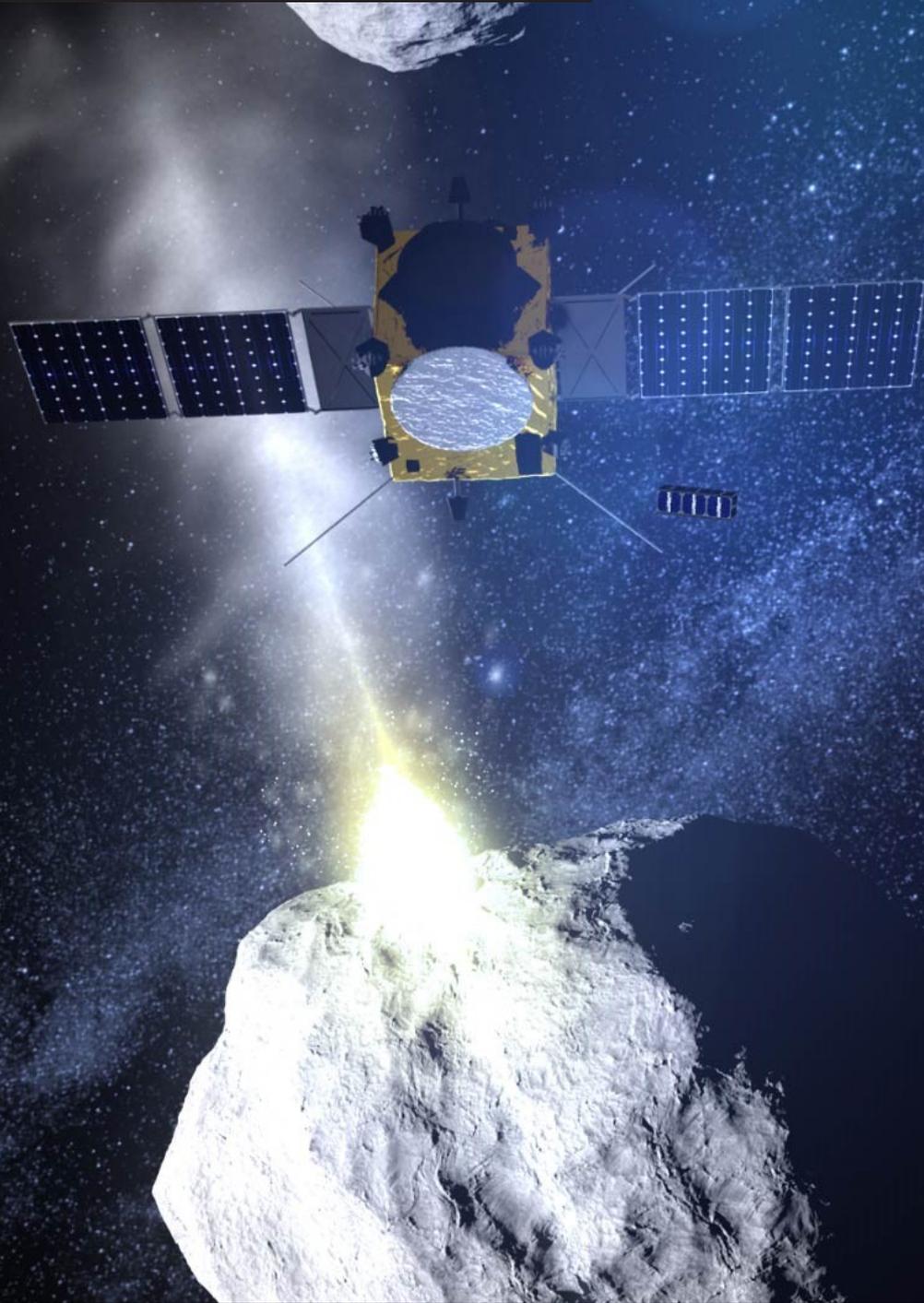


# Cubesat Deployment Trajectories for the Asteroid Impact Mission

MSc Thesis

P. Minguijon Pallas

Technische Universiteit Delft



Cover image credit: European Space Agency

# CUBESAT DEPLOYMENT TRAJECTORIES FOR THE ASTEROID IMPACT MISSION

MSC THESIS

by

**P. Minguijon Pallas**

in partial fulfillment of the requirements for the degree of

**Master of Science**  
in Aerospace Engineering

at Delft University of Technology,  
to be defended publicly on Friday February 19, 2018 at 09:30 AM.

Supervisor: Dr. ir. E. Mooij  
Ir. R. Noomen  
Thesis committee: Prof. dr. ir. P.N.A.M. Visser, Chair  
Dr. ir. H.G. Visser, External Examiner

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



# PREFACE

This thesis closes one chapter of the book of my life. It has been an intense and fruitful chapter, and I would like to dedicate words to three aspects of it:

As the student I still am, and the eager learner I will hopefully always be, I would like to firstly thank all the professors whose lectures I have attended throughout my life. You all tried to transfer some of your knowledge to me, and there is a deep beauty on that. In particular, I would like to thank Ron and Erwin for the helpful and honest feedback they have given me during the whole thesis. Your lessons will greatly help me during my entire life and career.

As the professional in which I have become after completing this thesis, I would like to greatly thank the support received by ESA, and specifically by Massimo Casasco. You have always welcomed my questions and suggestions, and have guided me through a not-so-easy project. I have definitely learnt a lot from your extremely professional yet relaxed way of approaching challenging engineering problems. Your experienced inputs have greatly helped me in this task and hopefully will continue to do so in my future ones. Still from a professional perspective I would also like to show my gratitude to GMV, company that kindly let me use some of their documents and figures on the mission at hand, and to M. Schlueter, who kindly granted me access to the full version of his commercial optimisation tool, MIDACO.

From a personal perspective, I would like to show my gratitude to TU Delft and the Netherlands for welcoming me as they have, as if there were no barriers and no frontiers. The real meaning and values of the European Union are reflected in these policies.

My gratitude goes to:

To all the people I have met during this chapter, it has been a pleasure getting to know all of you.

To the unexpected ones.

To my room-mates, because "that's what she said".

To the ones that started this chapter by my side and are no longer part of it, thank you for everything, you will never be forgotten. Never.

To my life-long buddies, you know I love you all, my weird guys.

To my dad, because you have taught me things you never really tried to teach me, and that's a valuable lesson itself.

To the best, my mum. There are no words to describe my gratitude for what you have done for me. I am here because of you. Gracias.

A todos y cada uno de vosotros,  
Gracias.

*P. Minguíjon Pallas  
Delft, January 2018*



# ABSTRACT

Missions to Near-Earth Objects (NEOs) are a growing trend in spaceflight. The interest for such bodies is not only justified by the threat that they represent for the Earth, but also because they are potential sources of extra-terrestrial materials, provide windows to the past of the Solar System and to planetary formation, and offer a good opportunity to demonstrate deep-space mission technologies.

The Asteroid Impact Mission (AIM) is the next European Space Agency mission to a NEO, which has among its main objectives the deployment of the Asteroid Spectral Imaging Mission Cubesat (ASPECT) in the vicinity of the binary asteroid Didymos during the Payload Delivery Phase (PDP). However, this deployment is affected by the uncertainties and errors coming from the strongly perturbed environment, the previous-phase dispersions, and the navigation and command inaccuracies.

The objective of this thesis is therefore to assess the viability of the cubesat deployment in terms of the safety of the reference trajectories and the accuracy of the deployment conditions achieved.

The reference trajectories that target the cubesat's commissioning orbit injection point were designed by implementing a two-arc hyperbolic trajectory design problem in a direct-shooting optimisation architecture in the commercial heuristics optimisation software known as Mixed Integer Distributed Ant Colony Optimization (MIDACO). These trajectories were obtained in an ideal, unperturbed scenario. The safety and accuracy of the trajectories obtained were tested in a series of Monte Carlo campaigns in which the uncertainties, errors and dispersions present in the system were used as Monte Carlo variables in a high-level Guidance, Navigation and Control (GNC) simulator implemented in Simulink 2016b.

After identifying the pericentre of the target Self-Stabilised Terminator Orbit (SSTO) as the best point to carry out the deployment, five different families of trajectories that bring the spacecraft within the safety deployment margins were found. The total velocity increment required for these trajectories were found to vary from 1.2 to 3.4 m/s. These relatively high velocity increments are caused by the combination of the constraints imposed in the design process that emanate from the mission operational requirements.

The first Monte Carlo set of simulations showed that the optimum number of guidance correction manoeuvres is assessed to be three. The timing of these manoeuvres is case dependent, but one correction must take place in the first arc flown, and the other two during the second arc and before the deployment.

Results of the second Monte Carlo campaign showed that the trajectories were unsafe according to the mission safety requirements on the minimum distance to the main asteroid and on the margin to the escape velocity. Of these two requirements, the escape velocity margin was the main cause of these safety violations, with several trajectories falling below the margin of 2 cm/s at certain epochs. This was due to the initial state vector dispersion coming from the previous phase and, to a lesser extent, due to the potential thrust failures.

In the final Monte Carlo campaign, the ASPECT injection success rates reached a maximum value of 75% for the trajectory that targets the SSTO with a semi-major axis equal to 5 km. The cause of this low success rate was attributed to a combination of high initial position dispersions and velocity navigation errors. Due to this velocity navigation inaccuracies, the corrections made by the guidance subsystem are not capable of compensating the dispersions in initial position. These corrections further deteriorate the final deployment conditions, since their calculation only takes into account deviations in position, and not in velocity.

The results of this study highlight the need of finding ways of reducing the position dispersion with respect to the nominal trajectory prior to the autonomous PDP. Furthermore, they indicate that the prediction of success rates can be improved by refining the navigation performance models, since these are considered conservative when compared to the real scenario.



# CONTENTS

<b>Glossary</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Asteroid Impact Mission</b>	<b>3</b>
2.1 AIM Phases and Trajectories . . . . .	4
2.2 Mission Modelling Assumptions and Requirements . . . . .	5
<b>3 Spacecraft Dynamics</b>	<b>9</b>
3.1 Binary Asteroid System 65803 Didymos . . . . .	9
3.2 Reference Frames . . . . .	12
3.3 Dynamics Model . . . . .	13
3.3.1 Two-Body Problem. . . . .	13
3.3.2 Third-Body Perturbation. . . . .	13
3.3.3 Solar Radiation Pressure . . . . .	14
3.3.4 Non-Sphericity Perturbation. . . . .	14
<b>4 Reference Trajectory and Cubesat Deployment Design</b>	<b>17</b>
4.1 Trajectory Design . . . . .	17
4.2 Cubesat Deployment and Injection analysis . . . . .	19
<b>5 Guidance and Navigation</b>	<b>23</b>
5.1 Guidance . . . . .	23
5.2 Navigation . . . . .	26
<b>6 Numerical Techniques</b>	<b>31</b>
6.1 Optimisation . . . . .	31
6.2 Integration . . . . .	34
6.3 Interpolation . . . . .	35
<b>7 Software</b>	<b>37</b>
7.1 Architectural Design . . . . .	37
7.1.1 Dynamics . . . . .	38
7.1.2 Optimisation with MIDACO . . . . .	39
7.1.3 Guidance Navigation and Control . . . . .	43
7.1.4 Monte Carlo Scheduler. . . . .	47
7.2 Verification and Validation . . . . .	49
7.2.1 Verification . . . . .	50
7.2.2 System Test . . . . .	55
<b>8 Results</b>	<b>57</b>
8.1 Simulation Plan . . . . .	57
8.2 Reference Trajectory Design and Optimisation . . . . .	58
8.3 Stochastic Guidance Tuning . . . . .	64
8.3.1 Number of stochastic corrections . . . . .	64
8.3.2 Stochastic Firing Times . . . . .	68
8.4 Mission Safety Assessment . . . . .	72
8.5 Deployment Success Assessment . . . . .	75
8.5.1 Uncertainties . . . . .	76
8.5.2 Command Errors . . . . .	78
8.5.3 Initial Dispersions . . . . .	79

---

<b>9</b>	<b>Conclusions and Recommendations</b>	<b>85</b>
9.1	Conclusions . . . . .	85
9.2	Recommendations . . . . .	86
<b>A</b>	<b>Reference Frames Transformations</b>	<b>89</b>
	<b>Bibliography</b>	<b>91</b>

# GLOSSARY

## LIST OF ABBREVIATIONS

<b>ABM</b>	Adams-Bashforth-Moulton
<b>ACO</b>	Ant Colony Optimisation
<b>AIDA</b>	Asteroid Impact & Deflection Assessment
<b>AIM</b>	Asteroid Impact Mission
<b>ASPECT</b>	Asteroid Spectral Imaging Mission Cubesat
<b>AU</b>	Astronomical Unit
<b>CRS</b>	Cubesat Release System
<b>DART</b>	Double Asteroid Redirect Test
<b>DCP</b>	Detailed Characterisation Phase
<b>DIP</b>	Dart Impact Phase
<b>DLR</b>	Deutsches Zentrum für Luft- und Raumfahrt
<b>DOF</b>	Degrees Of Freedom
<b>DOR</b>	Differential One-way Ranging
<b>ECP</b>	Early Characterisation Phase
<b>ESA</b>	European Space Agency
<b>ET</b>	Ephemeris Time
<b>FDIR</b>	Failure Detection, Isolation and Recovery
<b>FOV</b>	Field Of View
<b>GMAT</b>	General Mission Analysis Tool
<b>GNC</b>	Guidance, Navigation and Control
<b>ICP</b>	Initial Characterisation Phase
<b>IMU</b>	Inertial Measurement Unit
<b>ISL</b>	Inter-Satellite Link
<b>JHU/APL</b>	John Hopkins University Applied Physics Laboratory
<b>LOS</b>	Line Of Sight
<b>LVLH</b>	Local-Vertical Local-Horizontal
<b>MASCOT-2</b>	Mobile Asteroid Surface Scout 2
<b>MIDACO</b>	Mixed Integer Distributed Ant Colony Optimization
<b>NAIF</b>	Navigation and Ancillary Information Facility

<b>NASA</b>	National Aeronautics and Space Administration
<b>NEA</b>	Near-Earth Asteroid
<b>NEO</b>	Near-Earth Object
<b>NSP</b>	Non-Sphericity Perturbation
<b>ODE</b>	Ordinary Differential Equation
<b>OPM</b>	Oracle Penalty Method
<b>PaGMO</b>	Parallel Global Multiobjective Optimizer
<b>PDF</b>	Probability Density Function
<b>PDP</b>	Payload Delivery Phase
<b>PECE</b>	Predict-Evaluate-Correct-Evaluate
<b>PIG</b>	Propulsive Impulsive Guidance
<b>RAAN</b>	Right Ascension of the Ascending Node
<b>RH4BP</b>	Restricted Hill 4-Body Problem
<b>RK</b>	Runge-Kutta
<b>SRP</b>	Solar Radiation Pressure
<b>SSTO</b>	Self-Stabilised Terminator Orbit
<b>STM</b>	State-Transition Matrix
<b>TAP</b>	Transition to Autonomous Phase
<b>TBP</b>	Third-Body Perturbation
<b>UTC</b>	Coordinated Universal Time
<b>V&amp;V</b>	Verification and Validation
<b>2BP</b>	Two-Body Problem

## LIST OF SYMBOLS

### LATIN SYMBOLS

<i>a</i>	Semi-major axis [m]
$C_R$	Reflection coefficient [-]
<i>d</i>	Distance [m]
<i>e</i>	Eccentricity [-]
<i>i</i>	Inclination [rad]
<i>m</i>	Mass [kg]
<b>O</b>	Output matrix [various]
<b>P</b>	Parameter matrix [various]
<b>r</b>	Position vector [m]
<i>s</i>	Cross-sectional area [m <sup>2</sup> ]
<i>t</i>	Time [s]
<i>T</i>	Period [s]
<b>v</b>	Velocity vector [m/s]
<b>x</b>	State vector [m; m/s]

## GREEK SYMBOLS

$\alpha$	Phase angle [rad]
$\beta$	Spin angle [rad]
$\epsilon$	Mass ratio [-]
$\phi$	Latitude [rad]
$\Phi$	State Transition matrix [various]
$\gamma$	Command error angle [rad]
$\kappa$	Knowledge factor [-]
$\lambda$	Longitude [rad]
$\mu$	Gravitational parameter [ $\text{m}^3/\text{s}^2$ ]
$\theta$	True anomaly [rad]
$\rho$	Correlation factor [-]
$\sigma$	Standard deviation [various]

## NOTATION

$\hat{\phantom{x}}$	Unitary vector
$\tilde{\phantom{x}}$	Estimated scalar or vector
$\bar{\phantom{x}}$	Normalised scalar
$\dot{\phantom{x}}$	Time derivative of vector or matrix
Subscript	
$(\ )_Q$	Vector expressed in reference frame [Q]
${}_{Body_1}Body_2$	Vector describing the motion of Body2 with respect to Body1
Superscript	
$^T$	Transpose of vector or matrix

## CONSTANTS

$AU$	Astronomical Unit, $AU = 149597870700 \text{ m}$ (Wakker, 2015)
$G$	Universal Gravitational Constant, $G = 6.67428 \cdot 10^{-11} \text{ m}^3/\text{kg s}^2$ (Wakker, 2015)
$G_1$	Solar Radiation Constant, $G_1 = 10^{17} \text{ kg m/s}^2$ (Scheeres and McMahon, 2013)
$\mu_{Sun}$	Sun Gravitational Parameter, $\mu_{Sun} = 1.327124421 \cdot 10^{20} \text{ m}^3/\text{s}^2$ (Wakker, 2015)



# 1

## INTRODUCTION

Potential impacts of Near-Earth Objects (NEOs) are especially relevant for a planet like Earth, where a fragile ecosystem survives by adaptation to slowly changing environmental conditions. If these conditions suffer large, sudden changes, this equilibrium is likely to break and to produce major extinctions as those suffered in the past. The best known of these events is the Cretaceous–Paleogene extinction one, which ended the reign of the non-avian dinosaurs and many other species, and was caused by the impact of a 10 to 15 km wide asteroid in the Gulf of Mexico (Schulte et al., 2010).

This is a threat that mankind has always faced. However, now we are potentially able to actively reduce this risk. This is reflected in a growing global trend among the space agencies to dedicate entire missions to visit such small bodies, with some of these missions being asteroid-deflection experiments.

However, this is not the only reason to explore NEOs, their characterisation presents great opportunities in many fields (Scheeres, 2012). They are potential sources of extra-terrestrial resources, which can eventually make asteroid mining profitable. Besides, NEOs provide windows to the history of the Solar System and to different stages of the planetary formation, what makes their exploration scientifically valuable. Also to be considered is the fact that the close-proximity operations of missions to these systems offer a perfect opportunity to demonstrate technologies for future deep-space missions. The Asteroid Impact Mission (AIM) is born in this context. In particular, this European Space Agency (ESA) mission will explore the binary asteroid system Didymos, where it will pursue different objectives throughout the phases of the mission. These will include: cubesat deployment, detailed characterisation of the asteroid system and the assessment of the deflection caused by the National Aeronautics and Space Administration (NASA) part of the mission, the Double Asteroid Redirect Test (DART), that will impact on the system a few months after the arrival of AIM.

From an engineering point of view, particularly interesting are the technological challenges that the close-proximity operations of a mission like AIM implies. These are driven by two factors: the characteristics of the environment in which these activities take place and the limitations of the on-board subsystems.

The surroundings of a binary irregularly-shaped asteroid system, where a weak non-uniform gravity field couples with the perturbations from the Sun are considered as an uncertain and strongly perturbed environment. In such a system, a spacecraft in close orbit around the asteroid can develop trajectory instabilities within hours and end up crashing into or escaping the system (Dell’Elce et al., 2016). The inherent inaccuracies and constraints of the guidance and navigation subsystems are another source of uncertainty and errors. For the autonomous navigation subsystem, aspects such as the illumination conditions of the asteroid, the number of surface features that are seen at once, and the inherent limitation in accuracy of the navigation sensors will drive the errors yielded by the chosen navigation filters. On the other hand, the type of scheme used for the guidance subsystem will not only determine its performance but also the spacecraft operational baseline.

The deployment of the Asteroid Spectral Imaging Mission Cubesat (ASPECT) will take place in this context. This thesis focuses on the study of the translational accuracy that is achievable at the epoch at which the deployment of the cubesat takes place for a controlled spacecraft autonomously moving in this environment. Controlled hyperbolic trajectories emanating from the previous phase of the mission and targeting the cubesat deployment position will be analysed, and the conditions found at the deployment epoch will be used to check the feasibility of the cubesat deployment under the mission conditions.

The key research question of this thesis is:

*How can we design a transfer trajectory between two points in the close-proximity operations range of a binary asteroid while satisfying payload injection accuracy and mission safety requirements?*

with the following sub-questions:

- *What is the nominal trajectory that best fits the **AIM** requirements for the deployment of **ASPECT**?*
- *What are the guidance firing conditions that lead to the best deployment conditions?*
- *How do uncertainties in the strongly perturbed environment affect the deployment conditions?*
- *How do guidance command errors affect the deployment conditions?*
- *How do uncertainties and dispersions in the spacecraft state vector affect the deployment conditions?*

The thesis report is structured as follows: Chapter 2 presents **AIM**, its goals, phases, requirements and the mission modelling assumptions made. Chapter 3 covers the dynamics of a spacecraft in the vicinity of a binary asteroid. The governing equations for translational motion are presented, which will be later used for the development of the software tool. Chapter 4 covers the requirements that apply to the design of the reference trajectory and the cubesat deployment strategy. Chapter 5 covers the guidance scheme and the navigation model that will be used in the thesis. In Chapter 6, the optimisation, ordinary differential equations integration and interpolation techniques used during the thesis are described. Chapter 7 covers the software tool design, its validation and verification and the system test carried out. In Chapter 8, the results obtained for the different simulations are presented and explained. Chapter 9 presents the main conclusions drawn from the work carried out, and some recommendations for future work.

# 2

## ASTEROID IMPACT MISSION

With its launch planned for 2020, AIM will be the first ESA mission to a small body since 2004, in which Rosetta was launched. It is part of the joint Asteroid Impact & Deflection Assessment (AIDA) project of ESA, the Deutsches Zentrum für Luft- und Raumfahrt (DLR), the Observatoire de la Côte d'Azur, NASA and the John Hopkins University Applied Physics Laboratory (JHU/APL).

The objective of AIDA is to send two spacecraft to the binary asteroid 65803 Didymos:

- An asteroid impactor – the NASA DART mission.
- An asteroid rendezvous and observation spacecraft – the ESA AIM mission.

The AIM part of the mission will involve the determination of the orbital and rotation dynamics, together with the physical properties of the system using the available payload instruments. It will observe the impact of DART and characterise the change in the orbital and rotational state of the system as well as the impact crater and ejecta plume generated by the impact.

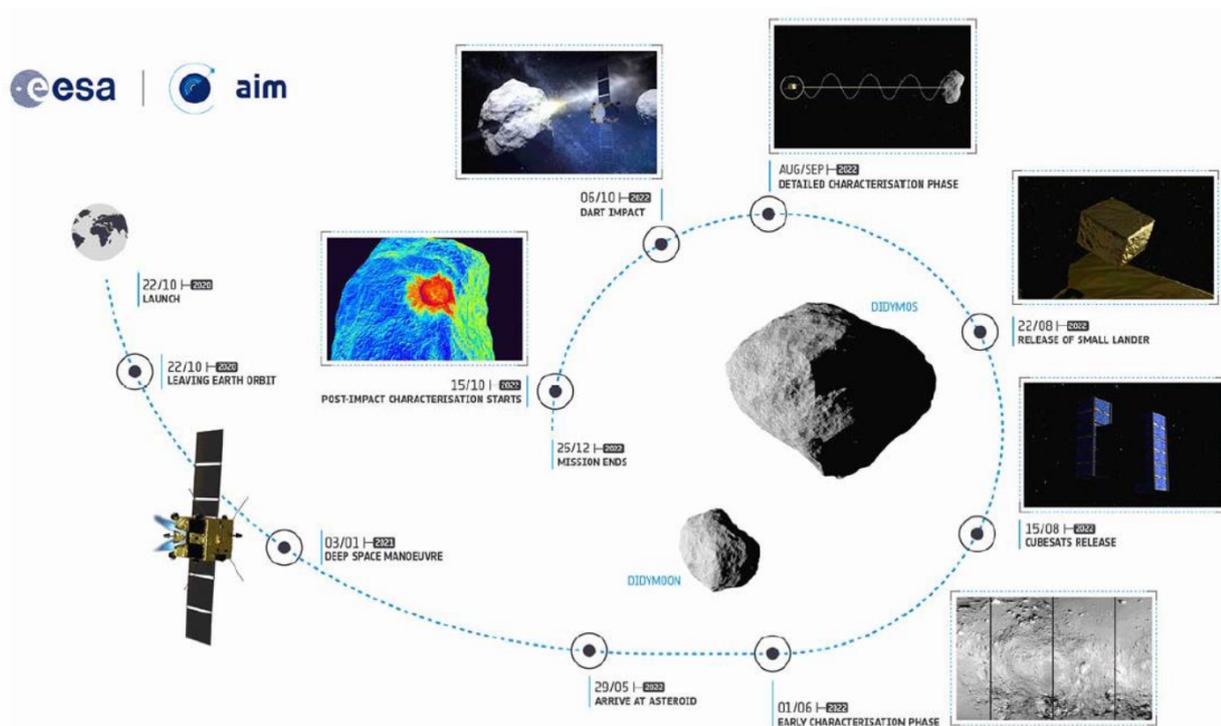


Figure 2.1: AIM phases from launch to end of mission (GMV, 2017).

It will also feature technology demonstration goals (ESA-CDF, 2014; ESA-ESTEC-AD1, 2016) such as a demonstration of laser communication in deep space and deployment of cubesats in deep space while keeping an Inter-Satellite Link (ISL) between them and the mothercraft.

The propulsion subsystem is inherited from the Rosetta mission and consists of two sets of 12 thrusters with a nominal force of 20 N each. These provide a six Degrees Of Freedom (DOF) control authority and are meant to control both the attitude and trajectory of the system. The spacecraft launch mass is 740 kg, including 306 kg of propellant (ESA-ESTEC-RD27, 2016).

The different phases and previously-described goals of the mission can be seen in Figure 2.1. It must be noted that the Mobile Asteroid Surface Scout 2 (MASCOT-2) ballistic lander deployment is no longer part of the mission as a consequence of the mission budget reduction after the ministerial meeting held in December 2016. Besides, in the new operational time line, there is a Detailed Characterisation Phase (DCP) previous to the cubesat deployment and after the Early Characterisation Phase (ECP). These phases and the trajectories flown during them will be further explained in Section 2.1. This thesis will focus on the trajectory that will be flown after the first DCP and will enable the cubesat release milestone shown in Figure 2.1.

It must be noted that since during the thesis only the AIM architecture and spacecraft subsystems will be modelled, there is no need to systematically cover the different approaches followed in other missions to small bodies. The corresponding aspects of AIM will be thoroughly analysed in the following sections and chapters, and comparisons to other missions' approaches will be made when necessary.

## 2.1. AIM PHASES AND TRAJECTORIES

As seen in Figure 2.1, AIM is scheduled to be launched in October 2020. After two years of deep space travel, the spacecraft will arrive at the asteroid system on April 22, 2022. The beginning of the ECP at 35 km altitude is meant to start one month later, in May 2022. The orbits flown during this stage of the mission will be such that it allows far observations of the illuminated side of the asteroid, but avoiding the sub-solar point, where the geometry of the illumination conditions would eliminate the presence of shadows.

On June 16, 2022, the spacecraft will move to the next phase, the DCP. This phase will be divided into two cycles of two weeks, and the spacecraft will fly hyperbolic trajectories at distances of around 10 km from the asteroid system. During the first cycle of this phase, the knowledge on some of the parameters that affect the motion of the spacecraft will be improved. This fact will be considered during the thesis and will be explained in Chapter 3. The trajectory flown during this phase can be seen in Figure 2.2.

As part of the MASCOT-2 deployment strategy, a previous phase that increased the safety and success expectations of the ballistic lander injection was required, the Transition to Autonomous Phase (TAP). These two objectives were addressed by designing this phase to reduce the high dispersions that are present during the whole DCP as much as possible. This TAP also provided the perfect scenario for the navigation to be handover from the ground segment to the on-board autonomous navigation subsystem. However, and as already mentioned, MASCOT-2 is no longer part of the mission baseline, which indicates the need to assess

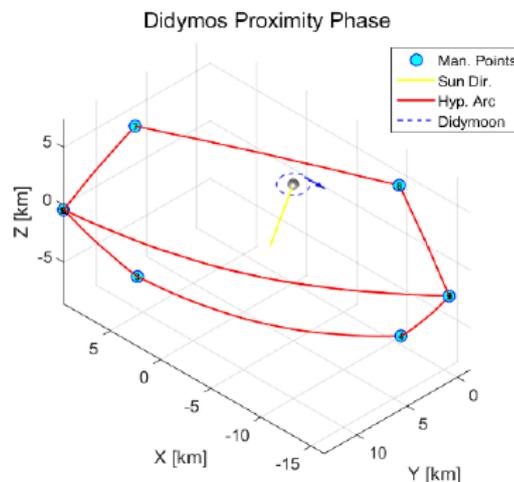


Figure 2.2: AIM DCP trajectory (GMV, 2017).

if the **TAP** is still required or not for the cubesat deployment scenario, which is less demanding in terms of accuracy and required distance to the asteroid bodies.

For this thesis, an scenario without **TAP** will be assumed. This will serve to evaluate the viability of the **ASPECT** deployment without such a phase. In this scenario, the **PDP** begins directly after the second cycle of the first **DCP**. During this second cycle, the handover to the autonomous navigation subsystem is assumed to happen before the injection into the **PDP** trajectory takes place.

During the **PDP**, the **ASPECT** cubesat will be deployed into its commissioning orbit. The **AIM** spacecraft will be injected into a hyperbolic trajectory from one of the points along the **DCP** trajectories. When the mothercraft reaches the target deployment point, the cubesat will be released and injected into its commissioning orbit. From this injection point the main spacecraft will continue, now without the cubesat since it has already been released, until reaching again a distance of around 10 km, where it will be injected into the next phase of the mission, ending the **PDP** (ESA-ESTEC-RD27, 2016).

From this phase onwards, the mission is under redesign due to the major changes introduced after the ministerial conference held in December 2016. The budget reduction strongly affected the payload delivered, which in return drastically changed the concept of operations after the **PDP**. However, a Dart Impact Phase (**DIP**) in which the spacecraft will retreat to a distance of around 100 km to safely record the **DART** impact followed by another **DCP** at a closer distance to measure changes due to the impact are still expected in the mission.

The thesis project is a high-level simulation of a transfer trajectory of the **AIM** spacecraft in the close-proximity operational phase of the mission. It will cover from the second cycle of the first **DCP** to the end of the **PDP** and will assess the deployment conditions and requirements for the **ASPECT** cubesat in its commissioning orbit. This deployment will take place in the middle of the **PDP**. The nominal trajectories that best satisfy the mission requirements will be designed. These trajectories will have their initial conditions along the **DCP** arcs shown in Figure 2.2. From the extremely perturbed and uncertain environment and from the dispersions coming from the initial condition of these trajectories, it emanates the need for guidance corrective manoeuvres, which will make sure the spacecraft follows the nominal arcs with an accuracy such that it enables the cubesat deployment. The sensitivity of the conditions at the cubesat deployment epoch will be analysed with respect to the different uncertainties, errors and dispersions that dominate the system, and the success rate of the cubesat deployment scenario will be determined.

Note that the strongly perturbed environment that dominates the dynamics of the spacecraft will be covered in Chapter 3. The **PDP** hyperbolic trajectory constraints and requirements will be extensively treated in Section 4.1. A detailed description of the target injection point and the commissioning orbit of the **ASPECT** cubesat and the deployment manoeuvre will be given in Section 4.2. The guidance and navigation models used will be described in Chapter 5. The sensitivity analysis implementation will be covered in Section 7.1.4.

## 2.2. MISSION MODELLING ASSUMPTIONS AND REQUIREMENTS

In this subsection, the assumptions strictly related to **AIM** will be given. Note that the rationale for lower-level assumptions of each subsystem will be given in their respective chapters.

To determine if the attitude of the vehicle is relevant for the scenario at hand, three different aspects are assessed: the coupling of the attitude with the command strategy, with the environment forces and with the cubesat deployment strategy. Since the spacecraft has a six-**DOF** thruster-control authority, the attitude of the vehicle can be treated as decoupled from its translational motion for the sake of a simplified thrusting analysis. The translational motion effect of the Solar Radiation Pressure (**SRP**) has a strong dependency with the attitude of the vehicle. However, the attitude of the spacecraft and solar arrays will be maintained relatively constant with respect to the Sun direction due to power budget requirements, which allows to neglect this coupling. The attitude of the vehicle can also be decoupled from the payload deployment strategy, since at this stage of the design the location of the Cubesat Release System (**CRS**) is not yet defined. For these three reasons, only the translational motion of a point-mass spacecraft will be studied.

Also coming from the six-**DOF** thruster-control authority, it is assumed that there are no delays between commanded and executed thruster actions. Considering that  $\Delta\mathbf{v}$ 's are in the order of 1 m/s (GMV, 2017), and assuming that at least three thrusters (i.e., 60 N) can contribute to the translational motion at any time, the impulse times for a spacecraft of 500 kg are lower than 10 seconds. Maximum velocities for the trajectories relative to the asteroid system will, on average, be lower than 1 m/s (GMV, 2017), which yields that a 10 seconds thrust would induce errors lower than 10 m. Taking into account that the cubesat deployment conditions safety margins are in the order of hundreds of meters, all manoeuvres will be considered impulsive.

Table 2.1: Spacecraft modelling assumptions for the PDP.

Symbol	Modelling aspect	Value
$m_{SC}$	Spacecraft mass	$500 \pm 50$ kg
$C_R$	Spacecraft reflection coefficient	1-2
$s_{SC}$	Spacecraft cross-sectional area	$5 \text{ m}^2$
$m_{cube}$	Cubesat mass	4.5 kg
$C_{R,cube}$	Cubesat reflection coefficient	1.5
$s_{cube}$	Cubesat cross-sectional area	$0.045 \text{ m}^2$
$ \Delta \mathbf{v}_{CommErr} $	Command error magnitude [3- $\sigma$ ]	5 %
$\Delta \gamma_{CommErr}$	Command misalignment angle [3- $\sigma$ ]	1 deg
$g$	Guidance scheme	Propulsive Impulsive Guidance (PIG), see Section 5.1
$Nav$	Navigation model	Performance error model, see Section 5.2

This will greatly simplify the modelling, while not compromising the consistency of the analysis.

The errors associated to the thrusters actuation, the control loop and the attitude knowledge will be given a simplified modelling, but still relevant and coherent with the usual treatment given in industry to them. A summary of the parameters that define this modelling can be found in Table 2.1, where the parameters  $|\Delta \mathbf{v}_{CommErr}|$  and  $\Delta \gamma_{CommErr}$  define an error cone around the nominal command. Note that the implementation of this modelling is explained in Section 7.1.

The robustness of the translational guidance algorithm to the mission environment must be assessed. For this reason, a full implementation of the PIG scheme that AIM will use will be given in the project. For the navigation of the spacecraft, performance models will be used to recreate the high-level behaviour of the subsystem that will provide an acceptable level of accuracy in the prediction of the magnitude of these errors but will keep the implementation simple. Note that the guidance and navigation modelling will be explained in Chapter 5.

A description of the mission parameters presented in Table 2.1 is given below.

- $m_{SC}$ : The mass of the spacecraft at the beginning of the PDP and the margins associated to them as reported by GMV (2017).
- $C_R$ : The reflection coefficient indicates the ratio of incident radiation that the spacecraft reflects, it can take any value from 1 to 2 due to the ongoing design process and to account for possible surface material degradation during the mission.
- $s_{SC}$ : The cross-sectional area of the spacecraft related to solar radiation (GMV, 2017).
- $m_{cube}$ : The nominal mass of the ASPECT cubesat (Näsilä et al., 2016).
- $C_{R,cube}$ : The reflection coefficient indicates the ratio of incident radiation that the cubesat reflects (Näsilä et al., 2016).
- $s_{cube}$ : The cross-sectional area of the cubesat related to solar radiation (Näsilä et al., 2016).
- $|\Delta \mathbf{v}_{CommErr}|$ : This parameter indicates the size of the deviation from the nominal command magnitude caused by the thrusters actuation, control loop and the attitude knowledge.
- $\Delta \gamma_{CommErr}$ : This parameter indicates the size of the deviation from the nominal command direction caused by the thrusters actuation, control loop and the attitude knowledge.

The requirements that apply to the PDP of the mission and to the task at hand are (ESA-ESTEC-AD1, 2016):

- **SC-SAFE-1**: All trajectory arcs shall be hyperbolic arcs that do not enter close orbit around Didymain. The velocity of the spacecraft shall always be 0.02 m/s over the escape velocity of the system.

- **SC-SAFE-2:** All trajectories shall be in a collision-free course at all times. A minimum distance of 2000 metres to Didymain shall be guaranteed.
- **SC-SAFE-3:** Requirements **SC-SAFE-1** and **SC-SAFE-2** shall be fulfilled even in case of a thruster failure at any manoeuvre.
- **SC-AUT-1:** The **AIM** spacecraft shall support autonomous operations for 72 hours without ground intervention.
- **SC-AUT-2.1:** The autonomous operations of the **PDP** shall be contained within a 10 km radius sphere around the Didymos system.
- **SC-AUT-2.2:** The **PDP** shall finish at a 10 km distance from the Didymos system, so the next **DCP-2** phase can begin.
- **SC-AUT-3:** Time-tagged commands shall be separated by at least 1800 seconds during autonomous operations. This number accounts for the time it takes for the navigation subsystem to reconstruct the attitude and position after a manoeuvre, and a 100 % margin on top of that (900 + 900 s).
- **SC-PAY-1:** The **ASPECT** cubesat shall be injected into a commissioning orbit that is passively stable for at least 30 days.

