

**ALMA MATER STUDIORUM
UNIVERSITA' DI BOLOGNA**

**SCHOOL OF ENGINEERING AND ARCHITECTURE
-Forlì Campus-**

**SECOND CYCLE MASTER'S DEGREE in
INGEGNERIA AEROSPAZIALE/ AEROSPACE ENGINEERING
Class LM20**

GRADUATION THESIS

In: Spacecraft Attitude Dynamics and Control

**“An optical navigation filter simulator for a CubeSat mission to Didymos
binary asteroid system”**

CANDIDATE

Alessandro Casadei

SUPERVISOR

Prof. Dario Modenini

EXAMINER

Prof. Fabrizio Giulietti

Academic Year 2017/2018

ABSTRACT

The orbital and the attitude control in real-time of an artificial satellite in orbit around a celestial body has been one of the most important aspects of space missions for decades, including those conducted through the use of CubeSats, namely nanosatellites and/or microsatellites of cubic form, increasingly used in space thanks to their simplified structure that requires low costs of design, production and putting into orbit, since the couplings between CubeSats and launchers take place through standardized processes that reduce the hours of work of the designers and offer the possibility of modifying the payload without having to fully re-evaluate the launch project, in addition to the fact that the size of a single CubeSat is so small that allows the launch of multiple nanosatellites simultaneously, going to further reduce the launch costs.

Moreover, the use of a CubeSat often offers the possibility to release it directly into orbit thanks to the transport on a "mother" satellite, necessary in many space missions to put the satellite in communication with the terrestrial operating centers, eventuality that allows further reduction of the launch costs.

Obviously, each CubeSat must be autonomous from the point of view of the production of electrical energy, which usually occurs through the usage of solar panels alternating with accumulators of electricity or batteries for the periods of occultation of sunlight, and from the point of view of navigation, as previously mentioned.

Although the artificial satellites often operate on terrestrial orbits LEO (low Earth orbit) or MEO (medium Earth orbit), for which it is possible to use the assisted navigation through Global Positioning System (GPS) for orbital determination, which requires the availability of a GPS receiver mounted onboard, there exist particular space missions outside the Earth's gravitational field that require autonomous navigation systems in real-time for both attitude control and orbital control.

The latter also includes AIDA, the joint NASA-ESA mission that will operate in the 65803 Didymos binary system and whose main purpose is to experiment and investigate the kinetic impact technique for the deviation of the asteroid trajectories in space with a view to a possible collision between one of them and the planet Earth in the future.

HERA, the "mother" satellite designed by ESA in collaboration with other European research institutes, will aim to collect data about the chemical-physical composition of the binary system and about the characteristics of the impact between DART, the bullet-satellite realized and run by NASA together with other US research institutes, and the minor of the two celestial bodies that compose Didymos, which should occur around October 2022.

Abstract

The "mother" HERA satellite will carry high-level technology onboard, including some CubeSats that will be released when HERA is already in orbit in the Didymos system.

This panorama also includes the DustCube mission, a project proposal for a CubeSat carried out by the Department of Industrial Engineering sited in Forlì (IT) of the University of Bologna in collaboration with other academic and research institutions of the European Union, which the main objective is to assist HERA in the acquisition of data concerning the impact between DART and the minor of the two celestial bodies that form Didymos.

The DustCube mission provides for the need to operate in complete autonomy, communicating in a continuous manner the data collected at HERA, which will send them to the terrestrial operating centers.

The physical configuration of the binary system and the type of mission, which requires the use of infrared cameras, focused on the two celestial bodies that form Didymos, which take pictures of them continuously, lead to prefer the development of an optical navigation filter able to estimate the position and the speed of the CubeSat starting from direction and range measurements with respect to a target body obtained precisely by means of optical technology.

This thesis project, which continues the work done during my curricular internship in the Microsatellite Laboratory of the University of Bologna sited in Forlì (IT), is part of the DustCube project with the aim of investigating in depth the orbital navigation of this CubeSat immersed in the Didymos system and the main one to develop an extended Kalman filter based on optical measurements that allows to simulate the DustCube real-time navigation.

SOMMARIO

Il controllo orbitale e d'assetto in tempo reale di un satellite artificiale in orbita attorno ad un corpo celeste riveste da decenni uno degli aspetti più importanti delle missioni spaziali, tra cui rientrano quelle condotte tramite l'utilizzo di CubeSats, ovvero nanosatelliti e/o microsattelliti di forma cubica, sempre più utilizzati in ambito spaziale grazie alla loro struttura semplificata che richiede bassi costi di progettazione, di produzione e di messa in orbita, dato che gli accoppiamenti tra CubeSats e lanciatori avvengono tramite processi standardizzati che riducono le ore di lavoro dei progettisti e che offrono la possibilità di modificare il carico utile senza dover rivalutare totalmente il progetto di lancio, oltre al fatto che l'ingombro di un singolo CubeSat è così ridotto da permettere il lancio di più nanosatelliti contemporaneamente, andando a ridurre ulteriormente i costi di lancio.

Inoltre, l'utilizzo di un CubeSat offre spesso la possibilità di un rilascio dello stesso direttamente in orbita grazie al trasporto su un satellite "madre", necessario in molte missioni spaziali per mettere in comunicazione il satellite con i centri operativi terrestri, eventualità che permette un'ulteriore riduzione dei costi di lancio.

Ovviamente, ogni CubeSat deve essere autonomo dal punto di vista della produzione di energia elettrica, che avviene solitamente tramite l'utilizzo di pannelli solari alternati ad accumulatori di energia elettrica o batterie per i periodi di occultazione della luce solare, e dal punto di vista della navigazione, come accennato precedentemente.

Sebbene spesso i satelliti artificiali operino su orbite terrestri LEO (low Earth orbit) o MEO (medium Earth orbit), per le quali è possibile utilizzare la navigazione assistita tramite Global Positioning System (GPS) per la determinazione orbitale, che richiede la disponibilità di un ricevitore GPS montato a bordo, esistono particolari missioni spaziali al di fuori dal campo gravitazionale terrestre che necessitano di sistemi di navigazione autonoma in tempo reale sia per il controllo d'assetto che per il controllo orbitale.

In quest'ultimo insieme rientra anche AIDA, la missione congiunta NASA-ESA che opererà nel sistema binario 65803 Didymos e che si prefigge come scopo principale quello di sperimentare ed indagare la tecnica di impatto cinetico per la deviazione delle traiettorie degli asteroidi nello spazio in un'ottica di possibile collisione tra uno di essi ed il pianeta Terra in futuro.

HERA, il satellite "madre" progettato dall'ESA in collaborazione con altri istituti di ricerca europei, avrà come obiettivo quello di raccogliere dati sulla composizione chimico-fisica del sistema binario e sulle caratteristiche dell'impatto tra DART, il satellite-proiettile realizzato e gestito dalla NASA

Sommario

assieme ad altri istituti di ricerca statunitensi, e il minore dei due corpi celesti che formano Didymos, che dovrebbe avvenire all'incirca nell'Ottobre del 2022.

Il satellite "madre" HERA trasporterà a bordo tecnologia di alto livello, tra cui alcuni CubeSats che verranno rilasciati quando HERA si troverà già in orbita nel sistema Didymos.

In questo panorama rientra anche la missione DustCube, una proposta di progetto di un CubeSat realizzato dal Dipartimento di Ingegneria Industriale con sede a Forlì (IT) dell'Università di Bologna in collaborazione con altri istituti accademici e di ricerca dell'Unione Europea, che ha come obiettivo principale quello di assistere HERA nell'acquisizione dei dati relativi all'impatto tra DART ed il minore dei due corpi celesti che compongono Didymos.

La missione DustCube prevede la necessità di operare in completa autonomia, comunicando in modo continuo i dati raccolti ad HERA, che provvederà all'invio degli stessi ai centri operativi terrestri.

La conformazione fisica del sistema binario e la tipologia di missione, che richiede l'utilizzo di fotocamere a raggi infrarossi, puntate sui due corpi celesti che formano Didymos, che scattino immagini degli stessi in modo continuo, portano a prediligere lo sviluppo di un filtro di navigazione ottica in grado di stimare la posizione e la velocità del CubeSat partendo da misure di direzione e di distanza rispetto ad un corpo-bersaglio ottenute per l'appunto tramite tecnologia di tipo ottico.

Questo progetto di tesi, che continua il lavoro svolto durante il mio tirocinio curriculare nel Laboratorio di Microsatelliti dell'Università di Bologna con sede a Forlì (IT), si inserisce nel progetto DustCube con l'obiettivo di indagare approfonditamente la navigazione orbitale di questo CubeSat immerso nel sistema Didymos e quello principale di sviluppare un filtro di Kalman di tipo esteso basato su misurazioni ottiche che permetta di simulare la navigazione in tempo reale di DustCube.

CONTENTS

LIST OF FIGURES	X
LIST OF TABLES	XII
LEGEND	XII
ACRONYMS	XIII
INTRODUCTION	1
1 DUSTCUBE'S MAIN FEATURES	7
1.1 Operational environment: Didymos system.....	7
1.2 DustCube's dynamical model	10
1.2.1 Reference Frames	11
1.2.2 The Circular Restricted 3-Body Problem (CR3BP) for DustCube.....	12
1.2.3 Gravitational attractions acted by Didymain and Didymoon	13
1.2.3 The Solar Radiation Pressure (SRP).....	23
1.2.5 Sun's gravitational attraction.....	31
2 DUSTCUBE'S ORBIT PROPAGATOR.....	33
2.1 The Cauchy problem for DustCube's propagator	33
2.2 The Runge-Kutta 4 th order method	35
2.3 The SPICE Toolkit	36
2.3.1 Spice Kernels for DustCube's orbit propagator.....	37
2.3.2 SPICE's useful routines	38
2.4 Implementation of the orbit propagator	39
2.4.1 The Didymos' Synodic Plane	39
2.4.2 The transformation in synodic coordinates.....	41
2.4.3 The orbital propagator's scripts	42
2.4.4 The orbital propagator's testing.....	49
3 OPTICAL NAVIGATION	56
3.1 Inertial direction evaluation: Lines of sight (LOS).....	56

Contents

3.2 Attitude evaluation	58
3.3 Image acquiring process.....	59
3.4 Image processing techniques	62
3.5 Optical Navigation concept for DustCube mission.....	68
4 KALMAN FILTERING	69
4.1 optimal controlling and filtering of dynamic systems	69
4.2 Kalman filtering for linear continuous dynamic systems	74
4.3 Kalman filtering for linear discrete dynamic systems	77
4.4 Linearized Kalman Filter (LKF).....	81
4.5 Extended Kalman Filter (EKF).....	84
5 DUSTCUBE'S FILTER.....	87
5.1 The non-linear Dynamic System of DustCube	87
5.1.1 The state vector function and its jacobian matrix	88
5.1.2 The measurement vector function and its jacobian matrix.....	90
5.1.3 The useful reference frames for the measurements	94
5.1.4 The process and the measurement errors.....	96
5.1.5 The covariance error matrices Q_{sat} and R_{sat}	102
5.2 The Extended Kalman Filter implementation	104
5.3 Initial validation of the DustCube's filter	106
5.4 The filtering of the corrupted state of DustCube	111
CONCLUSIONS.....	117
APPENDIX.....	119
A.1 Equations of motions.....	119
A.1.1 Inertial Reference Frame	119
A.1.2 Rotating Reference Frame	120
A.1.3 Translating and rotating Reference Frame.....	123
A.1.4 Rotation matrices	125
A.1.5 Transformations in non-inertial coordinates by using rotation matrices	128

A.2 The Circular Restricted 3-Body Problem (CR3BP)..... 130

A.3 General Cauchy problem..... 136

A.4 Statistics for random vectors 136

 A.4.1 Expected Value: 136

 A.4.2 Covariance Matrix: 137

 A.4.3 Correlation Matrix: 137

 A.4.4 Cross-correlation Matrix:..... 137

 A.4.5 Cross-covariance Matrix:..... 138

 A.4.6 Gaussian vectors: 138

REFERENCES..... 140

LIST OF FIGURES

Figure i.1: AIDA mission concept infographic (<i>European Space Agency (ESA)</i>).....	1
Figure i.2: Didymos orbit inside Solar System (<i>Jet Propulsion Laboratory (JPL), NASA</i>)	2
Figure i.3: DART Mission Concept (<i>National Aeronautics and Space Administration (NASA)</i>)	3
Figure i.4: HERA (AIM) Mission Concept infographic (<i>Credits: ESA/Science Office</i>).....	4
Figure i.5: DustCube CONOPS within Didymos system (<i>Pérez, et al., 2018 (In Press)</i>)	5
Figure 1.1: Didymos binary asteroid system (<i>Yu, Michel, Schwartz, Naidu, & Benner, 2017</i>)	9
Figure 1.2: The contribution of an infinitesimal element to the gravitational potential	14
Figure 1.3: General Legendre functions for $1 \leq l \leq 6$ and $0 \leq m \leq l$	15
Figure 1.4: Spherical harmonics with $l = 6$ in lateral and prospective views.....	16
Figure 1.5: Legendre functions for zonal (a), sectorial (b) and tesseral (c) harmonics	17
Figure 1.6: Didymos and Earth orbits from 11/20/202 (Christian, 2015)1 to 02/20/2023.....	23
Figure 1.7: Principles of an illuminated flat plate in prospective view and in top one.....	24
Figure 1.8: Solar flux impinging on DustCube from 2022-06-20 to 2022-11-20.....	25
Figure 1.9: Conical shadow model (<i>Montenbruck & Gill, Satellite Orbits, 2000</i>)	27
Figure 1.10: Illustration of apparent radii and apparent separation	28
Figure 1.11: Occultation of the solar disk caused by a spherical body	28
Figure 1.12: Occultation of the Sun caused by two spherical bodies	30
Figure 2.1: Didymos' Synodic Plane.....	40
Figure 2.2: DustCube's parking orbit in L5 from 2022-08-20 to 2022-08-25, considering Didymain and Didymoon as point masses and the SRP contribution.....	44
Figure 2.3: The solar flux which impinges on DustCube from 2022-08-20 to 2022-08-25	45
Figure 2.4: Shadow functions of Didymain and Didymoon from 2022-08-20 to 2022-08-25	45
Figure 2.5: Plot of two similar parking orbits in L5 obtained through the MATLAB propagator (blue line) and the Monte Python one (red line), both subjected to central attractions of the primaries and solar gravity.....	51
Figure 2.6: Absolute error trends for the two orbits of Figure 2.5	52
Figure 2.7: Relative percentage error trends for the two orbits of Figure 2.5.....	52
Figure 2.8: Max absolute, relative and relative percentage errors for the orbits of Figure 2.5.....	53
Figure 2.9: Plot of two similar parking orbits in L5, obtained by the MATLAB propagator (blue) and the Monte-Python one (red), due to central attractions of the primaries, solar gravity and SRP w/o sunlight occultations	54
Figure 2.10: Max absolute, relative and relative percentage errors for the orbits of Figure 2.9.....	55
Figure 3.1: Principle of 3D localization using two beacons LOS measurement (Polle, et al.).....	58

Figure 3.2: Principles of ideal pinhole camera. \mathbf{a} is the gnomonic projection for a star (<i>Owen, Methods of Optical Navigation, 2011, February 14</i>).....	58
Figure 3.3: Principles of emission and detection of light.....	61
Figure 3.4: LOS measurement acquisition using MTI.....	63
Figure 3.5: LOS measurement like stars in the background	64
Figure 3.6: CoM achievement using CoB determination technique	64
Figure 3.7: CoM achievement using analytic function fitting technique.....	65
Figure 3.8: Limb measurement principle (<i>Polle, et al., 2005, October 17-20</i>).....	65
Figure 3.9: Target viewed as an ellipsoid.....	66
Figure 3.10: Scanning process	67
Figure 4.1: Schematic representation of LQR.....	72
Figure 4.2: Kalman Filter's block diagram	80
Figure 4.3: Estimated trajectory of a S/C by using an LKF (<i>Montenbruck & Gill, Satellite Orbits</i>).....	83
Figure 4.4: Extended Kalman Filter's block diagram.....	85
Figure 4.5: Estimated trajectory of a S/C by using EKF (<i>Montenbruck & Gill, Satellite Orbits</i>)....	85
Figure 5.1: Tracking and ranging measurements of a target body	91
Figure 5.2: Representation of the longitude λ and the colatitude δ of a generic position vector \mathbf{r} ...	92
Figure 5.3: Body Reference Frame for DustCube (<i>Aguado, et al., 2016, May 6</i>)	94
Figure 5.4: Proposed cameras' configuration (<i>Aguado, et al., 2016, May 6</i>)	95
Figure 5.5: Small deviation errors from the CoM's pointing of the target body	99
Figure 5.6: Corrupted target pointing of a camera mounted onboard a satellite.....	100
Figure 5.7: Estimated state's error on x	108
Figure 5.8: Estimated state's error on y	108
Figure 5.9: Estimated state's error on z.....	109
Figure 5.10: Estimated state's error on V_x	109
Figure 5.11: Estimated state's error on V_y	110
Figure 5.12: Estimated state's error on V_z	110
Figure 5.13: Final estimated state's error on x.....	112
Figure 5.14: Final estimated state's error on y.....	113
Figure 5.15: Final estimated state's error on z.....	113
Figure 5.16: Final estimated state's error on V_x	114
Figure 5.17: Final estimated state's error on V_y	114
Figure 5.18: Final estimated state's error on V_z	115
Figure 5.19: Exact solution vs. filtered one in synodic coordinates	115
Figure 5.20: Exact solution vs. filtered one in synodic coordinates (zoom).....	116

Contents

Figure A.1: The position vector \mathbf{r} in an Inertial Reference Frame.....	120
Figure A.2: The position vector \mathbf{r} w.r.t. an Inertial RF and a rotating RF	121
Figure A.3: The position vector \mathbf{r} w.r.t. an Inertial RF and a roto-translating RF	124
Figure A.4: The generic 3-Body Problem (3BP).....	131
Figure A.5: The Circular Restricted 3-Body Problem (CR3BP).....	132
Figure A.6: Lagrange points in the Synodic Reference Frame	134
Figure A.7: Probability density function of a Gaussian random variable with zero mean	139

LIST OF TABLES

Table 1.1: Physical characteristics of Didymos binary asteroid.....	8
Table 1.2: Final proposed binary orbit solution for Didymos system	9
Table 1.3: Unnormalized exterior spherical harmonic coefficients of Didymain (<i>Takahashi</i>)	22
Table 1.4: Unnormalized exterior spherical harmonic coefficients of Didymoon	23
Table 2.1: The computed Lagrange equilibrium points of Didymos binary system	40
Table 4.1: Kalman Filter algorithm	81
Table 4.2: Linearized Kalman Filter algorithm	83
Table 4.3: Extended Kalman Filter algorithm	86

LEGEND

- The vectors are written in columns for convenience.
- Symbols in Cambria Math with bold, italic and lowercase characters are VECTORS: \mathbf{a} .
- Symbols in Cambria Math with bold, italic and uppercase characters are MATRICES: \mathbf{A} .
- Symbols in Cambria Math with no bold characters (lowercase/uppercase) are SCALARS: a, A .
- The words written in [blue characters](#) retrieve to other sections of the document.
The digital format of this thesis allows to click on these words to reconnect with the cited section.
- The words written in [blue and underlined characters](#) recall internet links. The digital format of this thesis allows the lecturer to click on these words to recover the cited site.
- The words written in *grey and italic characters between parenthesis* recall citations, figures, equations or demonstrations drawn from references available in bibliography section.
- The words written in *italic and bold characters with extensions* retrieve to some files.

ACRONYMS

ACRONYM	DESCRIPTION
3BP	3-Body Problem
AIDA	Asteroid Impact and Deflection Mission
AIM	Asteroid Impact Mission
ARE	Algebraic Riccati Equation
CoB	Center of Brightness
CoM	Center of Mass
CONOPS	Concept of Operations
COPINS	CubeSat Opportunity Payloads
CR3BP	Circular Restricted 3-Body Problem
CSG	Centre Spatial Guyanais
DART	Double Asteroid Redirection Test
DLR	Deutsches Zentrum für Luft- und Raumfahrt (German Aerospace Center)
DN	Digital Number
DRE	Differential Riccati Equation
DRO	Distant Retrograde Orbit
DYD	Didymos Reference Frame
ECI	Earth-centered Inertial Reference Frame
EKF	Extended Kalman Filter
ESA	European Space Agency
ET	Ephemeris Time
FoV	Field of View
GRC	Glenn Research Center
GSFC	Goddard Space Flight Center
i.e.	id est
iff	if and only if
INH	In-situ Nephelometer
IR	Infrared Radiation
ISL	Inter-Satellite (radio network) Link
ITRF	International Terrestrial Reference Frame
IVP	Initial Value Problem

Acronyms

JHU/APL	Johns Hopkins University Applied Physics Laboratory
JPL	Jet Propulsion Laboratory
JSC	Johnson Space Center
LKF	Linearized Kalman Filter
LOS	Line of Sight
LQE	Linear Quadratic Estimator
LQR	Linear Quadratic Regulator
LTI	Linear Time-Invariant
LTV	Linear Time-Variant
MTI	Multiple Time Integration
NAIF	Navigation and Ancillary Information Facility
NAIF	Navigation and Ancillary Information Facility
NASA	National Aeronautics and Space Administration
NEXT-C	NASA Evolutionary Xenon Thruster - Commercial
OCA	Observatoire de la Côte d'Azur
ODE	Ordinary Differential Equation
OPNAV	Optical Navigation
Pdf	Probability Density Function
RF	Reference Frame
RK4	Runge-Kutta 4 th order method
RNH	Remote Nephelometer
S/C	Spacecraft
SRP	Solar Radiation Pressure
SSB	Solar System Barycentre
SYN	Synodic Reference Frame
TBD	To Be Defined
TDB	Barycentric Dynamical Time
TOF	Time of Flight
UNIBO	University of Bologna
UVIGO	University of Vigo
w.r.t.	with respect to
w/o	without

INTRODUCTION

The University of Bologna takes part to AIDA (Asteroid Impact & Deflection Assessment) mission, that is divided in two main independent missions and, consequently, in two main large spacecrafts:

- DART (Double Asteroid Redirection Test), that will be the impactor S/C and it is directed by NASA to the Johns Hopkins University Applied Physics Laboratory (JHU/APL) with support of Jet Propulsion Laboratory (JPL), Goddard Space Flight Center (GSFC), Johnson Space Center (JSC) and other institutions and laboratories;
- HERA, named like the Greek goddess of marriage, that will be the observatory S/C. It has substituted AIM and is directed by ESA with support of German Aerospace Center (DLR), Observatoire de la Côte d'Azur (OCA) and other institutions and laboratories.

The main objective of the AIDA mission ([Figure i.1](#)) is to investigate the kinetic impact technique to change the motion of an asteroid in space, which is the selected near-Earth binary system 65803 Didymos ([Figure i.2](#)), composed by two celestial bodies, called Didymain and Didymoon.

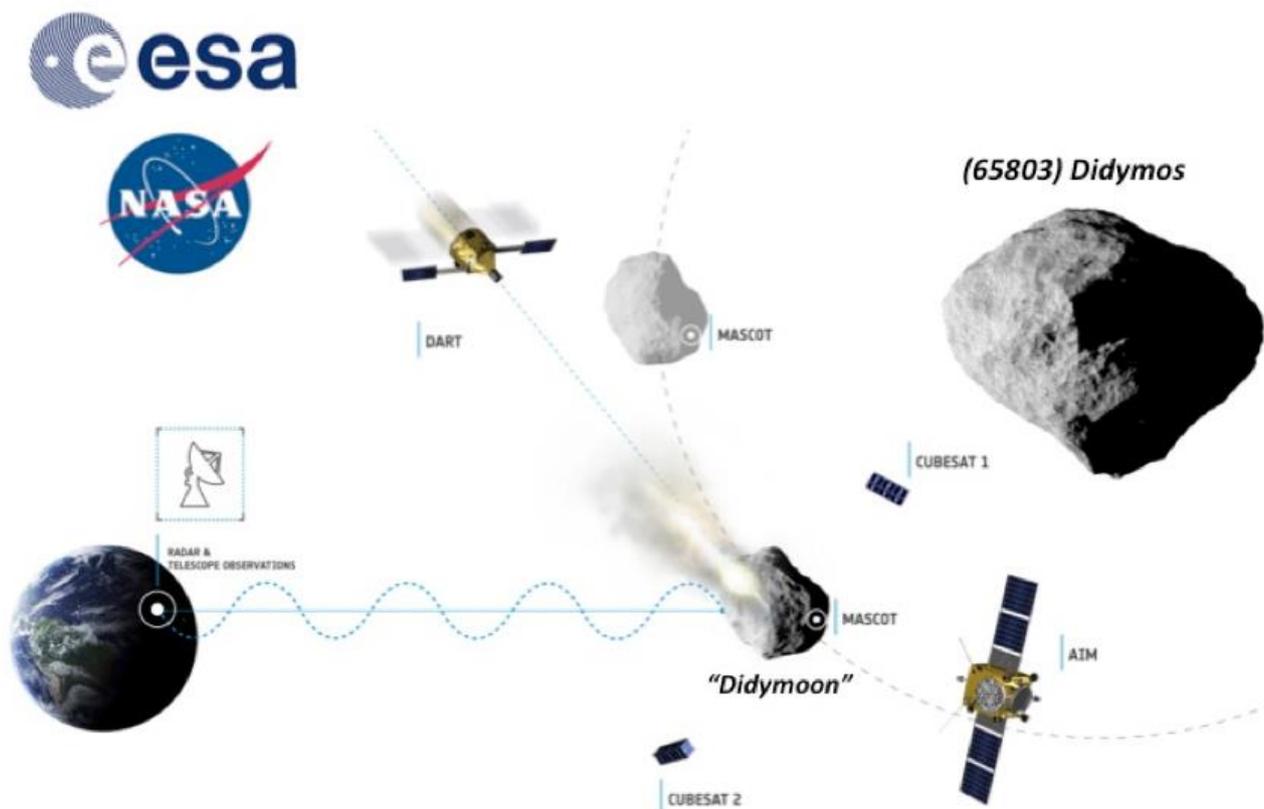


Figure i.1: AIDA mission concept infographic (*European Space Agency (ESA)*)

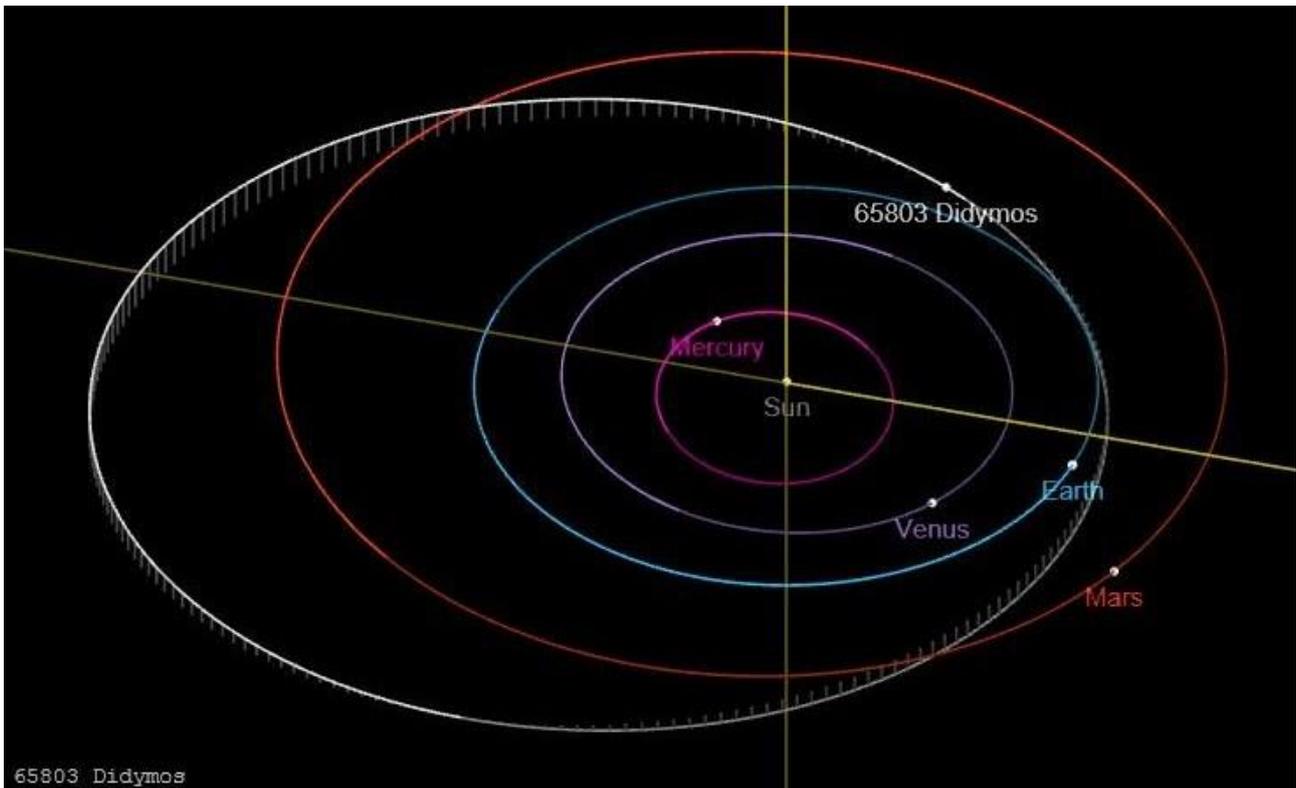


Figure i.2: Didymos orbit inside Solar System (*Jet Propulsion Laboratory (JPL), NASA*)

Obviously, the goal will be accomplished by divided the tasks into the two independent missions. Therefore, the main objectives of DART mission are:

1. to test the technologies developed to accomplish a rendezvous with a binary asteroid system, such as autonomous navigation and targeting, reducing the key risks;
2. to demonstrate the kinetic impact technique to change the motion of realistic scale asteroid, crashing into Didymoon with proper angle and velocity, in order to look for possible solutions to deflect asteroids in future events of close encounters with the Earth;
3. to improve impact models by comparing the motion of Didymoon in post-impact phase to one predicted in pre-impact phase;
4. to refine CONOPS for deflection missions.

Consequently, the main goals of the HERA mission are:

1. to acquire data about the deflection of Didymoon and about the cloud of dust generated by its collision with DART spacecraft;
2. to investigate the deep-space optical communication technology and the inter-satellite networking between CubeSats and a lander;

3. to study the interior structure of an asteroid;
4. to investigate the formation of binary asteroid systems and, more in general, the formation processes of our Solar System.

DART is considered as the first test in a Planetary Defense Technology Demonstration Plan.

It will be launched in January 2021, it will escape from the Earth's sphere of influence in August 2021 and will crash into Didymos in October 2022 (see [Figure i.3](#)).

For the launch window, DART will utilize the NEXT-C thruster, that is the next generation system based on the Dawn spacecraft propulsion system, developed by Glenn Research Center (GRC) in Cleveland, Ohio, USA.

NEXT-C will exploit solar electric propulsion system as its primary in-space propulsion one, so that it will be able to obtain significant flexibility for the mission timeline, to extend the launch window and to decrease the cost of the launch vehicle.

DART will intercept Didymos with the aim of an onboard optical camera and an autonomous navigation software and it will crash into the asteroid at an approximated velocity of 6 km/s, changing its motion after collision.

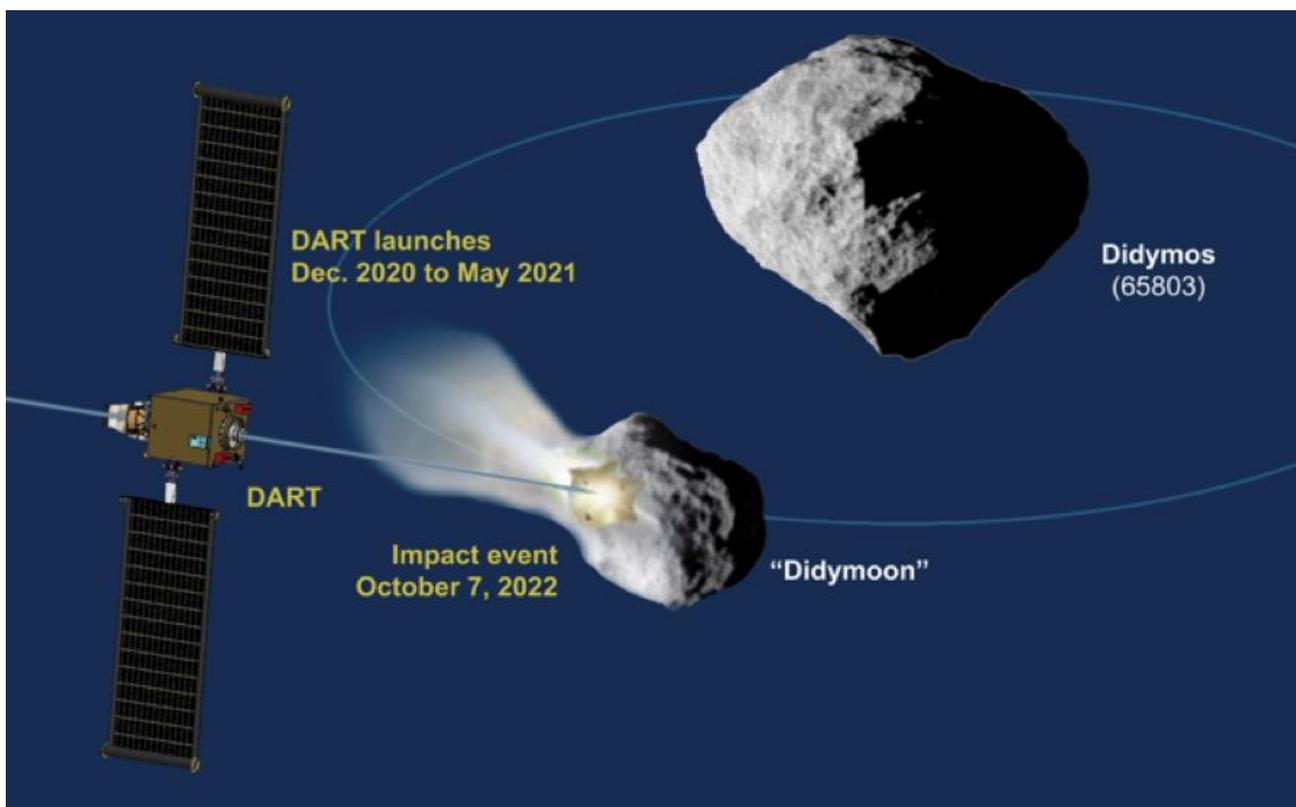


Figure i.3: DART Mission Concept (*National Aeronautics and Space Administration (NASA)*)

Introduction

The HERA spacecraft, which was named AIM (acronyms of Asteroid Impact Mission) in the first developing steps, will be launched in October 2020 from the Guiana Space Center (CSG) situated in Kourou, a city of the French Guiana, using an Ariane 6.2 rocket, so that it should enter in the gravity field of Didymos in May 2022 (Figure i.4).

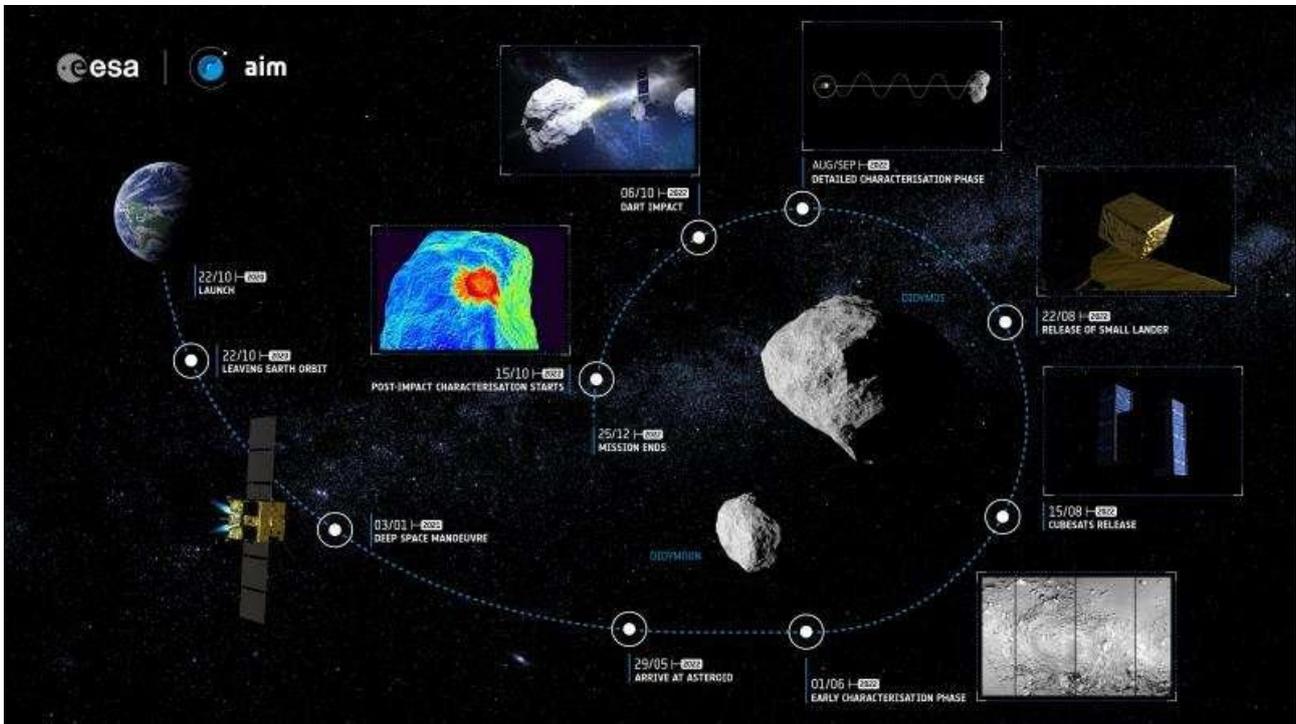


Figure i.4: HERA (AIM) Mission Concept infographic (Credits: ESA/Science Office)

HERA is planned to carry at least three smaller spacecrafts: Mascot-2 asteroid lander, developed by DLR, and two or more CubeSat Opportunity Payloads (COPINS).

Mascot-2 will be released by HERA in August 2022 and will be directed towards Didymoon, where its landing is expected and where it will establish a radar communication channel with HERA.

It will emit low frequency radar waves that will pass through Didymoon from side to side, before reaching HERA, to chart asteroid deep interior structure in pre-impact phase and to determine the variations in structure and surface of Didymoon in the post-impact phase.

The CubeSats will be released in August 2022 too and they will establish inter-satellite radio network through triangulation technique with HERA and Mascot-2.

HERA and the CubeSats will monitor the impact of DART, acquiring data about the change of velocity of Didymoon and its angle of deflection, together with the Earth's observatories.

After the collision, the emitted cloud of dust will be analysed by HERA using thermal images, obtaining important info about the type of debris ejected, their amount, the dimensions of the cloud, its ultimate distance reached and the shape and dimensions of the crater.

The collected data will be continuously sent to the Earth's observatories by using high-resolution laser communication.

The Department of Industrial Engineering of University of Bologna (IT) sited in Forlì, in collaboration with the Department of Telecommunication Engineering of the University of Vigo (ES) and the MICOS Engineering GmbH of Dübendorf (CH), developed the DustCube mission concept (Figure i.5), one of the five proposals that were selected by ESA for further study, that has the objective to design a 3U CubeSat platform with the main goal to measure the size, shape and concentration of fine dust ejected in the aftermath of the collision and its evolution over time, acquiring speeds of dust particles.

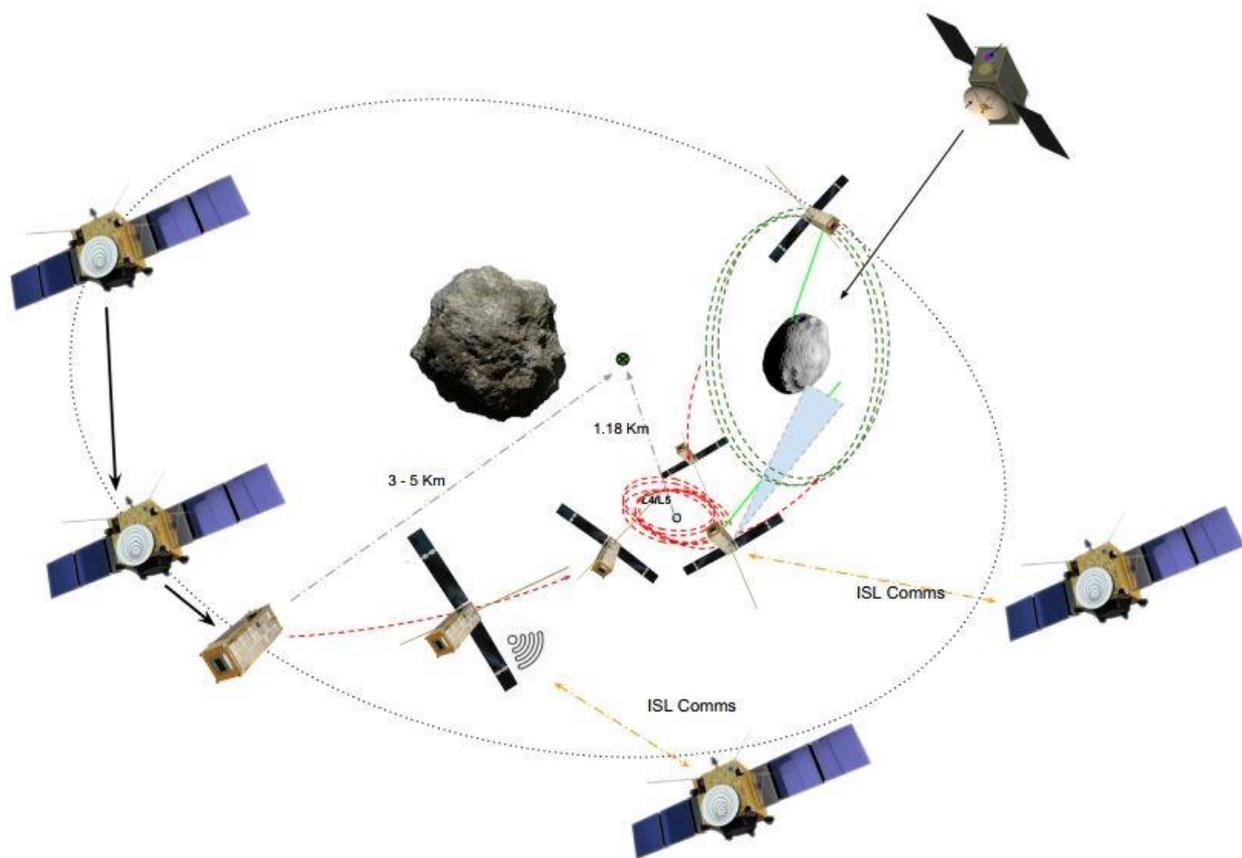


Figure i.5: DustCube CONOPS within Didymos system (Pérez, et al., 2018 (In Press))

Also, DustCube will help HERA to acquire data about physical characteristics, shape and quantity of the plume ejected during DART's impact by exploiting two light scattering Nephelometer, the remote one (RNH) and the in-situ one (INH), and it will test in orbit a new technique of laser altimetry, using the RNH for time of flight (TOF) measurement (Pérez, et al., 2018 (In Press)).

It will be released by HERA in August 2022 and since it will operate autonomously after the ejection, it will have to be independent for the navigation phases and for the acquisition of data ones, therefore an optical navigation system has been selected: a double IR camera configuration captures

Introduction

simultaneous images of the two celestial bodies to perform triangulation and to determine the relative position of the S/C w.r.t. them.

The objective of my work of thesis is to develop an optical navigation filter simulator for DustCube in MATLAB programming, starting from a proper propagation of its state.

The thesis is organised as follows:

- [Chapter 1](#) gives a more in-depth overview of DustCube mission, mainly the features of Didymos asteroid and the implemented dynamic model, which considers several levels of disturbances acted on DustCube;
- [Chapter 2](#) presents the DustCube's state propagator developed through MATLAB programming;
- [Chapter 3](#) describes the optical navigation technique, such as the link between optical measurements and the state/attitude of a S/C, the image acquiring and processing techniques and the selected concept for the optical navigation of DustCube;
- [Chapter 4](#) deepens the Kalman filtering technique, that will be the selected filter for DustCube because it is the most common in use for space navigation, since it is robust, reliable and it does not request great computational effort;
- [Chapter 5](#) describes the Kalman filter developed for DustCube mission.

1 DUSTCUBE'S MAIN FEATURES

DustCube will operate in Didymos binary asteroid system with the aim of an autonomous optical navigation system, i.e. a double IR camera configuration that will capture simultaneous images of Didymain and Didymoon.

Its CONOPS (Figure i.5) is composed by five operational phases (*Pérez, et al., 2018 (In Press)*):

1. **Birth phase**: DustCube will be released by HERA on August 2022;
2. **Injection phase**: DustCube will be inserted in a stable orbit at about 3-5 km from the CoM of the system, parking the S/C in a safe region for a time window of 7-14 days to permit early operations, such as the activation of the inter-satellite radio network link (ISL) with the mothership HERA;
3. **Pre-impact phase**: after the injection phase, DustCube will be transferred to a parking stable orbit around the L_4 or L_5 *Lagrange equilibrium point* (see Figure A.6 in Appendix) of Didymos binary system, where the S/C will activate the payload operations and will analyse the natural composition and physical characteristics of Didymoon before its collision with DART;
4. **Impact phase**: during and just after the impact of DART, DustCube will obtain data about the crater, the plume and dust generated by the collision, by using an In-situ Nephelometer (INH) and a Remote Nephelometer (RNH).

The acquisition phase will continue for a maximum of 4 days.

After that, DustCube will rendezvous with Didymoon from the parking orbit (L_4 or L_5) to the Distant Retrograde Orbit (DRO), an orbit at low altitude around Didymoon, where the S/C will remain for 24 days to acquire high-resolution images and measurements;

5. **Post-impact phase**: DustCube will be transferred from the DRO to the parking orbit (L_4 or L_5) to accomplish post-impact operations until the end of the mission.

To complete the mission, a detailed knowledge of the operational environment of DustCube and, therefore, of the forces acting on it is crucial, since the real-time dynamic filter requires a precise computation of the state (eq. (A.1) in Appendix) of DustCube.

In this chapter will be presented the main features of DustCube mission.

1.1 OPERATIONAL ENVIRONMENT: DIDYMOS SYSTEM

As said before, the selected asteroid for AIDA mission is 65803 Didymos, a near-Earth binary asteroid discovered in 1996 by Spacewatch, a group of the University of Arizona's Lunar and Planetary

Chapter 1

Laboratory founded in 1980 with the purpose to explore the various populations of small bodies in the Solar System to study the statistics of asteroids and comets in order to investigate the dynamical evolution of the Solar System.

The choice is not casual: it is more convenient to test the kinetic impact technique in a binary system instead of an individual asteroid, moreover Didymos will pass at the distance of just 11 million kilometers from Earth in 2022, facilitating the communications between HERA and Earth's bases and the measurements carried out by Earth's observatories.

Didymos belongs to the Apollo asteroids family, which includes asteroids with semi-major axis greater than 1 AU and perihelion lower than 1.017 AU, such as the Apollo asteroid.

In [Table 1.1](#) are shown the major physical characteristics of the binary system.

Official name of asteroid	65803
Diameter of Didymain	780 m +/- 10%
Diameter of Didymoon	163 m +/- 18 m
Bulk density of Didymain and Didymoon (assumed equal)	2104 kg m ⁻³ +/- 30%
Approximated Didymoon dimensions	a _s = 103 m b _s = 79 m c _s = 66m
Distance from CoM of Didymain and CoM of Didymoon (a_{orb})	1180 m +40/-20 m
Total mass of system	5.278e11 kg +/- 0.54e11 kg
Mass ratio Didymoon/Didymain	0.0093 +/- 0.0013
Rotation period of the primary	2.26 h +/- 0.0001 h
Heliocentric eccentricity e	$e = 0.383752501 +/- 7.7e-9$
Heliocentric semimajor axis a	1.6444327821 +/- 9.8e-9 AU
Heliocentric inclination to the ecliptic i	3.4076499° +/- 2.4e-6°

Table 1.1: Physical characteristics of Didymos binary asteroid

An observation campaign performed by the Discovery Channel Telescope on 2015-04-13 favors the retrograde orbit solution, with a synchronous rotation of Didymoon around Didymain (*Aguado, et al., 2016, May 6*), i.e. Didymoon rotational period is the same as Didymoon orbital period.

It is possible to assume that Didymain spin pole is the same as orbital spin pole, although it is important to remark that observations indicate that 25% of near-Earth asteroid binaries have non-zero inclinations (*Scheirich & Pravec, 2009*).

In [Table 1.2](#) is shown the final proposed binary orbit solution, while in [Figure 1.1](#) is possible to visualize the schematic representation of the Didymos system, where it is easy to recognize the rotation axis of Didymain and the motion of revolution of Didymoon around it.

Pole solution	$\lambda=310^\circ, \beta=-84^\circ$
Obliquity to the heliocentric orbit	$171^\circ \pm 9^\circ$
Diameter ratio Didymain/Didymoon	0.21 ± 0.01
Didymoon orbital period T_{orb}	$11.920\text{h} +0.004/-0.006$
Didymoon orbital eccentricity e_{orb}	0.03
Didymoon orbital inclination i_{orb} (assumed)	0°
Obliquity of the primary principal axis w.r.t. the mutual orbital plane (assumed)	0°
Obliquity of Didymoon principal axis with respect to the mutual orbital plane (assumed)	0°

Table 1.2: Final proposed binary orbit solution for Didymos system

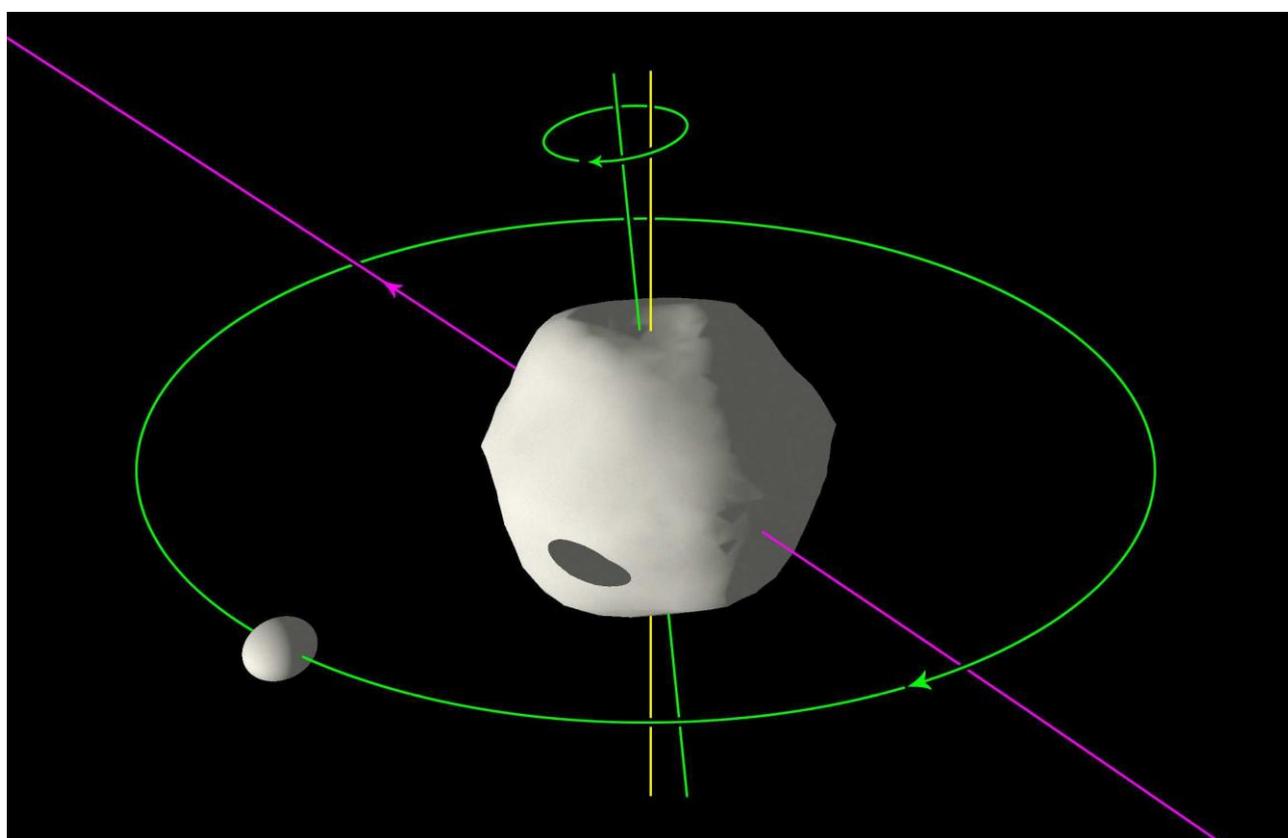


Figure 1.1: Didymos binary asteroid system (Yu, Michel, Schwartz, Naidu, & Benner, 2017)

The masses and, consequently, the standard *gravitational parameters* of Didymain (D subscript) and Didymoon (d subscript) can be computed from their diameter ratio, their mass ratio and the total mass of the system, assuming same and homogeneous density (Zannoni, et al., 2018), i.e.:

$$\begin{aligned} M_{\text{tot}} &= 5.278 \times 10^{11} \text{ kg} \\ M_{\text{D}} &= 5.229 \times 10^{11} \text{ kg} \end{aligned} \tag{1.1}$$

$$M_d = 4.866 \times 10^9 \text{ kg}$$

$$\mu_D = GM_D = 3.4903 \times 10^1 \text{ m}^3/\text{s}^2$$

$$\mu_d = GM_d = 3.23 \times 10^{-1} \text{ m}^3/\text{s}^2$$

Where G is the Universal Gravitation constant: $G = 6,67 \times 10^{-11} \text{ (N m}^2\text{)/kg}^2$.

Since the ratio between the masses of Didymoon and Didymain is equal to 0.93%, the CoM of the system is very close to the geometric center of Didymain, with an offset of about 10 meters.

The Didymos heliocentric orbit has been uploaded in my software by exploiting the Small Body Database Browser of JPL, which contains the ephemerides of Didymos in a wide time window that covers the entire phase of AIDA mission.

1.2 DUSTCUBE'S DYNAMICAL MODEL

The dynamical model of DustCube determines the evolution in time of its state (eq.(A.1) in Appendix), which is a vector of six components representing the position vector $\mathbf{r}(t)$ and the velocity vector $\mathbf{v}(t)$ w.r.t. the CoM of the exploited RF.

Since DustCube will be immersed in a binary system, it is possible to consider the estimation problem of the orbit of DustCube as a 3-Body Problem, i.e. the study of the behaviour of the dynamics of an isolated system composed by three punctual bodies of known masses moving due to the reciprocal gravitational attraction, perturbed by external forces that can be considered as disturbances for the isolated system.

Therefore, the most suitable solution is represented by the Circular Restricted 3-Body Problem (see section [A.2 The Circular Restricted 3-Body Problem \(CR3BP\)](#) in Appendix), which simplifies the general 3-Body Problem by assuming that one of the three bodies, the S/C, has a negligible mass, that is the same to recreate the actual situation in which the satellite does not affect the motion of the primaries, which revolve on a Keplerian orbit around their common CoM, so that the study of the dynamics is reduced to the analysis of the motion of the third body, DustCube, moving in a system composed by two massive bodies, Didymain as primary and Didymoon as secondary, with supposed known states.

This approximation is also valid for the masses of HERA, Mascot-2 and the other CubeSats, which can be considered as negligible w.r.t. the total mass of the system.

Since the masses of Didymain and Didymoon are insufficient to hold an atmosphere and to create a magnetosphere, we can compute the state of DustCube by considering a basic dynamical model due to the central attractions of Didymain and Didymoon, expanding and refining the solution step by step through the addition of the following contributions:

- gravitational attraction acted by Didymain by considering its spherical harmonics;
- gravitational attraction acted by Didymoon by considering its spherical harmonics;
- force generated by the Solar Radiation Pressure (SRP);
- gravitational disturbance of Sun.

Obviously, specific reference frames must be defined for the computation of the state of DustCube.

1.2.1 REFERENCE FRAMES

It is important to define proper reference frames to consistently compute the state of DustCube:

- *ECLIPJ2000 (EC subscript)*: is an inertial RF centered in SSB, with the x-axis directed towards the vernal equinox (intersection between the Earth's equatorial plane and the ecliptic one) at epoch J2000, that stands for 01/01/2000 at 12:00:00 TDB, the z-axis perpendicular to the mean ecliptic plane at epoch J2000 and the y-axis that completes the right-hand frame.

In alternative to *ECLIPJ2000*, is possible to use the *J2000* RF, an inertial RF centered in SSB, with the x-axis directed towards the vernal equinox at epoch J2000, the z-axis perpendicular to the Earth's mean equatorial plane at epoch J2000 and the y-axis that completes the right-hand frame;

- *Didymos Reference Frame (DYD subscript)*: is the RF linked to the CoM of the binary system with a fixed orientation in space, i.e. it can be considered as a quasi-inertial RF for a steep time-window, since it follows the CoM of the binary system w/o rotating w.r.t. the *ECLIPJ2000* RF. The x-y plane coincides with the mean orbital plane of the mutual orbit of Didymain and Didymoon, with the x-axis parallel to the line of nodes and directed towards the ascending node (N-axis), the z-axis directed as the first integral of motion (h-axis) and the y-axis which completes the right-hand frame.

- *Didymain quasi-inertial Reference Frame*: is the quasi-inertial RF linked to the Didymain's CoM, that can be considered as coincident with its geometric center, with a constant orientation in space and in time, which can be indifferently choose like the *ECLIPJ2000 RF*'s one or like the *Didymos RF*'s one, so that this frame simply follows the CoM of Didymain w/o rotations;

- *Synodic Reference Frame (SYN subscript)*: is the Synodic RF ([Figure A.6](#) in Appendix) related to Didymos, co-rotating with the binary system. It is centered in the CoM of Didymos, with the x-axis directed as the line of conjunction from Didymain to Didymoon, the z-axis directed as the first integral of motion (h-axis) and the y-axis that completes the right-hand frame.

This RF rotates w.r.t. to Didymos RF with an angular speed equal to the mean angular speed of the binary system.

Chapter 1

The x-y plane is the mean orbital plane of the mutual orbit of Didymain and Didymoon;

- *Didymain-fixed Reference Frame (DM subscript)*: is the body-fixed RF of Didymain, centered in its CoM, that coincides, for simplicity, with its geometric center. The x-y plane is parallel to one containing the mutual orbit of Didymain and Didymoon and the z-axis is directed as the first integral of motion (h-axis), since it can be considered as the rotation axis of Didymain too. This frame rotates around the z-axis with an angular speed equal to the mean rotating speed of the massive body. The x-axis is directed as the *ECLIPJ2000*'s x-axis at epoch J2000;
- *Didymoon-fixed Reference Frame (dm subscript)*: is the body-fixed RF of Didymoon, centered in its CoM, that is, for simplicity, its geometric center. The x-y plane is parallel to one containing the mutual orbit of Didymain and Didymoon and the z-axis is directed as the h-axis, since it can be considered as the rotation axis of Didymoon too. This frame rotates around z-axis with an angular speed equal to the mean rotating speed of the massive body, that is the mean orbiting speed of Didymoon around Didymain, since the binary system shows a synchronous rotation of Didymoon (*Aguado, et al., 2016, May 6*). The x-axis is directed as the *ECLIPJ2000*'s x-axis at epoch J2000.

1.2.2 THE CIRCULAR RESTRICTED 3-BODY PROBLEM (CR3BP) FOR DUSTCUBE

As said before, the first useful step is to compute the state of DustCube by exploiting the CR3BP (see section [A.2 The Circular Restricted 3-Body Problem \(CR3BP\)](#) in Appendix), i.e. by considering the mass of DustCube as negligible w.r.t. the masses of the primaries, which are approximated like punctual bodies with no shape.

The most useful Reference Frame to visualize the orbit of DustCube is the Synodic one ([Figure A.6](#)), which admits five equilibrium points, the so-called *Lagrange points* L_i ($i = 1, \dots, 5$), in which the velocity and the acceleration of the S/C are null if we consider only the gravitational attractions exerted by Didymain and Didymoon as point-masses.

Therefore, referring to [Didymos Reference Frame](#) and defining $\mathbf{r}_{\text{sat}_{\text{DYD}}}(\mathbf{t})$, $\mathbf{r}_{\text{D}_{\text{DYD}}}(\mathbf{t})$ and $\mathbf{r}_{\text{d}_{\text{DYD}}}(\mathbf{t})$ as the position vectors of DustCube, Didymain and Didymoon, respectively, w.r.t. the CoM of the binary system, we can compute the position vectors of the S/C w.r.t. Didymain and Didymoon:

$$\begin{aligned}\mathbf{r}_{\text{sat-D}}(\mathbf{t})|_{\text{DYD}} &\stackrel{\text{def}}{=} \mathbf{r}_{\text{sat}_{\text{DYD}}}(\mathbf{t}) - \mathbf{r}_{\text{D}_{\text{DYD}}}(\mathbf{t}) \\ \mathbf{r}_{\text{sat-d}}(\mathbf{t})|_{\text{DYD}} &\stackrel{\text{def}}{=} \mathbf{r}_{\text{sat}_{\text{DYD}}}(\mathbf{t}) - \mathbf{r}_{\text{d}_{\text{DYD}}}(\mathbf{t}),\end{aligned}\tag{1.2}$$

so that, it is possible to express the gravitational acceleration acting on DustCube by exploiting the Newton's Law of Universal Gravitation, i.e. by using the equation [\(A.55\)](#):

$$\mathbf{a}_{\text{sat}_{\text{DYD}}}(\mathbf{t}) = -\frac{\mu_{\text{D}}}{\|\mathbf{r}_{\text{sat-D}}(\mathbf{t})|_{\text{DYD}}\|^3}\mathbf{r}_{\text{sat-D}}(\mathbf{t})|_{\text{DYD}} - \frac{\mu_{\text{d}}}{\|\mathbf{r}_{\text{sat-d}}(\mathbf{t})|_{\text{DYD}}\|^3}\mathbf{r}_{\text{sat-d}}(\mathbf{t})|_{\text{DYD}} \quad (1.3)$$

Indeed, as said before, the Didymos RF can be considered as quasi-inertial, so that the equation (1.3) can be thought as the absolute acceleration of DustCube in inertial coordinates.

