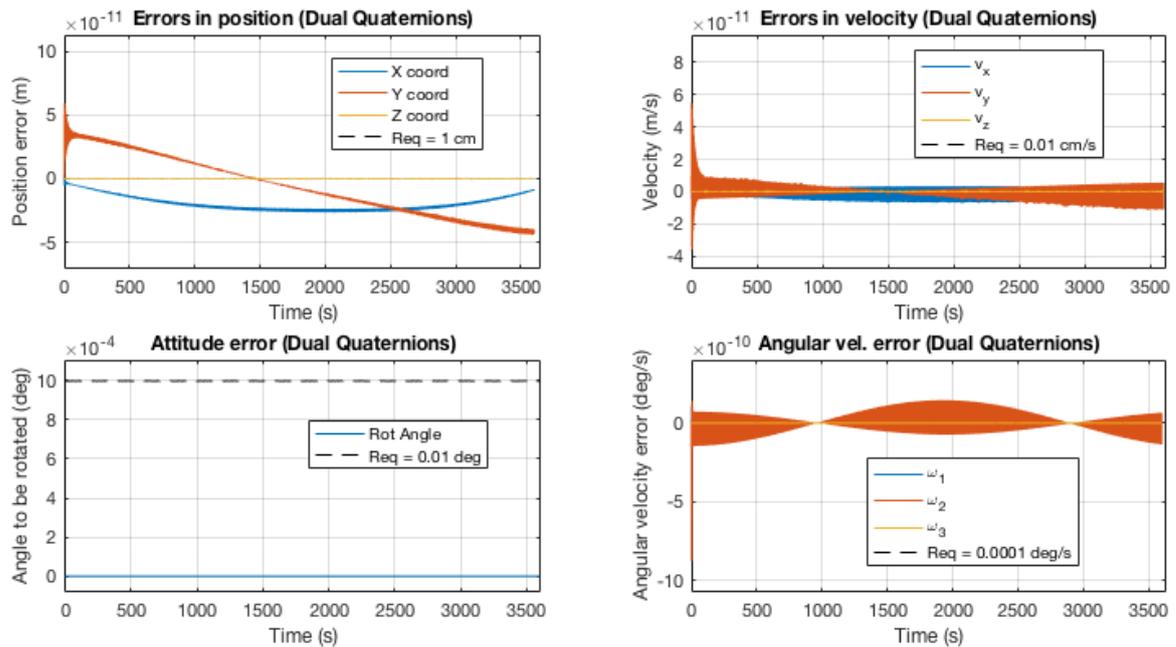


Figure 8.6: State variable errors with respect to reference trajectory with dual quaternion approach

they are exactly zero. This of course is not true, but because of the working precision, the matlab results are exactly zero. This means that the error in the first component of the attitude quaternion is below 10^{-16} . Roughly speaking: tinny small attitude errors. Finally, the last sub-figure on the bottom right, corresponds to the angular velocity errors. The order of magnitude of the errors is 10^{-10} degrees per second, whereas the requirement is $0.0001^\circ/\text{s}$. A clear overshoot of $8 \cdot 10^{-10}$ can be observed in the y body axis, before stabilization. The errors obtained with the classical method are almost identical than this ones shown in Figure 8.6. No differences can be appreciated comparing both results.

Again, as it has been carried out for the orbiting scenario, to numerically compare the errors of both methods, each error state component has been integrated over time. In this way, the total accumulated error can be compared. Figure 8.7 shows this comparison in a histogram. In this case, the blue bars represent the total accumulated position errors; the red bars, the velocity errors; the yellow bars, the pointing errors and the purple bars the angular velocity errors. The green bars still represent the total summation of all of them. All bars have been normalized with their specific requirement defined in Section 2.4. In this way, it is possible to add pointing errors with position errors, as has already been explained before.

As can be seen, the accumulated errors are again almost the same for both methods. To be more precise, the total accumulated error (green bar) for the dual quaternion representation is 0.03219, whereas for the classical representation is 0.03221. This difference is again insignificant. These results state that both methods have the same errors and behave in very similar ways for this hovering scenario.

A second aspect to be compared of both methods is the energy spent. Figure 8.8 shows the control forces and torques commands for each methods. This is different from the orbiting scenario, because now, control forces are also taken into account for the translational motion. As can be seen, they are almost identical

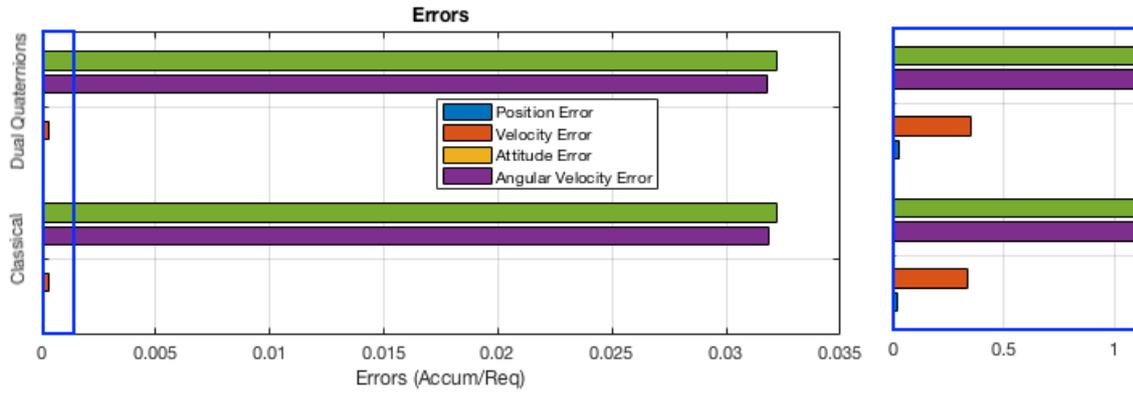


Figure 8.7: Translational and rotational motion errors for the hovering scenario

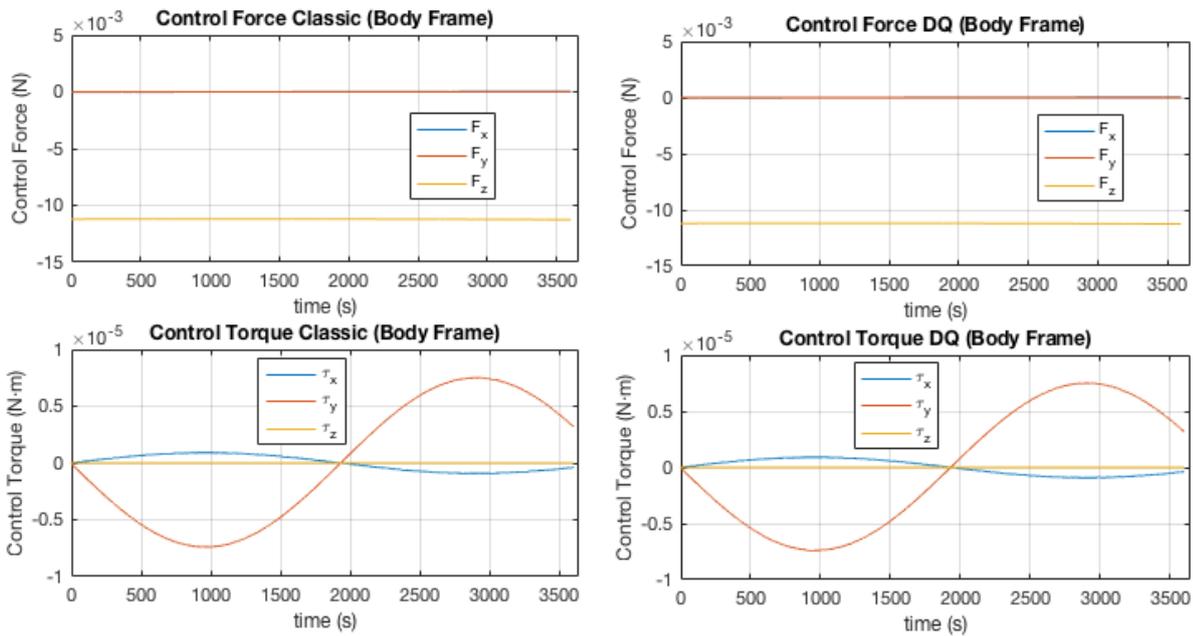


Figure 8.8: Control force and torque commands for the hovering scenario (classical and dual quaternion)

again. As it will be explained later on the results, they will only be different when correcting for perturbations. If both controllers are optimized in the same way, their performance turns out to be identical for the nominal scenarios. It is worth mentioning that the order of magnitude of the control torques is 10^{-5} N·m in this case, whereas the order of magnitude of the forces is 10^{-2} N.

To numerically compare both methods, the total energy spent has been accounted. This is shown in Figure 8.9. On the left part, the total accumulated forces are shown for both the classical and the dual quaternion representations. As can be seen, both methods consume the same force and it corresponds to the one on the z body axis. The reason for this is because the spacecraft have to point to the asteroid at all times with the z body axis. Therefore, the control force has to nullify the gravitation pull in this same direction. On the right side, the energy consumed for the rotational motion can be observed. Again, both methods consume almost the same amount energy. To be more precise, the difference in energy spent for the rotational motion

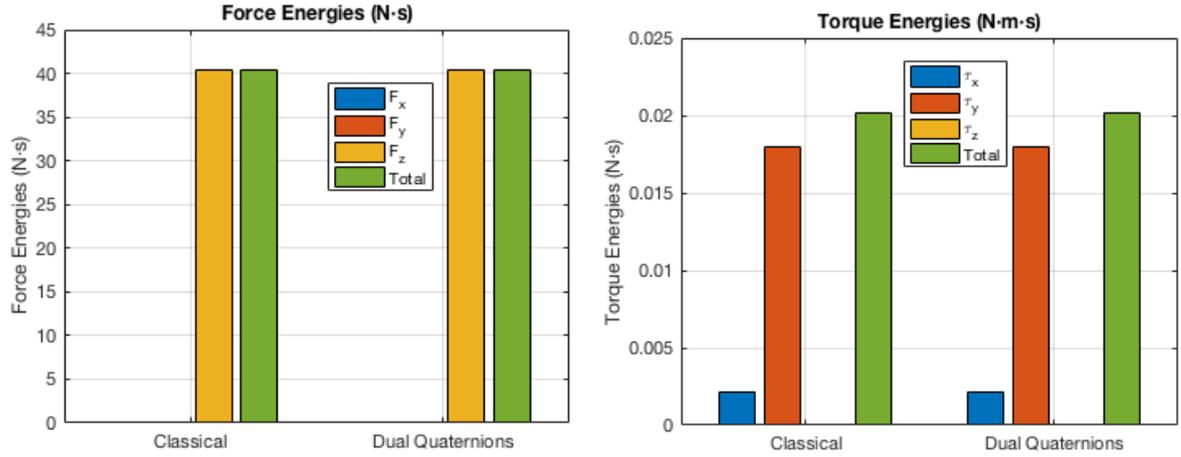


Figure 8.9: Rotational energy spent for the hovering scenario

Table 8.6: Time to run hovering scenario

	Classical Method	Dual quaternion method
Time	11.090 ± 0.021 s	12.516 ± 0.019 s

is only 10^{-11} N·m·s, whereas the magnitude of the values are of 10^{-2} N·m·s. This difference is completely insignificant. For the translational motion, this difference is of the order of magnitude of 10^{-9} N, whereas the magnitude of the values are of 10^1 N. With these results, one can affirm that both methods use the same amount of energy to achieve the same level of precision.

Finally, after proving that both methods are completely equivalent in terms of errors and energy, the time spent for each method has to be compared. After running the same simulation 1000 times, the computational times for each method are presented in Table 8.6. Again, the dual quaternion method takes clearly more time to achieve the same results than the classical method.

The same conclusion can be drawn from this hovering scenario than for the orbiting one: the dual quaternion representation takes more time to achieve the same level of precision and energy consumption than the classical representation.

8.3.3. LANDING

The last and third scenario defined in Section 2.3 is the landing scenario. It consists in a landing on the surface of Bennu following a certain given reference trajectory. The initial conditions and all the parameters for the simulations are summarized in Table 8.7. The selected gains are the ones found in Section 5.3.

Again, as happened in the previous simulations, the results from the landing scenario of the dual quaternion method and the classical one, are almost identical. For this reason and for the sake of readability, this time, only the classical results will be shown in this document. The tiny differences between methods will be explained.

As a first result, both control approaches succeeded in steering the spacecraft through the reference land-

Table 8.7: Initial conditions and settings for landing simulation

Parameter	Value
Initial position (ACI frame)	[345.46 ; 0 ; 0] m
Initial velocity (ACI frame)	[-0.1 ; -0.1403 ; 0] m/s
Initial attitude	[0.4743 ; 0.5244 ; -0.4743 ; -0.5244]
Initial angular velocity (Body frame)	[0.0932 ; 0.6336 ; -0.0191] · 10 ⁻³ rad/s
Integration and control freq.	IF = 4 Hz ; CF = 1 Hz
Time span	0 - 600 s
Classic control gains	$K_q = 3271.429, K_r = 3974.000, K_\omega = 1978.571, K_v = 2295.000$
Dual quaternion control gains	$K_q = 950.000, K_r = 0.267, K_\omega = 1050.000, K_v = 1.213$

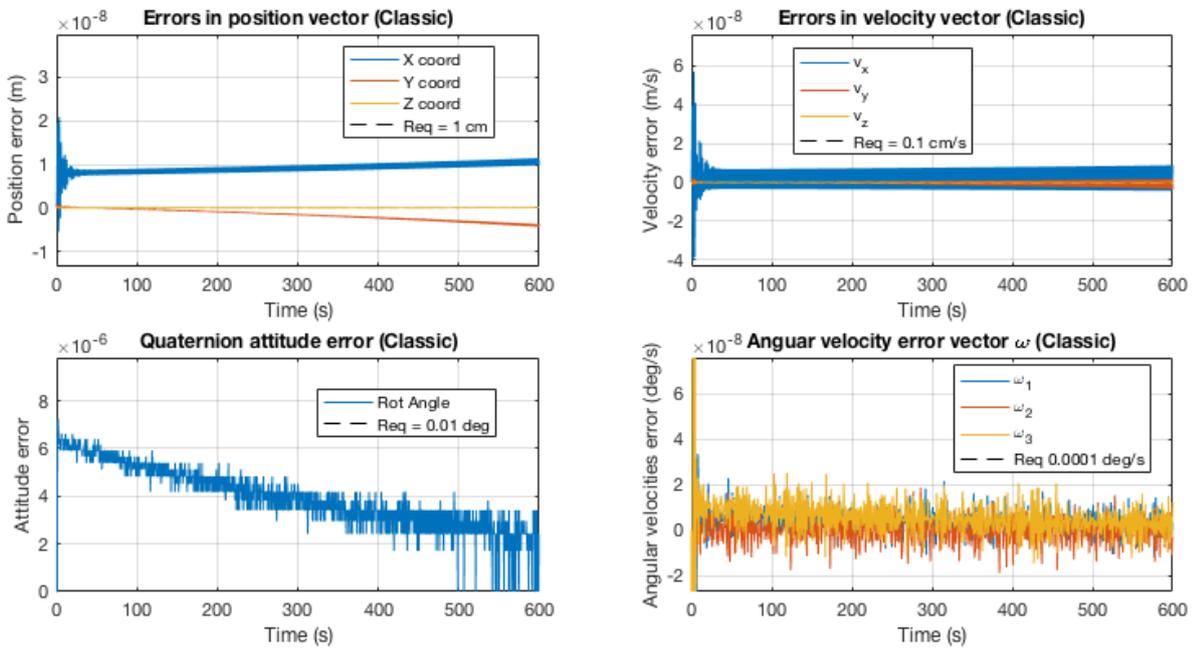


Figure 8.10: State variable errors with respect to reference trajectory with classical approach

ing trajectory, within the requirements defined in Sections 2.2. Figure 8.10 shows the errors between the classical solution and the reference one.