Note that the effect and implementation of these requirements will be detailed in the following chapters.



# 3

## SPACECRAFT DYNAMICS

In this chapter, the asteroid that the mission targets will be characterised in Section 3.1, the reference frames used in the thesis will be defined in Section 3.2 and the equations that describe the spacecraft dynamics in the binary asteroid environment will be presented in Section 3.3.

### 3.1. BINARY ASTEROID SYSTEM 65803 DIDYMOS

The target NEO is the binary asteroid system 65803 Didymos. Its orbit around the Sun has a semi-major axis of  $a_{Didymos} = 1.645$  Astronomical Unit (AU), an eccentricity of  $e_{Didymos} = 0.384$  and an inclination of  $i_{Didymos} = 3^\circ$  with respect to the ecliptic, which makes it a potentially hazardous asteroid of both the Apollo and Amor groups (Scheeres, 2012). The two bodies of the system will be referred to as Didymain or Main, for the most massive one, and Didymoon or Moon for the smaller one.

The model that will be implemented for the Didymos system will follow a simplified Restricted Hill 4-Body Problem (RH4BP) formulation, introduced by Bellerose and Scheeres (2005) for this kind of asteroid. Many authors have already used this formulation for this type of system, and even for this particular binary asteroid as in the work by Dell'Elce et al. (2016) and Ferrari et al. (2016). This specific dynamics model can be summarised as:

- The motion of the asteroid system around the Sun is assumed to obey a Two-Body Problem (2BP) formulation. This is consistent with the works by Dell'Elce et al. (2016) and Ferrari et al. (2016) and the usual modelling of the motion of asteroids reported in Scheeres and Bellerose (2008).

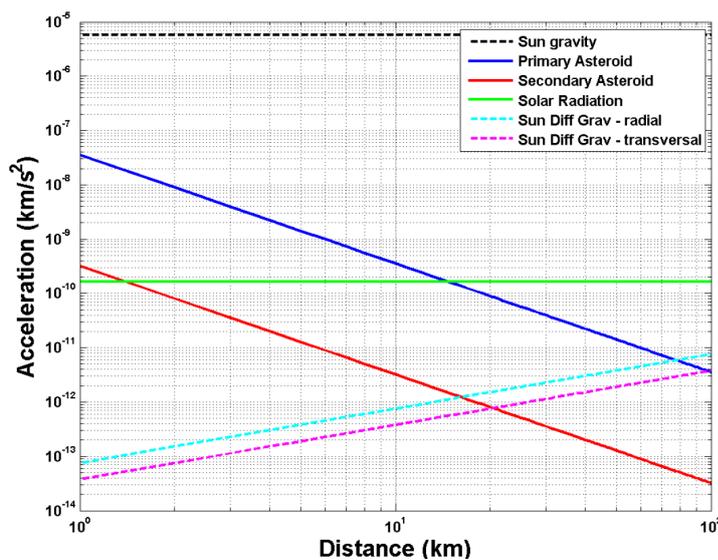


Figure 3.1: Acceleration comparison around Didymos system (ESA-ESTEC-RD27, 2016).

- The motion of Didymoon around Didymain is assumed to obey a **2BP** formulation. Other effects such as **SRP**, Third-Body Perturbation (**TBP**) from the Sun and their Non-Sphericity Perturbation (**NSP**) are neglected due to their small size relative to the main **2BP** mutual attraction.
- The motion of the spacecraft around Didymain is assumed to obey a perturbed **2BP** formulation. The perturbations that were identified as relevant for the problem are: **SRP**, **TBP** from both Didymoon and the Sun, and the **NSP** due to the gravity shape of Didymain and Didymoon. This is consistent with the work already available about the Didymos system, where the same relevant force field contributions were identified (Dell'Elce et al., 2016), (Ferrari et al., 2016), (ESA-HQ-AD3, 2016). The expected averaged magnitude of these perturbations can be seen in Figure 3.1.

It must be noted that by using this formulation for the dynamics, some simplifications have been made with respect to the modelling used by Dell'Elce et al. (2016) and Ferrari et al. (2016). For example, the coupling between the non-spheric gravity fields of the two asteroids would drive a small change in the eccentricity of the system and cause an also small libration of Didymoon. These two effects are considered negligible for trajectories like the ones used for this thesis, that will be flown well out of the ring described by the moon's orbit and for short periods of time. It is, however, taken into account in the work by Dell'Elce et al. (2016), where they analyse the stability of inner orbits ( $a < 1180$  m) for long periods of time ( $t > 1$  month). The gravity field modelling of both Didymain and Didymoon has also been simplified: a spherical harmonics gravity field expansion was used as opposed to a polyhedron model used by Dell'Elce et al. (2016) and Ferrari et al. (2016). The spherical harmonics expansions cannot calculate solutions for points that are close to the surface of the system, however in this case all the trajectories simulated will be far away from the asteroids. This means that the use of a spherical harmonics model will yield much better computational times and a negligible accuracy difference.

The escape velocity around the Didymos system can be seen in Figure 3.2. The values for the escape velocity in the distances that will be covered in the thesis range from 5 cm/s to 18 cm/s which are really low values when compared to the typical ones found when orbiting more massive bodies. This figure will become especially relevant when assessing the margin stated by mission requirement **SC-SAFE-1** in the following chapters.

A summary of the parameters that completely describe the modelled dynamics system can be found in Table 3.1.

- $a_{Moon}, i_{Moon}, e_{Moon}, \theta_{Moon,0}$ : The orbital elements of the orbit of Didymoon around Didymain, defined in the reference frame [P]. Note that the Right Ascension of the Ascending Node (**RAAN**) is not defined, since the inclination is equal to  $0^\circ$  and that the true longitude  $\theta_{Moon}$  is defined taking this into account with respect to the  $\hat{x}_P$  axis (Wakker, 2015).

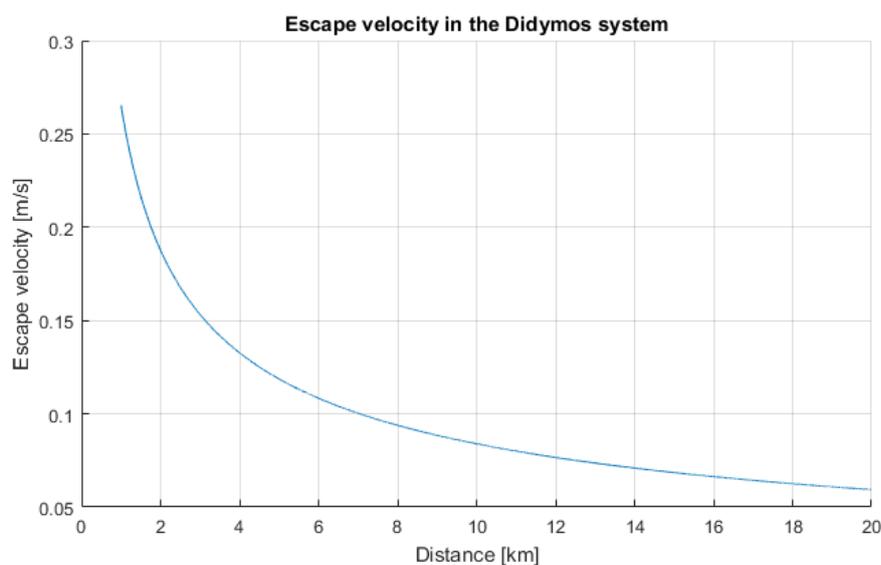


Figure 3.2: Escape velocity around Didymos system.

Table 3.1: 65803 Didymos system dynamics parameters (ESA-HQ-AD3, 2016).

Symbol	Parameter	Value and Uncertainties
$a_{Moon}$	Semi-major axis of Didymoon	$1.18 \pm 0.03$ km
$i_{Moon}$	Inclination of Didymoon	Assumed $0^\circ$
$e_{Moon}$	Eccentricity of Didymoon	0-0.03
$\theta_{Moon,0}$	Initial true longitude	0-360°
$T_{Moon}$	Didymoon orbital period	$11.92 \pm 0.005$ h
$m_{Didymos}$	Total mass of system	$5.278 \cdot 10^{11} \pm 0.54 \cdot 10^{11}$ kg
$\epsilon$	Mass ratio ( $m_{Moon}/m_{Didymos}$ )	$0.0093 \pm 0.0013$
$2\pi/\dot{\beta}_{Main}$	Spin period of Didymain	$2.2600 \pm 0.0001$ h
$2\pi/\dot{\beta}_{Moon}$	Spin period of Didymoon	$T_{Moon}$
$\beta_{Body,0}$	Initial prime meridian of Didymain/moon	0-360°
$(\lambda)_I$	Longitude of Didymos' north pole	$310^\circ$
$(\phi)_I$	Latitude of Didymos' north pole	$-84^\circ$
-	Obliquity of Moon and Main with	Both assumed $0^\circ$
-	respect to mutual orbital plane	
$R_{Main}$	Reference radius of Didymain	428.7 m
$R_{Moon}$	Reference radius of Didymoon	78.7 m
$C_{lm}, S_{lm}$	Spherical harmonics coefficients	See <a href="#">ESA-HQ-AD3 (2016)</a>
$\kappa_{Main}$	Knowledge on the gravity parameters at	0.0033%
$\kappa_{Moon}$	the beginning of the PDP [1- $\sigma$ ]	50%

- $T_{Moon}$ : The orbital period of the moon around Didymain.
- $m_{Didymos}$ : The total mass of the system  $m_{Main} + m_{Moon}$ .
- $\epsilon$ : The mass ratio of the system defined as  $m_{Moon}/m_{Didymos}$ .
- $\beta_{0,Body}$ : The initial prime meridian position of Didymain or Didymoon expressed with respect to the  $\hat{x}_P$  axis.
- $2\pi/\dot{\beta}_{Body}$ : The spin period of the bodies, defined as the time it takes for them to complete a  $360^\circ$  revolution around their rotation axis in an inertial frame. Both of them are assumed to be constant with time. In the case of Didymoon, it is equal to its orbital period around Didymain. This synchronised rotation is usually found in the moons of binary asteroids (Scheeres, 2012).
- $(\lambda)_I, (\phi)_I$ : The longitude and latitude coordinates of the north pole of the Didymos system expressed in the [I] reference system.
- $R_{Body}$ : The reference radius used for the gravity field spherical harmonics expansion, as described in Section 3.3.4.
- $C_{lm}, S_{lm}$ : The spherical harmonic expansion coefficients of the gravity potential of Didymain and Didymoon. The implementation used does not require the coefficients to be normalised.
- $\kappa_{Main}$ : The refined knowledge on the gravitational parameters of the bodies will be obtained after the DCP. It will be used by the estimation algorithm of the guidance system. It is expressed as a 1- $\sigma$  deviation around the actual  $\mu_{Body}$ .

It must be noted that all the initial conditions expressed in Table 3.1 are set at the beginning of the PDP. This will be further explained at the beginning of Chapter 8.

### 3.2. REFERENCE FRAMES

The following set of reference frames and transformations will be used during the thesis. All of them are formed by orthogonal and right-handed triads. The transformation matrices between these frames can be found in Appendix A.

In Figures 3.3 and 3.4, all the reference frames used in the thesis can be seen. Please note that in Figure 3.4 the spacecraft location is assumed to be at the exact cubesat deployment position. In reality, this will never be the case but it was represented like this for the sake of simplicity.

#### ECLIPTIC INERTIAL J2000 FRAME [I]

The Ecliptic Inertial J2000 frame is used centred at Didymain.

Its  $\hat{x}_I$  axis points towards the mean vernal equinox of the epoch J2000, while the  $\hat{z}_I$  axis is perpendicular to the ecliptic plane for that epoch. The  $\hat{y}_I$  axis completes a right-handed triad.

#### INERTIAL FRAME [P]

The inertial reference frame P is used centred at Didymain.

Its  $\hat{x}_P$  axis points towards the ascending node of the secondary body, while the  $\hat{z}_P$  axis is parallel to the pole of the system. The  $\hat{y}_P$  axis completes a right-handed triad.

#### LOCAL-VERTICAL LOCAL-HORIZONTAL FRAME [L]

The Local-Vertical Local-Horizontal (LVLH) reference frame is centred at the spacecraft location. This frame is used only in the navigation subsystem.

Its  $\hat{z}_L$  axis points towards Didymain's centre of mass, while the  $\hat{x}_L$  axis is contained in the plane of motion (defined by the instantaneous velocity vector of the spacecraft). The  $\hat{y}_L$  axis completes a right-handed triad.

#### $\Delta v$ FRAME [D]

The  $\Delta v$  frame is centred at the spacecraft location. This frame is used to compute the command errors.

Its  $\hat{z}_D$  axis points in the nominal  $\Delta v$  direction, while the  $\hat{y}_D$  and  $\hat{x}_D$  axis are defined in such a way that they define a perpendicular plane to the  $\hat{z}_D$  axis. The process followed to create this frame can be described as follows:

1. Calculate  $\hat{x}_D = \hat{z}_D \times \mathbf{c} / |\hat{z}_D \times \mathbf{c}|$ , where  $\mathbf{c}$  is a vector different than  $\hat{z}_D$ . Vector  $\mathbf{c}$  is created as:

$$\mathbf{c} = \begin{cases} \hat{z}_D + \mathbf{q}_1 & \text{if } |\hat{z}_D \times (\hat{z}_D + \mathbf{q}_1)| > |\hat{z}_D \times (\hat{z}_D + \mathbf{q}_2)| \\ \hat{z}_D + \mathbf{q}_2 & \text{if } |\hat{z}_D \times (\hat{z}_D + \mathbf{q}_2)| > |\hat{z}_D \times (\hat{z}_D + \mathbf{q}_1)| \end{cases} \quad \text{with } \mathbf{q}_1 \cdot \mathbf{q}_2 = 0 \quad (3.1)$$

2. Calculate  $\hat{y}_D = \hat{z}_D \times \hat{x}_D$ .

#### LOCAL ORBITAL FRAME [O]

The Local Orbital Frame [O] is centred at the spacecraft location. This frame is used to compute velocity margins for the deployment of the ASPECT cubesat.

Its  $\hat{z}_O$  points towards the osculating orbital angular momentum of the Self-Stabilised Terminator Orbit (SSTO) target orbit.  $\hat{x}_O$  is positive in the direction of the orbital speed and  $\hat{y}_O$  completes the right-handed triad.

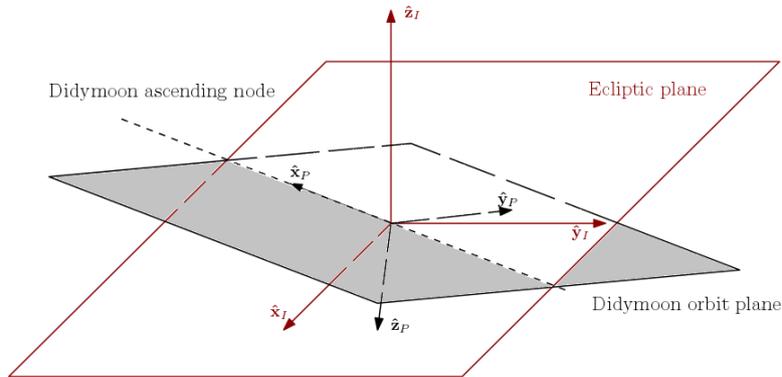


Figure 3.3: Geometry of the reference frames [I] and [P].

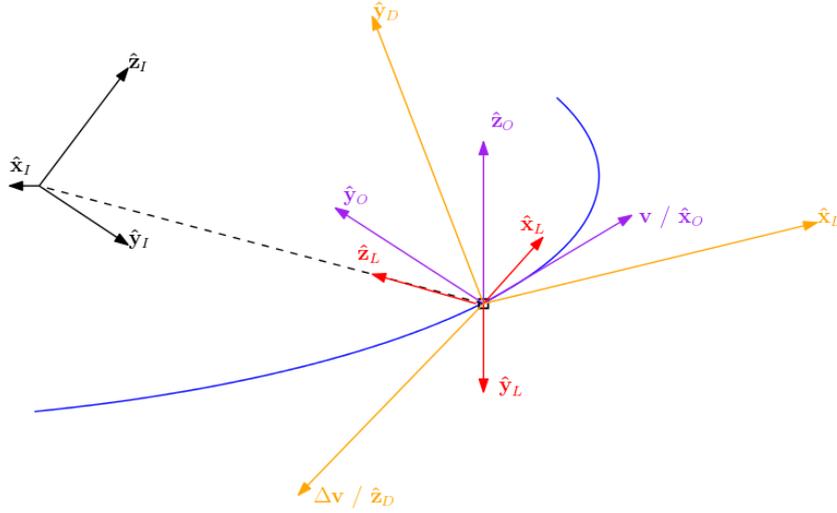


Figure 3.4: Geometry of the reference frames [L], [O] and [D].

#### DIDYMAIN-FIXED FRAME [B1]

The Didymain-fixed reference frame is centred at Didymain's centre of mass. This frame is used only to calculate the **NSP** coming from Didymain.

Its  $\hat{x}_{B1}$  axis points towards the prime meridian of Didymain, while the  $\hat{z}_{B1}$  axis is parallel to the pole of the system. The  $\hat{y}_{B1}$  axis completes a right-handed triad.

#### DIDYMOON-FIXED FRAME [B2]

The Didymoon-fixed reference frame is centred at Didymoon's centre of mass. This frame is used only to calculate the **NSP** coming from Didymoon.

Its  $\hat{x}_{B2}$ ,  $\hat{y}_{B2}$  and  $\hat{z}_{B2}$  axis point towards the minimum, medium and maximum axis of inertia of the ellipsoid that represents Didymoon's shape.

### 3.3. DYNAMICS MODEL

As described in the beginning of the chapter, Equation (3.2) shows the resulting acceleration that will be used to compute the motion of the spacecraft in the vicinity of the asteroid system.

$$\dot{\mathbf{v}}_{AIM} = \mathbf{f}_{Total} = \mathbf{f}_{2BP} + \mathbf{f}_{TBPSun} + \mathbf{f}_{TBPMoon} + \mathbf{f}_{SRP} + \mathbf{f}_{NSDidymain} + \mathbf{f}_{NSDidymoon} \quad (3.2)$$

The different terms found in Equation (3.2) are explained in the following sections.

#### 3.3.1. TWO-BODY PROBLEM

As already stated, the motion of Didymain around the Sun and Didymoon around Didymain are only described as (Wakker, 2015):

$$\mathbf{f}_{2BP} = -\mu_{CentralBody} \frac{\mathbf{r}}{|\mathbf{r}|^3} \quad (3.3)$$

where  $\mathbf{f}_{2BP}$  is the acceleration due to this effect,  $\mu_{Central}$  is the gravitational parameter of the central body (Sun or Didymain in this case) and  $\mathbf{r}$  is the radius vector from the central body to the body whose motion is described.

Note that the motion of the spacecraft around Didymain will be described by this same equation together with the ones corresponding to the perturbations, as already expressed in Equation (3.2).

#### 3.3.2. THIRD-BODY PERTURBATION

The effect that the gravitational attraction of a third particle has over the motion of a particle described by a main acceleration such as Equation (3.3), can be described as (Wakker, 2015):

$$\mathbf{f}_{TBP} = \mu_{TB} \left[ \frac{\mathbf{r}_{TB} - \mathbf{r}}{|\mathbf{r}_{TB} - \mathbf{r}|^3} - \frac{\mathbf{r}_{TB}}{|\mathbf{r}_{TB}|^3} \right] \quad (3.4)$$

where  $\mathbf{f}_{TBP}$  is the acceleration due to this effect,  $\mu_{TB}$  is the gravitational parameter of the third body (Didymoon or Sun in this case),  $\mathbf{r}$  has already been described for Equation (3.3) and  $\mathbf{r}_{TB}$  is the radius vector from the central body to the third body.

### 3.3.3. SOLAR RADIATION PRESSURE

The effect of the SRP in the motion of a spacecraft is described as (Scheeres and McMahon, 2015):

$$\mathbf{f}_{SRP} = -\frac{C_R G_1}{B} \frac{(\mathbf{r}_{Sun} - \mathbf{r})}{|\mathbf{r}_{Sun} - \mathbf{r}|^3} \quad (3.5)$$

where  $\mathbf{f}_{SRP}$  is the acceleration due to this effect,  $C_R$  is the dimensionless reflectivity coefficient of the spacecraft,  $B$  is the ratio between the mass and the wet surface of the spacecraft,  $G_1$  is the Solar radiation constant, and  $\mathbf{r}_{Sun}$  is defined as  $\mathbf{r}_{TB}$ . Note that the SRP effect on massive bodies such as the two asteroids is neglected in the formulation presented in (3.5) because the  $B$  parameter for these bodies is many orders of magnitude bigger than for a typical spacecraft.

### 3.3.4. NON-SPHERICITY PERTURBATION

The effect of the NSP on the spacecraft is divided in the one coming from Didymain and the one coming from Didymoon. The gravity field non-sphericity terms of the two bodies are described using the common spherical harmonics expansion.

For the 2BP, the acceleration accepts a reformulation in terms of a potential function as:

$$\mathbf{f}_{2BP} = -\nabla V \Rightarrow V = -\frac{\mu}{r} \quad (3.6)$$

Besides, it can be proven that a potential function that describes the gravity field of an arbitrary distribution of mass can be written as a spherical harmonics expansion as (Vallado, 1997):

$$V = -\frac{\mu}{r} \left\{ 1 + \sum_{l=2}^{\infty} \left( \frac{R_{Body}}{r} \right)^l \sum_{m=0}^l P_{lm}(\sin \phi) [C_{lm} \cos m\lambda + S_{lm} \sin m\lambda] \right\} \quad (3.7)$$

where  $C_{lm}$  and  $S_{lm}$  are the spherical harmonic coefficients, obtained by empirical fitting,  $P_{lm}(\sin \phi)$  are the associated Legendre functions of degree  $l$  and order  $m$  and  $R_{Body}$  is the reference radius of the expansion. The spherical coordinates, radius, longitude and latitude ( $r, \lambda, \phi$ ) are used in this expansion.

Following Equation (3.6), the perturbing acceleration is obtained applying the gradient operator to the potential function expressed in Equation (3.7), but only taking the perturbing terms (not taking the first one within the curly brackets). Doing this in spherical coordinates as:

$$\mathbf{f}_{NSP_{spherical}} = \frac{\partial V}{\partial r} \mathbf{u}_r + \frac{1}{r \cos \phi} \frac{\partial V}{\partial \lambda} \mathbf{u}_\lambda + \frac{1}{r} \frac{\partial V}{\partial \phi} \mathbf{u}_\phi \quad (3.8)$$

gives the the following results for each of the components:

$$\frac{\partial V}{\partial r} = \frac{-\mu}{r^2} \sum_{l=2}^{\infty} (l+1) \left( \frac{R_{Body}}{r} \right)^l \sum_{m=0}^l P_{lm}(\sin \phi) [C_{lm} \cos m\lambda + S_{lm} \sin m\lambda] \quad (3.9)$$

$$\frac{\partial V}{\partial \phi} = \frac{\mu}{r} \sum_{l=2}^{\infty} \left( \frac{R_{Body}}{r} \right)^l \sum_{m=0}^l [P_{l,m+1}(\sin \phi) - m \tan \phi P_{lm}(\sin \phi)] [C_{lm} \cos m\lambda + S_{lm} \sin m\lambda] \quad (3.10)$$

$$\frac{\partial V}{\partial \lambda} = \frac{\mu}{r} \sum_{l=2}^{\infty} \left( \frac{R_{Body}}{r} \right)^l \sum_{m=0}^l P_{lm}(\sin \phi) m [-C_{lm} \sin m\lambda + S_{lm} \cos m\lambda] \quad (3.11)$$

Since the motion is integrated with respect to Didymain, the NSP effect from Didymain can be expressed as:

$$\mathbf{f}_{NSPcartesian} = \begin{Bmatrix} \mathbf{i} \\ \mathbf{j} \\ \mathbf{k} \end{Bmatrix} = \begin{bmatrix} \cos\phi \cos\lambda & \cos\phi \sin\lambda & \sin\phi \\ -\sin\lambda & \cos\lambda & 0 \\ -\sin\phi \cos\lambda & -\sin\phi \sin\lambda & \cos\phi \end{bmatrix}^T \begin{Bmatrix} \mathbf{u}_r \\ \mathbf{u}_\lambda \\ \mathbf{u}_\phi \end{Bmatrix} = \mathbf{R}(\lambda, \phi) \cdot \mathbf{f}_{NSPspherical} \quad (3.12)$$

Equation (3.12) can be directly added to Equation (3.3), taking into account that the acceleration described in Equation (3.12) is given in a Didymain-fixed Cartesian reference frame.

However, for the NSP of Didymoon on the spacecraft, a small derivation must be conducted. As stated by Wakker (2015), the equations of motion of the spacecraft and Didymain with respect to the barycenter of the system can be written as:

$$\frac{d^2 \mathbf{r}_{SCBar}}{dt^2} = \frac{\mu_{Didymain}}{|\mathbf{r}_{SCMain}|^3} \mathbf{r}_{SCMain} + \left( \frac{\mu_{Didymoon}}{|\mathbf{r}_{SCMoon}|^3} \mathbf{r}_{SCMoon} + \mathbf{f}_{NSPcartesianMoonSC} \right) + \frac{\mu_{Sun}}{|\mathbf{r}_{SCSun}|^3} \mathbf{r}_{SCSun} \quad (3.13)$$

$$\frac{d^2 \mathbf{r}_{MainBar}}{dt^2} = \left( \frac{\mu_{Didymoon}}{|\mathbf{r}_{MainMoon}|^3} \mathbf{r}_{MainMoon} + \mathbf{f}_{NSPcartesianMoonMain} \right) + \frac{\mu_{Sun}}{|\mathbf{r}_{MainSun}|^3} \mathbf{r}_{MainSun} \quad (3.14)$$

Note that in Equation (3.13), the NSP from Didymain and the SRP to the spacecraft have been neglected for the sake of simplicity in the derivation.

If Equation (3.14) is subtracted from Equation (3.13), and the usual vector identities are used, the equation of motion for the spacecraft relative to Didymain, Equation (3.2), can be written as (Wakker, 2015):

$$\frac{d^2 \mathbf{r}}{dt^2} - \mathbf{f}_{SRP} - \mathbf{f}_{NSPMain} = \mathbf{f}_{2BP} + (\mathbf{f}_{TBPMoon} + \mathbf{f}_{NSPMoon}) + \mathbf{f}_{TBPSun} \quad (3.15)$$

The terms in Equation (3.15) can be easily identified to the subtraction of terms from Equations (3.13) and (3.14). Note that this equation shares nomenclature with Equations (3.3), (3.4) and (3.5).

From Equation (3.15) the NSP of Didymoon on the motion of the spacecraft around Didymain is identified as:

$$\mathbf{f}_{NSPMoon} = (\mathbf{f}_{NSPMoonSC} - \mathbf{f}_{NSPMoonMain}) \quad (3.16)$$

where the same structure as for the TBP in Equation (3.4) can be observed. This was a expected result, since the rationale behind both equations is exactly the same: a third body (or the NSP from that body) perturbing the motion of a spacecraft when orbiting around a primary body.



# 4

## REFERENCE TRAJECTORY AND CUBESAT DEPLOYMENT DESIGN

In this chapter, the requirements and constraints that apply to the trajectory design will be discussed in Section 4.1. In Section 4.2, the **ASPECT** cubesat deployment strategy and considerations will be covered.

### 4.1. TRAJECTORY DESIGN

The general and specific requirements that apply to any transfer trajectory design and exclusively to AIM, and how they will be dealt with are:

- **Transfer  $\Delta v$ .** This requirement is dependent on the type of transfer that is flown. Some aspects that have an influence on it are changes in the trajectory plane, the use of more than one manoeuvre and the guidance stochastic mid-course corrections. This parameter will be used as one of the objective functions to minimise during the design process, since reducing the propellant consumption is always desirable. Also note that the maximum allocated value for this phase  $|\Delta \mathbf{v}_{\max}| = |\Delta \mathbf{v}_1| + |\Delta \mathbf{v}_2|$  is set to 6 m/s (**ESA-ESTEC-AD1, 2016**), which will be implemented as an optimisation constraint.
- **Transfer time.** This is a major constraint in most of the trajectories in strongly perturbed environments. If the transfer time is too long, the risk for the spacecraft to suffer big trajectory deviations due to perturbations dramatically increases. On the other hand, if the transfer time is too short, accelerations might be too high and the process could end up being untraceable and thus uncontrollable. For the scenario at hand, the requirement **SC-AUT-1** indicates the minimum amount of time that the spacecraft shall be able to operate autonomously. Since the **PDP** must be fully autonomous, the maximum propagation time for the simulations  $t_f$  was set to 72 hours, as to guarantee that the phase could be done without ground intervention.
- **Departure and arrival conditions.** These conditions will be mainly driven by the previous phase conditions and the requirements on the release of the **ASPECT** cubesat in the target orbit. The initial conditions are taken from the **DCP** trajectory file made available courtesy of **ESA**; the uncertainty that applies to this initial condition will be described in Section 5.2. The precision of the deployment conditions will be part of the objective function to minimise when optimising the reference trajectory. In particular, the magnitude of the dispersion between the target position and the actual one,  $|\mathbf{r}_{AIM} - \mathbf{r}_T|$ , and the magnitude of the difference of velocities  $|\mathbf{v}_{AIM} - \mathbf{v}_T|$ , will be used as objective functions.

Coming from mission requirement **SC-AUT-2.2**, the final distance to the Didymos system should be greater or equal to 10 km for the **PDP** to end. This distance,  $d_{end}$ , will be checked at  $t_f$  as an optimisation constraint. When any trajectory leaves the 10 km sphere around Didymain, the mission phase is assumed finished, as stated specified by mission requirement **SC-AUT-2.1**.

- **Ground station communications.** This requirement will not be covered during the thesis work. The loss of communication link could be due to eclipses or attitude constraints. The former ones are not dependent on the small trajectories that will be carried out in the Sun side of the satellite, and the

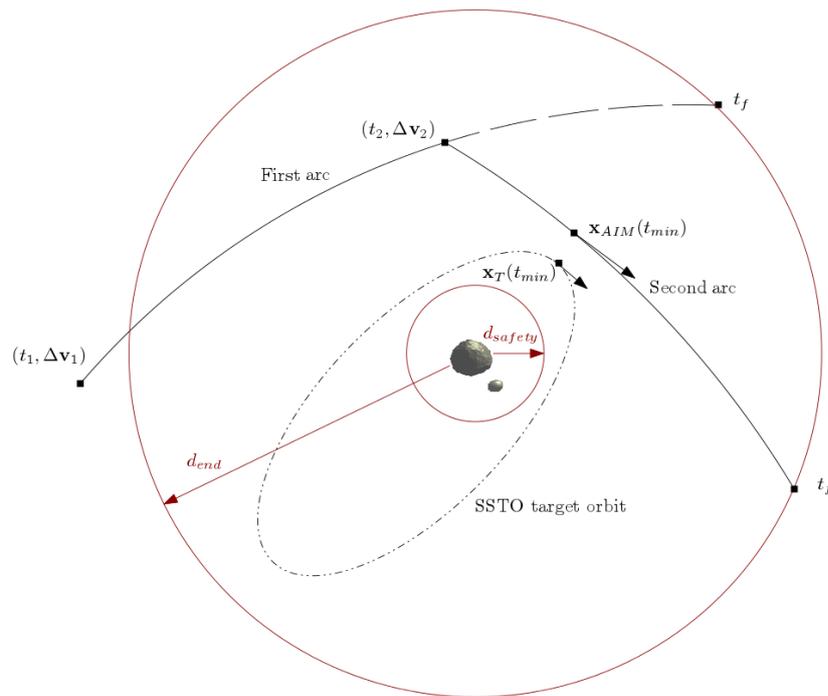


Figure 4.1: Sketch of the trajectory design geometry.

latter ones are not considered in this thesis, as explained in Chapter 2. Even though not considering communication constraints is consistent with the analysis that will be carried out during the thesis assignment, this assumption has effects on the validity of the results that will be obtained:

The pointing direction of communication antennas is usually constrained for a given spacecraft attitude. This means that when communications happen, the spacecraft usually has to orient its attitude in a certain direction to allow them to occur. This orientation manoeuvre translates into a time window during which no translational manoeuvres can take place, a fact that will not be considered in the analysis. Another effect is that when the attitude of the satellite is satisfying communication requirements, some of the navigation sensors can lose track of the features they were tracking, which would negatively affect the accuracy of this subsystem. This fact will not be considered either.

- Radiation environment. For the case under study, the radiation environment is considered as accepted at system level. This means that no radiation considerations will be made during the trajectory design process. The spacecraft will always move in the vicinity of the asteroid system and no relevant changes in the radiation dose are expected during the phase of the mission considered, regardless of the possible trajectories that it follows. This treatment is consistent with the analysis that will be done and no further effects are expected.
- Thermal environment. For missions orbiting big bodies and with long eclipses the thermal control might be strongly coupled with the trajectory design of the mission. For a mission such as AIM, which will be always in the vicinity of a small body and on the Sun side of the asteroid system, the thermal control is only related to the attitude control of the system. As such, this lies outside the scope of this work.
- Required navigation and command accuracy. This aspect will play a key role during the thesis project. In particular, the effect of the navigation errors on the state vector knowledge and thus, on the estimated  $\Delta v$  guidance commands will be one of the main couplings that will be assessed. The navigation and guidance design considerations will be treated in Chapter 5.
- Particular AIM requirements (ESA-ESTEC-AD1, 2016):

In the case of AIM, the transfer trajectories must never be in a collision course with any of the asteroid bodies. This makes any potential loss of communications or control much less dangerous for the safety

of the mission. This will be implemented as a design constraint in the optimisation problem. The safety distance to the escape velocity of the system is specified by the mission requirement **SC-SAFE-1**, with a value of  $v_{escMargin} = 0.02$  m/s. The safety distance margin with respect to Didymos is specified by mission requirement **SC-SAFE-2**, with a value  $d_{safety} = 2$  km.

Trajectories must assure continuous availability of optical feature-tracking navigation, which strongly constrains the illumination phase angle at which the spacecraft can fly. The Sun-asteroid-spacecraft phase angle must always be between  $\alpha_{lb} = 0$  and  $\alpha_{ub} = 100^\circ$ . This requirement comes from **ESA** navigation experience obtained by Rosetta operations around the comet 67P.

In Figure 4.1, a simplified sketch of the geometry of the problem characterised in this section is shown. Note that due to the simplifications made for this figure, some of the requirements are not represented, since they would lead to confusion. Also note that the dashed part of the first arc will not be nominally flown by the **AIM** spacecraft, but the collision constraints will still be checked along it during the optimisation process to ensure that the spacecraft is able to trigger a safe mode without collisions if the impulse at  $t_2$  fails.

The general requirements above described are based on the check-list available in **Wertz (2009)**, whereas the particular requirements are taken from (**ESA-ESTEC-AD1, 2016**).

## 4.2. CUBESAT DEPLOYMENT AND INJECTION ANALYSIS

As stated in Chapters 1 and 2, the assessment of the optimal deployment conditions for the **ASPECT** cubesat payload and the uncertainties therein is one of the main objectives of the thesis. It will consist in a feasibility study of the accuracy requirements in the deployment conditions of the cubesat that guarantee successfully injecting it into its target commissioning orbit. The relation between the different state and error vectors that will drive this study can be written as:

$$\mathbf{x}_T \pm \Delta\mathbf{x}_M = \mathbf{x}_{AIM} + \Delta\mathbf{x}_{Deployment} \quad (4.1)$$

where  $\mathbf{x}_T$  is the target deployment state vector,  $\Delta\mathbf{x}_M$  is the allowed deployment error margin,  $\mathbf{x}_{AIM}$  is the actual **AIM** spacecraft simulated state vector at the deployment epoch, and  $\Delta\mathbf{x}_{Deployment}$  are the deployment effects coming from the chosen deployment direction and speed and errors coming from the deployment **CRS**. These terms are depicted in Figure 4.2.

Note that all these terms will be further explained in the following sections. As it will be concluded, the position and velocity components of Equation (4.1) are decoupled. This means that the fulfilment of each one of them can be checked separately. The position component satisfaction is mainly driven by the accuracy with which the reference trajectory is followed, as it can be inferred from Figures 4.1 and 4.2a. The velocity component satisfaction is also driven by the accuracy with which the reference trajectory is followed but also by the  $\Delta\mathbf{v}_{Deployment}$  that the **CRS** gives to the cubesat, as seen in Figure 4.2b.