Once the solver has computed the state of DustCube in Didymos RF and has transformed it in synodic coordinates, he should obtain a result which satisfies the conditions of the *Lagrange points* of the Synodic RF (subscript SYN), i.e. $\mathbf{v}_{\text{sat}_{\text{SYN}}}(\mathbf{t}) = \mathbf{a}_{\text{sat}_{\text{SYN}}}(\mathbf{t}) = \mathbf{0}$, that means the S/C would remain steady w.r.t. the Synodic RF if it was parked in one of the five equilibrium points.

1.2.3 GRAVITATIONAL ATTRACTIONS ACTED BY DIDYMAIN AND DIDYMOON

The main contributions for the total punctual acceleration of DustCube in terms of orders of magnitude are, obviously, the gravitational attractions caused by Didymain and Didymoon.

In first approximation, is possible to simplify the primaries as point-masses, i.e. considering their masses as concentrated in their CoMs, to compute the DustCube's acceleration, which brings to the CR3BP, that has been explained in the previous section.

Proceeding with the refinement of the computed solutions, is necessary to account for the actual mass distributions, since the primaries are not punctual and have non-spherical shape.

Therefore, defining the position vector of the satellite w.r.t. the CoM of Didymain or Didymoon in body-fixed RF (b subscript) like $\mathbf{r}_{\text{sat}}(\mathbf{t})|_{\text{b}} = (x_{\text{b}}(\mathbf{t}), y_{\text{b}}(\mathbf{t}), z_{\text{b}}(\mathbf{t}))$, the acceleration of DustCube due to gravitational attractions of the primaries is derived by its gravitational potential $U_{\text{sat}_G}|_{\text{b}}$, i.e.:

$$\mathbf{a}_{\text{sat}_G}(\mathbf{t})|_{\text{b}} = \left. \frac{d^2\mathbf{r}_{\text{sat}}(\mathbf{t})}{dt^2} \right|_{\text{b}} = \nabla[U_{\text{sat}_G}(\mathbf{r}_{\text{sat}}(\mathbf{t}))]|_{\text{b}} = \left. \frac{\partial U_{\text{sat}_G}(\mathbf{r}_{\text{sat}}(\mathbf{t}))}{\partial \mathbf{r}(\mathbf{t})} \right|_{\text{b}} \quad (1.4)$$

In case of the bodies are considered as point-masses, applying the relation (1.4) for both the primaries in their body-fixed RFs and summing up the two contributions, the computed DustCube's acceleration simply becomes the already mentioned eq. (1.3), which represents the acceleration due to central attractions of the primaries.

Developing the state propagator for more precise computations, we must redefine the gravitational potential by considering the actual shapes and mass distributions of the primaries.

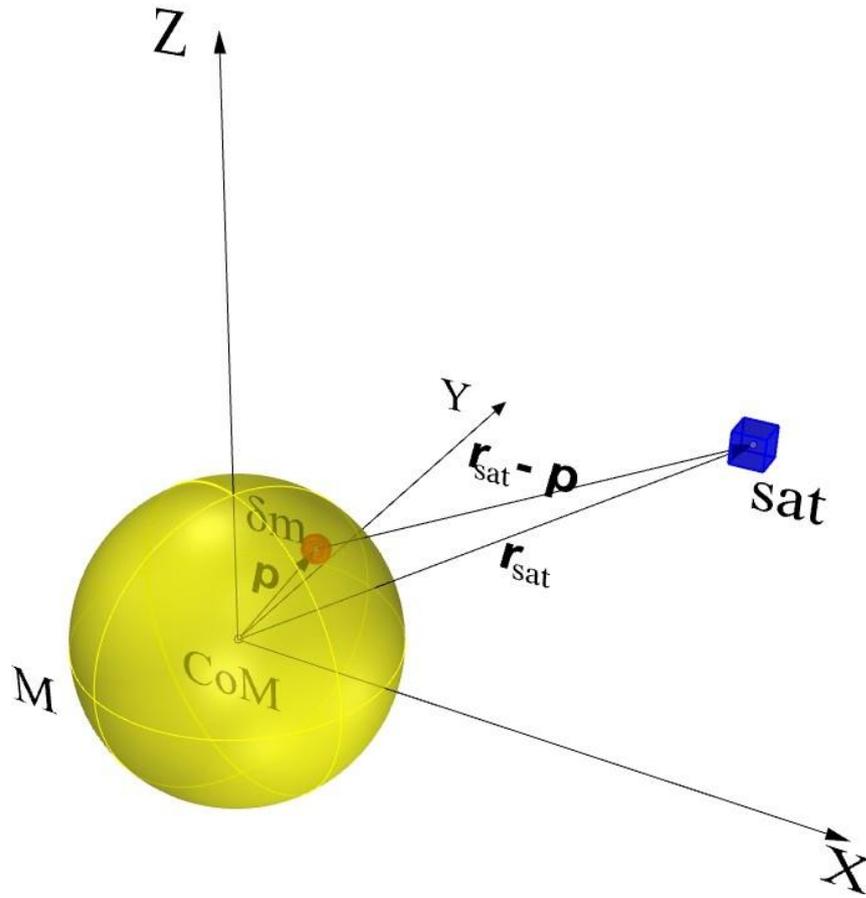


Figure 1.2: The contribution of an infinitesimal element to the gravitational potential

Indeed, starting from the [Figure 1.2](#), we can visualize the gravitational influence of every infinitesimal mass δm of a generic massive body on the satellite motion.

Defining $\mathbf{p}(t)|_b$ as the position vector of δm w.r.t. the body-fixed RF centered in the CoM of the celestial body, the gravitational potential due to a single primary can be computed by integrating the infinitesimal contributions on the entire domain, which is the body mass M :

$$U_{\text{satG}}(\mathbf{r}_{\text{sat}}(t))|_b = G \int \frac{\delta m}{\|\mathbf{r}_{\text{sat}}(t)|_b - \mathbf{p}(t)|_b\|}, \quad (1.5)$$

which requires an appropriate evaluation of the inverse of the distance for $r_{\text{sat}}(t) > p(t)$, i.e.:

$$\frac{1}{\|\mathbf{r}_{\text{sat}}(t)|_b - \mathbf{p}(t)|_b\|} = \frac{1}{r_{\text{sat}}(t)} \sum_{l=0}^{\infty} \left(\frac{p(t)}{r_{\text{sat}}(t)} \right)^l P_l[\cos(\gamma)], \quad (1.6)$$

where γ is the angle between $\mathbf{r}_{\text{sat}}(t)|_b$ and $\mathbf{p}(t)|_b$ and $P_l[\cos(\gamma)]$ is the Legendre polynomial or function of degree l for the specific function $\cos(\gamma)$.

The generic Legendre polynomial P_l of degree l for a scalar function u is defined like:

$$P_l[u] \stackrel{\text{def}}{=} \frac{1}{2^l l!} \frac{d^l}{du^l} (u^2 - 1)^l, \quad (1.7)$$

which can be used to obtain the associated Legendre polynomial $P_{lm}[u]$ of degree l and order m :

$$P_{lm}[u] \stackrel{\text{def}}{=} (1 - u^2)^{\frac{m}{2}} \frac{d^m}{du^m} P_l[u] = \frac{1}{2^l l!} (1 - u^2)^{\frac{m}{2}} \frac{d^{l+m}}{du^{l+m}} (u^2 - 1)^l \quad (1.8)$$

In [Figure 1.3](#) are shown the general Legendre functions for $1 \leq l \leq 6$ and $0 \leq m \leq l$.

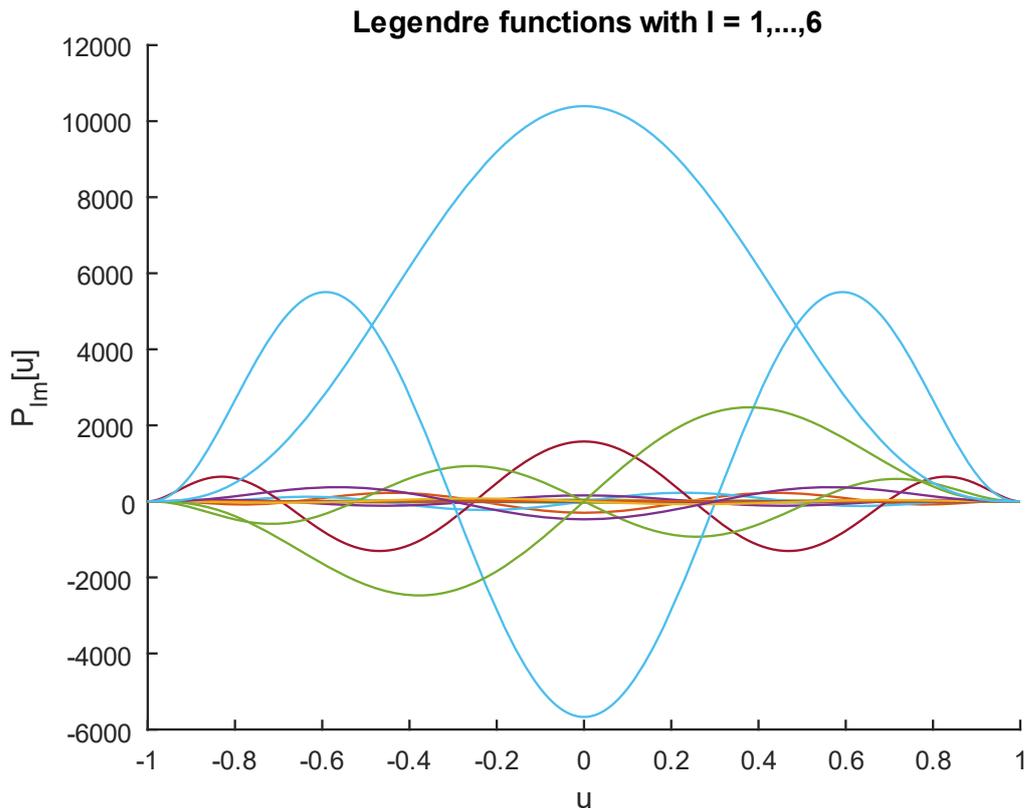


Figure 1.3: General Legendre functions for $1 \leq l \leq 6$ and $0 \leq m \leq l$

Introducing the planetocentric latitude and longitude of the satellite $\phi_{\text{sat}}(t)$ and $\lambda_{\text{sat}}(t)$, respectively, we can expand the generic Legendre polynomial $P_l[\cos(\gamma)]$ of eq. (1.6) by using some important properties, to finally obtain the gravitational potential of the satellite due to a single massive body in body-fixed RF as a series of Legendre polynomials (*Vallado & McClain, 2007, May 5*), (*Montenbruck & Gill, Satellite Orbits, 2000*):

$$U_{\text{satG}}|_{\text{b}} \stackrel{\text{def}}{=} \frac{GM}{r_{\text{sat}}(t)} \sum_{l=0}^{\infty} \sum_{m=0}^l \left(\frac{R_p}{r_{\text{sat}}(t)} \right)^l P_{l,m}[\sin(\phi_{\text{sat}}(t))] [C_{l,m} \cos(m\lambda_{\text{sat}}(t)) + S_{l,m} \sin(m\lambda_{\text{sat}}(t))], \quad (1.9)$$

where:

Chapter 1

- R_p is the mean equatorial radius of the massive body [m];
- $C_{l,m}$ and $S_{l,m}$ are unnormalized gravitational coefficients.

The Legendre polynomials $P_{l,m}[\sin(\phi_{\text{sat}}(t))]$ and the coefficients $C_{l,m}$ and $S_{l,m}$, which characterize the massive body, define the so-called spherical harmonics of degree l and order m , that can be grouped in three main classes:

- *Zonal harmonics*: $m = 0$ and $l \neq 0$;
- *Sectorial harmonics*: $l = m$;
- *Tesseral harmonics*: $l \neq m$, with $l \neq 0$ and $m \neq 0$.

The [Figure 1.4](#) displays some examples of spherical harmonics for a generic quasi-spherical massive body while the [Figure 1.5](#) shows the Legendre functions for zonal harmonics, sectorial harmonics and tesseral harmonics, for a maximum degree $l = 6$ and $u \in [-1,1]$.

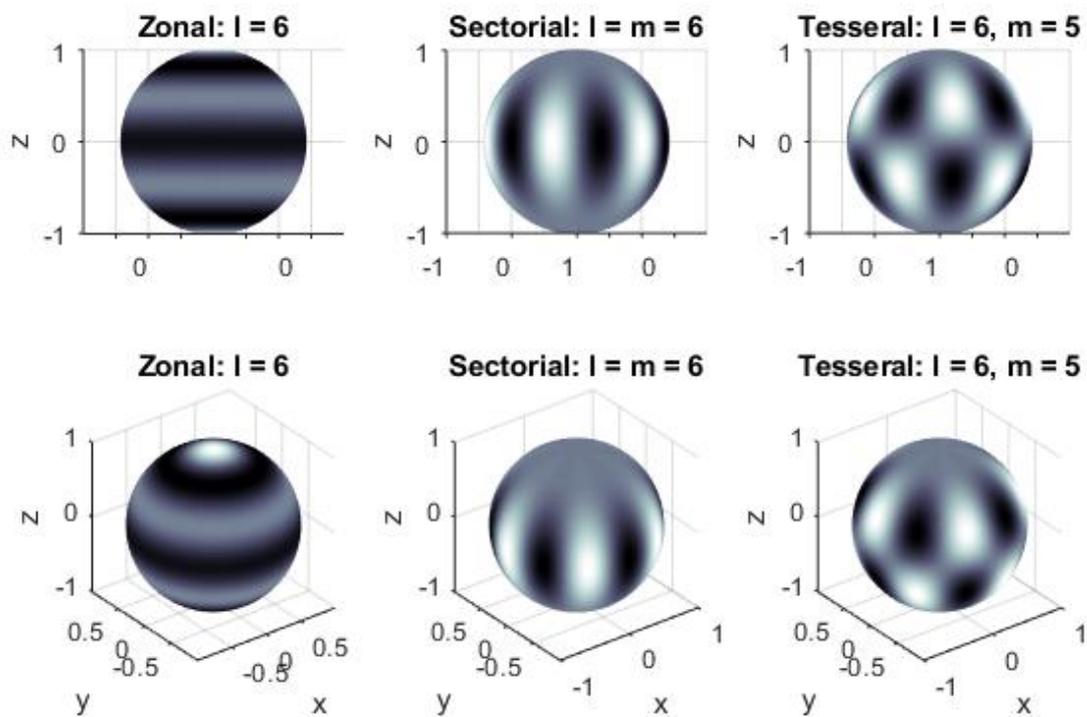


Figure 1.4: Spherical harmonics with $l = 6$ in lateral and prospective views

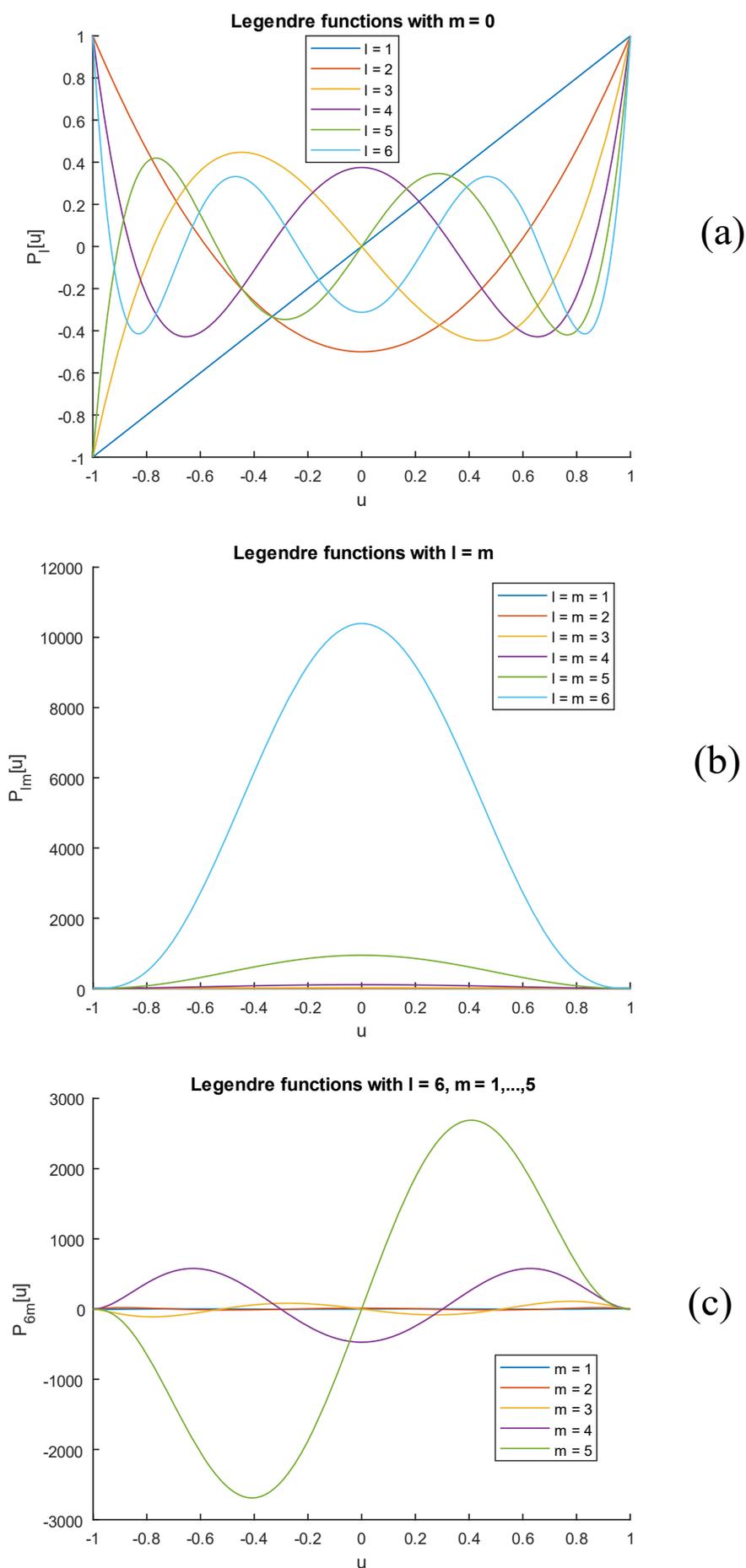


Figure 1.5: Legendre functions for zonal (a), sectorial (b) and tesseral (c) harmonics

Chapter 1

Finally, computing the gradient of the gravitational potential of eq. (1.9) in body-fixed coordinates, we obtain the acceleration of the S/C related to a single celestial body.

A useful and quite easy implementation of the accelerations due to gravitational attraction of non-spherical shaped celestial bodies is available in Montenbruck O., Gill E., *Satellite Orbits*, Springer, 3rd edition, 2005 (*Montenbruck & Gill, Satellite Orbits, 2000*), which considers the three components of acceleration vector due to spherical harmonics as the sum of partial accelerations computed for every harmonic, namely:

$$\ddot{x}_b(t) = \sum_{l,m} \ddot{x}_{l,m_b}(t) \quad \ddot{y}_b(t) = \sum_{l,m} \ddot{y}_{l,m_b}(t) \quad \ddot{z}_b(t) = \sum_{l,m} \ddot{z}_{l,m_b}(t) \quad (1.10)$$

Therefore, defining the useful scalar coefficients $V_{l,m}(t)$ and $W_{l,m}(t)$ as:

$$\begin{aligned} V_{l,m}(t)|_b &\stackrel{\text{def}}{=} \left(\frac{R_p}{r(t)}\right)^{l+1} P_{l,m}[\sin(\phi_{gc_{\text{sat}}}(t))] \cos(m\lambda_{\text{sat}}(t)) \\ W_{l,m}(t)|_b &\stackrel{\text{def}}{=} \left(\frac{R_p}{r(t)}\right)^{l+1} P_{l,m}[\sin(\phi_{gc_{\text{sat}}}(t))] \sin(m\lambda_{\text{sat}}(t)), \end{aligned} \quad (1.11)$$

is possible to simplify the gravitational potential of eq. (1.9):

$$U_{\text{sat}_G}(t)|_b = \frac{GM}{R_p} \sum_{l=0}^{\infty} \sum_{m=0}^l \left[C_{l,m} V_{l,m}(t)|_b + S_{l,m} W_{l,m}(t)|_b \right] \quad (1.12)$$

The scalar coefficients $V_{l,m}(t)$ and $W_{l,m}(t)$ of eqs (1.11) can be seemed difficult to compute, but since they follow recurrence relations (*Montenbruck & Gill, Satellite Orbits, 2000*), i.e.:

$$\begin{aligned} V_{m,m}(t)|_b &= (2m-1) \left[\frac{R_p x_b(t)}{r(t)^2} V_{m-1,m-1}(t) - \frac{R_p y_b(t)}{r(t)^2} W_{m-1,m-1}(t) \right] \\ W_{m,m}(t)|_b &= (2m-1) \left[\frac{R_p x_b(t)}{r(t)^2} W_{m-1,m-1}(t) + \frac{R_p y_b(t)}{r(t)^2} V_{m-1,m-1}(t) \right] \\ V_{l,m}(t)|_b &= \left(\frac{2l-1}{l-m}\right) \frac{R_p z_b(t)}{r(t)^2} V_{l-1,m}(t) - \left(\frac{l+m-1}{l-m}\right) \frac{R_p^2}{r(t)^2} V_{l-2,m}(t) \\ W_{l,m}(t)|_b &= \left(\frac{2l-1}{l-m}\right) \frac{R_p z_b(t)}{r(t)^2} W_{l-1,m}(t) - \left(\frac{l+m-1}{l-m}\right) \frac{R_p^2}{r(t)^2} W_{l-2,m}(t), \end{aligned} \quad (1.13)$$

they are easily computable with a recursive implementation, starting from the obvious values:

$$V_{0,0}(t) = \frac{R_p}{r(t)} \quad , \quad W_{0,0} = 0 \quad (1.14)$$

Finally, the partial accelerations in body-fixed coordinates become (omitting the time dependence and the body-fixed RF subscript of the scalar coefficients $V_{l,m}(t)$ and $W_{l,m}(t)$):

$$\begin{aligned} \ddot{x}_{l,0_b} &= \frac{GM}{R_p^2} (-C_{l,0} V_{l+1,1}) \\ \ddot{y}_{l,0_b} &= \frac{GM}{R_p^2} (-C_{l,0} W_{l+1,1}) \end{aligned} \quad (1.15)$$

$$\begin{aligned} \ddot{x}_{l,m_b} &= \frac{GM}{2R_p^2} \left[(-C_{l,m} V_{l+1,m+1} - S_{l,m} W_{l+1,m+1}) + \frac{(l-m+2)!}{(l-m)!} (C_{l,m} V_{l+1,m-1} + S_{l,m} W_{l+1,m-1}) \right] \\ \ddot{y}_{l,m_b} &= \frac{GM}{2R_p^2} \left[(-C_{l,m} W_{l+1,m+1} + S_{l,m} V_{l+1,m+1}) + \frac{(l-m+2)!}{(l-m)!} (-C_{l,m} W_{l+1,m-1} + S_{l,m} V_{l+1,m-1}) \right] \\ \ddot{z}_{l,m_b} &= \frac{GM}{R_p^2} [(l-m+1)(-C_{l,m} V_{l+1,m} - S_{l,m} W_{l+1,m})] \end{aligned}$$

As said before, these partial accelerations are computed in a body-fixed RF centered in the CoM of the celestial body, but the final acceleration must be written in an inertial RF.

So, we must transform the acceleration vector from the body-fixed RF to the inertial one by exploiting the proper rotational matrix and, eventually, the drift between the two CoMs.

For example, considering a S/C orbiting around the Earth, the partial accelerations are computed in ITRF, but the final acceleration vector must be transformed in ECI frame:

$$\mathbf{a}_{\text{ECI}}(t) = \mathbf{T}(t) \mathbf{a}_{\text{ITRF}}(t) \quad , \quad (1.16)$$

where $\mathbf{T}(t)$ is the transformation matrix between ITRF and ECI reference frames.

Obviously, the $C_{l,m}$ and $S_{l,m}$ unnormalized gravitational coefficients related to spherical harmonics must be known a-priori, since their computation requires difficult techniques to map the masses distribution of every celestial body considered.

Some publications prefer to adopt the normalized gravitational coefficients $\bar{C}_{l,m}$ and $\bar{S}_{l,m}$ instead of the unnormalized ones, since the first normally show lower changes in magnitude between a combination of degree l and order m and the subsequent one.

It is possible to easily shift from the unnormalized coefficients to the normalized ones, and vice versa, by exploiting the following equations (*Montenbruck & Gill, Satellite Orbits, 2000*):

$$\begin{cases} \bar{C}_{l,m} \\ \bar{S}_{l,m} \end{cases} = \sqrt{\frac{(l+m)!}{(2-\delta_{0m})(2l-1)(l-m)!}} \begin{cases} C_{l,m} \\ S_{l,m} \end{cases}, \quad (1.17)$$

where δ_{0m} is the Delta of Kronecker function.

The normalized associated Legendre functions are obtainable from the unnormalized ones too, i.e.:

$$\bar{P}_{l,m} = \sqrt{\frac{(2-\delta_{0m})(2l-1)(l-m)!}{(l+m)!}} P_{l,m}, \quad (1.18)$$

so that, it is easy to rewrite the gravitational potential of eq. (1.9) by using the normalized coefficients and Legendre functions instead of the unnormalized ones:

$$U_{\text{satG}}|_{\text{b}} \stackrel{\text{def}}{=} \frac{GM}{r_{\text{sat}}(t)} \sum_{l=0}^{\infty} \sum_{m=0}^l \left(\frac{R_p}{r_{\text{sat}}(t)} \right)^l \bar{P}_{l,m}[\sin(\phi_{\text{sat}}(t))] [\bar{C}_{l,m} \cos(m\lambda_{\text{sat}}(t)) + \bar{S}_{l,m} \sin(m\lambda_{\text{sat}}(t))], \quad (1.19)$$

Obviously, also the partial accelerations computed in eqs (1.15) must be rewritten by considering the normalized gravitational coefficients and Legendre functions.

Although the Montenbruck technique is quite easy to implement and is suitable in terms of computational effort, there exists a better solution to compute the accelerations due to gravitational spherical harmonics, which is the built-in MATLAB function *gravitysphericalharmonic*, which requires a greater effort to the computational machine than the Montenbruck technique, but it allows more precise solutions.

Since this function has been created to primarily extrapolate the accelerations due to well-known massive bodies in space, such as the Earth, the Moon and so on, for which there exists an excellent knowledge of their shapes and mass distributions, that are already contained inside MATLAB software, is possible to customize the computation process in case of particular celestial bodies by adding the main data available for the orbiting massive body.

Therefore, the calls:

- `[ax, ay, az] = gravitysphericalharmonic(r_rot,'custom',degree,{'Didymain.mat'}@load),'None');`
- `[ax, ay, az] = gravitysphericalharmonic(r_rot,'custom',degree,{'Didymoon.mat'}@load),'None');`

allow the user to compute the three components ax, ay, az of the gravitational accelerations due to spherical harmonics in the Planet-Centered Planet-Fixed reference frames by properly specifying the following fields:

- `r_rot` is the position vector of DustCube w.r.t. Didymain or Didymoon in planet coordinates;

- 'custom' suggests to MATLAB to consider the computation as personalized;
- degree is the maximum value l considered by the MATLAB function;
- {'Didymain.mat'@load} and {'Didymoon.mat'@load} permit to upload the main data of the celestial bodies Didymain and Didymoon inside the MATLAB function, which are contained in the MATLAB files *Didymain.mat* and *Didymoon.mat*, respectively;
- 'None' specifies the action in case of out of range input, which is not in our interest.

The MATLAB files *Didymain.mat* and *Didymoon.mat* hold binary data in form of matrices and scalars, which are necessary to properly compute the gravitational acceleration due to the spherical harmonics of the primaries of Didymos.

Every binary file contains the following data of a celestial body:

- the mass parameter $\mu = GM$ in $[m^3/s^2]$;
- the mean equatorial radius R_e in $[m]$;
- the maximum degree l_{max} of the spherical harmonics available for the massive body;
- the matrices C and L of dimensions $(l_{max} + 1) \times (l_{max} + 1)$ which hold the normalized exterior spherical harmonic coefficients $\bar{C}_{l,m}$ and $\bar{S}_{l,m}$ of the celestial body.

The Didymain's gravitational acceleration has been tested by using normalized coefficients up to degree 20 and order 20 computed for a homogeneous polyhedron of uniform density by Zannoni M., et al., *Radio science investigations with the Asteroid impact mission*, Adv. Space Res., 2018 (Zannoni, et al., 2018) and unnormalized exterior spherical harmonic coefficients up to degree 4 and order 4 available from Takahashi Y., *Gravity Field Characterization around Small Bodies*, University of Colorado, 2013 (Takahashi, 2013), shown in [Table 1.3](#).

The final acceleration of DustCube due to Didymain's gravitation has an order of magnitude of about $10^{-5} \div 10^{-6} [m/s^2]$, even if is important to remark that the major contribute comes from the central attraction $C_{1,0}$, while the other partial accelerations have orders of magnitude much smaller than the central one.

ORDER l	DEGREE m	$C_{l,m}$	$S_{l,m}$
0	0	1.0	–
1	0	0.0	–
1	1	0	0
2	0	-6.3422×10^{-2}	–
2	1	0.0	0.0
2	2	4.0949×10^{-3}	0.0
3	0	-1.5154×10^{-3}	–
3	1	2.8455×10^{-4}	1.1578×10^{-4}
3	2	2.89891×10^{-5}	-1.89599×10^{-5}
3	3	3.995×10^{-4}	-1.293×10^{-4}
4	0	4.66049×10^{-2}	–
4	1	-2.65537×10^{-5}	3.352119×10^{-5}
4	2	-9.588539×10^{-5}	-1.28121×10^{-6}
4	3	-8.305724×10^{-6}	-4.819896×10^{-6}
4	4	3.544874×10^{-5}	-7.124178×10^{-6}

Table 1.3: Unnormalized exterior spherical harmonic coefficients of Didymain (*Takahashi*)

The Didymoon's gravitational acceleration has been computed by using the degree-2 spherical harmonics expansion of a homogeneous triaxial ellipsoid with these approximated dimensions:

$$(a_d, b_d, c_d) = (103 \text{ m}, 79 \text{ m}, 66 \text{ m}), \quad (1.20)$$

by making use of the formulations available from Bills B.G. et al., *Harmonic and statistical analyses of the gravity and topography of Vesta*, ICARUS, 2014 (Bills, Asmar, Konopliv, Park, & Raymond, 2014) to compute the unnormalized spherical harmonic coefficients $J_2 = -C_{2,0}$ and $C_{2,2}$:

$$J_2 = \frac{1}{M_d R_d^2} \left(I_z - \frac{I_x + I_y}{2} \right) = -C_{2,0} \quad C_{2,2} = \frac{1}{M_d R_d^2} \left(\frac{I_y - I_x}{4} \right), \quad (1.21)$$

where R_d is the mean equatorial radius and I_x, I_y, I_z are the principal moments of inertia, namely:

$$I_x = \frac{M_d}{5} (b_d^2 + c_d^2) \quad I_y = \frac{M_d}{5} (a_d^2 + c_d^2) \quad I_z = \frac{M_d}{5} (a_d^2 + b_d^2) \quad (1.22)$$

The computed values of the Didymoon's harmonic coefficients are available in [Table 1.4](#).

ORDER l	DEGREE m	$C_{l,m}$	$S_{l,m}$
0	0	1.0	—
1	0	0.0	—
1	1	0	0
2	0	-9.8273×10^{-2}	—
2	1	0.0	0.0
2	2	2.6374×10^{-2}	0.0

Table 1.4: Unnormalized exterior spherical harmonic coefficients of Didymoon

The final acceleration of DustCube due to Didymoon's gravitation has an order of magnitude of about 10^{-7} [m/s²], where the central attraction stands for the major contribute.

Obviously, the gravitational accelerations due to Didymain or Didymoon are computed in body-fixed coordinates and translated in quasi-inertial RF at every iteration.

1.2.3 THE SOLAR RADIATION PRESSURE (SRP)

The Solar Radiation Pressure (SRP) contribution must be taken into account, since the acceleration due to it is not negligible during the DustCube mission, when Didymos will pass at a distance of just 0.11 AU from the Earth, as can be seen in [Figure 1.6](#).

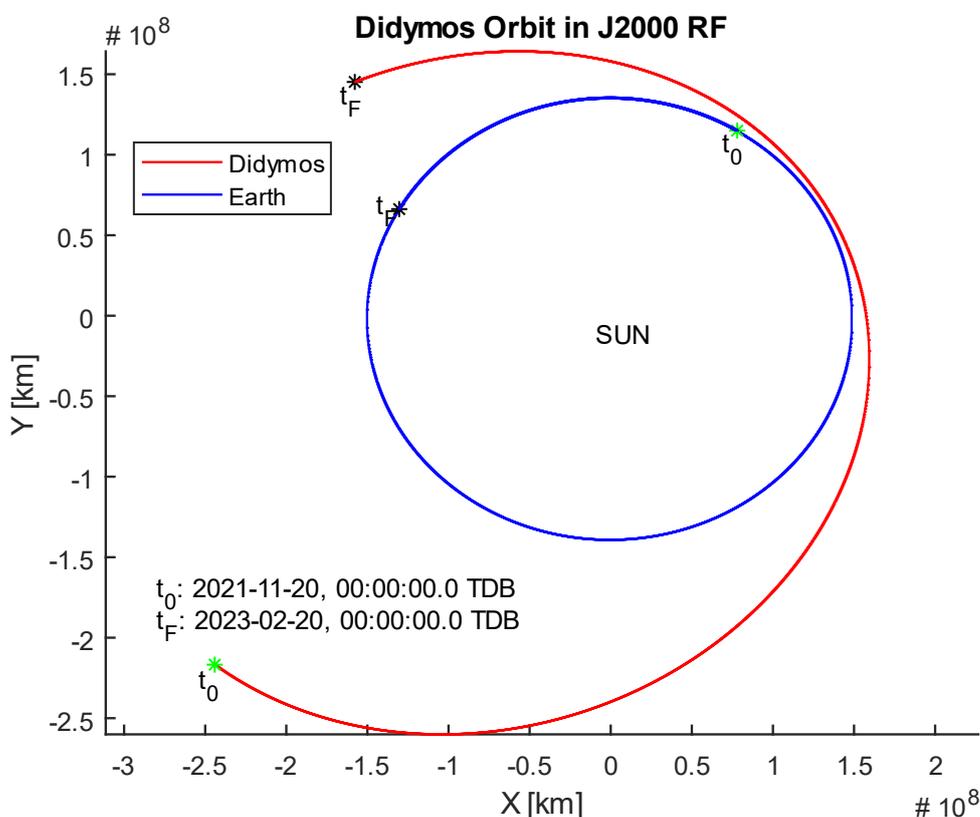


Figure 1.6: Didymos and Earth orbits from 11/20/202 (Christian, 2015)1 to 02/20/2023

Simplifying the geometry of the satellite, i.e. considering its shape as a flat plate, is possible to compute the force acting on DustCube due to SRP by exploiting the well-known behaviour of an opaque flat surface subjected to a flow of incoming luminous energy (Figure 1.7).

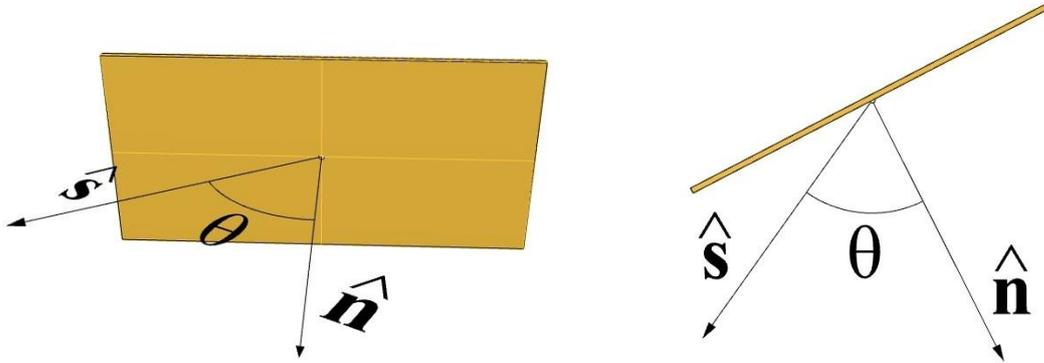


Figure 1.7: Principles of an illuminated flat plate in perspective view and in top one

Indeed, since the luminous energy that reaches an opaque surface is divided in absorbed, specularly reflected and diffusively reflected, the force acting on a flat plate due to SRP is computable as:

$$\mathbf{f}_{\text{SRP}}(t) = -P(t) \left\{ (1 - C_{\text{spec}}) \hat{\mathbf{s}}(t) + 2 \left[C_{\text{spec}} \cos(\theta(t)) + \frac{1}{3} C_{\text{diff}} \right] \hat{\mathbf{n}}(t) \right\} \cos(\theta(t)) A_{\text{tot}}, \quad (1.23)$$

where:

- $P(t)$ is the momentum flux regard the solar pressure [N/m^2];
- C_{spec} is the coefficient of the specular radiation emitted by the illuminated surface;
- C_{diff} is the coefficient of the diffusive radiation emitted by the illuminated surface;
- $\hat{\mathbf{s}}(t)$ is the unit vector directed from the surface towards the light source;
- $\hat{\mathbf{n}}(t)$ is the unit vector perpendicular to the illuminated surface;
- $\theta(t)$ is the angle between $\hat{\mathbf{s}}$ and $\hat{\mathbf{n}}$, which is always included between 0° and 90° ;
- A_{tot} is the approximated area of the illuminated surface [m^2].

In our case, since the light source is the Sun and the DustCube's camera will point towards Didymoon to acquire images during the DART's impact, the unit vector $\hat{\mathbf{s}}(t)$ will be directed towards the Sun and the versor $\hat{\mathbf{n}}(t)$ towards Didymoon or in the opposite direction, depending on which will be the illuminated surface.

The momentum flux $P(t)$ is a measure of the pressure exerted by the incoming light, therefore it depends on the luminous energy source, the distance from it and the light propagation medium.

The last feature affects the speed of movement of the electromagnetic waves while the first two ones are kept in consideration by employing the solar flux $\Phi(t)$ [W/m^2], which measures the luminous energy that impinges on the surface per unit time and per unit area, namely:

$$\Phi(t) \stackrel{\text{def}}{=} \frac{L_s}{4\pi\|\mathbf{s}(t)\|^2}, \quad (1.24)$$

where $L_s = 3.9 \times 10^{26}$ [W] is the luminosity of the Sun and $\|\mathbf{s}(t)\|$ is the distance from the energy source in [m], i.e. the distance between DustCube and the Sun, since $\mathbf{s}(t)$ is defined as the position vector of the last one, considered as a point in space, w.r.t. DustCube's CoM.

The evolution in time of the solar flux impinging on DustCube during its motion inside Didymos binary system is shown in [Figure 1.8](#), for a time window that spans from midnight of June 20, 2022 to midnight of November 20, 2022, covering five months entirely.

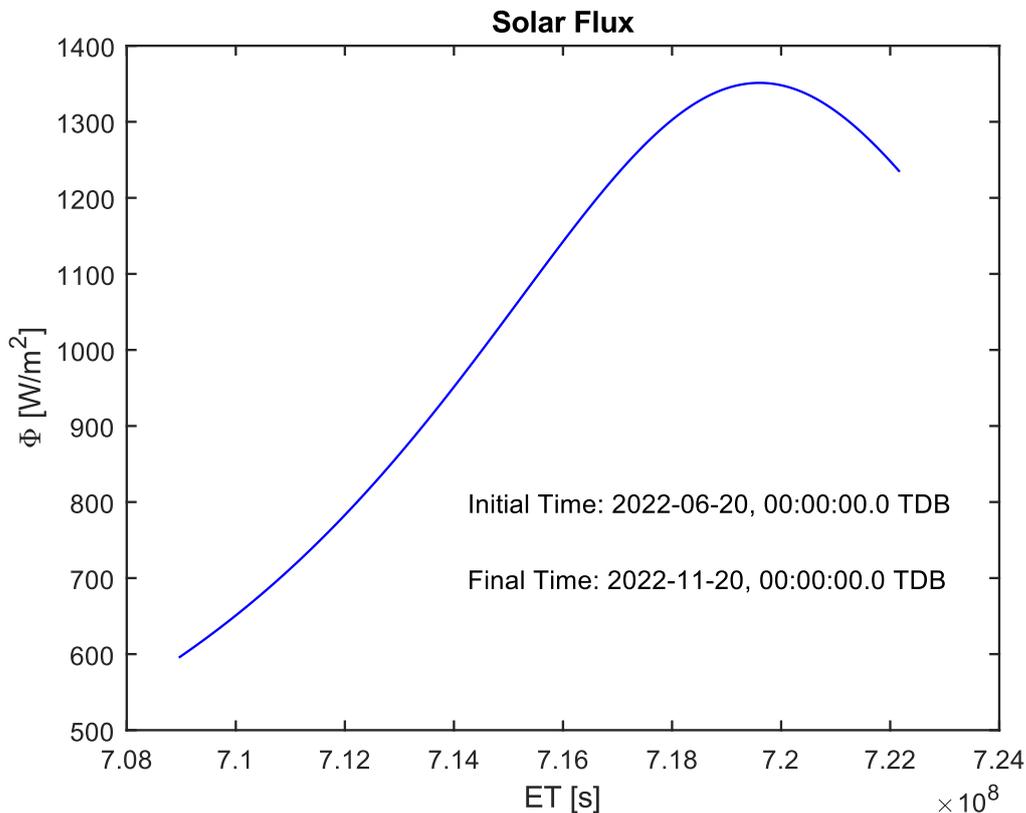


Figure 1.8: Solar flux impinging on DustCube from 2022-06-20 to 2022-11-20

Finally, since Didymos has no atmosphere and no ionosphere, the electromagnetic waves propagate in the vacuum, so that the momentum flux $P(t)$ for DustCube mission (sat subscript) depends on the solar flux (eq. (1.24)) and on the speed of light $c = 2,99792458 \times 10^8$ m/s:

$$P_{\text{sat}}(t) \stackrel{\text{def}}{=} \frac{\Phi(t)}{c} = \frac{L_s}{4\pi c \|\mathbf{s}(t)\|^2} \quad (1.25)$$

The adimensional coefficients C_{spec} , C_{diff} and C_{abs} are due to the considered material and are defined as energy ratios, namely:

$$C_{\text{spec}} \stackrel{\text{def}}{=} \frac{E_{\text{spec}}}{E_{\text{tot}}} \quad ; \quad C_{\text{diff}} \stackrel{\text{def}}{=} \frac{E_{\text{diff}}}{E_{\text{tot}}} \quad ; \quad C_{\text{abs}} \stackrel{\text{def}}{=} \frac{E_{\text{abs}}}{E_{\text{tot}}} \quad (1.26)$$

where E_{tot} is the total luminous energy that reaches the opaque surface measured in [J], while E_{spec} , E_{diff} and E_{abs} are the portions of the total energy that are specularly reflected, diffusively reflected and absorbed, respectively, so that every coefficient is lower than 1.

Therefore, since $E_{\text{tot}} = E_{\text{spec}} + E_{\text{diff}} + E_{\text{abs}}$, it is enough to know just two coefficients, because the third one can be easily computed by exploiting the obvious relation:

$$C_{\text{abs}} + C_{\text{spec}} + C_{\text{diff}} = 1 \quad (1.27)$$

Approximating DustCube as a flat plate with no thickness and considering the constrains about materials and payload, we can account for the following rough data:

- $m_{\text{sat}} = 4.365 \text{ kg}$;
- $A_{\text{tot sat}} = 0.09 \text{ m}^2$;
- $C_{\text{spec sat}} = 0.08$;
- $C_{\text{diff sat}} = 0.45$.

Finally, using the equation (1.23) and the previous data, it is possible to compute the force acting on DustCube due to SRP and the relative acceleration $\mathbf{a}_{\text{satSRP}}(t) = \mathbf{f}_{\text{satSRP}}(t)/m_{\text{sat}}$.

It is important to remark the relation (1.23) does not consider any attenuation factor, but since DustCube will orbit in a binary system, it is needed to take into account the shadows of the two massive bodies: Didymain and Didymoon.

Indeed, making geometric considerations, it is possible to understand the influence of a celestial body on an orbiting S/C in terms of shadow conditions (Figure 1.9).

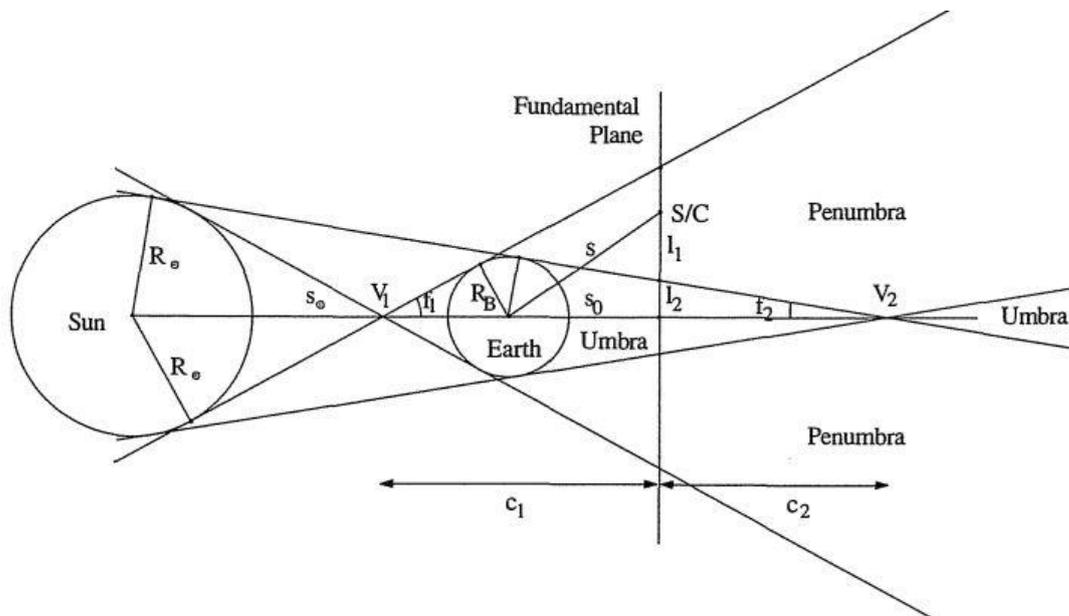


Figure 1.9: Conical shadow model (*Montenbruck & Gill, Satellite Orbits, 2000*)

Therefore, to refine the equation (1.23), is useful to introduce an adimensional factor, the so-called shadow function v_{sf} , which allows to take into account an attenuation of the force due to SRP when the S/C is subjected to partial illumination conditions.

Since the shadow function must tune the SRP's influence, that is maximum when the celestial bodies do not occult the sunlight, for which we can exploit the equation (1.23), the actual force vector acting on a flat surface due to SRP in presence of one occulting body can be computed as (*Montenbruck & Gill, Satellite Orbits, 2000*):

$$\mathbf{f}_{SRP}(t) \stackrel{\text{def}}{=} -v_{sf}(t)P(t) \left\{ (1 - C_{spec})\hat{\mathbf{s}}(t) + 2 \left[C_{spec} \cos(\theta(t)) + \frac{1}{3}C_{diff} \right] \hat{\mathbf{n}}(t) \right\} \cos(\theta(t)) A_{tot}, \quad (1.28)$$

so that $v_{sf}(t)$ assumes several values lower or equal to the unit related to the illumination conditions of the S/C, i.e.:

1. $v_{sf} = 1$ when the satellite is in **sunlight**;
2. $v_{sf} = 0$ when the satellite is in **umbra**;
3. $0 < v_{sf} < 1$ when the satellite is in **penumbra**.

Montenbruck & Gill (*Montenbruck & Gill, Satellite Orbits, 2000*) propose a computation of the shadow function which neglects the oblateness of the occulting body.

Indeed, simplifying the shapes of the three bodies (Sun, occulting body and S/C), i.e. considering the satellite like a flat plate and the Sun and the other celestial body as quasi-spherical, the degree of the

Therefore, making use of a quasi-inertial RF centered in the CoM of the system and considering the geometric illustrations of [Figure 1.10](#) and [Figure 1.11](#), we can define:

- $\mathbf{r}_s(t)$ as the position vector of the Sun;
- $\mathbf{r}_{occ}(t)$ as the position vector of the occulting body;
- $\mathbf{r}_{sat}(t)$ as the position vector of the S/C;
- R_s as the mean radius of the Sun;
- R_{occ} as the mean radius of the occulting body,

so that it is possible to compute the apparent separation between the geometric centers of the Sun and the occulting body (c) and they apparent radii (a and b) as:

$$\begin{aligned} a(t) &\stackrel{\text{def}}{=} \sin^{-1} \left[\frac{R_s}{\|\mathbf{r}_s(t) - \mathbf{r}_{sat}(t)\|} \right] \\ b(t) &\stackrel{\text{def}}{=} \sin^{-1} \left[\frac{R_{occ}}{\|\mathbf{r}_{occ}(t) - \mathbf{r}_{sat}(t)\|} \right] \\ c(t) &\stackrel{\text{def}}{=} \cos^{-1} \left[\frac{(\mathbf{r}_{occ}(t) - \mathbf{r}_{sat}(t)) \cdot (\mathbf{r}_s(t) - \mathbf{r}_{sat}(t))}{\|\mathbf{r}_{occ}(t) - \mathbf{r}_{sat}(t)\| \|\mathbf{r}_s(t) - \mathbf{r}_{sat}(t)\|} \right], \end{aligned} \quad (1.29)$$

which allows the computation of the occulted area of the apparent solar disk (A_{occ}), which is the turquoise one depicted in [Figure 1.11](#):

$$A_{occ}(t) = a(t)^2 \cos^{-1} \left[\frac{x(t)}{a(t)} \right] + b(t)^2 \cos^{-1} \left[\frac{c(t) - y(t)}{b(t)} \right] - c(t) y(t), \quad (1.30)$$

where $x(t)$ and $y(t)$ are two coefficients defined as:

$$\begin{aligned} x(t) &\stackrel{\text{def}}{=} \frac{c(t)^2 + a(t)^2 - b(t)^2}{2c(t)} \\ y(t) &\stackrel{\text{def}}{=} \sqrt{a(t)^2 - x(t)^2} \end{aligned} \quad (1.31)$$

It is important to remark that the eq. (1.30) is exploitable iff the following condition is verified:

$$|a(t) - b(t)| < c(t) < a(t) + b(t) \quad (1.32)$$

Once $A_{occ}(t)$ is defined, the computation of the shadow function $v_{sf}(t)$ becomes simply:

$$v_{sf}(t) = 1 - \frac{A_{occ}(t)}{\pi a(t)^2} \quad (1.33)$$

Chapter 1

Obviously, the different values of $v_{sf}(t)$ are linked to the occulting conditions, i.e.:

1. total occultation when: $b(t) - a(t) > c(t)$, i.e. $v_{sf}(t) = 0$;
2. partial but maximum occultation when: $a(t) - b(t) > c(t)$, i.e. $v_{sf}(t) = 1 - [b(t)^2/a(t)^2]$;
3. partial occultation when the condition (1.32) is verified, i.e. $v_{sf}(t)$ is computed from (1.33);
4. there is no occultation when: $a(t) + b(t) < c(t)$, i.e. $v_{sf}(t) = 1$.

The condition 1. brings to a null influence of the SRP on the S/C ($A_{occ} = \pi a^2$), while the condition 4. implies the S/C is totally in sunlight, i.e. $f_{satSRP}(t)$ is maximum ($A_{occ} = 0$).

In our case, since DustCube will operate in a binary system, we must consider the occultations of the primaries (Didymain and Didymoon), therefore the illuminating conditions of DustCube are influenced by two massive bodies with different shapes: a quasi-spherical body for Didymain and a quasi-ellipsoidal one for Didymoon.

Although the last one has not a quasi-spherical shape, it is useful to maintain the concept of Montenbruck & Gill to compute the Didymoon's shadow function, considering the mean of its dimensions (1.20), w/o making big errors.

Thus, the final solution of the force $f_{satSRP}(t)$ acting on DustCube, attenuated by the occultation of the sunlight due to the presence of the primaries, can be obtained by computing the two shadow functions separately, using two simplified quasi-spherical bodies (Figure 1.12).

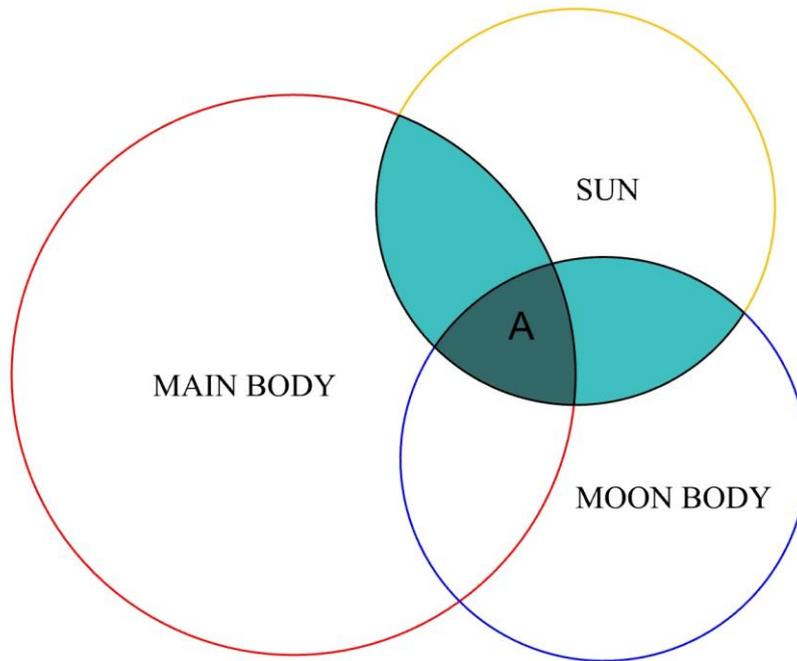


Figure 1.12: Occultation of the Sun caused by two spherical bodies

Since Didymoon has a quasi-ellipsoidal shape, the implementation of this solution is not optimal and a deeper study of the overlapping situations in case of two massive bodies with different shapes should be faced up in the future, although the method adopted in this work of thesis to compute $\mathbf{f}_{\text{satSRP}}(t)$ allows the user to reach a good compromise between precision of the solution and computational cost, to such an extent that the adopted method can be considered as the optimal one.

Finally, the SRP contribution on DustCube's motion is important, since the acceleration of the satellite has an order of magnitude of about $10^{-8} \div 10^{-9}$ [m/s²].

1.2.5 SUN'S GRAVITATIONAL ATTRACTION

The gravitational disturbance acted by Sun must be considered for the state's propagation of DustCube, since the 99.8% of the Solar System mass is contained in its unique star.

Since the Sun is far away from Didymos system, is suitable to simplify its gravitational influence as a central attraction, i.e. by considering it like a point-mass which contains its total mass.

Obviously, the Sun's gravitational influence acts on DustCube and on Didymos system too, so that, expressing the state of S/C in Didymos RF, which is centered in the CoM of the binary system, the final expression of the acceleration of DustCube due to Sun's gravitational disturbance must consider the positive solar central attraction, which directly acts on the S/C, and the negative contribution due to the same attraction working on Didymos barycenter (*Montenbruck & Gill, Satellite Orbits, 2000*). Therefore, approximating the solar mass and, consequently, the gravitational parameter as:

$$\begin{aligned} M_{\text{sun}} &= 1.98847 \times 10^{30} \text{ kg}, \\ \mu_{\text{sun}} &= GM_{\text{sun}} = 1.3271 \times 10^{20} \text{ m}^3/\text{s}^2, \end{aligned} \quad (1.34)$$

and referring again to [Figure 1.10](#), which has been used for the explanation of the SRP's influence, the DustCube's acceleration due to Sun's gravitation can be finally expressed as:

$$\mathbf{a}_{\text{satG}_{\text{sun}}}(t) \stackrel{\text{def}}{=} \mu_{\text{sun}} \left[\frac{\mathbf{r}_s(t) - \mathbf{r}_{\text{sat}}(t)}{\|\mathbf{r}_s(t) - \mathbf{r}_{\text{sat}}(t)\|^3} - \frac{\mathbf{r}_s(t)}{\|\mathbf{r}_s(t)\|^3} \right] = \mu_{\text{sun}} \left[\frac{\mathbf{s}(t)}{\|\mathbf{s}(t)\|^3} - \frac{\mathbf{r}_s(t)}{\|\mathbf{r}_s(t)\|^3} \right], \quad (1.35)$$

remembering that $\mathbf{s}(t)$ and $\mathbf{r}_s(t)$ are the Sun's position vectors w.r.t. DustCube and the Didymos' CoM, respectively. Finally, the Sun's gravitational acceleration acting on the S/C has an order of magnitude around 10^{-11} [m/s²], which is much lower than the other ones.

With the same reasoning, we can compute the accelerations of DustCube due to other solar bodies, such as Jupiter, which has a mass of about 10^{27} [kg], but their orders of magnitude are so small that we can omit them w/o making relevant errors.

Chapter 1

To compute the state of DustCube, is important to remark that we can indifferently choose to exploit the [Didymos Reference Frame](#), centered in the CoM of the binary system, or the [Didymain quasi-inertial Reference Frame](#), linked to the CoM of Didymain.

Therefore, in case of the selected RF is the second one, the propagator must consider not only the solar gravitational attraction acting on the CoM, but also the Didymoon's gravitational influence that works on the RF's CoM.

2 DUSTCUBE'S ORBIT PROPAGATOR

In section 1.2 [DustCube's dynamical model](#) of the previous chapter have been described the set of dynamic forces to which DustCube is subjected and the fundamental reference frames to describe its motion inside Didymos system.

Now, since the objective of the Kalman filter is to compute the evaluation of the satellite's state in real-time onboard DustCube, it has been important to implement an orbit propagator to simulate several S/C's orbits to finally test the Kalman filtering step by step, i.e. discretizing the time window with a constant time step, which has to be selected in a proper way with a compromise between precision of the computed solution and computational cost.

Therefore, an orbit propagator has been created in MATLAB exploiting the Runge-Kutta 4th order method, which is one of the most useful mathematical technique to approximate the solutions of ODE systems, and the SPICE Toolkit, an useful information system created by NAIF at JPL to assist NASA scientists in planning and interpreting scientific observations from space-borne instruments and to help NASA engineers involved in modelling, planning and executing activities needed to conduct planetary exploration missions (*Jet Propulsion Laboratory (JPL), NASA*).

2.1 THE CAUCHY PROBLEM FOR DUSTCUBE'S PROPAGATOR

Since the goal is to propagate the state of DustCube in a discretized time window, the problem is to evaluate a vector function $\mathbf{s}_{\text{sat}}(t)$ for a specific time-window, knowing its evolution in time, i.e. its first derivative $\mathbf{f}(t, \mathbf{s}_{\text{sat}}(t))$, and the initial value $\mathbf{s}_{\text{sat}}(t_0)$, considering a quasi-inertial reference frame centered in CoM of the binary system, which is the [Didymos Reference Frame](#) (DYD subscript) described in the first chapter.

Therefore, it's easy to recognize the classical Cauchy problem or initial value problem (IVP), as reported in section [A.3 General Cauchy problem](#), eq. [\(A.70\)](#), in Appendix:

$$\text{STATE VECTOR:} \quad \mathbf{s}_{\text{satDYD}}(t) = \begin{bmatrix} \mathbf{r}_{\text{sat}}(t) \\ \mathbf{v}_{\text{sat}}(t) \end{bmatrix}_{\text{DYD}} \quad (2.1)$$

$$\text{EVOLUTION IN TIME:} \quad \dot{\mathbf{s}}_{\text{satDYD}}(t) = \mathbf{f}(t, \mathbf{s}_{\text{satDYD}}(t)) = \begin{bmatrix} \dot{\mathbf{r}}_{\text{sat}}(t) \\ \dot{\mathbf{v}}_{\text{sat}}(t) \end{bmatrix}_{\text{DYD}} = \begin{bmatrix} \mathbf{v}_{\text{sat}}(t) \\ \mathbf{a}_{\text{sat}}(t) \end{bmatrix}_{\text{DYD}} \quad (2.2)$$

$$\text{INITIAL VALUE VECTOR:} \quad \mathbf{s}_{\text{satDYD}}(t_0) = \begin{bmatrix} \mathbf{r}_{\text{sat}}(t_0) \\ \mathbf{v}_{\text{sat}}(t_0) \end{bmatrix}_{\text{DYD}} = \begin{bmatrix} \mathbf{r}_{\text{sat}0} \\ \mathbf{v}_{\text{sat}0} \end{bmatrix}_{\text{DYD}}, \quad (2.3)$$

Chapter 2

where $\mathbf{r}_{\text{satDyD}}(\mathbf{t})$, $\mathbf{v}_{\text{satDyD}}(\mathbf{t})$ and $\mathbf{a}_{\text{satDyD}}(\mathbf{t})$ are the position, the velocity and the acceleration vectors of DustCube w.r.t. the CoM of Didymos.

Finally, the objective is to solve a system of ODEs with known initial values, where the acceleration is a sum of the elementary contributions acting on DustCube, as it has been described in section 1.2 [DustCube's dynamical model](#) of chapter 1.