The first sub-figure on the top left corresponds to the position errors. As can be seen, the order of magnitude of the errors is 10^{-8} m, whereas the control requirement is 1 cm. Despite the fact of growing with time, these errors are way smaller than the position requirement. These errors are totally dependent on the control gains selected in Section 5.3, which have been chosen taken into account energy consumption and accumulated errors. For this nominal trajectory, the selected gains achieve the goal with the minimum energy consumption. Similar results are obtained for the landing velocities. The order of magnitude is the same but the requirement is 0.01 cm/s, therefore, the requirement is completely met.

The sub-figure on the bottom left part of Figure 8.10 corresponds the pointing errors. As can be observed,

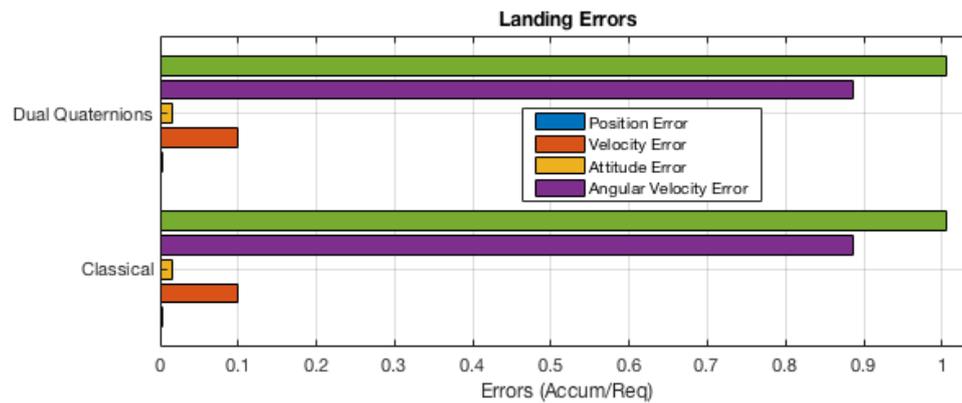


Figure 8.11: Translational and rotational motion errors for the landing scenario

the pointing error reaches its maximum at the beginning, and then slowly converges to almost zero during the landing trajectory. The order of magnitude of the pointing errors is 10^{-6} degrees, far below the attitude requirement. Finally, the last sub-figure on the bottom right, corresponds to the angular velocity errors. The order of magnitude of the errors is 10^{-8} degrees per second, whereas the requirement is $0.0001^{\circ}/s$. A clear overshoot of $8 \cdot 10^{-6}$ degrees per second can be observed in the z body axis, before stabilization. The errors obtained with the dual quaternion method are almost identical than these ones shown in Figure 8.10. No differences could be appreciated comparing both results.

Again, as it has been carried out for the previous scenarios, to numerically compare the errors of both methods, each error state component has been integrated over time. Figure 8.11 shows this comparison in a histogram. All bars are normalized with the specific requirements defined in Section 2.4.

As can be seen, the accumulated errors are again almost the same for both methods. To be more precise, the total accumulated error (green bar) for the dual quaternion representation is 1.0052, whereas for the classical representation is 1.0049. This difference is again insignificant. It might be important to notice that the longer bar is the purple one, corresponding to the angular velocity errors. These results state that both methods have almost the same errors and behave in very similar ways for this landing scenario.

The second aspect to be compared of both methods is the energy spent. Figure 8.11 shows control forces and torques commands for each methods. As can be seen, they are almost identical again. It is worth mentioning that the order of magnitude of the control torques is 10^{-4} N·m in this case, whereas the order of magnitude of the forces is 10^{-2} N.

To numerically compare both methods, the total energy spent has been accounted. This is shown in Figure 8.13. On the left part, the total accumulated forces are shown for both the classical and the dual quaternion representations. As can be seen, both methods consume the same force. On the right side, the energy consumed for the rotational motion can be observed. Again, both methods consume almost the same energy. To be more precise, the difference in energy spent for the rotational motion is only $1 \cdot 10^{-7}$ N·m·s, whereas the magnitude of the values are of 10^{-2} N·m·s. This difference is completely insignificant. With these results, one can state that both methods use the same amount of energy to achieve the same level of precision.

Finally, after proving that both methods are completely equivalent in terms of errors and energy, the time

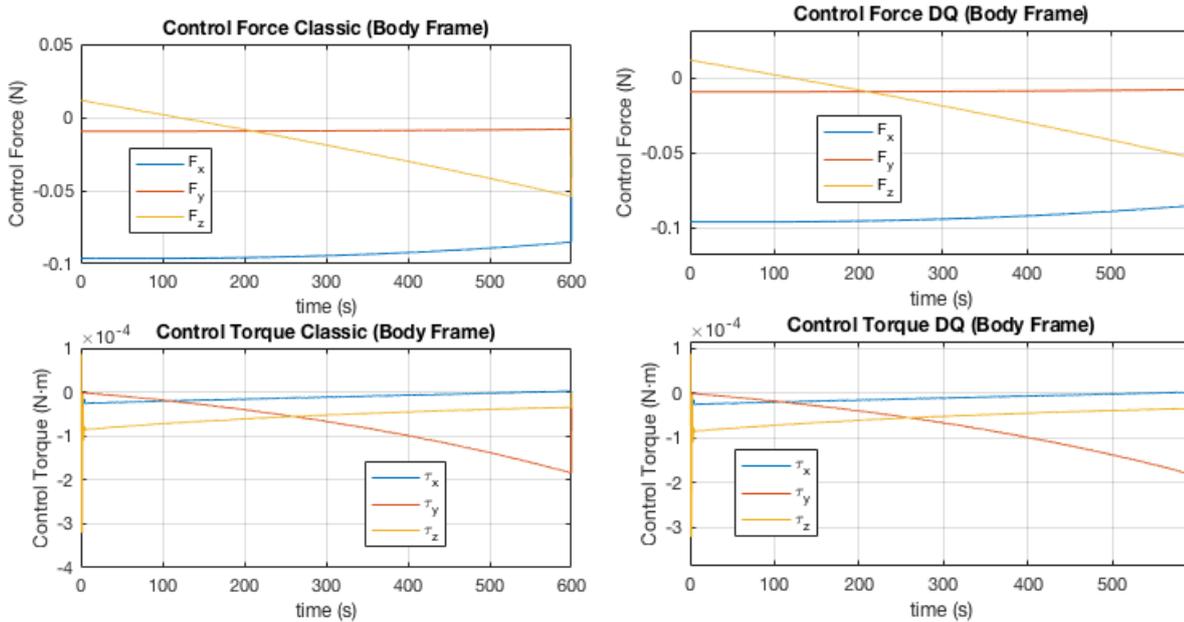


Figure 8.12: Control force and torque commands for the landing scenario (classical and dual quaternion)

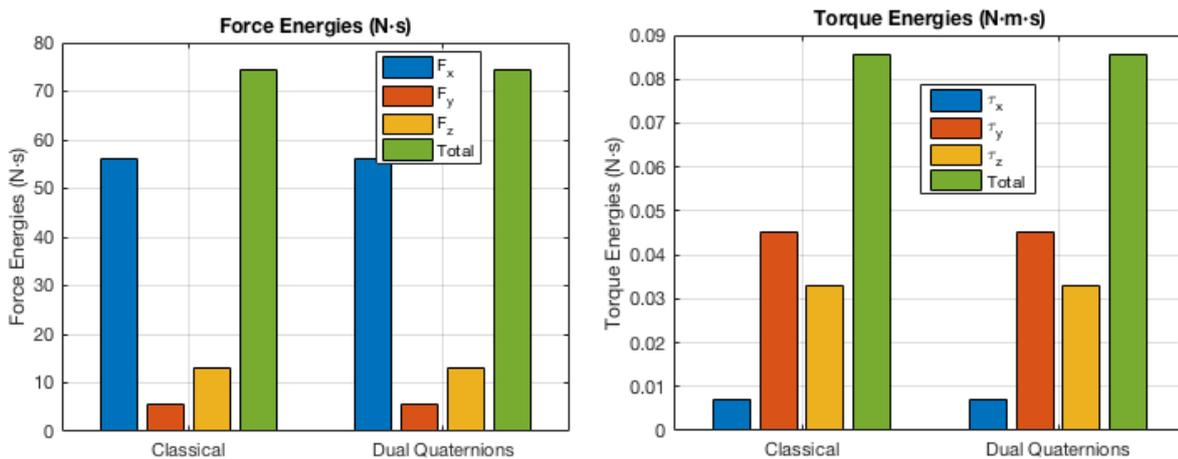


Figure 8.13: Rotational energy spent for the landing scenario

Table 8.8: Time to run landing scenario

	Classical Method	Dual quaternion method
Time	3.715 ± 0.008 s	4.152 ± 0.011 s

spent for each method has to be compared. After running the same simulation 1000 times, the computational times for each method are presented in Table 8.8. Again, the dual quaternion method takes clearly more time to achieve the same results than the classical method.

To conclude, in all three different asteroid close-proximity operations, the dual quaternion representation takes always more time to achieve the same level of accuracy and energy consumption than the classical

Table 8.9: Range of operational control frequencies meeting mission requirements

	Classical Method	Dual quaternion method
Orbiting scenario	< 0.5 Hz - 0.465 Hz >	< 0.5 Hz - 0.457 Hz >
Hovering scenario	< 0.5 Hz - 0.459 Hz >	< 0.5 Hz - 0.490 Hz >
Landing scenario	< 1 Hz - 0.834 Hz >	< 1 Hz - 0.541 Hz >

representation.

8.4. FREQUENCY ANALYSIS

In an attempt to further compare both controllers performances, a frequency analysis has been carried out. So far, it has been proven that both methods achieve the same accuracy and energy consumption with a certain given control frequency. However, what happens if this control frequency has to be reduced? Will both controllers still behave in the same way? The aim of this section is to answer this question, and determine which one pops up earlier.

To do so, the control frequency has been reduced little by little. The initial conditions have been kept the same, with no initial errors. In this way, it has been possible to determine the lowest frequency at which, each controller, still can meet the requirements for all three scenarios. The integration frequency and the time span of each simulation have been kept the same as well. The same applies for the gains. Only the control frequency has been reduced.

Table 8.9 shows the results of this analysis for each asteroid close-proximity operation. The results are given with an operation control frequency range. It represents the range of control frequencies at which, each controller, is still able to meet the mission requirements. As can be seen, for the orbiting and landing scenarios, the dual quaternion representation has a wider range of frequencies. This means that it can meet the mission requirements with lower control frequencies.

Results from the landing scenario, for instance, show that the control frequency can be reduced up to 0.834 Hz for the classical representation, and up to 0.541 Hz for the dual quaternion representation. For the latter case, this represents a reduction of almost the 50% of the nominal frequency. Due to nowadays flight proven GNC hardware and software frequencies limitations, this result can be very beneficial and save a lot of money. Results from the orbiting scenario only show a small advantage of the dual quaternion representation over the classical one. The possible control frequency reduction is a 7% for the classical method and a 8.6% for the dual quaternion one. This is only a difference of 1.6%.

Finally, results from the hovering scenario show the opposite. The dual quaternion representation stops meeting the mission requirements with higher frequencies than the classical representation. The reason for this is unknown. This is clearly a recommended future aspect to study. Several hypothesis have been formulated and proven wrong. For example, it was thought to be because of a local minimum of the optimization cost function close to the non-converging region. However, this was not the case. Bearing these results in

mind, it cannot be concluded that dual quaternions are less sensitive to control frequency reductions. Results are inconclusive and this can only be concluded for the specific case of orbiting and landing.

8.5. STABILITY ANALYSIS

Another important aspect to be compared is the stability of each controller. Previous results showed that both representations have the same performance in terms of accuracy and energy consumption, for the nominal trajectory. However, how does each controller deal with perturbations? What happens if the spacecraft does not start on the reference trajectory? Chapter 5, together with the stability theory summarized in Appendix A, proved that both controllers can reach almost global Lyapunov stability. However, is there a controller more stable than the other? How does each controller steer back the satellite to the reference trajectory? What is the maximum perturbation each controller can handle?

To answer all these questions, several tests have been carried out for the three asteroid close-proximity scenarios. The first one consists in determining the maximum initial perturbation that each controller can handle. However, a perturbation of 1 m in the initial position has not the same impact as a perturbation of 1 m/s in the initial velocity. To solve this dimensional problem, all perturbations have been normalized with their maximum allowed deviations. In this way, the perturbations have been expressed as the number of times exceeding the maximum allowed control deviations; becoming therefore, comparable.