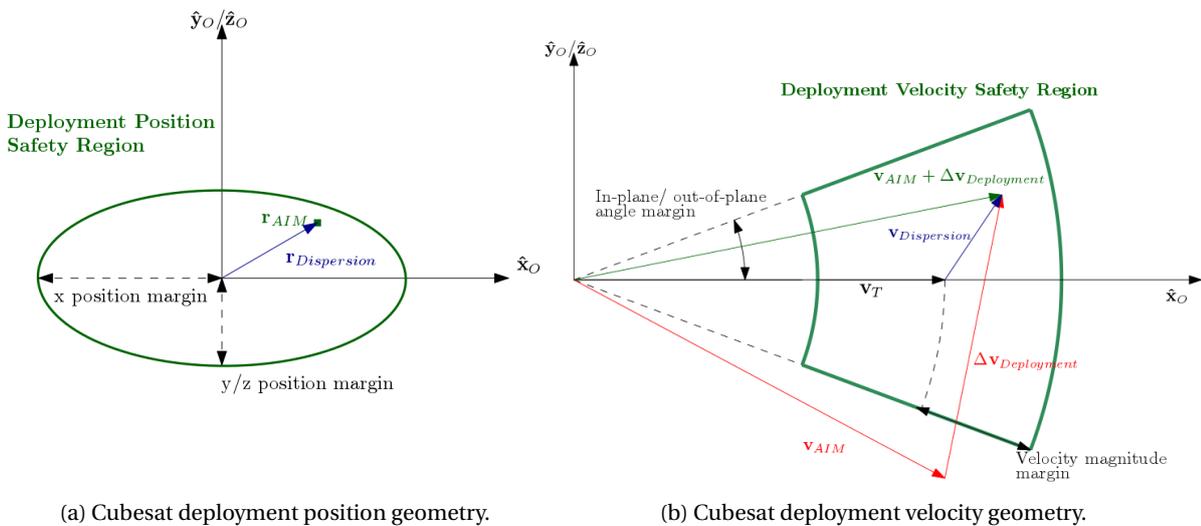


Figure 4.2: Cubesat deployment geometry.

#### TARGET DEPLOYMENT STATE VECTOR $\mathbf{x}_T$

The target deployment state vector  $\mathbf{x}_T$  corresponds to the state vector that the cubesat should have in order to fly a specific target orbit. The ASPECT cubesat first phase requires a commissioning orbit such that it is passively stable for a period of time of at least 30 days, as stated in the mission requirement **SC-PAY-1**. The only closed orbits that are stable in the Didymos system over periods longer than a few days are **SSTO** (Dell'Elce et al., 2016; GMV, 2017), and only for specific semi-major axis values. In particular, and due to operational constraints, three different orbits were considered as possible targets for the deployment, with semi-major axis of 3, 5 and 7 kilometres. **SSTO** are characteristic of environments strongly dominated by the **SRP** effect, which is frequent for asteroid systems. Its main feature is that the angular momentum vector of a spacecraft in such an orbit always points towards the Sun direction due to the effect of the **SRP** force. The full derivation of the Sun-synchronous status of these orbits can be found in the work by Scheeres (2012).

The conditions for an orbit around Didymos to be a **SSTO** are:

- The eccentricity of the orbit has a value linked with its semi-major axis, this can be expressed as:

$$e_{SSTO} = \cos \left( \arctan \left( \frac{3G_1 C_{R,cube}}{2B_{cube}} \sqrt{\frac{a_{SSTO}}{\mu_{Didymos} \mu_{Sun} a_{Didymos} (1 - e_{Didymos}^2)}} \right) \right) \quad (4.2)$$

where  $a_{SSTO}$  is the semi-major axis of the **SSTO** orbit,  $a_{Didymos}$  is the semi-major axis of the orbit of the Didymos system around the Sun and  $e_{Didymos}$  is the eccentricity vector of the orbit of the Didymos system around the Sun.

- The orbit lies in the terminator plane. If the orbit normal points towards the Sun, the periapsis must point above the asteroid orbit plane along the positive  $\hat{\mathbf{z}}$  axis, whereas if the orbit normal points away from the Sun, the periapsis must point below the orbit plane. These relations can be expressed as:

$$(\hat{\mathbf{z}} \cdot \hat{\mathbf{e}})(\hat{\mathbf{r}}_{DidymosSun} \cdot \hat{\mathbf{h}}) = 1 \quad (4.3)$$

where  $\hat{\mathbf{z}}$  is the unit vector pointing towards the angular momentum vector of the orbit of the asteroid about the Sun,  $\hat{\mathbf{e}}$  is the eccentricity unit vector that points to the pericentre of the **SSTO**,  $\hat{\mathbf{r}}_{DidymosSun}$  is the unit vector from the asteroid to the Sun and  $\hat{\mathbf{h}}$  is the unit angular momentum vector of the **SSTO** orbit.

Another fact to take into account when choosing the  $\mathbf{x}_T$  is the position within the **SSTO** that will be targeted. The main consideration for this choice is that the **AIM** spacecraft must never enter a closed orbit

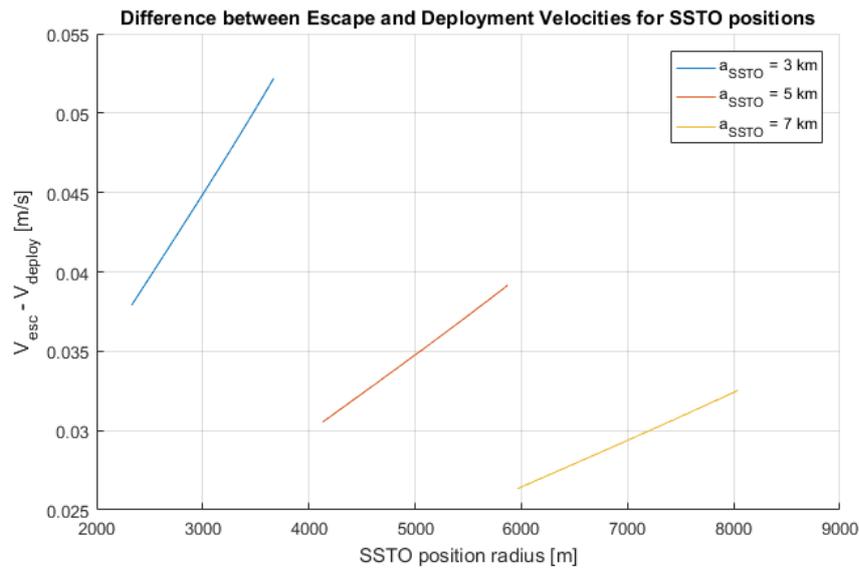


Figure 4.3: Difference between escape and deployment velocities at different distances to the system.

around the Didymos system, which translates into the margin to the escape velocity discussed in the previous section. However, at the deployment point, the AIM spacecraft and the ASPECT cubesat must be at the same position, but the cubesat velocity must be such that it remains in the target closed orbit. This implies that the  $\Delta \mathbf{v}$  given by the CRS, should at the same time allow AIM to stay above the escape velocity of the system and increase the cubesat speed in a way that fulfils the deployment. Assuming aligned deployment conditions, this coupled requirement can be expressed as the need to minimise the difference between the escape velocity and the orbital speed required in the target point of the SSTO selected for deployment,  $v_{esc} - |\mathbf{v}_T|$ . This relation for all the points of SSTO of 3,5 and 7 km can be seen in Figure 4.3.

As seen in Figure 4.3, the point where this difference is smaller is always at the pericentre of each of the three orbits studied. For this reason, the chosen deployment point will be the pericentre of the target commissioning SSTO.

Interesting to note is that the values shown at the pericentres in Figure 4.3 will be used later in the optimisation process as the theoretical lower boundaries that a trajectory can obtain for  $|\mathbf{v}_{AIM} - \mathbf{v}_T|$ . These boundaries are obtained when the 0.02 m/s margin to escape velocity is added on top of the values displayed.

#### DEPLOYMENT SAFETY MARGINS $\Delta \mathbf{x}_M$

In an operational scenario the target state vector will not be achieved with a 100% accuracy. For this reason, safety margins for these commissioning orbits were studied in Dell'Elce et al. (2016) and GMV (2017). If the deployment conditions are within the margins shown in Table 4.1, the studies concluded that the commissioning orbit is passively stable for periods longer than 30 days.

Table 4.1: ASPECT release safe intervals around nominal deployment state vector, expressed in a Local Orbital [O] reference frame (GMV, 2017).

Parameter	Safe Intervals					
	3		5		7	
Semi-major axis [km]						
Velocity magnitude [cm/s]	$\pm 2$	$\pm 2$	$\pm 1.6$	$\pm 2$	$\pm 1.4$	$\pm 2$
Angle in-Plane [deg]	$\pm 19$	$\pm 20$	$\pm 20$	$\pm 20$	$\pm 20$	$\pm 7$
Angle out-of-Plane [deg]	$\pm 20$	$\pm 20$	$\pm 20$	$\pm 20$	$\pm 20$	$\pm 16$
Position X [m]	$\pm 200$	$\pm 200$	$\pm 200$	$\pm 200$	$\pm 200$	$\pm 200$
Position Y [m]	$\pm 200$	$\pm 200$	$\pm 200$	$\pm 160$	$\pm 200$	$\pm 200$
Position Z [m]	$\pm 200$	$\pm 190$	$\pm 190$	$\pm 170$	$\pm 170$	$\pm 150$

It must be noted that Table 4.1 shows the maximum ranges of position uncertainties that produce hyper-ellipsoids containing errors that lead to a 100% success rate for the cubesat injection into a stable commissioning orbit. However, different configurations are available for such success rate. For each semi-major axis, the values given in the left column produce a position hyper-ellipsoid with maximum volume, whereas the values on the right allow for maximum velocity errors. When assessing the success rate of the cubesat injection, both intervals will be taken into account: if the cubesat deployment is successful for one of them, it will be considered a successful injection.

From what is presented in the table, it can be seen that smaller margins were found for larger  $a_{SSTO}$  values. This is explained by the fact that the SRP becomes stronger relative to the gravitational forces when the spacecraft gets further away from the system, which induces stronger instabilities for the same injection errors.

#### AIM STATE VECTOR $\mathbf{x}_{AIM}$

The AIM state vector  $\mathbf{x}_{AIM}$  will be given by the simulation carried out. This will include errors coming from the coupling of navigation and guidance, dispersions from the nominal state vector and from dynamical uncertainties and its knowledge.

#### DEPLOYMENT MANOEUVRE AND EFFECTS $\Delta \mathbf{x}_{Deployment}$

The effects due to the deployment itself, expressed as  $\Delta \mathbf{x}_{Deployment}$ , have a component coming from the chosen deployment direction and speed, and coming from errors in this choice due to the CRS. The first component is the tunable parameter that is meant to solve Equation 4.1 within the allowed margins presented in Subsection 4.2.

The deployment of the cubesat is assumed to be allowed in any direction with respect to the actual orientation of the AIM spacecraft. This assumption simplifies the solution of Equation 4.1 but is consistent with the design phase of the mission, where attitude constraints imposed by payloads/instrument are not well defined, as already stated in Section 4.1. Since the CRS injection is given by a mechanical spring, the magnitude of this impulse must be set on ground. Besides, the deployment strategy, i.e.  $\Delta\mathbf{v}_{CRS}$  will also be defined with the reference trajectory rather than the actually flown one. This prevents the on-board computer to take any on-line decision, and is consistent with the operational architecture of the mission proposed in (GMV, 2017).

The component due to the errors in the deployment can be characterised by the parameters given in Table 4.2. It must be noted that the CRS dispersion delay effect could be approximated to a position error equal to  $\Delta\mathbf{r} = \dot{\mathbf{r}}\Delta t_{delay}$ . Since the velocities at the deployment point will be close to the closed-orbit ones, i.e., in the order of few cm/s, the position errors introduced by this factor will be neglected.

Table 4.2: CRS characteristics (ESA-ESTEC-AD2, 2016).

Possible longitudinal release velocities	4-6 cm/s
Release velocity dispersion [3- $\sigma$ ]	2.8 %
Release direction dispersion (half cone) [3- $\sigma$ ]	3.9 °
CRS delay (+50°C)	2.6 s
CRS delay (+50°C) dispersion [3- $\sigma$ ]	0.2 s
CRS delay (-20°C)	4.35 s
CRS delay (-20°C) dispersion [3- $\sigma$ ]	0.44 s

# 5

## GUIDANCE AND NAVIGATION

In this chapter, the guidance and navigation subsystems will be described. The guidance scheme and its firing logic will be explained in Section 5.1 and the navigation performance model used will be introduced in Section 5.2. Note that, as already stated in Chapter 2, the control subsystem will be implemented only as a  $\Delta\mathbf{x}$  vector deviation in the form of a magnitude and direction error.

### 5.1. GUIDANCE

AIM will use a semi-autonomous PIG scheme in the proximity operations phase of the mission. This is an impulsive guidance scheme that makes use of the State-Transition Matrix (STM) of the force field to estimate (predict) the final miss-distance with respect to a reference trajectory and then tries to correct it. This kind of schemes have a relatively simple formulation, are not computationally demanding and are a proven and reliable way to follow reference trajectories in terminal guidance scenarios (Hawkins et al., 2012, 2010).

In contrast to guidance systems that continuously command accelerations to correct perturbations, impulsive schemes work with on-off cycles. These cycles are triggered when the deviation from the reference trajectory or the time since last command has reached or surpassed a certain threshold. This threshold severely affects the success of the corrections but its optimum value is also extremely difficult to determine, and many times practically impossible, as stated by Hawkins et al. (2012). This is due to the fact that the linearisation accuracy changes for different orbits (Hawkins et al., 2010). This fact will be assessed in this thesis, and results related to it can be seen in Chapter 8.

Also worth noting is that the first manoeuvre is usually not sufficient by itself to achieve acceptable positioning deviations in the end epoch, as the estimations are increasingly detrimental with increasing position and velocity deviations, as stated by Hawkins et al. (2010). For this reason, the number of impulsive corrections that will be used in this thesis project is three, which is a good trade-off between number of commands and final deviation achieved, as indicated by Hawkins et al. (2010), Hawkins et al. (2012) and obtained in the assessment found in Chapter 8.

To estimate the final miss distance, the scheme assumes that this relation holds:

$$\mathbf{x} = \mathbf{x}^* + \delta\mathbf{x} \quad (5.1)$$

where  $\mathbf{x}^*$  is the spacecraft state,  $\mathbf{x}$  is the reference state and  $\delta\mathbf{x}$  is the deviation between them. This geometry can be seen in Figure 5.1. Then, if the dynamical problem can be written as:

$$\begin{pmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \end{pmatrix} = \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}^* + \delta\mathbf{x}, t) \quad (5.2)$$

it can also be expanded in a Taylor series about  $\mathbf{x}^*$ , as:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}^*, t) + \left[ \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right]_* \delta\mathbf{x}(t) + \dots = \dot{\mathbf{x}}^* + \delta\dot{\mathbf{x}} \quad (5.3)$$

Assuming that  $\mathbf{F}(t) = \left[ \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right]_*$ , the equation of motion of the state deviation can then be written as:

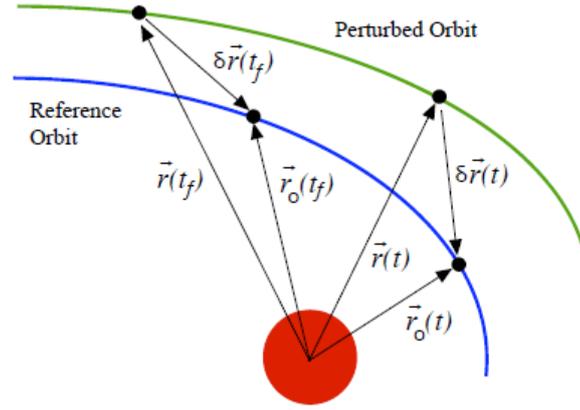


Figure 5.1: Guidance geometry [Hawkins et al. \(2010\)](#).

$$\delta \dot{\mathbf{x}} = \mathbf{F}(t)\delta \mathbf{x}(t) \quad (5.4)$$

The general solution for Equation (5.4) can be obtained as:

$$\delta \mathbf{x} = \Phi(t, t_0)\delta \mathbf{x}_0 \quad (5.5)$$

where  $\delta \mathbf{x}_0 = \delta \mathbf{x}(t_0)$ . To estimate the final miss-distance, the **STM** matrix  $\Phi(t, t_0)$  must be calculated for any epoch. Differentiating Equation (5.5):

$$\delta \dot{\mathbf{x}}(t) = \dot{\Phi}(t, t_0)\delta \mathbf{x}_0 \quad (5.6)$$

Making use of the results of Equations (5.4) and (5.5), Equation (5.6) can also be written as:

$$\dot{\Phi}(t, t_0) = \mathbf{F}(t)\Phi(t, t_0), \text{ with } \Phi(t_0, t_0) = \mathbf{I} \quad (5.7)$$

Even though this differential equation could be numerically integrated, in that case, it would make more sense to simply integrate the equations of motion, and have the best possible prediction at the desired epoch. To save computational time, however, an analytical form for the **STM** computation is required.

Now the expression for the state error is expanded into a Taylor series in time as:

$$\delta \mathbf{x}(t) = \delta \mathbf{x}_0 + \delta \dot{\mathbf{x}}_0 \Delta t + \frac{1}{2} \delta \ddot{\mathbf{x}}_0 \Delta t^2 + \dots \quad (5.8)$$

Taking into account the derivatives expressed as:

$$\begin{aligned} \delta \dot{\mathbf{x}}_0 &= \mathbf{F}(t_0)\delta \mathbf{x}_0 \\ \delta \ddot{\mathbf{x}}_0 &= \mathbf{F}(t_0)\delta \dot{\mathbf{x}}_0 = \mathbf{F}(t_0)^2 \delta \mathbf{x}_0 \end{aligned} \quad (5.9)$$

the expression for the analytical calculation of the **STM** is obtained:

$$\delta \mathbf{x} = \left[ \mathbf{I} + \mathbf{F}(t_0)\Delta t + \frac{1}{2} \mathbf{F}(t_0)^2 \Delta t^2 + \dots \right] \delta \mathbf{x}_0 = \Phi \delta \mathbf{x}_0 \quad (5.10)$$

Also, the **STM** can be decomposed in four submatrices as:

$$\Phi = \begin{bmatrix} \Phi_1 & \Phi_2 \\ \Phi_3 & \Phi_4 \end{bmatrix} \quad (5.11)$$

Then, making use of equations (5.5) and (5.11), the estimation of the final miss-distance is written as:

$$\delta \mathbf{r}(t_f) = \Phi_1(t_f)\delta \mathbf{r}(t) + \Phi_2(t_f)\delta \mathbf{v}(t) \quad (5.12)$$

The second term on the right-hand side of Equation (5.12) can be ignored. This is due to the fact that the **PIG** scheme will command a  $\Delta \mathbf{v}$  that will change the relative velocity, and thus will make the contribution

from that term meaningless in the estimation. With all the expressions that have been derived so far, four new terms are defined as:

$$\begin{aligned}
\delta\tilde{\mathbf{r}}(t_f) &= \Phi_1(t)\delta\mathbf{r}(t) \\
v_c(t) &= \delta\dot{r}(t) \\
\Lambda_c(t) &= \frac{\delta\tilde{\mathbf{r}}(t_f)}{|\delta\tilde{\mathbf{r}}(t_f)|} \\
t_{go} &= \Delta t
\end{aligned} \tag{5.13}$$

With the terms of Equation (5.13), the  $\Delta\mathbf{v}$  correction given by the **PIG** scheme is then written as:

$$\Delta\mathbf{v} = \delta\tilde{\mathbf{r}}(t_f)/t_{go} - \delta\mathbf{v} \tag{5.14}$$

where  $\delta\tilde{\mathbf{r}}(t_f)$  is the estimated miss-distance (only due to initial differences in position);  $v_c(t)$  is the closing velocity, that indicates the rate at which the spacecraft approaches the reference orbit;  $\Lambda_c(t)$  is the corrected Line Of Sight (**LOS**), that indicates the estimated **LOS** direction (and thus miss-distance direction) at the nominal end-of-mission time,  $t_f$ ; and  $t_{go}$  is the time-to-go, defined as  $t_{go} = t_f - t$ .

Now, the only thing missing is calculating the  $\mathbf{F}(t)$  force field gradient matrix. This matrix will be analytically computed from the expressions that describe the forces of the Didymos' environment. However, it must be noted that not all forces play the same role when their gradient is considered. To know which forces must be included in the  $\mathbf{F}(t)$  matrix, an assessment of their relative magnitude must be conducted.

The **2BP** force-field gradient analytical formulation, obtained from Equation (3.3), was calculated to be:

$$\mathbf{F}_{2BP}(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{3\mu x^2}{|\mathbf{r}|^5} - \frac{\mu}{|\mathbf{r}|^3} & \frac{3\mu xy}{|\mathbf{r}|^5} & \frac{3\mu xz}{|\mathbf{r}|^5} & 0 & 0 & 0 \\ \frac{3\mu xy}{|\mathbf{r}|^5} & \frac{3\mu y^2}{|\mathbf{r}|^5} - \frac{\mu}{|\mathbf{r}|^3} & \frac{3\mu yz}{|\mathbf{r}|^5} & 0 & 0 & 0 \\ \frac{3\mu xz}{|\mathbf{r}|^5} & \frac{3\mu yz}{|\mathbf{r}|^5} & \frac{3\mu z^2}{|\mathbf{r}|^5} - \frac{\mu}{|\mathbf{r}|^3} & 0 & 0 & 0 \end{bmatrix} \tag{5.15}$$

The **TBP** force-field gradient analytical formulation, obtained from Equation (3.4), was calculated to be:

$$\mathbf{F}_{TBP}(t) = -\mu_{TB} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{|\mathbf{r}_{TB}-\mathbf{r}|^2-3(x_{TB}-x)^2}{|\mathbf{r}_{TB}-\mathbf{r}|^5} & \frac{-3(x_{TB}-x)(y_{TB}-y)}{|\mathbf{r}_{TB}-\mathbf{r}|^5} & \frac{-3(x_{TB}-x)(z_{TB}-z)}{|\mathbf{r}_{TB}-\mathbf{r}|^5} & 0 & 0 & 0 \\ \frac{-3(x_{TB}-x)(y_{TB}-y)}{|\mathbf{r}_{TB}-\mathbf{r}|^5} & \frac{|\mathbf{r}_{TB}-\mathbf{r}|^2-3(y_{TB}-y)^2}{|\mathbf{r}_{TB}-\mathbf{r}|^5} & \frac{-3(y_{TB}-y)(z_{TB}-z)}{|\mathbf{r}_{TB}-\mathbf{r}|^5} & 0 & 0 & 0 \\ \frac{-3(x_{TB}-x)(z_{TB}-z)}{|\mathbf{r}_{TB}-\mathbf{r}|^5} & \frac{-3(y_{TB}-y)(z_{TB}-z)}{|\mathbf{r}_{TB}-\mathbf{r}|^5} & \frac{|\mathbf{r}_{TB}-\mathbf{r}|^2-3(z_{TB}-z)^2}{|\mathbf{r}_{TB}-\mathbf{r}|^5} & 0 & 0 & 0 \end{bmatrix} \tag{5.16}$$

The **SRP** force-field gradient analytical formulation, obtained from Equation (3.5), was calculated to be:

$$\mathbf{F}_{SRP}(t) = \frac{C_R G_1}{B} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{|\mathbf{r}_{Sun}-\mathbf{r}|^2-3(x_{Sun}-x)^2}{|\mathbf{r}_{Sun}-\mathbf{r}|^5} & \frac{-3(x_{Sun}-x)(y_{Sun}-y)}{|\mathbf{r}_{Sun}-\mathbf{r}|^5} & \frac{-3(x_{Sun}-x)(z_{Sun}-z)}{|\mathbf{r}_{Sun}-\mathbf{r}|^5} & 0 & 0 & 0 \\ \frac{-3(x_{Sun}-x)(y_{Sun}-y)}{|\mathbf{r}_{Sun}-\mathbf{r}|^5} & \frac{|\mathbf{r}_{Sun}-\mathbf{r}|^2-3(y_{Sun}-y)^2}{|\mathbf{r}_{Sun}-\mathbf{r}|^5} & \frac{-3(y_{Sun}-y)(z_{Sun}-z)}{|\mathbf{r}_{Sun}-\mathbf{r}|^5} & 0 & 0 & 0 \\ \frac{-3(x_{Sun}-x)(z_{Sun}-z)}{|\mathbf{r}_{Sun}-\mathbf{r}|^5} & \frac{-3(y_{Sun}-y)(z_{Sun}-z)}{|\mathbf{r}_{Sun}-\mathbf{r}|^5} & \frac{|\mathbf{r}_{Sun}-\mathbf{r}|^2-3(z_{Sun}-z)^2}{|\mathbf{r}_{Sun}-\mathbf{r}|^5} & 0 & 0 & 0 \end{bmatrix} \tag{5.17}$$

The results obtained for their effect on the **STM** submatrices of Equation (5.11) when compared to the term corresponding to the **2BP** can be seen in Figure 5.2.

It can be seen that the only relevant gradient is the one coming from the moon **TBP**, which for certain Monte Carlo runs reaches a 9% of the **2BP** term. The magnitudes of the other two evaluated perturbations

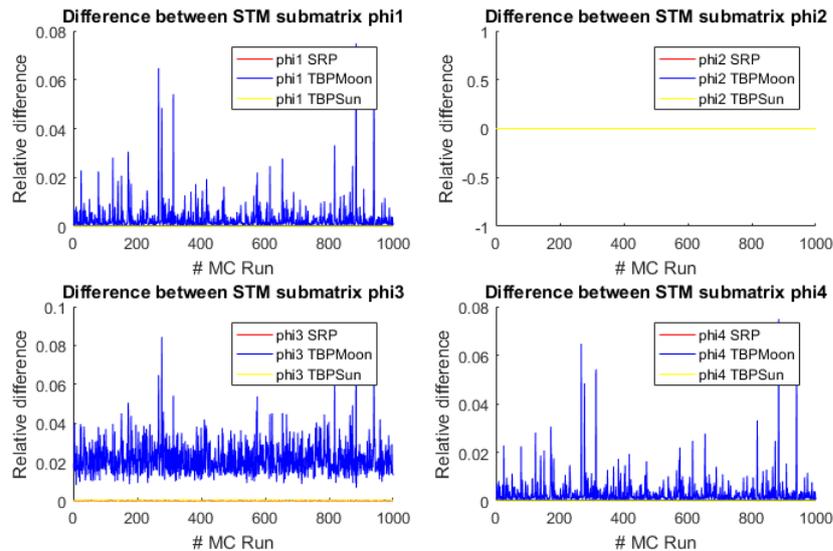


Figure 5.2: Perturbations effect on the STM  $\Phi$  of the state deviation.

are always 3 or 4 orders of magnitude smaller. From these results the conclusion is that only the 2BP and the Didymoon TBP terms will be used for the calculation of the STM.

Note that the difference for  $\Phi_2$  is always equal to zero because that submatrix is constant (i.e., not dependent on the position of the spacecraft, the Sun and Didymoon, which were the parameters of the Monte Carlo run).

Also note that the NSP effects were disregarded as their gradient behaves as the one coming from the 2BP and the TBP with distance, but is a number of orders of magnitude smaller than them.

The final form for the  $F(t)$  matrix can be read as:

$$\mathbf{F}(t) = \mathbf{F}_{2BP}(t) + \mathbf{F}_{TBP\_Moon}(t) + \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (5.18)$$

where the identity matrix come from the derivatives of the velocity with respect to the velocity, as indicated by  $\mathbf{F}(t) = \begin{bmatrix} \partial \mathbf{v} / \partial \mathbf{r} & \partial \mathbf{v} / \partial \mathbf{v} \\ \partial \dot{\mathbf{v}} / \partial \mathbf{r} & \partial \dot{\mathbf{v}} / \partial \mathbf{v} \end{bmatrix}$ .

Throughout the report, two different terms will be used to describe impulsive commands. Stochastic manoeuvres will refer to the ones that are calculated by the on-board computer using the equations described in this section. Deterministic manoeuvres will refer to the impulses that follow the scheduled nominal/reference trajectory that was designed. Both of them are scheduled based on the time of mission  $t$ .

## 5.2. NAVIGATION

Navigation in the close-proximity region of small-bodies involves complex systems and algorithms. In this thesis, spacecraft's position and velocity knowledge is strongly coupled with the guidance stochastic corrections, given to overcome dispersions with respect to the reference state, which determines the success of the cubesat deployment. Thus, the navigation subsystem performance must be taken into account for the thesis study.

Due to thesis time constraints, a full implementation of a navigation simulator is out of the scope of the project. The implementation of a navigation error performance model that produces realistic navigation error signals was identified as a compromise solution. This approach is consistent with the current development phase of the mission and also with other treatments given in the literature to this kind of preliminary-study modelling (Bayard and Brugarolas, 2008). The implementation of this model will be described in Section 7.1.3, where the rationales given in this section will be used to justify the choices made for the performance model.

To make this error performance model as representative as possible, a characterisation of the navigation errors expected for this mission will be done in the following paragraphs. This study will be based on the AIM

preliminary navigation analysis found in [GMV \(2017\)](#) and [Pellacani et al. \(2017\)](#).

There are different possible approaches to solve the problem of accurately estimating the state vector of the spacecraft with respect to the asteroid. The [AIM](#) spacecraft navigation architecture is based on Earth ground-based radiometric measurements for the phases where the spacecraft is distant to the asteroid, i.e., until the first [DCP](#). For [AIM](#), these radiometric techniques include range and range-rate measurements, Doppler tracking and Delta-Differential One-way Ranging ([DOR](#)) ([ESA-CDF, 2014](#)). After the [DCP](#), the knowledge on the environment dynamics parameters has been greatly improved, and the distance is small enough to switch to the (asteroid)relative navigation. Although different approaches are possible for this type of navigation, [AIM](#) only uses an optical navigation camera, as opposed to other missions, whose close-proximity navigation instrument suite included both a navigation camera and an altimeter. The latter is the case, for example, for the Hayabusa I and II spacecraft, reported by [Tsuda et al. \(2013\)](#), and for OSIRIS-REx, as reported by [Berry et al. \(2013\)](#). This fact will have consequences for the expected navigation signals, as will be explained later in this section.

During the phase considered in the thesis, the [PDP](#), the navigation is only based on navigation camera readings, with punctual corrections (resets) based on Earth ground-based radiometric measurements. These resets take place at the beginning of the phase, which means that the initial condition of the spacecraft for the simulations is subject to uncertainties coming from this ground-based navigation. For the thesis, this error is modelled following the navigation studies found in [GMV \(2017\)](#) for the [DCP](#). The results of these studies can be seen in [Figure 5.3](#).

It must be noted that the navigation architecture associated with the plots shown in [Figure 5.3](#) operates in cycles of 7 days (roughly 170 hours). These cycles include four arcs referred to as Nav1 (72 hours), Sci2 (22h), Sci3 (50h) and Sci4 (24h). As explained in [Section 2.1](#) and shown in [Figure 2.2](#), the full [DCP](#) consists of two cycles (14 days). A detailed description of the operational strategy during this phase is out of the scope of the thesis, however it is relevant to note that there is a data cut-off 48 hours before every manoeuvre. This implies that the ephemeris used as initial condition for the propagation of every subsequent manoeuvre has a knowledge error as indicated by the green dots shown in [Figure 5.3](#), which represent the knowledge error at the time of data cut-off propagated until the manoeuvre epoch. This uncertainty will be used as the initial value for the navigation performance models implemented in the thesis. As seen in the figure, these values for position range from 30 to 90 m and from 0.4 to 0.9 mm/s ( $1-\sigma$ ). The highest values (90 m and 0.9 mm/s) will be used as the  $1-\sigma$  value of the initial navigation error magnitude distribution for position and velocity respectively.

The other relevant information coming from [Figure 5.3](#) is the expected state vector dispersion of the actual [DCP](#) trajectory with respect to the nominal one. This information, given by the red line of the figure, will be used as an error source at the beginning of the [PDP](#) with respect to the reference [DCP](#) trajectory shown in [Figure 2.2](#). These dispersions range from 100 to 500 m and 3 to 5 mm/s ( $1-\sigma$ ) during the long [DCP](#) arcs.

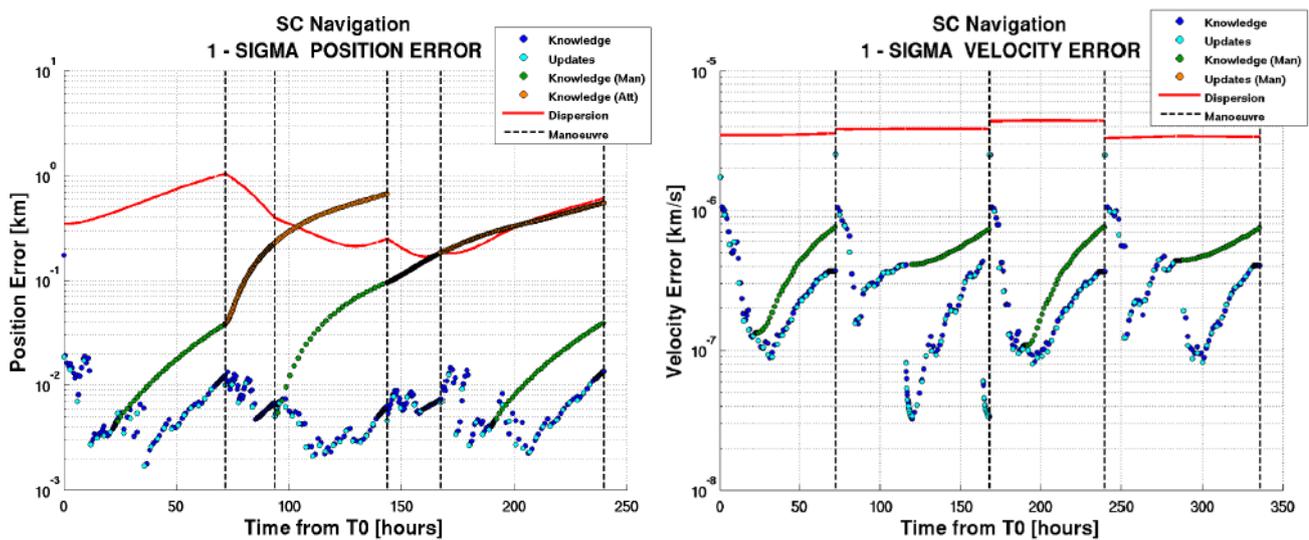


Figure 5.3: Earth ground-based navigation study signals for the [DCP](#) ([GMV, 2017](#)).

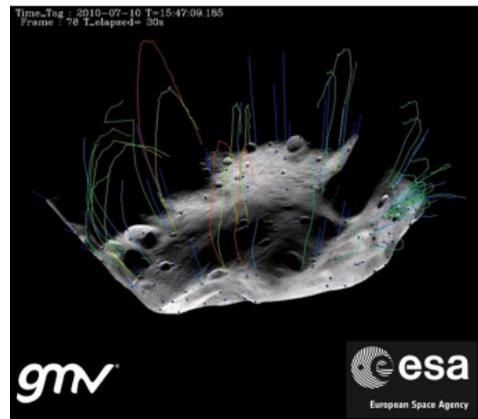


Figure 5.4: Feature tracking image processing algorithm tested on real Lutetia asteroid images taken by Rosetta (Pellacani et al., 2017).

The highest values (500 m and 5 mm/s) will be used as the  $1-\sigma$  value of the initial dispersion magnitude distribution for position and velocity respectively.

Since there will be no Didymos surface maps available from previous phases, the navigation during the PDP will be entirely based on the tracking of the relative movement between surface features in the flow of optical images taken. This relative movement will be used to reconstruct the motion of the spacecraft with respect to the asteroid. An example of the "feature tracking" algorithm can be seen in Figure 5.4.

This kind of navigation is specially challenging, since the information the filter receives is incomplete: it has an unknown scale factor (Bayard and Brugarolas, 2008). This factor comes from the lack of measurements in the radial direction: since the "size" of the features cannot be tracked from image to image, relative motion between the point features can be caused by either a purely radial displacement or a combination of lateral and radial. This uncertainty would be solved by the use of an altimeter, as already suggested in this section. However, this method presents an advantage over the Inertial Measurement Unit (IMU)-only navigation scenario: errors do not grow in an unbounded fashion (Bayard and Brugarolas, 2008). With this technique, errors build up within certain bounds until a ground-based navigation "correction" is done and the initial navigation error estimate is reset. In this scenario, the navigation filter does not improve the knowledge of the initial estimate over time as it usually does for other navigation techniques, but rather controls and minimises the estimation errors until a "reset" can be done (GMV, 2017; Pellacani et al., 2017).

One example of this kind of navigation signal can be seen in Figure 5.5. This example will be used as a reference when creating the performance models for the Navigation subsystem. The magnitudes of the  $1-\sigma$  errors observed in the plot (in red) and their behaviour will be used to generate the navigation signals of the

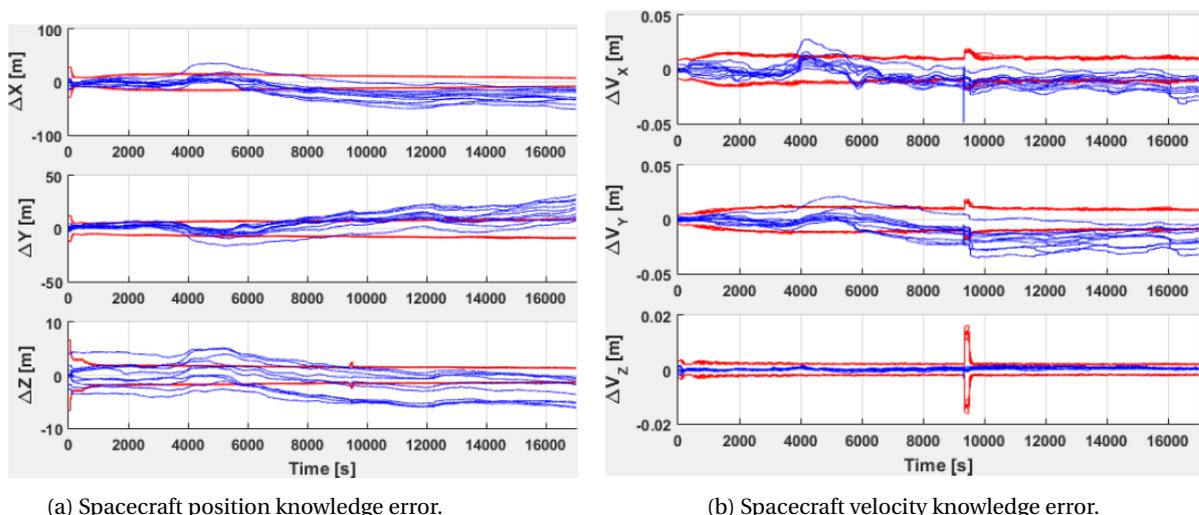


Figure 5.5: Relative navigation signal example for feature tracking algorithm in the Didymos system (Pellacani et al., 2017).

model. Note the dependency of this  $1-\sigma$  estimation with time and the impulsive manoeuvres. The latter will not be considered in the modelling, since they are caused by the manoeuvres but do not affect impulsive ones. This is consistent with the available current knowledge about the performance of this kind of navigation at this stage of the mission development.