For instance, if we only consider the central attractions acted by Didymain and Didymoon and the SRP w/o shadow function contribution, the expression of the derivative of the state vector becomes:

$$\dot{\mathbf{s}}_{\text{satDyD}}(\mathbf{t}) = \left[\begin{array}{c} \mathbf{v}_{\text{sat}}(\mathbf{t}) \\ -\frac{\mu_D}{\|\mathbf{r}_{\text{sat-D}}(\mathbf{t})\|^3} \mathbf{r}_{\text{sat-D}}(\mathbf{t}) - \frac{\mu_d}{\|\mathbf{r}_{\text{sat-d}}(\mathbf{t})\|^3} \mathbf{r}_{\text{sat-d}}(\mathbf{t}) + \mathbf{a}_{\text{satSRP}}(\mathbf{t}) \end{array} \right]_{\text{DYD}}, \quad (2.4)$$

which can be expanded in components:

$$\dot{\mathbf{s}}_{\text{satDyD}}(\mathbf{t}) = \left[\begin{array}{c} v_{x_{\text{sat}}}(\mathbf{t}) \\ v_{y_{\text{sat}}}(\mathbf{t}) \\ v_{z_{\text{sat}}}(\mathbf{t}) \\ -\frac{\mu_D}{\|\mathbf{r}_{\text{sat-D}}(\mathbf{t})\|^3} x_{\text{sat-D}}(\mathbf{t}) - \frac{\mu_d}{\|\mathbf{r}_{\text{sat-d}}(\mathbf{t})\|^3} x_{\text{sat-d}}(\mathbf{t}) + a_{x_{\text{satSRP}}}(\mathbf{t}) \\ -\frac{\mu_D}{\|\mathbf{r}_{\text{sat-D}}(\mathbf{t})\|^3} y_{\text{sat-D}}(\mathbf{t}) - \frac{\mu_d}{\|\mathbf{r}_{\text{sat-d}}(\mathbf{t})\|^3} y_{\text{sat-d}}(\mathbf{t}) + a_{y_{\text{satSRP}}}(\mathbf{t}) \\ -\frac{\mu_D}{\|\mathbf{r}_{\text{sat-D}}(\mathbf{t})\|^3} z_{\text{sat-D}}(\mathbf{t}) - \frac{\mu_d}{\|\mathbf{r}_{\text{sat-d}}(\mathbf{t})\|^3} z_{\text{sat-d}}(\mathbf{t}) + a_{z_{\text{satSRP}}}(\mathbf{t}) \end{array} \right]_{\text{DYD}} \quad (2.5)$$

Since the equations of the dynamic of the system are not linear, is not possible to compute an exact solution of the state vector in the selected time window, thus it must be approximated by exploiting a suited numerical method, that is an ODE solver, by discretizing the time window ΔT with a constant time-step h .

Therefore, the problem becomes to compute a discretized solution of the continuous function $\mathbf{s}_{\text{sat}}(\mathbf{t})$, i.e. $\mathbf{s}_{\text{sat}}(\mathbf{t}_k)$ with $\mathbf{t}_k \in \Delta T$ and $k = 0, \dots, n$, i.e. $\mathbf{t}_k = \mathbf{t}_0 + kh$, by knowing a-priori $\mathbf{s}_{\text{satDyD}}(\mathbf{t}_0)$ and $\mathbf{f}(\mathbf{t}, \mathbf{s}_{\text{satDyD}}(\mathbf{t}))$, which require to select an appropriate time-step h and to choose the best numerical method in terms of compromise between precision of the solution and computational cost.

There exists a lot of ODE solvers in MATLAB, such as *ode23t.m*, *ode23s.m*, *ode45.m* and so on, but I decided to implement an independent ODE solver exploiting the Runge-Kutta 4th order method to have more flexibility, since it allows flexibility in the choice of the time-step h , because it is a robust numerical method, and to easily access to the data of the problem at every instant of time of the propagation.

2.2 THE RUNGE-KUTTA 4TH ORDER METHOD

The Runge-Kutta 4th order method is a useful numerical explicit method to approximate the solutions of ordinary differential equations (ODEs), especially for initial values problems (IVPs).

Its name derives from the German mathematicians Carl David Tolmé Runge and Martin Wilhelm Kutta, which developed the Runge-Kutta methods around 1900 and published their work in 1901.

The RK4 method is an accurate and precise one-step method that allows to properly select the time-step of the integration, so that it is possible to try different time-steps with the goal to reach a compromise between local errors and computational costs.

Since it can be easily extended to ODEs systems and does not require a big computational effort, it is the most exploited method in spacecraft computations.

Considering the continuous Cauchy problem of eqs (2.1)÷ (2.3) and discretizing the time window with a constant time step h as described before, i.e. $\Delta T = (t_0, \dots, t_k, \dots, t_n)$ with $0 \leq k \leq n$, the general formulation of RK4 for a system of ODEs makes sure that the vector function is approximated at every time-step h as:

$$\mathbf{u}_k = \mathbf{u}_{k-1} + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4), \quad (2.6)$$

where \mathbf{u}_k and \mathbf{u}_{k-1} are the computations of the function at the instants of time k and $k - 1$, respectively, and $\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4$ are the evaluations of the temporal derivative of the function \mathbf{u} at different instants of time, so that they are defined as:

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}_{k-1} = \mathbf{f}(t_{k-1}, \mathbf{u}_{k-1}) \\ \mathbf{k}_2 &= \mathbf{f}\left(t_{k-1} + \frac{h}{2}, \mathbf{u}_{k-1} + \frac{h}{2}\mathbf{k}_1\right) \\ \mathbf{k}_3 &= \mathbf{f}\left(t_{k-1} + \frac{h}{2}, \mathbf{u}_{k-1} + \frac{h}{2}\mathbf{k}_2\right) \\ \mathbf{k}_4 &= \mathbf{f}(t_{k-1} + h, \mathbf{u}_{k-1} + h\mathbf{k}_3) \end{aligned} \quad (2.7)$$

The computational cost depends on the width of the time window ΔT and the selected time step h . Indeed, for the same time window, a smaller time step allows the user to reach a better accuracy, while a bigger one permits to achieve a lower computational cost.

2.3 THE SPICE TOOLKIT

The SPICE Toolkit (*Jet Propulsion Laboratory (JPL), NASA, 2017*) is based on **kernel**s, that are binary or text files containing important data such as ephemeris of celestial bodies, quaternions or Euler angles of reference frames, shape models of celestial bodies and so on.

There are several types of kernels that contain different data:

- **CK**: orientation information, called the "C-matrix," between S/Cs and their scientific instruments;
- **DSK**: digital shape model info, that offer the possibility of making use of higher fidelity shape models w.r.t. some data insert in PCK kernels. Obviously, they are available just for those solar bodies for which scientists have calculated detailed shapes, such as Earth, Moon, Mars and the most important asteroids and comets;
- **EK**: events information that summarize mission activities (rarely exploited);
- **FK**: info of a specific reference frame that is used for space missions;
- **IK**: instrument information, such as field-of-view size, shape and orientation parameters;
- **LSK**: leap-seconds info, that must be used together with SCLK kernels;
- **PCK**: physical, dynamical and cartographic constants for target bodies, such as size, shape, orientation of the spin axis and so on;
- **SCLK**: spacecraft clock info, that must be used together with LSK kernels to convert time tags between various time measurement systems;
- **SPK**: planets, satellites, comets, asteroid and spacecraft ephemeris given as a function of time.

NAIF/JPL offers the possibility to download a lot of kernels from its website at the link https://naif.jpl.nasa.gov/pub/naif/generic_kernels/, even if some kernels must be created by the user to properly exploit the potentialities of the SPICE Toolkit.

Once the useful kernels for the mission have been defined, the Toolkit allows the user to manage the data contained in these files by permitting the dialogue with the software packages C, Fortran, IDL, MATLAB.

The dialogue with MATLAB is guaranteed by the MICE library, that contains a lot of MATLAB functions and routines generated by NAIF/JPL that permit to exploit the loaded kernels and to access their data. The MICE library can be downloaded from NAIF website at the link https://naif.jpl.nasa.gov/naif/toolkit_MATLAB.html.

Also, the MICE library allows the user to exploit other utilities, such as execution programmes, useful to easily dialogue with SPICE kernels.

Since the usage of SPICE kernels and MICE library is not easy, NAIF website contains a database of tutorials and lessons in pdf extension to allows the user to properly manage the potentialities of the SPICE Toolkit at the link <https://naif.jpl.nasa.gov/naif/tutorials.html> (*Jet Propulsion Laboratory (JPL), NASA, 2018*).

2.3.1 SPICE KERNELS FOR DUSTCUBE'S ORBIT PROPAGATOR

The SPICE Toolkit has been used to recreate the environment in which DustCube will be immersed. To do this, some kernels have been uploaded from NAIF website:

- ***naif0012.tls.pc***, an LSK kernel which contains leap seconds to adjust time, released by NAIF/JPL on 14th July 2016;
- ***de432s.bsp***, an SPK binary file which contains planetary and lunar ephemeris of the Solar System, released by NAIF/JPL on 1st May 2014;
- ***pck00010.tpc***, a PCK kernel which contains orientation and shape data of several Solar System's celestial bodies.

Also, I've created other kernels to properly study the orbit of DustCube:

- ***2065803.bsp***: is an SPK binary file which contains the ephemeris of Didymos from 1st January 1990 to 1st January 2050. This file was created by making use of the Horizons SPK file generator (*Jet Propulsion Laboratory (JPL), NASA, 2018*), offered by the JPL website at the link <https://ssd.jpl.nasa.gov/x/spk.html>;
- ***didymosinertial.tf***: is an FK text file which contains the Didymos inertial reference frame linked to SSB, presented in section 1.2.1 Reference Frames, with the x-y plane parallel to the plane containing the mean mutual orbit of Didymain and Didymoon;
- ***didymosframe.tf***: is an FK text file which contains the pseudo-inertial reference frame linked to CoM of Didymos, presented in section 1.2.1 Reference Frames. This frame does not rotate, but it simply follows the CoM of Didymos system, with the x-y plane that is the mean orbital plane of the mutual orbit of Didymain and Didymoon;
- ***IAU_Didymos.tf***: is an FK text file created by PHD student Riccardo Lasagni Manghi of University of Bologna (*Lasagni Manghi, Modenini, Zannoni, & Tortora, 2018*) which contains the main features of the body-fixed reference frames of Didymain and Didymoon;
- ***Didymos_constants.pck***: is a PCK text file that contains the main features of the primaries.

2.3.2 SPICE'S USEFUL ROUTINES

The MICE library contains a lot of routines for space analysis. Obviously, just a small set of this library is useful for the DustCube orbit propagator.

Here below is possible to visualize the most exploited functions with some example calls:

- ***cspice_furnsh.m***: this function is useful to upload kernels inside MATLAB environment. For instance, the call `cspice_furnsh({'naif0012.tls.pc','de432s.bsp','2065803.bsp'})` uploads *naif0012.tls.pc*, *de432s.bsp*, *2065803.bsp* kernels;
- ***cspice_str2et.m***: this routine is employed to insert dates inside MATLAB environment. Once the user has written dates in strings, this function converts these dates in ET. For instance, the call `cspice_str2et({'2022 AUG 08 00:00:00.0','2022 AUG 20 00:00:00.0'})` is useful to insert the initial time and the final time for the orbit's propagation;
- ***cspice_spkpos.m***: this function is used to obtain the position vector of a target point in space, starting from an observation point in space, in rectangular coordinates, w.r.t. a specific reference frame and a precise time window. For instance, the command `cspice_spkpos('Sun',epochs,'eclipJ2000','NONE','2065803')` is used to compute the position of the CoM of Sun w.r.t. CoM of Didymos ('2065803') in ecliptic frame ('eclipJ2000') for the time window 'epochs', without aberration corrections ('NONE');
- ***cspice_spkezr.m***: this routine is useful to obtain the state of a target point in space, in rectangular coordinates, w.r.t. an observation point in space., for. a specific reference frame and a precise time window. For instance, the call `cspice_spkezr('2065803',epochs,'eclipJ2000','LT+S','0')` allows user to compute the state of the CoM of Didymos w.r.t. Solar System Barycenter ('0'), in ecliptic frame, during the time window 'epochs' and corrected for one-way light time (LT) and stellar (S) aberration using a Newtonian formulation. The result is the apparent position of the target;
- ***cspice_pxform.m***, this routine is useful to compute the rotation matrix between frames for a specified time window. For instance, the command `cspice_pxform('eclipJ2000','didymosframe',t0)` generates the rotation matrix between the ecliptic frame and the Didymos frame at epoch 't0'.

Finally, by making use of these routines and the previous kernels, it is possible to reconstruct the dynamical model of DustCube and, consequently, to compute its state.

2.4 IMPLEMENTATION OF THE ORBIT PROPAGATOR

Once defined the numerical method to exploit and the necessary kernels and SPICE routines, it is useful to decompose the problem in different integration steps.

Since the MATLAB propagator had to be tested step by step, the adopted solution for the purpose of this thesis was to split the acceleration of DustCube in its main contributions, each of which has a proper MATLAB routine created specifically for this mission.

Obviously, the computational complexity of the propagator increases together with the growth of the number of accelerations considered.

The states of DustCube are obtained in discretized form in [Didymos Reference Frame](#), which can be thought as quasi-inertial for steep time windows, then transformed in [Synodic Reference Frame](#), which represents the better RF to show the orbit of DustCube by exploiting its properties, such as the *Lagrange equilibrium points* (see [Figure A.6](#) in Appendix).

Therefore, the first step was computing the Synodic features for Didymos system, the second one integrating the DustCube's orbit in Didymos RF, the third one transforming the states of DustCube in synodic coordinates.

However, the user can autonomously decide to obtain the propagation of the motion of DustCube in discretized form [Didymain quasi-inertial Reference Frame](#) instead of in Didymos RF, as it will be clearer in section [2.4.3 The orbital propagator's scripts](#).

The propagator has been developed by creating the MATLAB script *DustCube_Main.m*.

2.4.1 THE DIDYMOS' SYNODIC PLANE

The most important features of Didymos' Synodic Plane can be obtained through the specifically created MATLAB function *Libration_Points.m*, which computes separately the *collinear* equilibrium points L_1 , L_2 , L_3 and the *triangular* ones L_4 , L_5 .

To exploit the computations shown in [A.2 The Circular Restricted 3-Body Problem \(CR3BP\)](#) in Appendix, is necessary to initially define the *mass parameter* η_{didy} related to Didymos (eq. [\(A.55\)](#)) through the data [\(1.1\)](#), namely:

$$\eta_{\text{didy}} = \frac{M_d}{M_D + M_d} = \frac{M_d}{M_{\text{tot}}} = 0.0092 \quad (2.8)$$

Subsequently, considering the *Lagrange quintic equation* [\(A.64\)](#), which is a non-linear 5th order equation that asymptotically converges to three real values iff the initial ones u_{10} , u_{20} and u_{30} are properly selected, as in the eqs [\(A.66\)](#), we can solve it by exploiting the well-known Newton-Raphson

Chapter 2

numerical method (*Quarteroni, Sacco, Saleri, & Gervasio, 2014*) to obtain three real solutions u_1, u_2 and u_3 , which allow to achieve the *collinear* equilibrium points L_1, L_2, L_3 by using the eq. (A.63):

$$\begin{aligned} L_1 &= u_1 + 1 - \eta \\ L_2 &= u_2 + 1 - \eta \\ L_3 &= u_3 + 1 - \eta, \end{aligned} \tag{2.9}$$

while the *triangular* equilibrium points L_4 and L_5 are directly computable through the eqs. (A.68).

The previous computations allow to obtain the *Lagrange equilibrium points* in adimensional form, but since the objective is to reconstruct the Didymos' Synodic Plane to use it for DustCube's orbital representations, is useful to transform L_1, L_2, L_3, L_4 and L_5 in dimensional coordinates by multiplying them with the mean distance between primaries, which is $d_{Dd} = 1180$ m.

Finally, in [Table 2.1](#) are shown the resulting *Libration points* for Didymos binary system.

COLLINEAR EQUILIBRIUM POINTS	$L_1 = (1005.91, 0, 0)$ m
	$L_2 = (1349.17, 0, 0)$ m
	$L_3 = (-1184.53, 0, 0)$ m
TRIANGULAR EQUILIBRIUM POINTS	$L_4 = (579.12, 1021.91, 0)$ m
	$L_5 = (579.12, -1021.91, 0)$ m

Table 2.1: The computed Lagrange equilibrium points of Didymos binary system

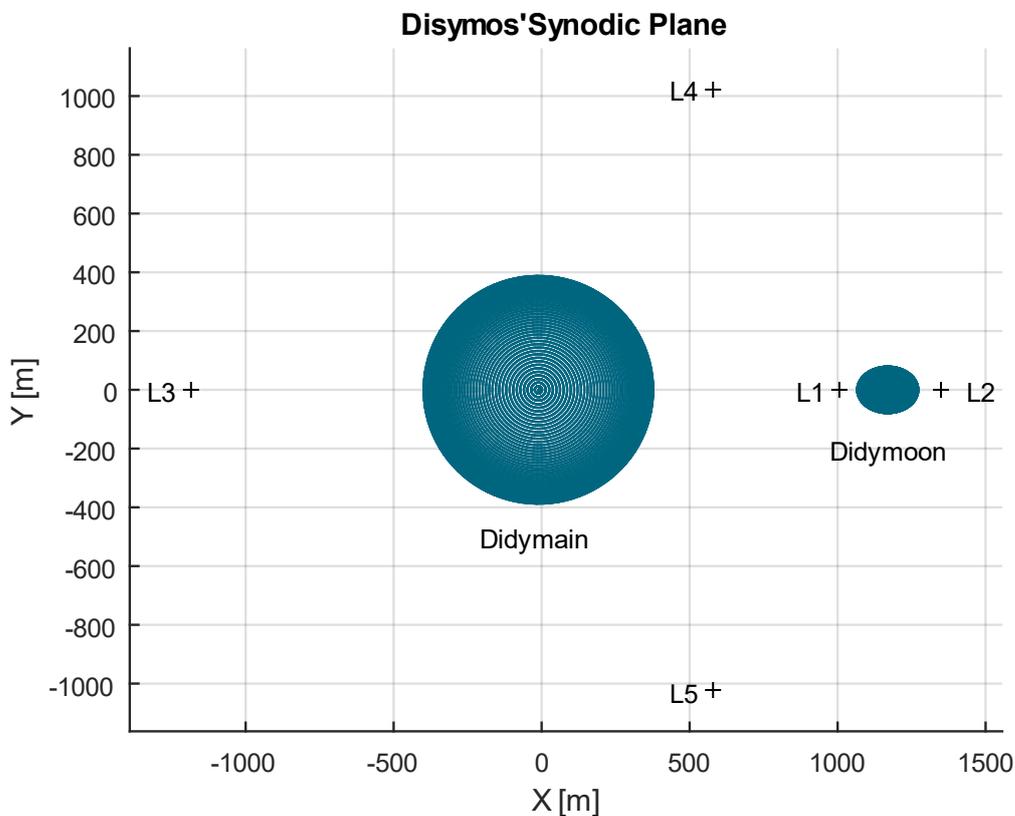


Figure 2.1: Didymos' Synodic Plane

In Figure 2.1 is shown the Didymos' Synodic Plane in dimensional form with the computed *Libration points* of Table 2.1 and the shapes and positions of Didymain and Didymoon.

2.4.2 THE TRANSFORMATION IN SYNODIC COORDINATES

Since it is convenient to evaluate the orbit of DustCube in the Synodic Plane, the MATLAB function *synodic_state.m* has been developed to transform the state of DustCube, which is computed in discretized form, from the Didymos RF to the Synodic one, which can be eased by exploiting the quasi-circularity of the orbit of Didymoon around Didymain, that means the module of the instantaneous angular velocity of Didymoon remains very close to its mean value.

Hence, considering the discretized Didymoon's state in DYD coordinates, namely:

$$\mathbf{s}_{\text{dDYD}}(t_k) = \begin{bmatrix} \mathbf{r}_d(t_k) \\ \mathbf{v}_d(t_k) \end{bmatrix}_{\text{DYD}} = \begin{bmatrix} \mathbf{r}_{\text{dDYD}}(t_k) \\ \mathbf{v}_{\text{dDYD}}(t_k) \end{bmatrix}, \quad (2.10)$$

is possible to compute the transformation matrix from DYD to SYN coordinates by exploiting the canonical base resulting from the vector triad composed by:

- the position vector of Didymoon: $\mathbf{r}_{\text{dDYD}}(t_k)$;
- the first integral of motion of Didymoon's orbit: $\mathbf{h}_{\text{dDYD}} = \mathbf{r}_{\text{dDYD}}(t_k) \times \mathbf{v}_{\text{dDYD}}(t_k)$;
- the vector which completes the right-hand frame: $\boldsymbol{\theta}_{\text{dDYD}}(t_k) = \mathbf{h}_{\text{dDYD}} \times \mathbf{r}_{\text{dDYD}}(t_k)$,

Indeed, the unit vectors which form the canonical base ($\hat{\mathbf{r}}_{\text{dDYD}}(t_k), \hat{\boldsymbol{\theta}}_{\text{dDYD}}(t_k), \hat{\mathbf{h}}_{\text{dDYD}}$), namely:

$$\hat{\mathbf{r}}_{\text{dDYD}}(t_k) = \frac{\mathbf{r}_{\text{dDYD}}(t_k)}{\|\mathbf{r}_{\text{dDYD}}(t_k)\|} \quad \hat{\boldsymbol{\theta}}_{\text{dDYD}}(t_k) = \frac{\boldsymbol{\theta}_{\text{dDYD}}(t_k)}{\|\boldsymbol{\theta}_{\text{dDYD}}(t_k)\|} \quad \hat{\mathbf{h}}_{\text{dDYD}} = \frac{\mathbf{h}_{\text{dDYD}}}{\|\mathbf{h}_{\text{dDYD}}\|}, \quad (2.11)$$

can be used to compute the rotation matrix from DYD to SYN coordinates, i.e.:

$$\mathbf{T}_{\text{DYD-SYN}}(t_k) = \begin{bmatrix} \hat{\mathbf{r}}_{\text{dDYD}}(t_k) & \hat{\boldsymbol{\theta}}_{\text{dDYD}}(t_k) & \hat{\mathbf{h}}_{\text{dDYD}} \end{bmatrix}^T = \begin{bmatrix} \hat{\mathbf{r}}_{\text{dDYD}}(t_k)^T \\ \hat{\boldsymbol{\theta}}_{\text{dDYD}}(t_k)^T \\ \hat{\mathbf{h}}_{\text{dDYD}}^T \end{bmatrix} \quad (2.12)$$

Also, to compute the DustCube's velocity in SYN coordinates, we need to obtain the instantaneous angular velocity of Didymoon w.r.t. the CoM of Didymos $\boldsymbol{\omega}_{\text{dDYD}}(t_k)$, namely:

$$\boldsymbol{\omega}_{\text{dDYD}}(t_k) = \frac{\mathbf{h}_{\text{dDYD}}}{\|\mathbf{r}_{\text{dDYD}}(t_k)\|^2} \quad (2.13)$$

Chapter 2

Once obtained the matrix $\mathbf{T}_{\text{DYD-SYN}}(t_k)$ and the vector $\boldsymbol{\omega}_{\text{dDYD}}(t_k)$, the state of DustCube in SYN coordinates can be computed by exploiting the classical relations useful to transform position and velocity vectors from an inertial reference frame to a rotating frame with the same CoM (see eqs (A.37) and (A.13) in Appendix):

$$\begin{aligned}\mathbf{r}_{\text{satSYN}}(t_k) &= \mathbf{T}_{\text{DYD-SYN}}(t_k) * \mathbf{r}_{\text{satDYD}}(t_k) \\ \mathbf{v}_{\text{satSYN}}(t_k) &= \mathbf{T}_{\text{DYD-SYN}}(t_k) * \mathbf{v}_{\text{satDYD}}(t_k) - \boldsymbol{\omega}_{\text{dDYD}}(t_k) \times \mathbf{r}_{\text{satDYD}}(t_k)\end{aligned}\quad (2.14)$$

The choice to use the vector $\boldsymbol{\omega}_{\text{dDYD}}(t_k)$ comes from its easier computation than the first derivative of the rotation matrix $\dot{\mathbf{T}}_{\text{DYD-SYN}}(t_k)$.

It is important to remark that the actual equations to transform the DustCube's state from DYD to SYN coordinates require a constant rotation matrix and a constant angular velocity, which are defined as the mean values of the instantaneous rotation matrix $\mathbf{T}_{\text{DYD-SYN}}(t_k)$ and the instantaneous angular velocity $\boldsymbol{\omega}_{\text{dDYD}}(t_k)$, so that:

$$\begin{aligned}\mathbf{r}_{\text{satSYN}}(t_k) &= \mathbf{T}_{\text{mean}} * \mathbf{r}_{\text{satDYD}}(t_k) \\ \mathbf{v}_{\text{satSYN}}(t_k) &= \mathbf{T}_{\text{mean}} * \mathbf{v}_{\text{satDYD}}(t_k) - \boldsymbol{\omega}_{\text{mean}} \times \mathbf{r}_{\text{satSYN}}(t_k)\end{aligned}\quad (2.15)$$

However, since the orbit of Didymoon around the CoM of Didymos is quasi-circular, i.e. its eccentricity is close to zero ($e_{\text{orb}} = 0.03$), is useful to compute the transformation of the DustCube's state from DYD to SYN coordinates by exploiting the eqs (2.14), which allows the user to obtain a final solution w/o large absolute errors, because the instantaneous angular velocity of Didymoon differs slightly from the rotation speed of the Synodic RF w.r.t. the Didymos one.

2.4.3 THE ORBITAL PROPAGATOR'S SCRIPTS

As said before, the state of DustCube is achieved by exploiting the already mentioned Cauchy problem (eqs (2.1)÷ (2.3)) with a growing complexity due to a step by step addition of acceleration contributions (see 1.2 DustCube's dynamical model for more details).

Therefore, several MATLAB functions have been created to propagate the DustCube's state, each of which employs the RK4 numerical method explained in section 2.2 The Runge-Kutta 4th order method, starting from an initial state defined a-priori:

1. The scripts ***RK_3bodies.m*** propagates the DustCube's state by just considering the central attractions of Didymain and Didymoon, that regards to consider the primaries as two punctual massive bodies w/o shape. Therefore, the evolution in time of the state vector can be easily

visualized by erasing the SRP contribution from eq. (2.5) and by switching from the continuous form to the discretized one.

The accelerations due to the central attractions of Didymain and Didymoon, $\mathbf{a}_{\text{sat}_{G_D}}(t_k)$ and $\mathbf{a}_{\text{sat}_{G_d}}(t_k)$, respectively, are computed by exploiting the specifically designed MATLAB function *central_attraction.m*, which uses the triangulation principle between the position vector of the primary w.r.t. CoM of the system and the position vector of the satellite w.r.t. the same CoM, as shown in Figure A.4 in Appendix, to obtain the gravitational forces acting on DustCube at every instant of the discretized time window.

Since the CR3BP (see A.2 The Circular Restricted 3-Body Problem (CR3BP) in Appendix) is based on the concept to consider the celestial bodies of a binary system as point-masses and the satellite's influence on the motion of the primaries as null (eq. (A.54)), if the user is going to park DustCube in one of the five *Lagrange equilibrium points*, he will obtain a solution of the state of DustCube as expressed in Didymos RF such that, extrapolating the position vector and transforming it in synodic coordinates, will demonstrate the validity of the computation, i.e. the S/C will not move from the equilibrium point w.r.t. the Synodic RF.

2. The script *RK_3bodies_SRP.m* propagates the state of DustCube by exploiting the central attractions of the primaries and the SRP contribution as shown in the eq. (2.5), where the force acting on DustCube due to SRP can be obtained by omitting the shadow function (eq. (1.23)) or by considering it (eq. (1.28)) and by switching from the continuous form to the discretized one. This script exploits the already mentioned *central_attraction.m* function to achieve the gravitational accelerations of the primaries, $\mathbf{a}_{\text{sat}_{G_D}}(t_k)$ and $\mathbf{a}_{\text{sat}_{G_d}}(t_k)$ respectively, and the specifically created one *SRP_Montenbruck_complete.m* to compute the acceleration of DustCube due to SRP $\mathbf{a}_{\text{sat}_{SRP}}(t_k)$ at every instant of the discretized time window.

Thus, this script estimates the shadow functions related to Didymain and Didymoon, $\nu_{sf_D}(t_k)$ and $\nu_{sf_d}(t_k)$ respectively, and thus the force acting on DustCube $\mathbf{f}_{\text{sat}_{SRP}}(t_k)$ from eq. (1.28) at every instant of time, which is possible by exploiting the specifically designed script *SRP_data.m*, that continuously computes the unit vectors $\hat{\mathbf{s}}(t_k)$ and $\hat{\mathbf{n}}(t_k)$ of Figure 1.7, the corresponding trigonometric function $\cos(\theta(t_k))$ and the solar flux $\Phi(t_k)$ defined in the equation (1.24).

For example, considering a time window of 120 hours spans from 2022-08-20 to 2022-08-25 and a parking orbit in L_5 , the user can obtain the evolution of the DustCube's state for the discretized time window in DYD coordinates and transform it in SYN ones to evaluate the deviation of DustCube from the L_5 equilibrium point due to SRP, as shown in Figure 2.2.

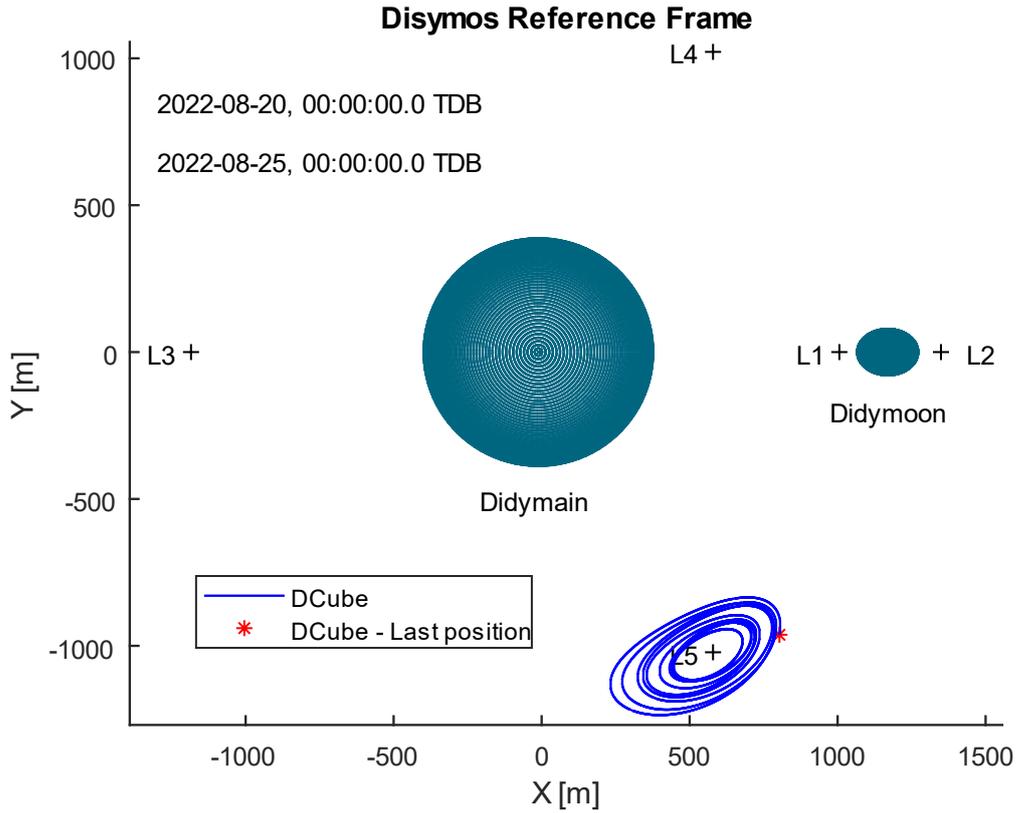


Figure 2.2: DustCube's parking orbit in L_5 from 2022-08-20 to 2022-08-25, considering Didymain and Didymoon as point masses and the SRP contribution

The script also computes the solar flux that impinges on DustCube during its motion inside Didymos system, which is shown in Figure 2.3, where the reader can evaluate the shadow periods, that are the time segments in which the solar flux is zero due to the occultation of the sunlight acted by the primaries, and the shadow function values of Didymain and Didymoon for the same discretized time window, that are shown in Figure 2.4.

From the last image, the user can separately assess the influences of Didymain and Didymoon on the occultation of the sunlight during the DustCube's parking orbit in L_5 and he can easily infer that just Didymain can conceal the Sun for the considered DustCube's orbit and time window, since the Didymoon's shadow function constantly remains at its maximum value, i.e.:

$$v_{sf_d}(t) = 1 \quad \forall t \in [2022/08/20 : 2022/08/25] \quad (2.16)$$

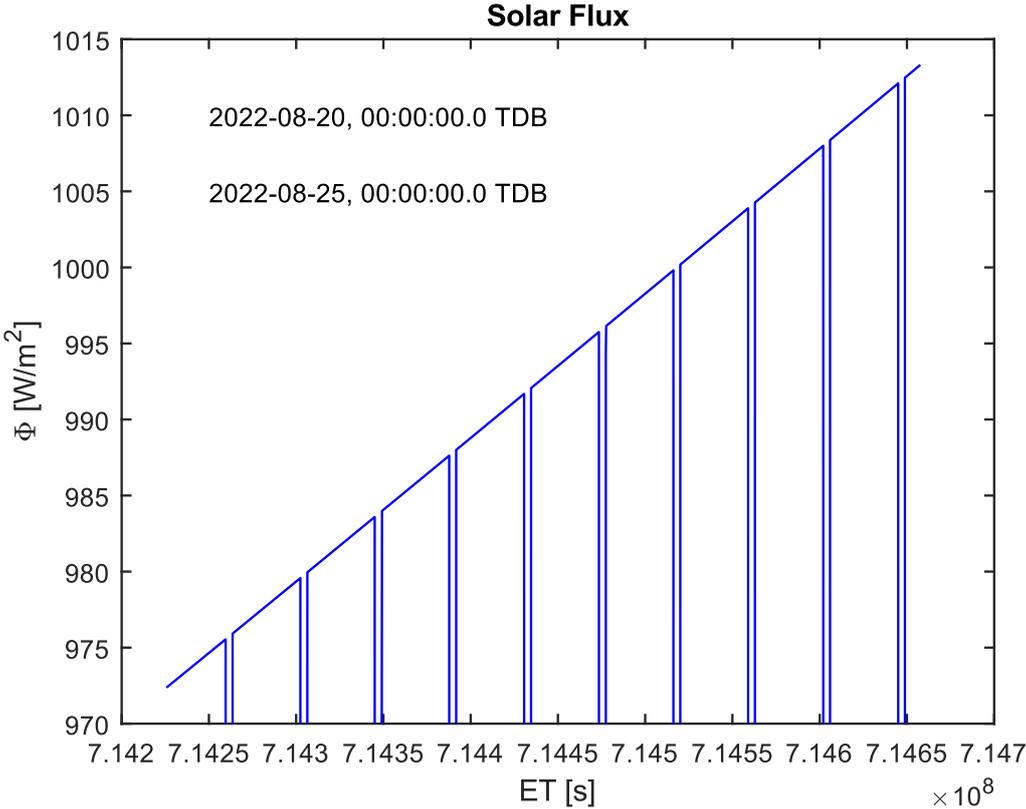


Figure 2.3: The solar flux which impinges on DustCube from 2022-08-20 to 2022-08-25

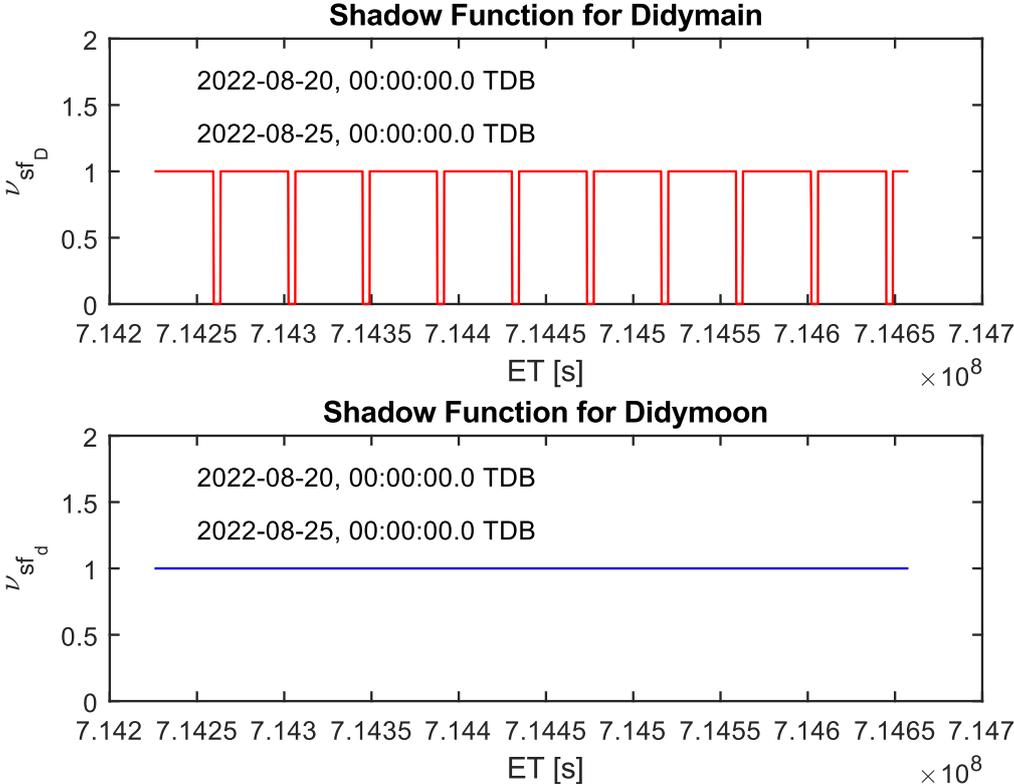


Figure 2.4: Shadow functions of Didymain and Didymoon from 2022-08-20 to 2022-08-25

3. The script *RK_3bodies_1harmonic.m* propagates the state of DustCube by separately computing the gravitational accelerations due to Didymain and Didymoon, $\mathbf{a}_{\text{sat}_{G_D}}(t_k)$ and $\mathbf{a}_{\text{sat}_{G_d}}(t_k)$, at every instant of the discretized time window, where the primary body is considered as volumetric with an actual distribution of its mass, while the secondary one as a punctual mass.

Therefore, the acceleration due to Didymoon is directly computed in DYD coordinates by exploiting the already cited *central_attraction.m* function, whereas the one due to Didymain is obtained by using the specifically created *Spherical_Harmonics.m* script, which allows the user to choose the method through which to calculate the gravitational acceleration, as it has been explained in section 1.2.3 [Gravitational attractions acted by Didymain and Didymoon](#), namely:

- the Montenbruck method, that computes the partial accelerations due to spherical harmonics in the [Didymain-fixed Reference Frame](#) (DM subscript) by exploiting the eqs (1.10) and summing the partial accelerations of eqs (1.15), that is possible through the computation of the coefficients $V_{l,m}$ and $W_{l,m}$ of eqs (1.11) and through the knowledge of the unnormalized exterior spherical harmonic coefficients $C_{l,m}$ and $S_{l,m}$ of Didymain.

The coefficients $C_{l,m}$ and $S_{l,m}$ can be uploaded in MATLAB by reading the text file *Didymos_spherical_harmonics.txt*, which contains the unnormalized spherical harmonic coefficients up to degree $l_{max} = 4$ and order $m_{max} = 4$ available from [Table 1.3](#) (Takahashi, 2013), so that the user can select the maximum value of l and m he wants to exploit, obviously with $l_{max} = m_{max}$.

Alternatively, instead of the Didymain's unnormalized coefficients $C_{l,m}$ and $S_{l,m}$ of [Table 1.3](#), is possible to exploit the normalized exterior spherical harmonic coefficients $\bar{C}_{l,m}$ and $\bar{S}_{l,m}$ as computed by Zannoni M. et al. (Zannoni, et al., 2018) up to degree 20 and order 20, by previously unnormalizing them through the inverse of eqs (1.21), even if is important to remark that this choice increases the computational cost a lot, due to the greater number of coefficients $V_{l,m}$ and $W_{l,m}$ of eqs (1.11) to be calculated.

The normalized coefficients can be uploaded in MATLAB by reading the text file *Didymos_spherical_harmonics_main.txt*, which allows the user to select the maximum value of degree and order to exploit;

- the built-in MATLAB function *gravitysphericalharmonic.m*, which computes the three components of the gravitational acceleration in the [Didymain-fixed Reference Frame](#) (DM subscript) by uploading the binary data-file *Didymain.mat*, that contains the normalized spherical harmonic coefficients up to degree 20 and order 20 (Zannoni, et al., 2018), in addition to the main info about Didymain's shape, or by uploading the binary data-file

Didymain_Taka.mat, that holds the normalization of the unnormalized coefficients of [Table 1.3](#), that is possible by exploiting the eqs (1.21), and the main info about Didymain's shape. As said in section [1.2.3 Gravitational attractions acted by Didymain and Didymoon](#), this method requires a greater computational effort to the machine than the Montenbruck one, but it is also more precise.

In both cases, since the final gravitational acceleration must be written in DYD coordinates, while the computation in *Spherical_Harmonics.m* is achieved in DM ones, the user must obtain the transformation matrix from one RF to the other to commute the position vector from DYD to DM coordinates and to obtain the final acceleration as done in eq. (1.16) in chapter 1, where we considered a S/C orbiting around the Earth, so that the partial accelerations were computed in ITRF, but the final acceleration vector would be transformed in ECI frame.

It is important to remark that, since the Didymos RF is centered in the CoM of the system and the Didymain-fixed RF in the CoM of Didymain, the *Spherical_Harmonics.m* script must also take into account the drift of the CoMs of the two reference frames (see section [A.1.5 Transformations in non-inertial coordinates by using rotation matrices](#) in Appendix).

4. The script *RK_3bodies_2harmonics.m* propagates the state of DustCube through separate computations of the gravitational accelerations due to Didymain and Didymoon, $\mathbf{a}_{\text{sat}_{G_D}}(t_k)$ and $\mathbf{a}_{\text{sat}_{G_d}}(t_k)$, at every instant of the discretized time window, by considering these bodies as volumetric with an actual distribution of their masses, i.e. by using the already cited *Spherical_Harmonics.m* script.

The concept for Didymain has been already explained in the previous point, while for Didymoon we only consider the harmonic coefficients $C_{0,0}$, $C_{2,0}$ and $C_{2,2}$, whose computations are possible by previously estimates the principal moments of inertia about the three principal axis of the volumetric body (eqs (1.22)) and then inserts these values in the eqs (1.21), finally obtaining the coefficients of [Table 1.4](#).

Therefore, the user can choose to exploit the Montenbruck method, which uses the specifically developed script *J2_C22_Moon.m*, or the built-in MATLAB function *gravitysphericalharmonic*, which exploits the script *J2_C22_Moon_normalized.m*, that uses the normalization of the coefficients $C_{0,0}$, $C_{2,0}$ and $C_{2,2}$.

Obviously, as said before, the accelerations are computed in body-fixed RFs centered in the CoMs of the two massive bodies, but the final results must be written in DYD coordinates; thus, the script requires the knowledge of the transformation matrices from Didymos RF to Didymain-fixed RF and from Didymos RF to [Didymoon-fixed RF](#).

5. The script *RK_3bodies_2harmonics_SRP.m*, propagates the state of DustCube by separately computing the gravitational accelerations due to Didymain and Didymoon, $\mathbf{a}_{\text{sat}_{G_D}}(t_k)$ and $\mathbf{a}_{\text{sat}_{G_d}}(t_k)$, considering these bodies as volumetric with an actual distribution of their masses, and by adding the acceleration due to SRP $\mathbf{a}_{\text{sat}_{SRP}}(t_k)$.

As said before, the first two are obtained by using the function *Spherical_Harmonics.m*, while the last one is computed through the script *SRP_Montenbruck_complete.m*, with or w/o shadow functions related to Didymain and Didymoon.

6. The specifically created scripts *RK_4bodies.m*, *RK_4bodies_SRP.m*, *RK_4bodies_1harmonic.m*, *RK_4bodies_2harmonics.m* and *RK_4bodies_2harmonics_SRP.m* propagate the state of DustCube as the similar functions presented in the previous points 1÷5 by adding a further contribution, which is the disturbance of the solar gravity, that has been explained in section 1.2.5 [Sun's gravitational attraction](#).

Therefore, these scripts also include the gravitational acceleration due to Sun's presence $\mathbf{a}_{\text{sat}_{G_{\text{sun}}}}(t_k)$, which can be computed at every instant of the discretized time window by exploiting the specially created MATLAB function *body_disturbance.m*, that follows the equation (1.35).

7. The script *RK_complete.m* propagates the state of DustCube by considering every contribution presented in the previous points, i.e. the gravitational accelerations due to the primaries $\mathbf{a}_{\text{sat}_{G_D}}(t_k)$ and $\mathbf{a}_{\text{sat}_{G_d}}(t_k)$, computed including the spherical harmonics, the acceleration due to SRP $\mathbf{a}_{\text{sat}_{SRP}}(t_k)$, obtained incorporating the shadow functions $\nu_{sf_D}(t_k)$ and $\nu_{sf_d}(t_k)$, the gravitational acceleration due to the Sun's presence $\mathbf{a}_{\text{sat}_{G_{\text{sun}}}}(t_k)$, and adding the gravitational accelerations due to other celestial bodies, such as the Jupiter's one $\mathbf{a}_{\text{sat}_{G_{\text{jup}}}}(t_k)$, which has an order of magnitude of about 10^{-18} [m/s²] and can be easily computable by exploiting the equation (1.35), with μ_{jup} and \mathbf{r}_{jup} that substitute μ_{sun} and \mathbf{r}_s , respectively.

Obviously, the scripts used by this implementation are the already cited *Spherical_Harmonics.m*, *SRP_Montenbruck_complete.m* and *body_disturbance.m*, where the last one can be exploited to compute the gravitational disturbances of every important celestial body different from the primaries.

The MATLAB script *DustCube_Main.m* allows the user to compute the state of DustCube through an interactive approach, i.e. requesting several info to the utilizer by showing different dialogue boxes, so that it can be easily usable from inexperienced consumers too.

Also, is possible to complete the same orbit propagations of the previous points 1÷7 by exploiting the [Didymain quasi-inertial Reference Frame](#), presented in section 1.2.1 [Reference Frames](#), which is centered in the CoM of Didymain, that coincides with its geometric center.

Obviously, since Didymain is subjected to the gravitational attraction of Didymoon, the orbit propagator considers the influence of the secondary body as a disturbance like the solar gravity one, so that its contribute on the satellite motion is computable from the equation (1.35) presented in section 1.2.5 [Sun's gravitational attraction](#).

Since the distance between Didymain and Didymoon is quite small, the influence of the secondary body's gravitational attraction on the CoM of Didymain is easy until we consider the primaries as punctual, vice versa is necessary to consider also the partial accelerations due to the spherical harmonics of Didymoon.

Therefore, the user can autonomously choose to actuate the propagation of DustCube through the scripts described in the previous points 1÷7 or through the properly created MATLAB functions related to the [Didymain quasi-inertial Reference Frame](#), which are, in particular: *RK_4bodies_Didymain.m*, *RK_4bodies_SRP_Didymain.m*, *RK_4bodies_1harmonic_Didymain.m*, *RK_4bodies_2harmonics_Didymain.m*, *RK_4bodies_2harmonics_SRP_Didymain.m* and *RK_complete_Didymain.m*.

The results, translated in [Didymos Reference Frame](#), will be the same computed through the MATLAB functions of the points 1÷7.

2.4.4 THE ORBITAL PROPAGATOR'S TESTING

The orbital propagator's functionality has been tested by comparing its results with the ones obtained with Monte Python's propagation by PhD student Riccardo Lasagni Manghi of the University of Bologna, Department of Industrial Engineering, Forlì (IT) (*Lasagni Manghi, Modenini, Zannoni, & Tortora, 2018*). Monte Python is a highly accurate software developed by the Jet Propulsion Laboratory (JPL) for modelling the dynamic of objects moving within the Solar System, so that it is routinely used for navigating NASA deep space probes.

The tests have been carried on through the MATLAB script *DustCube_orbit_evaluation.m*, which analyses step by step the error budget between DustCube's states, i.e. by considering the state computed with MATLAB propagator (MAT subscript) as the measured state and the the one obtained with Monte Python (monte subscript) as the exact state, namely:

- Trend of the absolute norm errors among the two position and velocity vectors in [m], i.e.:

$$\begin{aligned} E_{\text{pos}_{\text{ABS}}}(\mathbf{t}_k) &\stackrel{\text{def}}{=} \|\mathbf{r}_{\text{sat}_{\text{MAT}}}(\mathbf{t}_k) - \mathbf{r}_{\text{sat}_{\text{monte}}}(\mathbf{t}_k)\| \\ E_{\text{vel}_{\text{ABS}}}(\mathbf{t}_k) &\stackrel{\text{def}}{=} \|\mathbf{v}_{\text{sat}_{\text{MAT}}}(\mathbf{t}_k) - \mathbf{v}_{\text{sat}_{\text{monte}}}(\mathbf{t}_k)\| \end{aligned} \quad (2.17)$$

Chapter 2

- Trend of the absolute errors among the position and velocity vectors' components in [m], i.e.:

$$\begin{aligned}
 e_{x_{\text{ABS}}}(\mathbf{t}_k) &\stackrel{\text{def}}{=} |x_{\text{sat}_{\text{MAT}}}(\mathbf{t}_k) - x_{\text{sat}_{\text{monte}}}(\mathbf{t}_k)| \\
 e_{y_{\text{ABS}}}(\mathbf{t}_k) &\stackrel{\text{def}}{=} |y_{\text{sat}_{\text{MAT}}}(\mathbf{t}_k) - y_{\text{sat}_{\text{monte}}}(\mathbf{t}_k)| \\
 e_{z_{\text{ABS}}}(\mathbf{t}_k) &\stackrel{\text{def}}{=} |z_{\text{sat}_{\text{MAT}}}(\mathbf{t}_k) - z_{\text{sat}_{\text{monte}}}(\mathbf{t}_k)| \\
 e_{v_x_{\text{ABS}}}(\mathbf{t}_k) &\stackrel{\text{def}}{=} |\dot{x}_{\text{sat}_{\text{MAT}}}(\mathbf{t}_k) - \dot{x}_{\text{sat}_{\text{monte}}}(\mathbf{t}_k)| \\
 e_{v_y_{\text{ABS}}}(\mathbf{t}_k) &\stackrel{\text{def}}{=} |\dot{y}_{\text{sat}_{\text{MAT}}}(\mathbf{t}_k) - \dot{y}_{\text{sat}_{\text{monte}}}(\mathbf{t}_k)| \\
 e_{v_z_{\text{ABS}}}(\mathbf{t}_k) &\stackrel{\text{def}}{=} |\dot{z}_{\text{sat}_{\text{MAT}}}(\mathbf{t}_k) - \dot{z}_{\text{sat}_{\text{monte}}}(\mathbf{t}_k)|
 \end{aligned} \tag{2.18}$$

- Trend of the relative errors and, therefore, the relative percentage errors between the two position and velocity vectors, i.e.:

$$\begin{aligned}
 E_{\text{pos}_{\text{REL}}}(\mathbf{t}_k) &\stackrel{\text{def}}{=} \frac{E_{\text{pos}_{\text{ABS}}}(\mathbf{t}_k)}{\|\mathbf{r}_{\text{sat}_{\text{monte}}}(\mathbf{t}_k)\|} \\
 E_{\text{vel}_{\text{REL}}}(\mathbf{t}_k) &\stackrel{\text{def}}{=} \frac{E_{\text{vel}_{\text{ABS}}}(\mathbf{t}_k)}{\|\mathbf{v}_{\text{sat}_{\text{monte}}}(\mathbf{t}_k)\|} \\
 E_{\text{pos}_{\text{REL}\%}}(\mathbf{t}_k) &\stackrel{\text{def}}{=} E_{\text{pos}_{\text{REL}}}(\mathbf{t}_k) * 100 \\
 E_{\text{vel}_{\text{REL}\%}}(\mathbf{t}_k) &\stackrel{\text{def}}{=} E_{\text{vel}_{\text{REL}}}(\mathbf{t}_k) * 100
 \end{aligned} \tag{2.19}$$

- Maximum absolute errors in [m], maximum relative errors and maximum relative percentage errors in a single propagation, i.e.:

$$\begin{aligned}
 E_{\text{pos}_{\text{MAX-ABS}}} &\stackrel{\text{def}}{=} \max_{1 \leq k \leq n} \left(E_{\text{pos}_{\text{ABS}}}(\mathbf{t}_k) \right) \\
 E_{\text{vel}_{\text{MAX-ABS}}} &\stackrel{\text{def}}{=} \max_{1 \leq k \leq n} \left(E_{\text{vel}_{\text{ABS}}}(\mathbf{t}_k) \right) \\
 E_{\text{pos}_{\text{MAX-REL}}} &\stackrel{\text{def}}{=} \max_{1 \leq k \leq n} \left(E_{\text{pos}_{\text{REL}}}(\mathbf{t}_k) \right) \\
 E_{\text{vel}_{\text{MAX-REL}}} &\stackrel{\text{def}}{=} \max_{1 \leq k \leq n} \left(E_{\text{vel}_{\text{REL}}}(\mathbf{t}_k) \right) \\
 E_{\text{pos}_{\text{MAX-REL}\%}} &\stackrel{\text{def}}{=} \max_{1 \leq k \leq n} \left(E_{\text{pos}_{\text{REL}\%}}(\mathbf{t}_k) \right) \\
 E_{\text{vel}_{\text{MAX-REL}\%}} &\stackrel{\text{def}}{=} \max_{1 \leq k \leq n} \left(E_{\text{vel}_{\text{REL}\%}}(\mathbf{t}_k) \right)
 \end{aligned} \tag{2.20}$$

For instance, we can consider a DustCube's parking orbit in L_5 , a time window which spans from 2022-08-24 to 2022-08-28, the same initial state for the MATLAB's propagator and the Monte Python's one and two orbital evaluations depending on different contributions:

1. DustCube's motion driven by the central attractions of the primaries and the solar gravity;

2. DustCube's motion driven by the central attractions of the primaries, the solar gravity and the SRP w/o sunlight occultations.

Therefore, we can numerically compare the two computed states (measured and exact) between them in terms of trends of absolute and relative errors and graphically evaluate the shifts of the two orbits in synodic coordinates by exploiting the *DustCube_orbit_evaluation.m* script.

Here below, I will show the main results of the two evaluations:

1. Parking orbit in L_5 influenced by the central attractions of the primaries and solar gravity:

Just considering the central attractions of the primaries and the solar gravity, it is easy to verify the validity of the implemented MATLAB propagator.

Indeed, visualizing the [Figure 2.5](#), the reader can observe that the orbit achieved through it (blue line) is similar to one obtained by the Monte Python propagator (red line) and that the designed orbits comply with the theory which is at the base of the *Lagrange equilibrium points* (see section [A.2 The Circular Restricted 3-Body Problem \(CR3BP\)](#) in Appendix), since the two lines remain always close to the L_5 libration point.

Also, the reader can observe the absolute errors trends and relative percentage error trends related to position and velocity inertial computations from [Figure 2.6](#) and [Figure 2.7](#).

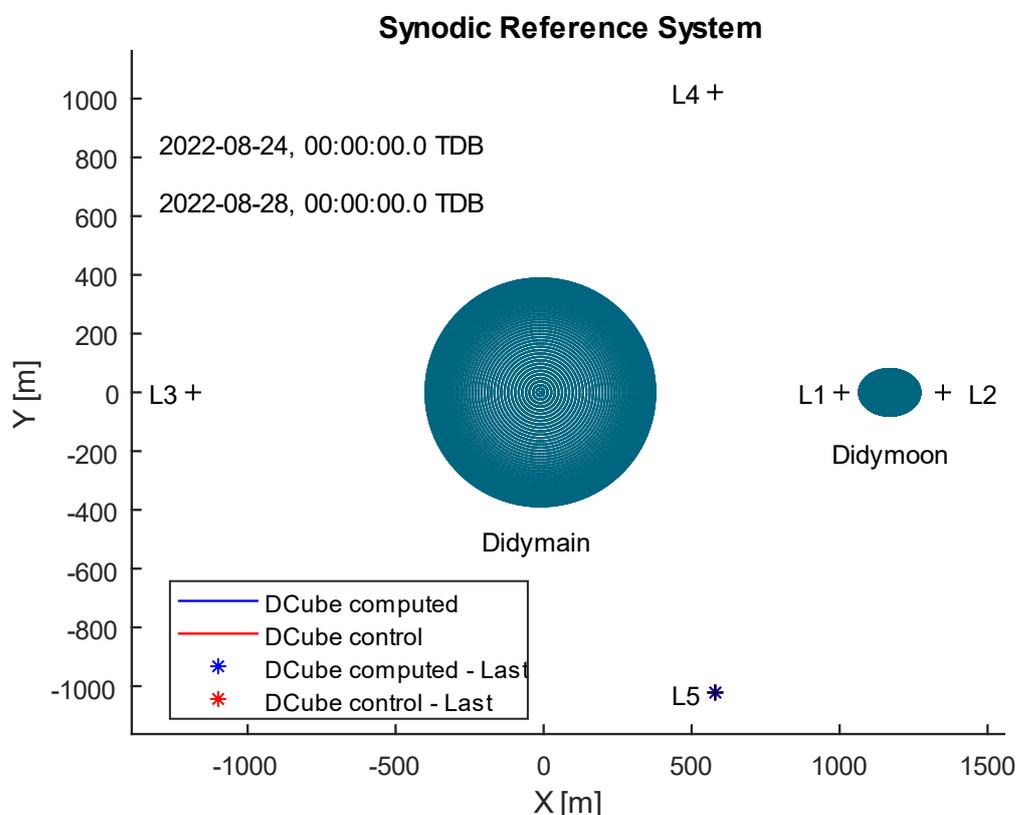


Figure 2.5: Plot of two similar parking orbits in L_5 obtained through the MATLAB propagator (blue line) and the Monte Python one (red line), both subjected to central attractions of the primaries and solar gravity

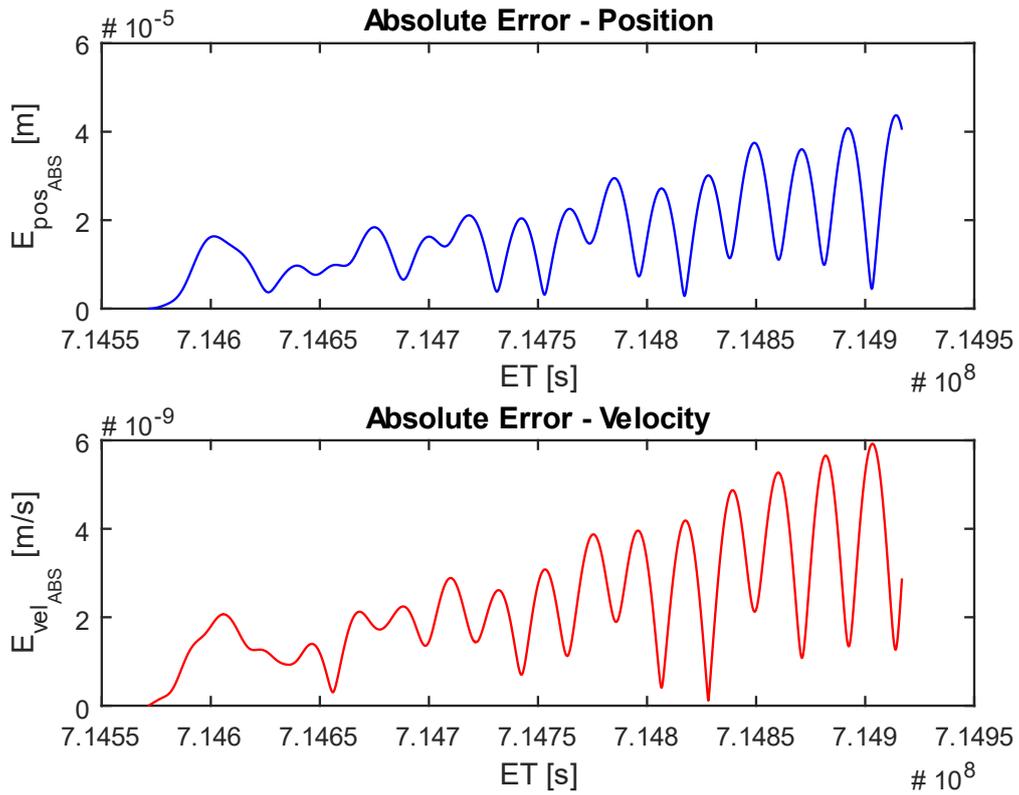


Figure 2.6: Absolute error trends for the two orbits of Figure 2.5

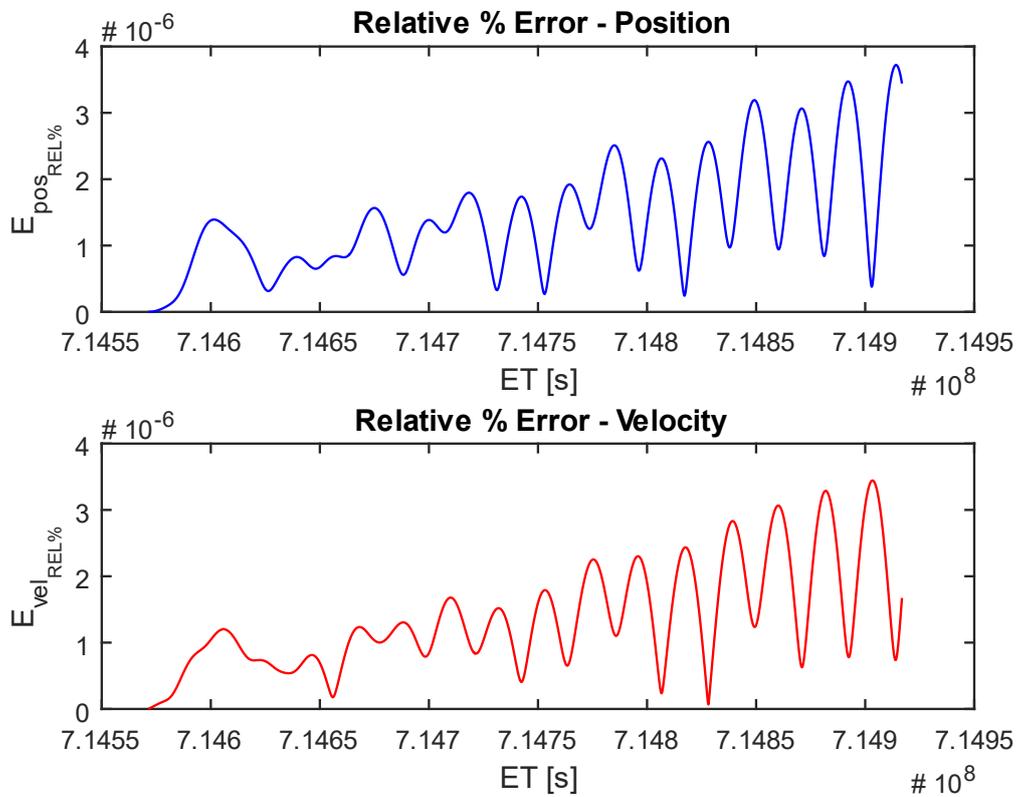


Figure 2.7: Relative percentage error trends for the two orbits of Figure 2.5

From the first image, it is easy to verify that, even if the absolute errors grow in time, they remain very small, i.e. in the order of about 10^{-5} [m] for the DustCube's position and 10^{-9} [m/s] for the DustCube's velocity in Didymos RF, so that it can be finally concluded that the two computed states practically coincide, which proves the reliability and the accuracy of the MATLAB propagator.