The test starts with an initial perturbation of 1 time the maximum allowed deviation in each state variable. For instance, if the requirement for the position is 1 cm, the initial error for the position has been set to be 1 cm. However, it is not the same to have 1 cm of error in one direction than in another direction. For this reason, 50 different sets of perturbations of 1 cm have been generated with totally random different directions. It is important to mention however, that these initial conditions have been the same for the classical and dual quaternion representations. The goal is to determine if each controller is able to handle initial perturbations of a number of times the maximum allowed deviation in 50 random different directions. It might be worth mentioning that the control and integration frequency has not been modified from the ones of the nominal scenarios. The gains are also the same. The only parameters that have changed are the initial conditions. If a controller is able to handle a perturbation of 3 times the maximum allowed deviation for each of the state variables, in 50 different directions, it is then considered to handle errors of this order of magnitude in all possible directions. This is of course an assumption but it works for comparing each representation. The magnitude of the perturbations has been increased until a controller is not able to handle the perturbations anymore.

Results from this test are shown in Table 8.10. They represent the order of magnitude of the initial perturbations that each controller can handle. These results show a very interesting discovery. The dual quaternion representation can handle much larger initial deviations than the classical representation in all three different scenarios.

As can be seen, for the orbiting scenario, the dual quaternion representation can meet the mission requirements with initial perturbations of 255 times the maximum allowed deviation. This represents 227 more

Table 8.10: Initial conditions deviations that still meet the requirements for each scenario

	Classical Method	Dual quaternion method
Orbiting scenario	28 times	255 times
Hovering scenario	4 times	202 times
Landing scenario	16 times	39 times

times than the classical representation. The same can be said for the hovering scenario, with 198 times more. For the landing scenario, the difference is a little bit smaller with only 23 times more. These results are very important because they prove that dual quaternions, despite the fact of taking more time to achieve the same level of accuracy and energy consumption, they can handle much larger perturbations than the classical representation. For real asteroids missions, this is a very important advantage. As has been explained in Chapter 2, the dynamical environment around asteroids is not always properly modeled and there are always perturbations. The results found in this test show that dual quaternions are more suited for these perturbations.

Dual quaternions might be able to deal with larger perturbations, however, given a perturbation, which method is more efficient in terms of energy? Which control approach takes more time? Which controller has a shorter settling time and brings the spacecraft back to the reference trajectory with less accumulated errors? To answer all these questions, a second test has been carried out. It consists in testing both controllers for a given set of initial conditions with certain deviations. For each set of initial conditions, four different aspects will be analyzed: the energy spent; the time it takes to run the simulation, the settling time and the total accumulated error.

It is important to mention, that the initial conditions are the same for both controllers. In the range of stability perturbations computed in the first test, 800 random initial conditions with deviations have been generated for each scenario. Results are presented here as histograms. The bars of the histograms represent the number of times each controller has a results in a specific range of values. For instance, how many of the 800 simulations took between 3 and 4 seconds to run. In this way, it is possible to compare both approaches and drew some conclusions.

8.5.1. ENERGY

The first aspect to be compared is the energy spent by each control approach. Figures 8.14 to 8.16 show the results. Figure 8.14 shows the histograms for the orbiting scenario. As can be observed, the dual quaternion controller uses much less energy in general than the classical controller. To take some statistical results, a normal distribution has been fitted to both histograms. The mean average energy spent by the classical controller is 604.988, while it is only 64.889 for the dual quaternion one. These numbers do not have units because they have been normalized with the maximum force and torque available by the spacecraft actuators. They represent the summation of all forces and torques over time, in all three body axes, normalized with these maximum available limits. Without any doubt, the classical controller consumes more energy.

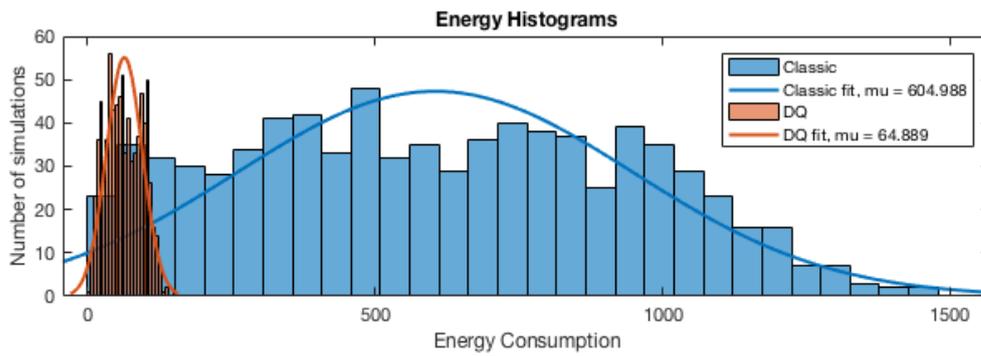


Figure 8.14: Energy spent by each method given a set of initial deviations (orbiting scenario)

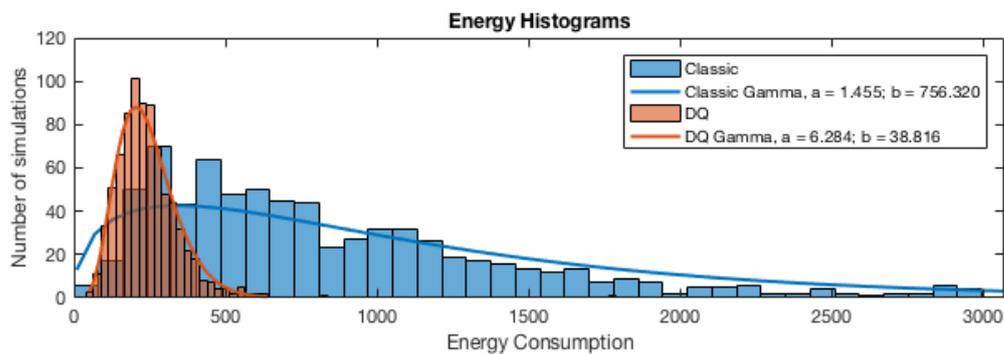


Figure 8.15: Energy spent by each method given a set of initial deviations (hovering scenario)

Figure 8.15 shows the same results but for the hovering scenario. As can be seen, the classical approach also consumes much more energy than the dual quaternion one. However, in this case, it was not possible to fit a normal distribution because the energy spent by each controller follows a gamma distribution instead. The gamma distribution is a two-parameter family of distributions used to model sums of exponentially distributed random variables. a is the shape parameter and b is the scale parameters. The estimated value is their multiplication. Gamma probability functions are used, for example, to model waiting times in airports or multi-path fading of signal power in wireless communications. After fitting the curves, the mean value for the classical controller is 1100.446, whereas it is 243.920 for the dual quaternion one.

Finally, Figure 8.16 shows the results for the landing scenario. In this case, it was possible to fit a normal distribution for each controller again. The results are the same as the previous ones: the classical representation consumes 3.2 more energy than the dual quaternion representation.

From these results, a very important conclusion can be drawn: dual quaternions can deal with initial perturbations in a more efficient way than the classical controllers in terms of energy consumption. This is also an outstanding advantage of the dual quaternions over the classical representation because efficiency in space means less fuel. And less fuel means less money.

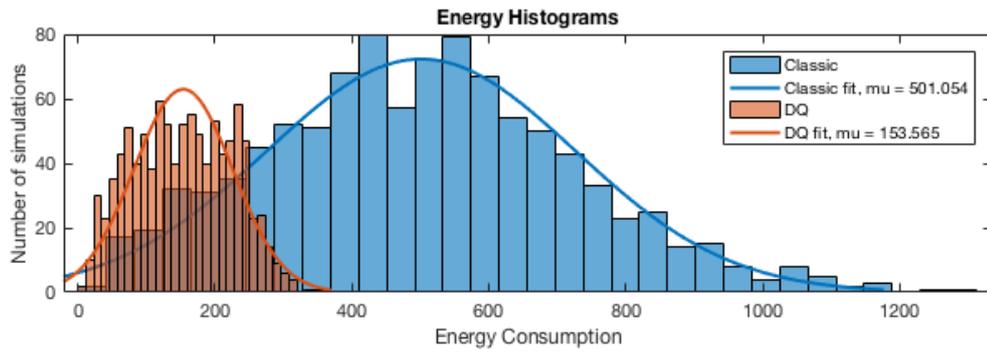


Figure 8.16: Energy spent by each method given a set initial deviations (landing scenario)

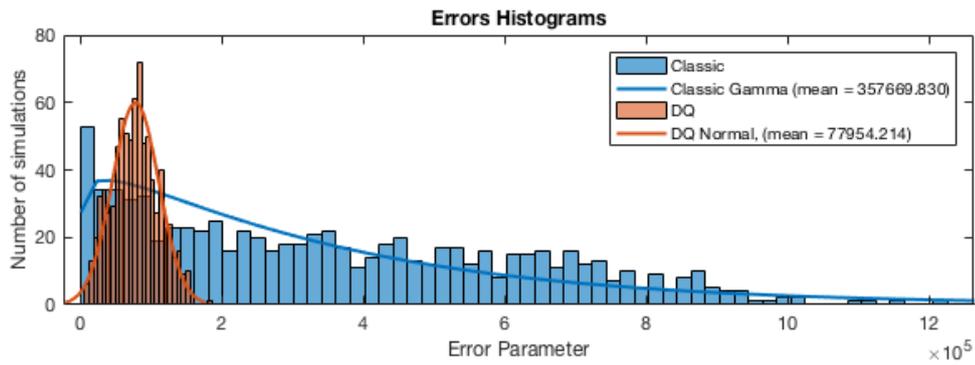


Figure 8.17: Accumulated errors by each method given a set initial deviations (orbiting scenario)

8.5.2. ERRORS

The second aspect to be compared is the total accumulated error of each controller. Figures 8.17 to 8.19 show the results for each of the three scenarios. It is important to notice that the accumulated error has been computed in the same way as previous tests. Since errors in position are not the same as errors in velocity, all errors have been normalized with their maximum allowed value. In this way, the total error plotted in these figures represent the summation of all the state variables normalized errors. Again, the results are shown in the form of histograms. Figure 8.17 shows the total accumulated errors for the orbiting scenario. As can be seen, the dual quaternion representation has much less errors than the classical one. In this case, the mean value of the total accumulated errors for the dual quaternion controller is 77954.214, whereas it is 357669.830 for the classical one. It is important to notice that, a normal distribution has been fitted for the dual quaternion controller histogram, whereas a gamma distribution has been fitted for the classical one.

Figure 8.18 shows the same results but for the hovering scenario. As can be seen, the classical approach also has much more accumulated errors than the dual quaternion one. In this case, the differences were so big that they had to be plotted in separate histograms to appreciate the information. Again, these histograms follow a gamma distribution. The mean value of the total accumulated error for the classical controller is $3.123494 \cdot 10^6$ and $1.65085 \cdot 10^5$ for the dual quaternion one.

Finally, Figure 8.19 shows the error results for the landing scenario. In this case, it was possible to fit a normal distribution for each controller again. The mean value of accumulated errors for the classical control

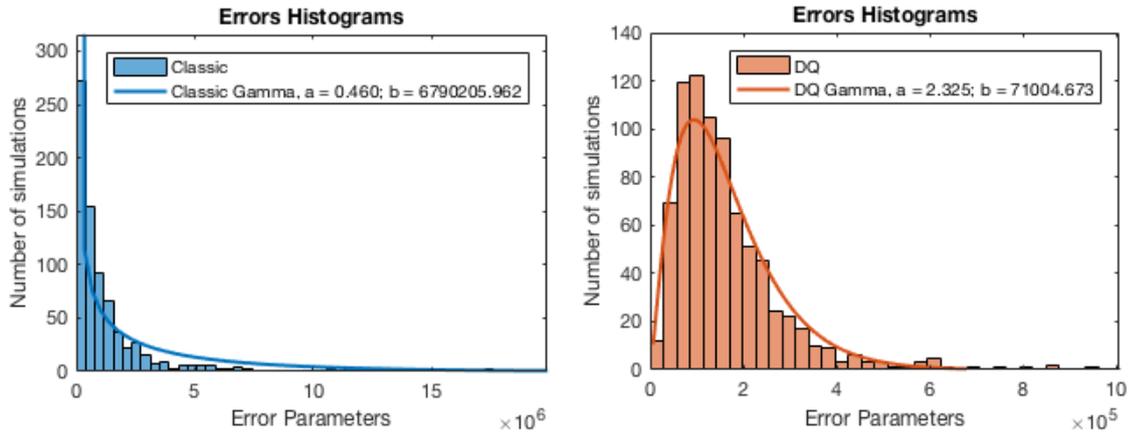


Figure 8.18: Accumulated errors by each method given a set initial deviations (hovering scenario)

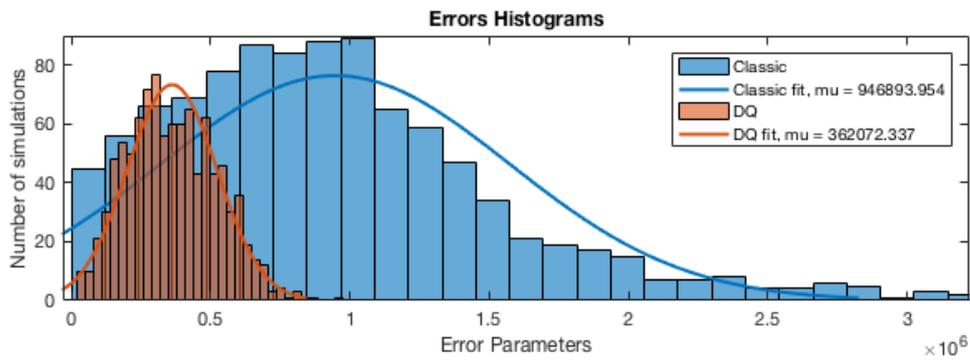


Figure 8.19: Accumulated errors by each method given a set initial deviations (landing scenario)

approach is $9.468 \cdot 10^5$, whereas it is only $3.620 \cdot 10^5$ for the dual quaternion approach.

Bearing these results in mind, an important conclusion can be drawn. In all simulations, despite the fact of having initial errors, both controllers are always available to steer the spacecraft back to the reference trajectory. After reaching stability, as it has been proven in Section 8.3, both controllers behave equally and have the same errors. For this reason, if the dual quaternion control approach is showing much less errors during these simulations, it is because it achieves stability much faster than the classical control approach. And for this reason, it can be stated that, the dual quaternion approach has a much shorter settling time than the classical one. This is a very important advantage to take into account when selected a control approach in real missions. To statistically prove this conclusion, the next subsection shows the settling times of each scenario.