It must be noted that this kind of navigation is extremely dependent on the illumination conditions of the surface whose features are tracked. For this reason a strict requirement for the Sun-asteroid-spacecraft phase angle ( $\alpha$ ) is set as:

$$0^\circ \leq \alpha \leq 100^\circ \quad (5.19)$$

Besides, the navigation within the allowed phase angles is not equally favourable for all of them. There is a strong dependency between the phase angle and the navigation errors expected, as the algorithm performance progressively degrades with sub-optimal lighting conditions. This phenomenon can be seen in Figure 5.5, causing the variation of the  $1-\sigma$  estimate with time. This fact will be considered in the implementation of the navigation performance models, in Section 7.1.3.



# 6

## NUMERICAL TECHNIQUES

In this chapter a description of the numerical techniques that were used during the thesis and the rationale behind their choice will be given. In Section 6.1, the MIDACO commercial optimisation tool working principle, its main features and advantages will be described. In Section 6.2, the numerical integrator chosen to solve the Ordinary Differential Equation (ODE)s that describe the motion of the thesis' bodies will be presented. In Section 6.3, the interpolation numerical scheme used will be briefly described.

### 6.1. OPTIMISATION

A problem of trajectory design is, essentially, a problem of optimisation subject to different constraints and requirements. Typically, the objective function to minimise is the propellant consumption or, for the same matter, the total increment of velocity given to the spacecraft. However, this is not always the case: some problems have a different objective function or even several ones. Some of these alternative objective functions can be the time of flight, the end-of-mission state vector, total radiation dose taken by the spacecraft during the mission, and in general, any mission design parameter. The classic optimisation problem can be read as:

$$\text{minimise } \mathbf{f}(\mathbf{x}) \text{ such that } \begin{cases} \mathbf{g}_{eq}(\mathbf{x}) = \mathbf{0} \\ \mathbf{g}_{ineq}(\mathbf{x}) \geq \mathbf{0} \\ \mathbf{x}_{lb} \leq \mathbf{x} \leq \mathbf{x}_{ub} \end{cases} \quad (6.1)$$

where  $\mathbf{f}(\mathbf{x})$  is the vectorial objective function,  $\mathbf{g}_{eq}(\mathbf{x})$  is the vectorial equality constraint function,  $\mathbf{g}_{ineq}(\mathbf{x})$  is the vectorial inequality constraint function and  $\mathbf{x}_{ub}$  and  $\mathbf{x}_{lb}$  are the lower and upper boundary constraints for the vector of optimisation variables.

For this project, and compliant with the trajectory generation procedures part of the operational baseline of AIM (ESA-CDF, 2014; ESA-ESTEC-AD1, 2016; ESA-ESTEC-RD27, 2016), an off-line trajectory optimisation algorithm is required. This will be used for the generation of the reference trajectory that the guidance sub-system will try to follow until the deployment of the ASPECT cubesat. The requirements and constraints that apply to the reference trajectory generation and to the cubesat deployment design problems were described in Sections 4.1 and 4.2.

For the problem at hand, three different objective functions were identified in Section 4.1. Multi-objective problems are addressed using Pareto fronts as the one depicted in Figure 6.1, where the Pareto front of an optimisation problem with two objective functions is shown. Note that the number of dimensions of these fronts is not limited to two, in the thesis Pareto fronts of three dimensions will be presented with the third dimension represented as a variable color for the solutions reported in a two-dimensional plot. These fronts are formed by the non-dominated solutions of the feasible design space. It can be said that a solution  $\mathbf{x}_1$  dominates a solution  $\mathbf{x}_2$  in a problem with three objective functions if:

$$\begin{cases} f_1(\mathbf{x}_1) < f_1(\mathbf{x}_2) \text{ and } f_2(\mathbf{x}_1) \leq f_2(\mathbf{x}_2) \text{ and } f_3(\mathbf{x}_1) \leq f_3(\mathbf{x}_2) \\ \text{or} \\ f_1(\mathbf{x}_1) \leq f_1(\mathbf{x}_2) \text{ and } f_2(\mathbf{x}_1) < f_2(\mathbf{x}_2) \text{ and } f_3(\mathbf{x}_1) \leq f_3(\mathbf{x}_2) \\ \text{or} \\ f_1(\mathbf{x}_1) \leq f_1(\mathbf{x}_2) \text{ and } f_2(\mathbf{x}_1) \leq f_2(\mathbf{x}_2) \text{ and } f_3(\mathbf{x}_1) < f_3(\mathbf{x}_2) \end{cases} \quad (6.2)$$

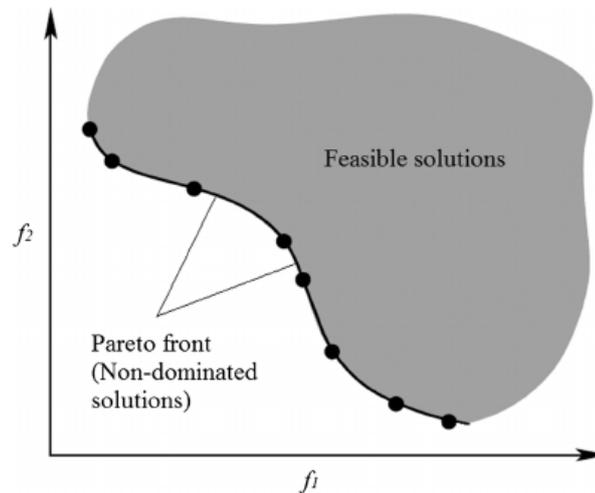


Figure 6.1: Pareto front in multi-objective optimisation example (Saboori et al., 2013).

Equation (6.2) basically expresses that solution  $\mathbf{x}_1$  dominates solution  $\mathbf{x}_2$  if  $\mathbf{x}_1$  has a lower cost than  $\mathbf{x}_2$  for at least one of the objective functions and is not worse with respect to the remaining objective functions (Saboori et al., 2013).

The problem at hand is essentially quite similar to most of the interplanetary trajectory optimisation problems presented in the literature. In this sense, the objective functions are expected to be complex, with several local minima that might make the use of local optimisers not ideal, since they tend to converge to the local minimum that is closer to the given initial condition. Due to the difficulty in determining the initial condition that would lead to the global optimum in this type of problem, the use of evolutionary algorithms is recommended in the trajectory optimisation literature (Conway, 2010).

These techniques are based on a high-level methodical research of the design space. As advantages, these techniques are robust, easy to use, they do not need the objective function to be differentiable, the initial value has a really low impact on the final result and they aim directly at solving the problem. As disadvantages, one should expect many evaluations or iterations for these methods and the convergence criteria is unclear, which makes it difficult to tell whether the algorithm has found the true global optimum or not. Besides, these algorithms scale poorly with the problem size, and become extremely slow for a large number of parameters.

The chosen optimisation tool, MIDACO, is a commercial optimisation software that uses an evolutionary optimisation algorithm based on Ant Colony Optimisation (ACO) metaheuristics (Schlueter and Munetomo, 2016).

This method was chosen because it had shown the best performance in space trajectory optimisation among the available optimisation suites. It holds the first and second records for the hardest ESA benchmark, the full Messenger mission problem (Schlueter, 2017; Schlueter and Munetomo, 2014), outperforming all the space trajectory optimisation methods distributed within ESA's Parallel Global Multiobjective Optimizer (PaGMO) suite<sup>1</sup>. The description of this benchmark problem and more simpler ones can be found in Izzo (2010). On top of that, MIDACO is available in many programming languages and its black-box architecture was easy to implement in the software architecture of the thesis. However, MIDACO license-free software version only allows the use of four optimisation variables. Since the problem at hand required eight variables, the full software version was requested to Mr. Schlueter, who kindly granted a full license for the duration of the thesis.

The use of ACO was firstly described for combinatorial optimisation. A later collaboration extended its applicability to continuous problems (Socha and Dorigo, 2008). This latter work is the one in which the ACO implementation found in MIDACO is based on.

To generate the ants, ACO makes use of a Probability Density Function (PDF) that consists of the weighted sum of several one-dimensional Gaussian PDF:

$$G^i(x) = \sum_{l=1}^k w_l^i \cdot g_l^i(x) = \sum_{l=1}^k w_l^i \cdot \frac{1}{\sigma^i \sqrt{2\pi}} e^{-\frac{(x-\mu_l^i)^2}{2(\sigma^i)^2}} \quad (6.3)$$

<sup>1</sup>PaGMO documentation and free download are available at: <http://www.esa.int/gsp/ACT/inf/projects/gtop/gtop.html>

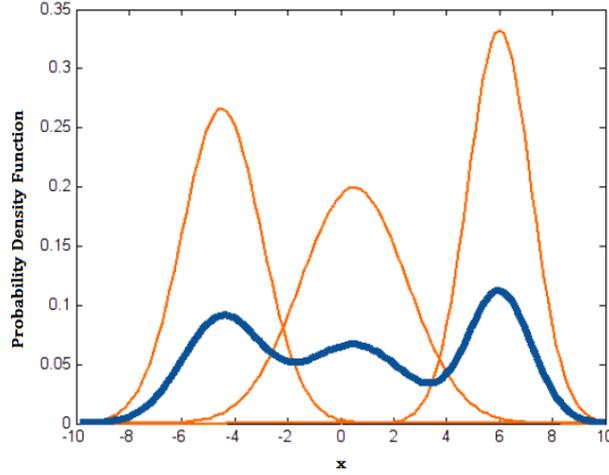


Figure 6.2: Continuous multi-kernel Gauss PDF (Schlueter and Munetomo, 2016).

where  $w_l^i$  is the weight of each Gauss function,  $\sigma^i$  is the standard deviation and  $\mu_l^i$  is the mean of the Gauss function. This function is characterised by these three values ( $w_l^i, \sigma^i, \mu_l^i$ ) that are given for every dimension of the search domain ( $i$ ) and each number of kernels ( $k$ ) of Gauss functions used to generate  $G^i(x)$  (Schlueter et al., 2009).

Figure 6.2 shows in thick blue the Gauss PDF  $G^i(x)$  that results from applying Equation (6.3) in one of the dimensions  $i$  for a kernel number  $k = 3$ . In orange, the individual Gaussian PDF  $g_l^i(x)$  used to generate  $G^i(x)$  are shown.

An archive of the  $k$  best solutions  $\mathbf{s}_l$  is built as the iterations occur and these are sorted in decreasing quality order (i.e.:  $\mathbf{s}_1$  is the best solution found so far). The weights are then defined as:

$$w_l^i = \frac{(k-l+1)}{\sum_{j=1}^k j} \quad (6.4)$$

As can be seen, the weighting function is linear with respect to  $l$ . This means that the better the solution  $\mathbf{s}_l$  is, the higher the weight associated to it will be. New ants (solutions) are evaluated, and their associated objective function values are compared to those in the solution archive, and if they are better than any  $\mathbf{s}_l$ , the new solution is stored in the corresponding spot (following the explained order) and  $\mathbf{s}_k$  is discarded.

The standard deviations shown in Equation (6.3) are calculated as:

$$\sigma^i = \frac{\text{distance}_{\max}(i) - \text{distance}_{\min}(i)}{\# \text{ generation}} \quad (6.5)$$

$$\left. \begin{aligned} \text{distance}_{\max}(i) &= \max \left\{ \left| s_p^i - s_q^i \right| \right\} \\ \text{distance}_{\min}(i) &= \min \left\{ \left| s_p^i - s_q^i \right| \right\} \end{aligned} \right\} \text{ with } p, q = 1, \dots, k \text{ and } p \neq q$$

And then, the mean is given directly by the components saved in the solution archive as:

$$\mu_l^i = s_l^i \quad (6.6)$$

Once the three parameters that describe the generation of a new ant have been defined, the actual algorithm steps for one iteration/generation can be enumerated as follows:

1. With the previous iteration solution (initial guess if it is the first iteration), the  $\mu_l^i$  is chosen as indicated in Equation (6.6). Then the weights are calculated as indicated in Equation (6.4) and the standard deviations as in Equation (6.5). The PDF distribution shown in Equation (6.3) can now be built.
2. A random number is generated using the newly created PDF function.
3. Repeat steps 1. and 2. for all dimensions  $i = 1, \dots, n$ , and the new ant ( $\mathbf{s}_l^{\text{new}}$ ) is generated.

4. Repeat steps 1., 2. and 3. for any number of ants. This number will be the number of ants per generation.
5. Evaluate solutions and go back to step number 1.

It must be noted that the algorithm described until now is a "basic" ACO as the one described by (Socha and Dorigo, 2008). However, MIDACO includes many additions that increase the capabilities of the ACO algorithm. These will be briefly described in the following paragraphs.

To enhance the constraint handling, convergence and general robustness of the algorithm, MIDACO makes use of the Oracle Penalty Method (OPM). The method is based only on one parameter  $\Omega$ , whose value is directly linked with the objective function value. The parameter is meant to be set as close as possible (or slightly above) the optimal value, which is of course tricky, since most of the times the optimal value of the target function is unknown *a priori*. For this reason, the implementation makes sure that the method performs well even for bad  $\Omega$  choices (Schlueter et al., 2009). A detailed description of the development, robustness and performance of the method can be found in Schlueter and Gerds (2010).

To enhance the global capability of the ACO algorithm, four new heuristic methods are included in the MIDACO implementation:

- A dynamical sizing of the ant population per generation is done. This allows the algorithm to use more ants for the most critical parts of the algorithm process.
- Single dimension tuning. For each generation, the best solution is locally optimised with small variations in all its dimensions. This helps when there is a large number of variables in one problem.
- A weighted average "best" ant is calculated from the solution archive. This solution is evaluated and introduced in the solution archive if positive results are obtained.
- According to the fitness progress between two different consecutive generations of ants, the algorithm enters a "final stage" mode in which local solvers are frequently called to try to solve around the current best solution. If the local solver is not able to find a solution, then the set of solutions is reset and a search around it is done, avoiding in this way getting stuck in that "current best solution".

Note that a further discussion of these heuristics methods can be found in the work done by Schlueter et al. (2009).

## 6.2. INTEGRATION

As it will be later discussed in Chapter 7, MATLAB 2016b was chosen as the programming environment and language in which the thesis tool was implemented. Since MATLAB includes many options as far as numerical integrators go, and they are commonly used for astrodynamics propagations (Shampine and Reichelt, 1997), there was no need to implement an ODE integrator. The most commonly used for astrodynamics are the solvers ode45, which is an explicit single-step Runge-Kutta (RK) of order 4(5), and ode113, which is a variable-step, variable-order Adams-Bashforth-Moulton (ABM) Predict-Evaluate-Correct-Evaluate (PECE) solver of orders 1 to 13. These methods are widely used due to their high accuracy at a low computational cost.

The choice of integration scheme was based on an preliminary accuracy analysis carried out, as depicted in Figure 6.3. The figure shows the end-epoch position accuracy for propagations of the AIM spacecraft under all the system perturbations described in Chapter 3 for a period of five days. The reference end-epoch position was obtained with a RK45 method and a relative tolerance setting of  $2e-14$ , which is the minimum value allowed by MATLAB. The use of the RK or the ABM with the lowest relative tolerance possible ( $2e-14$ ) as reference was assessed to be irrelevant for the case in question since they differed in the end-epoch state position by less than  $1e-6$  metres. Both methods would have been useful for the task at hand: assess the decrease in accuracy (aiming at a millimetre-level accuracy) as a function of the computational time for the two methods. It must be noted that the "errors" shown in Figure 6.3, are not the real errors that the method introduce with respect to the reality, but with respect to the RK45 method with a relative tolerance of  $2e-14$ . However, the "reality" is expected to be a number of orders of magnitude closer to the result coming from this reference than to the results coming from the cases analysed in the figure, which have higher relative tolerances.

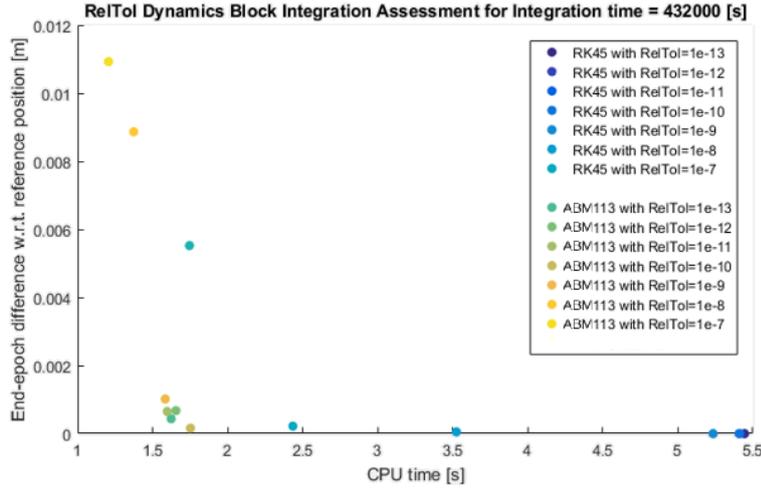


Figure 6.3: Relative Tolerance effect on **ABM** and **RK** solvers assessment.

From the results seen in the figure, a relative tolerance of  $1e-9$  on a **ABM113** was chosen, as it yields numerical errors on the order of millimetres, while notably reducing the computational time with respect to the **RK45** alternatives available in the MATLAB environment.

The **ABM113** method uses the explicit Adams-Bashforth and the implicit Adams-Moulton forms presented in Equations (6.7) and (6.8) respectively to solve an **ODE** initial value problem of the form  $\dot{x} = f(t, x)$  with  $x(t_0) = x_0$ .

$$x_{n+1}^* = x_n + h \sum_{j=0}^{k-1} \beta_j^* f_{n+j-k+1} \quad (6.7)$$

$$x_{n+1} = x_n + h \sum_{j=0}^k \beta_j f_{n+j-k+1} \quad (6.8)$$

where  $h$  is the step-size,  $k$  the number of steps of the methods,  $\beta_j$  are constant coefficients of the methods and  $f_n = f(t_n, x_n)$ .

The method is called **PECE** because it uses Equation (6.7) to predict  $x_{n+1}^*$ , which is then used to evaluate  $f(t_{n+1}, x_{n+1}^*)$  and inserted in the right hand side of Equation (6.8) to obtain the corrected value  $x_{n+1}$ . The function is then evaluated as  $f(t_{n+1}, x_{n+1})$  and the local truncation error can be estimated as (Ashino et al., 2000):

$$e_{n+1} \approx \frac{C_{p+1}}{C_{p+1}^* - C_{p+1}} (x_{n+1} - x_{n+1}^*) \quad (6.9)$$

where  $C_{p+1}$  are the corresponding error constants of the methods associated to the method's order  $p$ .

The evolution of this truncation error is closely monitored during the integration, and its value is used to control the step size and the order of the **ABM113** algorithm in every step (Shampine and Reichelt, 1997).

### 6.3. INTERPOLATION

To obtain values that lie within two nodes of the discretisation grid of a certain data set, interpolation is required. Depending on the nature of the data set and the grid resolution, different interpolation techniques are better suited for this task.

In the problem at hand, interpolation is only used to obtain the state vector of Didymoon around Didymain, which is tabulated in a previous integration. The state vector of the moon is tabulated with a high epoch density, which makes interpolations more likely to yield acceptable errors. As a compromise solution between the available methods in MATLAB, linear interpolation was chosen, since it yielded errors of less than a meter ( $< 0.1\%$  of the nominal Didymoon radius  $a_{Moon} = 1180$  m). These errors had negligible effects on the propagation of the spacecraft's state vector.

Equation (6.10) shows how to obtain the value  $y$  at the point  $x$ , which lies between the points  $(x_0, y_0)$  and  $(x_1, y_1)$ , by means of linear interpolation.

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0} \quad (6.10)$$

Simpler interpolation schemes, such as the ones that take the adjacent value to the queried one were assessed to introduce errors that surpassed the one metre margin. More complex interpolation schemes, like cubic spline interpolation, were assessed to increase the computational cost of the process too much when compared to the negligible estimation accuracy improvements they offered.

# 7

## SOFTWARE

In this chapter, the software tool implemented for the MSc thesis will be described. In Section 7.1 the software architecture will be explained and in Section 7.2 the Verification and Validation (V&V) of the tool will be presented.

### 7.1. ARCHITECTURAL DESIGN

In this section, the software architecture of the developed tool will be described at different levels. In Figure 7.1 the high-level architecture of the software implemented can be seen. In the following sections, the implementation of the software blocks shown in the figure will be detailed. This explanation will cover all the blocks that are part of the GNC subsystem, the implementation of the optimisation problem and the Monte Carlo analysis scheduler design. The latter one includes the description of the implementation of the cubesat deployment assessment algorithm described in Section 4.2. The post-processing block includes only simple conversion and plotting functions that are not worth describing in detail.

As can be seen, the software consists of two main high-level environments: an optimisation one, in blue, and a Monte Carlo scheduler one, in green. The former is in charge of obtaining the reference trajectory targeting the ASPECT commissioning orbit that will be later fed into the guidance subsystem. Its implementation will be described in Section 7.1.2. The latter uses the optimisation outputs,  $\mathbf{O}_{Optimisation}(t)$ , and the

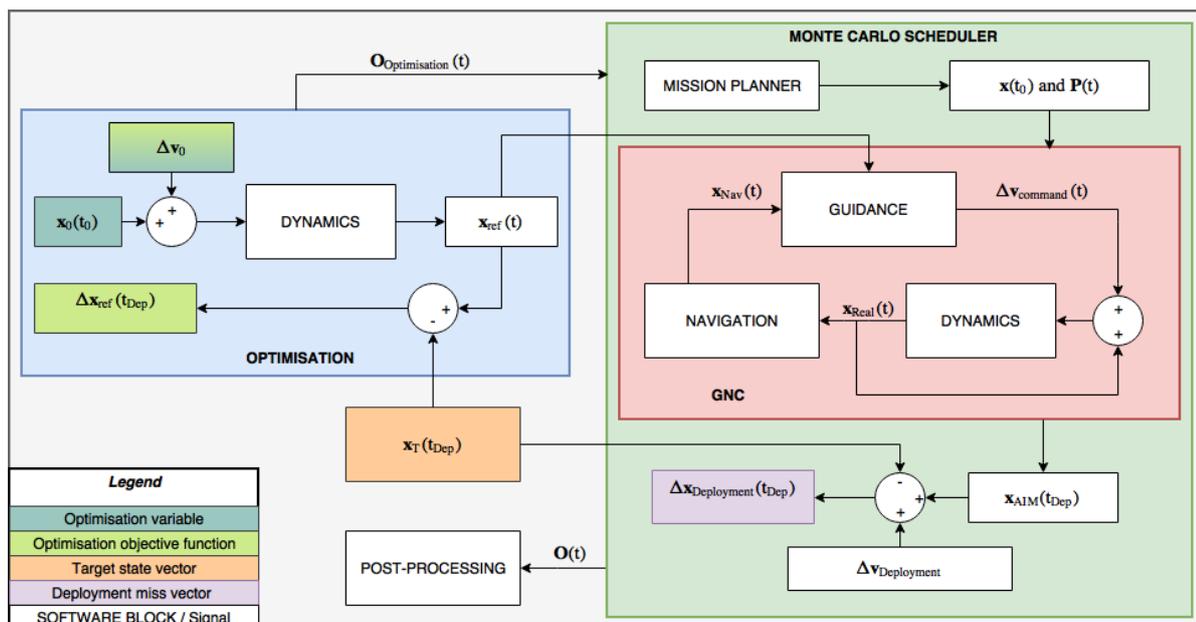


Figure 7.1: High-level software architecture.

uncertainties in some of the problem parameters (included in the parameter matrix  $\mathbf{P}(t)$ ) to launch a Monte Carlo campaign. The campaigns are set through the mission planner block, which simply structures the parameters and inputs of the simulations. The implementation of this environment will be described in Section 7.1.4. The output matrix,  $\mathbf{O}(t)$ , is later used for the final results presented in Chapter 8.

The GNC environment presented in red in Figure 7.1, embedded in the Monte Carlo scheduler, will be described in Section 7.1.3.

The dynamics block, shown embedded in both the Optimisation and the GNC environments, will be described in Section 7.1.1.

The choice of MATLAB as the programming environment for the thesis was greatly determined by the need for a GNC simulator. Simulink offers a perfect package for this task, really well documented, easy to use and widely used for this kind of task in the industry. As drawbacks it is a really slow language when compared to C++ or Fortran. This drawback was considered not extremely relevant for the thesis, since the running times of the simulations were always below one second. With the parallelisation and C++ compiling options that MATLAB and Simulink include, the running times of the longest Monte Carlo runs were always kept around 1 hour, a more than reasonable time for the task at hand.

The optimisation block was implemented in MATLAB 2016b with interfaces to the .mex functions in which MIDACO is contained. The Monte Carlo scheduler block was programmed in Matlab 2016b with interfaces with Simulink. The GNC block was implemented in Simulink 2016b. The dynamics block was implemented in Simulink 2016b. All of the programmed functions use functions compatible with C++ compiling, which makes simulations considerably faster.

### 7.1.1. DYNAMICS

The architecture implemented in the dynamics block can be seen in Figure 7.2 and the one of the perturbations block in Figure 7.3.

The grey trapezoidal boxes make reference to dynamical parameters characteristic of the different celestial bodies and the spacecraft. They are available at all times within each of the three different environments presented. The values for these parameters have been presented in Table 3.1 for the asteroid system and Sun, and in Table 2.1 for the spacecraft. The rationale/source of these parameters was also provided in Chapters 3 and 2, respectively. Note that when an uncertainty margin is present for a parameter in Tables 3.1 and 2.1, the nominal value is implemented and used for the simulations that are not part of the Monte Carlo analysis.

The initial condition for the Didymos system around the Sun,  $\mathbf{x}_{SunMain}(t_0)$ , is taken from SPICE for the epoch July 1, 2022 Coordinated Universal Time (UTC).

The initial condition for Didymoon around Didymain,  $\mathbf{x}_{MainMoon}(t_0)$ , is set at the epoch at which the injection from the DCP trajectory takes place (variable value). The nominal position is considered as  $\theta(t_0) = 0^\circ$  for this same epoch. The uncertainties are described in Table 3.1. The nominal value for the eccentricity is 0. The nominal value for  $\beta_{0,Body} = 0^\circ$ , also set at the beginning of the PDP.

The initial condition for the spacecraft around Didymain,  $\mathbf{x}_{MainSC}(t_0)$ , is a point from the trajectory followed in the previous phase of the mission (DCP). This state vector is obtained as a function of the time since the beginning of the second cycle of the first DCP. A file with the history of state vectors with the form  $\mathbf{x}_{DCP}(t)$

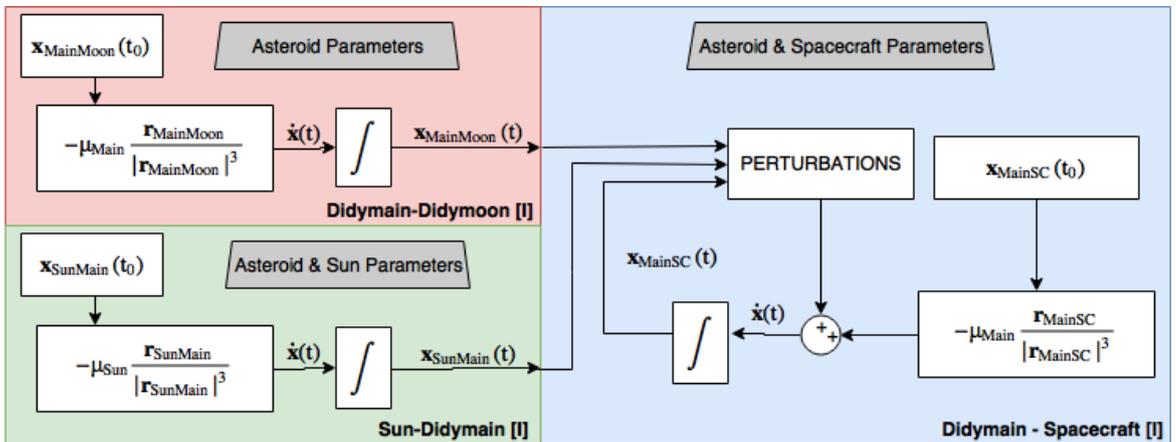


Figure 7.2: Dynamics block architecture.

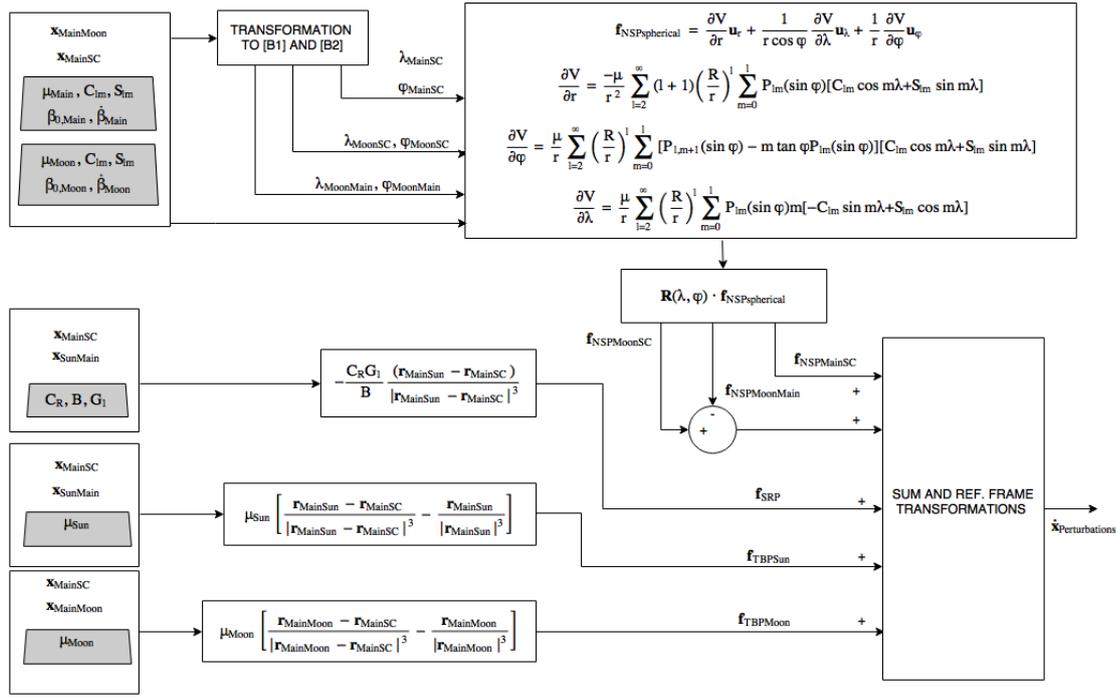


Figure 7.3: Perturbations block architecture.

is made available within the block. The time, which is one of the optimisation variables, is used to obtain the initial condition by means of linear interpolation. The nominal value for the  $C_R$  is 1.5.

The perturbations block introduces all the forces that are not part of the **2BP** described by Didymain and the spacecraft. To do so, the Sun and Moon state vectors with respect to Didymain are required in the block. All the forces are calculated in the [P] reference frame, except the **NSP** which must be computed in a fixed rotating frame around each of the bodies, [B1] and [B2]. All these forces are later transformed to the [I] reference frame and added to the spacecraft state vector derivative  $\dot{\mathbf{x}}_{MainSC}$ . Note that the definition of all these reference frames was done in Section 3.2.

Note that  $P_{lm}(x)$  are the not-normalised associated Legendre functions of degree  $l$  and order  $m$ . This is consistent with the spherical harmonic coefficients used in the implementation of the **NSP**, which are also not normalised.

### 7.1.2. OPTIMISATION WITH MIDACO

The architecture followed for the two-manoevres trajectory generation can be seen in Figure 7.4. All the elements presented in Equation (6.1) are present in the formulation. It must be noted that the reason as to why two arcs are followed during the **PDP** will be later explained in Chapter 8. Also note that the optimisation architecture implemented would be classified under the "direct shooting" trajectory optimisation methods according to the usual classification nomenclature used in the field (Stryck and Bulirsch, 1992).

For the algorithm to start, it is necessary to provide an initial condition for the optimisation variables  $\mathbf{x}$ . It is worth noting that the optimisation problem is formulated as a function of only eight variables (six  $\Delta \mathbf{v}$  components and two  $t$ ). The upper and lower boundaries that apply to the optimisation variables are:

$$\begin{pmatrix} -[6 \ 6 \ 6]^T \text{ m/s} \\ t_{\min, DCP} \\ -[6 \ 6 \ 6]^T \text{ m/s} \\ 1800 \text{ s} \end{pmatrix} \leq \begin{pmatrix} \Delta \mathbf{v}_1 \\ t_1 \\ \Delta \mathbf{v}_2 \\ t_2 \end{pmatrix} \leq \begin{pmatrix} [6 \ 6 \ 6]^T \text{ m/s} \\ t_{\max, DCP} \\ [6 \ 6 \ 6]^T \text{ m/s} \\ t_f \end{pmatrix} \quad (7.1)$$

where the  $\Delta \mathbf{v}$  boundaries were set according to constraint  $\Delta \mathbf{v}_{\max}$ ,  $t_1$  takes values from the **DCP** timeline and the 1800 s obeys the requirement **SC-AUT-3**.

With the time of the first deterministic manoeuvre  $t_1$ , the **DCP** state vector history file is queried and the initial condition for the transfer trajectory is obtained. Then, a first trajectory is integrated until the  $t_f$  is

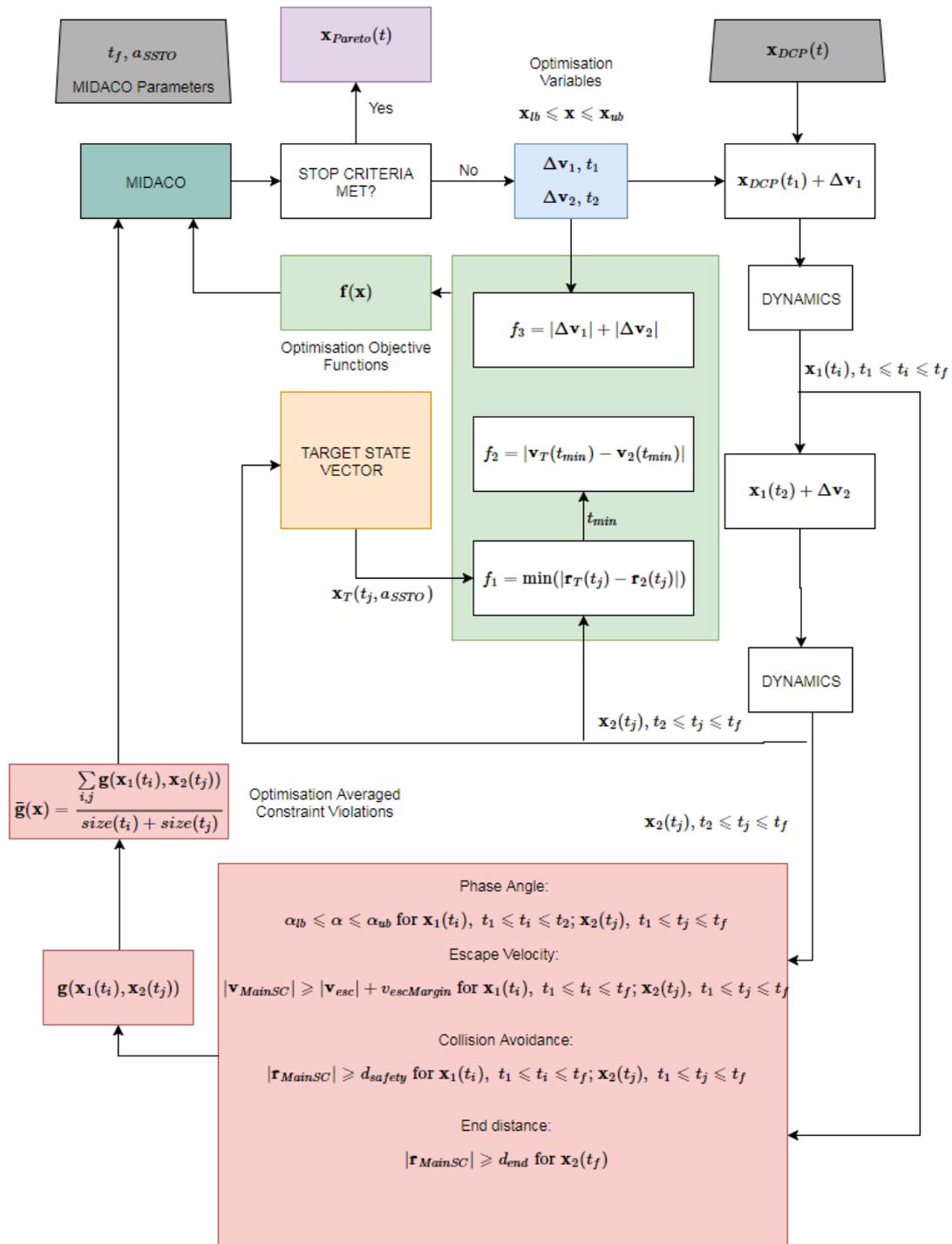


Figure 7.4: Optimisation block architecture.

reached or until the spacecraft leaves the region delimited by a sphere of radius 10 km centred at the barycentre of the system. This first trajectory is stored, since it will be later used for the constraint evaluation.