From the second figure, it is possible to see that the orders of magnitude of the relative percentage errors are of about 10^{-6} % for both DustCube's position and velocity, so that they confirm the accuracy of the MATLAB propagator.

Finally, the MATLAB script creates the Excel file *Errors.xlsx*, which contains the most important features of the propagation accomplished and the maximum absolute, relative and relative percentage errors, which are useful to evaluate the quality of the propagator.

Therefore, the Excel folder for the considered parking orbit in L_5 of [Figure 2.5](#), with a time window which spans from 2022-08-24 to 2022-08-28 and the same initial state for the MATLAB's propagator and the Monte Python's one, is shown in [Figure 2.8](#).

MISSION DESCRIPTION			
Celestial Body:	Didymos		
Mission Name:	DustCube		
Case Study:	L5 - Central attractions of Didymain and Didymoon + solar gravity disturbance		
Reference Frame:	Didymos RF		
Initial Time:	2022-08-24, 00:00:00.000 TDB		
Final Time:	2022-08-28, 00:00:00.000 TDB		
Time Step:	h = 10 s		
Number of Time Steps:	N = 34560		
Initial Position:	x0 = -990,92647 m	y0 = 630,82448 m	z0 = 0,05459 m
Initial Velocity:	Vx0 = -0,09233 m/s	Vy0 = -0,14509 m/s	Vz0 = 0,00000 m/s
TABLE	Max Absolute Error	Max Relative Error	Max Relative % Error
Position	4,37271E-05 m	3,72151E-08	0,0000037 (%)
Velocity	5,92249E-09 m/s	3,44393E-08	0,0000034 (%)

Figure 2.8: Max absolute, relative and relative percentage errors for the orbits of [Figure 2.5](#)

2. Parking orbit in L_5 influenced by the central attractions of the primaries, solar gravity and SRP w/o sunlight occultations:

Since the Solar Radiation Pressure causes a DustCube's acceleration in the order of magnitude of about $10^{-8} \div 10^{-9}$ [m/s²], the absolute and relative errors are a little bigger than ones obtained with the only accelerations due to central attractions of the primaries and solar gravity.

The [Figure 2.9](#) allows the reader to observe that the orbit achieved through the MATLAB propagator (blue line) is very similar to one obtained by the Monte Python one (red line), to such an extent that

the two orbits appear as overlapped, which again proves the reliability and the accuracy of the MATLAB propagator.

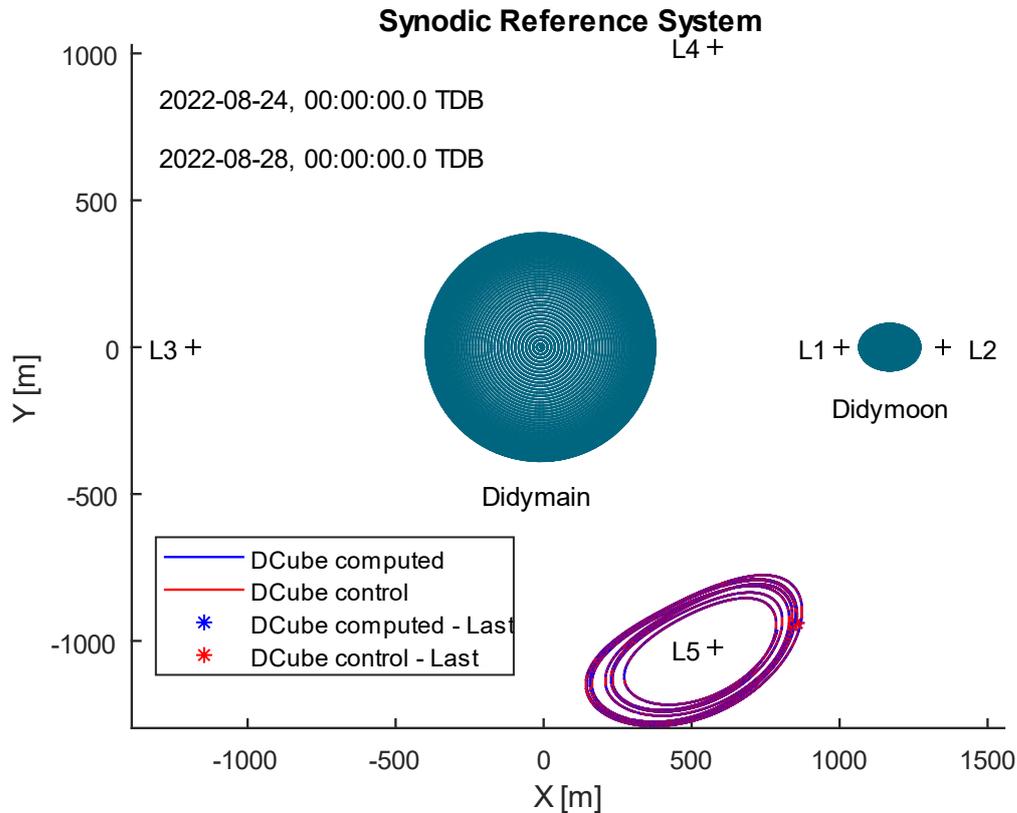


Figure 2.9: Plot of two similar parking orbits in L_5 , obtained by the MATLAB propagator (blue) and the Monte-Python one (red), due to central attractions of the primaries, solar gravity and SRP w/o sunlight occultations

Finally, the MATLAB script creates the Excel file *Errors.xlsx*, which contains the new maximum absolute, relative and relative percentage errors of the considered parking orbit in L_5 of Figure 2.9, with a time window which spans from 2022-08-24 to 2022-08-28 and the same initial state for the MATLAB's propagator and the Monte Python's one, which is shown in Figure 2.10.

It is important to remark that the approximated shape of the satellite, i.e. the flat plate, even if it is useful to easily compute the force due to SRP, creates some problems when the angle θ between \hat{n} and \hat{s} (see Figure 1.7 and eq. (1.23) in chapter 1) approaches the right angle, i.e. when the illuminated area almost vanishes, so that it increases the absolute errors, which, however, remain very small, as can be observed from Figure 2.10, which states that the orders of magnitude settle around 10^{-3} [m] for the position evaluation and 10^{-5} [m/s] for the velocity one.

MISSION DESCRIPTION			
Celestial Body:	Didymos		
Mission Name:	DustCube		
Case Study:	L5 - Central attractions of Didymain and Didymoon + SRP + solar gravity disturbance		
Reference Frame:	Didymos RF		
Initial Time:	2022 AUG 24 00:00:00.0		
Final Time:	2022 AUG 28 00:00:00.0		
Time Step:	h = 10 s		
Number of Time Steps:	N = 34560		
Initial Position:	x0 = -1148.89821 m	y0 = 479.59948 m	z0 = -31.78199 m
Initial Velocity:	Vx0 = -0.07749 m/s	Vy0 = -0.14276 m/s	Vz0 = -0.00186 m/s
TABLE	Max Absolute Error	Max Relative Error	Max Relative % Error
Position	8.09845E-03 m	6,08710E-06	0.0006087 (%)
Velocity	8.20153E-05 m/s	5,49964E-04	0.0549964 (%)

Figure 2.10: Max absolute, relative and relative percentage errors for the orbits of Figure 2.9

3 OPTICAL NAVIGATION

The optical navigation is the navigation technique selected for the mission.

The goal of the optical navigation prediction process is to determine the expected coordinates of the image of some sort of target within a picture taken by a camera. A target can be anything. In traditional optical navigation, targets typically are the coordinate centers of nearby Solar System objects: planetary satellites, asteroids, comets, and occasionally the planets themselves. Targets may also include features or landmarks. on the surface of a body, even another spacecraft. Stars are also considered targets, because it is important to measure their images too.

The first goal is to compute the inertial direction of the S/C with respect to a target, while the second goal is to evaluate the attitude of the camera, and so the attitude of the S/C.

3.1 INERTIAL DIRECTION EVALUATION: LINES OF SIGHT (LOS)

As it is described in Polle B., et al., *Autonomous Navigation for Interplanetary Missions Performance Achievements Based on Real and Flight Images*, presented in the 6th International ESA Conference on Guidance, Navigation and Control Systems, held 17th – 20th October 2005 in Loutraki, Greece (Polle, et al., 2005, October 17-20):

“Interplanetary vision-based navigation relies on observation of the line of sight (LOS) of close objects with known position to infer the vehicle position.”

The Line of Sight is the inertial (I subscript) geometric relative position $\mathbf{r}_{M-sat}(t)|_I$ between the camera mounted on S/C and the target, that can be evaluated as:

$$\mathbf{r}_{M-sat}(t)|_I = \mathbf{r}_M(t)|_I - \mathbf{r}_{sat}(t)|_I \quad (3.1)$$

Where $\mathbf{r}_M(t)|_I$ and $\mathbf{r}_{sat}(t)|_I$ are the inertial positions of the target and the S/C, respectively, w.r.t. the barycentre of the planetary system.

For a more realistic representation of the vector $\mathbf{r}_{M-sat}(t)|_I$, we must consider the light time τ , which is defined as the time required for light to travel from the target to the camera.

Therefore, the true position of the target $\mathbf{r}_{M-sat}'(t)$ can be computed as (omitting the RF):

$$\mathbf{r}_{M-sat}'(t) = \mathbf{r}_M(t - \tau) - \mathbf{r}_{sat}(t) \quad (3.2)$$

The light time is an important datum when the distances between the targets and the S/C are considerable. Since DustCube will be immersed in a binary system with relative positions in the order

of $10^2 - 10^3$ meters, it is possible to adopt the choice of neglecting the light time term in the next computations.

There is another effect that has to be considered, which is the stellar aberration, that shifts the direction toward the velocity apex of the camera and, therefore, brings to compute the apparent position $\mathbf{r}_{M-sat}''(t)$ of the target with respect to the S/C by taking into account the velocity of the camera $\mathbf{v}_{cam}(t)$:

$$\begin{aligned} \mathbf{r}_{M-sat}''(t) &= \mathbf{r}_{M-sat}'(t) + \|\mathbf{r}_{M-sat}'(t)\| \frac{\mathbf{v}_{cam}(t)}{c} = \\ &= [\mathbf{r}_M(t - \tau) - \mathbf{r}_{sat}(t)] + \|\mathbf{r}_M(t - \tau) - \mathbf{r}_{sat}(t)\| \frac{\mathbf{v}_{cam}(t)}{c}, \end{aligned} \quad (3.3)$$

where c is the speed of light.

This equation represents the Newtonian formulation for the inertial relative position of the target with respect to the camera and is commonly exploited in optical navigation missions.

Since for DustCube mission the light time can be omitted, the final expression of $\mathbf{r}_{M-sat}''(t)$ is:

$$\mathbf{r}_{M-sat}''(t) = [\mathbf{r}_M(t) - \mathbf{r}_{sat}(t)] + \|\mathbf{r}_M(t) - \mathbf{r}_{sat}(t)\| \frac{\mathbf{v}_{cam}(t)}{c} \quad (3.4)$$

Also, in first approximation for DustCube mission it is possible to omit the second term of the previous formula, since its value is in the order of 10^{-7} .

Therefore, the inertial relative positions of Didymain and Didymoon with respect to DustCube can be finally considered as the simple relations:

$$\begin{aligned} \mathbf{r}_{D-sat}(t)|_I &= \mathbf{r}_D(t)|_I - \mathbf{r}_{sat}(t)|_I \\ \mathbf{r}_{d-sat}(t)|_I &= \mathbf{r}_d(t)|_I - \mathbf{r}_{sat}(t)|_I \end{aligned} \quad (3.5)$$

Since for DustCube mission the Didymos RF can be considered as quasi-inertial, the eqs (3.5) become the opposite vectors of the eqs (1.2) of chapter 1 by substituting the subscript I with DYD one in eqs (3.5), so that we have demonstrated the validity of the approximations considered in chapter 1, where we had omitted the stellar aberration.

In general, the relative position vectors $\mathbf{r}_{D-sat}(t)|_{DYD}$ and $\mathbf{r}_{d-sat}(t)|_{DYD}$ can be affected by errors which have to be considered for the real-time evaluation of the images (Figure 3.1).

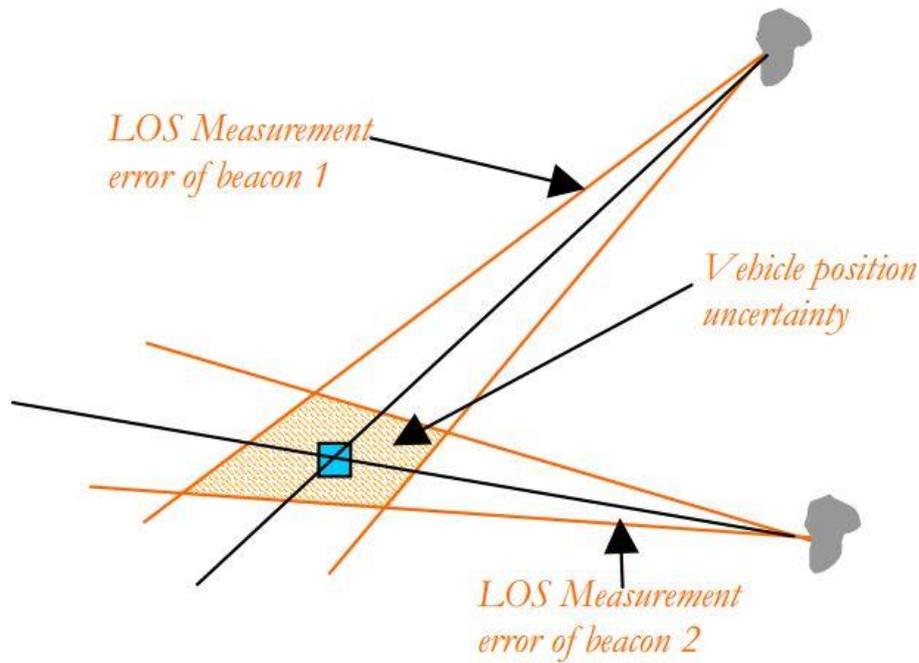


Figure 3.1: Principle of 3D localization using two beacons LOS measurement (Polle, et al.)

3.2 ATTITUDE EVALUATION

The second goal of the optical navigation is to compute the attitude of the camera and, consequently, the attitude of the S/C.

The transformation from inertial coordinates to camera coordinates is computed by a rotation matrix that is related to a right-handed coordinate system produced by M , N and L axes (Figure 3.2), where it is represented the simple schematization of the optical process for acquiring images of an ideal pinhole camera.

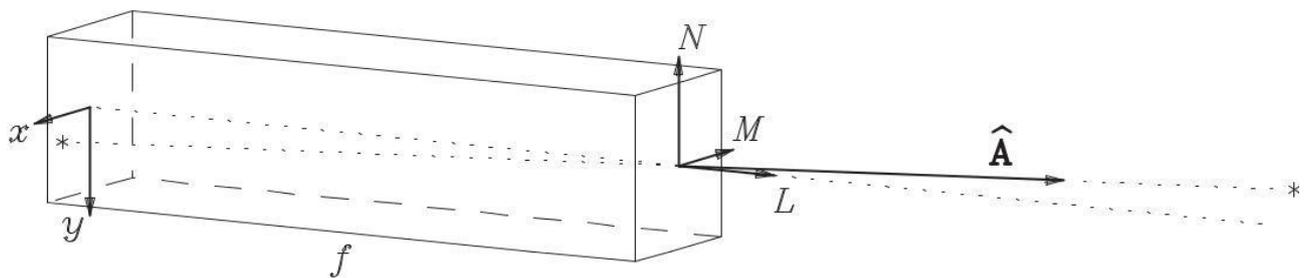


Figure 3.2: Principles of ideal pinhole camera. $\hat{\mathbf{a}}$ is the gnomonic projection for a star (Owen, *Methods of Optical Navigation*, 2011, February 14)

The objective is to find a relation that expressed the attitude of the camera with respect to the inertial tracking.

Considering the unit vector $\hat{\mathbf{a}}$ pointed towards the star, that is a fixed point in space, it is possible to transform this unit vector from *Inertial Coordinates (I)* to *Camera Coordinates (C)*:

$$\hat{\mathbf{a}}_C = \mathbf{C} \hat{\mathbf{a}}_I, \quad (3.6)$$

where \mathbf{C} is a rotation matrix which can be defined in the simplest way as a classical 3-1-3 sequence of elementary rotations, i.e.:

$$\mathbf{C} = \mathbf{R}_3(\varphi) \mathbf{R}_1\left(\frac{\pi}{2} - \delta\right) \mathbf{R}_3\left(\frac{\pi}{2} + \alpha\right), \quad (3.7)$$

where:

- α is the right ascension of the optical axis L ;
- δ is the declination of the optical axis L ;
- φ is a generic angle of rotation.

The expression (3.7) is useful iff the three angles are measurable and/or controllable.

There are other useful formulations to feature the matrix \mathbf{C} , which, in any case, expresses the attitude of the camera and, consequently, the attitude of S/C.

3.3 IMAGE ACQUIRING PROCESS

An ideal pinhole camera acts as the camera aperture was a point (Figure 3.2), that is also the centre of the coordinate frame (M, N, L).

The light rays coming from the target, such as a star, enter in the aperture and keep on until they encounter the detector, suited in a focal plane at a distance f from the aperture, called focal length.

An inverted image of the target appears on the detector (det subscript) in coordinates $(x_{\text{det}}, y_{\text{det}})$.

Since $\hat{\mathbf{a}}_C$ is the apparent position unit vector of the target, which components are (a_C^1, a_C^2, a_C^3) , the gnomonic projection of this one is:

$$\begin{bmatrix} x_{\text{det}} \\ y_{\text{det}} \end{bmatrix} = \frac{f}{a_C^3} \begin{bmatrix} a_C^1 \\ a_C^2 \end{bmatrix}, \quad (3.8)$$

where $(x_{\text{det}}, y_{\text{det}})$ and f are in units of length.

The real cameras differ from ideal ones, since several aberrations must be considered.

For the errors in position location of the images, the most important aberrations are radial distortions and tip and tilt misalignments (*Owen, Methods of Optical Navigation, 2011, February 14*), which must be taken into account by the corrected image coordinates, i.e.:

$$\begin{bmatrix} x_{\text{det}}' \\ y_{\text{det}}' \end{bmatrix} = \begin{bmatrix} x_{\text{det}} \\ y_{\text{det}} \end{bmatrix} + \begin{bmatrix} \Delta x_{\text{det}} \\ \Delta y_{\text{det}} \end{bmatrix} = \begin{bmatrix} x_{\text{det}} \\ y_{\text{det}} \end{bmatrix} + \begin{bmatrix} x_{\text{det}} r_{\text{det}}^2 & x_{\text{det}} y_{\text{det}} & x_{\text{det}}^2 \\ y_{\text{det}} r_{\text{det}}^2 & y_{\text{det}}^2 & x_{\text{det}} y_{\text{det}} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}, \quad (3.9)$$

where $r_{\text{det}}^2 = x_{\text{det}}^2 + y_{\text{det}}^2$ and $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \varepsilon_3)$ is the error vector which comprises radial distortions and tip and tilt misalignments; the other aberrations are not considered, since they affect the shape but not the position of the image.

The data used by optical navigation are digital pictures, independently by the camera exploited.

A digital picture is an array of data numbers or DN values and every DN measures the amount of light falling on a specific region of a detector (a pixel, acronyms of “picture element”) at the focal plane of a camera (Owen, *Methods of Optical Navigation*, 2011, February 14).

The images are captured by the camera in pixel coordinates s_p and l_p , where s_p is the “sample” and l_p is the “line” of the pixels’ plane, therefore the image processing consists of extracting the (s_p, l_p) coordinates of an image of a target within a picture.

In other words, the samples are the columns and the lines are the rows of the digital picture, where the first column is displayed at the left side of the picture and the first line is displayed at the top or at the bottom, differing from one device to the other.

The first pixel differs in representation between systems too, so it has coordinates (0,0) or (1,1).

The images are translated in units of length making use of the coordinates $(x_{\text{det}}', y_{\text{det}}')$, that are usually measured in millimetres.

The transformation in pixel coordinates is possible by exploiting a simple formula, i.e.:

$$\begin{bmatrix} s_p \\ l_p \end{bmatrix} = \boldsymbol{\Theta} \begin{bmatrix} x_{\text{det}}' \\ y_{\text{det}}' \end{bmatrix} + \begin{bmatrix} s_{p_0} \\ l_{p_0} \end{bmatrix} = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix} \begin{bmatrix} x_{\text{det}}' \\ y_{\text{det}}' \end{bmatrix} + \begin{bmatrix} s_{p_0} \\ l_{p_0} \end{bmatrix}, \quad (3.10)$$

where $\boldsymbol{\Theta}$ is a square matrix that links the pixel dimensions with the length dimensions and its components are measured in pixels/mm, (s_{p_0}, l_{p_0}) are the coordinates of the optical axes L , that has coordinates $(x_{\text{det}}, y_{\text{det}}) = (0,0)$ by definition.

Iff the ideal detector has perfectly square pixels, the matrix $\boldsymbol{\Theta}$ is diagonal with $\Theta_{11} = \Theta_{22}$.

The measured data are the pixel scales in radians/pixel, which, ignoring the distortions, can be featured like:

$$\begin{aligned} S_{s_p} &= \frac{\Theta_{11}}{f} \\ S_{l_p} &= \frac{\Theta_{22}}{f} \end{aligned} \quad (3.11)$$

Obviously, it is important to know the measured size of a single pixel for the camera in use.

Finally, we must pay attention for the rotations, since a rotated image could depend on the proper attitude of the camera (matrix \mathbf{C}) or on the construction of it (matrix $\mathbf{\Theta}$).

However, what is actually gauged by the detector is the incident intensity of the incoming light, that is defined, in general, as a measure of the luminous energy which reaches the pixels.

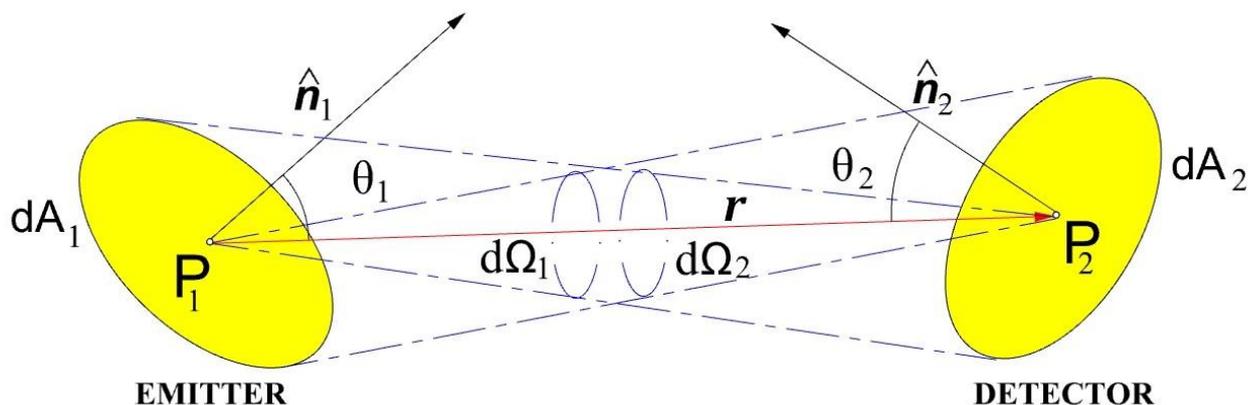


Figure 3.3: Principles of emission and detection of light

Considering the [Figure 3.3](#), we can define the infinitesimal energy dE_1 which leaves the emitter, depicted by the point P_1 , and the infinitesimal one dE_2 that reaches the detector, represented by the point P_2 , like, respectively:

$$\begin{aligned} dE_1 &\stackrel{\text{def}}{=} I(\cos(\theta_1), \nu, t) \cos(\theta_1) dA_1 d\Omega_1 d\nu dt \\ dE_2 &\stackrel{\text{def}}{=} I(\cos(\theta_2), \nu, t) \cos(\theta_2) dA_2 d\Omega_2 d\nu dt \end{aligned} \quad (3.12)$$

where:

- I is the specific or radiative intensity of the light $\left[\frac{\text{J}}{\text{m}^2 \cdot \text{sr} \cdot \text{Hz} \cdot \text{s}} \right]$;
- θ_1 is the angle between \hat{n}_1 and \mathbf{r} , while θ_2 is the one between \hat{n}_2 and \mathbf{r} [rad];
- dA_1 and dA_2 are, respectively, the infinitesimal areas that contain P_1 and P_2 ;
- $d\Omega_1$ and $d\Omega_2$ are the infinitesimal solid angles, measured in steradians [sr];
- $d\nu$ is the infinitesimal frequency [Hz];
- dt is the infinitesimal time [s].

Chapter 3

Finally, the principle of the invariance says that, if there is no matter between the emitter and the detector, such as in vacuum, the specific intensity I is the same for the emitter area and the detector one, so that dE_2 becomes:

$$dE_2 = I(\cos(\theta_1), \nu, t) \cos(\theta_2) dA_2 d\Omega_2 d\nu dt \quad (3.13)$$

In general, the specific intensity I is a function of the frequency, the time and the cosine of the angle θ , or, more in general, the position of the emitter, so that $I = I(\alpha, \delta, \nu, t)$, where α and δ are the right ascension and the declination of the inertial target position (the emitter P_2).

Also, the camera introduces an attenuation function $\epsilon(\nu)$, mainly caused by the filter, and the modern detectors convert some fraction of the incident photons in photoelectrons, called quantum efficiency $q(\nu)$.

The incident intensity of the incoming light is modulated by the camera and captured by the detector in the focal plane, so that, by using the aperture of the camera A_p , the signal measured by a single pixel in number of photoelectrons can be expressed as (*Owen, Methods of Optical Navigation, 2011, February 14*):

$$Y(s_p, l_p) = A_p \iiint I(\alpha, \delta, \nu, t) \epsilon(\nu) q(\nu) d\nu d\Omega dt, \quad (3.14)$$

Finally, the digital number (DN) value is directly dependent on the incident intensity, i.e.:

$$DN(s_p, l_p) = \frac{Y(s_p, l_p)}{g} + b, \quad (3.15)$$

where g is a gain measured in [electrons/DN] related to the analog-to-digital converter and b is a bias which is usually constant, useful to avoid negative voltages emanating from the detector. Obviously, the DN values are subjected to various sources of noises, which treatment goes beyond the topics of this work of thesis.

3.4 IMAGE PROCESSING TECHNIQUES

Since the goal of the optical navigation is to compute the inertial position and the attitude of the camera mounted on a S/C, it is important to select the proper image processing techniques for the mission features.

Usually, two phases of the vehicle's trajectory are distinguished:

- **Cruise Phase:** S/C is far away from massive bodies and the camera can observe several distant targets with a quite good separation angle. In this case the main issue is to perform accurate LOS measurements to compensate the large S/C-beacon distance separation, so it's preferable to compute the LOS with respect to the background stars simultaneously, since they can be considered as fixed in space;
- **Encounter Phase:** the beacons are generally bright, so it cannot possible to measure LOS and background stars at the same time. In this case, stars are measured by Star trackers and LOS can easily be measured by small instruments.

Obviously, the phase at which the S/C operates determines the best image processing technique, together with the mission purposes. The image processing consists of extracting the pixel coordinates (s_p, l_p) of the image of the target within a picture.

The most important techniques used in space missions *are* (Owen, *Methods of Optical Navigation*, 2011, February 14), (Polle, *et al.*, 2005, October 17-20):

- 1. Punctual direction by MTI:** in case of the target is so far that the beacon is punctual and very weak, such as its brightness is lower than the stars in the background, so that it is difficult to detect the target in a single image with an exposure time similar to one useful for tracking of stars in the background, Multiple Time Integration (MTI) technique is exploited, acquiring multiple images of several successive frames and obtaining the LOS measurement in post-integration (Figure 3.4).

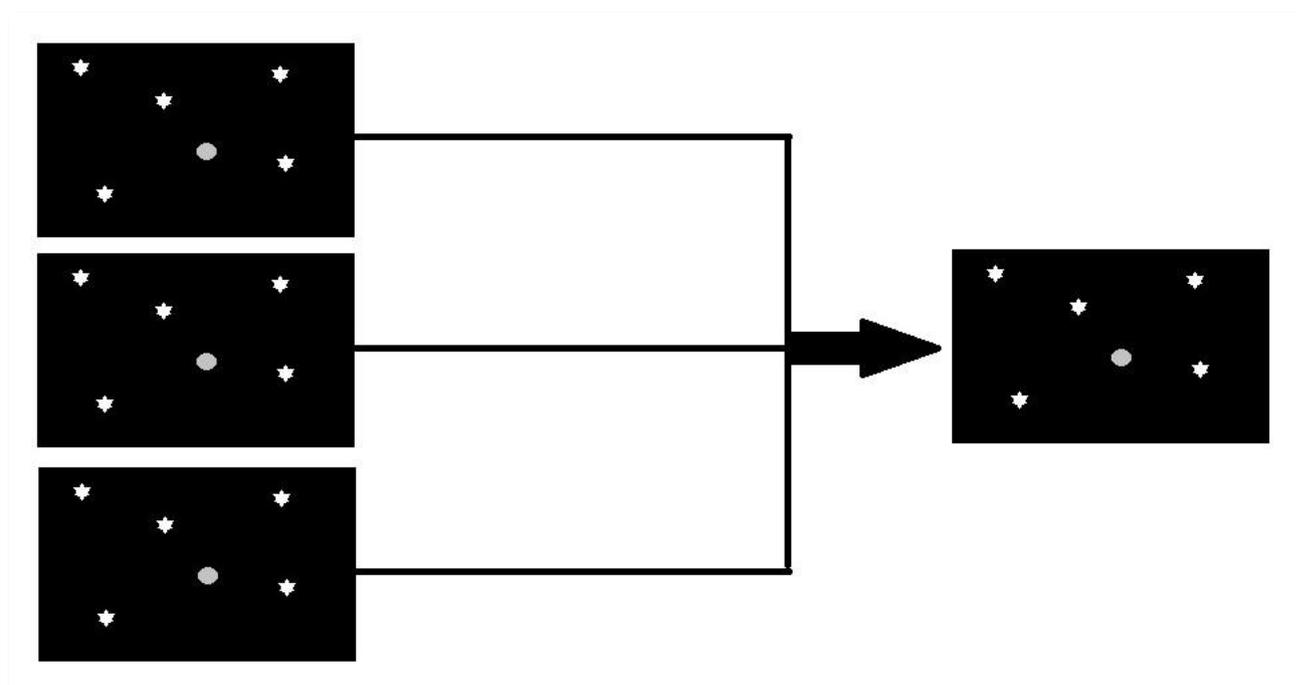


Figure 3.4: LOS measurement acquisition using MTI

- 2. Punctual direction using star technique:** in case of the target is far so that the beacon is punctual, but its brightness is similar to one of the stars in the background, the LOS measurement is obtained in the same way of acquisition of the picture for the stars in the background (Figure 3.5).

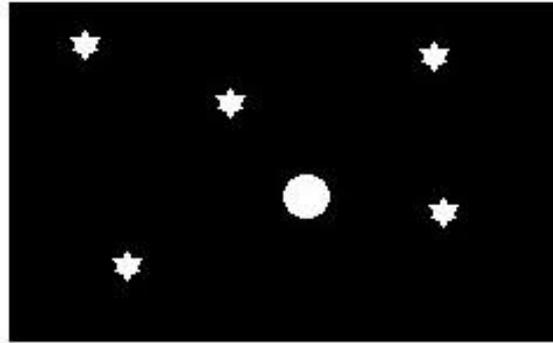


Figure 3.5: LOS measurement like stars in the background

- 3. Center-finding by Correlation:** when the target is contained in a few pixels of the digital picture (Figure 3.6), the target image contained within a picture is defined by computing the Center of Brightness (CoB) with phase effect compensation using a model, that is correlating the actual image of the target with the digital model picture and reconstructing the brightness of the target, thus determining the LOS measurement directed towards the target body. Considering an observed amount of DN values (an array of DN values) included in a portion of the picture that contains the image of the target and a predicted amount of DN values, where the predicted image is known a-priori by construction, the goal is to correlate the brightness sample in the second one within the first one;

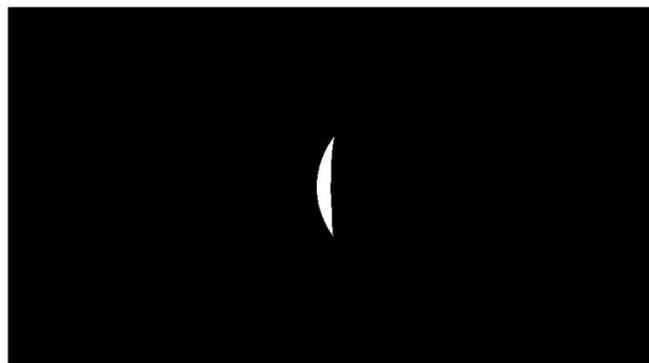


Figure 3.6: CoM achievement using CoB determination technique

4. **Center-finding by Analytic Function Fitting:** it consists to recover the Center of Mass (CoM) position of the target body (Figure 3.7). This technique is useful when the beacon is extended, but the image of the target is completely content in camera FoV (Field of View). The recovery of the CoM is achieved fitting an ellipse on the limb contour and comparing the image with a planet model, so that the shape of the planet must be known a-priori. The idea is to fit an analytic brightness function to the DN array, where each DN value is an observation. Then a least-squares solution determines the values of the parameters of the fitting function. Since the fitting function is not linear, the process is iterative;

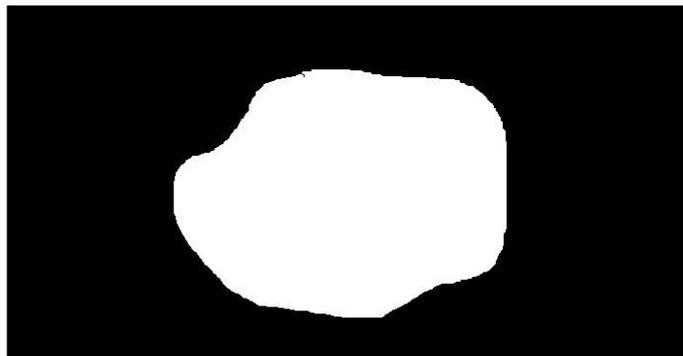


Figure 3.7: CoM achievement using analytic function fitting technique

5. **Limb Scanning:** it consists to determine the limb of a body target, that is the set of points that form the boundary of that body when it is projected into an image, comparing the image with models previously uploaded (Figure 3.8).

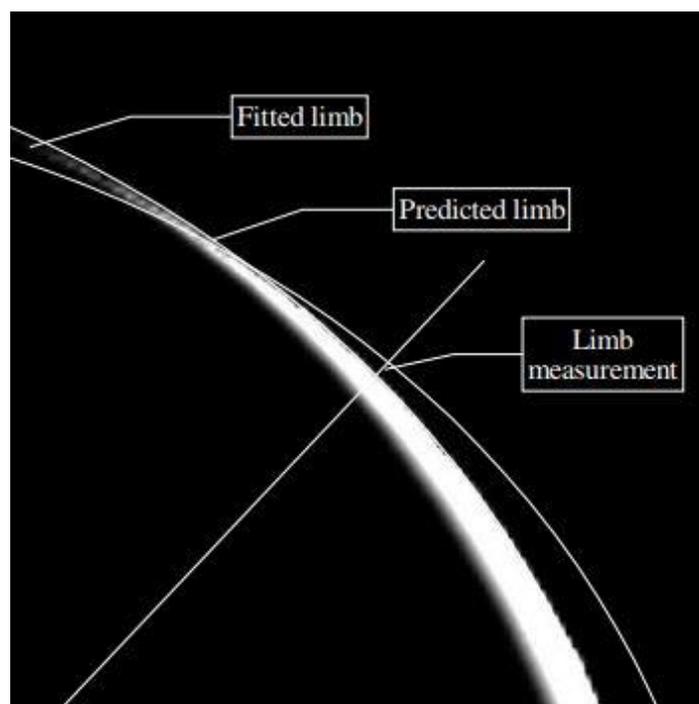


Figure 3.8: Limb measurement principle (Polle, et al., 2005, October 17-20)

Chapter 3

This technique is useful when the image is extended and is not content in camera FoV.

If we consider the target as a triaxial ellipsoid (Figure 3.9) with principal dimensions (a, b, c) and a body-fixed RF (b subscript) oriented as its three principal axes, the equation of the triaxial ellipsoid is:

$$\frac{x_b^2}{a^2} + \frac{y_b^2}{b^2} + \frac{z_b^2}{c^2} = 1 \quad (3.16)$$

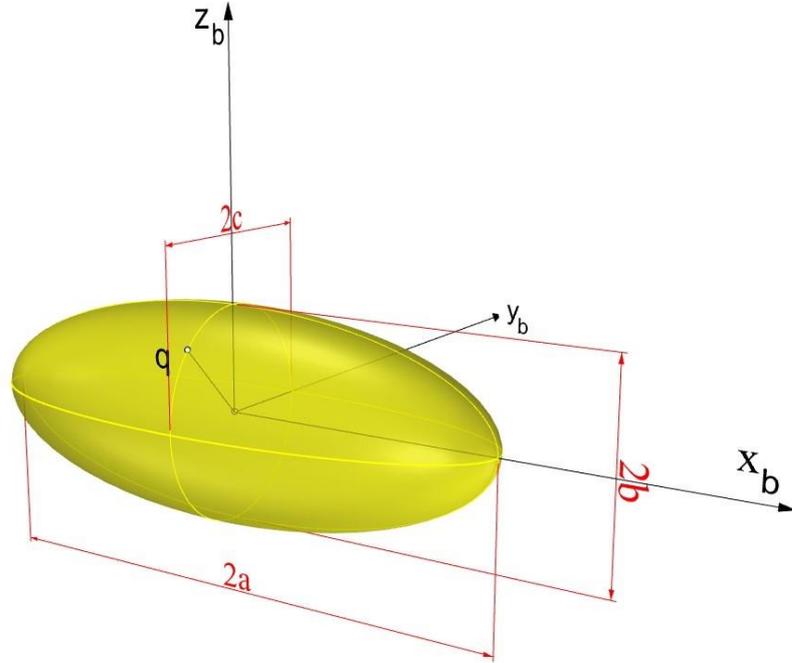


Figure 3.9: Target viewed as an ellipsoid

Defining the diagonal matrix $\Gamma \stackrel{\text{def}}{=} \text{diag}\left(\frac{1}{a^2}, \frac{1}{b^2}, \frac{1}{c^2}\right)$ and \mathbf{q} as a generic surface point of the triaxial ellipsoid, such that \mathbf{q} is a generic vector from the center of the body to \mathbf{q} , the equation (3.16) becomes:

$$\frac{x_b^2}{a^2} + \frac{y_b^2}{b^2} + \frac{z_b^2}{c^2} = \mathbf{q}^T \Gamma \mathbf{q} = 1 \quad (3.17)$$

Therefore, it is possible to define the points on the surface of the ellipsoid by using the scalar equation (3.17).

Also, the camera's direction is perpendicular to the superficial normal.

Indeed, the limb is the set of the points that form the boundary of the projection of the body into an image, so that, since the camera's direction is perpendicular to the superficial normal, called \mathbf{n} , considering \mathbf{u} as the vector from the camera to the center of the body, we can define the limb as the set of points satisfying the following equation:

$$(\mathbf{u} + \mathbf{q})^T \mathbf{n} = 0, \quad (3.18)$$

which is a scalar product that represents the equation of a plane.

Therefore, since it is possible to demonstrate that $\mathbf{n} = \Gamma \mathbf{q}$, so that $\mathbf{q}^T \mathbf{n} = 1$ from eq. (3.17), the relation (3.18) becomes:

$$\mathbf{u}^T \mathbf{n} = \mathbf{u}^T \Gamma \mathbf{q} = -\mathbf{q}^T \mathbf{n} = -1 \quad (3.19)$$

Also, considering the image of the target as captured by the detector (Figure 3.10), there is a scan vector $\hat{\mathbf{m}}$ that starts from the center of the image $\mathbf{c}_0 = (x_0, y_0, f)$ in camera coordinates so that the limb points satisfies:

$$(\mathbf{c}_0 \times \hat{\mathbf{m}})^T (\mathbf{u} + \mathbf{q}) = 0 \quad (3.20)$$

Where $\hat{\mathbf{m}} = (\cos(\theta), \sin(\theta), 0)$, with θ that is defined as the scan angle.

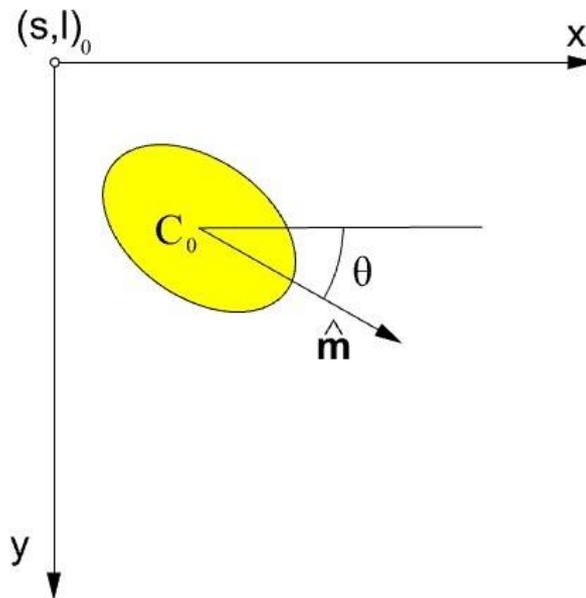


Figure 3.10: Scanning process

Finally, the limb points, represented by the vector \mathbf{q} , satisfy the three scalar equations (3.17), (3.19) and (3.20), thus defining a system of three scalar equations:

$$\begin{cases} \mathbf{q}^T \Gamma \mathbf{q} = 1 \\ \mathbf{u}^T \Gamma \mathbf{q} = -1 \\ (\mathbf{c}_0 \times \hat{\mathbf{m}})^T (\mathbf{u} + \mathbf{q}) = 0 \end{cases} \quad (3.21)$$

3.5 OPTICAL NAVIGATION CONCEPT FOR DUSTCUBE MISSION

Since the selected approach for DustCube navigation concept must consider the strong constraint in terms of mass, volume and available computing power, the choice for the image processing technique falls into the center-finding by analytic function one (4) to acquire LOS measurements, complemented by limb scanning one (5) when unfavourable geometric configurations occur (*Lasagni Manghi, Zannoni, Modenini, & Tortora, 2016, February 2*).

Obviously, LOS measurement alone does not give information about the distance from the reference body, therefore the ranges from Didymain and Didymoon must be computed by exploiting the binary nature of Didymos system, i.e. by continuously taking images of these bodies and by correlating the size of the target within the image with the known size of the corresponding celestial body (*Christian, 2015*).

The accuracy of this procedure is dependent on several factors: the ability to determine the exact CoB, the resolution of the cameras, the distance of Didymain and Didymoon and the knowledge of the heliocentric orbit of Didymos.

Since the configuration of the binary system does not create so many problems in terms of brightness of the detecting objects, the center-finding technique can be implemented in cooperation with stars background evaluation, that allows to acquire the LOS measurements with good precision, because the positions of the stars can be considered as fixed in space and in time. The heliocentric orbit of the binary system and the relative positions of Didymain and Didymoon are known, thus the LOS measurements can be computed at the same time by making use of two cameras mounted onboard DustCube. Since Didymain and Didymoon will be subjected to wide time-windows without lighting of sunlight, two IR cameras are needed.

The evaluation of simultaneous LOS measurements and of the ranges at the same time brings to the knowledge of the S/C's position and velocity in Didymos system.

Indeed, since the boresight of the IR cameras w.r.t. the S/C are known a-priori, the inertial pointing direction of every camera can be easily computed by comparing the image of the detected body with the stars in the background and the a-priori knowledge of the body's shape, which is also useful to determine the range from the target body, because every target body will be completely content in the image captured by the camera for almost the entire mission phase.

Individual and independent LOS measurements are acquired over some interval of time together with ranges computations.

Therefore, the data are processed using a Kalman filter to determine the DustCube's state.

4 KALMAN FILTERING

The Kalman filtering technique takes the name from professor Rudolf Emil Kalman, since it was investigated and published by Kalman in 1960 and together professor Richard Snowden Bucy in 1961 starting from the concept of the least-squares estimations of dynamic systems, although Thorvald Nicolai Thiele and Peter Swerling developed a similar algorithm earlier.

The first implementation of the algorithm of the Kalman filter was created by Stanley F. Schmidt, a NASA engineer and researcher that exploited the Kalman idea for the Apollo missions in the early 60 years. Schmidt made use of this technique to track and evaluate the trajectories of launch vehicles in real-time.

Kalman filtering is also known as **linear quadratic estimator (LQE)** (*Grewal & Andrews, 2001*), since it results from the theory of the linear quadratic regulator (LQR) used in the optimal control of linear dynamic systems, where the system's state is stabilized by minimizing a cost or loss function. Following from this theory, the Kalman filter can be considered as an efficient recursive one that evaluates the state of a dynamic system in an optimal way starting from a series of measurements affected by noises, minimizing a loss function that coincides with the trace of the covariance matrix of the a-posteriori error of the state (see section A.4 Statistics for random vectors.in Appendix for more details).

The Kalman filtering technique has been exploited in different technological fields for the following years after his development, especially for guidance, navigation and control of vehicles, such as aircrafts and spacecrafts.

In this chapter, we will analyze the working processes of the optimal controls and filters, showing the main demonstrations and operation principles, to finally reach the concept of the Extended Kalman Filter, which is suitable for non-linear systems, such as the equations that govern the motion of a spacecraft.

4.1 OPTIMAL CONTROLLING AND FILTERING OF DYNAMIC SYSTEMS

Considering a **generic linear continuous time-variant (LTV) dynamic system in finite time**, we can represent the dynamic evolution of the state as:

CONTINUOUS LTV DYNAMIC SYSTEM:

$$\begin{aligned}\frac{d\mathbf{x}(t)}{dt} &= \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \\ \mathbf{z}(t) &= \mathbf{H}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) \\ \mathbf{x}(t_0) &= \mathbf{x}_0 \\ \mathbf{x}(t_f) &= \mathbf{x}_f\end{aligned}\tag{4.1}$$

where:

- $\mathbf{x}(t)$ is a $nx1$ vector stands for the state of the system, for example a single vector composed by position vector and velocity vector of a spacecraft with respect to a generic inertial or non-inertial reference frame;
- t is the time;
- $\mathbf{u}(t)$ is a $rx1$ vector stands for the input vector;
- $\mathbf{F}(t)$ is the nxn state-matrix;
- $\mathbf{B}(t)$ is the nxr input-matrix;
- $\mathbf{z}(t)$ is the measurement $mx1$ vector, that comprises every single measurement available from the sensors of the dynamic system, for example a radial distance from an object;
- $\mathbf{H}(t)$ is the mxn observation matrix;
- $\mathbf{D}(t)$ is the mrx matrix that related the measurement vector with the input vector, that can be considered null in this treatment;
- \mathbf{x}_0 is the state vector in the initial instant of time;
- \mathbf{x}_f is the state vector in the final instant of time.

As said before, the optimal control of linear dynamic systems can be achieved by making us of the linear quadratic regulator (LQR), where the cost or loss function J that must be minimized is a scalar positive quadratic function (*Castaldi, AA 2015-2016*):

$$J = \mathbf{x}_f^T \mathbf{S}_f \mathbf{x}_f + \int_{t_0}^{t_f} (\mathbf{x}(t)^T \mathbf{Q}(t) \mathbf{x}(t) + \mathbf{u}(t)^T \mathbf{R}(t) \mathbf{u}(t)) dt, \tag{4.2}$$

where:

- \mathbf{S}_f is a semi-positive definite matrix, such that the quadratic function $\mathbf{x}_f^T \mathbf{S}_f \mathbf{x}_f \geq 0$;
- $\mathbf{Q}(t)$ is a semi-positive definite matrix, such that the quadratic function $\mathbf{x}(t)^T \mathbf{Q}(t) \mathbf{x}(t) \geq 0$;
- $\mathbf{R}(t)$ is a definite positive matrix, such that the quadratic function $\mathbf{x}(t)^T \mathbf{R}(t) \mathbf{x}(t) > 0$.

Therefore, $\mathbf{Q}(t)$ penalizes the state of the system and $\mathbf{R}(t)$ penalizes the input control.

It is possible to demonstrate (*Grewal & Andrews, 2001*) that the solution of the linear quadratic optimal control problem can be achieved by solving the Differential Riccati Equation (DRE), that contains every information of the Euler-Lagrange (EL) solution:

$$\text{DRE: } \dot{\mathbf{S}}(t) + \mathbf{S}(t)\mathbf{F}(t) + \mathbf{F}(t)^T\mathbf{S}(t) - \mathbf{S}(t)\mathbf{B}(t)\mathbf{R}(t)^{-1}\mathbf{B}(t)^T\mathbf{S}(t) + \mathbf{Q}(t) = \mathbf{0}, \quad (4.3)$$

where $\mathbf{S}(t)$ is a semi-positive definite matrix and $\mathbf{S}(t_0) = \mathbf{S}_0$.

If the quadratic loss function J penalizes the outputs of the system instead of the state, i.e.:

$$J = \frac{1}{2}\mathbf{x}_f^T\mathbf{S}_f\mathbf{x}_f + \frac{1}{2}\int_{t_0}^{t_f} (\mathbf{z}(t)^T\mathbf{Q}_z(t)\mathbf{z}(t) + \mathbf{u}(t)^T\mathbf{R}(t)\mathbf{u}(t))dt, \quad (4.4)$$

the DRE becomes:

$$\text{DRE: } \dot{\mathbf{S}}(t) + \mathbf{S}(t)\mathbf{F}(t) + \mathbf{F}(t)^T\mathbf{S}(t) - \mathbf{S}(t)\mathbf{B}(t)\mathbf{R}(t)^{-1}\mathbf{B}(t)^T\mathbf{S}(t) + \mathbf{H}(t)^T\mathbf{Q}_z(t)\mathbf{H}(t) = \mathbf{0} \quad (4.5)$$

It is possible to demonstrate (*Grewal & Andrews, 2001*) that the optimal control gain that satisfies the DRE and stabilizes the system around a control point is:

$$\mathbf{K}(t) \stackrel{\text{def}}{=} -\mathbf{R}(t)^{-1}\mathbf{B}(t)^T\mathbf{S}(t), \quad (4.6)$$

such that the optimal control input is defined as:

$$\mathbf{u}(t) \stackrel{\text{def}}{=} \mathbf{K}(t)\mathbf{x}(t) = -\mathbf{R}(t)^{-1}\mathbf{B}(t)^T\mathbf{S}(t)\mathbf{x}(t) \quad (4.7)$$

Therefore, the dynamic of the system becomes:

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) = [\mathbf{F}(t) + \mathbf{K}(t)]\mathbf{x}(t), \quad (4.8)$$

which means the system turns into an autonomous one with state matrix $\mathbf{F}(t) + \mathbf{K}(t)$, such that the control gain asymptotically stabilizes the system making the eigenvalues of the original state matrix completely negative, if they were positive, or more negative with respect to the previous ones.

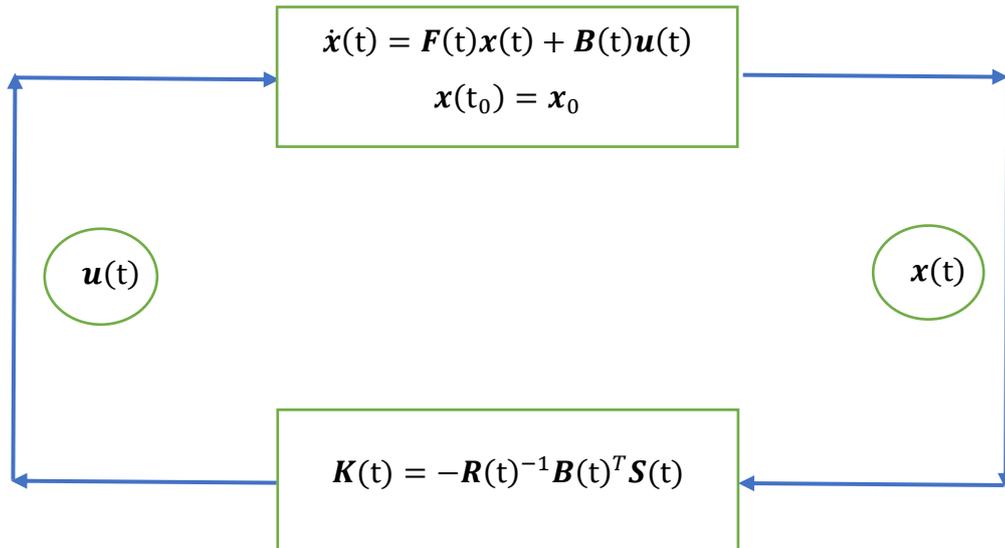


Figure 4.1: Schematic representation of LQR

In case of the system is a **generic linear continuous time-invariant (LTI) dynamic system in finite time**, i.e.:

CONTINUOUS LTI DYNAMIC SYSTEM:

$$\frac{dx(t)}{dt} = \mathbf{F}x(t) + \mathbf{B}u(t)$$

$$\mathbf{z}(t) = \mathbf{H}x(t) + \mathbf{D}u(t)$$

$$x(t_0) = x_0$$

$$x(t_f) = x_f$$

(4.9)

with the quadratic loss function J that becomes:

$$J = x_f^T S_f x_f + \int_{t_0}^{t_f} (z(t)^T Q_z z(t) + u(t)^T R u(t)) dt,$$
(4.10)

the optimal control gain remains time-variant, since $S = S(t)$:

$$K(t) = -R^{-1}B^T S(t)$$
(4.11)

Vice versa, if the final instant of time tends to infinite, the loss function can be defined as:

$$J = \int_{t_0}^{\infty} (z(t)^T Q_z z(t) + u(t)^T R u(t)) dt ;$$
(4.12)

in this case, if the system is completely controllable and completely observable, i.e. the couple (\mathbf{F}, \mathbf{B}) is stabilizable and the couple (\mathbf{F}, \mathbf{H}) is revealable, the regulator becomes stationary (*Grewal & Andrews, 2001*), while the Differential Riccati Equation turns into an Algebraic one (ARE) since $\mathbf{S} = \mathbf{S}_\infty = \mathbf{CONST}$, i.e.:

$$\text{ARE: } \mathbf{S}_\infty \mathbf{F} + \mathbf{F}^T \mathbf{S}_\infty - \mathbf{S}_\infty \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}_\infty + \mathbf{H}^T \mathbf{Q}_z \mathbf{H} = \mathbf{0}, \quad (4.13)$$

such that, the optimal control gain is constant, and the optimal control input becomes:

$$\mathbf{u}(t) = \mathbf{K}_\infty \mathbf{x}(t) = -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}_\infty \mathbf{x}(t) \quad (4.14)$$

As said before, the Kalman filter can be considered as a linear quadratic estimator (LQE), since its goal is to evaluate the state of the system minimizing a loss function J , defined in appropriate way, i.e. it is an optimal filter, such as the least-squares estimator.

Therefore, we can presume to know just an estimation of the state $\tilde{\mathbf{x}}(t)$ and we consider an estimator of the system in closed loop (*Castaldi, AA 2015-2016*), so that the dynamic equation of the estimation becomes:

$$\frac{d\tilde{\mathbf{x}}(t)}{dt} = \mathbf{F}(t)\tilde{\mathbf{x}}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{L}(t)[\mathbf{z}(t) - \tilde{\mathbf{z}}(t)], \quad (4.15)$$

so that, it is easy to define the evolution of the error vector:

$$\begin{aligned} \dot{\mathbf{e}}(t) &= \dot{\mathbf{x}}(t) - \dot{\tilde{\mathbf{x}}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) - \mathbf{F}(t)\tilde{\mathbf{x}}(t) - \mathbf{B}(t)\mathbf{u}(t) - \mathbf{L}(t)(\mathbf{z}(t) - \tilde{\mathbf{z}}(t)) = \\ &= \mathbf{F}(t)[\mathbf{x}(t) - \tilde{\mathbf{x}}(t)] - \mathbf{L}(t)\mathbf{H}(t)[\mathbf{x}(t) - \tilde{\mathbf{x}}(t)] = [\mathbf{F}(t) - \mathbf{L}(t)\mathbf{H}(t)]\mathbf{e}(t), \end{aligned} \quad (4.16)$$

which is a new dynamic autonomous system that features the estimator, i.e.:

$$\begin{cases} \dot{\mathbf{e}}(t) = [\mathbf{F}(t) - \mathbf{L}(t)\mathbf{H}(t)]\mathbf{e}(t) \\ \mathbf{e}(t_0) = \mathbf{e}_0 \end{cases} \quad (4.17)$$

Finally, it is possible to obtain the evolution in time of the error vector by solving the system (4.17):

$$\mathbf{e}(t) = e^{[\mathbf{F}(t) - \mathbf{L}(t)\mathbf{H}(t)]t} \mathbf{e}(t_0) \quad (4.18)$$

As for the optimal regulator, $\mathbf{L}(t)$ is the necessary gain matrix of the optimal estimator to minimize a scalar function J defined in appropriate way, i.e. to stabilize the estimation of the state.

The cost function J is usually defined through the weighted mean-squared errors (see eq. (A.72) in Appendix), i.e.:

$$J \stackrel{\text{def}}{=} E \left[(\mathbf{x}(t) - \hat{\mathbf{x}}(t))^T \mathbf{M} (\mathbf{x}(t) - \hat{\mathbf{x}}(t)) \right] = E[\mathbf{e}(t)^T \mathbf{M} \mathbf{e}(t)], \quad (4.19)$$

where \mathbf{M} is a semi-positive or positive definite weighting matrix.

For example, if we consider a white gaussian with zero mean $m \times l$ noise vector $\mathbf{v}(t)$ as the measurement error, the measurement vector related to the estimation of the system becomes:

$$\tilde{\mathbf{z}}(t) = \mathbf{H}(t)\tilde{\mathbf{x}}(t) + \mathbf{v}(t), \quad (4.20)$$

so that, is possible to exploit the LEAST-SQUARES METHOD to obtain the optimal estimation of the state, minimizing a scalar cost function J that represents the inner product of the error vector:

$$J = \mathbf{v}(t)^T \mathbf{v}(t) = [\mathbf{z}(t) - \mathbf{H}(t)\tilde{\mathbf{x}}(t)]^T [\mathbf{z}(t) - \mathbf{H}(t)\tilde{\mathbf{x}}(t)] \quad (4.21)$$

To minimize the cost function, we compute the gradient of J w.r.t. $\tilde{\mathbf{x}}(t)$ and pose it equal to zero:

$$\frac{\partial J}{\partial \tilde{\mathbf{x}}(t)} = 2\mathbf{H}(t)^T \mathbf{H}(t)\tilde{\mathbf{x}}(t) - 2\mathbf{H}(t)^T \mathbf{z}(t) = \mathbf{0} \quad (4.22)$$

so that, iff $\mathbf{H}(t)^T \mathbf{H}(t)$ is a non-singular matrix, the optimal estimation of the state becomes:

$$\tilde{\mathbf{x}}(t) = [\mathbf{H}(t)^T \mathbf{H}(t)]^{-1} \mathbf{H}(t)^T \mathbf{z}(t) \quad (4.23)$$

Therefore, exploiting the eq. (4.23), is possible to estimate the state of the system at every iteration. Starting from this theory, Rudolph Kalman and his colleagues developed the concepts of the Kalman filtering, as it will be clearer later.