8.5.3. SETTLING TIME

The third aspect to be compared is the settling time. It is important to mention that it has been computed as the time each controller takes to bring the state variables within the mission requirements, after correcting for a perturbation. Results are presented again as histograms, and shown in Figures 8.20 to 8.22. Figure 8.20 shows the settling time for each control approach in the orbiting scenario. As can be seen, both controllers

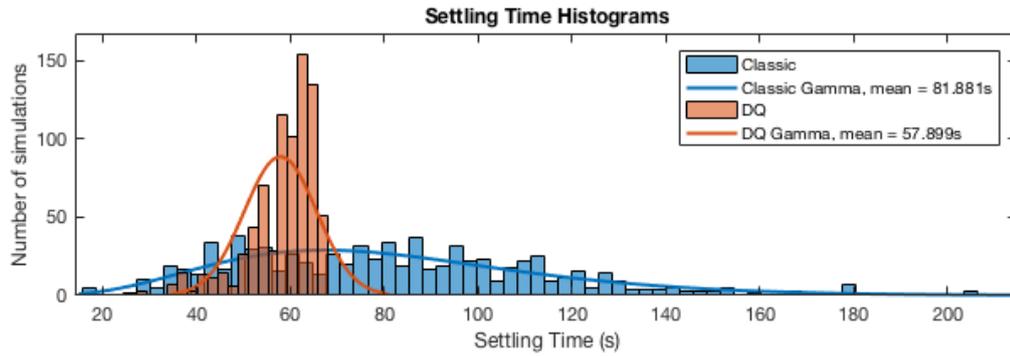


Figure 8.20: Settling time for each method (orbiting scenario)

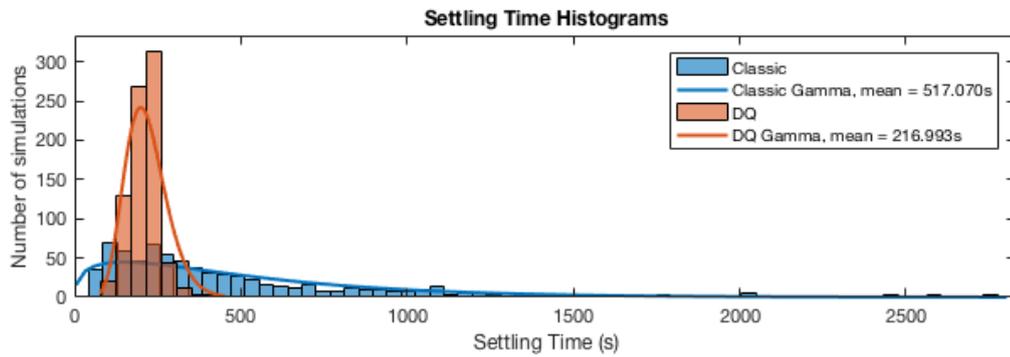


Figure 8.21: Settling time for each method (hovering scenario)

manage to stabilize the initial perturbations after less than 250 s. To be more precise, the average settling time for the dual quaternion controller is 57.9 s, whereas it is 81.9 s in the classical control. Clearly, the controller based on dual quaternions takes less time to bring the different state variables back within the mission requirements.

Figure 8.21 shows the histogram results of settling times for the hovering scenario. In this case, the average settling time for the classical approach is 517 s, whereas it is only 216.9 s for the dual quaternion controller. A gamma distribution have been fitted to both histograms. Again, the dual quaternion approach proves to be much faster in recovering from initial perturbations, specifically, 58.03% faster. It is also worth mentioning that dual quaternions manage to stabilize all perturbations before 500 s; whereas the classical approach can take up to more than 2500 s in some cases.

Finally, Figure 8.22 shows the settling time results for the landing scenario. As can be seen, the average settling time for the classical controller is 203.7 s, whereas it is only 15 s for the dual quaternion representation, which represents a reduction of 92.6%. It is important to notice that the dual quaternion controller manages to stabilize all initial perturbations before the first 25 s. However, the classical one takes longer times and follows a gamma distribution, rather than a normal one.

These histogram results prove what have been stated in the previous subsection: controllers based on dual quaternions have a shorter settling time when correcting for initial perturbations.

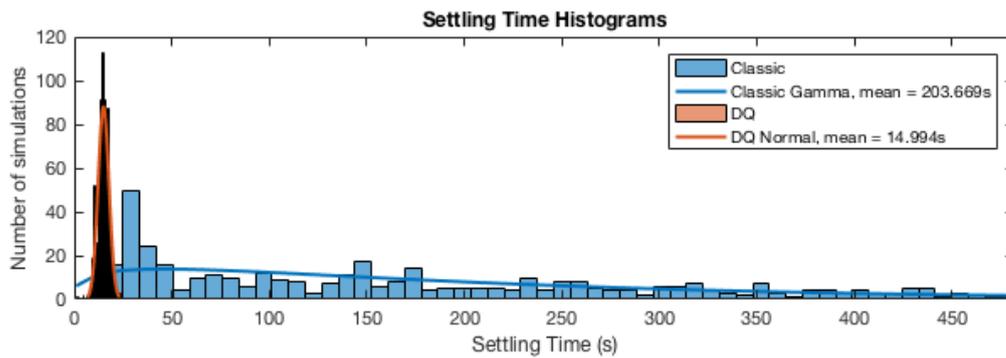


Figure 8.22: Settling time for each method (landing scenario)

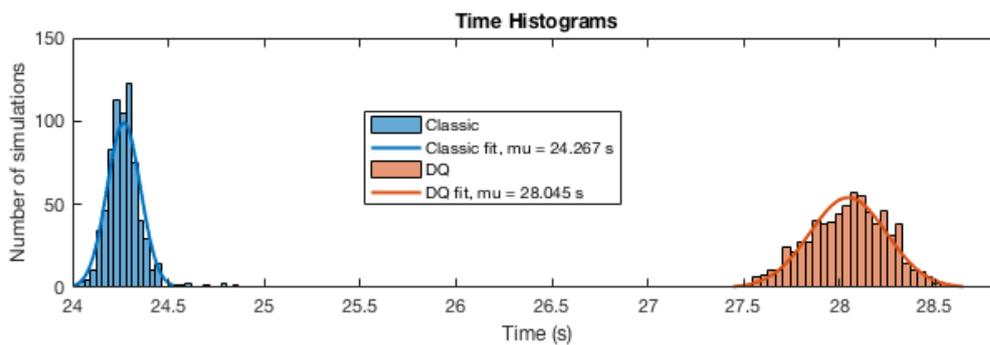


Figure 8.23: Computational time spent for each method (orbiting scenario)

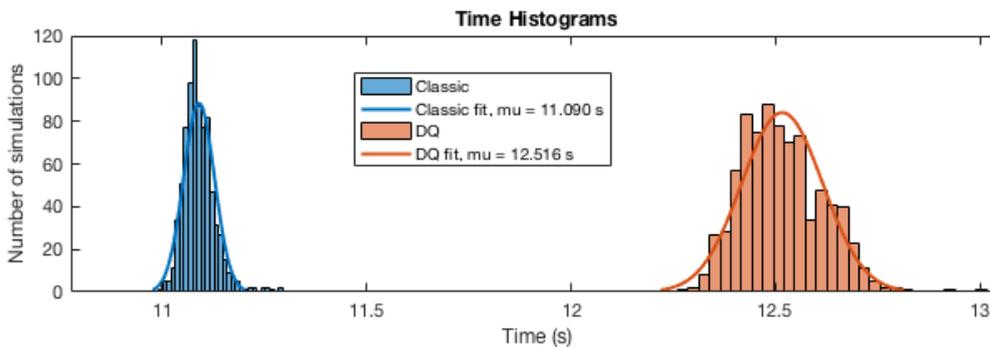


Figure 8.24: Computational time spent for each method (hovering scenario)

8.5.4. COMPUTATIONAL TIME

Finally, the last aspect to be compared is the computational time each controller is taking to run the simulation. Figures 8.23 to 8.25 show the time results. As has already been proven before, the dual quaternion approach takes more time to compute the control commands and integrate the solution. With no surprise, the results are the same as the ones obtained in Section 8.3, because the computational time depends on the number of operations to be carried out and not the efficiency of the control approach.

Several conclusion have been drawn in this section. The first one is that, dual quaternions can handle larger perturbations than the classical approach and still reach the mission requirements. Second, dual quaternions spend much less energy to correct perturbations. Third and final, dual quaternions have a set-

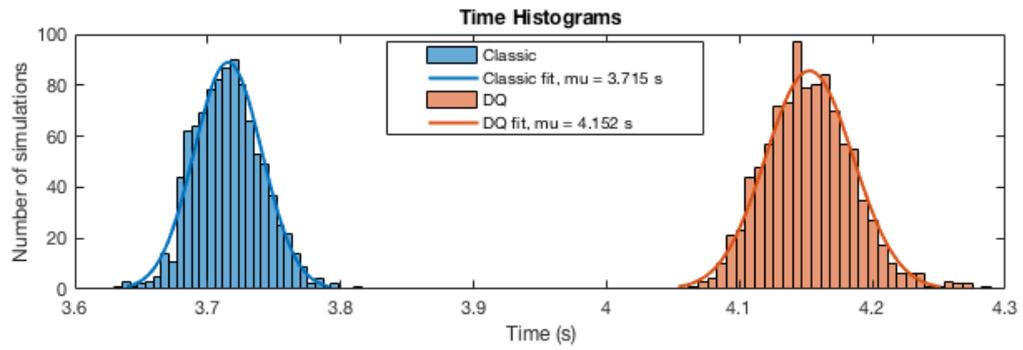


Figure 8.25: Computational time spent for each method (landing scenario)

ting time much shorter than the classical representation and thus, accumulate less error over the simulations. All of these advantages come with the price of taking more computational time.

9

CONCLUSIONS AND RECOMMENDATIONS

This chapter closes the cycle started in the introduction by providing all the conclusions drawn during the development of the thesis work, as well as the results obtained. It also answers the research questions and includes a set of recommendations for future work in the topic.

9.1. CONCLUSIONS

The first conclusion of this work is that dual quaternions have been successfully compared to classical methods for a full six **DOF** controller problem in asteroid close proximity operations. Three different scenarios have been successfully simulated to provide sufficient and valuable comparisons between the two methods to answer the research questions.

The research objective formulated in Chapter 1 was:

- *What advantages and disadvantages can a six **DOF** controller based on dual quaternions bring over classical methods regarding asteroid close proximity operations?*

The answer to this question can be summarized in the following advantages and disadvantages:

- Dual quaternions naturally couple the rotational and translational motion. They allow the integration of both motions to be carried out at the same time.
- Dual quaternions are a mathematical formulation that allows to represent the six **DOF** of a spacecraft in a very compact form.
- Only two equations of motions are sufficient to represent the full six **DOF** problem with dual quaternions. Four are necessary in the classical approach.

- Only one Lyapunov function is necessary to prove stability of the non-linear system with dual quaternions. The classical method requires two steps to prove its stability.
- Dual quaternions controllers can be designed following the same theory and properties of attitude-only controllers written with quaternions.
- Dual quaternion controllers take more time to optimize following the methodology described in this master thesis (Section 5.3).
- Dual quaternions take more time to integrate the equations of motion. The total number of basic operations to be computed is almost 4 times larger than for the classical representation.
- Given a fixed integration step-size, dual quaternions have less accuracy in the solutions than the classical representations, for the tested scenarios.
- For the described nominal scenarios, the dual quaternion control approach achieves the same order of magnitude in errors and energy consumption than the classical approach, taking more computational time (about 15% more).
- Dual quaternions can handle larger perturbations in the tested scenarios (between 23 and 227 times larger depending on the scenario).
- Dual quaternions are more efficient in terms of energy consumption when correcting deviations from the nominal trajectory in the tested scenarios (between 3.2 and 9.3 times less).
- Dual quaternions have shorter settling times. Specifically, between 29% and 92% shorter settling times depending on the tested scenarios.
- Dual quaternions accumulate less errors (between 2.6 and 19 times less than classical methods) when correcting deviations from the nominal trajectory in the tested scenarios.

9.2. RECOMMENDATIONS AND FUTURE WORK

Despite the fact that this thesis covered an extensive work and successfully compared the dual quaternion approach with classical methods for asteroid close-proximity operations; several further steps and recommendations can be given to further investigate on the topic. Therefore, this section states them.

- The reason why dual quaternions are very sensitive to control frequency reduction for the hovering scenario has to be further studied.
- The work done in this thesis could be further applied to other scenarios such as docking between spacecrafts in Earth orbit or close-proximity operations in other asteroids.
- The performance of controllers with adaptive algorithms could be compared between the classical and the dual quaternion approaches.

- The gain selection methodology could be further studied to reduce the computational time involved and improve the accuracy of the results.
- It would be great to study the full dimension of the problem by introducing a filter and statistical orbit determination

APPENDIX A

STABILITY THEORY

To understand and design controllers capable of stabilizing spacecraft it is important to bear some theory and definitions in mind. This appendix gives the necessary theory to understand the design of the controllers proposed in Chapter 5. All this theory has been taken from [Schaub and Junkins, 2009].

STABILITY DEFINITIONS

There are different types of stability. Let \mathbf{x} be a generalized state vector; then nonlinear dynamical systems can be written in the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \quad (\text{A.1})$$

To define stability of a dynamical system, some notions of an equilibrium state \mathbf{x}_e and nominal reference motion \mathbf{x}_r are required.

- **Equilibrium state:** A state vector point \mathbf{x}_e is said to be an equilibrium state (or equilibrium point) of a dynamical system described by $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ at time t_0 if

$$\mathbf{f}(\mathbf{x}_e, t) = 0 \quad \forall t > t_0 \quad (\text{A.2})$$

If the dynamical system is to follow a prescribed motion, then this motion is referred to as the nominal reference motion $\mathbf{x}_r(t)$. To describe the proximity of one state to another, the notion of neighborhoods is defined.

- **Neighborhood B_δ :** Given $\delta > 0$, a state vector $\mathbf{x}(t)$ is said to be in the neighborhood $B_\delta(\mathbf{x}_r(t))$ of the state $\mathbf{x}_r(t)$ if

$$\|\mathbf{x}(t) - \mathbf{x}_r(t)\| < \delta \implies \mathbf{x}(t) \in B_\delta(\mathbf{x}_r(t)) \quad (\text{A.3})$$

A simple form of stability is the concept of a motion simply being bounded (or Lagrange stable) relative to $\mathbf{x}_r(t)$. Note that $\mathbf{x}(t_0)$ could lie arbitrarily close to $\mathbf{x}_r(t_0)$ while $\mathbf{x}(t)$ may still deviate from $\mathbf{x}_r(t)$. The only stability guarantee made here is that this state vector difference will remain within a finite bound δ .