At a certain epoch ( $t_2$ ) during this first trajectory, a second deterministic  $\Delta \mathbf{v}_2$  is given. This new trajectory is propagated until one of the two criteria that applied to the first arc is met. This second arc is also stored, since it will be used for the constraint and objective function evaluation.

$$\mathbf{g} = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \end{pmatrix} = \begin{pmatrix} (\alpha_{ub} - \alpha) / \alpha_{ub} \\ (\alpha - \alpha_{lb}) / \alpha_{ub} \\ (|\mathbf{v}_{AIM}| - |\mathbf{v}_{esc}| - v_{escMargin}) / v_{escMargin} \\ (|r_{AIM}| - d_{safety}) / d_{safety} \\ (|r_{AIM}| - d_{end}) / d_{safety} \\ (|\Delta \mathbf{v}_{max}| - |\Delta \mathbf{v}_1| - |\Delta \mathbf{v}_2|) / |\Delta \mathbf{v}_{max}| \end{pmatrix} \geq \mathbf{0} \quad (7.2)$$

It must be noted that in Figure 7.4,  $t_i$  and  $t_j$  are the discrete times at which the state vectors are outputted by the integrator.

The implemented problem considers only five constraints ( $g_1$  and  $g_2$  are part of the same one), as shown in Equation (7.2).

The violations of the  $\mathbf{g}$  function are added for every epoch and the averaged constraint violation  $\bar{\mathbf{g}}$  is created and fed to the [MIDACO](#) software as shown in Figure 7.4.

It must be noted that the rationale behind these constraints and the values for the parameters found in Equation (7.2) was already given in Section 5.2 for  $g_1$  and  $g_2$ , and in Section 4.1 for the rest of the constraints.

For every epoch available of the second arc, the distance to the target point is calculated. The epoch at which this distance is minimal is chosen as the deployment point in the current optimisation iteration, and this distance value is used as the first objective function. The difference in velocity between the target state vector and the velocity at the current deployment point is the second objective function. The third objective function is constructed by adding the norm of the deterministic  $\Delta \mathbf{v}$ s given. With these three objective functions a global objective function must be constructed. The latter will be used by [MIDACO](#) to make the internal decisions during the optimisation iterations, but will not be considered in the dominance criteria filter for the generation of the Pareto front. Since the only role of this global objective function is to "direct" the optimisation evolution, the choice was to weight  $f_1$ ,  $f_2$  and  $f_3$  in such a way that variations of the end values had more or less the same contribution to the objective function. The form of the global objective function can be seen as:

$$f_4 = w_1 f_1 + w_2 f_2 + w_3 f_3 \quad (7.3)$$

For the first objective function,  $f_1$ , a value lower than 10 m was assumed to be acceptable for the reference trajectory. For the second objective function,  $f_2$ , a value with a difference smaller than 1 mm/s to the lower boundary shown in Figure 4.3 was assumed acceptable. Note that the reason why these values are a lower boundary for the velocity difference was already explained in Section 4.2. These two values are obtained by assuming an acceptable deviation in the reference trajectory of 5% from the typical deployment margins (200 m and 2 cm/s) that are shown in Table 4.1. These two values introduce a way to set the weight of the global objective function, since both of them "must" have the same relevance in the global function.

For this reason, the following weights were set  $w_1 = 1$ ,  $w_2 = 10000$  and  $w_3 = 1$ . The weight for  $f_3$  was set in a way that it did not drive the optimisation process but was still relevant for the algorithm, so Pareto fronts could identify different trajectory "families" according to the global  $\Delta \mathbf{v}$  of the manoeuvre. This was verified by preliminary tests, that proved that higher weights for  $f_3$  led to convergence problems and an overall bad quality of results. This was assumed to be due to the direct control that the algorithm has over the  $\Delta \mathbf{v}$ , since they are part of the optimisation variables, as opposed to the more indirect control that has over the position and velocity differences,  $f_1$  and  $f_2$ .

The objective functions are then fed to the [MIDACO](#) software, which determines what to do next in the optimisation process. Please note that the "target state vector" block implementation will be described at the end of the section.

[MIDACO](#) working principle has been explained in Section 6.1. The basic output of the block is the decision to either continue or stop the optimisation process. If the process does not fulfil any of the stop criteria present in the software, the algorithm generates a new set of variables and the next iteration takes place. On the contrary, if the optimisation stops, a file containing the Pareto points found during the process is created. This Pareto front will be used in Chapter 8 to obtain the best trajectories.

The internal [MIDACO](#) parameters that control the stop criteria and the behaviour of the algorithm will be explained later in this section.

#### TARGET STATE VECTOR

The algorithm implemented to calculate the state vector of a [SSTO](#) with a semi-major axis  $a_{SSTO}$  around the Didymos system, can be summarised as follows:

1. Obtain the eccentricity  $e_{SSTO}$  following Equation (4.2), and using the cubesat parameters described in Table 2.1.
2. Obtain the state vector  $\mathbf{x}_1$  of the pericentre of an orbit with semi-major axis equal to  $a_{SSTO}$  and eccentricity equal to  $e_{SSTO}$ . It must be expressed in an inertial frame centred at the barycentre of the system. This orbit can have any orientation (so  $\Omega, \theta, \omega$  and  $i$  are free).
3. Calculate the angular momentum of this orbit,  $\mathbf{h}_1$ , and the vector that points from the barycentre of the system to the Sun,  $\mathbf{R}_{Sun}$ . Now calculate the angle between them as:

$$\delta = \arctan2(|\mathbf{a} \times \mathbf{b}|, \mathbf{a} \cdot \mathbf{b}) \quad (7.4)$$

where  $\mathbf{a}$  and  $\mathbf{b}$  are any vector, and  $\delta$  the absolute value of the shortest angle between them.

In this case  $\mathbf{a} = \mathbf{h}$  and  $\mathbf{b} = \mathbf{R}_{Sun}$ . The resulting angle is  $\delta_1$ .

4. Rotate the state vector  $\mathbf{x}_1$  around the axis defined by  $\mathbf{h} \times \mathbf{R}_{Sun}$  by an angle equal to  $\delta_1$ . This operation will place the plane of the orbit perpendicular to the  $\mathbf{R}_{Sun}$  vector. The resulting state vector is  $\mathbf{x}_2$ .
5. Calculate the angular momentum of Didymos around the Sun,  $\mathbf{h}_{SunDidymos}$ . Also calculate the eccentricity vector  $\mathbf{e}$  of the state vector obtained in the last step,  $\mathbf{x}_2$ , and its angular momentum,  $\mathbf{h}_2$ . Obtain the angle between  $\mathbf{h}_{SunDidymos}$  and  $\mathbf{e}$  using again Equation (7.4). This angle is  $\delta_2$ .
6. Rotate state vector  $\mathbf{x}_2$  around  $\mathbf{h}_2$  by an angle  $\delta_2$ . The resulting orbit state vector,  $\mathbf{x}_{SSTO}$ , fulfils Equation (4.3).

The vector  $\mathbf{x}_{SSTO}$  is the state vector of the pericentre of a **SSTO**. It will be used to calculate the objective function values of the optimisation process, as shown in Figure 7.4 and explained in Section 7.1.2.

#### MIDACO PARAMETERS

**MIDACO** software has several parameters that determine the behaviour of the algorithm and its performance. The optimum values for these parameters are problem-specific, and thus they require specific tuning for every problem implementation done.

In this section, **MIDACO** parameters will be described according to [Schlueter and Munetomo \(2016\)](#). Besides, the typical values given to them during the optimisation campaign will be discussed.

- **ACCURACY**: **MIDACO** considers an inequality constraint to be feasible if  $g(x) \geq -\mathbf{ACCURACY}$ . For the cases considered an **ACCURACY** parameter of 0.00001 was used. This value was chosen considering that on average, the integration of the trajectories yielded around 500 epochs, which translates to a cumulative violation of 0.005. This translates into a maximum "peak" violation of 0.5%, considering that the constraint functions were normalised as shown in Equation (7.2).
- **SEED**: This parameter defines the initial seed for **MIDACO**'s internal pseudo random number generator. This allow **MIDACO** runs to be 100% reproducible. For the thesis, the number was varied a few times to check the different Pareto fronts produced when global searches were done, as it will be reported in Chapter 8.
- **FSTOP**: This parameter enables a stopping criterium based on the value of the objective function to be reached. This parameter was not used during the thesis.
- **ALGOSTOP**: This parameter enables a stopping criterium based on the number of internal **ACO** restarts without improvement of the objective function value. It is a good indicator of global optimality, although for CPU-time expensive evaluations, it is usually not efficient. For this thesis, it was not implemented due to the long times associated with its fine-tuning.
- **EVALSTOP**: This parameter enables a stopping criterium based on the number of function evaluations without improvements of the objective function value. In this thesis, good global optimal conditions were observed to be consistently reached for any run with the nominal parameter set-up and a number of function evaluations lower than 100000. The quality of the global solutions was assessed based on the Pareto fronts obtained and the objective function values. This was possible because there are known

lower ("ideal") boundaries for the objective function values (0 m for the position difference and the values shown at the pericentre in Figure 4.3 for the velocity difference). The **EVALSTOP** parameter was set to 100000, but was usually overrode by a manual stop.

When local optimisation was carried out, the process was always manually stopped.

- **FOCUS**: This parameter forces **MIDACO** to focus its search process around the current best solution. Its value is used to reduce the upper bound of the standard deviation used by the algorithm  $\sigma^i$ /**FOCUS**. This parameter is specially useful for refinement runs. For the global optimisation stage, a value of 0 was used, whereas for the local one values ranging from 1000 to 100000 were used.
- **ANTS**: This parameter allows the user to fix the number of *ants* (iterates) which **MIDACO** generates within one generation (major algorithm iteration). This parameter must be used in combination with **KERNEL** to tackle non-convex and difficult problems. A further discussion on both of them will be given in the next paragraph.
- **KERNEL**: This parameter allows the user to fix the number of kernels within **MIDACO**'s multi-kernel Gauss **PDF**. The kernel size corresponds also to the number of solutions stored in **MIDACO**'s solution archive  $\mathbf{s}_l$ . A low number of kernels increases the risk of **MIDACO** getting stuck in a local optimum, while a larger kernel number increases the chance of reaching a global optimum at the expense of more computational time per generation. Both the **KERNEL** and **ANTS** parameter performance effects are strongly problem-specific. The preliminary tests carried out to tune them showed that the problem implemented required high values of both of them to guarantee that the algorithm did not get stuck in a local optimum.

The **KERNEL** parameter was given a value of 40 and the **ANTS** one a value of 400 for the global optimisation runs. For the local runs, both values were reduced to 10 and 100 respectively.

- **ORACLE**: This parameter sets the  $\Omega$  value described in Section 6.1. For the thesis a value of 1000 was used. This value is obtained using Equation (7.3) and assuming  $f_1 = 200$  m,  $f_2 = 0.08$  m/s and  $f_3 = 6$  m/s which are the maximum possible values for the objective functions that would allow the cubesat deployment in only an ideal scenario.
- **PARETOMAX**: This parameter defines the maximal number of non-dominated solutions stored by **MIDACO**. These solutions are called Pareto points. For the thesis, the value was set to more than than the number that was usually found in the maximum number of evaluations, so that no non-dominated solutions were lost in the process. A value of 300 was enough for all the cases.
- **EPSILON**: This parameter defines the precision used by **MIDACO** for its multi-objective dominance filter. A new non-dominated solution is only introduced in **MIDACO**'s Pareto point archive if at least one of their objectives is at least  $100 * \text{EPSILON}\%$  better than the corresponding objective of any previous solution. For the thesis, the default value of 0.001 was set. Only for problems with many objective functions (more than four), it is recommended to increase that number to prevent slowing down **MIDACO** internal calculation times.
- **CHARACTER**: This parameter allows the user to activate special treatment for discrete optimisation problems. For the thesis, only continuous variables were used and thus this parameter was set to default (0).

### 7.1.3. GUIDANCE NAVIGATION AND CONTROL

In this section, the architecture implemented for the **GNC** block shown in Figure 7.1 will be explained. The Simulink implementation of the block can be seen in Figure 7.5.

It can be seen that the spacecraft state vector coming out from the dynamics block is used for both the navigation and the guidance blocks. The navigation feeds the state vector errors to the guidance block as a function of the mission time. The guidance block makes use of information coming from the navigation system and the pre-loaded reference trajectory to estimate the required manoeuvres. Also must be noted that the position of the Moon with respect to Didymain is assumed to be known perfectly because its translational dynamics are reduced to a **2BP** with a high-level of accuracy and also because its effect on the guidance

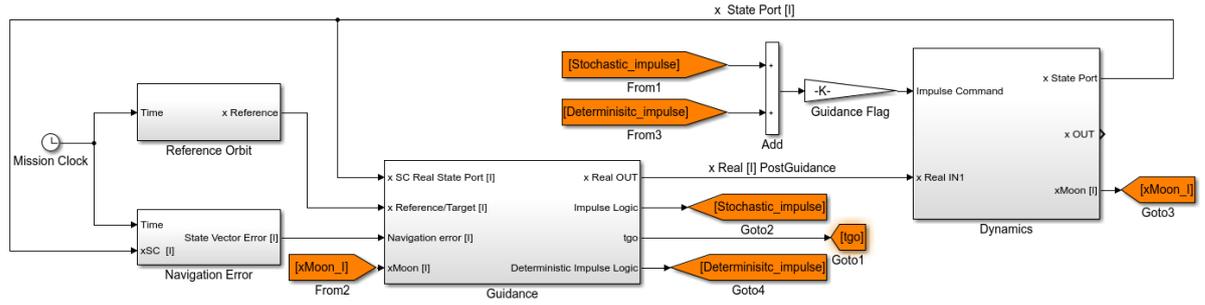


Figure 7.5: GNC block architecture.

manoeuvre estimation is small (and thus small errors would yield even smaller effects). These two facts were also explained in Section 5.1.

Besides the spacecraft state vector with the added  $\Delta\mathbf{v}$ 's, the guidance block also feeds two logic signals coming from the firing logic. These signals are used to reset the initial condition of the integrator of the dynamics block. This is done because integrators use the result of the previous epoch  $\mathbf{x}(t_{i-1})$  as initial condition to integrate  $\dot{\mathbf{x}}(t_{i-1})$  and obtain  $\mathbf{x}(t_i)$ . The reset is required because in this case you are changing the initial condition changes to  $\mathbf{x}(t_{i-1}) + \Delta\mathbf{v}(t_{i-1})$ .

### GUIDANCE

The guidance block architecture can be seen in Figure 7.6. It can be seen that the estimation of the relative position both present and future requires the reference trajectory,  $\mathbf{x}_{ref}(t)$ , the spacecraft position with respect to Didymain as known by the navigation system,  $\mathbf{x}_{Nav}(t)$ , the position of the Moon with respect to Didymain,  $\mathbf{x}_{MainMoon}(t)$  and the mission time,  $t$ .

The internal architecture of the estimation block can be seen in Figure 7.7. Note that the derivations and explanations behind this guidance scheme were given in Section 5.1.

Once the estimation of the required relative speed magnitude,  $|\delta\dot{\mathbf{r}}(t_{min})|/t_{go}$ , that would close the position gap,  $|\delta\dot{\mathbf{r}}(t_{min})|$ , in the available time,  $t_{go} = t_{min} - t$ , is done, the  $\Delta\mathbf{v}(t)$  required is calculated as a difference of the desired relative speed minus the current relative speed.

When the firing logic sends a "Yes" (1) command, the  $\Delta\mathbf{v}(t)$  is added to the real spacecraft state vector, whereas when the firing logic sends a "No" (0) command, the real spacecraft state vector is added to a 6x1 zeros vector.

The firing logic Simulink block can be seen in Figures 7.8 and 7.9. Note that the parameter "FiringCond" will be referred to as  $t_{firing}$  from now on.

In Figure 7.9, the firing check block makes sure than only activation impulses are sent for the first epoch

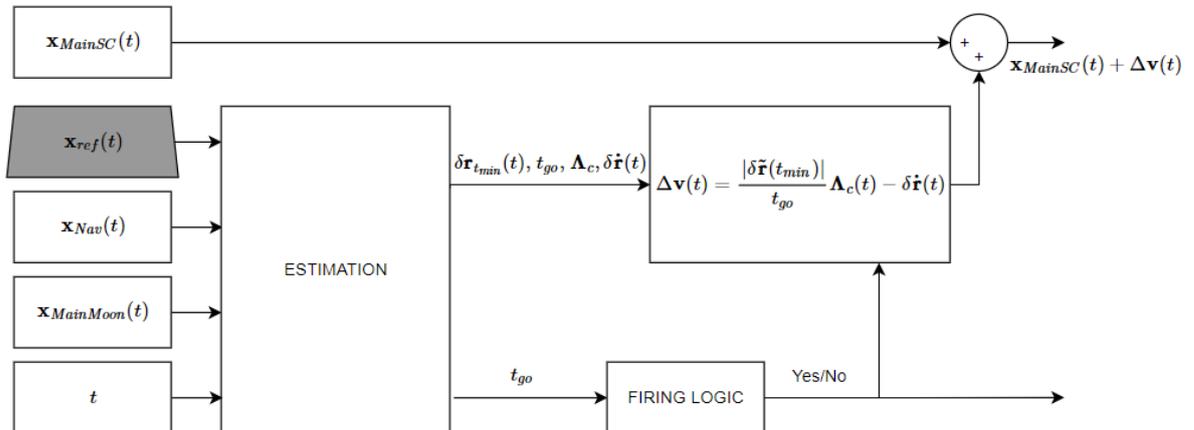


Figure 7.6: Guidance block architecture.

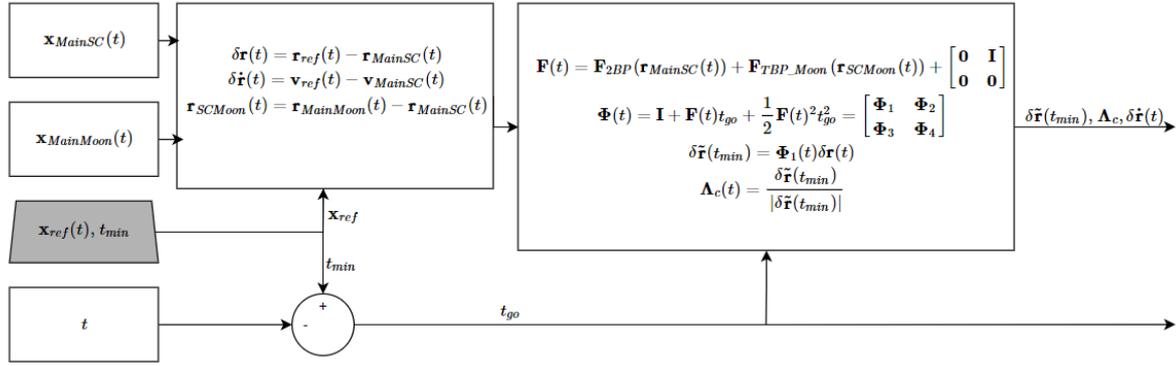


Figure 7.7: Estimation block architecture.

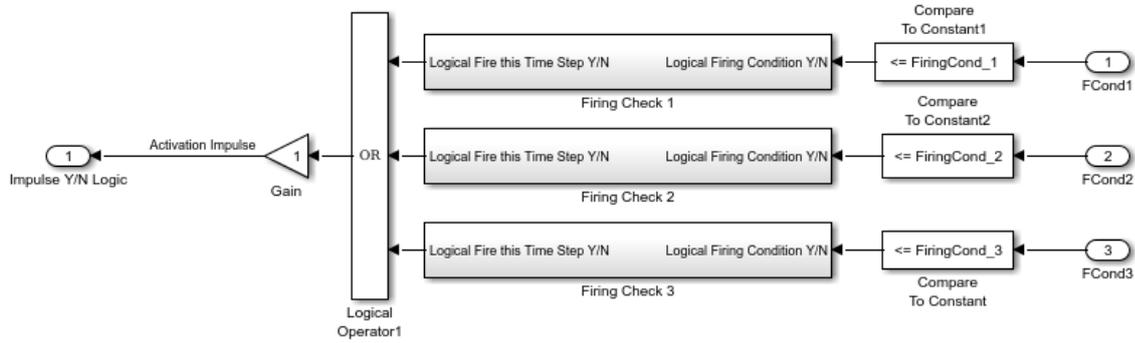


Figure 7.8: High-level Firing Logic block architecture.

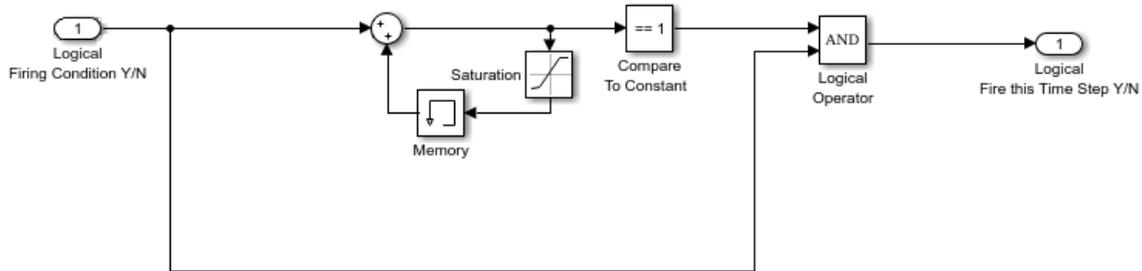


Figure 7.9: Firing Check block architecture.

that the  $t_{firing}$  is satisfied. Memory blocks store the first time the condition is met and keep feeding it back in the close loop shown in the figure. The saturation block makes sure the signal does not grow without bounds, which could eventually lead to numerical problems. The second "compare to constant" block only yields (1) for the first time the condition is met, which ultimately drives the activation of the "AND" logical operator.

#### NAVIGATION

The navigation block architecture can be seen in Figure 7.10. Here, the block makes use of the reference trajectory,  $\mathbf{x}_{ref}(t)$ , and a set of statistical parameters to create the navigation signals that are later added to the real spacecraft state vector,  $\mathbf{x}_{ref}(t)$ . This creates the spacecraft state vector as known by the navigation system,  $\mathbf{x}_{nav}(t)$ . The mission time,  $t_f$ , is discretised in periods delimited by  $t_{i,nav}$  and  $t_{i+1,nav}$ . For each of these periods, the navigation error signal is assigned a nominal  $1 - \sigma = \sigma_i$  value.

The strong dependency between the phase angle,  $\alpha$ , of the spacecraft position and the performance of the navigation system was already explained in Section 5.2. This relation was implemented in the thesis software

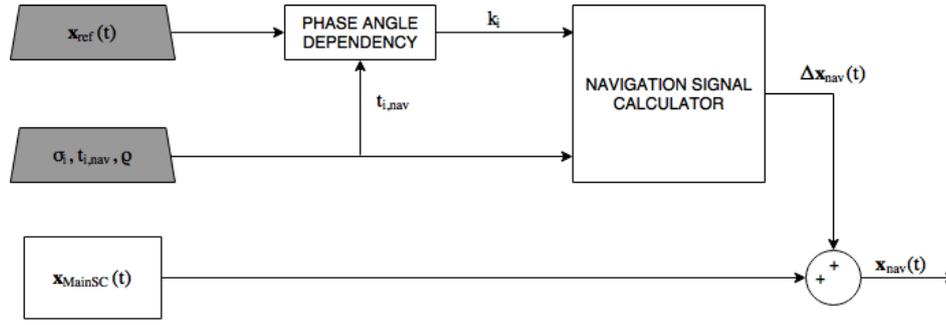


Figure 7.10: Navigation block architecture.

as a factor,  $k_i(\alpha)$ , that multiplies the  $1 - \sigma$  value used to generate the navigation error models. For every epoch,  $t_j$ , of  $\mathbf{x}_{ref}(t_j)$ , the phase angle is calculated using Equation (7.4), with  $\mathbf{a} = \mathbf{r}_{DidymainSun}$  and  $\mathbf{b} = \mathbf{r}_{ref}$ . The average phase angle  $\bar{\alpha}$  is calculated between the epochs  $t_{i,nav}$  and  $t_{i+1,nav}$  and assigned to the  $1 - \sigma$  value of that period,  $\sigma_i$ . The calculation of the factor  $k_i$  is done as:

$$k_i = \begin{cases} \left( \left( \frac{30^\circ - \bar{\alpha}}{30^\circ} \right) 0.2 + 1 \right) \sigma_i & \text{for } 0^\circ \leq \bar{\alpha} < 30^\circ \\ \sigma_i & \text{for } 30^\circ \leq \bar{\alpha} < 70^\circ \\ \left( \left( \frac{\bar{\alpha} - 70^\circ}{20^\circ} \right) 0.2 + 1 \right) \sigma_i & \text{for } 70^\circ \leq \bar{\alpha} < 90^\circ \\ \left( \left( \frac{\bar{\alpha} - 90^\circ}{10^\circ} \right) 0.2 + 1.2 \right) \sigma_i & \text{for } 90^\circ \leq \bar{\alpha} < 100^\circ \end{cases} \quad (7.5)$$

It can be seen that four different performance sectors are identified. The nominal performance is obtained between  $30^\circ$  and  $70^\circ$  because the illumination conditions are such that shadows are easily identifiable. For lower angles, there is too much sunlight and the shadows can be barely seen. For higher angles, the performance gets worse up to a 40% of the nominal performance for  $100^\circ$ . This is because the asteroid's face in the camera Field Of View (FOV) is mostly in shadows, as indicated in Pellacani et al. (2017).

The navigation signal is then created using two sampling times, set to 1000 s and 100 s. The fast one is the one responsible of the "noise" seen in the navigation plots shown in the report, whereas the the slow one is responsible for the big-magnitude error variations with time. The latter is the only one that is affected by the correlation coefficient set in the navigation signal calculator block,  $\rho$ . Two parts of the error signal calculation can be identified:

1. Slow sample time (1000 s): The time that lies between  $t_{i,nav}$  and  $t_{i+1,nav}$  is divided in segments of 1000 s. An error value is assigned to each of these segments,  $\Delta \mathbf{x}_p$ , where the subindex  $p$  identifies the different segments. This error is generated obeying an uncorrelated normal distribution with  $\mu = 0$  and  $\sigma = k_i \sigma_i$ . If there is a previous segment with an assigned error value, the value given to the next segment is correlated using the relation expressed in Equation (7.6). This happens even if the segments belong to different  $t_{i,nav}$  and  $t_{i+1,nav}$  periods (i.e. they have different associated  $\sigma_i$ ).

$$\Delta \mathbf{x}_{p,corr} = \Delta \mathbf{x}_{p-1} + \rho(\Delta \mathbf{x}_p - \Delta \mathbf{x}_{p-1}) \quad (7.6)$$

The parameter  $\rho$  was given a value of 0.18 that was obtained by a fine-tuning process to achieve the most realistic signals according to the results presented and discussed in Section 5.2.

2. Fast sample time (100 s): A new signal is created using  $\Delta \mathbf{x}_{p,corr}$  as input. A  $\Delta \mathbf{x}$  value is generated every 100 s using a normal distribution with  $\mu = \Delta \mathbf{x}_{p,corr}$  and  $\sigma = 0.1 k_i \sigma_i$ .

Note that the 10% value comes from fine-tuning process that yielded the most realistic signal.

The resulting signal from step 2 is used as the navigation signal  $\Delta \mathbf{x}_{nav}(t)$ , since it includes the results from both steps.

Note that the sample time of 1000 s corresponds to the time that is estimated to be required to generate a navigation estimation based on vision-based feature tracking. This value was presented as the rationale for a mission requirement in Chapter 2. Using a fast sample time came from the need of introducing a realistic source of noise in the signal.

#### 7.1.4. MONTE CARLO SCHEDULER

In this section, the Monte Carlo scheduler design will be explained. Its role in the whole software architecture is to launch the Simulink simulations with the generated set of parameters and save the outputs from these simulations in a way that can be post-processed and analysed easily.

It is important to note that a Monte Carlo sampling method was chosen because the system variables are extremely coupled and there is no easy way to deduct laws to describe this coupling. Between the random sampling techniques available, a Monte Carlo one was chosen over the near-random orthogonal sampling or Latin Hypercube sampling methods because of the easiness to implement the former one. Even though it is less efficient in terms of uniform coverage of the variables for the same amount of runs, the method is well accepted in the industry as a source of statistical confidence for the behaviour of complex systems.

As already explained in the beginning of this chapter, the Simulink model was implemented only using functions compatible with C++ compiling. This enabled the use of the so-called "Rapid Acceleration Simulation" option of Simulink, which together with the parallelisation of the simulations allowed for considerable increase in the number of simulations done per Monte Carlo campaign.

The parameters used in each Monte Carlo run can be seen in Table 7.1. Some of the parameters that are used in the Monte Carlo analysis are not directly characterised by a probability distribution, but a rather more complex derivation. These derivations are given in the following paragraphs. Please note that the source and rationale of the distributions were given in Chapters 2 and 3.

Table 7.1: Monte Carlo parameters.

Parameter	Distribution and Parameters
$a_{Moon}$	Normal: $\mu = 1.18$ km, $\sigma = 0.03$ km
$e_{Moon}$	Uniform: [0, 0.03]
$\theta_{Moon}(t_0)$	Uniform: [0, 360] °
$\beta_{Moon}(t_0)$	Uniform: [0, 360] °
$\beta_{Main}(t_0)$	Uniform: [0, 360] °
$m_{SC}$	Normal: $\mu = 500$ kg, $\sigma = 50$ kg
$C_R$	Uniform: [1, 2]
$\mu_{Didymos}$	See Equation (7.7).
$\mu_{Main}$	See Equation (7.7).
$\mu_{Moon}$	See Equation (7.7).
$\mu_{Main,g}$	See Equation (7.8).
$\mu_{Moon,g}$	See Equation (7.8).
$\mathbf{x}_{MainSC}(t_0)$	See Equation (7.9).
$\Delta \mathbf{x}_{Nav}(t)$	See Section 7.1.3.
$t_{firing}$	See Equation (7.10).
$\Delta \mathbf{v}_{CRSError}$	See 3 <sup>rd</sup> point of "Injection Assessment Procedure" below.

The gravitational parameter of the system is calculated using the "mass ratio",  $\epsilon$ , and "total mass of the system",  $m_{Didymos}$ , parameters given in Table 3.1. With the distributions characterised in that table, a  $\epsilon$  and a  $m_{Didymos}$  are generated for every Monte Carlo run. From those values, the different gravitational parameters can be obtained as:

$$\begin{aligned}
 \mu_{Didymos} &= Gm_{Didymos} \\
 \mu_{Moon} &= \epsilon \mu_{Didymos} \\
 \mu_{Main} &= (1 - \epsilon) \mu_{Didymos}
 \end{aligned} \tag{7.7}$$

The gravitational parameters known to the guidance system are generated with the "after DCP parameter knowledge",  $\kappa$  present in Table 3.1. The values used inside the guidance system are:

$$\begin{aligned}
 \mu_{Main,g} &= \kappa_{Main} \mu_{Main} \\
 \mu_{Moon,g} &= \kappa_{Moon} \mu_{Moon}
 \end{aligned} \tag{7.8}$$

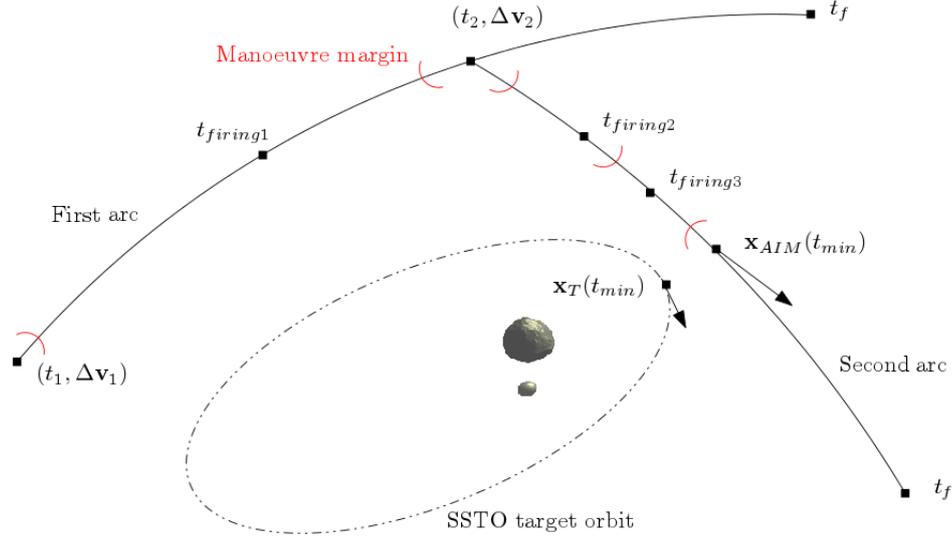


Figure 7.11: Firing times geometry.

where  $\kappa$  is generated from a distribution generated according to the parameters provided in Table 3.1.

The initial condition for the AIM spacecraft is created as:

$$\mathbf{x}_{MainSC}(t_0) = \mathbf{x}_{DCP}(t_1) + \Delta \mathbf{v}_1 + \Delta \mathbf{v}_{CommErr} + \Delta \mathbf{x}_{DCPdispersion} \quad (7.9)$$

where  $\Delta \mathbf{v}_{CommErr}$  is a function of the desired  $\Delta \mathbf{v}$  and characterised by the command errors introduced in Table 2.1, and  $\Delta \mathbf{x}_{DCPdispersion}$  is the state vector dispersion error that is characterised by the values presented in Figure 5.3 and discussed in Section 5.2.

The  $\Delta \mathbf{v}_{CommErr}$  is calculated in the [D] reference frame, defined in Section 3.2. The error is calculated by generating a magnitude error characterised by the parameter  $|\Delta \mathbf{v}_{CommErr}|$  shown in Table 2.1 and adding it to the z coordinate component of  $\Delta \mathbf{v}$  expressed in [D]. Then, this vector is rotated around the  $\hat{\mathbf{x}}_D$  by an angle characterised by the parameter  $\Delta \gamma_{CommErr}$  in Table 2.1, and finally it is rotated again by a random angle from 0 to 360° around the  $\hat{\mathbf{z}}_D$  axis. This vector is later transformed to the frame [I].

The  $\Delta \mathbf{x}_{DCPdispersion}$  is calculated in spherical coordinates in terms of the magnitude of the dispersion and the latitude,  $\lambda$ , and longitude,  $\phi$ , angles of the error vector. The magnitude is generated following a normal distribution characterised by the values discussed and given in Section 5.2. The angles can take random values between 0° and 360° for  $\lambda$ , and between -90° and 90° for  $\phi$ .

The  $\Delta \mathbf{x}_{Nav}(t)$  parameter is calculated as it was indicated in Section 7.1.3 for every Monte Carlo simulation.

The  $t_{firing}$  parameter calculation, which is only used for the tuning of the three firings of the guidance subsystem, is as follows:

$$\begin{aligned} t_{firing1} &= [0, 1](t_2 - 2 \cdot 1800) + 1800 \text{ s} \\ t_{firing2} &= [0, 1](t_{min} - t_2 - 3 \cdot 1800) + t_2 + 1800 \text{ s} \\ t_{firing3} &= [0, 1](t_{min} - t_{firing2} - 2 \cdot 1800) + t_{firing2} + 1800 \text{ s} \end{aligned} \quad (7.10)$$

where  $[0, 1]$  indicates a random number taken from a uniform distribution between 0 and 1 and the 1800 seconds margin value is taken from the mission requirement **SC-AUT-3**, as stated in Chapter 2. This equation distributes the first firing condition along the first arc, leaving margins at the beginning and at the end of the arc. The second is then set along the second arc, leaving a 1800 seconds margin just after  $t_2$  and two margins before  $t_{min}$ . The 3600 s before  $t_{min}$  are left in case the random process places  $t_{firing2}$  just at the end of the segment, so there is still room for a third manoeuvre with 1800 s margin at both sides. The third firing is set along the remaining second arc, leaving a margin after  $t_{firing2}$  and before  $t_{min}$ . Equation (7.10) geometry is depicted in Figure 7.11. Please note that the rationale as to why three firings are used will be given in Chapter 8.

The  $\Delta \mathbf{v}_{CRSError}$  parameter will be characterised in the final part of the section, where the cubesat injection assessment implementation is described.

### INJECTION ASSESSMENT PROCEDURE

To determine if the injection has been successful or not, Equation (4.1) has to be evaluated progressively.

It is assumed that the term  $\Delta \mathbf{x}_{Deployment}$  is composed only of the impulse given to the cubesat by the spring CRS and associated errors,  $\Delta \mathbf{v}_{CRS} + \Delta \mathbf{v}_{CRSErrors}$ . Note that this assessment takes place in the [O] reference frame, defined in Section 3.2.