4.2 KALMAN FILTERING FOR LINEAR CONTINUOUS DYNAMIC SYSTEMS

It is possible to generalize the operation of the Kalman filtering for a generic **continuous linear time-variant (LTV) dynamic system in finite time affected by noises**, which is defined as:

CONTINUOUS LTV DYNAMIC SYSTEM with noises:

$$\begin{aligned} \frac{d\mathbf{x}(t)}{dt} &= \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{w}(t) \\ \mathbf{z}(t) &= \mathbf{H}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) + \mathbf{v}(t) \\ \mathbf{x}(t_0) &= \mathbf{x}_0 \\ \mathbf{x}(t_f) &= \mathbf{x}_f \end{aligned} \quad (4.24)$$

where, as said before, $\mathbf{D}(t)$ is usually null and $\mathbf{w}(t)$ and $\mathbf{v}(t)$ are the process and the measurement noises, respectively, which have null means and, therefore, well-defined covariance matrices (see eq. (A.73) in Appendix) $\mathbf{Q}_w(t)$ and $\mathbf{R}_v(t)$, namely

$$\begin{aligned} E[\mathbf{w}(t)] &= E[\mathbf{v}(t)] = \mathbf{0}, \\ \mathbf{Q}_w(t) &\stackrel{\text{def}}{=} \text{cov}(\mathbf{w}(t), \mathbf{w}(t)) = E[\mathbf{w}(t)\mathbf{w}(t)^T] = \text{corr}(\mathbf{w}(t)) \\ \mathbf{R}_v(t) &\stackrel{\text{def}}{=} \text{cov}(\mathbf{v}(t), \mathbf{v}(t)) = E[\mathbf{v}(t)\mathbf{v}(t)^T] = \text{corr}(\mathbf{v}(t)) \end{aligned} \quad (4.25)$$

Indeed, the covariance matrix of a vector signal \mathbf{a} is defined as (Luise & Vitetta, 2009, June 1):

$$\begin{aligned} \text{cov}(\mathbf{a}, \mathbf{a}) &\stackrel{\text{def}}{=} E[(\mathbf{a} - E[\mathbf{a}])(\mathbf{a} - E[\mathbf{a}])^T] = E[\mathbf{a}\mathbf{a}^T] \quad \text{iff } E[\mathbf{a}] = \mathbf{0} \\ E[\mathbf{a}\mathbf{a}^T] &= \text{corr}(\mathbf{a}) \quad \text{by definition} \end{aligned} \quad (4.26)$$

Also, $\mathbf{Q}_w(t)$ and $\mathbf{R}_v(t)$ are diagonal positive definite matrices, namely:

$$\begin{aligned} \mathbf{Q}_w(t) &= \text{diag}(\sigma_1^2, \dots, \sigma_n^2) \quad \text{with } \sigma_i^2 = \text{var}(w_i) \\ \mathbf{R}_v(t) &= \text{diag}(\sigma_1^2, \dots, \sigma_m^2) \quad \text{with } \sigma_i^2 = \text{var}(v_i), \end{aligned} \quad (4.27)$$

which means that:

$$\begin{aligned} \text{cov}(w_i, w_j) &= 0 \quad \forall i, j = 1, \dots, n \quad \text{with } i \neq j \\ \text{cov}(v_i, v_j) &= 0 \quad \forall i, j = 1, \dots, m \quad \text{with } i \neq j \end{aligned} \quad (4.28)$$

Therefore, we can finally conclude that the components of every noise vector are uncorrelated each other, since $E[w_i] = E[v_i] = 0$, so that $\mathbf{w}(t)$ and $\mathbf{v}(t)$ are white Gaussian with zero mean noises and can be represented by the following common notations:

$$\begin{aligned} \mathbf{w}(t) &\sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_w(t)) \\ \mathbf{v}(t) &\sim \mathcal{N}(\mathbf{0}, \mathbf{R}_v(t)) \end{aligned} \quad (4.29)$$

Since $\mathbf{w}(t)$ and $\mathbf{v}(t)$ are stochastic processes, $\mathbf{x}(t)$ and $\mathbf{z}(t)$ are vector random variables too.

Exploiting the cross-correlation matrices (see eq. (A.78) in Appendix), it is possible to show that the noises $\mathbf{w}(t)$ and $\mathbf{v}(t)$ are uncorrelated each other and are uncorrelated with the same noises at different instants of time:

$$\begin{aligned} \text{corr}(\mathbf{w}(t), \mathbf{v}(t)) &= \text{corr}(\mathbf{v}(t), \mathbf{w}(t)) = \mathbf{0} \\ \text{corr}(\mathbf{w}(t_k), \mathbf{w}(t_h)) &= E[\mathbf{w}(t_k)\mathbf{w}(t_h)^T] = \mathbf{0} \quad \forall k, h \quad \text{with } k \neq h \end{aligned} \quad (4.30)$$

$$\text{corr}(\mathbf{v}(t_k), \mathbf{v}(t_h)) = E[\mathbf{v}(t_k)\mathbf{v}(t_h)^T] = \mathbf{0} \quad \forall k, h \text{ with } k \neq h$$

Now, recalling the dynamic equation of the estimation in closed loop (4.15) and the cost or loss function J as defined in eq. (4.19) with $\mathbf{M} = \mathbf{I}$, it is possible to demonstrate (Grewal & Andrews, 2001) that J becomes:

$$J = E[\mathbf{e}(t)^T \mathbf{e}(t)] = \text{trace}(\mathbf{e}(t)) \quad (4.31)$$

If the system is completely controllable and completely observable, i.e. the couple $(\mathbf{F}(t), \mathbf{B}(t))$ is stabilizable and the couple $(\mathbf{F}(t), \mathbf{H}(t))$ is revealable, the estimator in closed loop can be stationary and optimal if the estimator gain is selected as:

$$\mathbf{L}(t) = -\mathbf{R}_v(t)^{-1} \mathbf{H}(t)^T \mathbf{P}(t), \quad (4.32)$$

where $\mathbf{P}(t)$ is the solution of the Cauchy problem composed by the DRE and the initial value \mathbf{P}_0 :

$$\begin{cases} \dot{\mathbf{P}}(t) = \mathbf{P}(t)\mathbf{F}(t) + \mathbf{F}(t)^T \mathbf{P}(t) - \mathbf{P}(t)\mathbf{H}(t)\mathbf{R}_v(t)^{-1} \mathbf{H}(t)^T \mathbf{P}(t) + \mathbf{Q}_w(t) \\ \mathbf{P}(t_0) = \mathbf{P}_0 \end{cases} \quad (4.33)$$

Vice versa, if we consider a generic **continuous linear time-invariant (LTI) dynamic system in infinite time affected by noises** with constant covariance matrices \mathbf{Q}_w and \mathbf{R}_v , namely:

CONTINUOUS LTI DYNAMIC SYSTEM with noises:

$$\begin{aligned} \frac{d\mathbf{x}(t)}{dt} &= \mathbf{F}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{w}(t) \\ \mathbf{z}(t) &= \mathbf{H}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) + \mathbf{v}(t) \\ \mathbf{x}(t_0) &= \mathbf{x}_0 \end{aligned} \quad (4.34)$$

if the system is completely controllable and completely observable, it is possible to project an optimal and stationary LQR, minimizing the loss function J_{LQR} as defined in eq. (4.12), by selecting the following constant control gain \mathbf{K}_{LQR} :

$$\mathbf{K}_{LQR} \stackrel{\text{def}}{=} -\mathbf{R}^{-1} \mathbf{B}^T \mathbf{S}_\infty, \quad (4.35)$$

where \mathbf{S}_∞ is the solution of the ARE (4.13), and, at the same time, to develop an independent optimal and stationary LQE, minimizing the loss function J_{LQE} as defined in eq. (4.19) with $\mathbf{M} = \mathbf{I}$, by selecting the following constant estimator gain \mathbf{K}_{LQE} :

$$\mathbf{K}_{LQE} = -\mathbf{R}_v^{-1} \mathbf{H}^T \mathbf{P}_\infty, \quad (4.36)$$

where \mathbf{P}_∞ is the solution of the following ARE:

$$\text{ARE: } \mathbf{P}_\infty \mathbf{F} + \mathbf{F}^T \mathbf{P}_\infty - \mathbf{P}_\infty \mathbf{H} \mathbf{R}_v^{-1} \mathbf{H}^T \mathbf{P}_\infty + \mathbf{Q}_w = \mathbf{0} \quad (4.37)$$

Therefore, projecting an optimal LQR and an optimal LQE for a generic LTI dynamic system, we have developed the so called **linear quadratic Gaussian control (LQG)**.

4.3 KALMAN FILTERING FOR LINEAR DISCRETE DYNAMIC SYSTEMS

A generic **linear discrete dynamic system in finite time affected by noises** can be represented by the discrete (or discretized) equations of the state and the measurement vectors:

LINEAR DISCRETE DYNAMIC SYSTEM:

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k + \mathbf{w}_k$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{D}_k \mathbf{u}_k + \mathbf{v}_k$$

$$\mathbf{x}(t_0) = \mathbf{x}_0$$

$$\mathbf{x}(t_f) = \mathbf{x}_f$$

(4.38)

where:

- \mathbf{x}_{k+1} is the $nx1$ state vector at time t_{k+1} ;
- \mathbf{x}_k is the $nx1$ state vector at time t_k ;
- \mathbf{z}_k is the $mx1$ measurement vector at time t_k ;
- \mathbf{u}_k is the $rx1$ input vector at time t_k ;
- $\mathbf{\Phi}_k$ is the nxn state transition matrix, which is defined as: $\mathbf{\Phi}_k = \frac{\partial \mathbf{x}_{k+1}}{\partial \mathbf{x}_k}$;
- \mathbf{B}_k is the nxr control matrix at time t_k ;
- \mathbf{H}_k is the mxn observation matrix at time t_k ;
- \mathbf{D}_k is the mxr matrix that relates the measurement vector with the control one;
- \mathbf{x}_0 is the state vector in the initial time instant;
- \mathbf{x}_f is the state vector in the final time instant;
- \mathbf{w}_k is the process noise;
- \mathbf{v}_k is the measurement noise.

As said before, \mathbf{w}_k and \mathbf{v}_k are white Gaussian with zero mean noise vectors, namely:

$$\begin{aligned}\mathbf{w}_k &\sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \\ \mathbf{v}_k &\sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k),\end{aligned}\tag{4.39}$$

which leads to the conclusion that also \mathbf{x}_k , \mathbf{z}_k and \mathbf{x}_{k+1} are vector random variables.

Usually, in astronomical applications \mathbf{u}_k is a null vector, since the Kalman filter is exploited to determine the trajectories and the attitudes of the spacecrafts w/o controlling inputs.

Therefore, at every iteration we have a measurement vector and a state vector affected by errors, which have covariance matrices \mathbf{Q}_k and \mathbf{R}_k , namely:

$$\begin{aligned}\mathbf{Q}_k &\stackrel{\text{def}}{=} \text{cov}(\mathbf{w}_k, \mathbf{w}_k) = E[\mathbf{w}_k \mathbf{w}_k^T] = \text{corr}(\mathbf{w}_k) = \text{diag}(\sigma_1^2, \dots, \sigma_n^2) \quad \text{with } \sigma_i^2 = \text{var}(w_k^i) \\ \mathbf{R}_k &\stackrel{\text{def}}{=} \text{cov}(\mathbf{v}_k, \mathbf{v}_k) = E[\mathbf{v}_k \mathbf{v}_k^T] = \text{corr}(\mathbf{v}_k) = \text{diag}(\sigma_1^2, \dots, \sigma_m^2) \quad \text{with } \sigma_i^2 = \text{var}(v_k^i),\end{aligned}\tag{4.40}$$

which means that the components of every noise vector are uncorrelated each other and the measurement and the process noises are uncorrelated considering different instants of time, i.e.:

$$\begin{aligned}\text{cov}(v_k^i, v_k^j) &= 0 \quad \forall i, j = 1, \dots, m \text{ with } i \neq j \\ \text{cov}(w_k^i, w_k^j) &= 0 \quad \forall i, j = 1, \dots, n \text{ with } i \neq j, \\ \text{corr}(\mathbf{v}_k, \mathbf{v}_h) &= E[\mathbf{v}_k \mathbf{v}_h^T] = E[\mathbf{v}_h \mathbf{v}_k^T] = \mathbf{0} \\ \text{corr}(\mathbf{w}_k, \mathbf{w}_h) &= E[\mathbf{w}_k \mathbf{w}_h^T] = E[\mathbf{w}_h \mathbf{w}_k^T] = \mathbf{0}\end{aligned}\tag{4.41}$$

Therefore, we can consider an a-priori estimation of the state $\tilde{\mathbf{x}}_k^-$ and an a-posteriori one $\tilde{\mathbf{x}}_k^+$, which both have \mathbf{x}_k as mean value and the following a-priori and a-posteriori errors, respectively:

$$\begin{aligned}\mathbf{e}_k^- &\stackrel{\text{def}}{=} \tilde{\mathbf{x}}_k^- - \mathbf{x}_k \\ \mathbf{e}_k^+ &\stackrel{\text{def}}{=} \tilde{\mathbf{x}}_k^+ - \mathbf{x}_k\end{aligned}\tag{4.42}$$

The a-priori and a-posteriori errors have both null means and covariance matrices \mathbf{P}_k^- and \mathbf{P}_k^+ , respectively, i.e.:

$$\begin{aligned}E[\mathbf{e}_k^-] &= E[\mathbf{e}_k^+] = \mathbf{0} \\ \mathbf{P}_k^- &\stackrel{\text{def}}{=} \text{cov}(\mathbf{e}_k^-, \mathbf{e}_k^-) = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}] \\ \mathbf{P}_k^+ &\stackrel{\text{def}}{=} \text{cov}(\mathbf{e}_k^+, \mathbf{e}_k^+) = E[\mathbf{e}_k^+ \mathbf{e}_k^{+T}]\end{aligned}\tag{4.43}$$

Also, the a-priori and a-posteriori estimations of the state have the same covariance matrices of the a-priori and a-posteriori errors, respectively, indeed:

$$\begin{aligned} cov(\tilde{\mathbf{x}}_k^-, \tilde{\mathbf{x}}_k^-) &= E[(\tilde{\mathbf{x}}_k^- - E[\tilde{\mathbf{x}}_k^-])(\tilde{\mathbf{x}}_k^- - E[\tilde{\mathbf{x}}_k^-])^T] = E[(\tilde{\mathbf{x}}_k^- - \mathbf{x}_k)(\tilde{\mathbf{x}}_k^- - \mathbf{x}_k)^T] = \mathbf{P}_k^- \\ cov(\tilde{\mathbf{x}}_k^+, \tilde{\mathbf{x}}_k^+) &= E[(\tilde{\mathbf{x}}_k^+ - E[\tilde{\mathbf{x}}_k^+])(\tilde{\mathbf{x}}_k^+ - E[\tilde{\mathbf{x}}_k^+])^T] = E[(\tilde{\mathbf{x}}_k^+ - \mathbf{x}_k)(\tilde{\mathbf{x}}_k^+ - \mathbf{x}_k)^T] = \mathbf{P}_k^+ \end{aligned} \quad (4.44)$$

Since the mean values of the errors are null (1st eq. of (4.43)), the Kalman filter is an *unbiased filter*. Also, the a-priori and a-posteriori errors and the measurement and process noises are uncorrelated each other, i.e. the cross-correlation matrices are null:

$$\begin{aligned} corr(\mathbf{w}_k, \mathbf{v}_k) &= E[\mathbf{w}_k \mathbf{v}_k^T] = E[\mathbf{v}_k \mathbf{w}_k^T] = \mathbf{0} \\ corr(\mathbf{e}_k, \mathbf{w}_k) &= E[\mathbf{e}_k \mathbf{w}_k^T] = E[\mathbf{w}_k \mathbf{e}_k^T] = \mathbf{0} \\ corr(\mathbf{e}_k, \mathbf{v}_k) &= E[\mathbf{e}_k \mathbf{v}_k^T] = E[\mathbf{v}_k \mathbf{e}_k^T] = \mathbf{0} \end{aligned} \quad (4.45)$$

Therefore, the recursive scheme relates the a-posteriori estimation of the state with the a-priori one and the measurement vector (*Tortora & Modenini, AA 2014-2015*), namely:

$$\tilde{\mathbf{x}}_k^+ \stackrel{\text{def}}{=} \mathbf{A}_k \tilde{\mathbf{x}}_k^- + \mathbf{B}_k \mathbf{z}_k, \quad (4.46)$$

where \mathbf{A}_k and \mathbf{B}_k are two matrices of dimensions $n \times n$ and $n \times m$, respectively.

Since the filter is unbiased, it is possible to demonstrate that (*Grewal & Andrews, 2001*):

$$\mathbf{A}_k = \mathbf{I} - \mathbf{B}_k \mathbf{H}_k, \quad (4.47)$$

which brings to a new definition of the a-posteriori error $\tilde{\mathbf{x}}_k^+$ starting from the eq. (4.46):

$$\tilde{\mathbf{x}}_k^+ = \mathbf{A}_k \tilde{\mathbf{x}}_k^- + \mathbf{B}_k \mathbf{z}_k = (\mathbf{I} - \mathbf{B}_k \mathbf{H}_k) \tilde{\mathbf{x}}_k^- + \mathbf{B}_k \mathbf{z}_k = \tilde{\mathbf{x}}_k^- + \mathbf{B}_k [\mathbf{z}_k - \mathbf{H}_k \tilde{\mathbf{x}}_k^-] = \tilde{\mathbf{x}}_k^- + \mathbf{B}_k \mathbf{i}_k, \quad (4.48)$$

where \mathbf{i}_k is defined as the *innovation* of the recursive scheme.

Therefore, the a-posteriori error covariance matrix is:

$$\mathbf{P}_k^+ = E[\mathbf{e}_k^+ \mathbf{e}_k^{+T}] = E\{[(\mathbf{I} - \mathbf{B}_k \mathbf{H}_k) \mathbf{e}_k^- + \mathbf{B}_k \mathbf{v}_k][(\mathbf{I} - \mathbf{B}_k \mathbf{H}_k) \mathbf{e}_k^- + \mathbf{B}_k \mathbf{v}_k]^T\}, \quad (4.49)$$

which, after some manipulations, becomes:

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{B}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{B}_k \mathbf{H}_k)^T + \mathbf{B}_k \mathbf{R}_k \mathbf{B}_k^T \quad (4.50)$$

It can be shown that minimizing the mean quadratic value of the a-posteriori error is like to minimize the trace of the a-posteriori error covariance matrix (*Grewal & Andrews, 2001*).

Indeed, the classical expression of the loss function J of eq. (4.19) with $\mathbf{M} = \mathbf{I}$ can be redefined as: (4.51)

$$J = trace(\mathbf{P}_k^+) = trace(\mathbf{P}_k^- + \mathbf{B}_k \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{B}_k^T - \mathbf{B}_k \mathbf{H}_k \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{B}_k^T) + trace(\mathbf{B}_k \mathbf{R}_k \mathbf{B}_k^T),$$

so that, minimizing it w.r.t. \mathbf{B}_k , i.e.:

$$\frac{\partial J}{\partial \mathbf{B}_k} = \frac{\partial \text{trace}(\mathbf{P}_k^+)}{\partial \mathbf{B}_k} = 2(\mathbf{B}_k \mathbf{H}_k - \mathbf{I}) \mathbf{P}_k^- \mathbf{H}_k^T + 2\mathbf{B}_k \mathbf{R}_k = \mathbf{0}, \quad (4.52)$$

and substituting \mathbf{B}_k with \mathbf{K}_k , it is possible to obtain the optimal gain of the filter:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (4.53)$$

Also, we can redefine the a-posteriori error covariance matrix, exploiting the optimal gain, and we must propagate the a-priori error covariance matrix, namely:

$$\begin{aligned} \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \\ \mathbf{P}_{k+1}^- &= E[\mathbf{e}_{k+1}^- \mathbf{e}_{k+1}^{-T}] = E[(\Phi_k \mathbf{e}_k^+ - \mathbf{w}_k)(\Phi_k \mathbf{e}_k^+ - \mathbf{w}_k)^T] = \Phi_k \mathbf{P}_k^+ \Phi_k^T + \mathbf{Q}_k \end{aligned} \quad (4.54)$$

Since the a-priori error covariance matrix is computed through the second one of eqs (4.54), the initial a-priori error covariance matrix \mathbf{P}_0^- can be chosen as equal to the process noise covariance matrix of the initial state, i.e. \mathbf{Q}_0

In Figure 4.2 is shown the Kalman Filter's block diagram, while in Table 4.1 the recursive scheme of the classical Kalman Filter algorithm.

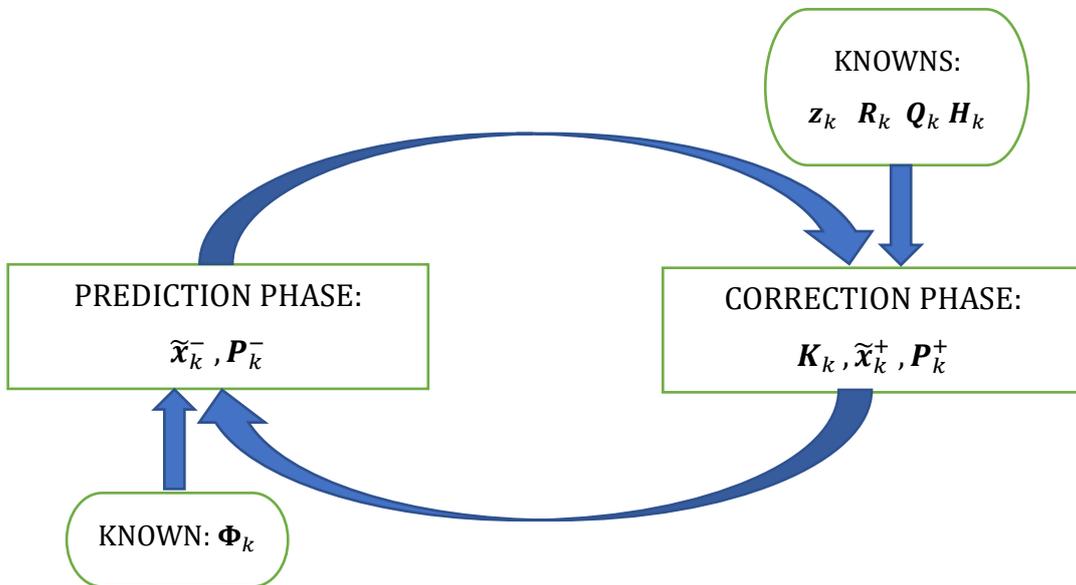


Figure 4.2: Kalman Filter's block diagram

Initial conditions:

$$\tilde{\mathbf{x}}_0, \mathbf{P}_0$$

Knowns:

$$\mathbf{z}_k \mathbf{H}_k \mathbf{R}_k \mathbf{Q}_k \Phi_k$$

Initialization:

$$\tilde{\mathbf{x}}_k^-, \mathbf{P}_k^-$$

Recursive scheme:

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

$$\tilde{\mathbf{x}}_k^+ = \tilde{\mathbf{x}}_k^- + \mathbf{K}_k [\mathbf{z}_k - \mathbf{H}_k \tilde{\mathbf{x}}_k^-]$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$$

Project ahead:

$$\tilde{\mathbf{x}}_{k+1}^- = \Phi_k \tilde{\mathbf{x}}_k^+$$

$$\mathbf{P}_{k+1}^- = \Phi_k \mathbf{P}_k^+ \Phi_k^T + \mathbf{Q}_k$$

Table 4.1: Kalman Filter algorithm

4.4 LINEARIZED KALMAN FILTER (LKF)

As said before, the Kalman filter is optimal in the sense that minimizes the trace of the a-posteriori error covariance matrix (*Grewal & Andrews, 2001*), while the least-squares method technique minimizes the loss function J represented by the inner product of the error vector \mathbf{v} (eq. (4.21)).

It is possible to demonstrate that the Kalman filter minimizes the same loss function J of eq. (4.21), so that it can be regarded as an extension of the concept of the least-squares technique.

Indeed, considering **non-linear discretized equations** for measurements, namely:

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, \quad (4.55)$$

we can linearize the measurement function $\mathbf{h}(\mathbf{x})$ by exploiting the Taylor's method:

$$\mathbf{h}_k(\tilde{\mathbf{x}}_k^-) = \mathbf{h}_k(\mathbf{x}_k^{ref} + \Delta \mathbf{x}_k^-) \approx \mathbf{h}_k(\mathbf{x}_k^{ref}) + \mathbf{H}_k(\tilde{\mathbf{x}}_k^- - \mathbf{x}_k^{ref}), \quad (4.56)$$

where \mathbf{x}_k^{ref} is the reference state at the k^{th} iteration and \mathbf{H}_k is the Jacobian matrix in numerator layout representation, i.e.:

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}_k} \right|_{(\mathbf{x}_k = \tilde{\mathbf{x}}_k^-)} = \begin{bmatrix} \frac{\partial h_{1k}}{\partial x_{1k}} & \dots & \frac{\partial h_{1k}}{\partial x_{n_k}} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{m_k}}{\partial x_{1k}} & \dots & \frac{\partial h_{m_k}}{\partial x_{n_k}} \end{bmatrix}_{(\mathbf{x}_k = \tilde{\mathbf{x}}_k^-)} \quad (4.57)$$

Therefore, considering the term $\Delta \mathbf{z}_k = \mathbf{z}_k - \mathbf{h}_k(\mathbf{x}_k^{ref})$, the Kalman gain matrix \mathbf{K}_k and recalling that $\mathbf{P}_k^- = cov(\tilde{\mathbf{x}}_k^-, \tilde{\mathbf{x}}_k^-) = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}]$, the objective of the Linearized Kalman Filter is to minimize the loss function J , which can be defined as:

$$J \stackrel{\text{def}}{=} [\Delta \mathbf{z}_k - \mathbf{H}_k(\tilde{\mathbf{x}}_k^- - \mathbf{x}_k^{ref})]^T [\Delta \mathbf{z}_k - \mathbf{H}_k(\tilde{\mathbf{x}}_k^- - \mathbf{x}_k^{ref})] + [\mathbf{x}_k - \tilde{\mathbf{x}}_k^-]^T (\mathbf{P}_k^-)^{-1} [\mathbf{x}_k - \tilde{\mathbf{x}}_k^-], \quad (4.58)$$

so that, we finally reach the a-posteriori evaluation of the state:

$$\tilde{\mathbf{x}}_k^+ = \tilde{\mathbf{x}}_k^- + \mathbf{P}_k^+ \mathbf{H}_k^T \mathbf{R}_k^{-1} [\Delta \mathbf{z}_k - \mathbf{H}_k(\tilde{\mathbf{x}}_k^- - \mathbf{x}_k^{ref})] = \tilde{\mathbf{x}}_k^- + \mathbf{K}_k [\Delta \mathbf{z}_k - \mathbf{H}_k(\tilde{\mathbf{x}}_k^- - \mathbf{x}_k^{ref})] \quad (4.59)$$

Now, exploiting the following relation:

$$\mathbf{P}_k^+ = [(\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k]^{-1}, \quad (4.60)$$

after some manipulations, we obtain the expression of the Kalman gain matrix \mathbf{K}_k as the eq. (4.53):

Manipulating the a-priori error covariance matrix \mathbf{P}_k^- too, since $\mathbf{K}_k = \mathbf{P}_k^+ \mathbf{H}_k^T \mathbf{R}_k^{-1}$, we find out:

$$\begin{aligned} \mathbf{P}_k^- &= \mathbf{P}_k^+ (\mathbf{P}_k^+)^{-1} \mathbf{P}_k^- = \mathbf{P}_k^+ [(\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k] \mathbf{P}_k^- = \mathbf{P}_k^+ (\mathbf{I} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \mathbf{P}_k^-) \\ &= \mathbf{P}_k^+ + (\mathbf{P}_k^+ \mathbf{H}_k^T \mathbf{R}_k^{-1}) \mathbf{H}_k \mathbf{P}_k^- = \mathbf{P}_k^+ + \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^-, \end{aligned} \quad (4.61)$$

which brings to the expression of the a-posteriori error covariance matrix \mathbf{P}_k^+ as the eq. (4.54).

Once again, is easy to conclude that minimizing the trace of \mathbf{P}_k^+ is like to minimize the loss function J , since they are the same functions expressed in different ways.

In this case, we define the state transition matrix as the Jacobian one, namely:

$$\Phi_k \stackrel{\text{def}}{=} \frac{\partial \mathbf{x}_{k+1}^{ref}}{\partial \mathbf{x}_k^{ref}} \quad (4.62)$$

In Figure 4.3 is shown the step by step estimated trajectory of an orbiting S/C by using an LKF, while in Table 4.2 the recursive scheme of the **Linearized Kalman Filter (LKF)** algorithm.

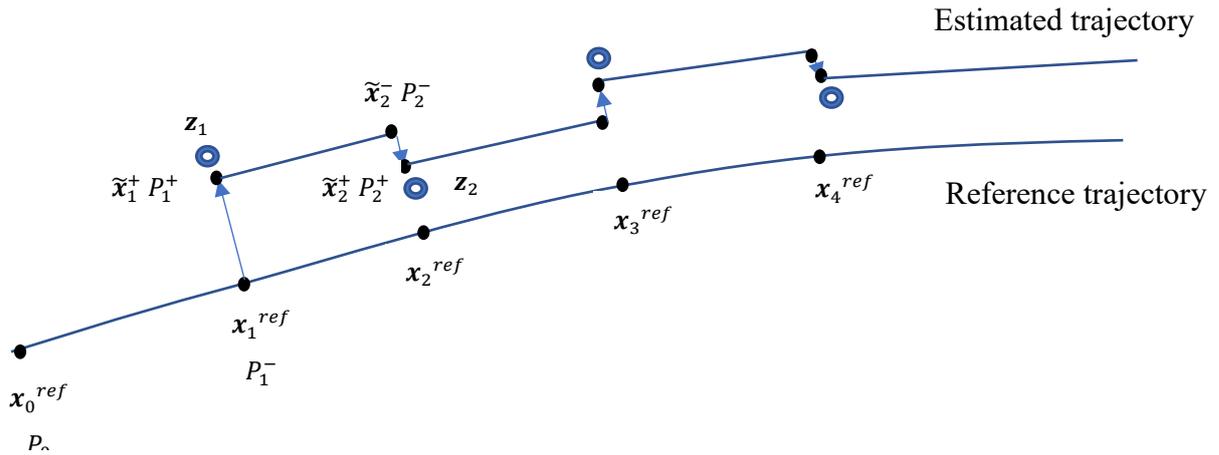


Figure 4.3: Estimated trajectory of a S/C by using an LKF (Montenbruck & Gill, *Satellite Orbits*)

Initial conditions:

$$\tilde{\mathbf{x}}_0, \mathbf{P}_0$$

Knowns:

$$\mathbf{z}_k, \mathbf{R}_k, \mathbf{Q}_k, \mathbf{h}_k(\mathbf{x}_k^{ref}), \mathbf{x}_k^{ref}$$

Initialization:

$$\tilde{\mathbf{x}}_k^-, \mathbf{P}_k^-$$

Recursive scheme:

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{(\mathbf{x}_k = \tilde{\mathbf{x}}_k^-)}$$

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

$$\tilde{\mathbf{x}}_k^+ = \tilde{\mathbf{x}}_k^- + \mathbf{K}_k [\mathbf{z}_k - \mathbf{h}_k(\mathbf{x}_k^{ref}) - \mathbf{H}_k (\tilde{\mathbf{x}}_k^- - \mathbf{x}_k^{ref})]$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$$

Project ahead:

$$\Phi_k = \frac{\partial \mathbf{x}_{k+1}^{ref}}{\partial \mathbf{x}_k^{ref}}$$

$$\tilde{\mathbf{x}}_{k+1}^- = \mathbf{x}_{k+1}^{ref} + \Phi_k (\tilde{\mathbf{x}}_k^+ - \mathbf{x}_k^{ref})$$

$$\mathbf{P}_{k+1}^- = \Phi_k \mathbf{P}_k^+ \Phi_k^T + \mathbf{Q}_k$$

Table 4.2: Linearized Kalman Filter algorithm

4.5 EXTENDED KALMAN FILTER (EKF)

Usually, in astronautic navigation the discretized equations are **non-linear**, but is preferable to use the **Extended Kalman Filter (EKF)** algorithm instead of the linearized one, since is not necessary to have a reference trajectory, thanks to the replacement of \mathbf{x}_k^{ref} with $\hat{\mathbf{x}}_k^+$ at every iteration.

Thus, considering a generic **non-linear discrete dynamic system in finite time affected by noises**:

NON-LINEAR DISCRETE DYNAMIC SYSTEM:

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \quad (4.63)$$

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k$$

recalling the definitions (4.40), (4.42), the 2nd and the 3rd of (4.43), the (4.57) one and the main features of noise and errors vectors (4.39) and the 1st of (4.43), which continue to remain valid also for the EKF algorithm, and defining the new state transition matrix \mathbf{F}_k as the Jacobian one in numerator layout representation, i.e.:

$$\mathbf{F}_k \stackrel{\text{def}}{=} \left. \frac{\partial \mathbf{f}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{(\mathbf{x}_k = \hat{\mathbf{x}}_k^+)} = \begin{bmatrix} \frac{\partial f_{1k}}{\partial x_{1k}} & \dots & \frac{\partial f_{1k}}{\partial x_{nk}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{nk}}{\partial x_{1k}} & \dots & \frac{\partial f_{nk}}{\partial x_{nk}} \end{bmatrix}_{(\mathbf{x}_k = \hat{\mathbf{x}}_k^+)} , \quad (4.64)$$

we can obtain the classical recursive scheme of the EKF algorithm.

In Figure 4.4 is shown the Extended Kalman Filter's block diagram, while in Figure 4.5 the step by step estimated trajectory of an orbiting S/C by using an **EKF**.

The Table 4.3 resumes the main features of the EKF algorithm.

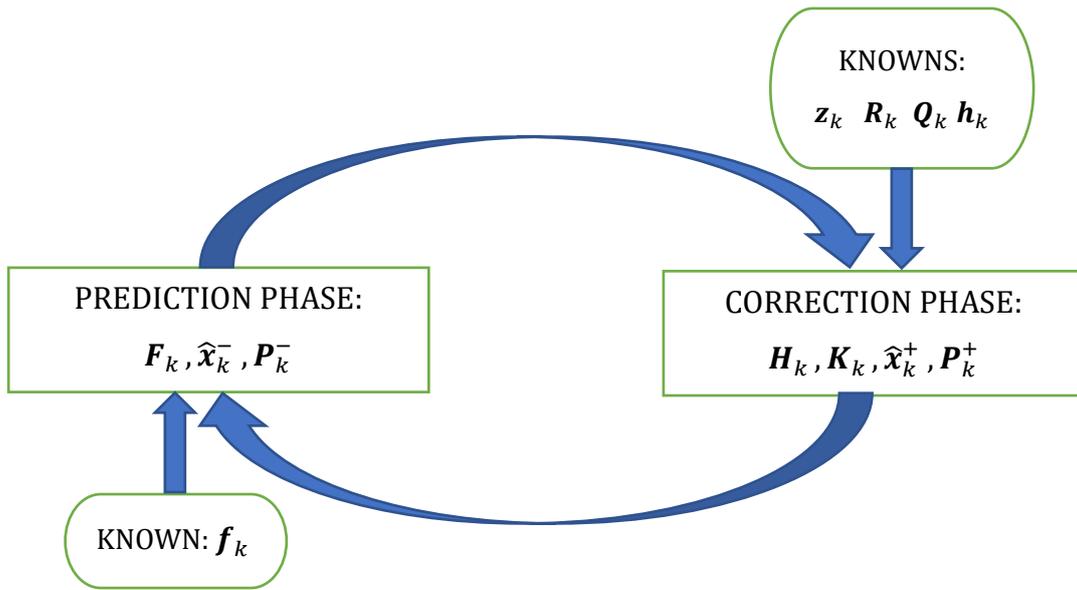


Figure 4.4: Extended Kalman Filter's block diagram

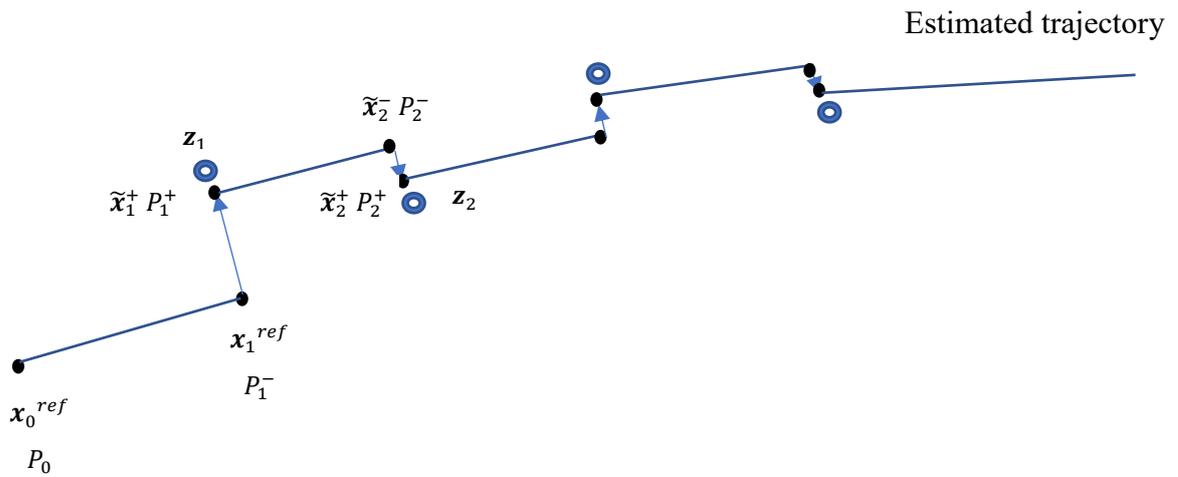


Figure 4.5: Estimated trajectory of a S/C by using EKF (Montenbruck & Gill, *Satellite Orbits*)

Initial conditions:

$$\tilde{\mathbf{x}}_0, \mathbf{P}_0$$

Knowns:

$$\mathbf{z}_k, \mathbf{R}_k, \mathbf{Q}_k, \mathbf{h}_k(\mathbf{x}_k), \mathbf{f}_k(\mathbf{x}_k)$$

Initialization:

$$\tilde{\mathbf{x}}_k^-, \mathbf{P}_k^-$$

Recursive scheme:

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{(\mathbf{x}_k = \tilde{\mathbf{x}}_k^-)}$$

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

$$\tilde{\mathbf{x}}_k^+ = \tilde{\mathbf{x}}_k^- + \mathbf{K}_k [\mathbf{z}_k - \mathbf{h}_k(\tilde{\mathbf{x}}_k^-)]$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$$

$$\mathbf{F}_k = \left. \frac{\partial \mathbf{f}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{(\mathbf{x}_k = \tilde{\mathbf{x}}_k^+)}$$

Project ahead:

$$\tilde{\mathbf{x}}_{k+1}^- = \mathbf{f}_k(\tilde{\mathbf{x}}_k^+)$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}_k \mathbf{P}_k^+ \mathbf{F}_k^T + \mathbf{Q}_k$$

Table 4.3: Extended Kalman Filter algorithm

5 DUSTCUBE'S FILTER

The created DustCube's filter is an Extended Kalman Filter, since the equations that govern the motion of satellites are non-linear.

Even if there exists the built-in MATLAB function *extendedKalmanfilter.m* which creates an EKF object for a continuous dynamic system through the definitions of the state function $\mathbf{f}(\mathbf{x})$, the measurement function $\mathbf{h}(\mathbf{x})$ and the initial state \mathbf{x}_0 , useful to predict and correct the estimation of the state of a system affected by noises, it has been more convenient to write an autonomous and independent EKF to appropriately control the ephemerides of the bodies which affect the motion of DustCube during the computation of the discretized estimated states $\tilde{\mathbf{x}}_{\text{sat}_k}^+$.

Therefore, the EKF has been created by adopting the classical Kalman Filter algorithm in extended version shown in [Table 4.3](#).

The main concepts of the filter will be explained in the following sections.

5.1 THE NON-LINEAR DYNAMIC SYSTEM OF DUSTCUBE

The first important aspect is to define the non-linear discretized dynamic system which characterizes the evolution in time of the state of DustCube, as has been described in [\(4.63\)](#).

The objective is to project a filter which can predict and correct the state of a satellite affected by process and measurement noises for an optical navigation concept, hence it has been decided to control the DustCube's motion as it was just affected by the central attractions of the primaries, by the gravitational disturbance of the Sun and by the Solar Radiation Pressure.

Since is necessary to select an appropriate reference frame to compute the estimation of the state, the user can choose to use the built-in [ECLIPJ2000 RF](#) or the [Didymos quasi-inertial RF](#).

Therefore, the discretized dynamic of the system [\(4.63\)](#) has been defined by considering a simpler model w.r.t. the Runge-Kutta 4th order method adopted to compute the exact solutions, since the EKF requires the expression of the Jacobian matrix $\mathbf{F}_{\text{sat}_k}$ of the discretized state vector function $\mathbf{f}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})$, which could become too complicated if $\mathbf{f}(\mathbf{x})$ is not defined in a suitable way.

Subsequently, is important to choose an appropriate measurement vector function $\mathbf{h}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})$, which brings to the computation of the Jacobian matrix $\mathbf{H}_{\text{sat}_k}$.

Indeed, remembering the EKF algorithm, which is shown in [Table 4.3](#), is easy to note that the matrix $\mathbf{H}_{\text{sat}_k}$ directly affects the Kalman gain matrix $\mathbf{K}_{\text{sat}_k}$, while $\mathbf{F}_{\text{sat}_k}$ influences it through the a-priori error covariance matrix $\mathbf{P}_{\text{sat}_k}^-$.

Chapter 5

The other contributions on the dynamic system of DustCube come from the measurement noise vector $\mathbf{v}_{\text{sat}_k}$ and the process noise vector $\mathbf{w}_{\text{sat}_k}$, since the covariance matrix of the first one $\mathbf{R}_{\text{sat}_k}$ directly affects the Kalman gain matrix $\mathbf{K}_{\text{sat}_k}$, while the covariance matrix of the second one $\mathbf{Q}_{\text{sat}_k}$ influences $\mathbf{K}_{\text{sat}_k}$ through the a-priori error covariance matrix $\mathbf{P}_{\text{sat}_k}^-$.

To keep the same writing notation of the equations of chapter 2, the DYD coordinates system and the uniformly spaced time window $\Delta T = (t_0, \dots, t_k, \dots, t_n)$ with $0 \leq k \leq n$ will be adopted in the following sections, i.e. with a time-step h , defined in seconds, which is maintained constant during the computation, so that:

$$t_{k+1} = t_k + h \quad \forall t_k \in \Delta T, k = 0, \dots, N \quad (5.1)$$

Hence, in the next pages we will define every term which is present in the non-linear discretized dynamic system (4.63) of DustCube, by taking into account the [Didymos quasi-inertial RF](#) and the time-window ΔT .

5.1.1 THE STATE VECTOR FUNCTION AND ITS JACOBIAN MATRIX

As said before, we proceed to define the state vector function $\mathbf{f}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})$ in DYD coordinates by considering a simpler dynamic model than the previously exploited RK4 method for the exact propagation of the satellite's motion, i.e. by using a single time-step propagator, which is the so-called Explicit Euler method (*Quarteroni, Sacco, Saleri, & Gervasio, 2014*), which allows to obtain the evolution in time of the state vector function as:

$$\mathbf{f}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})|_{\text{DYD}} = \begin{bmatrix} x_{\text{sat}_k} \\ y_{\text{sat}_k} \\ z_{\text{sat}_k} \\ \dot{x}_{\text{sat}_k} \\ \dot{y}_{\text{sat}_k} \\ \dot{z}_{\text{sat}_k} \end{bmatrix}_{\text{DYD}} + h \begin{bmatrix} \dot{x}_{\text{sat}_k} \\ \dot{y}_{\text{sat}_k} \\ \dot{z}_{\text{sat}_k} \\ \ddot{x}_{\text{sat}_k} \\ \ddot{y}_{\text{sat}_k} \\ \ddot{z}_{\text{sat}_k} \end{bmatrix}_{\text{DYD}} = \begin{bmatrix} x_{\text{sat}_k} \\ y_{\text{sat}_k} \\ z_{\text{sat}_k} \\ \dot{x}_{\text{sat}_k} \\ \dot{y}_{\text{sat}_k} \\ \dot{z}_{\text{sat}_k} \end{bmatrix}_{\text{DYD}} + h \begin{bmatrix} \dot{x}_{\text{sat}_k} \\ \dot{y}_{\text{sat}_k} \\ \dot{z}_{\text{sat}_k} \\ a_{x_{\text{sat}_k}} \\ a_{y_{\text{sat}_k}} \\ a_{z_{\text{sat}_k}} \end{bmatrix}_{\text{DYD}} \quad (5.2)$$

The set of the accelerations acting on DustCube that influences acceleration contributions $a_{x_{\text{sat}_k}}$, $a_{y_{\text{sat}_k}}$ and $a_{z_{\text{sat}_k}}$ of equation (5.2) are computable by summing up the single accelerations due to central attractions of the primaries, to solar gravity disturbance and to SRP.

This issue has been already faced up in points 2 and 6 of the section [2.4.3 The orbital propagator's scripts](#) of chapter 2, so that the accelerations acting on DustCube, which are $\mathbf{a}_{\text{sat}_{G_D}}(t_k)$ and $\mathbf{a}_{\text{sat}_{G_d}}(t_k)$ for Didymain and Didymoon, respectively, $\mathbf{a}_{\text{sat}_{G_{\text{sun}}}}(t_k)$ for the Sun and $\mathbf{a}_{\text{sat}_{\text{SRP}}}(t_k)$, with or w/o considering the shadow functions of the primaries, for the SRP, are computed by exploiting

the already mentioned MATLAB scripts *central_attraction.m*, *body_disturbance.m*, *SRP_Montenbruck_complete.m* and *SRP_data.m*.

The final expression of the acceleration $\mathbf{a}_{\text{sat}_k} \equiv \mathbf{a}_{\text{sat}}(t_k)$ is (omitting DYD subscript):

$$\mathbf{a}_{\text{sat}_k} = \mathbf{a}_{\text{sat}_{G_D}}(t_k) + \mathbf{a}_{\text{sat}_{G_d}}(t_k) + \mathbf{a}_{\text{sat}_{SRP}}(t_k) + \mathbf{a}_{\text{sat}_{G_{\text{sun}}}}(t_k), \quad (5.3)$$

where the single contributions have been defined in the previous chapters as (omitting DYD):

$$\begin{aligned} \mathbf{a}_{\text{sat}_{G_D}}(t_k) &= -\frac{\mu_D}{\|\mathbf{r}_{\text{sat-D}}(t)\|^3} \mathbf{r}_{\text{sat-D}}(t) \\ \mathbf{a}_{\text{sat}_{G_d}}(t_k) &= -\frac{\mu_d}{\|\mathbf{r}_{\text{sat-d}}(t)\|^3} \mathbf{r}_{\text{sat-d}}(t) \\ \mathbf{a}_{\text{sat}_{G_{\text{sun}}}}(t_k) &= \mu_{\text{sun}} \left[\frac{\mathbf{s}(t_k)}{\|\mathbf{s}(t_k)\|^3} - \frac{\mathbf{r}_s(t_k)}{\|\mathbf{r}_s(t_k)\|^3} \right] \\ \mathbf{a}_{\text{sat}_{SRP}}(t_k) &= \frac{\mathbf{f}_{\text{sat}_{SRP}}(t_k)}{m_{\text{sat}}}, \end{aligned} \quad (5.4)$$

where $\mathbf{r}_{\text{sat-D}}(t)$ and $\mathbf{r}_{\text{sat-d}}(t)$ have been shown in eqs (1.2), $\mathbf{s}(t)$ and $\mathbf{r}_s(t)$ have been clarified in eq. (1.35) and $\mathbf{f}_{\text{sat}_{SRP}}(t)$ has been demonstrated in eq. (1.23).

Therefore, renaming the state of the satellite as:

$$\mathbf{x}_{\text{sat}_k} = (x_{\text{sat}_k}, y_{\text{sat}_k}, z_{\text{sat}_k}, \dot{x}_{\text{sat}_k}, \dot{y}_{\text{sat}_k}, \dot{z}_{\text{sat}_k})^T = (x_{1k}, x_{2k}, x_{3k}, x_{4k}, x_{5k}, x_{6k})^T, \quad (5.5)$$

the Jacobian $\mathbf{F}_{\text{sat}_k}$ (eq. (4.64)) is easily computable for the first three components of the state vector function $\mathbf{f}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})$, while is useful to separate the last three components of $\mathbf{f}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})$ in their fundamental contributions.

Finally, the Jacobian $\mathbf{F}_{\text{sat}_k}$ is a 6×6 matrix composed by four 3×3 “blocks”, namely (omitting DYD):

$$\mathbf{F}_{\text{sat}_k} = \left. \frac{\partial \mathbf{f}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})}{\partial \mathbf{x}_{\text{sat}_k}} \right|_{(\mathbf{x}_{\text{sat}_k} = \tilde{\mathbf{x}}_k^+)} = \begin{bmatrix} 1 & 0 & 0 & h & 0 & 0 \\ 0 & 1 & 0 & 0 & h & 0 \\ 0 & 0 & 1 & 0 & 0 & h \\ \frac{\partial f_{4k}}{\partial x_{1k}} & \dots & \frac{\partial f_{4k}}{\partial x_{3k}} & 1 & 0 & 0 \\ \vdots & \ddots & \vdots & 0 & 1 & 0 \\ \frac{\partial f_{6k}}{\partial x_{1k}} & \dots & \frac{\partial f_{6k}}{\partial x_{3k}} & 0 & 0 & 1 \end{bmatrix}_{(\mathbf{x}_{\text{sat}_k} = \tilde{\mathbf{x}}_k^+)} \quad (5.6)$$

The components of $\mathbf{F}_{\text{sat}_k}(4:6;1:3)$ consist of the partial derivatives of the accelerations previously shown in eqs (5.4).

The formulations of the state vector function and of its Jacobian matrix F_k have been implemented in the MATLAB script *statetransition_4bodies_SRP.m*.

The partial derivatives of $\mathbf{a}_{\text{sat}_{G_D}}$, $\mathbf{a}_{\text{sat}_{G_d}}$ and $\mathbf{a}_{\text{sat}_{G_{\text{sun}}}}$ are straightforward to compute, while the ones of $\mathbf{a}_{\text{sat}_{\text{SRP}}}$ require the usage of the *Symbolic Tool* of MATLAB, due especially to the presence of the terms $\cos(\theta(t_k))$ and $\cos^2(\theta(t_k))$ in equation (1.23).

5.1.2 THE MEASUREMENT VECTOR FUNCTION AND ITS JACOBIAN MATRIX

Once defined the state vector function $\mathbf{f}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})$ and its Jacobian matrix $\mathbf{F}_{\text{sat}_k}$, we proceed to state the measurement vector function $\mathbf{h}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})$ and its Jacobian matrix $\mathbf{H}_{\text{sat}_k}$.

The measurement vectors $\mathbf{z}_{\text{sat}_k}$ are strictly connected to the considered navigation concept.

Since DustCube will operate inside Didymos binary system by using the optical measurements from the images captured by the cameras, the measurement vector function $\mathbf{h}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})$ must consider the LOS vectors (see section 3.1 [Inertial direction evaluation: Lines of sight \(LOS\)](#) for more details), which are the vectors directed from DustCube towards the target bodies (Figure 5.1).

Therefore, the adopted dynamic system is based on measurements in terms of range from the target body, i.e. the distance from the target, and in terms of inertial direction, i.e. the unit vector directed from the S/C to the target body (*Lasagni Manghi, Zannoni, Modenini, & Tortora, 2016, February 2*), so that these measurements are coherent with the image processing techniques adopted for the mission, as described in section 3.5 [Optical Navigation concept for DustCube mission](#).

In particular, since DustCube will acquire images through two infrared cameras pointed towards the primaries, the most exploited technique will be the [Center-finding by Analytic Function Fitting](#) (Figure 3.7), because the pictures of the two target bodies will always be content in the cameras' FoVs, which allows to compute the LOS inertial direction measurements and the ranges of the observations too, since is always possible to correlate the size of every target body within the image with the known size of the corresponding celestial body (*Christian, 2015*).

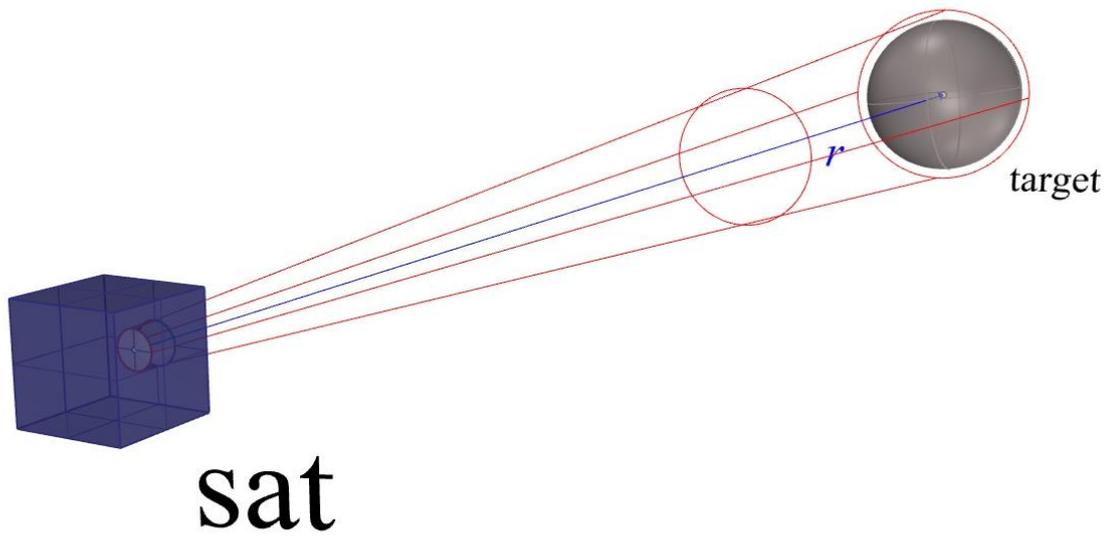


Figure 5.1: Tracking and ranging measurements of a target body

Therefore, since DustCube will navigate in a binary system, it will carry on at least two cameras mounted onboard:

- one camera pointed towards Didymain;
- one camera pointed towards Didymoon.

Hence, the measurements will be:

- the norm of the vector directed from DustCube towards Didymain;
- the norm of the vector directed from DustCube towards Didymoon;
- the three components of the unit vector directed towards Didymain;
- the three components of the unit vector directed towards Didymoon.

Decomposing the position vectors of DustCube, Didymain and Didymoon w.r.t. the CoM of Didymos in DYD coordinates as (omitting DYD subscript and the time dependence):

$$\begin{aligned}
 \mathbf{r}_{\text{sat}} &= (x_{\text{sat}_1}, x_{\text{sat}_2}, x_{\text{sat}_3})^T \\
 \mathbf{r}_{\text{D}} &= (x_{\text{D}_1}, x_{\text{D}_2}, x_{\text{D}_3})^T \\
 \mathbf{r}_{\text{d}} &= (x_{\text{d}_1}, x_{\text{d}_2}, x_{\text{d}_3})^T,
 \end{aligned} \tag{5.7}$$

is possible to obtain the LOS unit vectors directed towards the primaries of Didymos system by considering the opposite of $\mathbf{r}_{\text{sat-D}}$ and $\mathbf{r}_{\text{sat-d}}$ described in eqs (1.2), namely:

$$\begin{aligned}\widehat{\mathbf{LOS}}_D &= \frac{\mathbf{r}_{D-\text{sat}}}{\|\mathbf{r}_{D-\text{sat}}\|} = \frac{\mathbf{r}_D - \mathbf{r}_{\text{sat}}}{\|\mathbf{r}_D - \mathbf{r}_{\text{sat}}\|} \\ \widehat{\mathbf{LOS}}_d &= \frac{\mathbf{r}_{d-\text{sat}}}{\|\mathbf{r}_{d-\text{sat}}\|} = \frac{\mathbf{r}_d - \mathbf{r}_{\text{sat}}}{\|\mathbf{r}_d - \mathbf{r}_{\text{sat}}\|}\end{aligned}\quad (5.8)$$

Since every LOS direction is determined through the unit vector directed from the camera towards the target body, which has three components that are not mutually independent, is not convenient to operate with 8 measurements.

This problem is easily avoidable by switching from the Cartesian coordinate system to the spherical one, i.e. by reasoning in terms of range ρ , longitude λ and latitude ϕ .

Indeed, the pseudo-range, which would be the norm of the LOS vector, represents the distance from the target body, while the angles λ and ϕ determine the direction of the LOS unit vector.

Since the built-in MATLAB function *acosd* computes an angle that spans from 0° to 180° , is useful to switch from the latitude angle ϕ to the colatitude one δ (Figure 5.2).

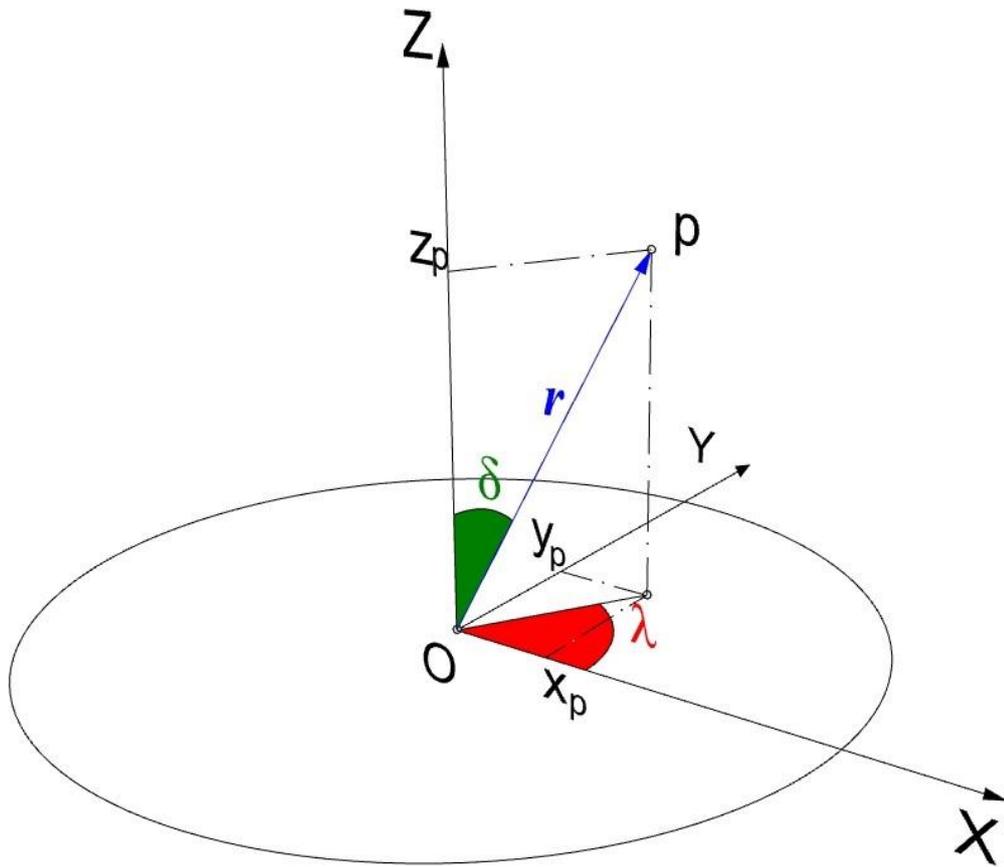


Figure 5.2: Representation of the longitude λ and the colatitude δ of a generic position vector \mathbf{r}

Finally, the considered measurement vector \mathbf{z}_k is featured by 6 components, which are the spherical coordinates of the two position vectors $\mathbf{r}_{D-\text{sat}}$ and $\mathbf{r}_{d-\text{sat}}$ of eqs (5.8).

The passage from the Cartesian coordinate system to the spherical one is possible by exploiting the inverse of the following classical transformation equations:

$$\begin{cases} x = \rho \sin(\delta) \cos(\lambda) \\ y = \rho \sin(\delta) \sin(\lambda) \\ z = \rho \cos(\delta) \end{cases}, \quad (5.9)$$

and the well-known Euclidean distance, which bring to the following definitions for the range ρ , the longitude λ and the colatitude δ :

$$\begin{cases} \rho = \sqrt{x^2 + y^2 + z^2} \\ \lambda = \tan^{-1}\left(\frac{y}{x}\right) \\ \delta = \cos^{-1}\left(\frac{z}{\rho}\right) \end{cases} \quad \text{with} \quad \begin{cases} \rho > 0 \\ \lambda \in [-180^\circ, 180^\circ] \\ \delta \in [0^\circ, 180^\circ] \end{cases} \quad (5.10)$$

Operating in degrees, since the built-in MATLAB functions *atan2* and *acosd* compute angles included in the intervals $[-90^\circ, 90^\circ]$ and $[0^\circ, 180^\circ]$ respectively, the colatitude δ can be directly calculated w/o restrictions, since $\rho > 0$, while the longitude λ requires an appropriate definition of the conditions on x and y to compute the actual longitude of the position vector, which are:

$$\begin{cases} \lambda = \tan^{-1}\left(\frac{y}{x}\right) & \text{iff } x > 0 \\ \lambda = \tan^{-1}\left(\frac{y}{x}\right) + 180^\circ & \text{iff } x < 0 \text{ and } y > 0 \\ \lambda = \tan^{-1}\left(\frac{y}{x}\right) - 180^\circ & \text{iff } x < 0 \text{ and } y < 0 \\ \lambda = 90^\circ & \text{iff } x = 0 \text{ and } y > 0 \\ \lambda = -90^\circ & \text{iff } x = 0 \text{ and } y < 0 \\ \lambda = \text{undefined} & \text{iff } x = y = 0 \end{cases} \quad (5.11)$$

Finally, considering the two LOS vectors, the measurement function $\mathbf{h}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})$ is defined as:

$$\mathbf{h}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})|_{\text{DYD}} = \begin{bmatrix} \|\mathbf{r}_{\text{D-sat}_k}\| \\ \lambda_{\text{D-sat}_k} \\ \delta_{\text{D-sat}_k} \\ \|\mathbf{r}_{\text{d-sat}_k}\| \\ \lambda_{\text{d-sat}_k} \\ \delta_{\text{d-sat}_k} \end{bmatrix}_{\text{DYD}} \quad (5.12)$$

Therefore, the Jacobian matrix $\mathbf{H}_{\text{sat}_k}$ becomes a 6×6 matrix which can be obtained by computing the single partial derivatives shown in equation (4.57), which has been implemented in the MATLAB script *jacobian_H_thesis.m*.

5.1.3 THE USEFUL REFERENCE FRAMES FOR THE MEASUREMENTS

To maintain a coherent LOS errors computation, it is convenient to define other three non-inertial reference frames, which are the satellite body-fixed RF and the two cameras tracking-fixed RFs:

1. DustCube body-fixed reference frame (SAT subscript): is the body-fixed reference frame of DustCube, which is centered in the CoM of the satellite. Since its main objective will be to examine the DART impact, is convenient to adopt an orientation of DustCube which is coherent with this specific constrain. Therefore, the body-axes of the satellite are defined like the final concept approved by the technical report *DustCube Technical Note 2: Mission Concept and Preliminary Assessment* (Aguado, et al., 2016, May 6), which is shown in Figure 5.3.