- **Lagrange stability:** The motion $\mathbf{x}(t)$ is said to be Lagrange stable (or bounded) relative to $\mathbf{x}_r(t)$ if there exists a $\delta > 0$ such that

$$\mathbf{x}(t) \in \mathbf{B}_\delta(\mathbf{x}_r(t)) \quad (\text{A.4})$$

Declaring a motion to be Lyapunov stable (also referred to simply as being stable) is a stronger statement than saying it is Lagrange stable.

- **Lyapunov stability:** The motion $\mathbf{x}(t)$ is said to be Lyapunov stable (or stable) relative to $\mathbf{x}_r(t)$ if for each $\epsilon > 0$ there exists a $\delta(\epsilon) > 0$ such that

$$\mathbf{x}(t_0) \in \mathbf{B}_\delta(\mathbf{x}_r(t_0)) \implies \mathbf{x}(t) \in \mathbf{B}_\epsilon(\mathbf{x}_r(t)) \quad \forall t > t_0 \quad (\text{A.5})$$

In other words, if the state vector $\mathbf{x}(t)$ is to remain within any arbitrarily small neighborhood \mathbf{B}_ϵ of $\mathbf{x}_r(t)$, then there exists a corresponding initial neighborhood $\mathbf{B}_\delta(\mathbf{x}_r(t_0))$ from which all $\mathbf{x}(t)$ must originate. A stronger stability statement is to say the motion $\mathbf{x}(t)$ is asymptotically stable. In this case the difference between $\mathbf{x}(t)$ and $\mathbf{x}_r(t)$ will approach zero over time.

- **Asymptotic Stability:** The motion $\mathbf{x}(t)$ is asymptotically stable relative to $\mathbf{x}_r(t)$ if $\mathbf{x}(t)$ is Lyapunov stable and there exists a $\delta > 0$ such that

$$\mathbf{x}(t_0) \in \mathbf{B}_\delta(\mathbf{x}_r(t_0)) \implies \lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{x}_r(t) \quad (\text{A.6})$$

Except for the Lagrange stability definition, all other types of stabilities defined are referred to as local stability. The initial state vector has to be within a certain neighborhood \mathbf{B}_δ relative to the desired state vector for stability to be guaranteed. If stability is guaranteed for any initial state vector $\mathbf{x}(t_0)$, then the system is said to be globally stable or stable at large.

LYAPUNOV DIRECT METHOD

Proving stability of nonlinear systems with the basic stability definitions and without resorting to local linear approximations can be quite tedious and difficult. Lyapunov's direct method provides a tool to make rigorous, analytical stability claims of nonlinear systems by studying the behavior of a scalar, energy-like Lyapunov function. A major benefit of this method is that this can be done without having to solve the nonlinear differential equations.

To prove stability of a dynamical system, special positive definite functions called Lyapunov functions are sought.

- **Lyapunov function:** The scalar function $V(\mathbf{x})$ is a Lyapunov function for the dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ if it is continuous and there exists a $\delta > 0$ such that for any $\mathbf{x} \in \mathbf{B}_\delta(\mathbf{x}_r)$

1. $V(\mathbf{x})$ is a positive definite function about \mathbf{x}_r .
2. $V(\mathbf{x})$ has continuous partial derivatives.
3. $\dot{V}(\mathbf{x})$ is negative semidefinite.

Even though $V(\mathbf{x})$ explicitly depends on only the state vector \mathbf{x} , because $\mathbf{x}(t)$ is time varying, the Lyapunov function V is time varying, too. Using the chain rule, the derivative of V is found to be

$$\dot{V} = \frac{\partial V}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) \quad (\text{A.7})$$

Bearing this definitions in mind, the following theorems can be stated:

- **Theorem A.1 - Lyapunov stability** If a Lyapunov function $V(\mathbf{x})$ exists for the dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, then this system is stable about the origin.
- **Theorem A.2 - Asymptotic stability** Assume $V(\mathbf{x})$ is a Lyapunov function about $\mathbf{x}_r(t)$ for the dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$; then the system is asymptotically stable if
 1. the system is stable about $\mathbf{x}_r(t)$.
 2. $\dot{V}(\mathbf{x})$ is negative definite about $\mathbf{x}_r(t)$.

And this is theorem that has been used to proof the stability of the controllers developed in this master thesis work.

BARBALAT'S THEOREM

Sometimes, for time varying systems, it is difficult to find a Lyapunov function with a negative definite derivative. This is where Barbalat's theorem comes into play. It says:

- **Barbalat's lemma** if $V(\mathbf{x}, t)$ satisfies the following conditions:
 1. $V(\mathbf{x}, t)$ is lower bounded
 2. $\dot{V}(\mathbf{x}, t)$ is negative semi-definite
 3. $\dot{V}(\mathbf{x}, t)$ is uniformly continuous in time (satisfied if \ddot{V} is finite); then

$$\dot{V}(\mathbf{x}, t) \rightarrow 0 \quad \text{as} \quad t \rightarrow \infty \quad (\text{A.8})$$

This lemma is used during the proof of stability of the dual quaternion controller.

APPENDIX B

OTHER SYSTEM TEST RESULTS

This appendix contains all the results from the system tests explained in Subsection 7.2.4. For the sake of readability, only the results from the landing scenario with classical representation were shown. Thus, this appendix shows all the remaining results to show that all system tests were successful.

ORBITING - CLASSIC

To start with, results from the system test with the classical approach for the orbiting scenario are presented below. The order in which the results are presented follows the same structure as in the main document, (Subsection 7.2.4). First of all, the different state variables evolution over time is presented in Figure B.1. As can be seen, it corresponds to one orbital period. The upper-left plot corresponds to the position components, whereas the upper-right plot shows the linear velocities. Both of them start and finish at the same values due to the stable orbit around Bennu. On the bottom-left side, the quaternion orientation is presented. Then, on the bottom-right side of Figure B.1, the angular velocities are shown. The blue box corresponds to a zoom-in of a specific part of the plot for better appreciation of how variables converge to their reference values.

Next, Figure B.2 shows the errors of the attitude quaternion and the angular velocity with respect to the reference values. It is worth remembering that for the orbiting scenarios, only the rotational motion was controlled. For this reason, only the pointing errors and the angular velocity errors are presented. As can be seen, after a settling time, both motions reach stability and remain within the requirements specified. For the pointing error, there is no overshoot and the error goes directly to zero from its starting point. However, for the angular velocity, there is a clear overshoot before reaching stability. The settling time is about 400 s. The blue box is again the zoom-in area for better visualizing how each error is converging to its reference value.

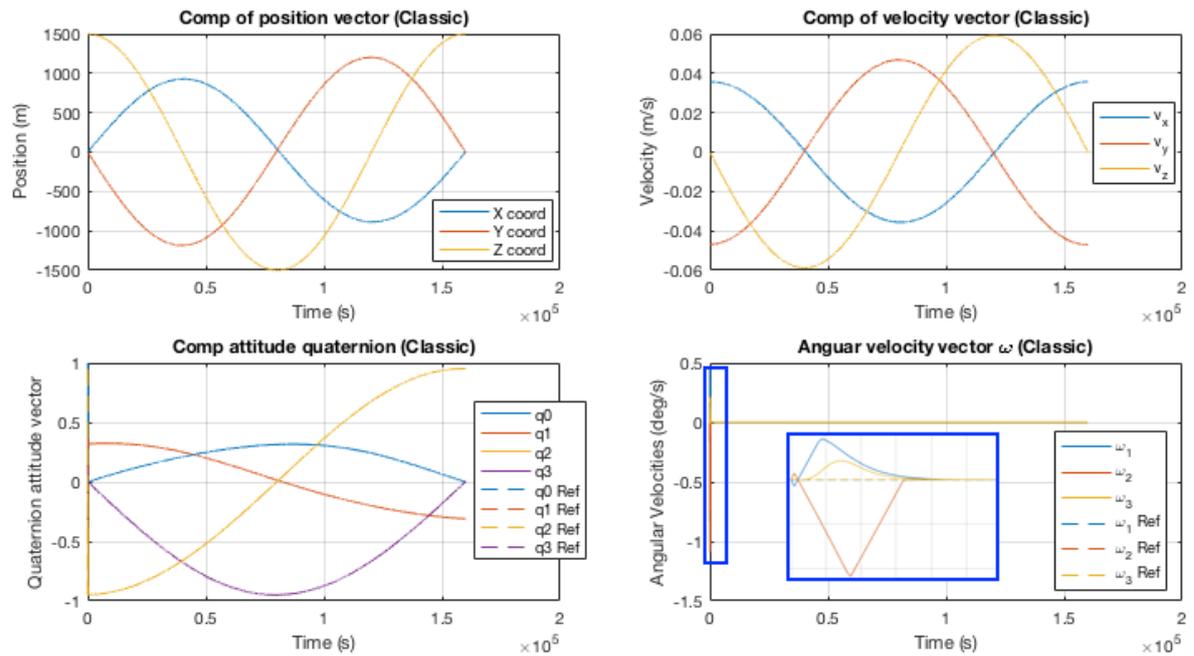


Figure B.1: Orbiting state variables and reference values with classic approach

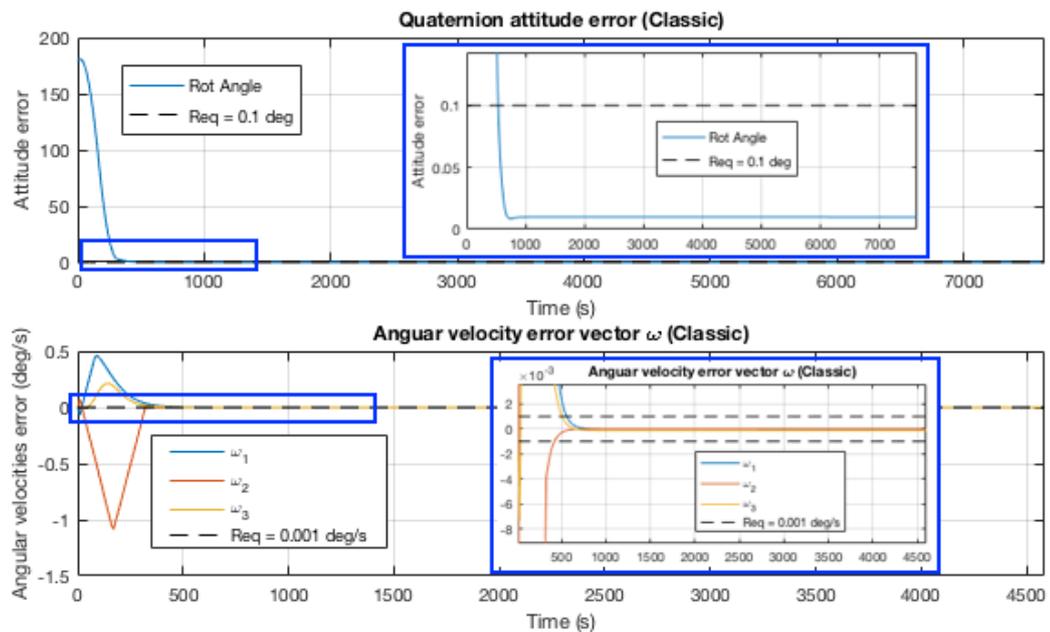


Figure B.2: Orbiting state variable errors with respect to reference trajectory with classic approach

Finally, the different control commands torques are presented in Figure B.3. It is interesting to notice that only the first 900 s are shown in this figure, for better appreciating how the controller manage to steer the spacecraft back to the reference trajectory. During the first seconds, the controller is clearly saturated reaching its maximum allowed values. Due to actuators limitations, there is a maximum torque of 0.2 Nm in each body direction that cannot be exceeded. During the first 300 s, the actuators are saturated before stabilizing the spacecraft to the reference trajectory. It is also worth mentioning that, after stabilization, the

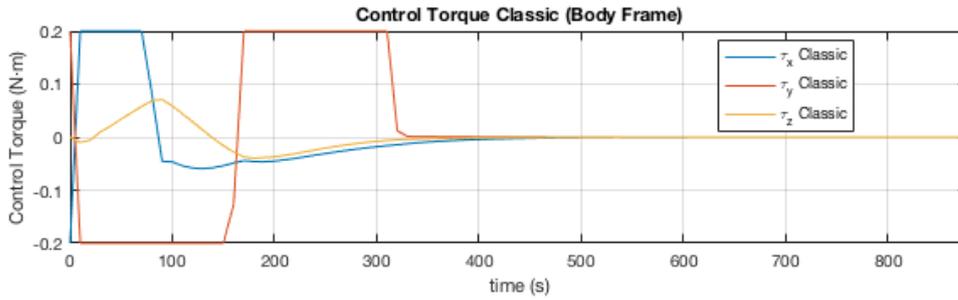


Figure B.3: Orbiting control forces and torques commands with classic approach

control torque commands are not zero. They have the values required to follow the reference trajectory. However, since these values are so small, they cannot be appreciated in this figure.

As a conclusion, the system test for the classical orbiting scenario is a success. All variables manage to get stabilized in the reference trajectory and remain there for the duration of the simulation. This proves that the simulator and the controller as a whole system works as expected.

ORBITING - DUAL QUATERNIONS

Next, results from the system test with the dual quaternion approach, for the orbiting scenario, are presented. They are very similar to the ones of the classical approach. First of all, the different state variables evolution over time is presented in Figure B.4. The upper-left plot corresponds to the position components, whereas the upper-right plot shows the linear velocities. Their values are identical to the previous system test with classical control. However, on the bottom-left side, the quaternion orientation plot shows some differences. More specifically, they look anti-symmetric with respect to the classical control. The reason for this is because the initial orientation error was set to be 180° , as it can be seen in Table 7.3. The classical control approach forced the spacecraft to rotate in one direction, whereas the dual quaternion controller chose the other one. This can also be observed in the bottom-right side of Figure B.4. The angular velocities of the body x and y axes are exactly the opposite than the ones with classical control. The blue box corresponds to a zoom-in of a specific part of the plot for better appreciation of how variables converge to their reference values.