The following procedure will assess the injection manoeuvre:

1. Position margins, ie.: first three components of Equation (4.1), are checked. From Equation (4.1), the position of the AIM spacecraft,  $\mathbf{x}_{AIM}$ , must fulfil the conditions:

$$\begin{cases} \mathbf{r}_T + \Delta \mathbf{r}_M \geq \mathbf{r}_{AIM} \\ \mathbf{r}_T - \Delta \mathbf{r}_M \leq \mathbf{r}_{AIM} \end{cases} \quad (7.11)$$

To check these requirements, the equation of an ellipsoid is used:

$$\frac{(x_T - x_{AIM})^2}{a^2} + \frac{(y_T - y_{AIM})^2}{b^2} + \frac{(z_T - z_{AIM})^2}{c^2} \leq 1 \quad (7.12)$$

where  $a$ ,  $b$ , and  $c$  are the position safety margins presented in Table 4.1 for the  $x$ ,  $y$  and  $z$  coordinates respectively.

2. As discussed in Section 4.2, the CRS injection velocity  $\Delta \mathbf{v}_{CRS}$  must be set on-ground. The way to represent this operational constraint is to set the  $\Delta \mathbf{v}_{CRS}$  after obtaining the reference trajectory. The optimum  $\Delta \mathbf{v}_{CRS}$  can be chosen so that the cubesat velocity difference with respect to the target is minimised by following Equation (7.13).

$$\begin{cases} \Delta \mathbf{v}_{CRS} = 0.06 \frac{\mathbf{v}_T - \mathbf{v}_{ref}(t_{min})}{|\mathbf{v}_T - \mathbf{v}_{ref}(t_{min})|} & \text{if } |\Delta \mathbf{v}_{CRS}| > 0.06 \text{ m/s} \\ \Delta \mathbf{v}_{CRS} = 0.04 \frac{\mathbf{v}_T - \mathbf{v}_{ref}(t_{min})}{|\mathbf{v}_T - \mathbf{v}_{ref}(t_{min})|} & \text{if } |\Delta \mathbf{v}_{CRS}| < 0.04 \text{ m/s} \\ \Delta \mathbf{v}_{CRS} = \mathbf{v}_T - \mathbf{v}_{ref}(t_{min}) & \text{if } 0.04 \text{ m/s} \leq |\Delta \mathbf{v}_{CRS}| \leq 0.06 \text{ m/s} \end{cases} \quad (7.13)$$

Note that Equation (7.13) takes into account the CRS deployment velocity upper and lower bounds expressed in Table 4.2. Also note that the  $\Delta \mathbf{v}_{CRS}$  is determined for each reference trajectory, i.e. there is one  $\Delta \mathbf{v}_{CRS}$  value for each Monte Carlo campaign.

3. The errors associated with  $\Delta \mathbf{v}_{CRS}$  are generated for every Monte Carlo run and added to the original vector,  $\Delta \mathbf{v}_{CRS} + \Delta \mathbf{v}_{CRSErrors}$ . The generation of these errors follows the same procedure as the one described previously in this section for the  $\Delta \mathbf{v}_{CommErr}$ , but with the distribution parameters found in Table 4.2.
4. As a last step, the cubesat deployment is considered successful if:

$$\begin{cases} \mathbf{v}_T + \Delta \mathbf{v}_M - \mathbf{v}_{AIM} \geq \Delta \mathbf{v}_{CRS} + \Delta \mathbf{v}_{CRSErrors} \\ \mathbf{v}_T - \Delta \mathbf{v}_M - \mathbf{v}_{AIM} \leq \Delta \mathbf{v}_{CRS} + \Delta \mathbf{v}_{CRSErrors} \end{cases} \quad (7.14)$$

To check the velocity safety margins expressed in Table 4.1, the following formulation of Equation (7.14) is used:

$$\begin{cases} \text{Velocity magnitude margin} \geq |\mathbf{v}_{ASPECT}| - |\mathbf{v}_T| \\ \text{In-plane angle} \geq |\arctan 2(\dot{y}_{ASPECT}, \dot{x}_{ASPECT})| \\ \text{Out-of-plane angle} \geq \left| \arctan 2(\dot{z}_{ASPECT}, \sqrt{\dot{x}_{ASPECT}^2 + \dot{y}_{ASPECT}^2}) \right| \end{cases} \quad (7.15)$$

where  $\mathbf{v}_{ASPECT}$  is the velocity of the cubesat after the deployment  $\mathbf{v}_{ASPECT} = \mathbf{v}_{AIM} + \Delta \mathbf{v}_{CRS} + \Delta \mathbf{v}_{CRSErrors}$ .

## 7.2. VERIFICATION AND VALIDATION

To prove that the software is ready to answer the research questions, the tool that has been implemented for this thesis must undergo V&V. Besides, those blocks that are considered commercial software must undergo an acceptance test. In this section, these processes applied to the different blocks of the model, and then to the whole integrated system, will be described.

### 7.2.1. VERIFICATION

Besides from the high-level blocks described in Section 7.1, there are a high number of lower-level unit blocks that are frequently used across the different high-level functional blocks. These individual and single-function blocks underwent a verification process that is summarised in Table 7.2. Note that these functions were described in Section 7.1.

Table 7.2: Individual function block verification summary.

Function	Method	Status
Reference frame transformations	Random vectors were transformed and checked against hand calculations.	Verified
Monte Carlo Scheduler	A matrix of parameters was fed into the Simulink model and the parameter values were checked from the simulation output file.	Verified
SSTO Calculator	Output state vector's orbit was used to evaluate Equation (4.3), checked and compared to expected values.	Verified
Kep. Elements to Cartesian Cartesian to Kep. Elements	Visual inspection of hand-calculated output-input relations.	Verified
Dynamical Perturbations	Perturbations modelled were particularised for the Earth case and compared to the values given in Wakker (2015).	Verified
Firing Logic	A simulation clock signal was fed into the block and the logic response was plotted against time to check the consistency.	Verified
Command Errors	The magnitude and angle of the calculated command error vector was checked against statistically generated values.	Verified

### DYNAMICS BLOCK

In this section, the verification carried out for the dynamics block will be explained. As mentioned in Section 7.1, the dynamics block is formed by a main 2BP with different perturbations added, TBP from the Moon and Sun, SRP and NSP from Didymain and the moon.

For the verification, results obtained for a target state vector propagated during five days under the different perturbations were compared to the same propagations done by the NASA's open-source software General Mission Analysis Tool (GMAT).

The orbit parameters used for the verification are shown in Table 7.3. These parameters are set so that the numbers obtained in the verification process are representative of the ones that will be obtained during the thesis simulations. Typical propagation times for the thesis assignment will range from a few hours to a couple of days, and typical distances of the spacecraft to the Didymos system will range from tens of kilometres to as close as three kilometres. As it is seen, conservative parameter values have been set for the verification simulations.

Since the TBP from the moon could not be set up in GMAT (there were no SPICE ephemeris files available for Didymoon), an analytical implementation of the TBP was used following the equations found in Dell'Elce

Table 7.3: Dynamics block verification parameters.

Parameter	Value
Simulation time	5 days
$a_{Moon}$	1180 m
$a_{SC}$	3000 m
$e_{Moon}$	0.01
$e_{SC}$	0
$i_{Moon}$	0°
$\theta_{0,Moon}$	0°

Table 7.4: Dynamics block verification results.

Perturbation	Deviation from <b>GMAT</b> result
<b>2BP</b>	$\ll 1$ mm
<b>TBP Moon</b>	$< 1$ cm
<b>TBP Sun</b>	$< 1$ mm
<b>SRP</b>	$\ll 1$ mm
<b>NSP Main</b>	$< 0.1$ m
<b>NSP Moon</b>	Non-tested

et al. (2016).

It must be noted that the verification results shown in Table 7.4 do not include **NSP** from Didymoon because **GMAT** does not allow such perturbation. However, numerically speaking, the block used for this perturbation is exactly the same one as the one used for the **NSP** from Didymain. Since the verification includes this latter block, the **NSP** from the moon is considered verified together with it. From what is seen in the table, the propagation accuracies obtained are satisfactory, since they are well below the resolution required to assess the cubesat safety deployment intervals, which are close to 200 m. The block is considered verified.

To further test the implementation of the dynamics block, the results from Dell’Elce et al. (2016) as far as the stability of the **SSTO** of the system were used as reference for the following set of test simulations.

The test objective is to prove that the current system model can simulate the behaviour of a spacecraft in the vicinity of the system and, despite the differences between the two models (that are partly due to a more accurate system characterisation), still be coherent with respect to the results presented in the work by Dell’Elce et al. (2016), whose results are considered relevant for the **AIM** mission development. It could be considered a validation of the dynamics block.

Moreover, most of the misalignments between the two models (paper vs thesis) are more relevant for extremely close approaches to the system, since they are gravity related. This fact implies that the results obtained with the developed software should be relatively similar for distances representative of the ones that will be found during the thesis simulations, ranging from 3 to 10 kilometres.

In Figure 7.12, failure rates for different **SSTO** in the Didymos system are presented. The relevant values from this plot are also found in the second column of Table 7.5 and can be compared to the values obtained with the software coded for the thesis in the third column of the same table. It can be seen that the relative difference between results is bigger the closer the spacecraft is to Didymain, and that there is a perfect match for orbits beyond 9 km. Please note that the initial condition for the spacecraft are set according to the **SSTO** conditions expressed in Section 4.2.

This behaviour was predicted previously in this section, and the difference in magnitude is explained by the different characterisations given to the system in the two models:

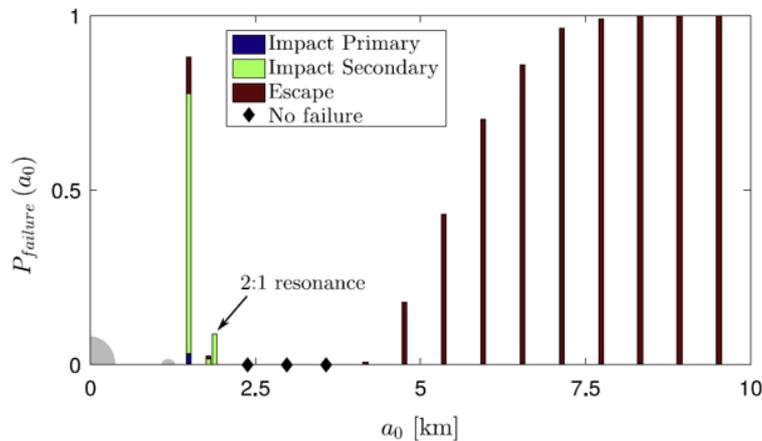


Figure 7.12: **SSTO** stability for different semi-major axis values (Dell’Elce et al., 2016).

Table 7.5: Dynamics block test results (1000 runs).

Semi-major axis [km]	Failure Dell'Elce et al. (2016) [%]	Failure Dynamics block [%]
1.8	2.4	0.0
4.7	13	9
6.25	75	70
9.0	100	100

- Dynamical parameters, including gravitational shape of the asteroids, are slightly different in both implementations: in the work by Dell'Elce et al. (2016), they use a polyhedron gravity model. The ones used for the thesis come from an ulterior characterisation activity of the binary system, thus they are expected to be more accurate.
- The full coupling between the gravity fields of both asteroids has not been implemented in the software for the thesis but it was included in the work by Dell'Elce et al. (2016). This coupling strongly affects the NSP from the secondary and primary and the eccentricity of the system.

Besides these two strong differences in the modelling, there are other factors to consider when assessing if the test was successful:

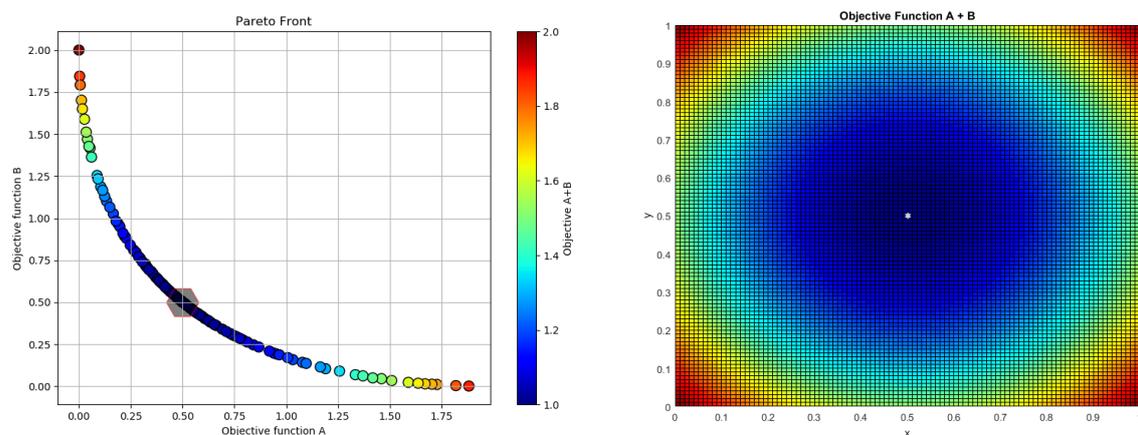
- Trajectories analysed in the test are not the ones that will be flown during the thesis.
- Typical propagation times for the thesis simulations will range from a few hours to a couple of days at maximum, whereas the simulation time for this test was 400 days.

These two coupled factors have a huge impact in the effect that this difference in results shown in Table 7.5 has on the test of the block. Trajectories flown in the thesis will only be as close as 3 km with respect to the system for few hours at most, so the big behavioural difference at this distance for 400 days will never become a relevant effect in the thesis.

In conclusion, the dynamics block of the software is considered verified and ready to fully represent the longitudinal dynamical behaviour of a spacecraft in the vicinity of Didymos in agreement with recent study models of the system.

### OPTIMISATION BLOCK

The tool that was chosen for the optimisation tasks in this thesis is MIDACO. Since this is a commercial tool, it is considered already V&V. The only test to be performed corresponds to an acceptance test that confirms



(a) Pareto Front of the acceptance test function.

(b) Objective function values map.

Figure 7.13: Acceptance test results.

that the user is aware of the functionality of the toolbox and that this toolbox performs as expected. For this purpose, a simple multi-objective problem will be solved with **MIDACO**.

$$f_A = x^2 + y^2 \quad (7.16)$$

$$f_B = (x-1)^2 + (y-1)^2 \quad (7.17)$$

And combining the two objective functions in Equations (7.16) and (7.17) into the one shown in Equation (7.18).

$$f_T = f_A + f_B \quad (7.18)$$

To prove that the minimum point found by the software is accurate, the analytical solution to the problem will be briefly discussed. From the local minima conditions, the possible optimum point can be found as shown in Equation (7.19).

$$\frac{\partial f_T}{\partial \mathbf{x}} = \mathbf{0} \Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} \quad (7.19)$$

Now, the sufficient condition for a minimum states that the Hessian matrix must be positive definite. This can be checked as seen in Equation (7.20).

$$\frac{\partial^2 f_T}{\partial \mathbf{x}^2} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \Rightarrow \lambda_{1,2} = 4 \quad (7.20)$$

Since both eigenvalues of the matrix are positive, it can be concluded that the Hessian of the target objective function is positive definite, and thus, the point obtained in Equation (7.19) is a local minimum.

The results obtained with **MIDACO** can be seen in Figures 7.13a and 7.13b. From these plots, and comparing the results to the analytical ones, it can be concluded that the acceptance test has been successfully carried out.

Whether the target problem was well implemented in **MIDACO** was checked by visual inspection of the results yielded by the tool after one iteration for certain initial conditions. Due to the tool internal architecture, the first iteration only evaluates the objective function and constraints for the initial conditions, which allows to check their expected values.

#### GUIDANCE BLOCK

To verify the guidance subsystem, its behaviour in a simple case was tested. A particle in a constant force field was guided along a reference trajectory. This reference trajectory was created in the same forcefield but with an offset in its initial conditions.

Since the force field is constant, all the components of its gradient are equal to 0. This implies that, following the same procedure as the one followed in Section 5.1, the **STM** for this forcefield can be expressed as indicated in Equation (7.21).

$$\Phi = \begin{bmatrix} \mathbf{I} & \mathbf{I} \cdot t_{go} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (7.21)$$

The initial conditions for the two particles can be seen in Equations (7.22) and (7.23).

$$\mathbf{x}_{Ref} = ( 0 \ 0 \ 0 \ 10 \text{ m/s} \ 0 \ 0 )^T \quad (7.22)$$

$$\mathbf{x}_{Chaser} = ( 100 \text{ m} \ 0 \ 0 \ 11 \text{ m/s} \ 0 \ 0 )^T \quad (7.23)$$

Because of the form of the **STM** and the lack of uncertainties in the dynamical model, only one corrective manoeuvre is required to hit the target at the nominal end-of-mission time. This is shown in Figure 7.14. It can be seen how the dispersion grows at a rate of 1 m/s from a position offset of 100 m, which is the difference in the initial conditions, and how the guidance subsystem corrects this growing dispersion with just one manoeuvre at  $t_{impulse} = 300$  s.

After the test conducted, the guidance subsystem is considered verified.

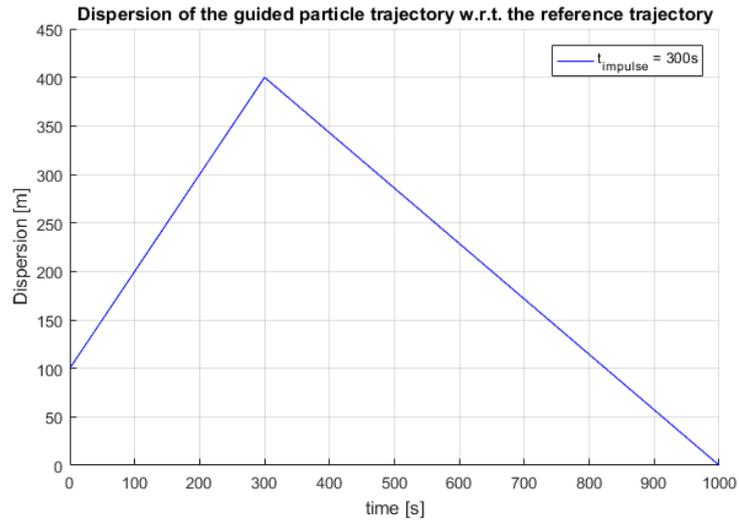


Figure 7.14: Dispersion of the simulated particle trajectory with respect to the reference one.

### NAVIGATION BLOCK

As stated in Section 7.1, the function of the navigation block is to mimic the error signal that the navigation subsystem of a spacecraft such as the one flown in this mission would have during close proximity operations around a binary asteroid. This block receives an input epoch which is used to query the navigation signal previously generated, obtaining the corresponding navigation error. The verification of this block implies verifying that the signal created is coherent with the inputs given for its generation and with a reference navigation model.

The first step described in the previous paragraph was done by calculating the  $1-\sigma$  and mean of the distributions used as base for the navigation errors. These values were compared to those used as input for the model generation. This part of the verification was successful.

To make the navigation signal more realistic and as explained in Section 7.1.3, variability of this error as a function of the Sun-Asteroid-Spacecraft phase angle ( $\alpha$ ) was implemented. To verify this feature, a signal of  $\alpha$  was manually generated and its effect on the signal was checked.

As it can be seen in Figure 7.15, for a monotonously increasing  $\alpha$  value, the factors that scale the  $1-\sigma$  value used to create the navigation signals correspond to the expected ones, as it would be obtained from Equation (7.5).

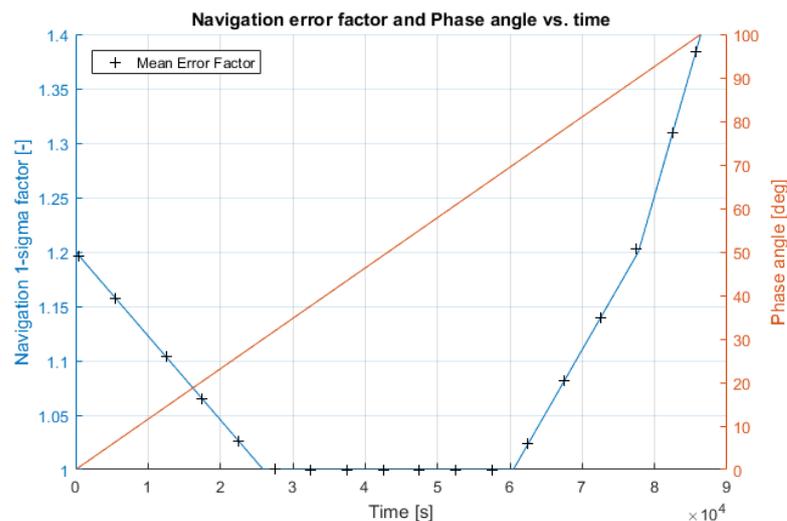
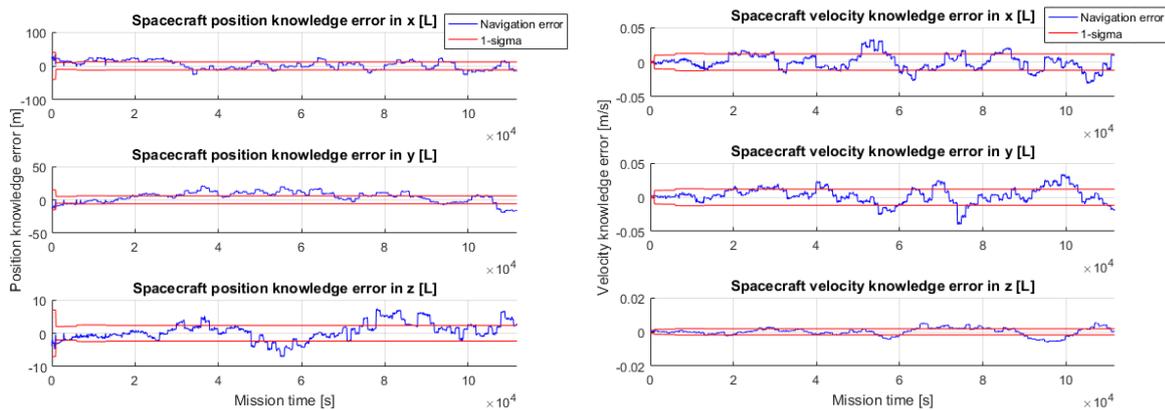


Figure 7.15: Navigation error as a function of  $\alpha$  verification.



(a) Spacecraft position navigation signal.

(b) Spacecraft velocity navigation signal.

Figure 7.16: Navigation signal for the reference trajectory targeting an SSTO with  $a_{SSTO} = 5$  km.

Note that the black crosses in Figure 7.15 correspond to the mean factor values that will be used for the generation of the actual error signal. This is done in this way because the error signal generation process is discretised in a finite number of epoch intervals, as explained in Section 7.1.3.

The next step consisted in comparing the main characteristics of a generated error signal to a reference navigation error signal taken from the literature. These two signals can be seen in Figures 7.16 and 5.5.

Both plots share the characteristics of optical camera based feature tracking navigation systems, which were explained in Section 5.2. It is worth noting that the  $1\text{-}\sigma$  values observed in Figure 7.16 include the  $\alpha$  dependency with respect the real reference trajectory that will be described in Section 8.2.

The navigation performance model is considered verified.

### 7.2.2. SYSTEM TEST

As a last step for the V&V of the software tool, a system test was carried out in substitution of what is usually called system validation. A substitute is required because there are no available reference results to use that are relevant for all the subsystems integrated in the model.

However, since it is crucial to determine whether the tool has been well implemented and whether it will be able to answer the research questions, a relatively similar test must be designed and conducted.

Prior to the test, all the blocks were integrated in the Simulink environment and the interfaces between

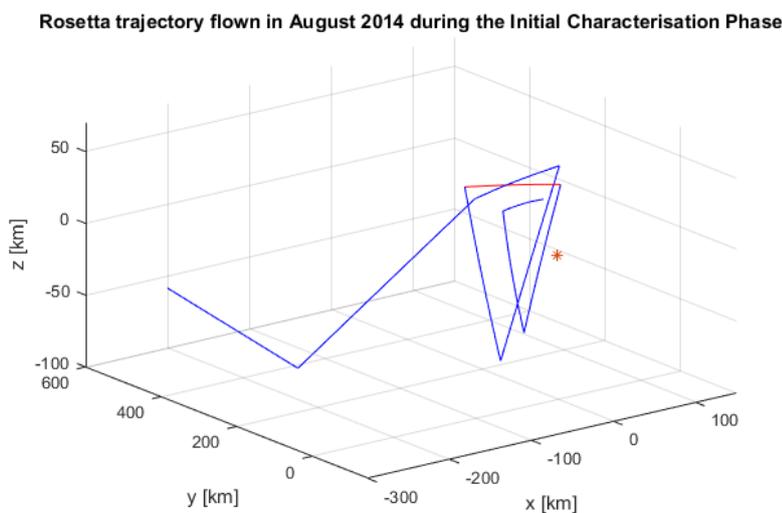


Figure 7.17: Initial characterisation phase Rosetta trajectory around the comet 67P in August 2014.

Table 7.6: Dynamic parameters for the system test scenario.

Parameter	Value
$\mu_{67P}$	$666.226 \text{ m}^3/\text{s}^2$
$m_{Rosetta}$	1613 kg
$s_{Rosetta}$	$66 \text{ m}^2$
$C_R$	1.5

them were verified as indicated in the previous section.

From what has been discussed before, the test objective was to prove that the software is able to model a real spacecraft translational behaviour in a complex dynamical environment with a guidance system that is robust to this environment and is able to guide the spacecraft along a desired reference trajectory.

The test used one of Rosetta trajectories flown during its Initial Characterisation Phase (ICP) in August 2014. The trajectory's state vectors were retrieved from COSMOS, ESA's SPICE website and integrated in the system with the SPICE tool from the Navigation and Ancillary Information Facility (NAIF).

The actual Rosetta trajectories took place in a dynamically complex environment with perturbations difficult to characterise and that are not included in the software developed for this thesis. The misalignment between the dynamics simulated and the real ones will make the spacecraft trajectory depart from the reference one taken from SPICE. The guidance system will therefore try to correct these dispersions in the state vector, in a similar way (and for similar reasons) as in the simulations that will be conducted in the thesis project.

The trajectory that is going to be used as reference can be seen in Figure 7.17 in red. It corresponds to the sixth hyperbolic arc flown during the ICP of the Rosetta mission, which greatly resembles to the trajectories that will be simulated in this thesis.

The dynamical model used to model the comet 67P and the Rosetta spacecraft can be found in Table 7.6. The ephemeris for the comet 67P were also retrieved from SPICE NAIF. The only perturbations included in the simulated dynamical model were the SRP and the TBP coming from the Sun.

In Figure 7.18 the dispersions caused by the differences in the dynamical models can be seen for a case where no corrective manoeuvres were applied and for a case where three corrective manoeuvres were applied. As it can be seen, for a relatively short period of time as the one considered (slightly less than three days), the dispersions reach a relatively high value of 40 m when they are not corrected. However, three impulsive manoeuvres are enough to reduce this dispersion to 5 m.

Results obtained from this test were consistent with the implementation, and proved that the latter one had been done successfully. The tool is considered tested at system and subsystem level and able to answer the research questions stated in Chapter 1.

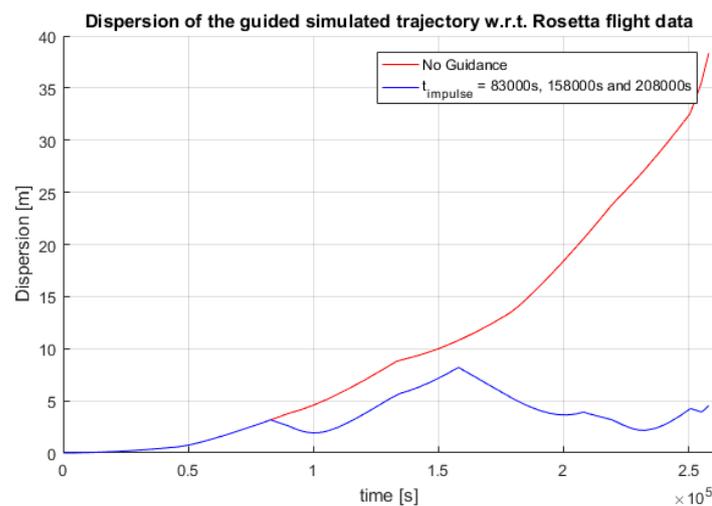


Figure 7.18: Deviations from the reference trajectory for a non-guided propagation (red) and guided one (blue).

# 8

## RESULTS

In this chapter, the results of the thesis simulations and the discussion about them will be presented. In Section 8.1 the simulation set-up will be given. In Section 8.2 the generation of the reference trajectories will be described. In Section 8.3, the timing of the stochastic manoeuvres will be tuned. In Section 8.4, the safety of the trajectories will be assessed. Finally, in Section 8.5, the cubesat deployment success rate for a final guidance configuration will be covered.

### 8.1. SIMULATION PLAN

The simulation plan to achieve the final results consists of basically a first trajectory design and optimisation task followed by three different sets of Monte Carlo campaigns.

The simulation plan is as follows:

1. To evaluate the success rate of the deployment of the ASPECT cubesat, a trajectory must be designed that takes the AIM mothercraft from a point of the previous phase to the deployment one. Because of this, the first task is to design a trajectory that fulfils all the requirements and constraints that were presented in Section 4.1. The optimisation architecture and the MIDACO parameter choices follow what was described in Section 7.1.2. In particular, the presented results have a SEED parameter value of 0. Besides, the nominal parameters for the spacecraft and the dynamical system, which can be found in Tables 2.1 and 3.1, will be used. The reference trajectories obtained can be found in Section 8.2.
2. Once the reference trajectories are obtained, the uncertainties and dispersions of the system make the deployment of the cubesat infeasible without the presence of guidance correction manoeuvres. The need of stochastic guidance manoeuvres also unveils the need to determine the best times to carry out these manoeuvres, since they drive the accuracy achieved at the deployment epoch. A Monte Carlo campaign with all the system parameters as presented in Table 7.1 will be launched. From this, the statistically optimal number of manoeuvres and the epochs at which they should be carried out,  $t_{firing}$ , will be chosen.
3. Every generated trajectory must be safe to thruster malfunctions. The safety of the transfer orbit is assessed as a function of the distance to the escape velocity,  $|\mathbf{v}| - |\mathbf{v}_{esc}|$ , and to Didymain,  $d_{MainSC}$ , and Didymoon,  $d_{MoonSC}$ . A Monte Carlo campaign is launched, in which the first injection  $\Delta\mathbf{v}_1$  is scaled with a factor ranging from 1 to 0 to see its effect on the safety margins. The system parameters will be set as presented in Table 7.1, except for the  $t_{firing}$ . The same process is repeated with failure only during the second deterministic impulse,  $\Delta\mathbf{v}_2$ . For the latter one, the stochastic correction at  $t_{firing_1}$  will be assumed perfectly given. This assessment should prove that injecting into the PDP following the trajectories generated is safe in the sense that the spacecraft can trigger a safe mode before a collision.
4. A final Monte Carlo campaign with all the system uncertainties will be launched. The firing times,  $t_{firing}$ , for the guidance stochastic corrections will be set according to the previous simulation campaign. The system parameters will be set as presented in Table 7.1, except for  $t_{firing}$ . From this assessment, the cubesat deployment success rate for the different trajectories will be calculated making use

of the values presented in Table 4.1. Furthermore, the sensitivity analysis of the cubesat injection conditions to the uncertainties, dispersions and command errors will be made, to identify the main cause of dispersions in both deployment position and velocity.

All the simulations take place using initial conditions from the DCP history file provided by ESA. This file includes the full-phase history (14 days) that will take place from midnight 01-07-2022 until midnight 15-07-2022, both UTC. It must be noted however, that this DCP will take place after a first full DCP cycle, which will begin 14 days before and that can not be interrupted by any activity that belongs to another phase.

Every Monte Carlo campaign was composed of 2000 simulations with the parameters for each of the campaigns specified at the beginning of this section. The number of simulations was maximised as to increase the statistical consistency of the results presented. Especially relevant is to note that statistical parameters shown in the plots of this chapter are calculated for batches of 200 samples.

Propagation ends when the spacecraft reaches a distance of 10 km again or the simulation time hits  $t_f = 72$  h. These requirements were stated in Chapter 2 and explained in Section 4.1.

Collisions with Didymain and Didymoon are checked at every epoch of the simulation. If one of them takes place, the simulation is interrupted and a warning flag is recorded in the log.

Other aspects of the simulation environment were already described in Section 7.1 or will be specifically discussed in the following sections.

## 8.2. REFERENCE TRAJECTORY DESIGN AND OPTIMISATION

In this section, the reference trajectories obtained in the optimisation process will be presented.

Please note that the rationale as to why deployment points of SSTO of  $a_{SSTO}$  equal to 3, 5 and 7 km was given in Section 4.2. Also in that section, the reasons why the pericentre was chosen as the deployment point in all of them were also given. Trajectories that lead to deployment points with distance and velocity differences lower than 200 m and 8 cm/s respectively will be assumed to be potentially feasible, although lower values are expected. The distance value comes from Table 4.1, and the velocity one from the magnitude error indicated in the same table plus the maximum CRS velocity increment,  $\Delta v_{CRS}$ , indicated in Table 4.2.

The requirements and constraints that apply to the trajectory design and optimisation process were presented in Section 4.1, and their implementation in the software was treated in Section 7.1.2.

As already mentioned in Section 6.1, MIDACO algorithm is a metaheuristic optimisation method. As such, the trajectories identified during the optimisation process and shown in this section should be considered neither the "best" nor the only solutions that satisfy the requirements and constraints that apply to their generation. There is however confidence in that they represent acceptable solutions that satisfy all the constraints and could be used as reference trajectories in a real-operations scenario. Besides, they are also assumed to be "good" solutions in the sense that the convergence of the optimisation algorithm around a global best solution was used as a stopping criteria.

Although later discarded, the first optimisation campaign aimed at finding trajectories that only use one arc during the entire mission. For these runs, the same constraints as the ones described in Sections 4.1

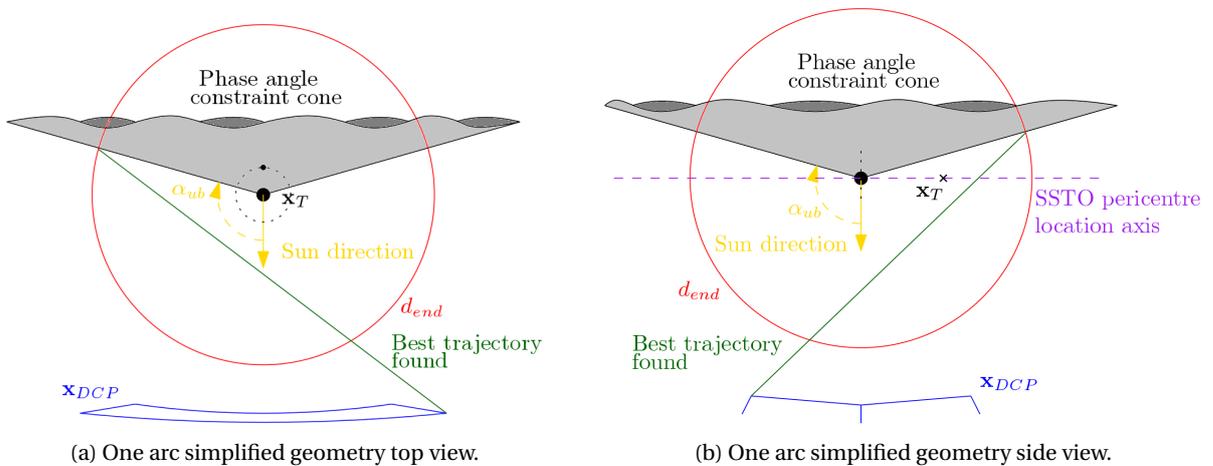


Figure 8.1: One arc geometric limitations.

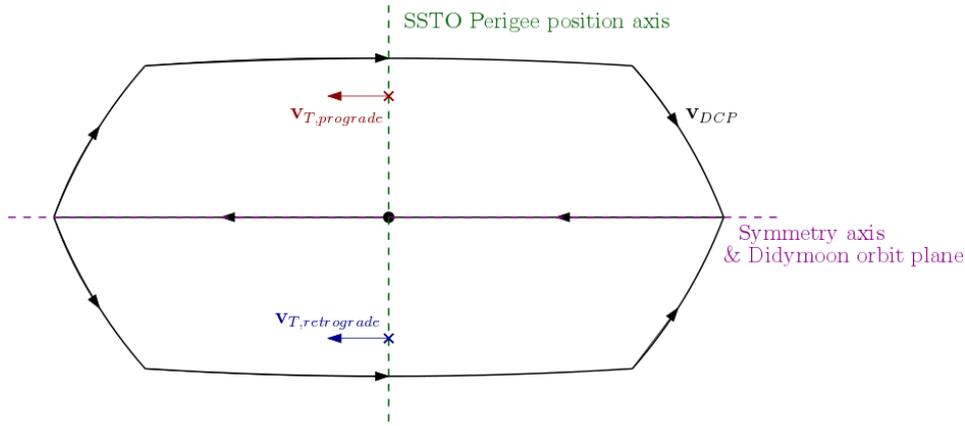


Figure 8.2: Prograde and retrograde SSTO deployment conditions symmetry.

and 7.1.2 applied. However, after the first runs, it was proven that the phase angle and the end-of-mission minimum distance constraints were too severe to allow trajectories to get closer than 1 km from the target deployment point. This could be argued from a strictly geometric point of view, since the DCP trajectories are too far from the system to allow a deployment in the terminator plane without hitting the phase angle constraint cone before leaving the 10 km sphere around the system. A simplification of this geometry is shown in Figure 8.1, where a hypothetical "best trajectory" under these conditions is represented in dark green. It can be seen that the minimum distance between the trajectory and the target point,  $\mathbf{x}_T$ , is driven by the phase-angle and the end-of-mission distance constraints.