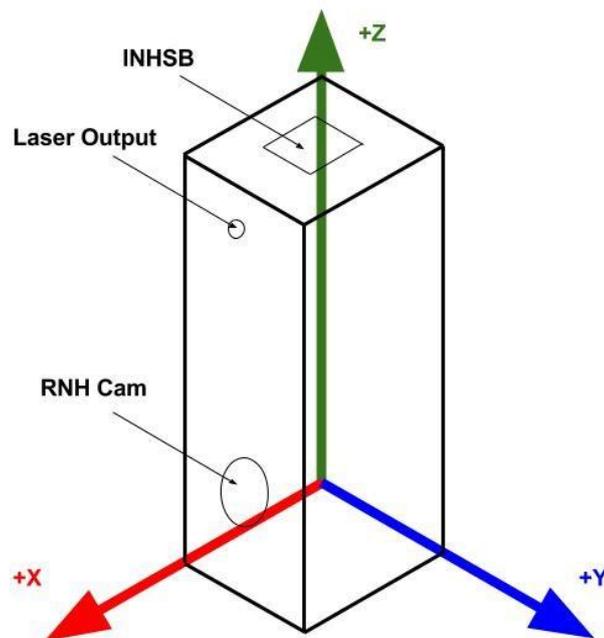


Figure 5.3: Body Reference Frame for DustCube (Aguado, et al., 2016, May 6)

Therefore, the axes of the Reference Frame are (Aguado, et al., 2016, May 6):

- **+X axis:** from the CoM to the face of the RNH directed towards Didymoon;
- **+Z axis:** from the CoM to the INH Sample Bay Inlet;
- **+Y axis:** it completes the right-handed RF.

Furthermore, the six faces of the DustCube S/C have been named adopting this RF.

As said before, since DustCube will analyze the DART impact with Didymoon and it will always move inside Didymos system remaining in the mean plane of the mutual orbit of the primaries, its adopted orientation is defined by maintaining the **+X axis** pointing towards Didymoon, the **+Y axis** directed as the cross product between the position vectors of Didymain and Didymoon w.r.t. the satellite frame and the **+Z axis** completes the right-handed orientation frame.

The MATLAB script *sat_RF.m* defines the body-fixed RF of DustCube and continuously computes the rotation matrix between this frame and the adopted inertial reference frame (see section [A.1.5 Transformations in non-inertial coordinates by using rotation matrices](#) for more details).

2. *Didymain and Didymoon tracking cameras RFs (DCAM e dCAM subscripts, respectively)*: are the camera-fixed RFs related to the two tracking cameras mounted onboard DustCube, pointing towards Didymain and Didymoon and centered in the respective geometric centers of the cameras' lens. Since DustCube will acquire images of Didymoon after the collision of DART and will track the two primaries as target bodies for navigation purposes, the solution adopted for the configuration of the two cameras onboard DustCube is the final concept approved by the technical report *DustCube Technical Note 2: Mission Concept and Preliminary Assessment* (Aguado, et al., 2016, May 6), which is shown in [Figure 5.4](#).

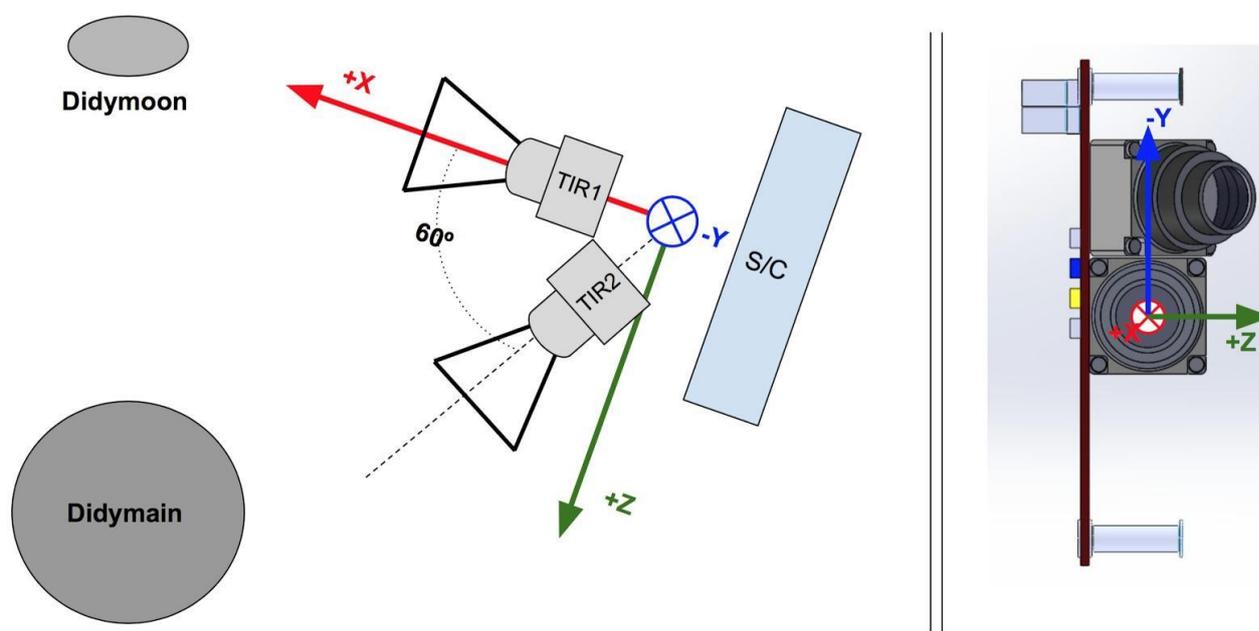


Figure 5.4: Proposed cameras' configuration (Aguado, et al., 2016, May 6)

Therefore, every camera mounted onboard DustCube has a fixed RF defined as:

- **+X axis:** is the pointing direction, ideally directed towards the target body's CoM;
- **-Y axis:** perpendicular to the main symmetry axis;
- **+Z axis:** it completes the right-handed RF.

As said before, since DustCube will track the primaries to satisfy the navigations purposes, the adopted orientation of every camera defined by maintaining every pointing axis **+X axis** of the two cameras directed towards the target body.

The MATLAB scripts *cam_RF_main.m* and *cam_RF_moon.m* define the relative cameras' RFs of DustCube and continuously compute the rotation matrices between the cameras' frames and the adopted DustCube's body fixed RF.

5.1.4 THE PROCESS AND THE MEASUREMENT ERRORS

After the definition of the state vector function $\mathbf{f}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})$ and of the measurement vector function $\mathbf{h}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})$, with their relative Jacobian matrices $\mathbf{F}_{\text{sat}_k}$ and $\mathbf{H}_{\text{sat}_k}$, is important to define a suitable set of process error vectors $\mathbf{w}_{\text{sat}_k}$ and measurement ones $\mathbf{v}_{\text{sat}_k}$ for the considered mission.

As said before, since the objective of this work of thesis is to implement a navigation filter simulation, it has been decided to consider the state of DustCube as it was subjected to the influence of the central attractions of the primaries, Didymain and Didymoon, to the solar gravitational disturbance and to the Solar Radiation Pressure, so that the acceleration vectors which arise from the addition of the spherical harmonics due to the non-spherical shape of the primaries in the computation of the gravitational attractions can be considered as the process noise vectors.

Therefore, considering the evaluation of the state vector of DustCube obtained through the simplified state's dynamic, i.e. by exploiting the Explicit Euler method for the propagation of the state, its evolution in time becomes (see 1st equation of (4.63) w/o input vector):

$$\mathbf{x}_{\text{sat}_{k+1}} = \begin{bmatrix} x_{\text{sat}_k} + h \dot{x}_{\text{sat}_k} \\ y_{\text{sat}_k} + h \dot{y}_{\text{sat}_k} \\ z_{\text{sat}_k} + h \dot{z}_{\text{sat}_k} \\ \dot{x}_{\text{sat}_k} + h a_{x_{\text{sat}_k}} \\ \dot{y}_{\text{sat}_k} + h a_{y_{\text{sat}_k}} \\ \dot{z}_{\text{sat}_k} + h a_{z_{\text{sat}_k}} \end{bmatrix}_{\text{DYD}} + \begin{bmatrix} w_{x_{\text{sat}_k}} \\ w_{y_{\text{sat}_k}} \\ w_{z_{\text{sat}_k}} \\ w_{\dot{x}_{\text{sat}_k}} \\ w_{\dot{y}_{\text{sat}_k}} \\ w_{\dot{z}_{\text{sat}_k}} \end{bmatrix}_{\text{DYD}}, \quad (5.13)$$

where the first term comes from the eq (5.2), while the second one is the process error vector \mathbf{w}_k , which must be linked to the augmented set of accelerations acting on DustCube due to spherical harmonic coefficients.

Considering a simplified second-order model in continuous time for the evaluation of the position and the velocity of a point mass in 1D, which is the so-called accelerated rectilinear motion, where $p(t)$ is the position, $v(t) = \dot{p}(t)$ is the velocity and $w(t) = \ddot{p}(t)$ is the acceleration, the dynamic of the state $\mathbf{x}(t) = [p(t), \dot{p}(t)]^T$ w/o process errors simply becomes (*Shimkin, 2009*):

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t), \quad (5.14)$$

where $w(t)$ can be considered as the input term.

Since is more convenient to operate in a discretized time-window for a computational point of view, discretizing the equation (5.14) through the well-known forward finite difference formula (*Quarteroni, Sacco, Saleri, & Gervasio, 2014*), we obtain the Explicit Euler method in 1D:

$$\mathbf{x}_{k+1} = \begin{bmatrix} p_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} + \begin{bmatrix} 0 \\ h \end{bmatrix} w_k \quad (5.15)$$

Now we can suppose that there exists a non-avoidable scalar small random acceleration Δa_k , which is a white gaussian noise with zero mean term, that acts on the point mass during its motion, corrupting the dynamic of the system and therefore inhibiting the exact evaluation of the point mass' state (*Shimkin, 2009*).

We also suppose that this small acceleration remains constant during the passage from the k^{th} instant of time to the $(k+1)^{\text{th}}$ one, which always occurs with a constant time step h .

Under these assumptions, the random small acceleration Δa_k produces an error vector which directly modifies the velocity evolution of the system and therefore its position too.

This error vector can be easily obtained as:

$$\begin{bmatrix} \Delta p_k \\ \Delta v_k \end{bmatrix} = \begin{bmatrix} h^2/2 \\ h \end{bmatrix} \Delta a_k \quad (5.16)$$

Starting from this simple example in 1D, is possible to extend the reasoning to our 3D case, which brings to the definition of the process error vector $\mathbf{w}_{\text{sat}_k}$ for DustCube mission:

$$\mathbf{w}_{\text{sat}_k} = \begin{bmatrix} h^2/2 \Delta a_{x_{\text{sat}_k}} \\ h^2/2 \Delta a_{y_{\text{sat}_k}} \\ h^2/2 \Delta a_{z_{\text{sat}_k}} \\ h \Delta a_{x_{\text{sat}_k}} \\ h \Delta a_{y_{\text{sat}_k}} \\ h \Delta a_{z_{\text{sat}_k}} \end{bmatrix}, \quad (5.17)$$

so that, the final dynamic of the state of DustCube, starting from the equation (5.19), becomes (omitting DYD subscript):

$$\mathbf{x}_{\text{sat}_{k+1}} = \begin{bmatrix} x_{\text{sat}_k} + h \dot{x}_{\text{sat}_k} \\ y_{\text{sat}_k} + h \dot{y}_{\text{sat}_k} \\ z_{\text{sat}_k} + h \dot{z}_{\text{sat}_k} \\ \dot{x}_{\text{sat}_k} + h a_{x_{\text{sat}_k}} \\ \dot{y}_{\text{sat}_k} + h a_{y_{\text{sat}_k}} \\ \dot{z}_{\text{sat}_k} + h a_{z_{\text{sat}_k}} \end{bmatrix} + \begin{bmatrix} h^2/2 \Delta a_{x_{\text{sat}_k}} \\ h^2/2 \Delta a_{y_{\text{sat}_k}} \\ h^2/2 \Delta a_{z_{\text{sat}_k}} \\ h \Delta a_{x_{\text{sat}_k}} \\ h \Delta a_{y_{\text{sat}_k}} \\ h \Delta a_{z_{\text{sat}_k}} \end{bmatrix}, \quad (5.18)$$

where the last term represents the so-called process error vector $\mathbf{v}_{\text{sat}_k}$, which can be properly evaluated to compute the process covariance matrix $\mathbf{Q}_{\text{sat}_k}$, which has an important influence on the behavior of the dynamic filter.

The measurement errors $\mathbf{v}_{\text{sat}_k}$ are strictly linked to the typology of the measurements.

Since the measurement vector $\mathbf{z}_{\text{sat}_k}$ is characterized by 6 components which are the ranges of DustCube from Didymain and Didymoon and the LOS directions of the two cameras pointed towards the primaries, which are summarized by the longitude λ and the colatitude δ (Figure 5.2) of the unit vectors $\hat{\mathbf{r}}_{\text{D-sat}}$ and $\hat{\mathbf{r}}_{\text{d-sat}}$, we have to consider possible random errors in the computation of the measurements of the ranges from the CoMs of the primaries and in the tracking of the position of the CoMs of the primaries.

Therefore, the possible measurement errors should be due to the tracking errors of the CoMs of Didymain and Didymoon and to the fitting errors of the bodies' shapes (Figure 5.1), so that they are strictly linked to the camera's FoVs.

Since the ranges ρ_{D} and ρ_{d} between DustCube and the CoMs of the primaries are the norms of the two vectors $\mathbf{r}_{\text{D-sat}}$ and $\mathbf{r}_{\text{d-sat}}$, they are independent by the RF used to achieve them, so that they can be easily obtained by computing the norms of the two vectors as defined in [Didymos coordinates](#).

Therefore, it can be suitable to consider norm errors which are 1% of the ideal measured ranges and which can be positive or negative, that is easy to implement by multiplying the two ranging errors with a scalar factor $k = \pm 1$ that varies randomly during the computation:

$$\begin{aligned} \Delta \rho_{\text{D}}(t_k) &= k \frac{\rho_{\text{D}_{ideal}}(t_k)}{100} \Rightarrow \rho_{\text{D}_{true}}(t_k) = \rho_{\text{D}_{ideal}}(t_k) + \Delta \rho_{\text{D}}(t_k) \\ \Delta \rho_{\text{d}}(t_k) &= k \frac{\rho_{\text{d}_{ideal}}(t_k)}{100} \Rightarrow \rho_{\text{d}_{true}}(t_k) = \rho_{\text{d}_{ideal}}(t_k) + \Delta \rho_{\text{d}}(t_k) \end{aligned} \quad (5.19)$$

Regarding the LOS direction errors, the unit vectors $\widehat{\mathbf{LOS}}_{D_{ideal}}$ and $\widehat{\mathbf{LOS}}_{d_{ideal}}$ have been corrupted by considering small deviation angles, that are suitably computed through the definition of small deviations from the ideal pointing in camera's RFs.

Considering a target picture captured by one camera as in [Figure 5.5](#), is easy to visualize the random small deviations \mathbf{e}_p from the ideal pointing directed towards the CoM of the target body, which have a norm that is maintained between 0.5% - 1% of the ideal measured range.

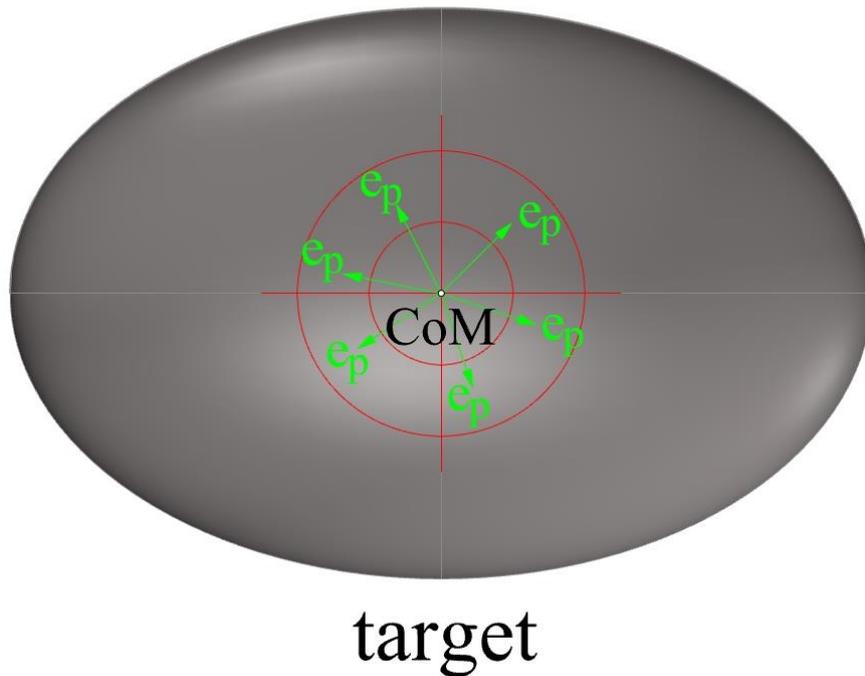


Figure 5.5: Small deviation errors from the CoM's pointing of the target body

Defining the vector directed from the satellite towards the target body's CoM with \mathbf{r}_{ideal} , which is aligned with the unit vector $\hat{\mathbf{i}}_{CAM}$, and the corrupted one with \mathbf{r}_{true} , is easy to understand how to compute the corrupted direction measurement $\hat{\mathbf{r}}_{true}$ by visualizing the [Figure 5.6](#), where the reader can also observe the camera RF, identified with the canonical basis $(\hat{\mathbf{i}}_{CAM}, \hat{\mathbf{j}}_{CAM}, \hat{\mathbf{k}}_{CAM})$.

Hence, considering the ideal LOS directions of the two cameras mounted onboard DustCube, $\widehat{\mathbf{LOS}}_{D_{ideal}}$ and $\widehat{\mathbf{LOS}}_{d_{ideal}}$, is possible to obtain the corrupted directions $\widehat{\mathbf{LOS}}_{D_{true}}$ and $\widehat{\mathbf{LOS}}_{d_{true}}$ through the computation of the small deviations \mathbf{e}_{p_D} and \mathbf{e}_{p_d} related to the two targets bodies.

Obviously, since the objective is to simulate the actual behaviour of DustCube, the two small deviations are calculated in the cameras' RFs, so that the corrupted LOS measurements are computed in the cameras' RFs too.

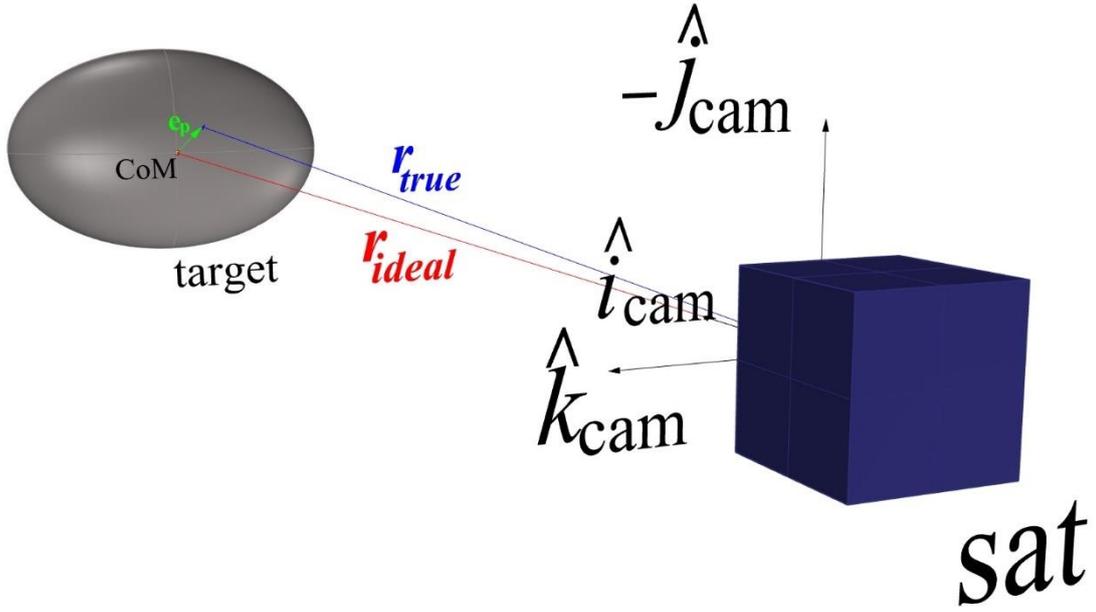


Figure 5.6: Corrupted target pointing of a camera mounted onboard a satellite

Hence, starting from the two ideal ranges $\rho_{D_{ideal}}$ and $\rho_{d_{ideal}}$, we can fix the two following norms:

$$\begin{aligned} \|\mathbf{e}_{p_D}|_{CAM}(t_k)\| &= \frac{\rho_{D_{ideal}}(t_k)}{\mathbb{C}} & \mathbb{C} = 100, 200 \\ \|\mathbf{e}_{p_d}|_{CAM}(t_k)\| &= \frac{\rho_{d_{ideal}}(t_k)}{\mathbb{C}} & \mathbb{C} = 100, 200, \end{aligned} \quad (5.20)$$

where \mathbb{C} determines if the software must consider two deviation errors of 1% or 0.5% of the two norms $\rho_{D_{ideal}}$ and $\rho_{d_{ideal}}$, that means the deviation angle α is equal to approximately 0.56° or 0.28° respectively, which represent almost 2% and 1% of the general FoV., that can be regarded as similar to the actual deviation angles

Obviously, the two small deviations \mathbf{e}_{p_D} and \mathbf{e}_{p_d} are ideally contained in the cameras' planes, upon which lay the unit vectors $(\hat{\mathbf{j}}_{CAM_D}, \hat{\mathbf{k}}_{CAM_D})$ and $(\hat{\mathbf{j}}_{CAM_d}, \hat{\mathbf{k}}_{CAM_d})$, therefore, since we want to obtain random errors in terms of direction, is possible to rotate the unit vectors $\hat{\mathbf{k}}_{CAM_D}$ and $\hat{\mathbf{k}}_{CAM_d}$ (or $\hat{\mathbf{j}}_{CAM_D}$ and $\hat{\mathbf{j}}_{CAM_d}$, which is the same) around the versors $\hat{\mathbf{i}}_{CAM_D}$ and $\hat{\mathbf{i}}_{CAM_d}$, respectively, of casual angles γ_D and γ_d , which are computed as:

$$\begin{aligned} \gamma_D &= c_D k_D \pi & \text{with } c_D \in [0,1] \text{ and } k_D = 1,2,3, \dots, \infty \\ \gamma_d &= c_d k_d \pi & \text{with } c_d \in [0,1] \text{ and } k_d = 1,2,3, \dots, \infty, \end{aligned} \quad (5.21)$$

where the fortuity on the choice of γ_D and γ_d is achieved in MATLAB through the call `rand(1)`, which is exploited to obtain the two random values c_D and c_d , and the call `randi(intmax)`, which is used to compute the two random integer values k_D and k_d , that are comprises from 1 to `intmax`, that is the maximum integer value storable by MATLAB.

Once the unit vectors $\widehat{\mathbf{k}}_{\text{CAM}_D}$ and $\widehat{\mathbf{k}}_{\text{CAM}_d}$ have been randomly rotated, we have obtained the unit vectors $\widehat{\mathbf{e}}_{p_D}$ and $\widehat{\mathbf{e}}_{p_d}$, that are necessary to finally achieve the expressions of the two small vectors \mathbf{e}_{p_D} and \mathbf{e}_{p_d} , namely:

$$\begin{aligned}\mathbf{e}_{p_D}|_{\text{CAM}}(t_k) &= \frac{\rho_{D_{ideal}}(t_k)}{200} \widehat{\mathbf{e}}_{p_D}|_{\text{CAM}}(t_k) \\ \mathbf{e}_{p_d}|_{\text{CAM}}(t_k) &= \frac{\rho_{d_{ideal}}(t_k)}{200} \widehat{\mathbf{e}}_{p_d}|_{\text{CAM}}(t_k),\end{aligned}\quad (5.22)$$

which bring to the following definitions of the two corrupted LOS vectors:

$$\begin{aligned}\mathbf{LOS}_{D_{true}}|_{\text{CAM}}(t_k) &= \mathbf{LOS}_{D_{ideal}}|_{\text{CAM}}(t_k) + \mathbf{e}_{p_D}|_{\text{CAM}}(t_k) \\ \mathbf{LOS}_{d_{true}}|_{\text{CAM}}(t_k) &= \mathbf{LOS}_{d_{ideal}}|_{\text{CAM}}(t_k) + \mathbf{e}_{p_d}|_{\text{CAM}}(t_k)\end{aligned}\quad (5.23)$$

The consequent unit vectors $\widehat{\mathbf{LOS}}_{D_{true}}$ and $\widehat{\mathbf{LOS}}_{d_{true}}$ must be transformed in Didymos coordinates through their proper rotation matrices, namely:

$$\begin{aligned}\widehat{\mathbf{LOS}}_{D_{true}}|_{\text{DYD}}(t_k) &= [\mathbf{T}_{\text{DYD-SAT}}(t_k)]^T [\mathbf{T}_{\text{SAT-DCAM}}(t_k)]^T \widehat{\mathbf{LOS}}_{D_{true}}|_{\text{CAM}}(t_k) \\ \widehat{\mathbf{LOS}}_{d_{true}}|_{\text{DYD}}(t_k) &= [\mathbf{T}_{\text{DYD-SAT}}(t_k)]^T [\mathbf{T}_{\text{SAT-dCAM}}(t_k)]^T \widehat{\mathbf{LOS}}_{d_{true}}|_{\text{CAM}}(t_k),\end{aligned}\quad (5.24)$$

Once obtained the true unit vectors in Didymos coordinates, they must be compared to the ideal unit vectors $\widehat{\mathbf{LOS}}_{D_{ideal}}$ and $\widehat{\mathbf{LOS}}_{d_{ideal}}$, always defined in Didymos coordinates, in terms of longitude λ and colatitude δ (see eqs (5.10)), which brings to the definition of the measurement errors $\Delta\lambda_D$, $\Delta\lambda_d$, $\Delta\delta_D$ and $\Delta\delta_d$, that finally compose the measurement error vector $\mathbf{v}_{\text{sat}_k}$:

$$\mathbf{v}_{\text{sat}_k} = \begin{bmatrix} \Delta\rho_{Dk} \\ \Delta\lambda_{Dk} \\ \Delta\delta_{Dk} \\ \Delta\rho_{dk} \\ \Delta\lambda_{dk} \\ \Delta\delta_{dk} \end{bmatrix}, \quad (5.25)$$

Chapter 5

which determines the covariance measurement error matrix $\mathbf{R}_{\text{sat}_k}$, that will be explained in the following section.

The error vector $\mathbf{v}_{\text{sat}_k}$ is obtained, for every iterative computation, through the MATLAB scripts *measurement_error_thesis_cam_main.m* and *measurement_error_thesis_cam_main.m*.

5.1.5 THE COVARIANCE ERROR MATRICES \mathbf{Q}_{sat} AND \mathbf{R}_{sat}

The covariance error matrices \mathbf{Q}_k and \mathbf{R}_k , which have been presented in the eqs (4.40) of Chapter 4, are determined starting from the process and the measurement error vectors \mathbf{w}_k and \mathbf{v}_k .

It is important to remark that the rigorous definition of the matrices \mathbf{Q}_k and \mathbf{R}_k states that they are two diagonal matrices, where the principal diagonal is composed by the variances of the single components of the two random vectors \mathbf{w}_k and \mathbf{v}_k i.e.:

$$\begin{aligned}\mathbf{Q}_k &= \text{diag}(\sigma_1^2, \dots, \sigma_n^2) && \text{with } \sigma_i^2 = \text{var}(w_k^i) \\ \mathbf{R}_k &= \text{diag}(\sigma_1^2, \dots, \sigma_m^2) && \text{with } \sigma_i^2 = \text{var}(v_k^i),\end{aligned}\tag{5.26}$$

while in the case of DustCube the error vectors $\mathbf{w}_{\text{sat}_k}$ and $\mathbf{v}_{\text{sat}_k}$ are not strictly random, which brings to the conclusion that the out-of-diagonal elements of $\mathbf{Q}_{\text{sat}_k}$ and $\mathbf{R}_{\text{sat}_k}$ are not null, that means single components of $\mathbf{w}_{\text{sat}_k}$ and $\mathbf{v}_{\text{sat}_k}$ are not strictly uncorrelated each other.

However, even if the process and the measurement error vectors are slightly deterministic, the Extended Kalman Filter remains valid to evaluate the state of DustCube, despite the control method is not optimal. Therefore, the matrices $\mathbf{Q}_{\text{sat}_k}$ and $\mathbf{R}_{\text{sat}_k}$ must be defined starting from the error vectors and considering some useful techniques.

Since the simplified dynamic of the state has been defined through the forward finite difference formula, obtaining the Explicit Euler method in 3D as done in the equation (5.18), is necessary to define a process error covariance matrix $\mathbf{Q}_{\text{sat}_k}$ that properly considers the influence of the single components of the acceleration error, which are $\Delta a_{x_{\text{sat}_k}}$, $\Delta a_{y_{\text{sat}_k}}$ and $\Delta a_{z_{\text{sat}_k}}$, on the estimations of the position and the velocity of DustCube.

Visualizing the state equation (5.18), is easy to observe that these errors is linked to the squared of the time-step h for the position evaluation and to the single time-step h for the velocity evaluation.

Without losing consistency, it is predictable that the principal diagonal of the matrix $\mathbf{Q}_{\text{sat}_k}$ is composed by the squares of the errors occurred for every component of the state vector $\mathbf{x}_{\text{sat}_k}$, while the out-of-diagonal elements of $\mathbf{Q}_{\text{sat}_k}$ are related to the crossing influences between the components of the same state vector $\mathbf{x}_{\text{sat}_k}$

Also, since the errors are slightly deterministic, is useful to maintain the matrix constant during the computation and equal to a matrix \mathbf{Q}_{sat} , that allows to reduce the computational cost a lot with a good approximation of the matrix $\mathbf{Q}_{\text{sat}_k}$ at every iteration.

Therefore, the matrix \mathbf{Q}_{sat} , which remains a symmetric matrix, can be computed by exploiting the acceleration error vector at the first instant of time $\Delta\mathbf{a}_{\text{sat}_0}$, or the mean of the acceleration error vectors $\Delta\mathbf{a}_{\text{sat}_k}$, and considering the simplified discretization model of the Lecture Notes of professor Nahum Shimkin, *Estimation and Identification in Dynamical Systems*, Technion - Israel Institute of Technology, Department of Electrical Engineering, Haifa (IL), 2009 (*Shimkin, 2009*), which brings to the following assumption:

$$\mathbf{Q}_{\text{sat}} = \mathbf{w}_{\text{sat}_0} \mathbf{w}_{\text{sat}_0}^T, \quad (5.27)$$

that is coherent with the definition of \mathbf{Q}_k in case of pure random vectors (1st of eqs (4.40)), i.e.:

$$\mathbf{Q}_k = E[\mathbf{w}_k \mathbf{w}_k^T] \quad (5.28)$$

Hence, the process error matrix \mathbf{Q}_{sat} becomes (omitting the subscripts sat and DYD):

$$\mathbf{Q} = \begin{bmatrix} \frac{h^2}{4} (\Delta a_{x_0})^2 & \frac{h^2}{4} \Delta a_{x_0} \Delta a_{y_0} & \frac{h^2}{4} \Delta a_{x_0} \Delta a_{z_0} & \frac{h^3}{2} (\Delta a_{x_0})^2 & \frac{h^3}{2} \Delta a_{x_0} \Delta a_{y_0} & \frac{h^3}{2} \Delta a_{x_0} \Delta a_{z_0} \\ \frac{h^2}{4} \Delta a_{x_0} \Delta a_{y_0} & \frac{h^2}{4} (\Delta a_{y_0})^2 & \frac{h^2}{4} \Delta a_{y_0} \Delta a_{z_0} & \frac{h^3}{2} \Delta a_{x_0} \Delta a_{y_0} & \frac{h^3}{2} (\Delta a_{y_0})^2 & \frac{h^3}{2} \Delta a_{y_0} \Delta a_{z_0} \\ \frac{h^2}{4} \Delta a_{x_0} \Delta a_{z_0} & \frac{h^2}{4} \Delta a_{y_0} \Delta a_{z_0} & \frac{h^2}{4} (\Delta a_{z_0})^2 & \frac{h^3}{2} \Delta a_{x_0} \Delta a_{z_0} & \frac{h^3}{2} \Delta a_{y_0} \Delta a_{z_0} & \frac{h^3}{2} (\Delta a_{z_0})^2 \\ \frac{h^3}{2} (\Delta a_{x_0})^2 & \frac{h^3}{2} \Delta a_{x_0} \Delta a_{y_0} & \frac{h^3}{2} \Delta a_{x_0} \Delta a_{z_0} & h^2 (\Delta a_{x_0})^2 & h^2 \Delta a_{x_0} \Delta a_{y_0} & h^2 \Delta a_{x_0} \Delta a_{z_0} \\ \frac{h^3}{2} \Delta a_{x_0} \Delta a_{y_0} & \frac{h^3}{2} (\Delta a_{y_0})^2 & \frac{h^3}{2} \Delta a_{y_0} \Delta a_{z_0} & h^2 \Delta a_{x_0} \Delta a_{y_0} & h^2 (\Delta a_{y_0})^2 & h^2 \Delta a_{y_0} \Delta a_{z_0} \\ \frac{h^3}{2} \Delta a_{x_0} \Delta a_{z_0} & \frac{h^3}{2} \Delta a_{y_0} \Delta a_{z_0} & \frac{h^3}{2} (\Delta a_{z_0})^2 & h^2 \Delta a_{x_0} \Delta a_{z_0} & h^2 \Delta a_{y_0} \Delta a_{z_0} & h^2 (\Delta a_{z_0})^2 \end{bmatrix} \quad (5.29)$$

Even if the measurement error vector $\mathbf{v}_{\text{sat}_k}$ is slightly deterministic too, is preferable to maintain the measurement error matrix $\mathbf{R}_{\text{sat}_k}$ as diagonal one, which allows to obtain a good simulation for the DustCube's online process to take tracking measurements of the target bodies.

Without losing consistency, is useful to maintain the matrix $\mathbf{R}_{\text{sat}_k}$ constant during the computation and equal to a diagonal matrix \mathbf{R}_{sat} which has the principal diagonal composed by the squares of the measurement errors at the first instance of time, so that it remains coherent to the definition of the measurement error covariance matrix in case of pure random vectors (2nd of eqs (4.40)), i.e.:

Once defined the selected estimate of the initial state $\tilde{\mathbf{x}}_{\text{sat}0}$, the error matrix $\mathbf{P}_{\text{sat}0}$ can be approximated by simply considering it as diagonal and formed by the square of the components of the initial error vector \mathbf{e}_0 .

Several estimates of the initial state have been tested to verify the qualities of the Kalman filter.

Indeed, the selected initial state $\tilde{\mathbf{x}}_{\text{sat}0}$ with its computed error matrix $\mathbf{P}_{\text{sat}0}$, together with the process error matrix \mathbf{Q}_{sat} and the measurement error one \mathbf{R}_{sat} , influences the computation of the corrected state vectors $\tilde{\mathbf{x}}_{\text{sat}k}^+$ in terms of convergence speed and convergence behaviour.

The formulation of the DustCube's filter continues with the implementation of the iterative procedure, which is accomplished by the MATLAB script *EKF_thesis.m*, where a FOR CYCLE has been exploited to continuously compute the prediction and the correction of the state vector, i.e. $\tilde{\mathbf{x}}_{\text{sat}k}^-$ and $\tilde{\mathbf{x}}_{\text{sat}k}^+$.

The initialization of every iteration requires the computation of the a-priori estimation of the state $\tilde{\mathbf{x}}_{\text{sat}k}^-$ and of the a-priori error covariance matrix $\mathbf{P}_{\text{sat}k}^-$, which are defined as:

$$\begin{aligned}\tilde{\mathbf{x}}_{\text{sat}k+1}^- &= \mathbf{f}_{\text{sat}k}(\tilde{\mathbf{x}}_{\text{sat}k}^+) \\ \mathbf{P}_{\text{sat}k+1}^- &= \mathbf{F}_{\text{sat}k} \mathbf{P}_{\text{sat}k}^+ \mathbf{F}_{\text{sat}k}^T + \mathbf{Q}_{\text{sat}},\end{aligned}\quad (5.32)$$

where $\mathbf{F}_{\text{sat}k}$ is the Jacobian matrix of the vector $\mathbf{f}_{\text{sat}k}(\tilde{\mathbf{x}}_{\text{sat}k}^+)$ and is computed at every iteration as it has been explained in the equation (5.6), while the computation of the evolution $\mathbf{f}_{\text{sat}k}(\tilde{\mathbf{x}}_{\text{sat}k}^+)$ of the state vector is accomplished through the eqs (5.2), (5.3) and (5.4).

As said before, $\mathbf{f}_{\text{sat}k}(\tilde{\mathbf{x}}_{\text{sat}k}^+)$ and $\mathbf{F}_{\text{sat}k}$ are computed by exploiting the specially created MATLAB script *statetransition_4bodies_SRP.m*.

Therefore, the script proceeds with the classical iteration of the EKF, like has been explained in section 4.5 [Extended Kalman Filter \(EKF\)](#) and is summarised here below for completeness:

$$\begin{aligned}\mathbf{H}_{\text{sat}k} &= \left. \frac{\partial \mathbf{h}_{\text{sat}k}(\mathbf{x}_{\text{sat}k})}{\partial \mathbf{x}_{\text{sat}k}} \right|_{(\mathbf{x}_{\text{sat}k} = \tilde{\mathbf{x}}_{\text{sat}k}^-)} \\ \mathbf{K}_{\text{sat}k} &= \mathbf{P}_{\text{sat}k}^- \mathbf{H}_{\text{sat}k}^T (\mathbf{H}_{\text{sat}k} \mathbf{P}_{\text{sat}k}^- \mathbf{H}_{\text{sat}k}^T + \mathbf{R}_{\text{sat}})^{-1} \\ \tilde{\mathbf{x}}_{\text{sat}k}^+ &= \tilde{\mathbf{x}}_{\text{sat}k}^- + \mathbf{K}_{\text{sat}k} [\mathbf{z}_{\text{sat}k} - \mathbf{h}_{\text{sat}k}(\tilde{\mathbf{x}}_{\text{sat}k}^-)] \\ \mathbf{P}_{\text{sat}k}^+ &= (\mathbf{I} - \mathbf{K}_{\text{sat}k} \mathbf{H}_{\text{sat}k}) \mathbf{P}_{\text{sat}k}^- \\ \mathbf{F}_{\text{sat}k} &= \left. \frac{\partial \mathbf{f}_{\text{sat}k}(\mathbf{x}_{\text{sat}k})}{\partial \mathbf{x}_{\text{sat}k}} \right|_{(\mathbf{x}_{\text{sat}k} = \tilde{\mathbf{x}}_{\text{sat}k}^+)},\end{aligned}\quad (5.33)$$

where $\mathbf{K}_{\text{sat}k}$ represents the Kalman gain matrix at the k^{th} iteration and $\mathbf{P}_{\text{sat}k}^+$ the a-posteriori error covariance matrix at the k^{th} iteration.

The measurement function $\mathbf{h}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})$ has been obtained as shown in the equation (5.12), while the general form of its Jacobian matrix $\mathbf{H}_{\text{sat}_k}$ has been achieved by computing the single partial derivatives shown in equation (4.57), which has been implemented in the MATLAB script *jacobian_H_thesis.m*, as said before.

The results obtained by the DustCube's filter will be discussed in the next sections.

5.3 INITIAL VALIDATION OF THE DUSTCUBE'S FILTER

After the implementation of the Extended Kalman Filter of DustCube, is necessary to tune the filter, which must converge to the actual state vectors of DustCube for a wide range of initial states $\tilde{\mathbf{x}}_{\text{sat}_0}$. Therefore, the adopted solution to preliminarily validate the filter implemented was to test a perfect matching between the "true" dynamic used to generate the measurements, and the filter implemented dynamic. In other words, the true state was computed through a simplified orbit propagator that exploits the Explicit Euler method and takes into account the same force sources acting on DustCube as the filter's state function $\mathbf{f}_{\text{sat}_k}(\mathbf{x}_{\text{sat}_k})$, namely the central attractions of the primaries, the Solar Radiation Pressure and the solar gravitational disturbance.

This means the dynamic of the orbital propagator has been equalized to the filter's dynamic.

This method allows to compute a process error matrix \mathbf{Q}_{sat} that is almost a null 6×6 matrix.

Also, the estimate of the initial state vector $\tilde{\mathbf{x}}_{\text{sat}_0}$ has been varied many times to test the convergence of the filter, which finally proved to be very satisfactory.

For example, exploiting the specially created MATLAB script *tuning.m*, we can consider a time window spanning from August 24th, 2022 to August 31st, 2022, a time-step $h = 10$ seconds and an initial estimate of state vector $\tilde{\mathbf{x}}_{\text{sat}_0}|_{\text{DYD}}$ defined as:

$$\tilde{\mathbf{x}}_{\text{sat}_0}|_{\text{DYD}} = \mathbf{x}_{\text{sat}_0}|_{\text{DYD}} + \mathbf{e}_0|_{\text{DYD}} = \mathbf{x}_{\text{sat}_0}|_{\text{DYD}} + \begin{bmatrix} -22.978 \text{ m} \\ 9.592 \text{ m} \\ -1.9069 \text{ m} \\ 0.0015 \text{ m/s} \\ -0.0029 \text{ m/s} \\ -0.0002 \text{ m/s} \end{bmatrix}_{\text{DYD}}, \quad (5.34)$$

so that, the considered scenario represents an extreme situation, since every element of the error vector $\mathbf{e}_0|_{\text{DYD}}$ is between 2% and 10% of the reciprocal component of the exact state $\mathbf{x}_{\text{sat}_0}|_{\text{DYD}}$.

The error vector \mathbf{e}_0 of the previous equation shows non-uniform errors in the components of the position and in the one of the velocity. Indeed, since we are interested to set up a more realistic scenario, this error has been selected by considering the orbital plane of DustCube and by taking into account the position of the *Lagrange point* L_5 w.r.t. the Didymos' synodic plane.

Therefore, remembering the definition of the a-posteriori estimation error \mathbf{e}_k^+ , i.e. the 2nd equation of (4.42), and filtering the state of DustCube of this awful set-up, is possible to really appreciate the convergence of the corrected solution obtained by the implemented EKF through the visualization of the behaviour of every component of the a-posteriori estimation error, namely:

- the Figure 5.7, the Figure 5.8 and the Figure 5.9 show the convergence of the corrected position of the S/C as divided in its components, starting from the exaggerated initial error $\mathbf{e}_0|_{D\bar{Y}D}$. Observing every plot, it is possible to appreciate the good performance of the estimation process: the absolute value of the initial error in x coordinate evolves from 23 meters to just 2.7 centimetres, the absolute value of the initial error in y coordinate from 9 meters to just 2.8 centimetres, the absolute value of the initial error in z coordinate from 1.9 meters to 7.9 centimeters;
- the Figure 5.10, the Figure 5.11 and the Figure 5.12 show the convergence of the corrected velocity of the S/C as divided in its components, starting from the exaggerated initial error $\mathbf{e}_0|_{D\bar{Y}D}$. From these plots, it is possible to evaluate the good performance of the estimation process: the absolute value of the initial error in \dot{x} decreases from 1.5×10^{-3} [m/s] to just 5.3×10^{-6} [m/s], the absolute value of the initial error in \dot{y} from 2.9×10^{-3} [m/s] to just 4.1×10^{-6} [m/s], the absolute value of the initial error in \dot{z} from 1×10^{-4} [m/s] to just 1.3×10^{-6} [m/s].

Obviously, the same tuning operation could be performed by exploiting other initial errors $\mathbf{e}_0|_{D\bar{Y}D}$, which would show similar convergences.

The results obtained through the tuning scenario allow to validate the DustCube's filter, which is now ready to filter actual situations in which we will consider the process error matrix \mathbf{Q}_{sat} too.

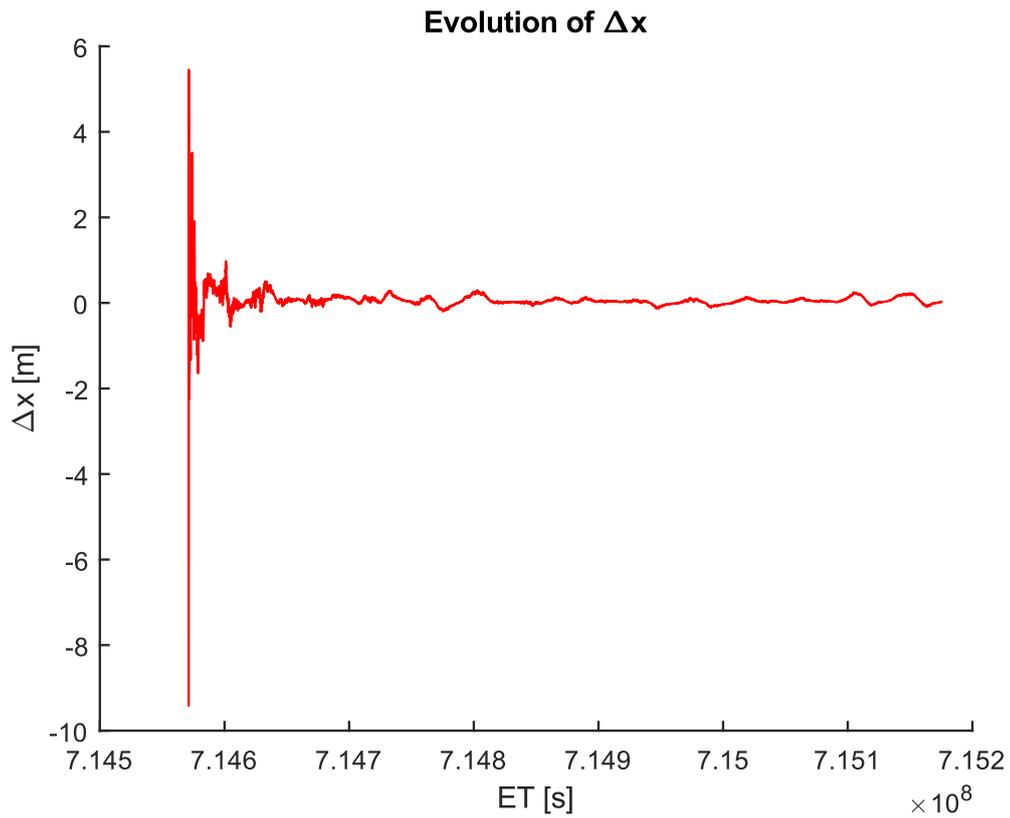


Figure 5.7: Estimated state's error on x

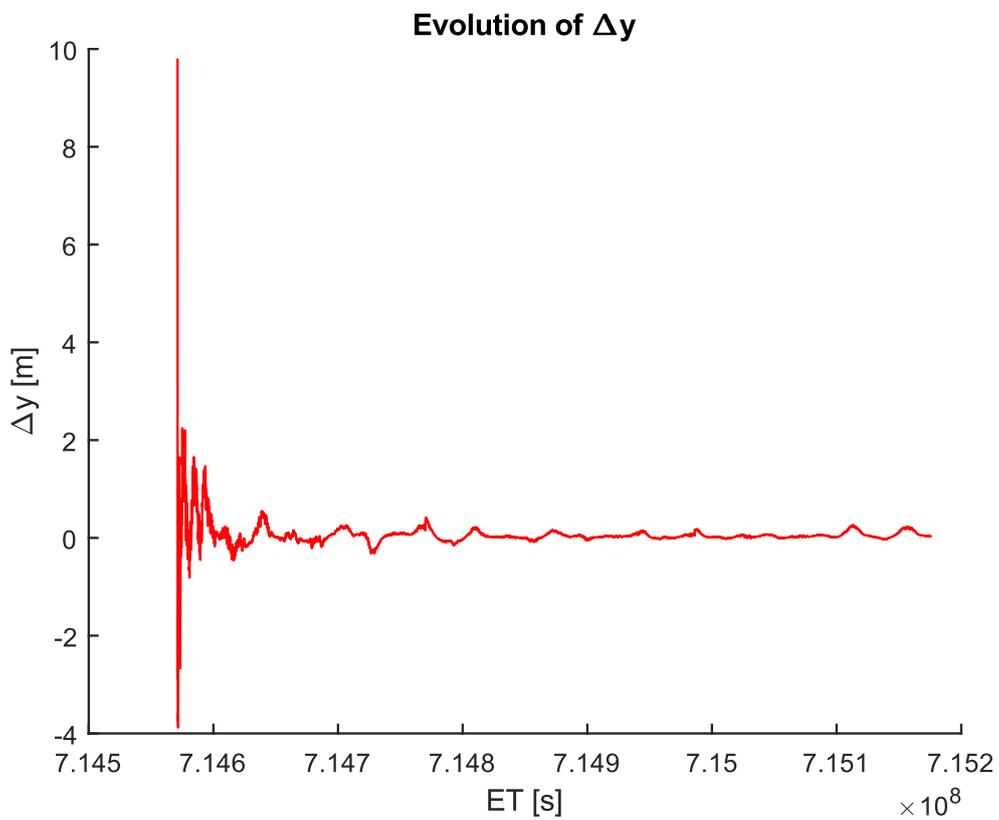


Figure 5.8: Estimated state's error on y

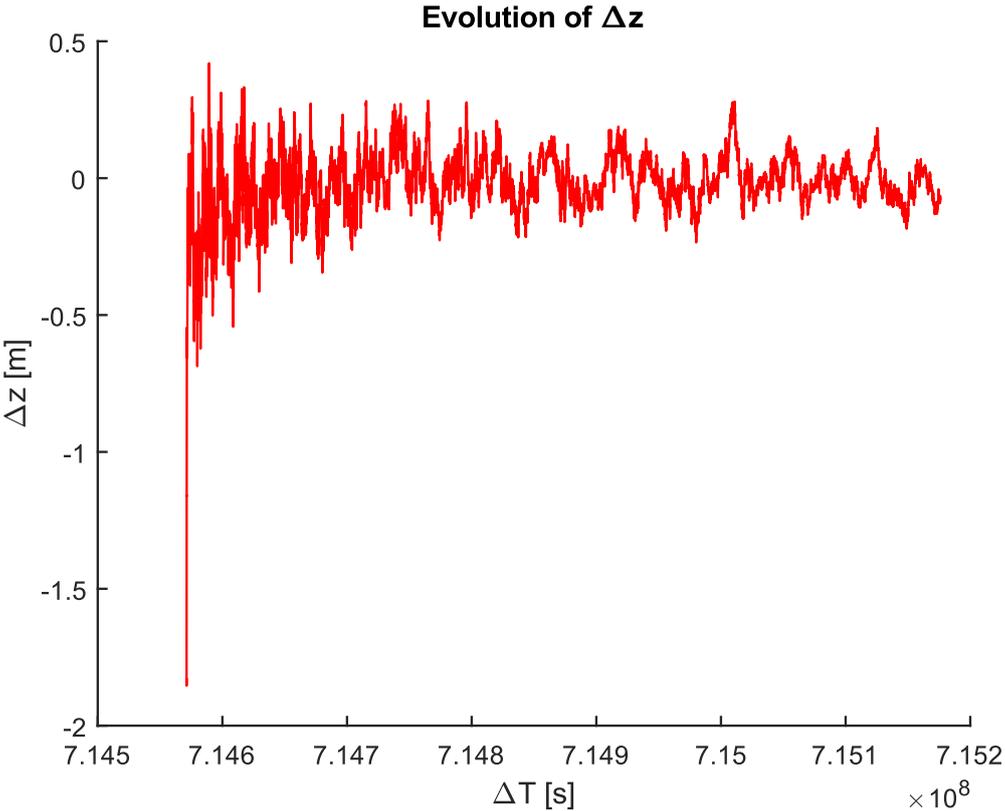


Figure 5.9: Estimated state's error on z

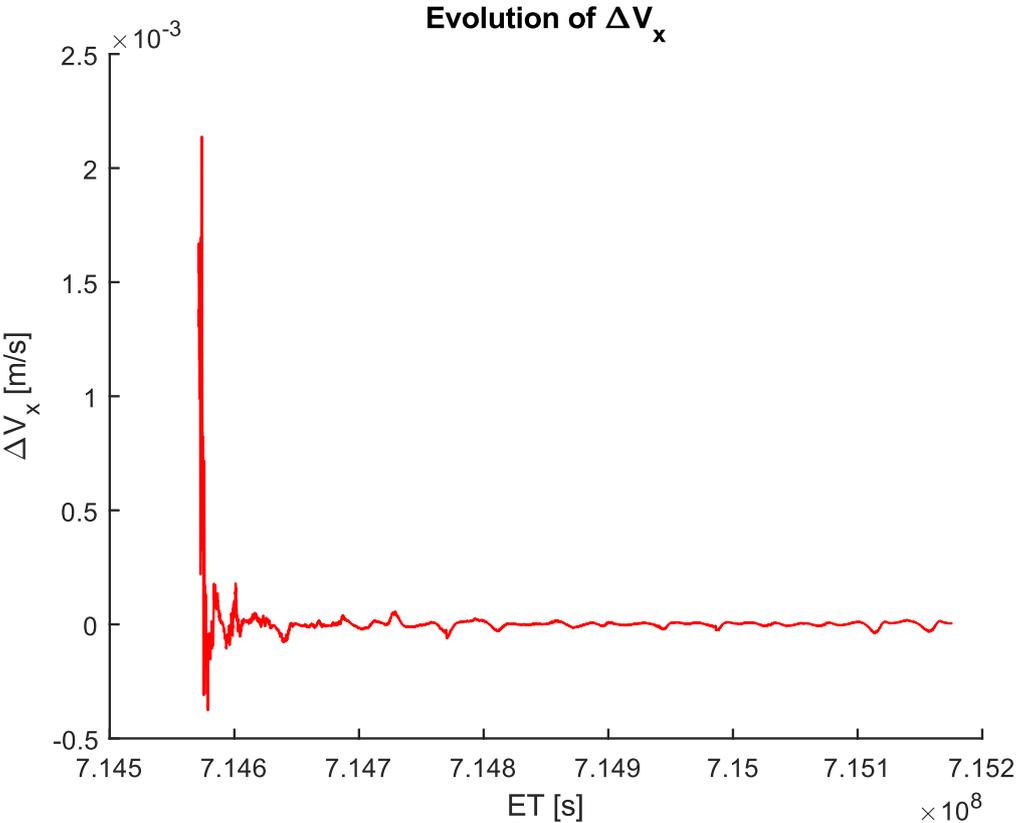


Figure 5.10: Estimated state's error on Vx

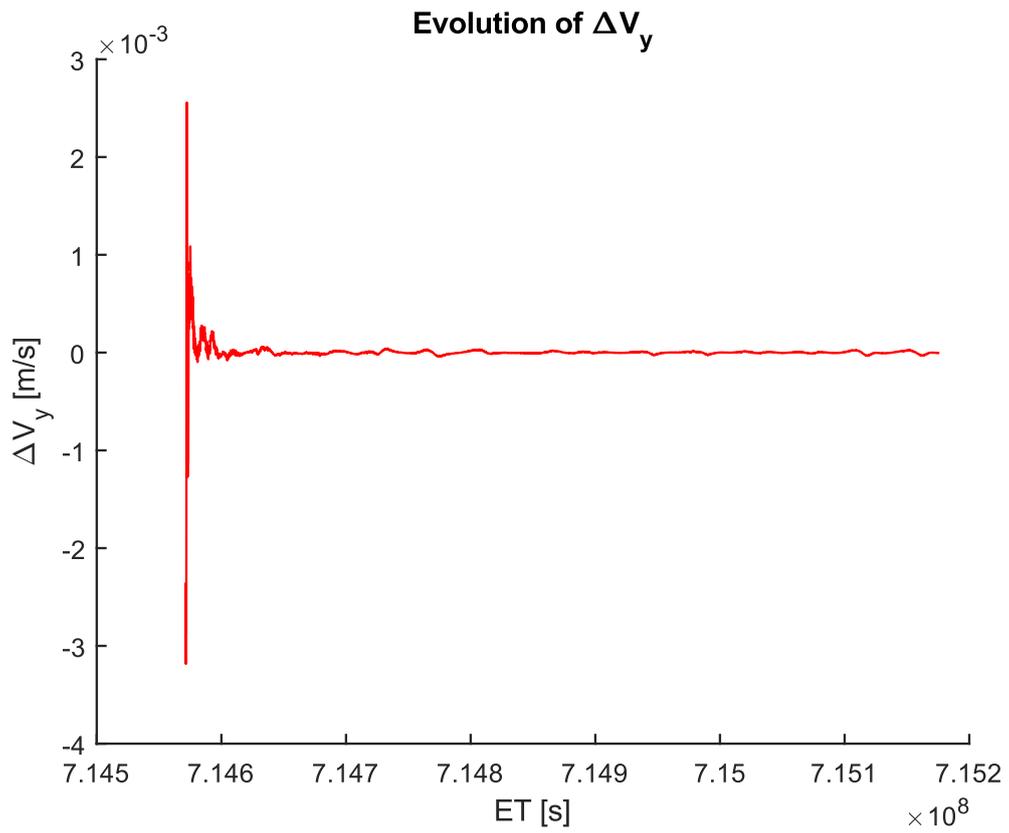


Figure 5.11: Estimated state's error on V_y

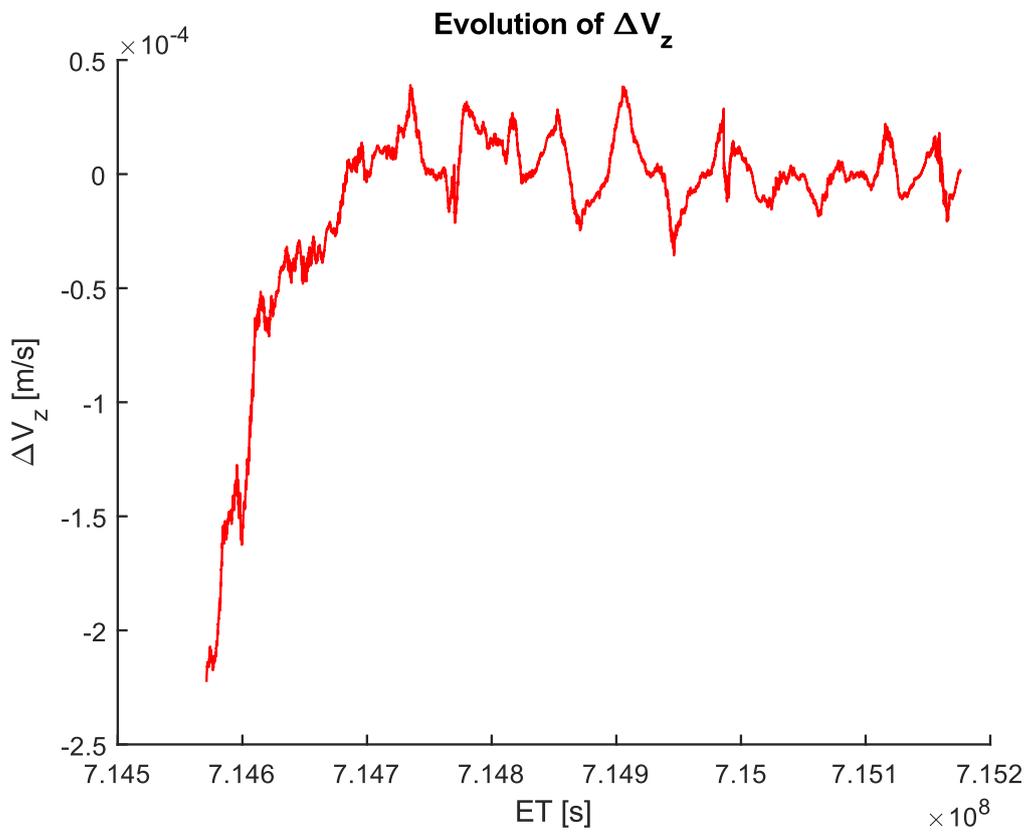


Figure 5.12: Estimated state's error on V_z

5.4 THE FILTERING OF THE CORRUPTED STATE OF DUSTCUBE

The filtering of the corrupted state of DustCube is obtained by initializing the EKF with an estimated initial state vector $\tilde{\mathbf{x}}_{\text{sat}0}$.

The measurements are taken, as said before, from the actual states of DustCube.

Also, a process error \mathbf{w}_k is considered to simulate possible noises acting on the system, which involves the existence of the process error matrix \mathbf{Q}_{sat} .

To compute the matrix \mathbf{Q}_{sat} , we suppose to consider the acceleration due to Didymain and Didymoon which includes their exterior spherical harmonics as the processed one and we consider the eq. (5.29). Therefore, since we are interest to represent the scenario as realistic as possible, we take into account the normalized exterior spherical harmonic coefficients up to degree 20 and order 20 (Zannoni, et al., 2018), which have been explained in section 1.2.3 Gravitational attractions acted by Didymain and Didymoon of Chapter 1, and we consider a deviation angle of 1% for the LOS measurements.

As said before, several estimations of initial state vector have been tested to properly evaluate the dynamic behaviour of the filtering process.

For example, considering a time window spanning from August 24th, 2022 to August 28th, 2022, a time-step $h = 10$ s and starting from the following estimate of the initial state vector $\tilde{\mathbf{x}}_{\text{sat}0}|_{\text{DYD}}$:

$$\tilde{\mathbf{x}}_{\text{sat}0}|_{\text{DYD}} = \mathbf{x}_{\text{sat}0}|_{\text{DYD}} + \mathbf{e}_0|_{\text{DYD}} = \mathbf{x}_{\text{sat}0}|_{\text{DYD}} + \begin{bmatrix} 5 \text{ m} \\ 15 \text{ m} \\ 1 \text{ m} \\ 0.001 \text{ m/s} \\ 0.001 \text{ m/s} \\ 0.001 \text{ m/s} \end{bmatrix}_{\text{DYD}}, \quad (5.35)$$

the filter can efficiently estimate the state of DustCube.