Moreover, Figure B.5 shows the errors of the attitude quaternion and the angular velocity with respect to the reference values, for the dual quaternion orbiting system test. Again, only the pointing error and the angular velocity errors are shown because no orbit control has been applied. On the upper part, the pointing error goes from 180° to almost 0° very quickly (in about 500 s). The difference between the dual quaternion approach and the classical one is that the first one reduces the error much faster than the second one. The same can be noticed from the angular velocity errors shown on the bottom part of the figure. The overshoot for the angular velocities is as high as the classical one. The blue box is again the zoom-in area for better visualizing how each error is converging to its reference value.

Finally, the different control commands torques for the dual quaternion representation are presented in Figure B.6. Two aspects are worth mentioning when comparing these results with the classical ones. The first

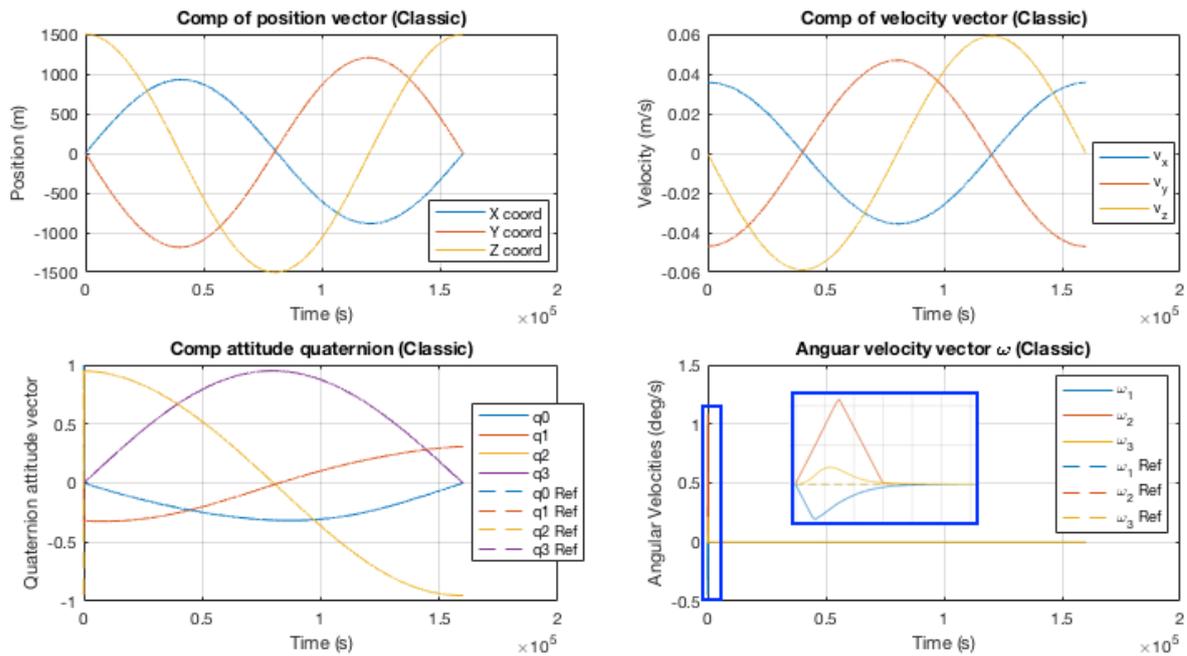


Figure B.4: Orbiting state variables and reference values with dual quaternion approach

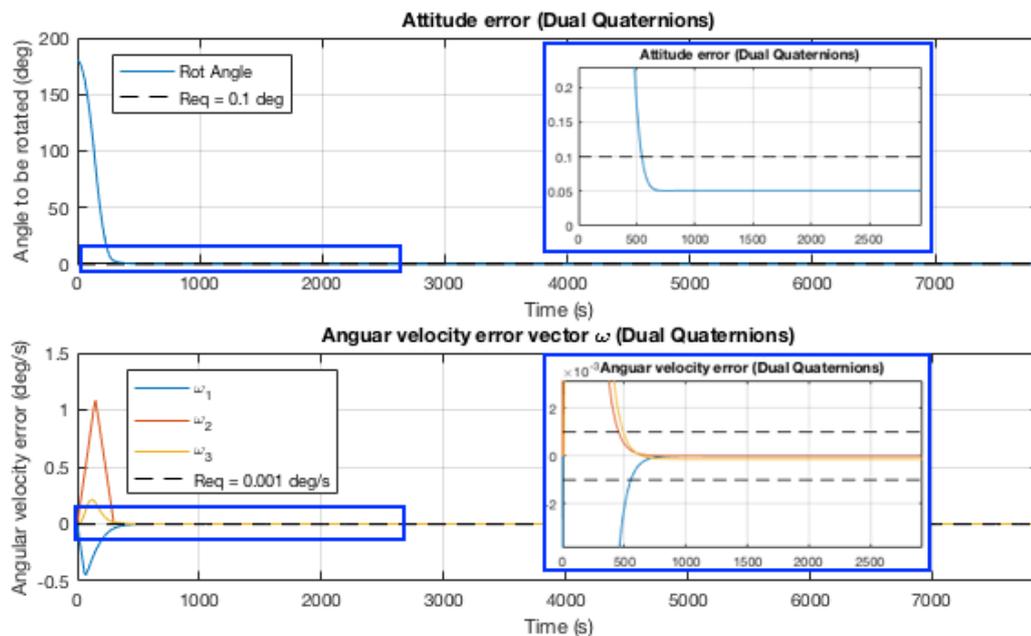


Figure B.5: Orbiting state variable errors with respect to reference trajectory with dual quaternion approach

one is that the settling time is shorter. The second one is that the control torque commands have the opposite shape than for the classical approach. This is the reason why the state variables look anti-symmetric. As has been said already, the dual quaternion approach chose to steer the spacecraft in the exact opposite direction than the classical control, when starting with 180° of pointing initial error. Furthermore, it is also important to notice that only the first 900 s have been represented in this figure to appreciate how the control commands evolve overtime. Otherwise, it would have been a straight line very close to zero, over the more than 7000 s of

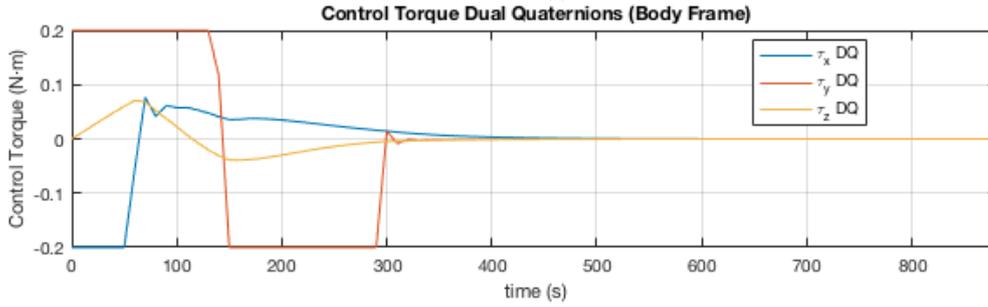


Figure B.6: Orbiting control forces and torques commands with dual quaternion approach

simulation.

This system test proves that the simulator developed with the implemented controller work as expected and manage to achieve the mission requirements, both with a classical representation and with the dual quaternion one. All sub-blocks defined in Section 7.1 work together as a system successfully.

HOVERING - CLASSIC

Next, results from the system test with the classical approach, for the hovering scenario, are presented below. As has been done in previous sections, the different state variables evolution over time are presented in Figure B.7. The upper-left plot corresponds to the position components. The reference values are also shown in dashed lined. Since the trajectory is an inertial hovering about Bennu, the position components are straight lines. The upper-right plot of the figure shows the velocity components. As can be seen, there is a clear overshoot to correct the position of the satellite and then, all velocity components go to almost zero. In this case, the overshoot has an order of magnitude of 0.1 m/s. It is important to remember that the initial conditions and gain settings for this simulation have been stated in Sub-section 7.2.4. The bottom-left side of Figure B.7 shows the different quaternion components over time, with the reference values. And last, in the bottom-right side of the figure, the angular velocity components are represented. They also show a clear overshoot, as the linear velocity did. In this case, the order of magnitude is about $0.08^{\circ}/s$. The reason for that is because the orientation of the satellite had to be corrected changing the angular velocity, to finally stabilize on the reference value.

Figure B.8 shows the errors of each state variable over time. As can be seen, the linear and angular velocities look exactly the same as in Figure B.7. This is because the reference velocities are zero. However, the a blue box have been inserted to zoom in and appreciate that both state variables actually converge within the requirements of the mission. Regarding the left part of the figure, the position and attitude errors are shown. They quickly go to zero (in about 40 s). It is important to mention that, in contrast with the orbiting scenario, orbit control was required for hovering. For this reason, position and velocity errors are presented in Figure B.8.

Finally, both the force and torque control commands are represented in Figure B.9. There is a clear saturation of the actuators in both cases before reaching stability. It is worth mentioning that only the first 55 s are

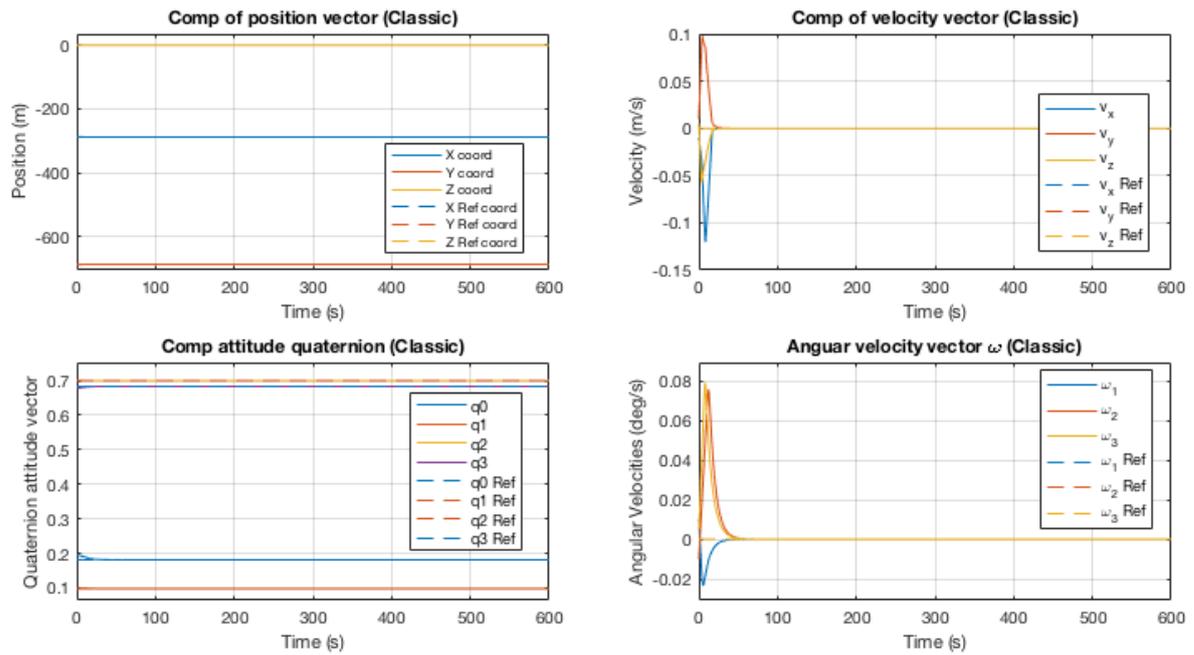


Figure B.7: Hovering state variables and reference values with classic approach

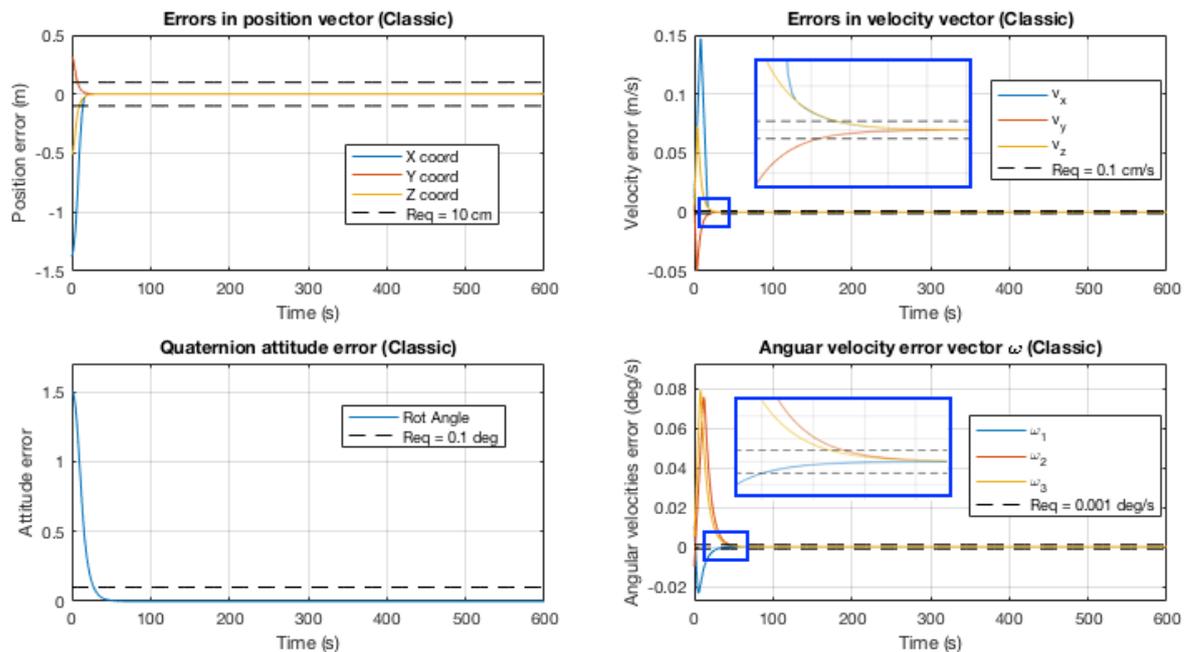


Figure B.8: Hovering state variable errors with respect to reference trajectory with classic approach

shown in this figure, for better appreciating how each controller manage to stabilize the spacecraft. As can be seen, it takes a little bit longer for the attitude motion to reach stability. Last, it is important to realize that the control commands never go to exactly zero values. There is always a tinny torque or force to be compensated so that inertial hovering can be achieved.