Also note that even in the case that no end-of-mission minimum distance was required, the one arc approach would have had problems satisfying the position and velocity deployment conditions at the same time. The velocity at the deployment epoch is required to be almost parallel to the target one for the deployment to be feasible, which is also incompatible with the one arc geometry. This fact will become more clear when the results for the two-arcs trajectories are presented.

In the trajectory generation process, only the pericentre of prograde SSTO will be targeted. A SSTO is called prograde in this report when its angular momentum vector,  $\mathbf{h}$ , points towards the Sun. This decision was taken based on the symmetry that the DCP trajectory (used as initial condition for the PDP trajectory generation) offers with respect to the orbital plane of Didymoon and the target deployment state vector,  $\mathbf{x}_T$ . As seen in Figure 2.2, during the DCP the spacecraft describes a hexagon-shaped trajectory that can be decomposed in a lower and upper trapezoids. If seen from the Sun-asteroid direction, the lower trapezoid is first flown by the spacecraft in clock-wise direction. When the first one is completed, the spacecraft describes the upper trapezoid in a counter-clock-wise direction. As shown in Equation (4.3), the pericentre of the target SSTO changes from the upper to the lower side of the orbital plane between the prograde and retrograde cases which leaves the velocity direction unchanged. This means that the exact same trajectory generation conditions can be found by injecting the spacecraft from DCP trajectory points that are symmetric with respect to the plane described by the orbit of Didymoon when going from the prograde to the retrograde case, as shown in Figure 8.2. This rationale was tested by optimisation runs that proved it right by finding only solutions that were symmetric to the ones reported in this section.

In Figure 8.3, the Pareto fronts obtained for the global and local optimisation of a trajectory that deploys the cubesat at the pericentre of an SSTO with  $a_{SSTO} = 3$  km is shown. In Figure 8.3a, three different predominant colors can be seen in the global run of the MIDACO algorithm. These colors link the solutions found with different  $\Delta v$  values, which can also be assumed to identify different "families" of solutions. The term "family" is used here to refer to trajectories that are similar in all their optimisation variables. The optimisation algorithm usually gets stuck in these families and "travels" along them, which ends up describing Pareto fronts as the ones seen in Figure 8.3a in the cases where these families are not dominated by others found later.

It must be noted that the solutions described by the family of red dots in Figure 8.3a is of no operational interest due to its large distance and velocity differences, and could have been discarded straight away. Its presence can be explained by the extremely low  $\Delta v$  that the trajectories in that family present, which clearly dominates any operationally feasible solution.

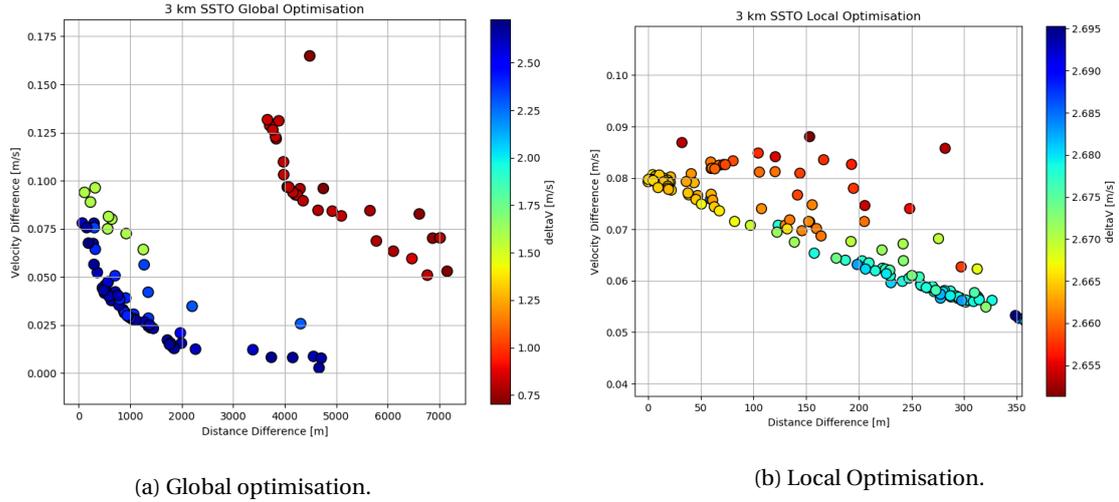


Figure 8.3: Reference trajectory optimisation for ASPECT deployment in a prograde SSTO with  $a_{SSTO} = 3$  km.

The family of feasible solutions is plotted in blue, with  $\Delta \mathbf{v}$  around 2.6 m/s. A third family, cheaper in terms of  $\Delta \mathbf{v}$  but not promising in terms of velocity difference magnitude is seen in green.

From the results seen in Figure 8.3a, trajectories that belong to the operational feasible family were chosen to carry out a local optimisation campaign, whose results can be seen in Figure 8.3b.

From there, the trajectory that will be used as reference when targeting a deployment at a SSTO with  $a_{SSTO} = 3$  km is chosen. Since the point chosen according to the weighted global objective function (described in Section 7.1.2) is close to the upper allowed margin for the deployment velocity difference magnitude, another "backup" trajectory of the same family was chosen. For this secondary trajectory, miss-distances of up to 100 m were allowed, so the velocity difference magnitude could go slightly lower.

This "sacrifice" in end-position accuracy is justified by the enhanced performance that the guidance shows at following the reference position as opposed to following the reference velocity. The rationale behind this idea was already explained in Section 5.1 and will be further developed in the following sections.

Both trajectories and the objective function values associated to them are shown in Table 8.1, where  $f_1$ ,  $f_2$  and  $f_3$  are the values shown in the x-axis, y-axis and color bar of Figure 8.3, respectively. Also note that the  $t_{min}$  value indicates the epoch at which the deployment takes place. From what was shown in Figure 4.3 and what was discussed in Section 4.2, there is a lower bound for the magnitude of the difference between the spacecraft velocity and the deployment one. This lower boundary can be read from Figure 4.3 to be close to 0.038 m/s. To this value, one must add the 0.02 m/s margin that is imposed by the requirements specified in Chapter 2. This means that theoretically, the optimisation could have reached a value as low as 0.058 m/s, but

Table 8.1: Chosen reference trajectory initial conditions and objective function values for  $a_{SSTO} = 3$  km, expressed in [I].

Trajectory	$a_{SSTO} = 3$ km, Nominal	$a_{SSTO} = 3$ km, Backup
$\Delta \mathbf{v}_1$	$\begin{bmatrix} 0.963 & -0.715 & -0.636 \end{bmatrix}^T$ m/s	$\begin{bmatrix} 0.963 & -0.715 & -0.636 \end{bmatrix}^T$ m/s
$t_1$	1073296.6 s	1073235.9 s
$\Delta \mathbf{v}_2$	$\begin{bmatrix} -0.734 & 0.875 & 0.634 \end{bmatrix}^T$ m/s	$\begin{bmatrix} -0.734 & 0.876 & 0.642 \end{bmatrix}^T$ m/s
$t_2$	6767.5 s	6768.5 s
$f_1$	9.4 m	97.1 m
$f_2$	0.078 m/s	0.071 m/s
$f_3$	2.664 m/s	2.669 m/s
$t_{min}$	16950.1 s	16829.2 s

instead both the nominal and backup trajectories have values well above this lower limit. The explanation for this can be found in Figure 8.4, where the constraints applied during the optimisation process are presented as a function of the mission time for all the reference trajectory families that will be identified in this section.

The trajectory for  $a_{SSTO} = 3$  km is limited by the distance to Didymain and Didymoon and the margin to the escape velocity constraints. These three constraints reach the limit value at some point during the mission time, which indicates that the optimisation algorithm is moving along that constraint in its search for the optimum point. From the fact that MIDACO did not find any result better than the ones presented here, it can be concluded that the constraints mentioned were limiting the search space too much, either by leaving outside "better" points or by creating a non-convex search space in which the algorithm could not advance further within the allowed maximum number of function evaluations explained in Section 7.1.2.

In Figure 8.5, the optimisation process for the reference trajectory targeting a SSTO with  $a_{SSTO} = 5$  km is shown. When compared to the results obtained for  $a_{SSTO} = 3$  km, it can be seen that the convergence of the algorithm to the "best" family is quite straightforward, since the process does not generate other non-dominated families. In Figure 8.5b, the local optimisation run used to choose the point that will be used as reference trajectory can be seen. It must be noted that MIDACO features a local optimisation mode that switches on when the algorithm detects that has found a relatively convex region potentially containing optimum points, as it was explained in Section 6.1. For this reason, the local optimisation runs carried out aimed at increasing the density of the Pareto fronts in the objective function margins desired rather than actually making the algorithm advance further.

In Table 8.2 the trajectory chosen for the deployment at  $a_{SSTO} = 5$  km is presented. This trajectory gets closer to the objective function theoretical value predicted by the lower limit shown in Figure 4.3. For  $a_{SSTO} = 5$  km, this limit is slightly higher than 0.05 m/s which is similar to the achieved 0.051 m/s. In this case, the safety distance to Didymain is not an active constraint for the family found, what allows the injection of the spacecraft into a second arc whose velocity gets much closer to the desired one at the deployment point.

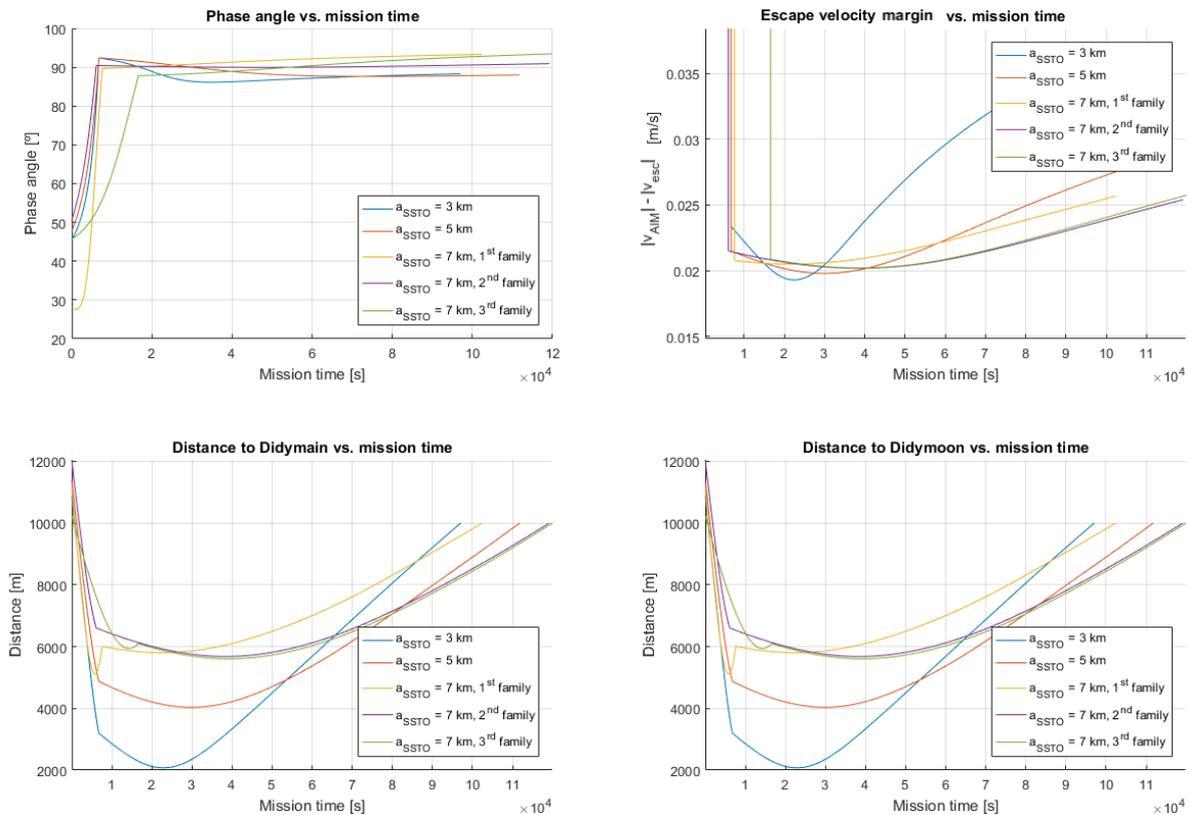


Figure 8.4: Reference trajectory families constraint values.

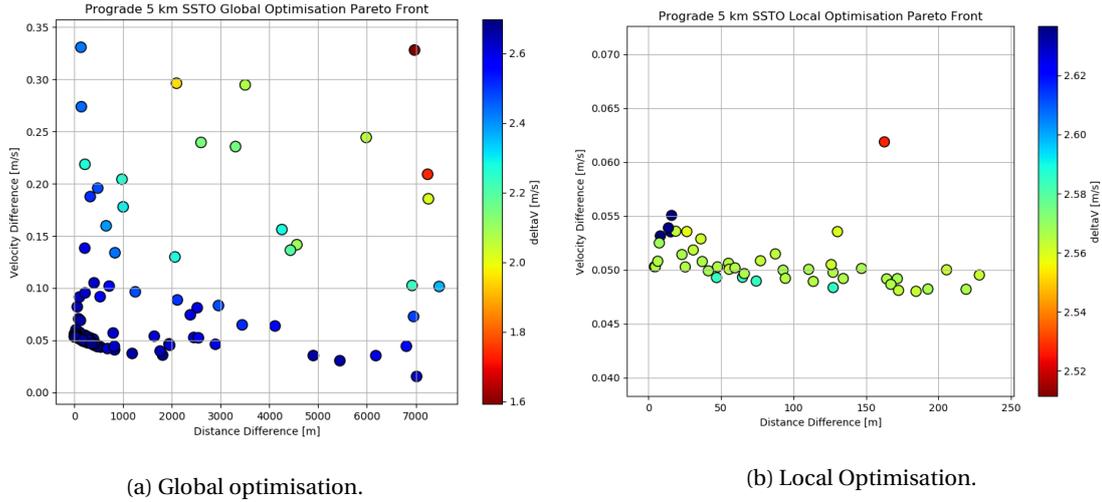


Figure 8.5: Reference trajectory optimisation for ASPECT deployment in a prograde SSTO with  $a_{SSTO} = 5$  km.

However, it must be noted that the spacecraft margin to the escape velocity is still an active constraint for this family of trajectories, as seen in Figure 8.4.

In Figure 8.6, the Pareto fronts associated to the global and local optimisation runs for trajectories that allowed the cubesat deployment at  $a_{SSTO} = 7$  km are shown. From what can be seen in the figure, three different families are identified. In contrast with the alternative families found for the case  $a_{SSTO} = 3$  km, all of these families have potential to achieve a successful deployment since their distance and velocity differences are within the upper margins that were specified for the cubesat deployment.

The first family, presented in dark blue, has  $\Delta \mathbf{v}$ 's of around 3.5 m/s. This family also presents the "best" objective function values in terms of position and velocity difference. Again, the lower limit for the velocity difference, found around 0.047 m/s as deduced from Figure 4.3, is accurately met by this family, which presents values around 0.048 m/s for close-to-zero distance differences. This is again due to the lack of active constraints besides from the one associated to the margin to the escape velocity one, as seen in Figure 8.4. The local optimisation Pareto front for this family can be seen in Figure 8.6b. Note that this local run was carried out to increase the density of the Pareto front around close-to-zero distance differences values. This was not required for the other families found in Figure 8.6 since their regions were already densely populated.

The second family, presented in light blue, has  $\Delta \mathbf{v}$ 's of around 2.6 m/s, which resembles a bit more to the values found for the trajectories which targeted orbits with  $a_{SSTO} = 3$  and 5 km. This family presents worse velocity difference values, while providing a decrease in the overall  $\Delta \mathbf{v}$ , which translates into mission benefits like fuel savings and decreased thruster failure risk.

Table 8.2: Chosen reference trajectory initial conditions and objective function values for  $a_{SSTO} = 5$  km, expressed in [I].

Trajectory	$a_{SSTO} = 5$ km
$\Delta \mathbf{v}_1$	$\left[ \begin{array}{ccc} 0.958 & -0.737 & -0.468 \end{array} \right]^T$ m/s
$t_1$	1086362.6 s
$\Delta \mathbf{v}_2$	$\left[ \begin{array}{ccc} -0.757 & 0.874 & 0.523 \end{array} \right]^T$ m/s
$t_2$	6800.3 s
$f_1$	4.2 m
$f_2$	0.051 m/s
$f_3$	2.566 m/s
$t_{min}$	30147.4 s

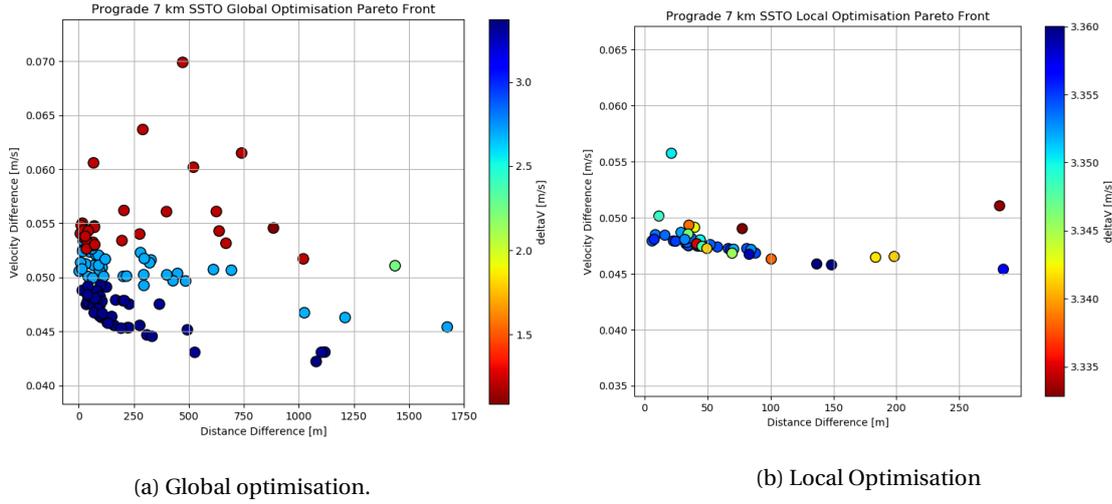


Figure 8.6: Reference trajectory optimisation for ASPECT deployment in a prograde SSTO with  $a_{SSTO} = 7$  km.

The third family, presented in red, has  $\Delta \mathbf{v}$ 's of around 1.3 m/s. It also shows worse velocity difference values and slightly worse position accuracy.

According to the criteria based on the upper margin for the distance and velocity differences presented at the beginning in this section, these two families also present potentially feasible deployment scenarios. This potential will be studied in the following sections. The objective functions and optimisation variables that characterise these trajectories can be found in Table 8.3.

A 3D view of the reference trajectories created in this section can be seen in Figure 8.7. The rationale as to why the reference trajectories have that similar shape can be inferred as follows:

- The second arcs are contained (or close to contained) in the terminator plane of the system. Since one of the best ways to minimise the difference in velocity between the spacecraft and the target deployment point is to make the two vectors parallel, this result was expected. This fact also explains the orientation of the second arc and its shape, which is determined by the minimum speed allowed by the margin to escape velocity at the deployment point.
- The first arc brings the spacecraft from a point of the (given) DCP trajectory to the already characterised second arc. In this sense, this first arc only drives the  $\Delta \mathbf{v}$  objective function, but must also satisfy the constraints used during the optimisation, which were described in Section 7.1.2. Especially relevant to note is that some of these constraints are checked for the propagation of the first arc as if there had not been any second deterministic manoeuvre. The shape of these first arcs is strongly affected by this fact,

Table 8.3: Chosen reference trajectory initial conditions and objective function values for  $a_{SSTO} = 7$  km, expressed in [I].

Trajectory	$a_{SSTO} = 7$ km, 1 <sup>st</sup> family	$a_{SSTO} = 7$ km, 2 <sup>nd</sup> family	$a_{SSTO} = 7$ km, 3 <sup>rd</sup> family
$\Delta \mathbf{v}_1$ [m/s]	$\begin{bmatrix} 0.993 & -0.998 & 0.920 \end{bmatrix}^T$	$\begin{bmatrix} 0.970 & -0.848 & -0.430 \end{bmatrix}^T$	$\begin{bmatrix} 0.002 & -0.498 & -0.188 \end{bmatrix}^T$
$t_1$ [s]	85615.4	1100878.5	1022097.8
$\Delta \mathbf{v}_2$ [m/s]	$\begin{bmatrix} -0.975 & 0.990 & -0.930 \end{bmatrix}^T$	$\begin{bmatrix} -0.792 & 0.966 & 0.509 \end{bmatrix}^T$	$\begin{bmatrix} 0.195 & 0.601 & 0.257 \end{bmatrix}^T$
$t_2$ [s]	7650.7	6119.0	16614.9
$f_1$ [m]	5.9	2.3	18.3
$f_2$ [m/s]	0.048	0.050	0.053
$f_3$ [m/s]	3.354	2.706	1.215
$t_{min}$ [s]	15009.1	51477.2	54319.7

in particular in relation with the constraint that checks the safety distance to Didymain, which does not allow hyperbolic trajectories to have low pericentre heights.

On top of that, no impulses are given from the lower part of the DCP trapezoid because they would violate the distance constraint. Also, the vertical sides of the DCP have a big  $z$  component that is not desired for the deployment, since it happens in a direction almost perpendicular to  $z$ .

Besides, a number of points should be noted about the different families found:

- The consistency of the optimisation algorithm is highlighted by the fact that for the three different orbit heights,  $a_{SSTO} = 3, 5$  and  $7$  km, orbits that share the same shape are found, and they all are non-dominated. This means that the rationale that makes each one of them non-dominated, still applies from one to the other, despite the slightly different conditions. MIDACO is able to reach these solutions despite these differences. The similarity between these trajectories can also be seen in the reported  $\Delta v$  values reported for them in Tables 8.1, 8.2 and 8.3.
- The 1<sup>st</sup> family trajectories for the  $a_{SSTO} = 7$  km case show a distinctly different behaviour. The absence of this family in the rest of cases is explained by the fact that it emanates from the DCP arc that lies in the plane of Didymoon orbit. As such, and due to basic geometry, the spacecraft gets closer to Didymain during the first arc of the PDP, as it can be observed in Figure 8.4. When the target point is at  $7$  km, this distance to Didymain never gets below  $2$  km, but if the target is lower, i.e.  $3$  or  $5$  km, the safety distance to Didymain constraint is violated, what prevents MIDACO from reaching those solutions.
- The 3<sup>rd</sup> family also shows a slightly different behaviour. From what is seen in Figures 8.4 and 8.7, the second arc of this family looks quite similar to the one of the second arc of the 2<sup>nd</sup> family. The only difference is that in the 3<sup>rd</sup> family the spacecraft is injected in a point of that arc that is closer to the deployment. As seen in the projections of Figure 8.7 and in particular in the projection in the XY plane, all the first arcs have the same orientation with respect to the normal to the terminator plane of the system, except the one of the 3<sup>rd</sup> family, which forms the same angle but negative. In conclusion, the normal to the terminator plane is an axis of symmetry of the first arcs of the 2<sup>nd</sup> and 3<sup>rd</sup> families. That orientation of the initial velocities with respect to the impulses seems to be optimal in terms to  $\Delta v$  for any given family. On the other hand, the 3<sup>rd</sup> family geometry brings the spacecraft closer to Didymain, as seen in Figure 8.4, which makes these trajectories infeasible for the  $a_{SSTO} = 3$  and  $5$  km cases.

### 8.3. STOCHASTIC GUIDANCE TUNING

As mentioned in Section 8.2, five different families of trajectories were identified during the trajectory design and optimisation process. The nominal trajectories within these families will be used for fine-tuning of the stochastic firing times.

It must be noted that for each  $t_{firing}$ , the firing allowed times were discretised in ten groups to average the effect of other sources of uncertainty in the results. This is imposed by a constraint in the number of Monte Carlo runs that could be run within a reasonable computational time and memory margins. Also note that removing these sources of uncertainty would have reduced the reach of the study, since this analysis aims at obtaining the firing conditions that create the most robust deployment strategy with respect to the uncertainties themselves. This approach reduces the resolution of the analysis done, but increases the consistency of the choices made for a given number of Monte Carlo runs.

In the following paragraphs, the number of stochastic manoeuvres given will be discussed. Next, the  $t_{firing}$  tuning process for the final number of manoeuvres will be described.

The analysis carried out to determine the optimum number of manoeuvres and their schedule showed the same trends and behaviour with respect to the number and timing of the manoeuvres. For this reason, and to avoid a large amount of not really insightful plots, the qualitative analysis of this section will only be done for the trajectory that targets a SSTO with  $a_{SSTO} = 5$  km. In any case, the optimum firing times for all the trajectories reported in the previous section will be given.

#### 8.3.1. NUMBER OF STOCHASTIC CORRECTIONS

The study started with the parametrisation of the possible  $t_{firing}$  as a function of their position along the trajectory, as it was described in Section 7.1.4. The use of only one manoeuvre was ruled out, since it was assessed to only yield infeasible deployment scenarios. Two distinct cases are found for one manoeuvre:

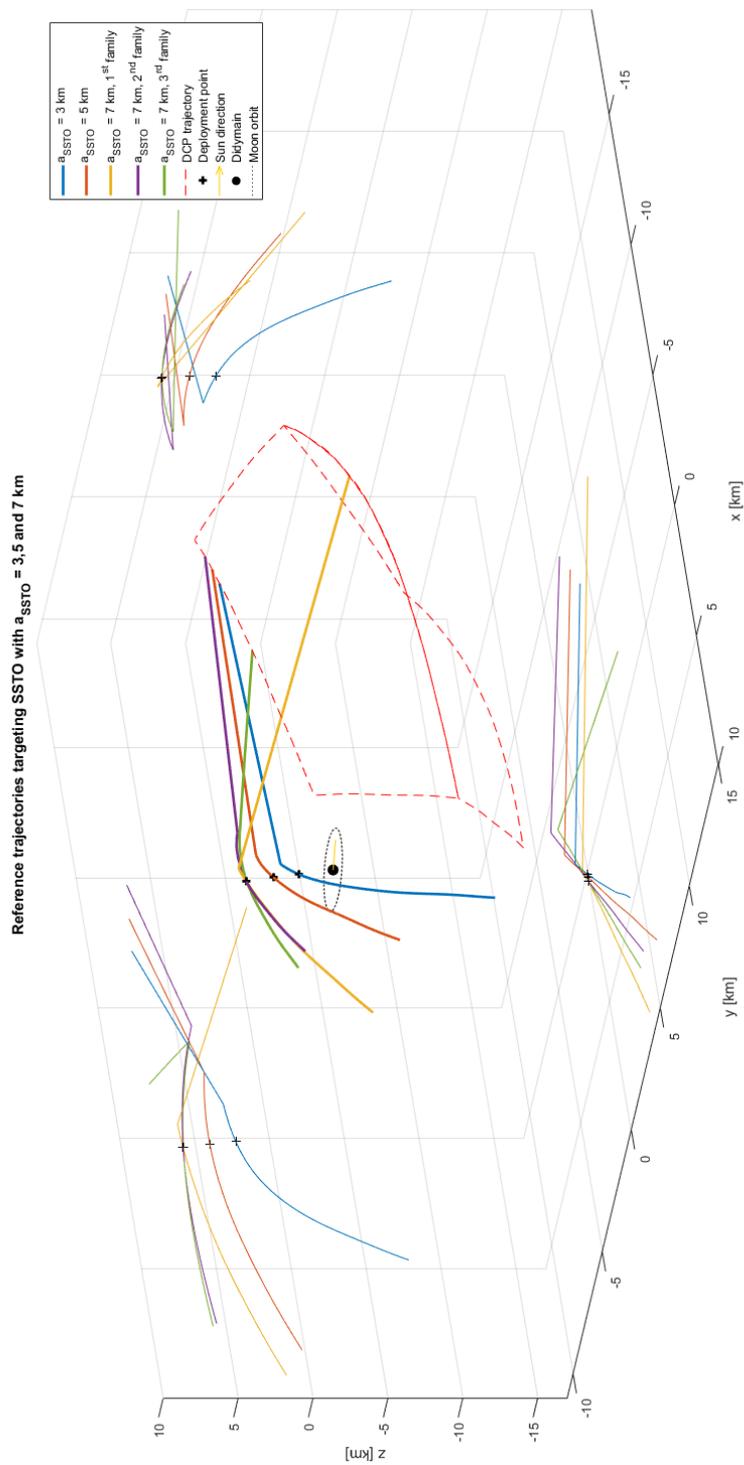


Figure 8.7: Reference trajectory families, expressed in [I].

- $t_{firing} < t_2$ : When the stochastic manoeuvre takes place before the injection into the second arc, the estimation that the guidance scheme is able to make of  $|\delta\tilde{\mathbf{r}}(t_{min})|$  becomes really poor, since the linearised estimation gets worse for distant epochs. This firing schedule yields too big values for end position dispersion, which is always greater than the allowed margins shown in Table 4.1. Besides, the errors induced by the deterministic manoeuvre at  $t_2$  are not corrected at any point, which enlarges the expected end miss-distance. On the other hand, the manoeuvre takes place early in the trajectory and thus the correction has a reasonably low  $\Delta\mathbf{v}$ . This makes the velocity less likely to be out of the safety deployment intervals shown in Table 4.1.
- $t_{firing} > t_2$ : When the stochastic manoeuvre takes place after the injection into the second arc, the dispersion with respect to the reference trajectory has become large. This dispersion has been growing since the first deterministic manoeuvre at  $t_1$  without any correction being given. In this scenario, the commanded  $\Delta\mathbf{v}$  is always big enough to bring the end velocity out of the margins shown in Table 4.1. The end position is better, since the guidance estimation considerably improves.

Note that neither of these two scenarios are desirable, since they do not leave room for successful cubesat deployments. Also note that the relations mentioned here will be further described below, as they are the main drivers for most of the choices that are made in this section.

From the rationale given above it is clear that there must be at least two firings. Also from what has been discussed it emanates that each of them should be scheduled in a different trajectory arc. The firing in the first arc will prevent the dispersion from growing as much as to make the second correction too big for an adequate end velocity. The second firing will ensure that the guidance estimation is good enough so that the correction brings the spacecraft close to the desired target end position.

In Figure 8.8, the simulation values that are used to express the different deployment safety intervals of Table 4.1 are shown as a function of when the second (and last) firing took place. It is especially relevant to compare the values with the ones found in Figure 8.9, where the same results are shown but for a third (and last) firing. Note that these two Monte Carlo runs are absolutely independent, one of them has two stochastic

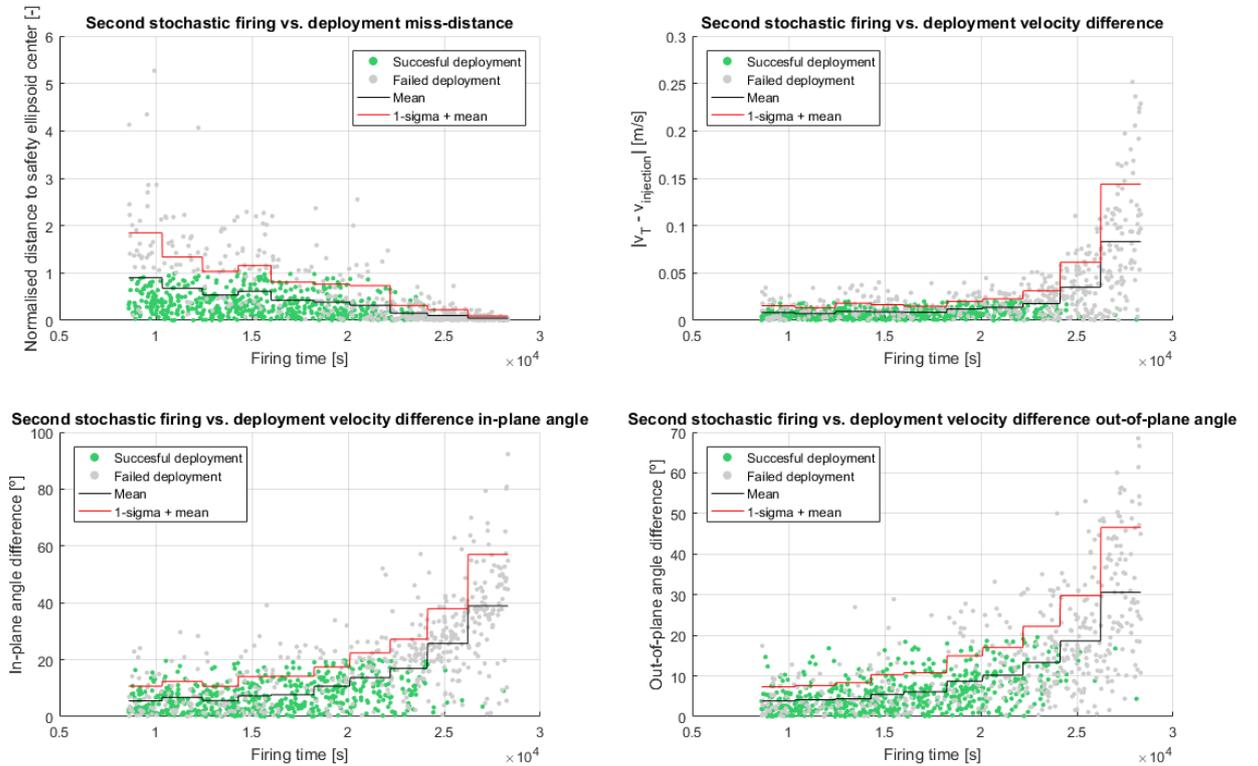


Figure 8.8: Second (out of two) firing condition effect on the deployment safety intervals, for a 5 km SSTO.

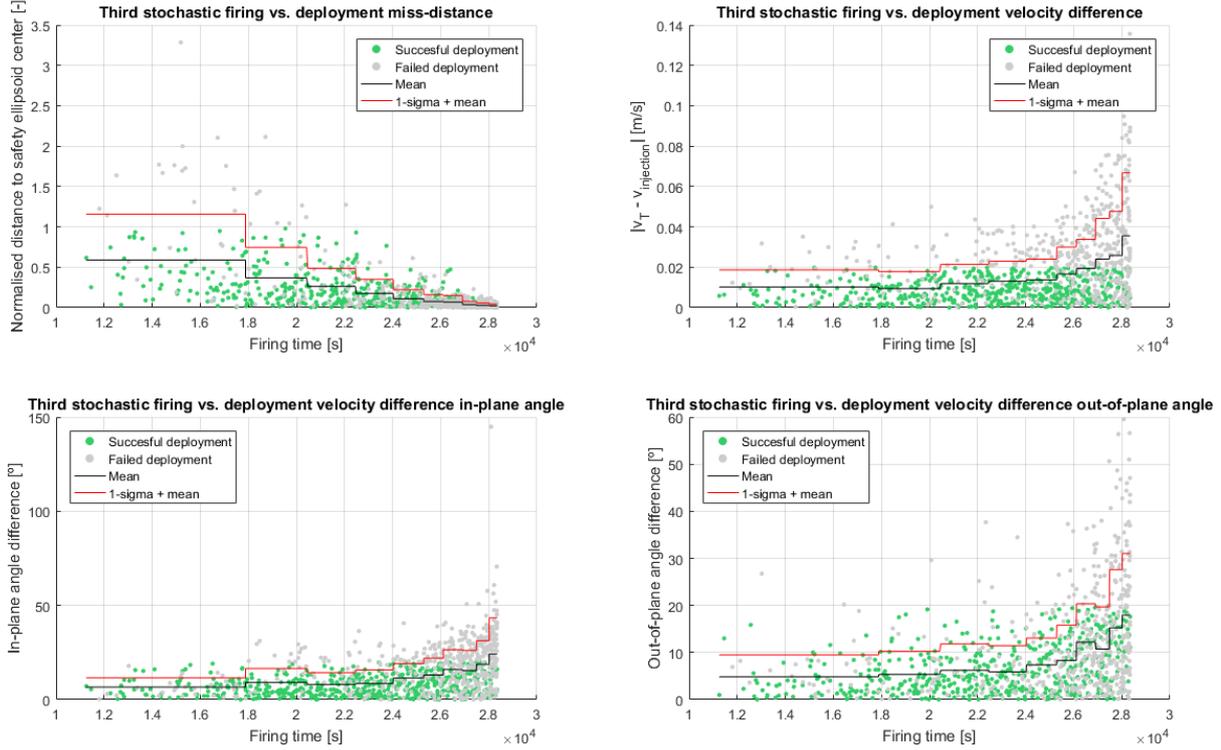


Figure 8.9: Third (out of three) firing condition effect on the deployment safety intervals, for a 5 km SSTO.

manoeuvres scheduled, and the other one, three.

Also note that in these plots the mean and  $1 - \sigma + \text{mean}$  values are used to represent the average of the parameter and the deviation in the sample, which are good measures of average quality of the deployment conditions.