The error vector \mathbf{e}_0 of the previous equation shows, as in the previous section, non-uniform errors in the components of the position. Again, since we are interested to set up a more realistic scenario, this error has been selected by considering the orbital plane of DustCube, which is, in this case, always coincident with the orbital plane of the mutual orbit of the primaries and by taking into account the position of the *Lagrange point* L_5 w.r.t. the Didymos' synodic plane.

The figures from Figure 5.13 to Figure 5.18 show that every component of the corrected state vector $\tilde{\mathbf{x}}_{\text{sat}k}^+|_{\text{DYD}}$ converges to the actual state of DustCube, since the a-posteriori error's components decreases from the initial error.

The Figure 5.19 shows the convergence of the DustCube's estimated orbit to the exact solution, as can be easier to visualize in the zoomed section of Figure 5.20.

Chapter 5

In general, we can observe that the errors in the in-plane components (x-y) are greater than the errors in the out-of-plane component (z). Furthermore, the errors show a clear deterministic periodic pattern, which is a direct consequence of the reduced dynamic modelling implemented in the filter, that can be only partially compensated by the process noise matrix.

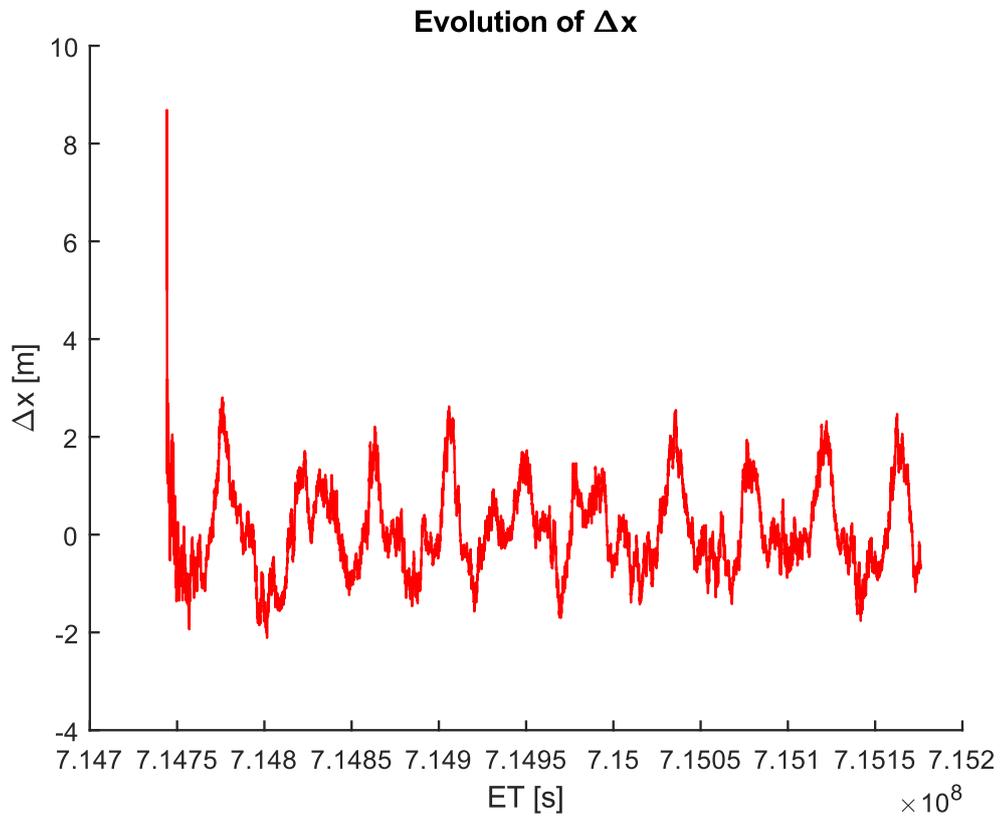


Figure 5.13: Final estimated state's error on x

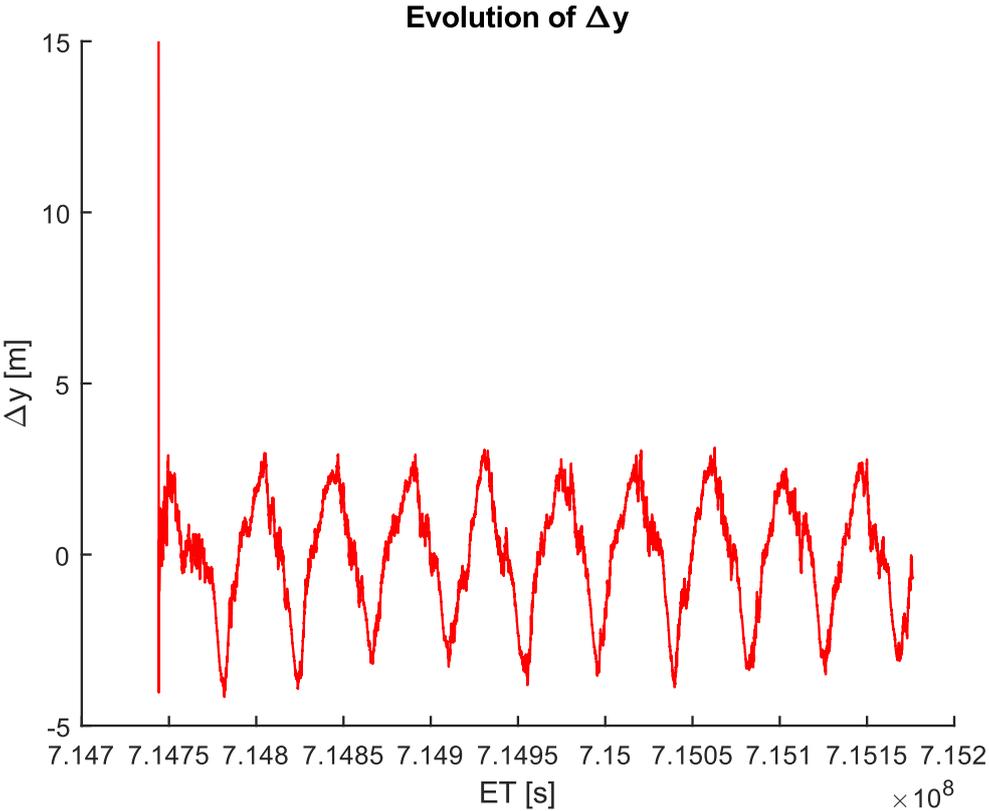


Figure 5.14: Final estimated state's error on y

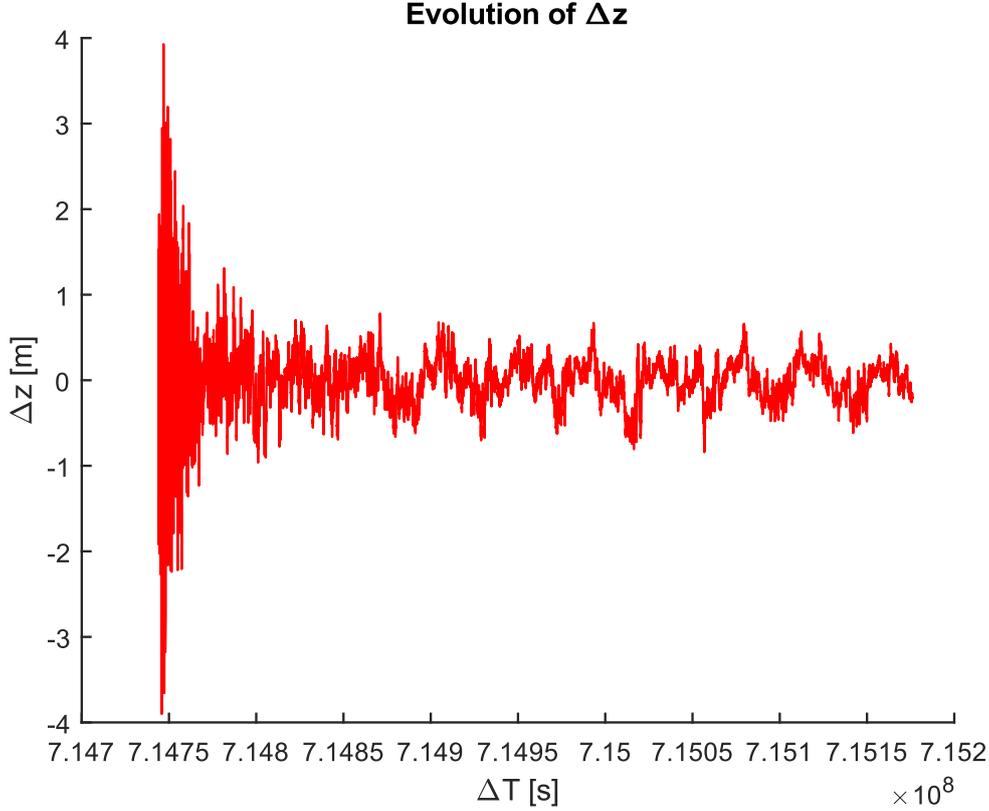


Figure 5.15: Final estimated state's error on z

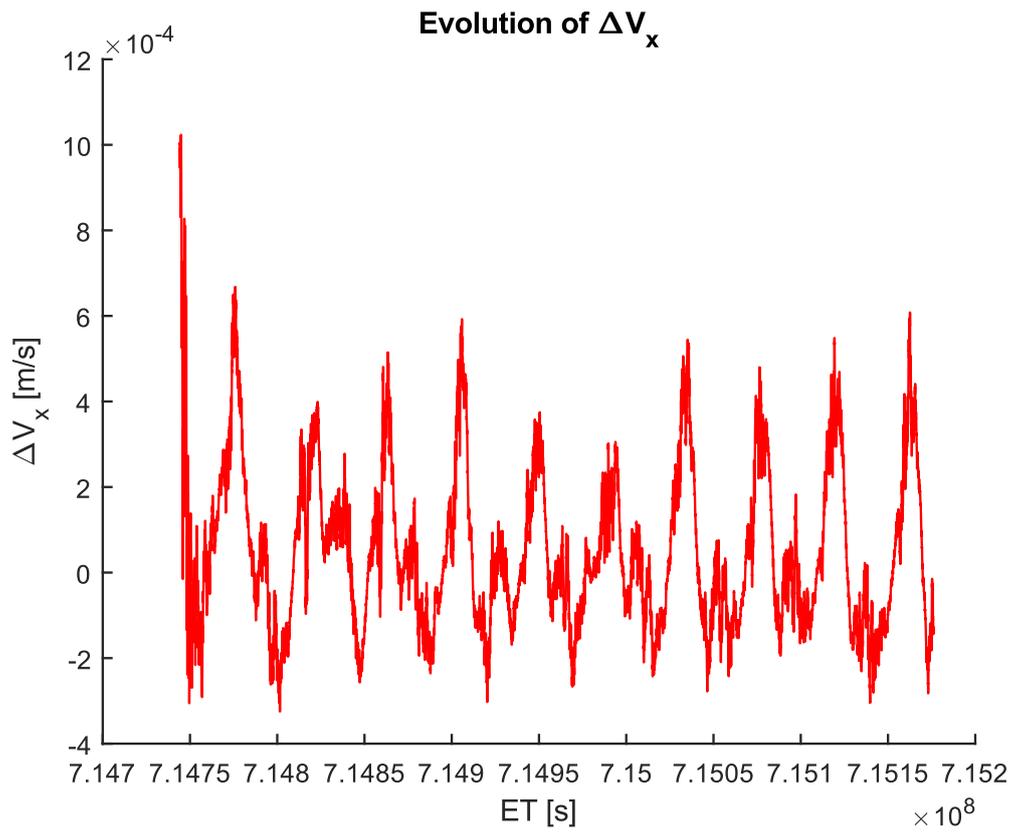


Figure 5.16: Final estimated state's error on Vx

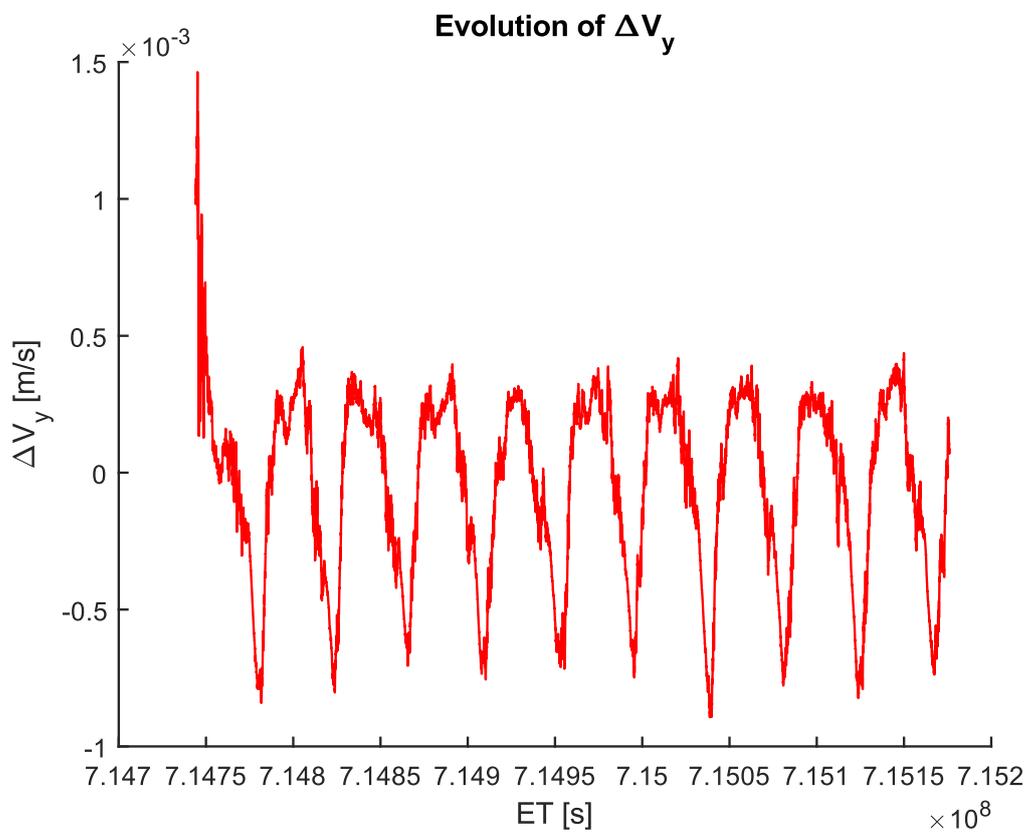


Figure 5.17: Final estimated state's error on Vy

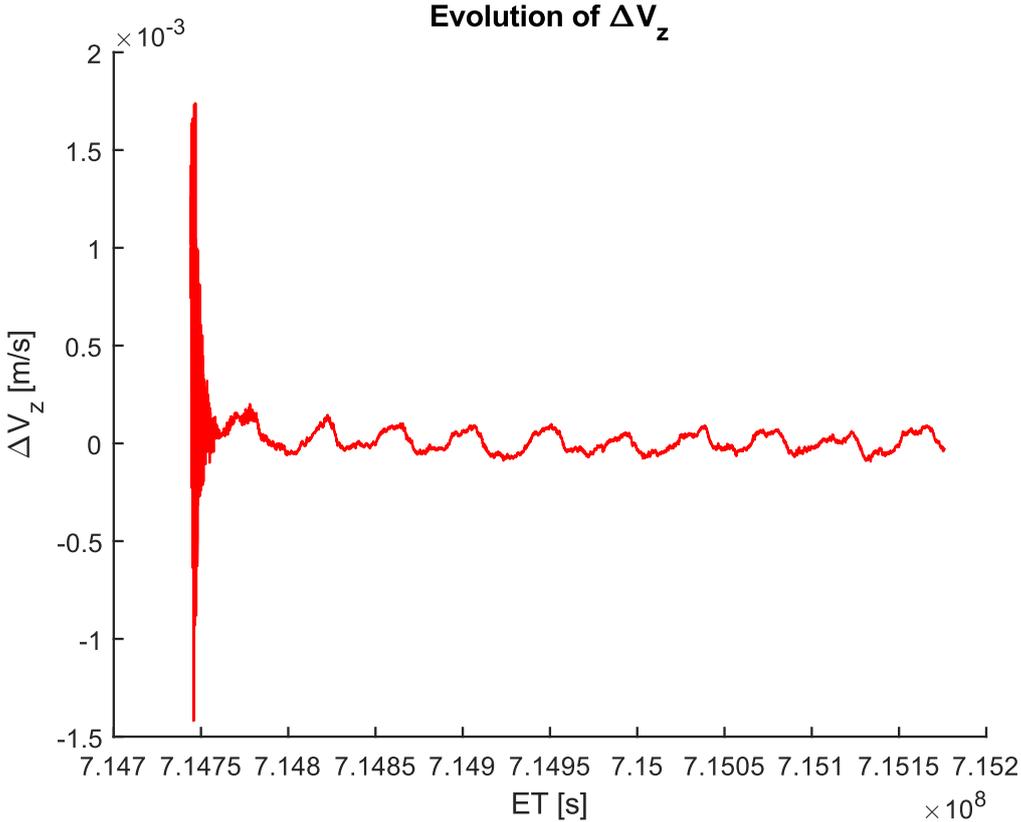


Figure 5.18: Final estimated state's error on Vz

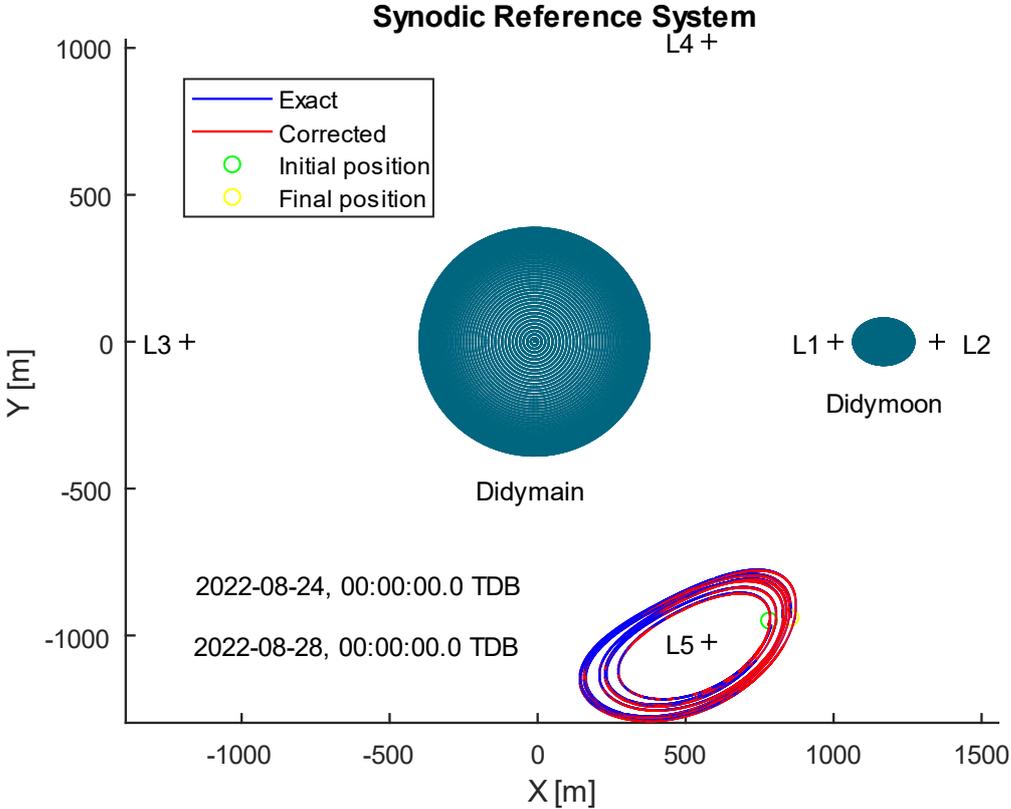


Figure 5.19: Exact solution vs. filtered one in synodic coordinates

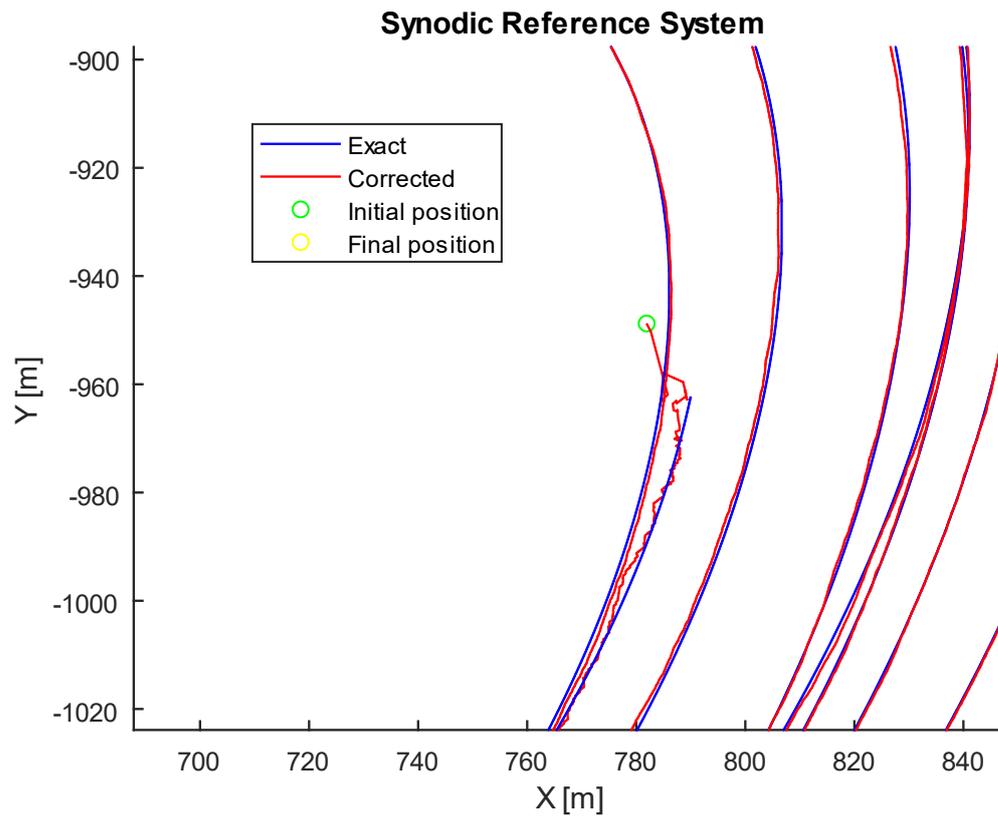


Figure 5.20: Exact solution vs. filtered one in synodic coordinates (zoom)

CONCLUSIONS

The thesis discussed in the previous chapters summarizes all the work done by the candidate, namely:

1. the implementation of a discrete orbital propagator in MATLAB environment, which is able to compute the state of the artificial satellite DustCube, which should operate in the gravitational field of the Didymos asteroid, a binary system composed by two massive bodies, called Didymain and Didymoon, starting from a known initial state and taking into account the main disturbances that would influence the total acceleration of DustCube;
2. the development, always in MATLAB environment, of a discrete optical navigation filter simulator, which can estimate the DustCube's state starting from some direction and range measurements w.r.t. the two celestial bodies that form the asteroid Didymos, obtained through two infrared cameras that constitute a preponderant part of the satellite payload.

The main part of the thesis concerns the phase 2, which necessarily depends on the phase 1, since the discretized state and the measurements obtained in real-time are subject to Gaussian noises with zero mean that heavily influence the dynamics of the filter, which requires a precise knowledge of the evolution of the state of DustCube.

The orbital propagator developed through the use of the Runge Kutta 4th (RK4) order method proved to be able to compute the satellite's state with an excellent numerical stability, which allows to vary the integration time-step with a wide range, reducing the computational cost and maintaining the accuracy of the final solution at the same time, and an excellent numerical precision, demonstrated by comparison with the solutions obtained through the Monte Python software.

These features of the developed orbital propagator allow to consider it as a valid MATLAB programming base for other space missions engaged on small celestial bodies.

Since the state dynamics of DustCube are governed by non-linear equations and the optical measurements are non-linear too, the navigation filter selected for DustCube mission is the Extended Kalman Filter (EKF) one, implemented in MATLAB programming too.

The navigation filter has been tested by exploiting range and direction measurements in terms of norms, colatitudes and longitudes of the position vectors directed from DustCube towards the primaries, which represents a realistic concept for actual optical measurements in space.

The results obtained by the Kalman filtering proved to be stable and coherent, widely satisfying the minimal requirements for DustCube mission with accuracies in the estimation of the spacecraft position in the order of $1 \div 2$ meters if we consider a wide error in the first estimated position, which

Conclusions

correspond to an error of about $0.1\% \div 0.2\%$ of the nominal position of DustCube, and $0.5 \div 0.1$ meters if we consider an acceptable error in the first estimated position, which correspond to an error of about $0.05\% \div 0.1\%$ of the nominal position of the S/C.

Possible future developments could concern a sensitivity analysis of the filter performance as a function of different magnitudes of measurement noise and level of detail of the implemented filter dynamic and a Monte Carlo analysis which for providing more statistically meaningful measures of the expected filter accuracy.

APPENDIX

A.1 EQUATIONS OF MOTIONS

The state $\mathbf{s}(t)$ of a body in space is a time-dependent vector of six components that comprises the position vector $\mathbf{r}(t)$ and the velocity vector $\mathbf{v}(t)$ of its CoM:

$$\mathbf{s}(t) = \begin{bmatrix} \mathbf{r}(t) \\ \mathbf{v}(t) \end{bmatrix} \quad (\text{A.1})$$

Obviously, the position and the velocity vectors depend on the Reference Frame, which can be inertial or non-inertial, centered in a specific point in space, that is usually the CoM of the considered system. Since the state of a body is referred to its CoM, we can consider the S/C as a point-mass and evaluate its trajectory without taking into account its shape.

A.1.1 INERTIAL REFERENCE FRAME

Considering an Inertial RF (I subscript) with axes (X_I, Y_I, Z_I) , centered in the CoM of the system, and a point-mass m in space, the expression of the position vector $\mathbf{r}(t)$ is possible by making use of the canonical basis $(\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}})$, which is composed by the unit vectors $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$ that are fixed in space, so constant in time (Figure A.1):

$$\mathbf{r}(t)|_I \stackrel{\text{def}}{=} \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = x_I \hat{\mathbf{i}} + y_I \hat{\mathbf{j}} + z_I \hat{\mathbf{k}} \quad (\text{A.2})$$

Differentiating the position vector $\mathbf{r}(t)|_I$ by the time we obtain the so-called absolute (or inertial) velocity, which is the velocity the point-mass holds w.r.t. the CoM of the Inertial RF, that can be considered as fixed in space and in time, while differentiating $\mathbf{r}(t)|_I$ twice we compute the so-called absolute (or inertial) acceleration, that is the acceleration the point-mass holds w.r.t. the CoM of the Inertial RF:

$$\mathbf{v}_A(t) \stackrel{\text{def}}{=} \left. \frac{d\mathbf{r}(t)|_I}{dt} \right|_I = \begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{z}_I \end{bmatrix} = \dot{x}_I \hat{\mathbf{i}} + \dot{y}_I \hat{\mathbf{j}} + \dot{z}_I \hat{\mathbf{k}} \quad (\text{A.3})$$

$$\mathbf{a}_A(t) \stackrel{\text{def}}{=} \left. \frac{d^2\mathbf{r}(t)|_I}{dt^2} \right|_I = \left. \frac{d\mathbf{v}_A(t)|_I}{dt} \right|_I = \begin{bmatrix} \ddot{x}_I \\ \ddot{y}_I \\ \ddot{z}_I \end{bmatrix} = \ddot{x}_I \hat{\mathbf{i}} + \ddot{y}_I \hat{\mathbf{j}} + \ddot{z}_I \hat{\mathbf{k}} \quad (\text{A.4})$$

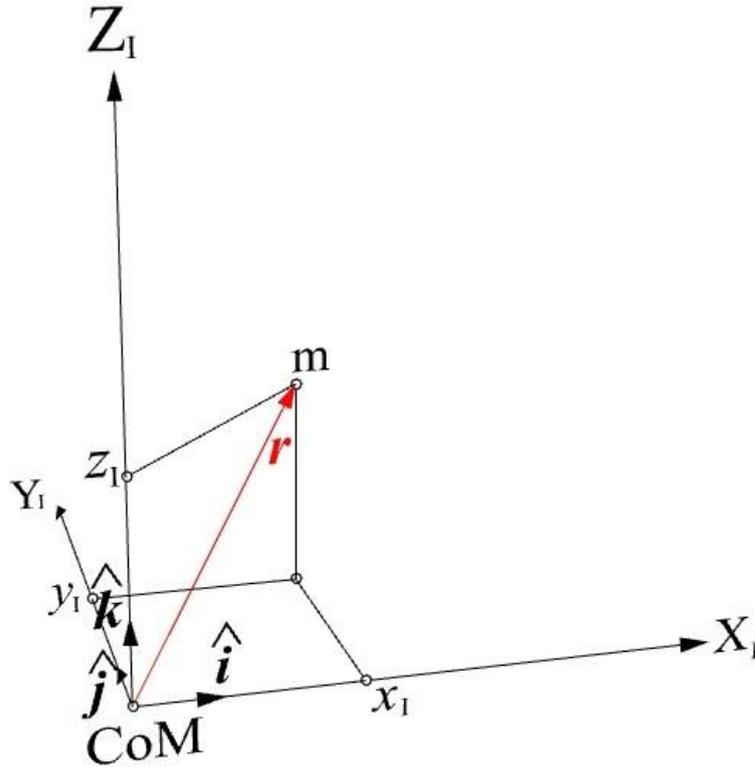


Figure A.1: The position vector \mathbf{r} in an Inertial Reference Frame

Therefore, the knowledge of the evolution in time of the inertial vectors $\mathbf{r}(t)|_I$, $\mathbf{v}_A(t)$ and $\mathbf{a}_A(t)$ allows to reconstruct the trajectory of the point-mass w.r.t. the Inertial RF.

A.1.2 ROTATING REFERENCE FRAME

Considering a Non-inertial RF (N subscript) with axes (X_N, Y_N, Z_N) , centered in the same CoM of the previous Inertial RF and rotating w.r.t. it with angular velocity $\boldsymbol{\omega}(t)$, we can express the position vector $\mathbf{r}(t)$ by exploiting the canonical basis $(\hat{\xi}, \hat{\lambda}, \hat{\zeta}) = (\hat{\xi}(t), \hat{\lambda}(t), \hat{\zeta}(t))$, which is composed by the unit vectors $\hat{\xi}$, $\hat{\lambda}$ and $\hat{\zeta}$ that are not constant in time w.r.t. the Inertial RF (Figure A.2), since their direction varies continuously.

Omitting the time dependence of $\hat{\xi}$, $\hat{\lambda}$ and $\hat{\zeta}$ for convenience, the position vector $\mathbf{r}(t)|_N$ can be defined as:

$$\mathbf{r}(t)|_N = \begin{bmatrix} x_N \\ y_N \\ z_N \end{bmatrix} = x_N \hat{\xi} + y_N \hat{\lambda} + z_N \hat{\zeta} \quad (\text{A.5})$$

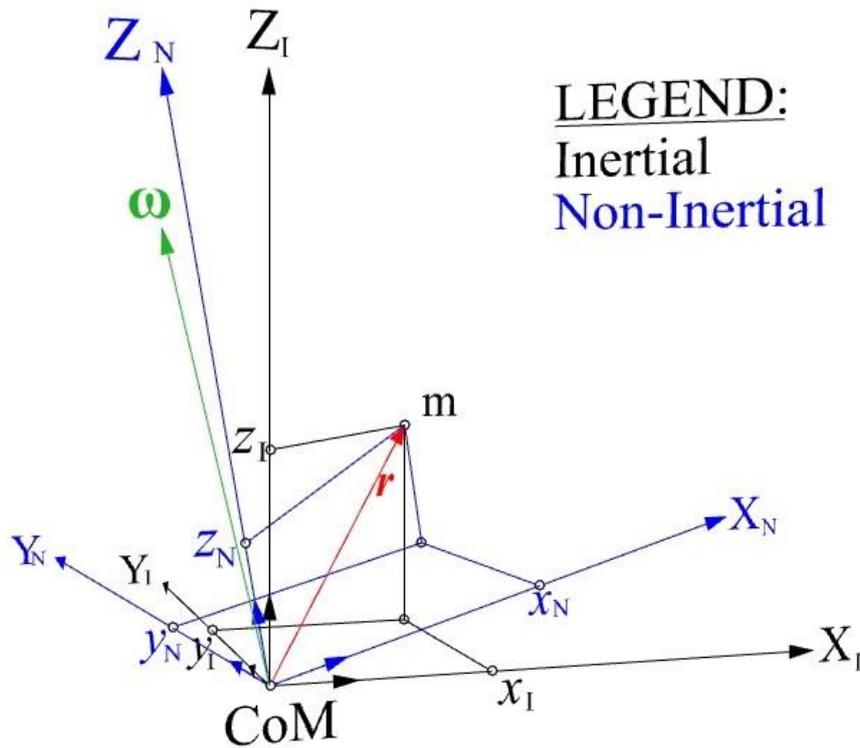


Figure A.2: The position vector \mathbf{r} w.r.t. an Inertial RF and a rotating RF

Also, the non-inertial triad $(\hat{\xi}, \hat{\lambda}, \hat{\zeta})$ rotates w.r.t. to the inertial one $(\hat{i}, \hat{j}, \hat{k})$ by the angular velocity $\boldsymbol{\omega}(t)$, which can be expressed in inertial coordinates or in non-inertial ones like the position vector:

$$\boldsymbol{\omega}(t)|_I = \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \Omega_x \hat{i} + \Omega_y \hat{j} + \Omega_z \hat{k} \quad (\text{A.6})$$

$$\boldsymbol{\omega}(t)|_N = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \omega_x \hat{\xi} + \omega_y \hat{\lambda} + \omega_z \hat{\zeta} \quad (\text{A.7})$$

Differentiating the position vector $\mathbf{r}(t)|_N$ by the time in non-inertial coordinates, for which the unit vectors $\hat{\xi}, \hat{\lambda}, \hat{\zeta}$ are constant in time, we obtain the so-called relative velocity, which is the velocity the point-mass holds w.r.t. the CoM of the Non-inertial RF, while differentiating $\mathbf{r}(t)|_N$ twice in non-inertial coordinates we compute the so-called relative acceleration, that is the acceleration the point-mass holds w.r.t. the CoM of the Non-inertial RF:

$$\mathbf{v}_R(t) = \left. \frac{d\mathbf{r}(t)|_N}{dt} \right|_N = \begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix} = \dot{x}_N \hat{\xi} + \dot{y}_N \hat{\lambda} + \dot{z}_N \hat{\zeta} \quad (\text{A.8})$$

$$\mathbf{a}_R(t) = \left. \frac{d^2\mathbf{r}(t)|_N}{dt^2} \right|_N = \begin{bmatrix} \ddot{x}_N \\ \ddot{y}_N \\ \ddot{z}_N \end{bmatrix} = \ddot{x}_N \hat{\xi} + \ddot{y}_N \hat{\lambda} + \ddot{z}_N \hat{\zeta} \quad (\text{A.9})$$

Appendix

The knowledge of the evolution in time of the non-inertial vectors $\mathbf{r}(t)|_N$, $\mathbf{v}_R(t)$ and $\mathbf{a}_R(t)$ allows to reconstruct the trajectory of the point-mass w.r.t. the Non-inertial RF.

Since the objective is to link the state of the S/C between reference frames, we can obtain the absolute velocity $\mathbf{v}_A(t)$ by differentiating $\mathbf{r}(t)|_N$ w.r.t. the Inertial RF, for which the unit vectors $\hat{\xi}, \hat{\lambda}, \hat{\zeta}$ are time-dependent:

$$\mathbf{v}_A(t) = \left. \frac{d\mathbf{r}(t)|_N}{dt} \right|_I = \dot{x}_N \hat{\xi}|_I + \dot{y}_N \hat{\lambda}|_I + \dot{z}_N \hat{\zeta}|_I + x_N \left. \frac{d\hat{\xi}}{dt} \right|_I + y_N \left. \frac{d\hat{\lambda}}{dt} \right|_I + z_N \left. \frac{d\hat{\zeta}}{dt} \right|_I \quad (\text{A.10})$$

Now, omitting the RF's subscripts, the derivatives of the unit vectors can be computed by exploiting the well-known Poisson's relation

$$\frac{d\hat{\xi}}{dt} = \boldsymbol{\omega}(t) \times \hat{\xi} \quad \frac{d\hat{\lambda}}{dt} = \boldsymbol{\omega}(t) \times \hat{\lambda} \quad \frac{d\hat{\zeta}}{dt} = \boldsymbol{\omega}(t) \times \hat{\zeta}, \quad (\text{A.11})$$

so that, the absolute velocity can be expressed as:

$$\mathbf{v}_A(t) = \mathbf{v}_R(t) + x_N \boldsymbol{\omega}(t) \times \hat{\xi} + y_N \boldsymbol{\omega}(t) \times \hat{\lambda} + z_N \boldsymbol{\omega}(t) \times \hat{\zeta} = \mathbf{v}_R(t) + \boldsymbol{\omega}(t) \times (x_N \hat{\xi} + y_N \hat{\lambda} + z_N \hat{\zeta}) \quad (\text{A.12})$$

Finally, we have obtained the absolute velocity $\mathbf{v}_A(t)$ as a sum of two contributions:

$$\mathbf{v}_A(t) = \mathbf{v}_R(t) + \boldsymbol{\omega}(t) \times \mathbf{r}(t) = \mathbf{v}_R(t) + \mathbf{v}_T(t), \quad (\text{A.13})$$

where $\mathbf{v}_T(t) \stackrel{\text{def}}{=} \boldsymbol{\omega}(t) \times \mathbf{r}(t)$ is the tangential velocity.

Differentiating twice the position vector $\mathbf{r}(t)|_N$ w.r.t the Inertial RF, we obtain the absolute acceleration $\mathbf{a}_A(t)$:

$$\begin{aligned} \mathbf{a}_A(t) &= \left. \frac{d^2\mathbf{r}(t)|_N}{dt^2} \right|_I = & (\text{A.14}) \\ &= \ddot{x}_N \hat{\xi}|_I + \ddot{y}_N \hat{\lambda}|_I + \ddot{z}_N \hat{\zeta}|_I + 2 \left(\dot{x}_N \left. \frac{d\hat{\xi}}{dt} \right|_I + \dot{y}_N \left. \frac{d\hat{\lambda}}{dt} \right|_I + \dot{z}_N \left. \frac{d\hat{\zeta}}{dt} \right|_I \right) + x_N \left. \frac{d^2\hat{\xi}}{dt^2} \right|_I + y_N \left. \frac{d^2\hat{\lambda}}{dt^2} \right|_I + z_N \left. \frac{d^2\hat{\zeta}}{dt^2} \right|_I = \\ &= \mathbf{a}_R(t) + 2 \boldsymbol{\omega}(t) \times \left(\dot{x}_N \hat{\xi}|_I + \dot{y}_N \hat{\lambda}|_I + \dot{z}_N \hat{\zeta}|_I \right) + x_N \left. \frac{d^2\hat{\xi}}{dt^2} \right|_I + y_N \left. \frac{d^2\hat{\lambda}}{dt^2} \right|_I + z_N \left. \frac{d^2\hat{\zeta}}{dt^2} \right|_I \end{aligned}$$

Omitting the RF's subscripts and using the Poisson's relation, we can reduce the previous terms as:

$$2 \left(\dot{x}_N \frac{d\hat{\xi}}{dt} + \dot{y}_N \frac{d\hat{\lambda}}{dt} + \dot{z}_N \frac{d\hat{\zeta}}{dt} \right) = 2 \boldsymbol{\omega}(t) \times (\dot{x}_N \hat{\xi} + \dot{y}_N \hat{\lambda} + \dot{z}_N \hat{\zeta}) = 2 \boldsymbol{\omega}(t) \times \mathbf{v}_R(t) \quad (\text{A.15})$$

$$x_N \frac{d^2 \hat{\xi}}{dt^2} = x_N \frac{d}{dt} (\boldsymbol{\omega}(t) \times \hat{\xi}) = \dot{\boldsymbol{\omega}}(t) \times x_N \hat{\xi} + \boldsymbol{\omega}(t) \times \boldsymbol{\omega}(t) \times x_N \hat{\xi} \quad (\text{A.16})$$

$$y_N \frac{d^2 \hat{\lambda}}{dt^2} = y_N \frac{d}{dt} (\boldsymbol{\omega}(t) \times \hat{\lambda}) = \dot{\boldsymbol{\omega}}(t) \times y_N \hat{\lambda} + \boldsymbol{\omega}(t) \times \boldsymbol{\omega}(t) \times y_N \hat{\lambda} \quad (\text{A.17})$$

$$z_N \frac{d^2 \hat{\zeta}}{dt^2} = z_N \frac{d}{dt} (\boldsymbol{\omega}(t) \times \hat{\zeta}) = \dot{\boldsymbol{\omega}}(t) \times z_N \hat{\zeta} + \boldsymbol{\omega}(t) \times \boldsymbol{\omega}(t) \times z_N \hat{\zeta} \quad (\text{A.18})$$

Summing up the computed terms, the absolute acceleration $\mathbf{a}_A(t)$ can be expressed as:

$$\mathbf{a}_A(t) = \mathbf{a}_R(t) + 2 \boldsymbol{\omega}(t) \times \mathbf{v}_R(t) + \dot{\boldsymbol{\omega}}(t) \times \mathbf{r}(t) + \boldsymbol{\omega}(t) \times \boldsymbol{\omega}(t) \times \mathbf{r}(t) \quad (\text{A.19})$$

The last three terms of the previous equation are defined as:

- $\mathbf{a}_{\text{COR}}(t) = 2 \boldsymbol{\omega}(t) \times \mathbf{v}_R(t)$ is the Coriolis acceleration;
- $\mathbf{a}_T(t) = \dot{\boldsymbol{\omega}}(t) \times \mathbf{r}(t)$ is the tangential acceleration;
- $\mathbf{a}_F(t) = \boldsymbol{\omega}(t) \times \boldsymbol{\omega}(t) \times \mathbf{r}(t)$ is the centrifugal or centripetal acceleration.

Finally, the absolute acceleration becomes:

$$\mathbf{a}_A(t) = \mathbf{a}_R(t) + \mathbf{a}_{\text{COR}}(t) + \mathbf{a}_T(t) + \mathbf{a}_F(t) \quad (\text{A.20})$$

In case of the angular velocity $\boldsymbol{\omega}(t)$ is constant in time, the tangential acceleration $\mathbf{a}_T(t)$ is null.

Exploiting the equations (A.13) and (A.20), it is possible to switch from an Inertial RF to Non-inertial one centered in the same CoM, and vice versa, simply knowing their relative angular velocity and its derivative w.r.t. the time $\dot{\boldsymbol{\omega}}(t)$.

A.1.3 TRANSLATING AND ROTATING REFERENCE FRAME

The general case relating to coordinate systems is a Non-inertial RF (N subscript), with axes (X_N, Y_N, Z_N) , translating and rotating (roto-translation) w.r.t. an Inertial one (I subscript), with axes (X_I, Y_I, Z_I) , so that they have different CoMs, canonical bases $(\hat{\xi}, \hat{\lambda}, \hat{\zeta})$ and $(\hat{i}, \hat{j}, \hat{k})$, respectively, and relative angular velocity $\boldsymbol{\omega}(t)$ (Figure A.3).

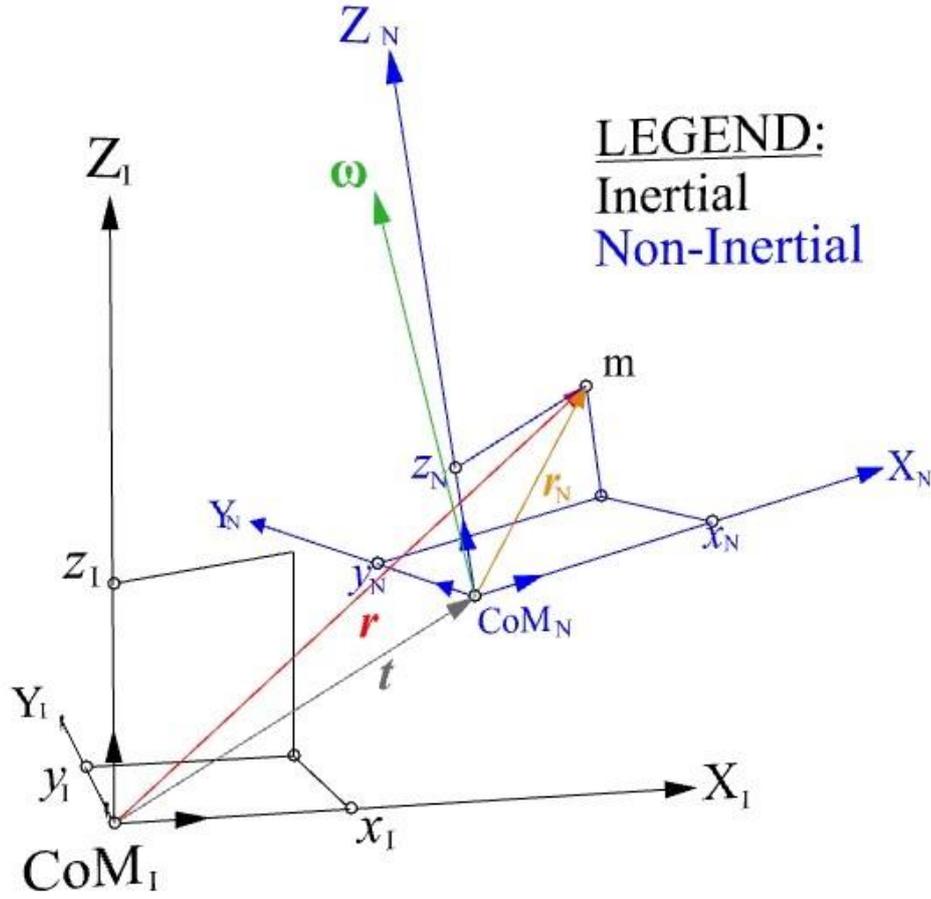


Figure A.3: The position vector \mathbf{r} w.r.t. an Inertial RF and a roto-translating RF

Now, the position vector $\mathbf{r}(t)$ can be formulated as a sum of two contributions: the translation vector between the RFs $\mathbf{t}(t)$ and the position vector $\mathbf{r}_N(t)$ w.r.t. the Non-inertial RF's CoM:

$$\mathbf{r}(t) = \mathbf{t}(t) + \mathbf{r}_N(t), \quad (\text{A.21})$$

which can be expanded by exploiting the canonical bases $(\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}})$ and $(\hat{\xi}, \hat{\lambda}, \hat{\zeta})$:

$$x_I \hat{\mathbf{i}} + y_I \hat{\mathbf{j}} + z_I \hat{\mathbf{k}} = x_O \hat{\mathbf{i}} + y_O \hat{\mathbf{j}} + z_O \hat{\mathbf{k}} + x_N \hat{\xi} + y_N \hat{\lambda} + z_N \hat{\zeta}, \quad (\text{A.22})$$

where $\mathbf{t}(t)$ has been defined as the vector $[x_O, y_O, z_O]^T$.

Obviously, the equation (A.22) can be expressed in inertial coordinates or in non-inertial ones, but since we are searching for the formulation of the absolute velocity $\mathbf{v}_A(t)$ w.r.t. the Inertial RF, we hold this frame as the reference one and we derive the equation (A.22) w.r.t. the time:

$$\dot{x}_I \hat{\mathbf{i}} + \dot{y}_I \hat{\mathbf{j}} + \dot{z}_I \hat{\mathbf{k}} = \dot{x}_O \hat{\mathbf{i}} + \dot{y}_O \hat{\mathbf{j}} + \dot{z}_O \hat{\mathbf{k}} + \dot{x}_N \hat{\xi}|_I + \dot{y}_N \hat{\lambda}|_I + \dot{z}_N \hat{\zeta}|_I + x_N \left. \frac{d\hat{\xi}}{dt} \right|_I + y_N \left. \frac{d\hat{\lambda}}{dt} \right|_I + z_N \left. \frac{d\hat{\zeta}}{dt} \right|_I \quad (\text{A.23})$$

The equation (A.23) represents the augmented expression of the formulation of the absolute velocity in a rotating frame (eq. (A.10)), with the addition of the translation velocity $\mathbf{v}_O(t)$ of the CoM of the Non-inertial RF w.r.t. the Inertial one:

$$\mathbf{v}_O(t)|_I = \dot{x}_O \hat{\mathbf{i}} + \dot{y}_O \hat{\mathbf{j}} + \dot{z}_O \hat{\mathbf{k}}, \quad (\text{A.24})$$

so that, omitting the RF's subscripts, the final expression of the absolute velocity becomes:

$$\mathbf{v}_A(t) = \mathbf{v}_O(t) + \mathbf{v}_R(t) + \boldsymbol{\omega}(t) \times \mathbf{r}_N(t) \quad (\text{A.25})$$

Also, differentiating the eq. (A.23) maintaining the Inertial RF as reference system, we obtain:

$$\begin{aligned} \ddot{x}_I \hat{\mathbf{i}} + \ddot{y}_I \hat{\mathbf{j}} + \ddot{z}_I \hat{\mathbf{k}} = & \ddot{x}_O \hat{\mathbf{i}} + \ddot{y}_O \hat{\mathbf{j}} + \ddot{z}_O \hat{\mathbf{k}} + \ddot{x}_N \hat{\boldsymbol{\xi}}|_I + \ddot{y}_N \hat{\boldsymbol{\lambda}}|_I + \ddot{z}_N \hat{\boldsymbol{\zeta}}|_I + 2 \left(\dot{x}_N \frac{d\hat{\boldsymbol{\xi}}}{dt} \Big|_I + \dot{y}_N \frac{d\hat{\boldsymbol{\lambda}}}{dt} \Big|_I + \dot{z}_N \frac{d\hat{\boldsymbol{\zeta}}}{dt} \Big|_I \right) + \\ & + \dot{x}_N \frac{d^2 \hat{\boldsymbol{\xi}}}{dt^2} \Big|_I + \dot{y}_N \frac{d^2 \hat{\boldsymbol{\lambda}}}{dt^2} \Big|_I + \dot{z}_N \frac{d^2 \hat{\boldsymbol{\zeta}}}{dt^2} \Big|_I, \end{aligned} \quad (\text{A.26})$$

where is possible to recognize the expression (A.14).

Defining the translation acceleration $\mathbf{a}_O(t)$ of the CoM of the Non-inertial RF w.r.t. the Inertial one:

$$\mathbf{a}_O(t)|_I = \ddot{x}_O \hat{\mathbf{i}} + \ddot{y}_O \hat{\mathbf{j}} + \ddot{z}_O \hat{\mathbf{k}}, \quad (\text{A.27})$$

recalling the equations (A.15) ÷ (A.20) and omitting the RF's subscripts, the final expression of the absolute acceleration becomes:

$$\mathbf{a}_A(t) = \mathbf{a}_O(t) + \mathbf{a}_R(t) + \mathbf{a}_{COR}(t) + \mathbf{a}_T(t) + \mathbf{a}_F(t) \quad (\text{A.28})$$

Exploiting the equations (A.25) and (A.28), it is possible to switch from an Inertial RF to a Non-inertial one with different CoM, and vice versa, simply knowing their relative angular velocity $\boldsymbol{\omega}(t)$, its derivative w.r.t. the time $\dot{\boldsymbol{\omega}}(t)$, the vectors $\mathbf{t}(t)$ and $\mathbf{r}_N(t)$ and their first and second derivatives w.r.t. time $\mathbf{v}_O(t)$, $\mathbf{a}_O(t)$, $\mathbf{v}_R(t)$, $\mathbf{a}_R(t)$.

A.1.4 ROTATION MATRICES

Since often the eqs (A.13), (A.20), (A.25) and (A.28) are not easily manageable, it is possible to use similar relations which exploit the rotation matrices.

Indeed, starting from the rotating RF (Figure A.2) and knowing the expressions of the position vector $\mathbf{r}(t)$ in inertial and in non-inertial coordinates (eqs. (A.2) and (A.5)), it is possible to reformulate the equation (A.5) by linking the canonical basis ($\hat{\boldsymbol{\xi}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\zeta}}$) with the inertial one:

Appendix

$$\begin{aligned}
 \hat{\xi}|_I &= (\hat{\xi} \cdot \hat{i})\hat{i} + (\hat{\xi} \cdot \hat{j})\hat{j} + (\hat{\xi} \cdot \hat{k})\hat{k} = \xi_x \hat{i} + \xi_y \hat{j} + \xi_z \hat{k}; \\
 \hat{\lambda}|_I &= (\hat{\lambda} \cdot \hat{i})\hat{i} + (\hat{\lambda} \cdot \hat{j})\hat{j} + (\hat{\lambda} \cdot \hat{k})\hat{k} = \lambda_x \hat{i} + \lambda_y \hat{j} + \lambda_z \hat{k}; \\
 \hat{\zeta}|_I &= (\hat{\zeta} \cdot \hat{i})\hat{i} + (\hat{\zeta} \cdot \hat{j})\hat{j} + (\hat{\zeta} \cdot \hat{k})\hat{k} = \zeta_x \hat{i} + \zeta_y \hat{j} + \zeta_z \hat{k},
 \end{aligned} \tag{A.29}$$

where the nine scalar products $\hat{\xi} \cdot \hat{i}, \dots, \hat{\zeta} \cdot \hat{k}$ are the direction cosines that the unit vectors $\hat{\xi}$, $\hat{\lambda}$ and $\hat{\zeta}$ form with the inertial axes (X_I, Y_I, Z_I) , or, more simply, the components of these unit vectors in inertial coordinates ξ_x, \dots, ζ_z , i.e. the projections of $\hat{\xi}$, $\hat{\lambda}$ and $\hat{\zeta}$ on the inertial axes.

Thus, the position vector $\mathbf{r}(t)$ as expressed in non-inertial coordinates can be transformed in inertial ones thanks to the relations (A.29):

$$\begin{aligned}
 \mathbf{r}(t)|_N &= x_N \hat{\xi} + y_N \hat{\lambda} + z_N \hat{\zeta} = \\
 &= x_N (\xi_x \hat{i} + \xi_y \hat{j} + \xi_z \hat{k}) + y_N (\lambda_x \hat{i} + \lambda_y \hat{j} + \lambda_z \hat{k}) + z_N (\zeta_x \hat{i} + \zeta_y \hat{j} + \zeta_z \hat{k}) = \\
 &= (x_N \xi_x + y_N \lambda_x + z_N \zeta_x) \hat{i} + (x_N \xi_y + y_N \lambda_y + z_N \zeta_y) \hat{j} + (x_N \xi_z + y_N \lambda_z + z_N \zeta_z) \hat{k} = \\
 &= x_I \hat{i} + y_I \hat{j} + z_I \hat{k} = \mathbf{r}(t)|_I
 \end{aligned} \tag{A.30}$$

Therefore, the expression (A.30) represents the transformation of the position vector $\mathbf{r}(t)$ from the Non-inertial RF to the Inertial one, which consists in a rotation of $\mathbf{r}(t)$:

$$\mathbf{r}(t)|_I = \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = \begin{bmatrix} x_N \xi_x + y_N \lambda_x + z_N \zeta_x \\ x_N \xi_y + y_N \lambda_y + z_N \zeta_y \\ x_N \xi_z + y_N \lambda_z + z_N \zeta_z \end{bmatrix} \tag{A.31}$$

Omitting the time dependence of the unit vectors $\hat{\xi}$, $\hat{\lambda}$ and $\hat{\zeta}$, we can define the rotation matrix $\mathbf{T}(t)$ as the matrix composed by the direction cosines that the unit vectors $\hat{\xi}$, $\hat{\lambda}$ and $\hat{\zeta}$ form with the inertial axes (X_I, Y_I, Z_I) :

$$\mathbf{T}(t) \stackrel{\text{def}}{=} \begin{bmatrix} \hat{\xi} \cdot \hat{i} & \hat{\xi} \cdot \hat{j} & \hat{\xi} \cdot \hat{k} \\ \hat{\lambda} \cdot \hat{i} & \hat{\lambda} \cdot \hat{j} & \hat{\lambda} \cdot \hat{k} \\ \hat{\zeta} \cdot \hat{i} & \hat{\zeta} \cdot \hat{j} & \hat{\zeta} \cdot \hat{k} \end{bmatrix} = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \lambda_x & \lambda_y & \lambda_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix}, \tag{A.32}$$

therefore, the computation of the position vector $\mathbf{r}(t)$ in inertial coordinates starting from the non-inertial ones can be obtained as a matrix-vector product looking to the equation (A.31):

$$\mathbf{r}(t)|_I = \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = \begin{bmatrix} \xi_x & \lambda_x & \zeta_x \\ \xi_y & \lambda_y & \zeta_y \\ \xi_z & \lambda_z & \zeta_z \end{bmatrix} \begin{bmatrix} x_N \\ y_N \\ z_N \end{bmatrix} = \mathbf{T}(t)^T \mathbf{r}(t)|_N \tag{A.33}$$

The rotation matrices are orthogonal, i.e.:

$$\mathbf{T}(t) \mathbf{T}(t)^T = \mathbf{T}(t)^T \mathbf{T}(t) = \mathbf{I}_{3 \times 3} \Rightarrow \mathbf{T}(t)^T = \mathbf{T}(t)^{-1} \quad (\text{A.34})$$

Since the determinant for two generic matrices \mathbf{B} and \mathbf{C} has the following properties:

1. $\det(\mathbf{B}^T) = \det(\mathbf{B})$;
2. $\det(\mathbf{B}^{-1}) = 1/\det(\mathbf{B})$;
3. $\det(\mathbf{BC}) = \det(\mathbf{B}) \det(\mathbf{C})$,

it is possible to obtain the determinant of a generic rotation matrix by using the property (A.34):

$$\det[\mathbf{T}(t)] = \det[\mathbf{T}(t)^T] = \det[\mathbf{T}(t)^{-1}] = 1/\det[\mathbf{T}(t)] , \quad (\text{A.36})$$

from which follows that every rotation matrix has a determinant with unitary value.

Coming back to the relation (A.33) and using the property (A.34), it is easy to obtain the opposite rotation, i.e. the position vector $\mathbf{r}(t)$ in non-inertial coordinates starting from the knowledge of the same vector as expressed in inertial ones:

$$\mathbf{r}(t)|_N = \mathbf{T}(t) \mathbf{r}(t)|_I \quad (\text{A.37})$$

With the same reasoning that brings to the expression (A.30) and to the definition of the rotation matrix $\mathbf{T}(t)$ (A.32), it is possible to obtain the equation (A.37) by projecting the inertial canonical basis on the non-inertial axes (X_N, Y_N, Z_N):

$$\begin{aligned} \hat{\mathbf{i}}|_N &= (\hat{\mathbf{i}} \cdot \hat{\boldsymbol{\xi}})\hat{\boldsymbol{\xi}} + (\hat{\mathbf{i}} \cdot \hat{\boldsymbol{\lambda}})\hat{\boldsymbol{\lambda}} + (\hat{\mathbf{i}} \cdot \hat{\boldsymbol{\zeta}})\hat{\boldsymbol{\zeta}} = i_\xi \hat{\boldsymbol{\xi}} + i_\lambda \hat{\boldsymbol{\lambda}} + i_\zeta \hat{\boldsymbol{\zeta}} ; \\ \hat{\mathbf{j}}|_N &= (\hat{\mathbf{j}} \cdot \hat{\boldsymbol{\xi}})\hat{\boldsymbol{\xi}} + (\hat{\mathbf{j}} \cdot \hat{\boldsymbol{\lambda}})\hat{\boldsymbol{\lambda}} + (\hat{\mathbf{j}} \cdot \hat{\boldsymbol{\zeta}})\hat{\boldsymbol{\zeta}} = j_\xi \hat{\boldsymbol{\xi}} + j_\lambda \hat{\boldsymbol{\lambda}} + j_\zeta \hat{\boldsymbol{\zeta}} ; \\ \hat{\mathbf{k}}|_N &= (\hat{\mathbf{k}} \cdot \hat{\boldsymbol{\xi}})\hat{\boldsymbol{\xi}} + (\hat{\mathbf{k}} \cdot \hat{\boldsymbol{\lambda}})\hat{\boldsymbol{\lambda}} + (\hat{\mathbf{k}} \cdot \hat{\boldsymbol{\zeta}})\hat{\boldsymbol{\zeta}} = k_\xi \hat{\boldsymbol{\xi}} + k_\lambda \hat{\boldsymbol{\lambda}} + k_\zeta \hat{\boldsymbol{\zeta}} , \end{aligned} \quad (\text{A.38})$$

where the nine scalar products $\hat{\mathbf{i}} \cdot \hat{\boldsymbol{\xi}}, \dots, \hat{\mathbf{k}} \cdot \hat{\boldsymbol{\zeta}}$ are the direction cosines which the inertial unit vectors $\hat{\mathbf{i}}, \hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$ form with the non-inertial axes (X_N, Y_N, Z_N).