The conclusion for this system test is exactly the same than for the previous ones. It proves that the sim-

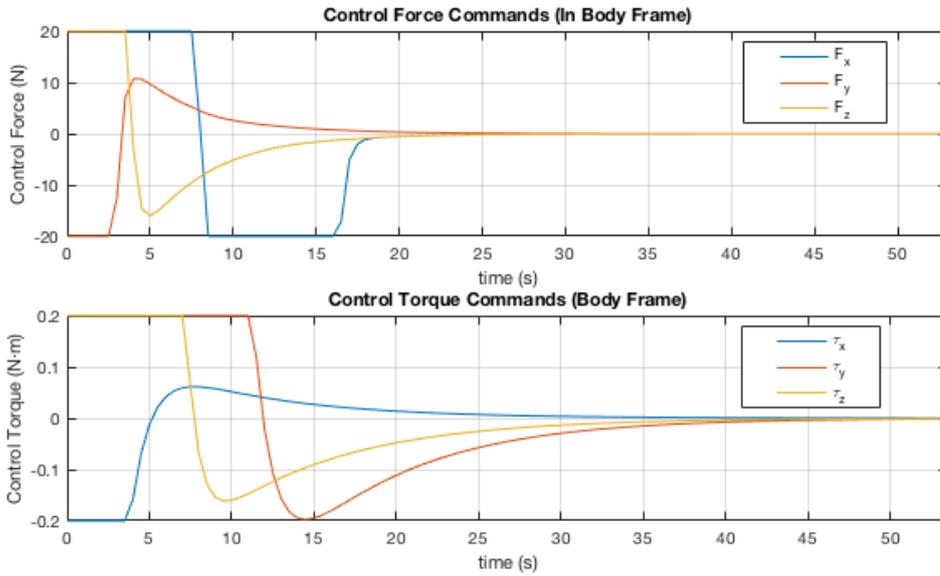


Figure B.9: Hovering control forces and torques commands with classic approach

ulator developed and the classical control for hovering works as expected and provides us with valid results.

HOVERING - DUAL QUATERNIONS

Here, results from the system test with the dual quaternion approach, for the hovering scenario, are presented. Again, they are very similar to the ones of the classical approach. The different state variables evolution over time is presented in Figure B.10. The distributions of the different state variables is the same as in the previous cases. The main difference with respect to the classical approach is that the overshoot for the angular velocity is slightly smaller. In this case, the order of magnitude is $0.07^\circ/s$ instead of $0.08^\circ/s$. For the linear velocity, they remain equal.

Figure B.11 shows the different errors for each state variable. The performance of the dual quaternion controller is very similar to the classical one for this hovering scenario. The only difference is the already mentioned smaller overshoot for the angular velocity. The blue box areas are a zoom-in of specific regions of the plot to clarify how each state variable converges to the reference solution.

Finally, Figure B.12 shows both the force and torque commands for the dual quaternion controller in the hovering scenario. As can be seen, the actuators are saturated for almost 15 s and then manage to stabilize the spacecraft to the hovering trajectory. The settling time in this case is the same as for the classical representation.

LANDING - DUAL QUATERNIONS

Last but not least, results from the landing scenario with the dual quaternion approach are represented below. For the sake of simplicity, only the classical representation for the landing scenario was shown in the

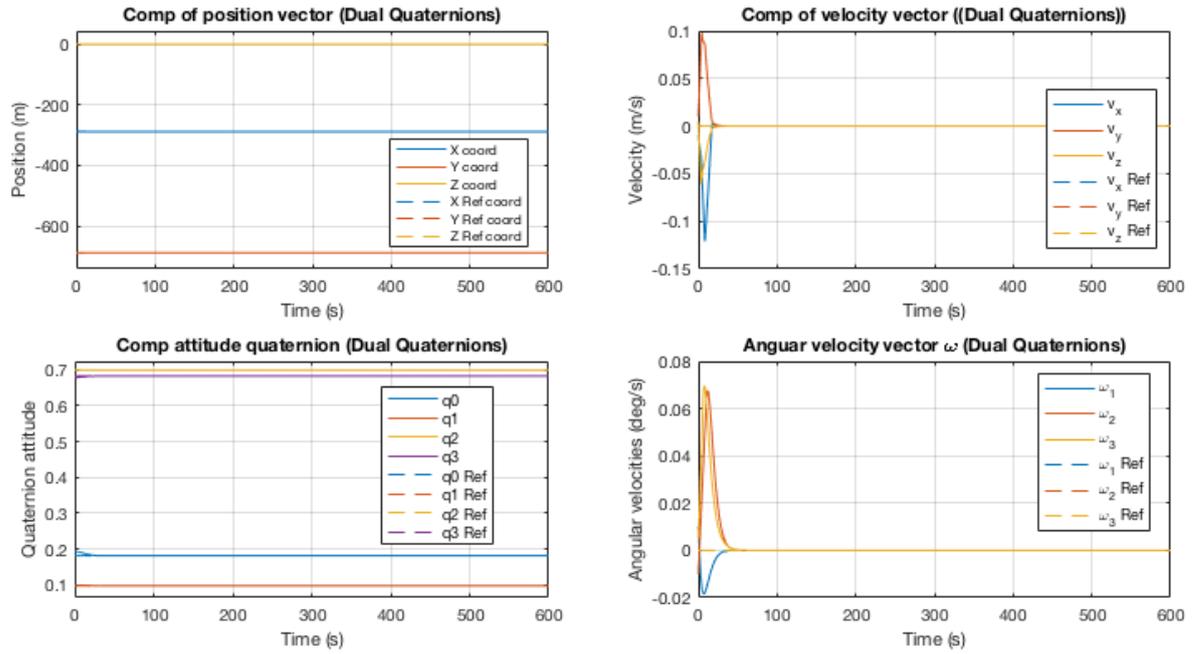


Figure B.10: Hovering state variables and reference values with dual quaternion approach

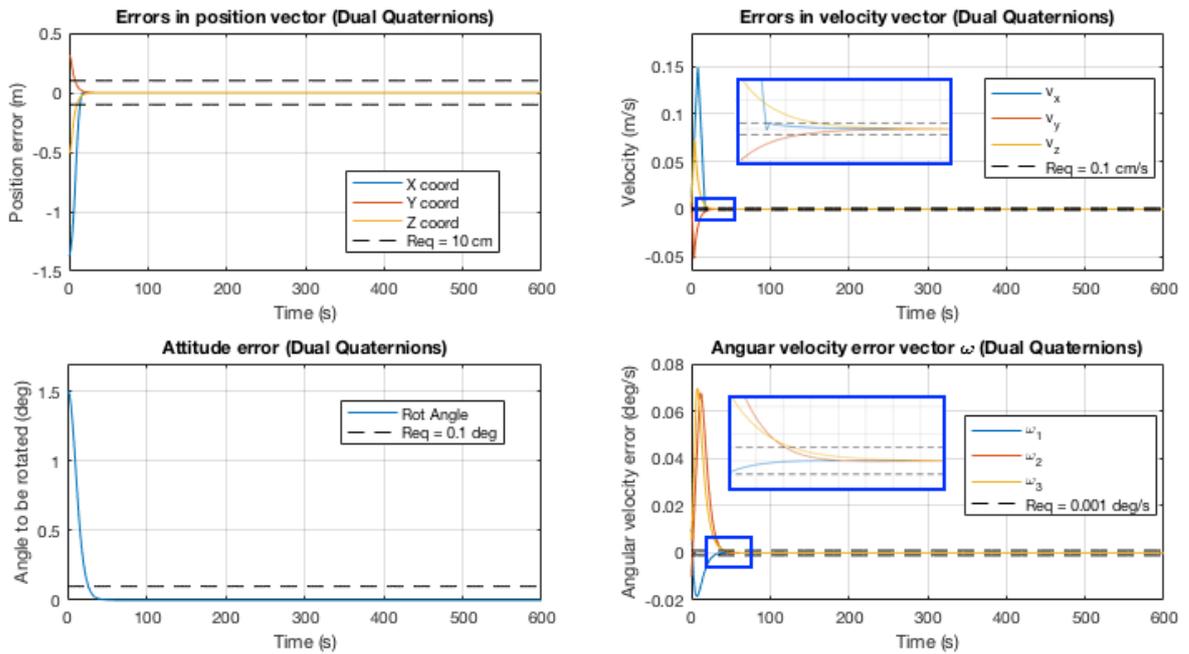


Figure B.11: Hovering state variable errors with respect to reference trajectory with dual quaternion approach

main document. Here, system test results using the dual quaternion approach are shown to prove that everything implemented in this thesis, works properly and as expected. First, Figure B.13 shows the different state variables evolution over time. All simulation settings and initial conditions are described in Subsection 7.2.4 of the main document. The gains used for this simulations are also stated. As can be seen, all different state variables manage to follow the reference trajectory (shown in dashed lines) to finally land on the surface of

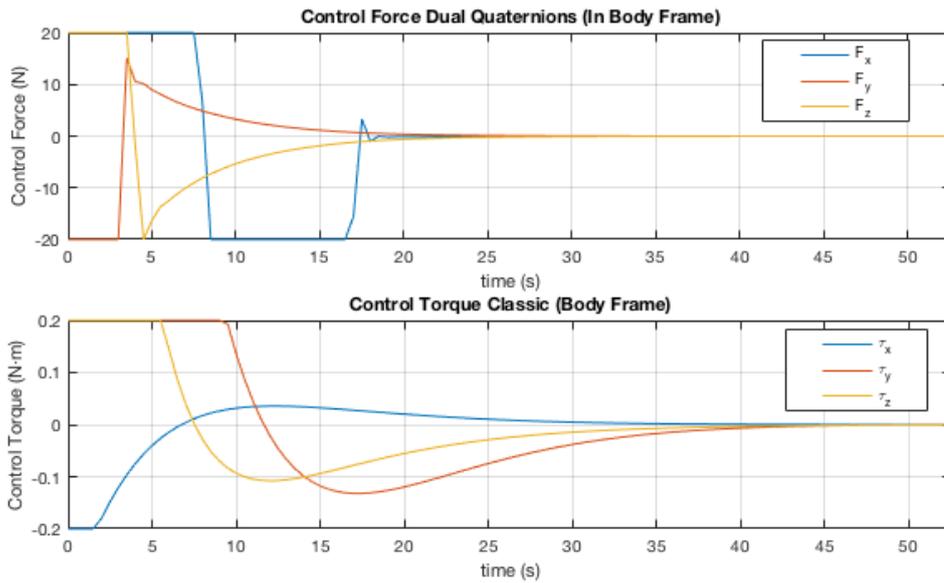


Figure B.12: Hovering control forces and torques commands with dual quaternion approach

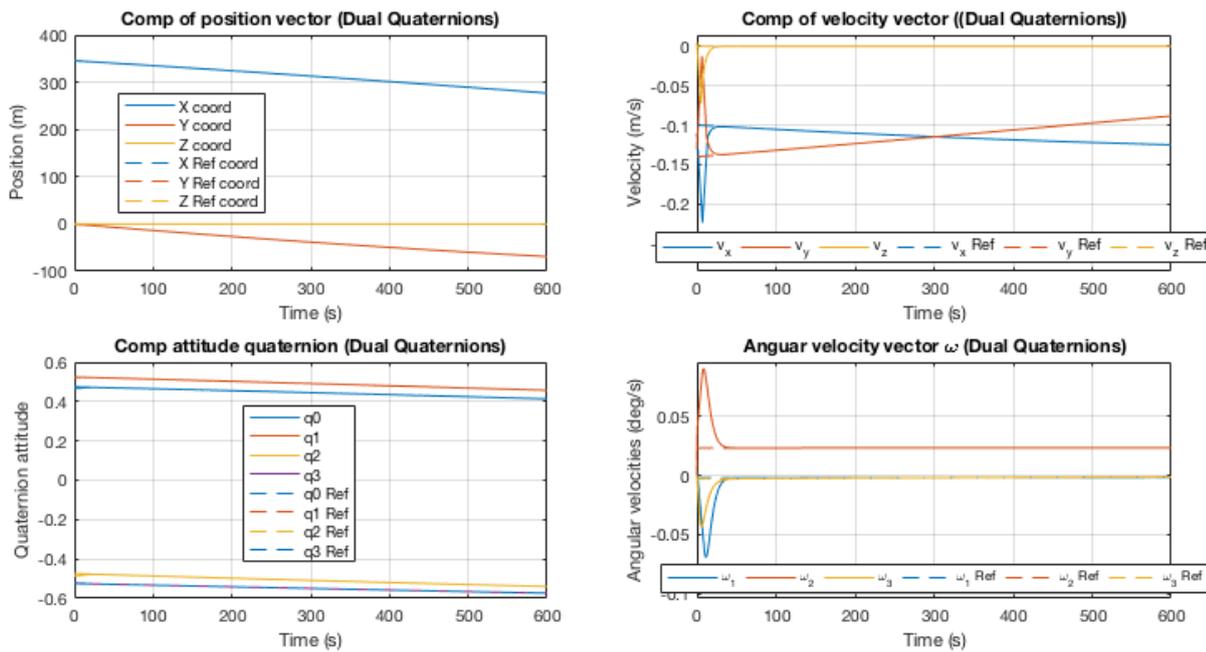


Figure B.13: Landing state variables and reference values for the landing scenario with dual quaternion approach

Bennu meeting the specified requirements.