Therefore, the position vector $\mathbf{r}(t)$ as expressed in inertial coordinates can be transformed in non-inertial ones thanks to the relations (A.38):

$$\begin{aligned} \mathbf{r}(t)|_I &= x_I \hat{\mathbf{i}} + y_I \hat{\mathbf{j}} + z_I \hat{\mathbf{k}} = \\ &= x_I (i_\xi \hat{\boldsymbol{\xi}} + i_\lambda \hat{\boldsymbol{\lambda}} + i_\zeta \hat{\boldsymbol{\zeta}}) + y_I (j_\xi \hat{\boldsymbol{\xi}} + j_\lambda \hat{\boldsymbol{\lambda}} + j_\zeta \hat{\boldsymbol{\zeta}}) + z_I (k_\xi \hat{\boldsymbol{\xi}} + k_\lambda \hat{\boldsymbol{\lambda}} + k_\zeta \hat{\boldsymbol{\zeta}}) = \\ &= (x_I i_\xi + y_I j_\xi + z_I k_\xi) \hat{\boldsymbol{\xi}} + (x_I i_\lambda + y_I j_\lambda + z_I k_\lambda) \hat{\boldsymbol{\lambda}} + (x_I i_\zeta + y_I j_\zeta + z_I k_\zeta) \hat{\boldsymbol{\zeta}} = \end{aligned} \quad (\text{A.39})$$

Appendix

$$= x_N \hat{\xi} + y_N \hat{\lambda} + z_N \hat{\zeta} = \mathbf{r}(t)|_N,$$

which can be shown in matrix form:

$$\mathbf{r}(t)|_N = \begin{bmatrix} x_N \\ y_N \\ z_N \end{bmatrix} = \begin{bmatrix} x_1 i_\xi + y_1 j_\xi + z_1 k_\xi \\ x_1 i_\lambda + y_1 j_\lambda + z_1 k_\lambda \\ x_1 i_\zeta + y_1 j_\zeta + z_1 k_\zeta \end{bmatrix} \quad (\text{A.40})$$

Now, it is easy to note the connection between the eqs (A.30) and (A.39).

Indeed, recalling the definition of the matrix $\mathbf{T}(t)$, we can show its alternative expression:

$$\mathbf{T}(t) \stackrel{\text{def}}{=} \begin{bmatrix} \hat{\xi} \cdot \hat{\mathbf{i}} & \hat{\xi} \cdot \hat{\mathbf{j}} & \hat{\xi} \cdot \hat{\mathbf{k}} \\ \hat{\lambda} \cdot \hat{\mathbf{i}} & \hat{\lambda} \cdot \hat{\mathbf{j}} & \hat{\lambda} \cdot \hat{\mathbf{k}} \\ \hat{\zeta} \cdot \hat{\mathbf{i}} & \hat{\zeta} \cdot \hat{\mathbf{j}} & \hat{\zeta} \cdot \hat{\mathbf{k}} \end{bmatrix} = \begin{bmatrix} i_\xi & j_\xi & k_\xi \\ i_\lambda & j_\lambda & k_\lambda \\ i_\zeta & j_\zeta & k_\zeta \end{bmatrix}, \quad (\text{A.41})$$

so that, the computation of the position vector $\mathbf{r}(t)$ in non-inertial coordinates can be obtained as:

$$\mathbf{r}(t)|_N = \begin{bmatrix} x_N \\ y_N \\ z_N \end{bmatrix} = \begin{bmatrix} i_\xi & j_\xi & k_\xi \\ i_\lambda & j_\lambda & k_\lambda \\ i_\zeta & j_\zeta & k_\zeta \end{bmatrix} \begin{bmatrix} x_I \\ y_I \\ z_I \end{bmatrix} = \mathbf{T}(t) \mathbf{r}(t)|_I, \quad (\text{A.42})$$

which is obviously like the equation (A.37).

Finally, the goal is to define the rotation matrix $\mathbf{T}(t)$; once it has been defined, we can compute the absolute velocity and the absolute acceleration of the eqs (A.13), (A.20),(A.25) and (A.28) by operating with matrix products instead of vector one.

A.1.5 TRANSFORMATIONS IN NON-INERTIAL COORDINATES BY USING ROTATION MATRICES

Starting from the rotating frame of Figure A.2 explained in section A.1.2 Rotating Reference Frame and recalling the eqs (A.33) and (A.37), it is possible to obtain the absolute velocity by differentiating in time the position vector $\mathbf{r}(t)$, expressed in non-inertial coordinates, w.r.t. the Inertial RF:

$$\mathbf{v}_A(t) = \left. \frac{d\mathbf{r}(t)|_N}{dt} \right|_I = \mathbf{T}(t) \dot{\mathbf{r}}(t)|_I + \dot{\mathbf{T}}(t) \mathbf{r}(t)|_I = \mathbf{T}(t) \dot{\mathbf{r}}(t)|_I + \dot{\mathbf{T}}(t) \mathbf{T}(t)^T \mathbf{r}(t)|_N \quad (\text{A.43})$$

Obviously, the equation (A.43) must be coincident with the (A.13) one, i.e.:

$$\mathbf{T}(t) \dot{\mathbf{r}}(t)|_I + \dot{\mathbf{T}}(t) \mathbf{T}(t)^T \mathbf{r}(t)|_N = \mathbf{v}_R(t) + \boldsymbol{\omega}(t) \times \mathbf{r}(t)|_N, \quad (\text{A.44})$$

which brings to the following equations:

$$\begin{aligned} \mathbf{T}(t) \dot{\mathbf{r}}(t)|_I &= \mathbf{v}_R(t) \\ \dot{\mathbf{T}}(t) \mathbf{T}(t)^T \mathbf{r}(t)|_N &= \boldsymbol{\omega}(t) \times \mathbf{r}(t)|_N \end{aligned} \quad (\text{A.45})$$

therefore, we can define the operation of the vector product $\boldsymbol{\omega}(t) \times$ as the skew-symmetric matrix $[\boldsymbol{\omega}(t)]^\otimes$, i.e.:

$$[\boldsymbol{\omega}(t)]^\otimes \stackrel{\text{def}}{=} \dot{\mathbf{T}}(t) \mathbf{T}(t)^T, \quad (\text{A.46})$$

With the same reasoning, it is possible to obtain the expression of the absolute acceleration of the S/C as a sum of four contributions, like the equation (A.20), by computing the second derivative w.r.t. the time of $\mathbf{r}(t)|_N$ w.r.t. the Inertial RF, i.e. differentiating the equation (A.43):

$$\begin{aligned} \mathbf{a}_A(t) &= \left. \frac{d^2 \mathbf{r}(t)|_N}{dt^2} \right|_I = \mathbf{T}(t) \ddot{\mathbf{r}}(t)|_I + \dot{\mathbf{T}}(t) \dot{\mathbf{r}}(t)|_I + \dot{\mathbf{T}}(t) \dot{\mathbf{r}}(t)|_I + \ddot{\mathbf{T}}(t) \mathbf{r}(t)|_I = \\ &= \mathbf{T}(t) \ddot{\mathbf{r}}(t)|_I + 2 \dot{\mathbf{T}}(t) \dot{\mathbf{r}}(t)|_I + \ddot{\mathbf{T}}(t) \mathbf{T}(t)^T \mathbf{r}(t)|_N \end{aligned} \quad (\text{A.47})$$

Obviously, the equation (A.47) must be coincident with the (A.19) one, i.e.:

$$\begin{aligned} \mathbf{T}(t) \ddot{\mathbf{r}}(t)|_I &= \mathbf{a}_R(t) \\ 2 \dot{\mathbf{T}}(t) \dot{\mathbf{r}}(t)|_I &= 2 \boldsymbol{\omega}(t) \times \mathbf{v}_R(t) = \mathbf{a}_{\text{COR}}(t) \\ \ddot{\mathbf{T}}(t) \mathbf{T}(t)^T \mathbf{r}(t)|_N &= \boldsymbol{\omega}(t) \times \boldsymbol{\omega}(t) \times \mathbf{r}(t)|_N + \dot{\boldsymbol{\omega}}(t) \times \mathbf{r}(t)|_N = \mathbf{a}_F(t) + \mathbf{a}_T(t) \end{aligned} \quad (\text{A.48})$$

Once determined the absolute velocity and the absolute acceleration of a rotating frame centered in the same CoM of the Inertial RF, it is possible to extend the reasoning to the roto-translating RF of Figure A.3, where the position vector $\mathbf{r}(t)$ can be obtained from the equation (A.21) by making use of the rotation matrix $\mathbf{T}(t)$, that is defined in the same way of (A.32) regardless the translation of the Non-inertial RF.

Indeed, $\mathbf{r}_N(t)$ can be transformed from a RF to another by exploiting $\mathbf{T}(t)$:

$$\mathbf{r}_N(t)|_N = \mathbf{T}(t) \mathbf{r}_N(t)|_I, \quad (\text{A.49})$$

thus, computing the derivative w.r.t. the time of the equation (A.21), we obtain:

$$\begin{aligned} \mathbf{v}_A(t) &= \left. \frac{d\mathbf{r}(t)}{dt} \right|_I = \left. \frac{d\mathbf{t}(t)}{dt} \right|_I + \left. \frac{d\mathbf{r}_N(t)|_N}{dt} \right|_I = \dot{\mathbf{t}}(t) + \mathbf{T}(t) \dot{\mathbf{r}}_N(t)|_I + \dot{\mathbf{T}}(t) \mathbf{r}_N(t)|_I = \\ &= \dot{\mathbf{t}}(t) + \mathbf{T}(t) \dot{\mathbf{r}}_N(t)|_I + \dot{\mathbf{T}}(t) \mathbf{T}(t)^T \mathbf{r}_N(t)|_N, \end{aligned} \quad (\text{A.50})$$

which must be coincident with the equation (A.25), i.e.:

$$\begin{aligned}
 \dot{\mathbf{i}}(t) &= \mathbf{v}_O(t) \\
 \mathbf{T}(t) \dot{\mathbf{r}}_N(t)|_I &= \mathbf{v}_R(t) \\
 \dot{\mathbf{T}}(t) \mathbf{T}(t)^T \mathbf{r}_N(t)|_N &= \boldsymbol{\omega}(t) \times \mathbf{r}_N(t)|_N
 \end{aligned} \tag{A.51}$$

With the same reasoning, it is possible to obtain the expression of the absolute acceleration for the roto-translating RF like the equation (A.28) by differentiating the relation (A.50) w.r.t. the time:

$$\begin{aligned}
 \mathbf{a}_A(t) &= \left. \frac{d^2 \mathbf{r}(t)}{dt^2} \right|_I = \ddot{\mathbf{i}}(t) + \mathbf{T}(t) \ddot{\mathbf{r}}_N(t)|_I + \dot{\mathbf{T}}(t) \dot{\mathbf{r}}_N(t)|_I + \ddot{\mathbf{T}}(t) \mathbf{r}_N(t)|_I = \\
 &= \ddot{\mathbf{i}}(t) + \mathbf{T}(t) \ddot{\mathbf{r}}_N(t)|_I + 2 \dot{\mathbf{T}}(t) \dot{\mathbf{r}}_N(t)|_I + \ddot{\mathbf{T}}(t) \mathbf{r}_N(t)|_I,
 \end{aligned} \tag{A.52}$$

which must be coincident with the equation (A.28), i.e.:

$$\begin{aligned}
 \ddot{\mathbf{i}}(t) &= \mathbf{a}_O(t) \\
 \mathbf{T}(t) \ddot{\mathbf{r}}_N(t)|_I &= \mathbf{a}_R(t) \\
 2 \dot{\mathbf{T}}(t) \dot{\mathbf{r}}_N(t)|_I &= 2 \boldsymbol{\omega}(t) \times \mathbf{v}_R(t) = \mathbf{a}_{COR}(t) \\
 \ddot{\mathbf{T}}(t) \mathbf{r}_N(t)|_I &= \ddot{\mathbf{T}}(t) \mathbf{T}(t)^T \mathbf{r}_N(t)|_N = \boldsymbol{\omega}(t) \times \boldsymbol{\omega}(t) \times \mathbf{r}_N(t)|_N + \dot{\boldsymbol{\omega}}(t) \times \mathbf{r}_N(t)|_N = \mathbf{a}_F(t) + \mathbf{a}_T(t)
 \end{aligned} \tag{A.53}$$

Finally, the transformations of $\mathbf{v}_A(t)$ and $\mathbf{a}_A(t)$ from the Non-inertial RF to the Inertial one and vice versa are possible by exploiting the eqs (A.13), (A.20),(A.25) and (A.28), which require the knowledge of the angular speed $\boldsymbol{\omega}(t)$, or, similarly, by using the eqs (A.43), (A.47), (A.50) and (A.52), which involve the acquaintance of the rotation matrix $\mathbf{T}(t)$.

Usually, the relations in matrix form are easier manageable than the ones in vector form.

A.2 THE CIRCULAR RESTRICTED 3-BODY PROBLEM (CR3BP)

Considering a binary system in space, such as the Earth-Moon one, and a S/C immersed in it, the general 3-Body Problem is the study of the behaviour of the dynamics of this system as it was composed by three punctual massive bodies of known masses moving due to the reciprocal gravitational attraction, perturbed by external forces that can be considered as disturbances for the isolated system.

The CR3BP simplifies the general 3-Body Problem by assuming that one of the three bodies, the S/C, has a negligible mass, i.e. imposing the satellite does not exert any force on the other two punctual massive bodies, which have known masses M and m ($M > m$) (Figure A.4).

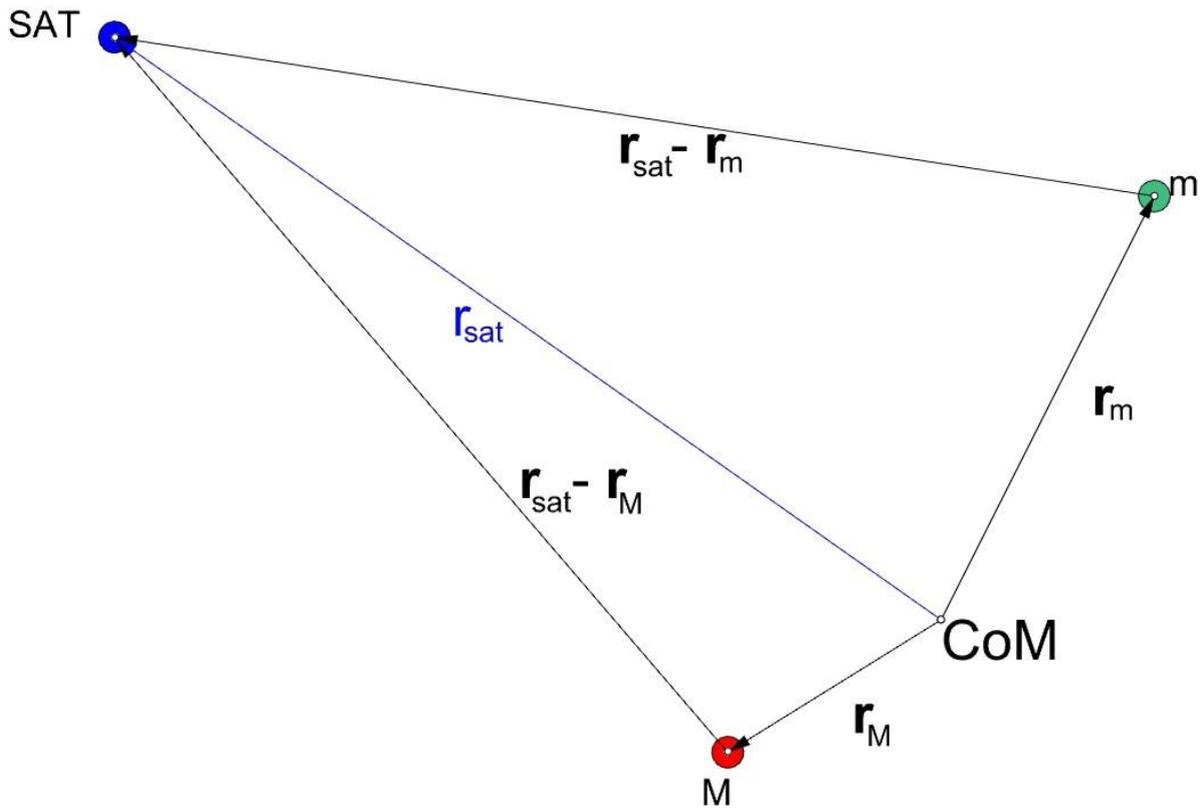


Figure A.4: The generic 3-Body Problem (3BP)

Considering a Reference Frame centered in the CoM of the isolated system, with the x-y plane coincident with the mean one of the mutual orbit of the two massive bodies, which just follows the CoM of the system w/o rotating, so that it can be minded as quasi-inertial, and referring to the [Figure A.4](#), the first approximation needed to compute the state of the S/C is to take into account only the gravitational attractions exerted by M and m exploiting the well-known Newton's Law of Universal Gravitation, i.e. expressing the acceleration acting on the S/C like:

$$\begin{aligned} \mathbf{a}_{\text{sat}}(t) &= -\frac{GM}{\|\mathbf{r}_{\text{sat}}(t) - \mathbf{r}_M(t)\|^3} [\mathbf{r}_{\text{sat}}(t) - \mathbf{r}_M(t)] - \frac{Gm}{\|\mathbf{r}_{\text{sat}}(t) - \mathbf{r}_m(t)\|^3} [\mathbf{r}_{\text{sat}}(t) - \mathbf{r}_m(t)] = \\ &= -\frac{\mu_M}{\|\mathbf{r}_1(t)\|^3} \mathbf{r}_1(t) - \frac{\mu_m}{\|\mathbf{r}_2(t)\|^3} \mathbf{r}_2(t), \end{aligned} \quad (\text{A.54})$$

where μ_M and μ_m are the gravitational parameters of the massive bodies M and m, respectively, and where we have abbreviated the vectors $\mathbf{r}_{\text{sat}}(t) - \mathbf{r}_M(t)$ and $\mathbf{r}_{\text{sat}}(t) - \mathbf{r}_m(t)$ with $\mathbf{r}_1(t)$ and $\mathbf{r}_2(t)$.

Since M and m orbit around the same CoM, the most suitable RF to show and study the trajectory of the SAT is the Synodic one, which is co-rotating with the binary system, i.e. rotating w.r.t. the quasi-inertial RF with angular velocity $\boldsymbol{\omega}(t)$, which, conventionally, is considered as constant in time and equal to the mean angular velocity of the mass m around M.

Indeed, it is centered in the CoM of the system, with the x-axis always directed as the line of

Appendix

conjunction M-m, from M to m, the z-axis directed as the first integral of motion (h-axis) and the y-axis that completes the right-hand frame.

Now, considering the equation (A.54), we take advantage, for convention, of non-dimensional units, i.e. the total mass of the system, the distance between the primaries and the modulus of the angular velocity are supposed unitary: $M_{tot} = M + m = 1$, $d = 1$, $\|\boldsymbol{\omega}\| = 1$.

These assumptions constrain to a unitary universal gravitational constant ($G = 1$).

To proceed in this analysis, it is useful to define the so-called *mass parameter* $\eta < 0.5$ such that:

$$\eta \stackrel{\text{def}}{=} \frac{m}{M+m} \quad 1-\eta = \frac{M}{M+m} \quad (\text{A.55})$$

Also, the massive bodies M and m are posed at distance, respectively, $(-\eta)$ and $(1-\eta)$ along the x-axis (Figure A.5), so that we can redefine the distances $r_1(t)$ and $r_2(t)$ like:

$$\begin{aligned} r_1(t) &= \sqrt{(x(t) + \eta)^2 + y(t)^2 + z(t)^2} \\ r_2(t) &= \sqrt{(x(t) - 1 + \eta)^2 + y(t)^2 + z(t)^2} \end{aligned} \quad (\text{A.56})$$

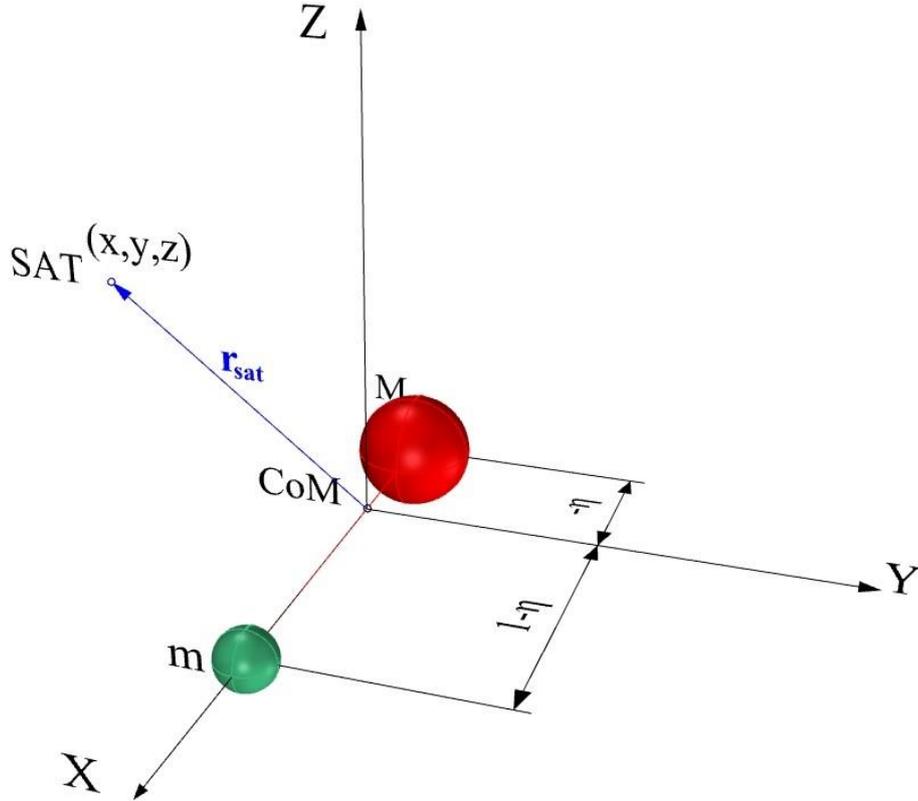


Figure A.5: The Circular Restricted 3-Body Problem (CR3BP)

Once the previous simplifications have been made, we can proceed to compute the equation of motion in vector form for the massless S/C.

Since the Synodic RF is non-inertial and has the same CoM of the quasi-inertial one, we can express it by exploiting the relation (A.19), recalling that $\boldsymbol{\omega}$ is constant:

$$\frac{d^2 \mathbf{r}_{\text{sat}}(t)}{dt^2} = \mathbf{a}_R(t) + 2 \boldsymbol{\omega} \times \mathbf{v}_{R_{\text{sat}}}(t) + \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{r}_{\text{sat}}(t) \quad (\text{A.57})$$

Also, the equation of motion can be computed by using its classical expression in vector form for a generic RF, namely:

$$\frac{d^2 \mathbf{r}_{\text{sat}}(t)}{dt^2} = \nabla U(\mathbf{r}_{\text{sat}}(t)) = \frac{\partial U(\mathbf{r}_{\text{sat}}(t))}{\partial \mathbf{r}_{\text{sat}}(t)} \quad (\text{A.58})$$

Comparing the eqs (A.57) and (A.58) among them, we can define the effective potential U can be as the sum of the gravitational one U_G with the centrifugal one U_C , i.e.:

$$U(\mathbf{r}_{\text{sat}}(t)) \stackrel{\text{def}}{=} U_G(\mathbf{r}_{\text{sat}}(t)) + U_C(\mathbf{r}_{\text{sat}}(t)) = \frac{1-\eta}{r_1(t)} + \frac{\eta}{r_2(t)} + \frac{1}{2} [\boldsymbol{\omega} \times \mathbf{r}_{\text{sat}}(t)] \cdot [\boldsymbol{\omega} \times \mathbf{r}_{\text{sat}}(t)] \quad (\text{A.59})$$

Finally, substituting the effective potential in the equation (A.58), expanding the equations of motion in the three coordinates and omitting the time dependence, we obtain:

$$\begin{cases} \ddot{x}_{\text{sat}} - 2\dot{y}_{\text{sat}} = x_{\text{sat}} - \frac{(1-\eta)}{r_1^3} (x_{\text{sat}} + \eta) - \frac{\eta}{r_2^3} (x_{\text{sat}} - 1 + \eta) \\ \ddot{y}_{\text{sat}} + 2\dot{x}_{\text{sat}} = y_{\text{sat}} - \frac{(1-\eta)}{r_1^3} y_{\text{sat}} - \frac{\eta}{r_2^3} y_{\text{sat}} \\ \ddot{z}_{\text{sat}} = -\frac{(1-\eta)}{r_1^3} z_{\text{sat}} - \frac{\eta}{r_2^3} z_{\text{sat}} \end{cases} \quad (\text{A.60})$$

We are interesting to compute the points of the Synodic RF in which the velocity and the acceleration are null. To do this, we impose the following condition:

$$(\ddot{x}_{\text{sat}}, \ddot{y}_{\text{sat}}, \ddot{z}_{\text{sat}}) = (\dot{x}_{\text{sat}}, \dot{y}_{\text{sat}}, \dot{z}_{\text{sat}}) = (0,0,0), \quad (\text{A.61})$$

so that, the equations of motion (A.60) simply become:

$$\begin{cases} x_{\text{sat}} - \frac{(1-\eta)}{r_1^3} (x_{\text{sat}} + \eta) - \frac{\eta}{r_2^3} (x_{\text{sat}} - 1 + \eta) = 0 \\ y_{\text{sat}} - \frac{(1-\eta)}{r_1^3} y_{\text{sat}} - \frac{\eta}{r_2^3} y_{\text{sat}} = 0 \\ -\frac{(1-\eta)}{r_1^3} z_{\text{sat}} - \frac{\eta}{r_2^3} z_{\text{sat}} = 0 \end{cases} \quad (\text{A.62})$$

Appendix

Trying to solve the eqs (A.62), we can make some considerations:

- the third eq. places all equilibrium points on the x-y plane, since it's verified only for $z_{\text{sat}} = 0$;
- the second eq. allows just two solutions: $y_{\text{sat}} = 0$ or $1 - \frac{(1-\eta)}{r_1^3} - \frac{\eta}{r_2^3} = 0$;
- the first eq. provides the last coordinate, having solved the last two eqs before.

Finally, omitting the complete resolution of the eqs (A.62), the solver obtains just five equilibrium points, the so-called *Lagrange points* (Figure A.6):

1. L_1, L_2 and L_3 are the *collinear* equilibrium points, since they lie on the x-axis, where $y_{\text{sat}} = 0$;
2. L_4 and L_5 are the *triangular* equilibrium points, since $y_{\text{sat}} \neq 0$ and $1 - \frac{(1-\eta)}{r_1^3} - \frac{\eta}{r_2^3} = 0$

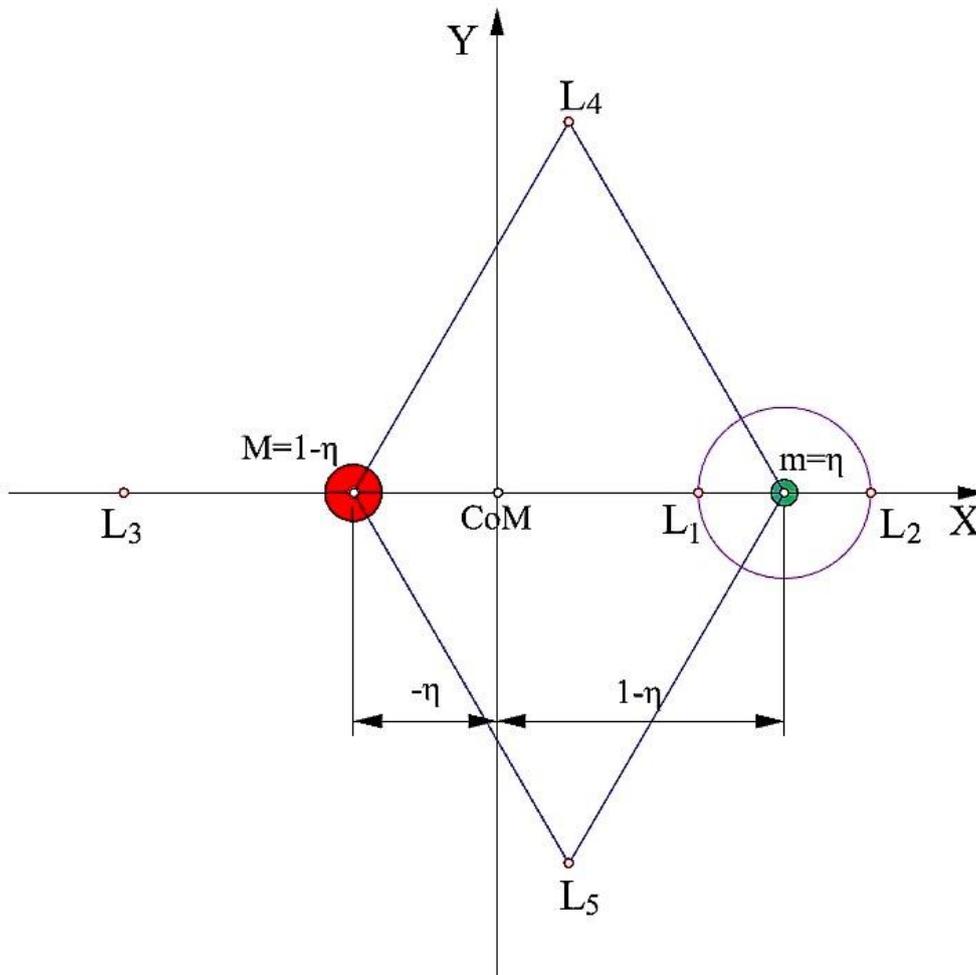


Figure A.6: Lagrange points in the Synodic Reference Frame

Indeed, starting from the condition $y_{\text{sat}} = 0$ and defining the parameter u as:

$$u = x_{\text{sat}} - 1 + \eta, \quad (\text{A.63})$$

it is possible to simplify the eqs (A.62), obtaining the well-known *Lagrange quintic equation*, which is an algebraic equation of degree 5 in u :

$$u^2(1 - s_1 + 3u + 3u^2 + u^3) = \eta[s_0 + 2s_0u + (1 + s_0 - s_1)u^2 + 2u^3 + u^4], \quad (\text{A.64})$$

where $s_0 = \text{sign}(u)$ and $s_1 = \text{sign}(u + 1)$.

The equation can be solved by suitable iterative methods, where the convergence of the solution is guaranteed only for a proper choice of the initial values, finally obtaining just three real solutions, which are, as said before, the *collinear* equilibrium points L_1, L_2, L_3 .

The experience suggests that the initial values of u_1, u_2 and u_3 must be selected by exploiting three adimensional parameters α, ν and β , which are defined as:

$$\alpha \stackrel{\text{def}}{=} \left[\frac{\eta}{3(1 - \eta)} \right]^{\frac{1}{3}}, \quad \nu \stackrel{\text{def}}{=} \frac{7}{12} \eta, \quad \beta \stackrel{\text{def}}{=} \eta \left(1 + \frac{23}{84} \eta^2 \right), \quad (\text{A.65})$$

so that, the initial values become:

$$\begin{aligned} u_{1_0} &\stackrel{\text{def}}{=} \alpha \left(1 - \frac{\alpha}{3} - \frac{\alpha^2}{9} \right) \\ u_{2_0} &\stackrel{\text{def}}{=} \alpha \left(1 + \frac{\alpha}{3} - \frac{\alpha^2}{9} \right) \\ u_{3_0} &\stackrel{\text{def}}{=} \beta - 2 \end{aligned} \quad (\text{A.66})$$

The *triangular* equilibrium points L_4 and L_5 are computed considering the other condition on the second equation of the system (A.62), i.e. solving the following couple of equations:

$$\begin{cases} x_{\text{sat}} - \frac{(1 - \eta)}{r_1^3} (x_{\text{sat}} + \eta) - \frac{\eta}{r_2^3} (x_{\text{sat}} - 1 + \eta) = 0 \\ 1 - \frac{(1 - \eta)}{r_1^3} - \frac{\eta}{r_2^3} = 0, \end{cases} \quad (\text{A.67})$$

which brings to the exact values:

$$\begin{aligned} L_4 &= \left(\frac{1}{2} - \eta, \frac{\sqrt{3}}{2}, 0 \right)^T \\ L_5 &= \left(\frac{1}{2} - \eta, -\frac{\sqrt{3}}{2}, 0 \right)^T \end{aligned} \quad (\text{A.68})$$

A.3 GENERAL CAUCHY PROBLEM

The Cauchy problem, also known as Initial Value Problem (IVP), associated to one ODE of the first order requests to find a function $y \in C^1(I)$ such that:

$$\begin{cases} \dot{y}(t) = f(t, y(t)) & t \in I \\ y(t_0) = y_0 & t_0 \in I, \end{cases} \quad (\text{A.69})$$

where $f(t, y)$ is a function assigned in the strip $S = I \times (-\infty, +\infty)$, composed by real values and continuous w.r.t. both variables, which represents the first derivative of $y(t)$.

In case of a system of ODEs of the first order, the goal is to find a vector function $\mathbf{y} \in \mathbb{R}^l$ composed by l functions $y^1 \in C^1(I), \dots, y^l \in C^1(I)$ such that:

$$\begin{cases} \dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t)) & t \in I \\ \mathbf{y}(t_0) = \mathbf{y}_0 & t_0 \in I, \end{cases} \quad (\text{A.70})$$

where $\mathbf{f}(t, \mathbf{y})$ is a vector function which represents the first derivative of $\mathbf{y}(t)$, with f^1, \dots, f^l assigned in the strip $S = I \times (-\infty, +\infty)$, composed by real values and continuous w.r.t. both variables.

A.4 STATISTICS FOR RANDOM VECTORS

Considering two random vector signals \mathbf{X} in \mathbb{R}^n and \mathbf{Y} in \mathbb{R}^m , these are defined as two vectors composed by n and m random variables, respectively, which can be continuous or discrete, i.e.:

$$\begin{aligned} \mathbf{X} &= [X_1 \quad \dots \quad X_n]^T \\ \mathbf{Y} &= [Y_1 \quad \dots \quad Y_m]^T, \end{aligned} \quad (\text{A.71})$$

we can make use of several operators in signal theory (*Luise & Vitetta, 2009, June 1*), which are useful to describe the main features of vector signals.

Here below, will be presented the most important statistics for the optimal controlling and filtering, omitting the theory of the statistics for the scalar random variables.

A.4.1 EXPECTED VALUE:

The expected or mean value of a random vector signal \mathbf{X} in \mathbb{R}^n is:

$$\boldsymbol{\eta}_X = E[\mathbf{X}] \stackrel{\text{def}}{=} [E[X_1] \quad \dots \quad E[X_n]]^T = [\eta_{X_1} \quad \dots \quad \eta_{X_n}]^T, \quad (\text{A.72})$$

where $\eta_{X_1}, \dots, \eta_{X_n}$ are the mean values of the random variables X_1, \dots, X_n .

A.4.2 COVARIANCE MATRIX:

The covariance matrix of a random vector signal \mathbf{X} in \mathbb{R}^n is an $n \times n$ symmetric matrix defined as:

$$\mathbf{C}_X = \text{cov}(\mathbf{X}, \mathbf{X}) \stackrel{\text{def}}{=} E[(\mathbf{X} - \boldsymbol{\eta}_X)(\mathbf{X} - \boldsymbol{\eta}_X)^T] = \begin{bmatrix} \text{cov}(X_1, X_1) & \cdots & \text{cov}(X_1, X_n) \\ \vdots & \ddots & \vdots \\ \text{cov}(X_n, X_1) & \cdots & \text{cov}(X_n, X_n) \end{bmatrix}, \quad (\text{A.73})$$

which is the matrix of the covariances of the random variables X_1, \dots, X_n , that are defined through the following well-known relations:

$$\begin{aligned} \text{cov}(X_i, X_j) &\stackrel{\text{def}}{=} E[(X_i - \eta_{X_i})(X_j - \eta_{X_j})] \\ \text{cov}(X_i, X_i) &= E[(X_i - \eta_{X_i})^2] \stackrel{\text{def}}{=} \text{var}(X_i) = \sigma_{X_i}^2, \end{aligned} \quad (\text{A.74})$$

where $\text{var}(X_i)$ is the variance of the scalar random variable X_i , which is the squared value of the standard deviation $\sigma_{X_i} = \text{std}(X_i)$.

A.4.3 CORRELATION MATRIX:

The correlation matrix of a random vector signal \mathbf{X} in \mathbb{R}^n is an $n \times n$ symmetric matrix defined as:

$$\mathbf{R}_X = \text{corr}(\mathbf{X}) \stackrel{\text{def}}{=} E[\mathbf{X}\mathbf{X}^T] = \begin{bmatrix} \text{corr}(X_1, X_1) & \cdots & \text{corr}(X_1, X_n) \\ \vdots & \ddots & \vdots \\ \text{corr}(X_n, X_1) & \cdots & \text{corr}(X_n, X_n) \end{bmatrix}, \quad (\text{A.75})$$

which is the matrix of the correlations of the random variables X_1, \dots, X_n , that are defined through the following well-known relations:

$$\begin{aligned} \text{corr}(X_i, X_j) &\stackrel{\text{def}}{=} \frac{\text{cov}(X_i, X_j)}{\sigma_{X_i}\sigma_{X_j}} \\ \text{corr}(X_i, X_i) &= \frac{\text{cov}(X_i, X_i)}{\sigma_{X_i}\sigma_{X_i}} = 1, \end{aligned} \quad (\text{A.76})$$

so that, the covariance matrix \mathbf{C}_X of eq. (A.73) can be rewritten as:

$$\mathbf{C}_X = \mathbf{R}_X - \boldsymbol{\eta}_X \boldsymbol{\eta}_X^T \quad (\text{A.77})$$

A.4.4 CROSS-CORRELATION MATRIX:

The cross-correlation matrix of two random vector signals \mathbf{X} in \mathbb{R}^n and \mathbf{Y} in \mathbb{R}^m is an $n \times m$ matrix, which is defined as:

Appendix

$$\mathbf{R}_{XY} = \text{corr}(\mathbf{X}, \mathbf{Y}) \stackrel{\text{def}}{=} E[\mathbf{X}\mathbf{Y}^T] = \begin{bmatrix} \text{corr}(X_1, Y_1) & \cdots & \text{corr}(X_1, Y_n) \\ \vdots & \ddots & \vdots \\ \text{corr}(X_n, Y_1) & \cdots & \text{corr}(X_n, Y_n) \end{bmatrix} \quad (\text{A.78})$$

As in scalar theory, the random vectors \mathbf{X} in \mathbb{R}^n and \mathbf{Y} in \mathbb{R}^m are said to be uncorrelated iff the following relation is verified:

$$\text{corr}(\mathbf{X}, \mathbf{Y}) = \mathbf{0} \quad (\text{A.79})$$

A.4.5 CROSS-COVARIANCE MATRIX:

The cross-covariance matrix of two random vector signals \mathbf{X} in \mathbb{R}^n and \mathbf{Y} in \mathbb{R}^m is an $n \times m$ matrix, that is defined as:

$$\begin{aligned} \mathbf{C}_{XY} = \text{cov}(\mathbf{X}, \mathbf{Y}) &\stackrel{\text{def}}{=} E[(\mathbf{X} - \boldsymbol{\eta}_X)(\mathbf{Y} - \boldsymbol{\eta}_Y)^T] = \begin{bmatrix} \text{cov}(X_1, Y_1) & \cdots & \text{cov}(X_1, Y_m) \\ \vdots & \ddots & \vdots \\ \text{cov}(X_n, Y_1) & \cdots & \text{cov}(X_n, Y_m) \end{bmatrix} = \\ &= E[\mathbf{X}\mathbf{Y}^T] - \boldsymbol{\eta}_X \boldsymbol{\eta}_Y^T = \mathbf{R}_{XY} - \boldsymbol{\eta}_X \boldsymbol{\eta}_Y^T \end{aligned} \quad (\text{A.80})$$

A.4.6 GAUSSIAN VECTORS:

A random vector \mathbf{X} in \mathbb{R}^n is a Gaussian vector iff it is composed by Gaussian random variables, i.e. every random variable has a probability density function (pdf) defined as:

$$f_{X_i}(x_i) = \frac{1}{\sqrt{2\pi\sigma_X^2}} e^{-\frac{(x-\eta_X)^2}{2\sigma_X^2}}, \quad (\text{A.81})$$

so that, it is possible to represent every random variable through a simplified notation:

$$X_i \sim \mathcal{N}(\eta_X, \sigma_X^2) \quad (\text{A.82})$$

In the case of the Gaussian random variable having zero mean, the plot of the probability density function is symmetric w.r.t. the y-axis ([Figure A.7](#)).

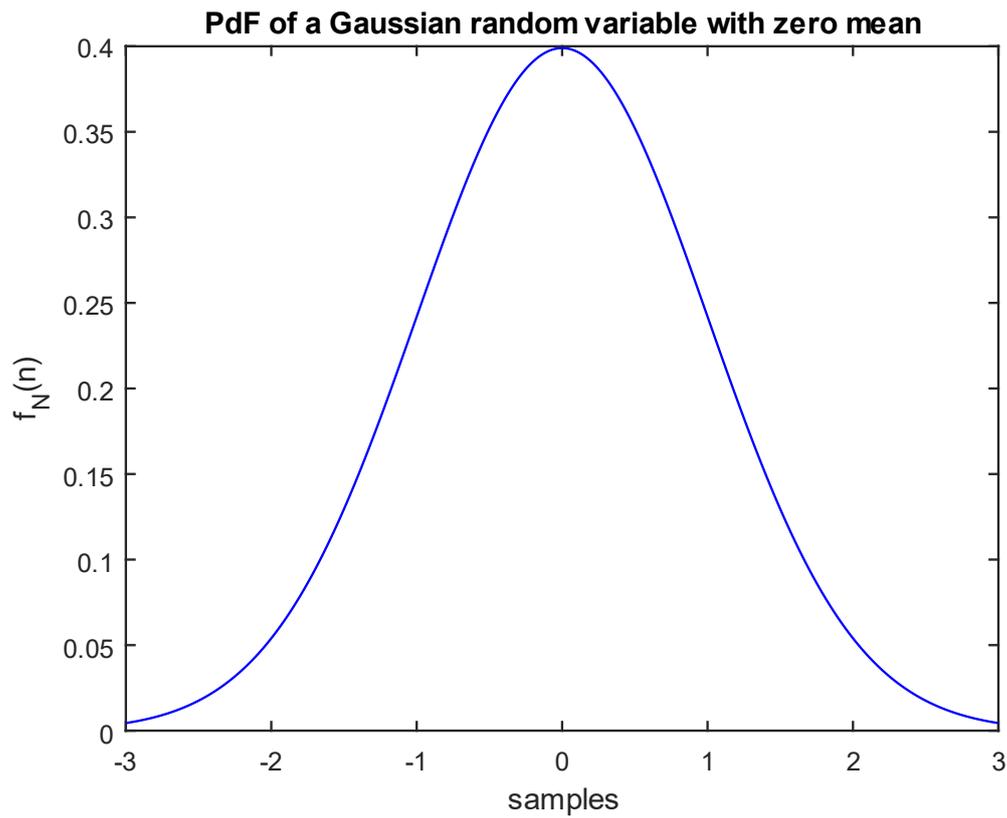


Figure A.7: Probability density function of a Gaussian random variable with zero mean

Adopting the same notation of the scalar random variables, a Gaussian vector can be represented as:

$$\mathbf{X} \sim \mathcal{N}(\eta_X, \sigma_X^2), \quad (\text{A.83})$$

which owns a covariance matrix \mathbf{C}_X in diagonal form, since the random variables of a random vector are independent each other, namely:

$$\mathbf{C}_X = \begin{bmatrix} \sigma_{X_1}^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{X_n}^2 \end{bmatrix} \quad (\text{A.84})$$

REFERENCES

- [1] Abrahamson, M. J., Kennedy, B. A., & Bhaskaran, S. (2012, June 14). Autonomous Navigation Performance During The Hartley 2 Comet Flyby. *SpaceOps 2012 Conference*. Stockholm (SW): Jet Propulsion Laboratory (JPL), NASA. Retrieved from <http://hdl.handle.net/2014/42446>
- [2] Aguado, F., Tortora, P., Modenini, D., Dolgos, G., Pérez, F., Vázquez, A., . . . Lasagni Manghi, R. (2016, May 6). *DustCube Technical Note 2: Mission Concept and Preliminary Assessment*. Technical Report, University of Vigo (UVIGO), University of Bologna (UNIBO), Micos Engineering.
- [3] Avanzi, A. (2012). *Design and implementation of a novel multi-constellation FPGA-based dual frequency GNSS receiver for space applications*. Dissertation of Doctorate of Philosophy (PhD) in Aerospace Plants and Systems, University of Bologna, Faculty of Aerospace Engineering, Forlì (IT). Retrieved from http://amsdottorato.unibo.it/4751/1/avanzi_alessandro_tesi.pdf
- [4] Bhaskaran, S. (2012, June 11-15). Autonomous Navigation for Deep Space Missions. *SpaceOps 2012 Conference*. Stockholm (SW): Jet Propulsion Laboratory (JPL), NASA. doi:10.2514/6.2012-1267135
- [5] Bhaskaran, S., Desai, S. D., Dumont, P. J., Kennedy, B. M., Null, G. W., Owen, W. J., . . . Werner, R. A. (1998). Orbit Determination Performance Evaluation of the Deep Space 1 Autonomous Navigation System. *Spaceflight Mechanics Meeting*. Monterey, CA (USA). Retrieved from <http://hdl.handle.net/2014/19040>
- [6] Biagiotti, L. (AA 2015-2016). *Stima dello stato in presenza di disturbi: il filtro di Kalman*. Lecture Notes, University of Modena and Reggio Emilia (UNIMORE), Department of Engineering “Enzo Ferrari” (DIEF), Modena (IT). Retrieved from <http://www.dii.unimore.it/~lbiagiotti/MaterialeTSC1516/TSC-03-KalmanFilter.pdf>
- [7] Bills, B. G., Asmar, S. W., Konopliv, A. S., Park, R. S., & Raymond, C. A. (2014, September 15). Harmonic and statistical analyses of the gravity and topography of Vesta. *ICARUS*, 240, 161-173. doi:10.1016/j.icarus.2014.05.033
- [8] Cangahuala, A., Bhaskaran, S., & Owen, W. J. (2012, May 1). Science Benefits of Onboard Spacecraft Navigation. *EOS, Transactions, American Geophysical Union*, 93(18), 177-178. doi:10.1029/2012EO180001

- [9] Castaldi, P. (AA 2015-2016). *Automatic Flight Control*. Lecture Notes, Faculty of Aerospace Engineering, University of Bologna (UNIBO), Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi", Forlì (IT).
- [10] Christian, J. A. (2015). Optical Navigation Using Planet's Centroid and Apparent Diameter in Image. *Journal of Guidance, Control, and Dynamics*, 38(2), 192-204.
doi:<https://doi.org/10.2514/1.G000872>
- [11] DART Team. (2017, August 15). *DART Mission Overview*. NASA, Johns Hopkins Applied Physics Laboratory, University of Colorado Boulder, University of Maryland, Lawrence Livermore National Laboratory, Pasadena, CA (USA). Retrieved from <http://www.lcpm12.org/wp-content/uploads/2017/08/1355-1415-Reed.pdf>
- [12] Dolgos, G., Tortora, P., González, A., Pérez, F., Vázquez, A., Hurtado, D., . . . Crespo, X. (2015, April 7). *DustCube Study Proposal*. Technical Report, University of Vigo (UVIGO), University of Bologna (UNIBO), Micos Engineering.
- [13] European Space Agency (ESA). (2014, September 22). Video on AIDA Mission. *AIDA: Asteroid Impact and Deflection Assessment study*. European Union (EU). Retrieved from <https://www.youtube.com/watch?v=OQ8vp-u-9A4>
- [14] European Space Agency (ESA). (2015, May 21). Video on Asteroid Impact Mission (AIM). *Asteroid Imoact Mission (AIM)*. European Union (EU). Retrieved from <https://www.youtube.com/watch?v=h4lpu8HbpFY>
- [15] European Space Agency (ESA). (2016). *Asteroid Impact and Deflection Assessment (AIDA) mission*. Retrieved from European Space Agency (ESA): https://www.esa.int/Our_Activities/Space_Engineering_Technology/Hera/Asteroid_Impact_Deflection_Assessment_mission
- [16] European Space Agency (ESA). (2016). *Asteroid Impact Mission (AIM)*. Retrieved from European Space Agency (ESA): www.esa.int/Our_Activities/Space_Engineering_Technology/Hera/Asteroid_Impact_Mission2
- [17] European Space Agency (ESA). (2016, February 2). *DustCube*. Retrieved from European Space Agency (ESA): http://www.esa.int/Our_Activities/Space_Engineering_Technology/Hera/DustCube
- [18] Gaskell, R. W., Barnouin-Jha, O. S., & Scheeres, D. J. (2007, March 12-16). Modeling Eros with Stereophotoclinometry. *38th Lunar and Planetary Science Conference*. League City, TX (USA). Retrieved from <https://www.lpi.usra.edu/meetings/lpsc2007/pdf/1333.pdf>

References

- [19] Grewal, M. S., & Andrews, A. P. (2001). *Kalman Filtering: Theory and Practice Using MATLAB* (Second ed.). New York, NY (USA): A Wiley-Interscience Publication, John Wiley & Sons Inc.
- [20] Jet Propulsion Laboratory (JPL), NASA. (n.d.). Retrieved from Navigation and Ancillary Information Facility (NAIF): <https://naif.jpl.nasa.gov/naif/index.html>
- [21] Jet Propulsion Laboratory (JPL), NASA. (2017, April 10). *The SPICE Toolkit*. Retrieved from Navigation and Ancillary Information Facility (NAIF): <https://naif.jpl.nasa.gov/naif/toolkit.html>
- [22] Jet Propulsion Laboratory (JPL), NASA. (2018). *Advanced Horizons Asteroid & Comet SPK File Generation Request*. Retrieved from NASA Jet Propulsion Laboratory - Space Mission and Science News, Videos and Images: <https://ssd.jpl.nasa.gov/x/spk.html>
- [23] Jet Propulsion Laboratory (JPL), NASA. (2018). *HORIZONS Web-Interface*. Retrieved from NASA Jet Propulsion Laboratory - Space Mission and Science News, Videos and Images: <https://ssd.jpl.nasa.gov/horizons.cgi>
- [24] Jet Propulsion Laboratory (JPL), NASA. (2018). *JPL Small-Body Database Browser - 65803 Didymos (1996 GT)*. Retrieved from NASA Jet Propulsion Laboratory - Space Mission and Science News, Videos and Images: <https://ssd.jpl.nasa.gov/sbdb.cgi>
- [25] Jet Propulsion Laboratory (JPL), NASA. (2018, January). *SPICE Tutorial, Navigation and Ancillary Information Facility (NAIF)*. (C. Acton, N. Bachman, M. Liukis, F. Thomson, B. Semenov, & E. Wright, Eds.) Retrieved from Navigation and Ancillary Information Facility (NAIF): <https://naif.jpl.nasa.gov/naif/tutorials.html>
- [26] Jet Propulsion Laboratory (JPL), NASA. (n.d.). *SPICE Required Reading Documents*. Retrieved from Navigation and Ancillary Information Facility (NAIF): ftp://naif.jpl.nasa.gov/pub/naif/toolkit_docs/C/req/index.html
- [27] Keller, H. U., Barbieri, C., Lamy, P., Rickman, H., Rodrigo, R., Wenzel, K.-P., . . . Zaccariotto, M. (2007, February). OSIRIS - The Scientific Camera System Onboard Rosetta. *Space Science Reviews*, 128(1), 433-506. doi:10.1007/s11214-006-9128-4
- [28] Kubitschek, D. G. (2003, August 3-7). Impactor Spacecraft Targeting for the Deep Impact Mission to Comet Tempel 1. *AAS/AIAA Astrodynamics Specialist Conference (AAS-03 615)* (p. 18 pp). Big Sky, MT (USA): American Astronautical Society (AAS); American Institute of Aeronautics and Astronautics (AIAA). Retrieved from <https://trs.jpl.nasa.gov/handle/2014/37300?show=full>

- [29] Kubitschek, D. G. (2005, November 10). Impactor Spacecraft Encounter Sequence Design for the Deep Impact Mission. *Georgia Tech - Space Systems Engineering Conference (GT-SSEC.C.3)* (p. 24 pp). Atlanta, GA (USA): Georgia Tech Library. Retrieved from https://smartech.gatech.edu/bitstream/handle/1853/8031/SSEC_SC3_ppt.pdf
- [30] Kubitschek, D. G., Mastrodemos, N., & Synnott, S. P. (2003, February 5). Deep Impact Spacecraft Flyby Instrument Pointing in the Presence of a Non-active Impactor Spacecraft. In I. J. Gravseth, & R. D. Culp (Ed.), *26th Annual AAS Rocky Mountain Guidance and Control Conference (AAS 03-026)*. 113, p. 16 pp. Breckenridge, CO (USA): Published for the American Astronautical Society (AAS) by Univelt. Retrieved from <http://hdl.handle.net/2014/6309>
- [31] Lasagni Manghi, R., Modenini, D., Zannoni, M., & Tortora, P. (2018, October 15). Preliminary orbital analysis for a CubeSat mission to the Didymos binary asteroid system. *Advances in Space Research*, 62(8), 2290-2305. doi:10.1016/j.asr.2017.12.014
- [32] Lasagni Manghi, R., Zannoni, M., Modenini, D., & Tortora, P. (2016, February 2). *DustCube Navigation Concept*. Technical Report, Faculty of Aerospace Engineering, University of Bologna (UNIBO), Department of Industrial Engineering, Forli (IT).
- [33] Lauer, M., Herfort, U., Hocken, D., & Kielbassa, S. (2009, September 28 - October 2). Optical Measurements for the Flyby Navigation of Rosetta at Asteroid Steins. *21st International Symposium on Space Flight Dynamics (ISSFD - AOCS II) at Centre national d'études spatiales (CNES)*, (p. 15 pp). Toulouse (FR). Retrieved from http://issfd.org/ISSFD_2009/AOCSII/Lauer.pdf
- [34] Lemaitre, G. R. (2013, March). Optical Design and Active Optics Methods in Astronomy. *Optical Review*, 20(2), 103-117. doi:10.1007/s10043-013-0015-4
- [35] Luise, M., & Vitetta, G. M. (2009, June 1). *Teoria dei Segnali* (3rd ed.). Italy: McGraw-Hill Education.
- [36] Mastrodemos, N., Rush, B., Vaughan, D., & Owen, W. M. (2012, October 29 - November 3, October 29). Optical Navigation for the Dawn Mission at Vesta. *23rd International Symposium on Space Flight Dynamics* (p. 26 pp). Pasadena, CA (USA): Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2012. Retrieved from <https://trs.jpl.nasa.gov/handle/2014/43000>
- [37] Montenbruck, O. (2008, September 26). Algorithms for GPS Based Real-Time Navigation. *IGPS Progress Meeting - European Space Research and Technology Centre (ESTEC)*, (p. 45 pp). Noordwijk (NL).

References

- [38] Montenbruck, O., & Gill, E. (2000). *Satellite Orbits* (1st ed.). Springer.
- [39] National Aeronautics and Space Administration (NASA). (2017, June 30). Video on DART Moon Collision. *DART Moon Collision*. United States of America (USA). Retrieved from <https://www.youtube.com/watch?v=8zooPRmgUPI>
- [40] National Aeronautics and Space Administration (NASA). (2018, June 25). *Double Asteroid Redirection Test (DART) Mission*. (B. Dunbar, & T. Talbert, Eds.) Retrieved from National Aeronautics and Space Administration (NASA): <https://www.nasa.gov/planetarydefense/dart>
- [41] Nodar, D., Pérez, F., Vázquez, A., González, A., Tubío, R., Aguado, F., . . . Dolgos, G. (2016, April 21). DustCube, a 3U Cubesat to Characterize the Natural Dust Environment and Microscopic Ejecta due to DART High Speed Impact on the Binary Asteroid 65803 Didymos. 25 pp. Vigo, ES. Retrieved from http://mstl.atl.calpoly.edu/~bklofas/Presentations/DevelopersWorkshop2016/3_DiegoNodar.pdf
- [42] Owen, W. J. (2011, February 14). Methods of Optical Navigation. *AAS Spaceflight Mechanics Conference*. New Orleans, Louisiana: Jet Propulsion Laboratory, NASA., 2011. Retrieved from <https://trs.jpl.nasa.gov/bitstream/handle/2014/41942/11-0589.pdf?sequence=1&isAllowed=y>
- [43] Owen, W. J., Duxbury, T. C., Acton, C. J., Synnott, S. P., Riedel, J. E., & Bhaskaran, S. (2008, February 1-6). A Brief History of Optical Navigation at JPL. In M. E. Drews, & R. D. Culp (Ed.), *31st Annual AAS Guidance and Control Conference*. 131, pp. 329-348. Breckenridge, CO (USA): Published for the American Astronautical Society by Univelt at San Diego, CA (USA). Retrieved from <http://www.univelt.com/book=1625>
- [44] Pérez, F. (2016, January 20). *DustCube Concept of Operations*. Technical Report, University of Vigo (UVIGO), Department of Electrical Engineering and Telecommunications, Vigo (ES).
- [45] Pérez, F., Modenini, D., Vázquez, A., Aguado, F., Tubío, R., Dolgos, G., . . . Carnelli, I. (2018 (In Press)). DustCube, a nanosatellite mission to binary asteroid 65803 Didymos as part of the ESA AIM mission. *Advances in Space Research*. doi:10.1016/j.asr.2018.06.019
- [46] Polle, B., Frapard, B. M., Milic, E., Gil-Fernandez, J., Graziano, M., Rebordão, J., & Motrena, P. (2005, October 17-20). Autonomous Navigation for Interplanetary Missions Performance Achievements Based on Real and Flight Images. *6th International ESA Conference on Guidance, Navigation and Control Systems*. ESA SP-606, January 2006. Loutraki, Greece: D. Danesy, ESA Publications Division.

- [47] Ponzio, F. (2018). *Il filtro di Kalman*. Politecnico di Torino, (POLITO), Automatic and Informatic Department (DAUIN), Torino (IT). Retrieved from https://www.academia.edu/6213290/Il_filtro_di_Kalman
- [48] Quarteroni, A., Sacco, R., Saleri, F., & Gervasio, P. (2014). *Matematica Numerica* (4th ed.). Milano (IT): Springer - UNITEXT. doi:10.1007/978-88-470-5644-2
- [49] Richardson, D. C., Barnouin, O. S., Benner, L. A., Bottke, W. J., Campo Bagatin, A., Cheng, A. F., . . . Zhang, Y. (2016, March 21-25). Dynamical and Physical Properties of 65803 Didymos. *47th Lunar and Planetary Science Conference*. Woodlands, TX (USA). Retrieved from <https://www.hou.usra.edu/meetings/lpsc2016/pdf/1501.pdf>
- [50] Scheirich, P., & Pravec, P. (2009, April). Modeling of Lightcurves of Binary Asteroids. *ICARUS*, 200(2), 531-547. doi:10.1016/j.icarus.2008.12.001
- [51] Shimkin, N. (2009). *Estimation and Identification in Dynamical Systems*. Lecture Notes, Technion - Israel Institute of Technology, Department of Electrical Engineering, Haifa (IL). Retrieved from http://webee.technion.ac.il/people/shimkin/Estimation09/ch8_target.pdf
- [52] Silvestri, D. (2014). *Enhancement of Trajectory Determination of Orbiter Spacecraft by using pairs of Planetary Optical Image*. Dissertation of Doctorate of Philosophy (PhD) in Mechanics and Advanced Engineering Sciences, University of Bologna, Faculty of Aerospace Engineering, Forlì (IT). Retrieved from http://amsdottorato.unibo.it/6290/1/Silvestri_Davide_PhD_Thesis.pdf
- [53] Takahashi, Y. (2013). *Gravity Field Characterization around Small Bodies, of the ,.* Dissertation of Doctorate of Philosophy (PhD), University of Colorado, Department of Aerospace Engineering Sciences - Graduate Thesis and Dissertations. Retrieved from https://scholar.colorado.edu/asen_gradetds/70
- [54] Tortora, P., & Modenini, D. (AA 2014-2015). *Spacecraft Attitude Dynamics and Control*. Lecture Notes, Faculty of Aerospace Engineering, University of Bologna (UNIBO), Department of Industrial Engineering, Forlì (IT).
- [55] Tortora, P., Modenini, D., & Alessi, E. M. (AA 2013-2014). *Spacecraft Orbital Dynamics and Control*. Lecture Notes, Faculty of Aerospace Engineering, University of Bologna (UNIBO), Department of Industrial Engineering, Forlì (IT).
- [56] Vallado, D. A., & McClain, W. D. (2007, May 5). *Fundamentals of Astrodynamics and Applications* (3rd ed.). Space Technology Library (STL), Microcosm Press, Springer. Retrieved from <https://celestrak.com/software/vallado-sw.php>

References

- [57] van der Hilst, R., Frederik, S. J., & Zuber, M. T. (n.d.). The Earth's Gravitational Field. 26-64. Boston, Massachusetts, USA. Retrieved from http://www-gpsg.mit.edu/12.201_12.501/BOOK/chapter2.pdf
- [58] Welch, G., & Bishop, G. (1995). *An Introduction to the Kalman Filter*. Technical Report, University of North Carolina (UNC) at Chapel Hill, Department of Computer Science, Chapel Hill, NC (USA). Retrieved from https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf
- [59] Yu, Y., Michel, P., Schwartz, S. R., Naidu, S. P., & Benner, L. A. (2017, January 15). Ejecta Cloud from the AIDA Space Project Kinetic Impact on the Secondary of a Binary Asteroid: I. Mechanical Environment and Dynamic Model. *ICARUS*, 282, 313-325. doi:10.1016/j.icarus.2016.09.008
- [60] Zacharias, N., Finch, C. T., Girard, T. M., Henden, A., Bartlett, J. L., Monet, D. G., & Zacharias, M. I. (2013, February). The Fourth US Naval Observatory CCD Astrograph Catalog (UCAC4). *The Astronomical Journal*, 145:45(2), 14 pp. doi:10.1088/0004-6256/145/2/44
- [61] Zannoni, M., Tommei, G., Modenini, D., Tortora, P., Mackenzie, R., Scoubeau, M., . . . Carnelli, I. (2018, October 15). Radio science investigations with the Asteroid impact mission. *Advances in Space Research*, 62(8), 2273-2289. doi:10.1016/j.asr.2017.12.003
- [62] Zhang, R., Tu, R., Zhang, P., Liu, J., & Lu, X. (2018 (In Press)). Study of satellite shadow function model considering the overlapping parts of Earth shadow and Moon shadow and its application to GPS satellite orbit determination. *Advances in Space Research*. doi:10.1016/j.asr.2018.02.002