Figure B.14 shows the different errors for each state variable. As can be appreciated, the initial conditions are not the nominal ones. For this reason, there is a clear overshoot in both the angular and linear velocities to compensate for these errors, and then follow the reference trajectory. The maximum overshoot for the linear velocity has an order of magnitude of 0.15 m/s, whereas it is 0.06°/s for the angular velocity. On the bottom-left side of the figure, the pointing error is represented showing how quickly the controller manages to correct the initial attitude error. Comparing these results with the ones in the main document for the

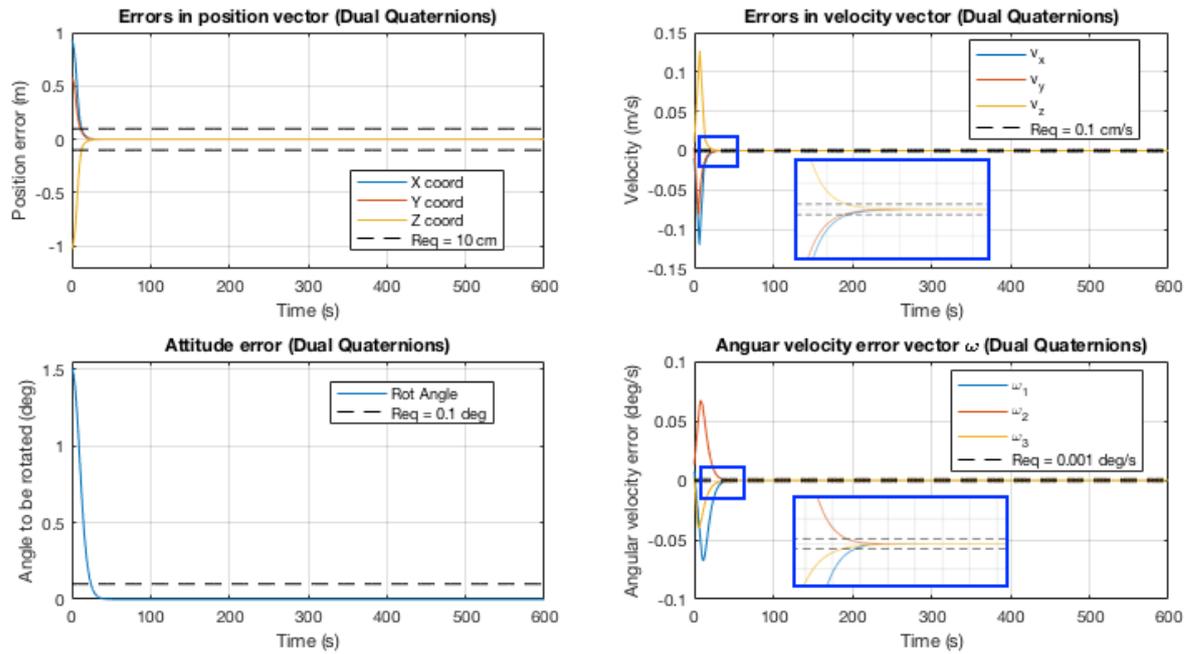


Figure B.14: Landing state variable errors with respect to reference trajectory with dual quaternion approach

classical controller, it can be said that the overshoot for the angular velocity here, is slightly smaller than the $0.08^\circ/\text{s}$ for the classical control. However, for the linear velocity, it is a little bit bigger (compared to 0.1 m/s of the classical representation). The blue boxes still represent the zoom-in areas for better appreciating how each error converges to zero.

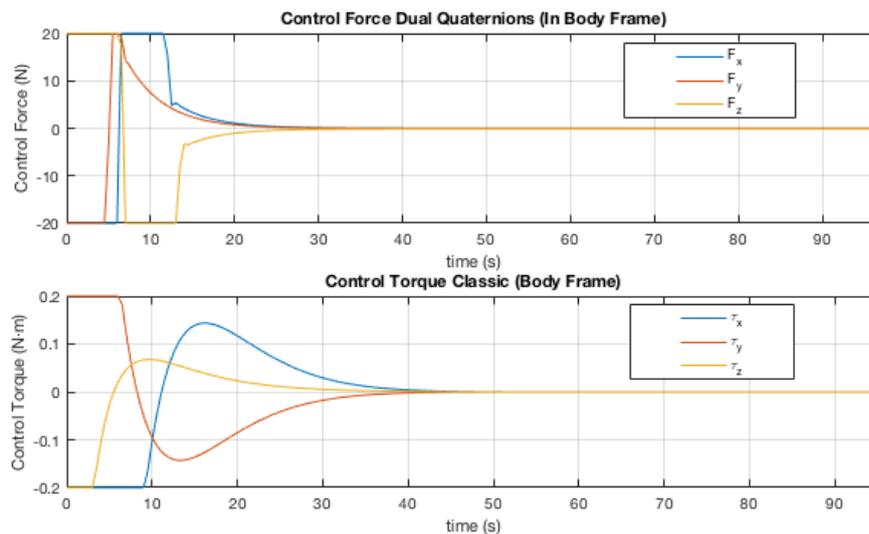


Figure B.15: Landing control forces and torques commands with dual quaternion approach

Finally, Figure B.15 shows both the force and torque commands of the dual quaternion controller used to guide the spacecraft through the landing trajectory. In this case, only for the first 15 s the actuators are saturated. Then, they converge to almost zero, where the spacecraft reaches convergence. It is important

noticing that they never go to exactly zero values, because there are always some tinny torques or forces to be performed so that the spacecraft follows the reference trajectory.

The final conclusion of this Appendix B is that all system tests carried out have been successful. The different blocks defined in Section 7.1, can work together as a system and provide the user with interesting and valid results, which can be used to answer the research question of this master thesis. For all different scenarios defined in this work, both the classical and the dual quaternion controllers manage to steer the spacecraft through the reference trajectory meeting the mission requirements. The purpose of this Appendix is to prove and show the system test results of all the different simulations that have been carried out in this thesis, and that could not be shown in the main document for the sake of simplicity.

BIBLIOGRAPHY

- Aspragathos, N. A. and Dimitros, J. K. (1998). A Comparative Study of Three Methods for Robot Kinematics. *Ieee Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 28(2):135–145.
- Beckman, M., Barbee, B. W., Williams, B. G., Williams, K., Sutter, B., and Berry, K. (2013). Trajectory and Mission Design for the Origins Spectral Interpretation Resource Identification Security Regolith Explorer Asteroid Sample Return Mission. *Planetary Defence Conference 2013, IAA*, 19:1–13.
- Broschart, S. B. and Scheeres, D. J. (2005). Control of Hovering Spacecraft Near Small Bodies : Application to Asteroid 25143 Itokawa. *Journal of guidance, control and dynamics*, 28(2):343–354.
- Celledoni, E., Fassò, F., Säfström, N., and Zanna, A. (2008). The exact computation of the free rigid body motion and its use in splitting methods. *SIAM Journal on Scientific Computing*, 30(4):2084–2112.
- Clifford (1871). Preliminary sketch of biquaternions. *Proceedings of the London Mathematical Society*, s1-4(1):381–395.
- Daniilidis, K. (1999). Hand-eye calibration using dual quaternions. *The International Journal of Robotics Research*, 18(3):286–298.
- Everett, D., Linn, T., Mink, R., and Wood, J. (2017). Designing to Sample the Unknown : Lessons from OSIRIS-REx Project Systems Engineering. *IEEE Aerospace Conference, 2017*, 40(2):1–19.
- Filipe, N. and Tsiotras, P. (2013). Rigid Body Motion Tracking Without Linear and Angular Velocity Feedback Using Dual Quaternions. *European Control Conference (ECC), 2013*, (125):329–334.
- Filipe, N. and Tsiotras, P. (2014). Adaptive position and attitude tracking controller for satellite proximity operations using dual quaternions. *Advances in the Astronautical Sciences*, 150(4):2313–2332.
- Filipe, S. and Ricardo, N. (2014). *Nonlinear pose control and estimation for space proximity operations: An approach based on dual quaternions*. PhD thesis, Georgia Institute of Technology.
- Funda, J. and Paul, R. P. (1990). A Computational Analysis of Screw Transformations In Robotics. *Transactions on robotics and automation*, 6(3):348–356.
- Ge, Q. and Ravani, B. (1994). Computer aided geometric design of motion interpolants. *Journal of Mechanical Design*, 116(3):756–762.
- Gottlieb, R. G. (1993). Fast gravity, gravity partials, normalized gravity, gravity gradient torque and magnetic field: derivation, code and data. *NASA Contractor Report 188243*, pages 64–65.

- Gui, H. and Vukovich, G. (2015). Dual-quaternion-based adaptive motion tracking of spacecraft with reduced control effort. *Nonlinear Dynamics*, 83(1):597–614.
- Han, D., Wei, Q., Li, Z., and Sun, W. (2008). Control of oriented mechanical systems: A method based on dual quaternion. *IFAC Proceedings Volumes*, 41(2):3836–3841.
- Hergenrother, C. W. e. a. (2014). The Design Reference Asteroid for the OSIRIS-REx Mission Target (101955) Bennu. *arXiv preprint arXiv:1409.4704*, 1409(101955):4704–4708.
- Jones, B. A. (2010). *Efficient models for the evaluation and estimation of the gravity field*. PhD thesis, University of Colorado at Boulder.
- Kumar, K. D. (2008). Attitude dynamics and control of satellites orbiting rotating asteroids. *Acta Mechanica*, 198(1):99–118.
- Lee, D., Sanyal, A. K., Butcher, E. A., and Scheeres, D. J. (2015). Finite-time control for spacecraft body-fixed hovering over an asteroid. *IEEE Transactions on Aerospace and Electronic Systems*, 51(1):506–520.
- Lissauer, J. J. and de Pater, I. (2013). *Fundamental Planetary Science*. Cambridge University Press, ISBN 9780521618557.
- Lorenz, D. A., Olds, R., May, A., Mario, C., Perry, M. E., Palmer, E. E., and Daly, M. (2017). Lessons learned from osiris-rex autonomous navigation using natural feature tracking. *IEEE Aerospace Conference, 2017*, 32(1):1–12.
- Markley, F. L. and Crassidis, J. L. (2014). *Fundamentals of Spacecraft Attitude Determination and Control*, volume 33. Springer, ISBN - 978493908011.
- Masashi, U., Kenichi, S., Tatsuaki, H., Takashi, K., and Jun'ichiro, K. (2006). Attitude Control Challenges and Solutions for Hayabusa Spacecraft. *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, (28):1–10.
- May, A., Sutter, B., Linn, T., Bierhaus, B., Berry, K., and Mink, R. (2014). OSIRIS-REx Touch-and-Go (TAG) Mission Design for Asteroid Sample Collection. *International Astronautical Congress; 65th*, (121):1–11.
- McMahon, J., Scheeres, D., Hesar, S., Farnocchia, D., Chesley, S., and Lauretta, D. (2018). The osiris-rex radio science experiment at bennu. *Space Science Reviews*, 214(1):43.
- McMahon, J. W. (2011). *An analytical theory for the perturbative effect of solar radiation pressure on natural and artificial satellites*. PhD thesis, University of Colorado at Boulder.
- Misra, G. and Sanyal, A. K. (2015). Analysis of orbit-attitude coupling of spacecraft near small solar system bodies. *AIAA Guidance, Navigation, and Control Conference*, page 1777.
- Montenbruck, O. and Gill, E. (2000). *Satellite Orbits: Models, Methods and Applications*, volume 134. Springer.
- Murray, R. M., Li, Z., Sastry, S. S., and Sastry, S. S. (1994). *A mathematical introduction to robotic manipulation*. CRC press.

- Nelson, R. L. (2001). 433 Eros Landing - Development of NEAR Shoemaker's Controlled Descent Sequence. *15th Annual AIAA/USU Conference on Small Satellites*, 15(1):1–7.
- Razgus, B., Mooij, E., and Choukroun, D. (2017a). Relative Navigation in Asteroid Missions: A Dual Quaternion Approach. *AIAA Guidance, Navigation, and Control Conference*, page 1521.
- Razgus, B., Mooij, E., and Choukroun, D. (2017b). Relative navigation in asteroid missions using dual quaternion filtering. *Journal of Guidance, Control, and Dynamics*, pages 1–16.
- Regnier, P., Ecale, E., and Serra, S. V. (2000). AOCS Design for scientific Missions ROSETTA/MARS EXPRESS. *Spacecraft Guidance, Navigation and Control Systems*, 425:3–7.
- Schaub, H. and Junkins, J. L. (2009). *Analytical Mechanics of Space Systems*. American Institute of Aeronautics and Astronautics, Virginia ISBN - 9781600867217, 4th edition.
- Scheeres, D., Hesar, S., Tardivel, S., Hirabayashi, M., Farnocchia, D., McMahon, J., Chesley, S., Barnouin, O., Binzel, R., Bottke, W., et al. (2016). The geophysical environment of bennu. *Icarus*, 276:116–140.
- Scheeres, D., Khushalani, B., and Werner, R. A. (2000). Estimating asteroid density distributions from shape and gravity information. *Planetary and Space Science*, 48(10):965–971.
- Scheeres, D. J. (2012). Orbit Mechanics About Asteroids and Comets. *Journal of Guidance, Control, and Dynamics*, 35(3).
- Takahashi, Y. (2013). *Gravity Field Characterization around Small Bodies*. PhD thesis, University of Colorado at Boulder.
- Tsuda, Y., Yoshikawa, M., Abe, M., Minamino, H., and Nakazawa, S. (2013). System design of the hayabusa 2-asteroid sample return mission to 1999 JU3. *Acta Astronautica*, 91:356–362.
- Vukovich, G. and Gui, H. (2017). Robust Adaptive Tracking of Rigid-Body Motion with Applications to Asteroid Proximity Operations. *IEEE Transactions on Aerospace and Electronic Systems*, 9251:1–11.
- Wang, J., Liang, H., and Sun, Z. (2010). Relative Coupled Dynamics and Control Using Dual Number. *Systems and Control in Aeronautics and Astronautics (ISSCAA)*, 36(2):277–282.
- Wang, X. and Yu, C. (2010). Feedback linearization regulator with coupled attitude and translation dynamics based on unit dual quaternion. *IEEE International Symposium on Intelligent Control - Proceedings*, pages 2380–2384.
- Wang, Y., Guan, H., and Xu, S. (2012). Fourth-order gravity gradient torque of spacecraft orbiting asteroids. *Advances in the Astronautical Sciences*, 143:471–488.
- Wang, Y. and Xu, S. (2013). Gravity gradient torque of spacecraft orbiting asteroids. *Aircraft Engineering and Aerospace Technology*, 85(1):72–81.

- Wang, Y. and Xu, S. (2014). Gravitational Orbit-Rotation Coupling of a Rigid Satellite around a Spheroid Planet. *Journal of aerospace engineering ASCE*, 27:140–150.
- Werner, R. and Scheeres, D. (1997). Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 Castalia. *Celestial Mechanics and Dynamical Astronomy*, 65(3):313–344.
- Yuanxin, W., Xiaoping, H., Dewen, H., Tao, L., and Juanxiang, L. (2005). Strapdown Inertial Navigation System Algorithms Based on Dual Quaternions. *IEEE Transactions on Aerospace and Electronic Systems*, 41(1):135–137.
- Zhang, F and Duan, G. (2011). Robust Integrated Translation and Rotation Finite-Time Maneuver of a Rigid Spacecraft Based on Dual Quaternion. *AIAA Guidance, Navigation, and Control Conference*, 18(92):1–17.