Figures 8.8 and 8.9 show that the deployment conditions for the case in which only one manoeuvre is scheduled in the second arc are in general much worse than when two manoeuvres take place. The position difference, represented as the normalised distance to the center of an ellipsoid with the margins expressed in Table 4.1, show an almost twice as big deviation. The magnitude of the velocity difference, as well as the in-plane and out-of-plane angles between the target and actual velocities are again almost twice as big for late firings. For early corrections, the velocity difference magnitude and in-plane angle are still around 50% bigger whereas the out-of-plane angle shows almost the same numbers in both plots for similar firing times.

In Figures 8.10 and 8.11, the  $\Delta v$  of the manoeuvre is plotted against the timing of the last firing for two and three stochastic manoeuvres, respectively. It can be seen that the  $\Delta v$  gets even four times bigger for late corrections. For early ones the magnitude is around 50% bigger.

From what has been presented in this section, it becomes clear that there is no advantage in scheduling only two firing conditions over three. It must be noted however that the analysis carried out only took into account the success rate of the deployment, assuming that factors like the operability of both configurations or the safety against thruster malfunction are not part of the trade-off process. These factors should be taken into account in further iterations on the PDP.

According to the results found, the firing strategy that will be followed is:

- The first manoeuvre will take place between the first,  $t_1$ , and second,  $t_2$ , deterministic guidance firings. This manoeuvre is required to avoid the first arc to greatly depart from the reference trajectory due to the big initial dispersions,  $\Delta v_1$  associated command errors and the dynamical uncertainties that act between  $t_1$  and  $t_2$ . Without this correction manoeuvre, the injection into the second arc would have higher dispersions than the first one, what would lead to a really demanding scenario for the rest of stochastic corrections.

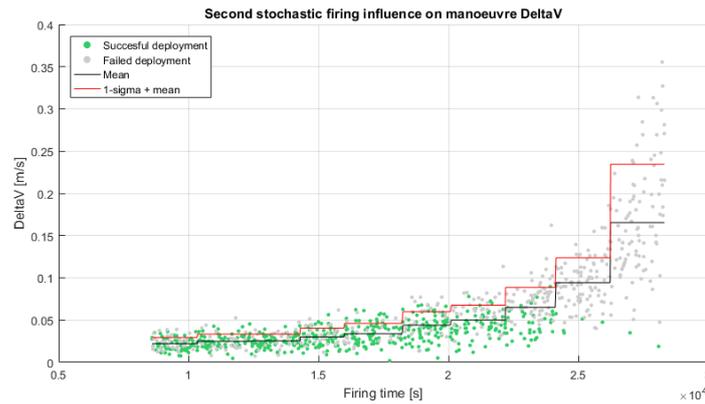


Figure 8.10: Second (out of two) firing condition effect on the manoeuvre  $\Delta v$ , for a 5 km SSTO.

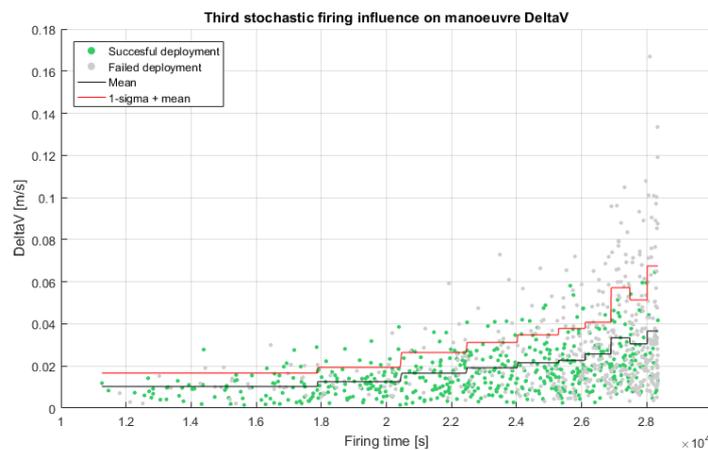


Figure 8.11: Third (out of three) firing condition effect on the manoeuvre  $\Delta v$ , for a 5 km SSTO.

- The second stochastic manoeuvre will take place after the second deterministic manoeuvre. Its goal is to command a small  $\Delta v$  ( $t_{go}$  is still big) impulse such that the trajectory starts converging to the reference one instead of diverging due to the command errors associated to  $\Delta v_2$  and dynamical uncertainties.
- The third stochastic manoeuvre will take place close to the deployment epoch in such a way that a small  $\Delta v$  (the position dispersion is small) that makes use of the improved guidance estimate, brings the spacecraft to the target deployment point within the acceptable margins expressed in Table 4.1.

The timing of these manoeuvres will be discussed in the next section.

### 8.3.2. STOCHASTIC FIRING TIMES

In Figures 8.12, 8.14 and 8.16, the success rate of the cubesat deployment as a function of the firing times can be seen. It must be noted that the overall success rate of the Monte Carlo campaign for this case was of 46.9 and 52.0% for the Safety Interval 1 and 2 from Table 4.1, respectively. The combined intervals yield a success rate of 53.5%.

From Figure 8.12, it can be seen that there exists a low correlation between the success rate of the injection and the time at which the first firing takes place. It must be noted however, that this does not mean that the first stochastic manoeuvre is not required. As it was argued in the previous section, a stochastic manoeuvre during the first arc notably increases the mission success rate.

A slightly higher success rate is observed for late stochastic corrections, but no clear trends are observed. A detailed analysis of the failure modes found as a function of when the  $t_{firing_1}$  takes place is presented in Figure 8.13. It must be noted that percentages of both figures are not meant to sum up to 100%, since some failures are caused by violation in both the position and the velocity conditions shown in Table 4.1.

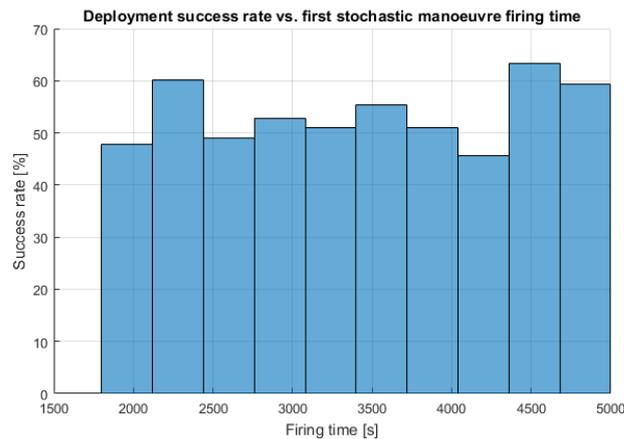


Figure 8.12: First firing condition effect on the succes rate, for a 5 km SSTO.

Although no clear trends can be inferred from Figure 8.12, some facts can be highlighted when combined with Figure 8.13:

- It can be seen that the injection position failure events are less frequent for late firing conditions. This is explained by the fact that the linearisation predicts the corrections better when its closer to the target time,  $t_{min}$ . This effect will become much more obvious for  $t_{firing_2}$  and  $t_{firing_3}$ . It must be noted however, that all the effects that are observed for the  $t_{firing_1}$  are greatly "dampened" by the later errors introduced by the deterministic and stochastic manoeuvres given and the ones caused by the dynamical uncertainties in the system.
- The velocity failure is the predominant mode of cubesat injection failure. This is explained by the fact that the guidance scheme used in this thesis only corrects for the distance dispersions and does not take into account the target end-of-mission velocity, as it has already been explained previously in the report. However, no trend can be observed as a function of  $t_{firing_1}$ , in contrast with what will be discussed in the following paragraphs for  $t_{firing_2}$  and  $t_{firing_3}$ . This difference can be explained by the same reason that was given for the position failure mode "dampened" relation with  $t_{firing_1}$ .

From what has been shown in Figure 8.12, it can be concluded that the first firing condition has a low impact on the final success rate of the cubesat injection, although it is not negligible. As seen in the figures,

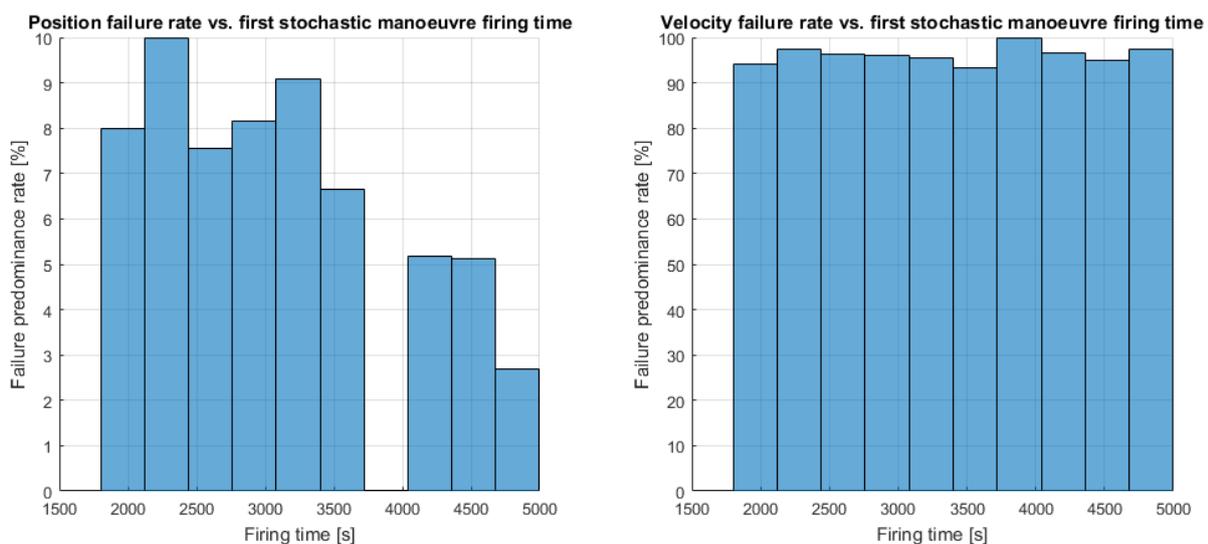


Figure 8.13: First firing condition effect on the failure mode, for a 5 km SSTO.

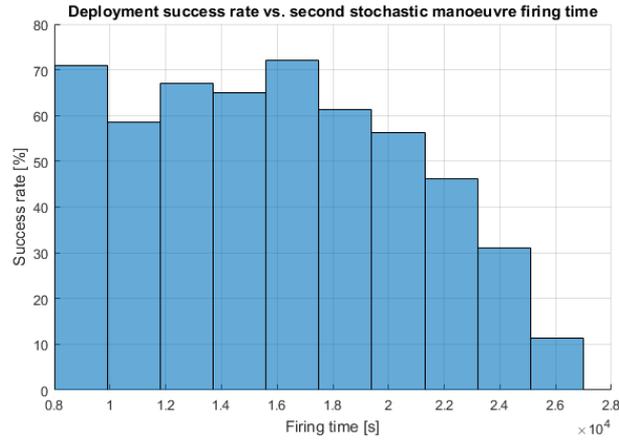


Figure 8.14: Second firing condition effect on the succes rate, for a 5 km SSTO.

a late correction manoeuvre is preferred in terms of the success, but not for the  $\Delta v$ . After what has been discussed, the choice for the first firing condition is  $t_{firing_1} = 4500$  s.

In Figure 8.14 the injection success rate as a function of  $t_{firing_2}$  can be seen. In this case, there is relevant correlation between both variables. The later the manoeuvre is given, the worse the chances of a successful deployment get. This can be explained by the increasing  $\Delta v$  magnitude that is commanded by the guidance. This  $\Delta v$  magnitude increase can be easily inferred looking at Equation (5.14): the longer you wait to carry out the manoeuvre, the less time ( $t_{go}$ ) there is to correct a practically "constant" estimated end-of-mission distance,  $|\delta\mathbf{r}(t_{min})|$ . Please note that  $|\delta\mathbf{r}(t_{min})|$  is "constant" because it is the product of a linearised estimation, and thus it changes with time as the linearisation becomes worse with the distance to the reference trajectory,  $|\mathbf{x}_{MainSC}(t) - \mathbf{x}_{ref}(t)|$ .

In Figure 8.14 it can be seen that the best scenario is encountered for times around  $t_{firing_2} = 16500$  s. For times after this one, the rate gets worse and reaches a minimum of around 10% for the last possible firing epoch. In this worst-case scenario,  $t_{firing_2}$  takes place so late that the dispersion has become really big. The required  $\Delta v$  magnitude to correct for such a dispersion is big and has a direction too perpendicular to the reference trajectory, what leads to a resulting velocity out of the injection safety margins.

The effect described in the previous paragraph is shown in Figure 8.15, where the velocity failures reach a 100 % for late manoeuvres. As already explained, the position-induced failures are minimised for late manoeuvres, since the estimation done by the guidance system becomes better.

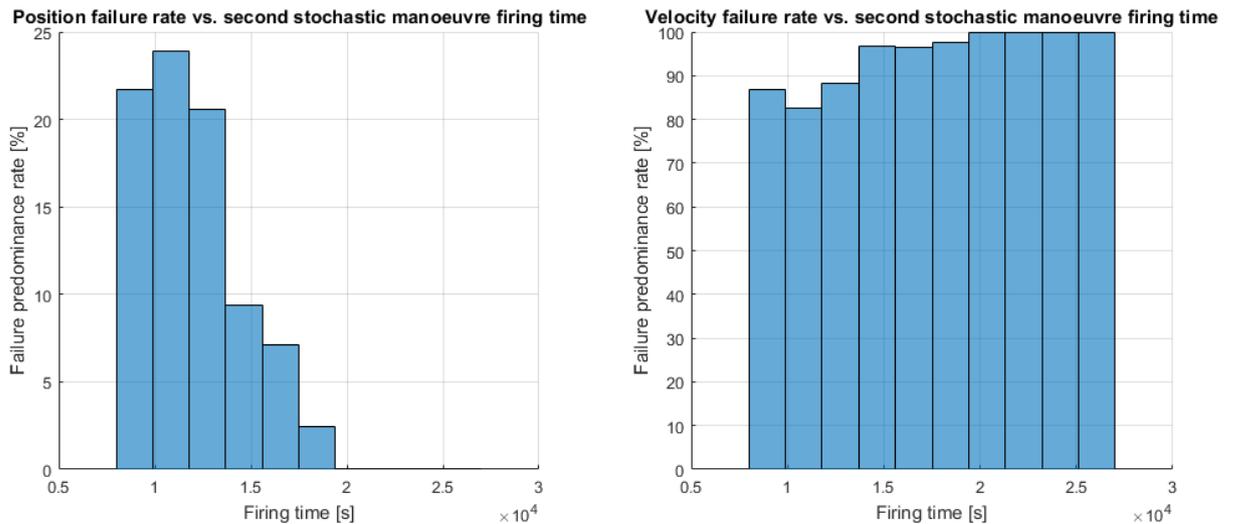


Figure 8.15: Second firing condition effect on the failure mode, for a 5 km SSTO.

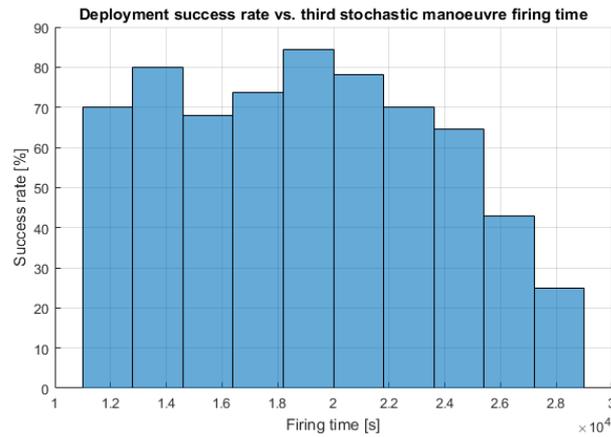


Figure 8.16: Third firing condition effect on the succes rate, for a 5 km SSTO.

In Figure 8.16, the deployment success rate as a function of the third firing time,  $t_{firing_3}$ , is shown. The behaviour is extremely similar to the one shown in Figure 8.14, what is explained by three main reasons:

- Both manoeuvres take place after the last deterministic impulsive manoeuvre. This fact makes both of the manoeuvres ( $t_{firing_2}$ ,  $t_{firing_3}$ ) act "directly" on the deployment conditions.
- The time between  $t_{firing_2}$  or  $t_{firing_3}$  and the deployment one,  $t_{min}$  is much smaller than for  $t_{firing_1}$ , which leaves the dynamical uncertainties and the command-induced errors less time to produce high dispersions.
- As explained in Section 7.1.4, the parametrisation of the firing times for the Monte Carlo analysis leaves  $t_{firing_2}$  and  $t_{firing_3}$  strongly correlated. This happens because the time at which  $t_{firing_3}$  is allowed to happen is delimited by the first choice of  $t_{firing_2}$  and the no-action margins imposed around them.

From the results shown in Figure 8.16, the best firing time is  $t_{firing_3} = 19000$  s.

The same behaviour that was explained for the previous firing times can be inferred from Figure 8.17. However a few differences can be observed. For early manoeuvres, the failures are predominantly due to the position. This is explained by the fact that the third manoeuvre is the last one before the deployment and, if it takes place too early, the miss-distance estimation from the guidance system is really poor. As soon as the

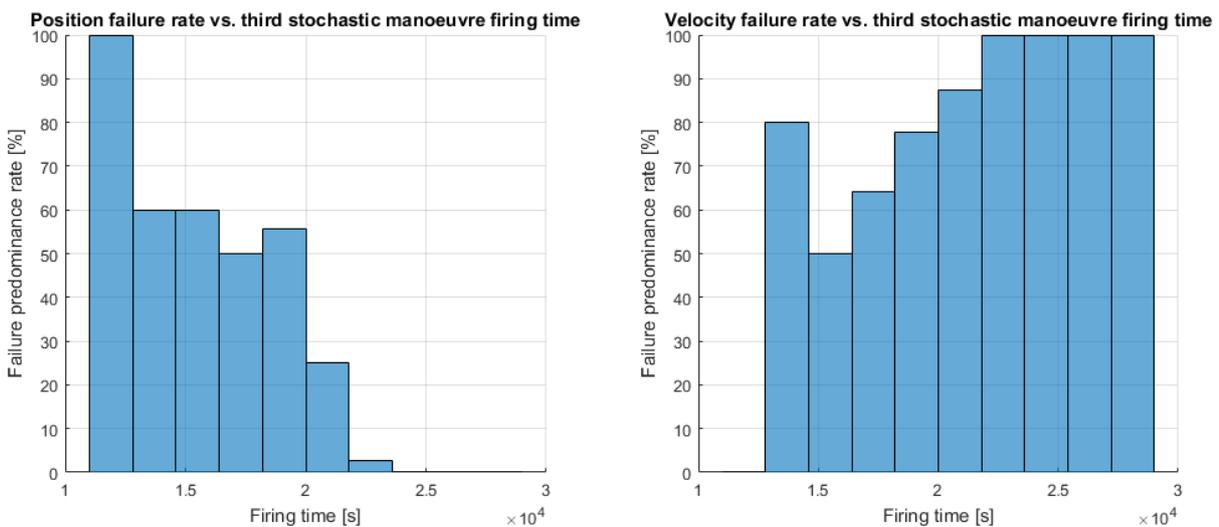


Figure 8.17: Third firing condition effect on the failure mode, for a 5 km SSTO.

firing takes place slightly closer to the deployment a drastic reduction in the position failure mode is seen. For late manoeuvres' behaviour, the same rationale as for the other firings apply.

As already mentioned at the beginning of the section, the analysis carried out for the firing times is cyclic for every reference trajectory. This means that the behaviour observed for the deployment success rate and the rationale for this behaviour repeats in each of the trajectories. Since the plots, trends and rationales behind the choices do not offer any notable difference, they are omitted from the report. A summary of the results found for the other trajectories described in Section 8.2, can be found in Table 8.4.

Table 8.4: Firing times choice for the reference trajectories.

Trajectory	$t_{firing_1}$ [s]	$t_{firing_2}$ [s]	$t_{firing_3}$ [s]
3 km	2000	8700	11300
5 km	4500	16500	19000
7 km, 1 <sup>st</sup> family	2300	9700	11500
7 km, 2 <sup>nd</sup> family	3300	25000	40000
7 km, 3 <sup>rd</sup> family	3000	30000	43000

#### 8.4. MISSION SAFETY ASSESSMENT

In this section, the safety of the families of reference trajectories previously generated will be assessed with respect to thruster failures. To model this misbehaviour, the magnitude of the commanded impulse will be scaled down with a factor  $k$ , which will be generated using a uniform distribution between 0 and 1. When the failure is scheduled to happen in the first deterministic manoeuvre, there will not be more firings in the simulation, which will end at  $t_f = 72$  h. When the failure is scheduled for the second deterministic manoeuvre, the first deterministic and stochastic manoeuvres happen as if there was no failure scheduled, i.e. as described in Sections 8.2 and 8.3.

As already explained in Section 2.1, one of the TAP objectives was to guarantee the safety of the following PDP in terms of distance to the system and margin to the escape velocity. The violation of these constraints was covered in the generation of the reference trajectories, but the dynamical uncertainties, command errors and initial state dispersions can change the situation encountered in the real operational scenario. All these factors are included in the analysis presented in this section, where the need of a TAP will be assessed from a mission safety perspective.

It must be noted however, that the effects of the dynamical uncertainties, command errors and initial state dispersions are also a main driver of the mission success ratio, which will be thoroughly analysed in Section 8.5.

As a general remark, in all the figures that will be presented in this section, the mean behaviour of the reported Monte Carlo results is assumed to be caused by the thrusting failure scaling factor. The "noise" around this mean is assumed to be caused by the dynamical uncertainties, command errors and initial state dispersions.

In Figure 8.18, the minimum values for the safety constraints are shown as a function of the first deterministic impulse factor,  $k$ . From what can be seen in Figure 8.18a, the distance to Didymain only gets slightly lower than 2000 m for few epochs. Besides, this constraint violation happens for relatively high impulsive scale factors in an almost-uniform way, which seems to indicate that it is closely related to the other factors already mentioned at the beginning of the section, i.e. dynamical uncertainties, command errors and initial state dispersions. Figure 8.18b shows that the margin to the escape velocity gets close to 0.02 m/s but never goes below it. These two facts prove that the first deterministic manoeuvre will always inject the spacecraft into a safe first arc, despite thrust failures. From this safe arc, the on-board computer would have time to trigger a safe mode without any risk of losing the spacecraft.

In Figure 8.19, the minimum values for the safety constraints are shown as a function of the second deterministic impulse factor,  $k$ . Figure 8.19a shows that the distance to Didymain gets as close as 1400 m for small thruster failures (from 5 to 10 % error in magnitude). This can be explained by the fact that the scaled manoeuvre  $\Delta v$  can lead to smaller and differently oriented final  $v_f$  vectors, whose trajectories can have different shapes, as depicted in Figure 8.20. The fact that it only happens for relatively high scale factors is case specific, and is related to the particular direction and magnitude of  $v_f$ 's that create trajectories that bring the

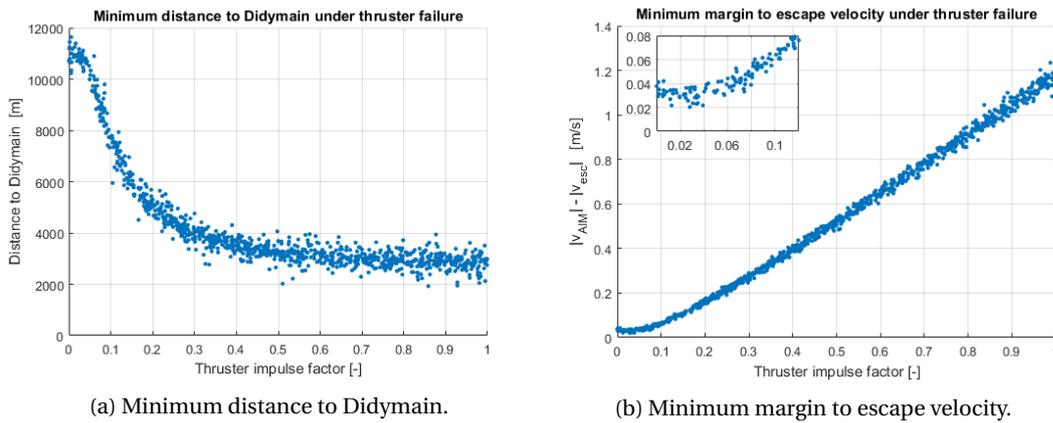


Figure 8.18: Minimum values of the safety constraints as a function of the first impulse scaling factor for the  $a_{SSTO} = 3$  km.

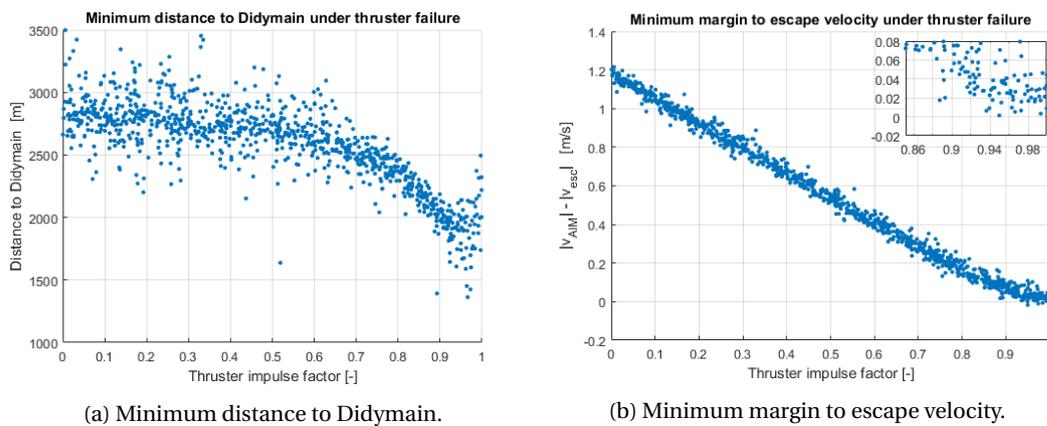


Figure 8.19: Minimum values of the safety constraints as a function of the second impulse scaling factor for the  $a_{SSTO} = 3$  km.

spacecraft closer to Didymain.

Figure 8.19b shows that the margin to the escape velocity gets below the 0.02 m/s threshold for numerous cases, but again this happens for high impulsiveness factor values, which indicates that this violation is more related to the "noise" that the other uncertainties, errors and dispersions create than to the thrust failure.

In any case, the results show that the safety of the reference trajectory targeting a  $a_{SSTO} = 3$  km, is not guaranteed, at least under the margins and conditions established in Chapter 2. This calls for a detailed study of the situations in which these violations occur, to see if they suppose an actual threat for the mission, or to revise the safety margins, since they might be too tight for a mission of these characteristics. The fact that the margin to the escape velocity is so small for nominal thrusting conditions is pointed out as the biggest source of concern about the safety of the mission, which might indicate the need of a TAP to mitigate the dispersions and uncertainties affecting the trajectory.

In Figure 8.21, a similar behaviour is seen for the case  $a_{SSTO} = 5$  km. This time, the distance never gets

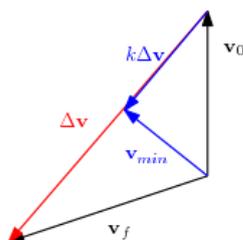


Figure 8.20: Scaled  $\Delta v$  effect on the final velocity  $v_f$ .

below 3 km and the margin to the escape velocity gets below the 0.02 m/s threshold for some cases with low impulse scaling factor. This violation is not caused by the thrusting failure but by deviations caused by the different sources of errors and dispersions in the trajectory initial condition.

Figure 8.22 shows that the spacecraft never gets closer than 3 km to Didymain, while the margin to the escape velocity has the same behaviour as for the  $a_{SSTO} = 3$  km case. It must be noted that the impulse failure has a much lower effect on the minimum distance reached during the second arc, as shown in Figure 8.22a. This is because the spacecraft is further from the asteroid and regardless of the  $\Delta v$  given, the trajectory generated never has a low enough speed as to fall towards the system. The same conclusions as for the  $a_{SSTO} = 3$  km case can be drawn.

In Figures 8.23 and 8.24 a similar behaviour to the past cases is observed. A slightly different behaviour can be seen in Figure 8.24a, which shows that the minimum distance to Didymos of the second arc for the first family of the case with  $a_{SSTO} = 7$  km does not suffer any variation with the thruster impulsive failure. This is partially due to the increased height of the injection point, as it was reported for the case with  $a_{SSTO} = 5$  km, but is mainly caused by the different geometry between the  $\Delta v$  and  $v_f$ . This difference in geometry can be appreciated in Figure 8.7. Again, similar conclusions as for the  $a_{SSTO} = 3$  km and 5 km cases can be drawn.

The results for the second and third families of the case with  $a_{SSTO} = 7$  km are omitted from the report, since they showed exactly the same behaviour as for the first family for both types of manoeuvre failures and both safety constraints.

The results obtained highlight the need for further studies addressing the safety of these trajectories and/or the presence of a TAP trajectory that mitigates the initial condition dispersions. What is more, the dynamical uncertainties, dispersions and command errors seem to be the cause of the most frequently found safety violation. Thruster malfunction is only found cause a safety constraint violation in one case, found in Figure 8.19a, and always in combination with the other factors already mentioned.

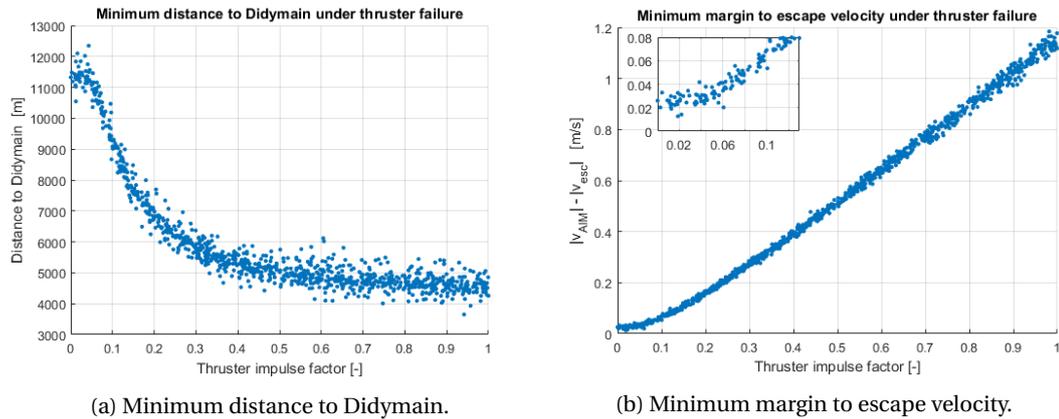


Figure 8.21: Minimum values of the safety constraints as a function of the first impulse scaling factor for the  $a_{SSTO} = 5$  km.

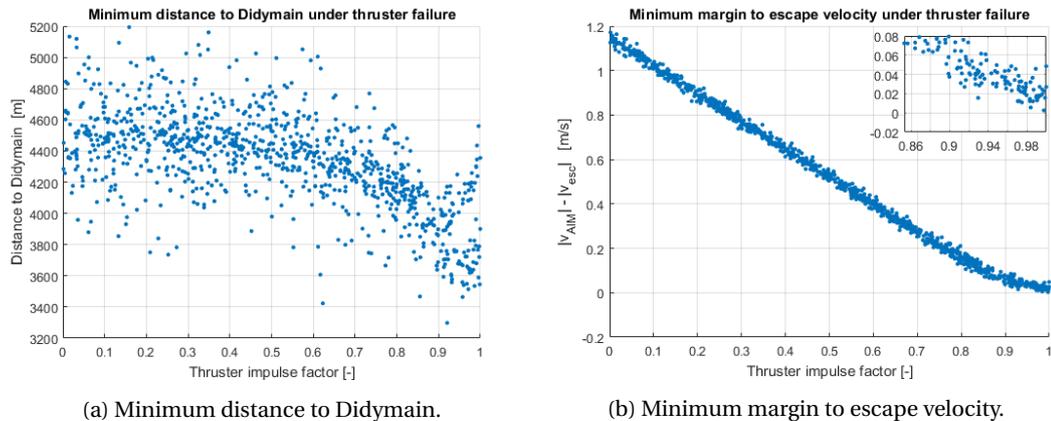


Figure 8.22: Minimum values of the safety constraints as a function of the second impulse scaling factor for the  $a_{SSTO} = 5$  km.

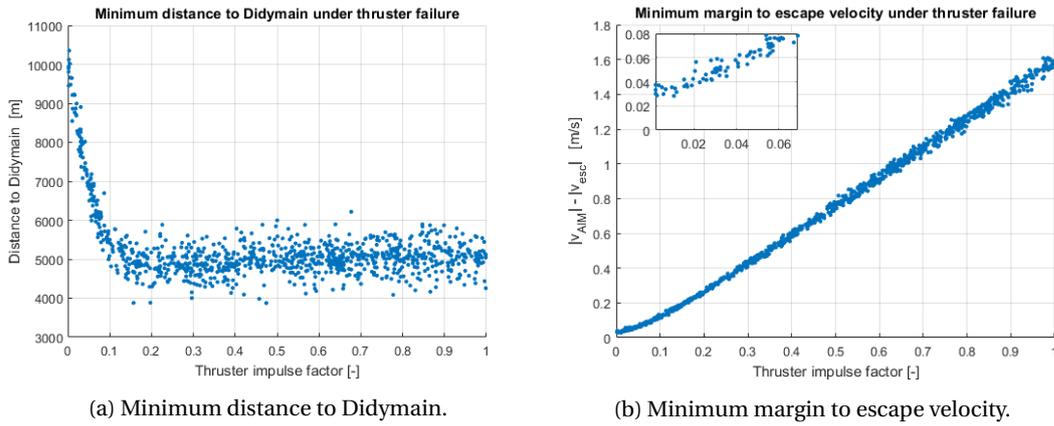


Figure 8.23: Minimum values of the safety constraints as a function of the first impulse scaling factor for the  $a_{SSTO} = 7$  km, first family.

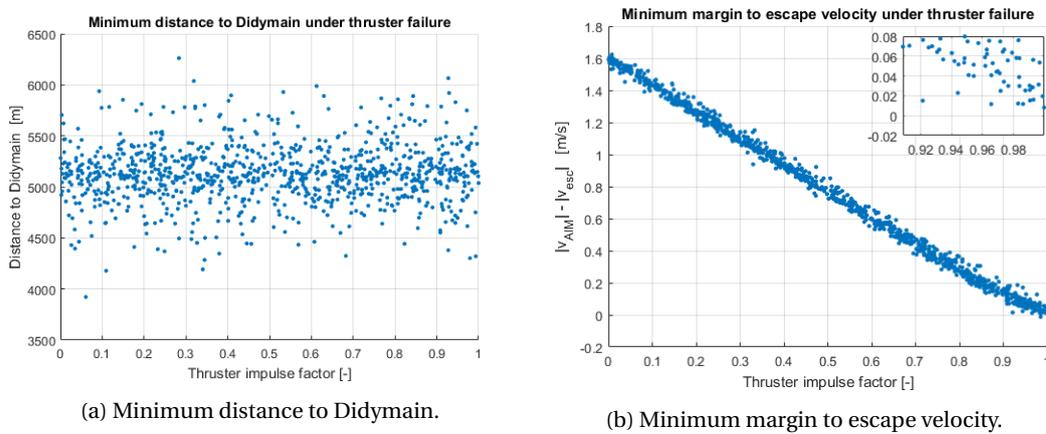


Figure 8.24: Minimum values of the safety constraints as a function of the second impulse scaling factor for the  $a_{SSTO} = 7$  km, first family.

### 8.5. DEPLOYMENT SUCCESS ASSESSMENT

In this section, the deployment success rate for the different reference trajectories presented in Section 8.2, under the uncertainties characterised in Section 7.1.4 and with the guidance firing times set according to what was discussed in Section 8.3, will be presented. These results can be directly translated to the PDP success expectations.

The summary of the results coming from the Monte Carlo campaigns can be seen in Table 8.5, where both safety deployment intervals presented in Table 4.1 will be assessed. It must be noted that the combined success rate is calculated as the union of the success rates for the two presented intervals, i.e., if the deployment is successful under one of the criteria, it is considered successful.

Table 8.5: ASPECT cubesat injection success rates for the different reference trajectories and safety deployment intervals 1 and 2, all values in %.

Trajectory	Combined success	Success 1/2	Position success 1/2	Velocity success 1/2
3 km	61.75	61.25 / 61.75	100 / 100	61.25 / 61.75
3 km, backup	65.91	63.20 / 65.48	87.13 / 82.09	69.34 / 71.27
5 km	75.65	69.9 / 71.95	91.95 / 86.95	74.05 / 81.35
7 km, 1 <sup>st</sup> family	50.15	47.5 / 25.45	100 / 100	47.5 / 25.45
7 km, 2 <sup>nd</sup> family	44.4	38.75 / 25	66.8 / 67.1	50.2 / 31.7
7 km, 3 <sup>rd</sup> family	40.75	36.4 / 22.65	60.55 / 60.95	50.75 / 30.5

From what is seen in Table 8.5 it can be concluded that the deployment of the ASPECT cubesat is not feasible with an acceptable confidence margin. The best case is obtained for the trajectory targeting a SSTO with  $a_{SSTO} = 5$  km, but the success rate is still far from what could be acceptable for a realistic operational scenario. This maximum is reached for this trajectory because it presents the best balance between low objective function values and wide safety deployment intervals. Trajectories with lower objective function values (i.e., 7 km ones), also have much narrower safety deployment intervals, and vice versa.

Note that the success for each of the intervals in Table 8.5 is assumed to happen when both the position and the velocity would allow a successful deployment at the same time and for the same interval. This means that only when the final state vector of the cubesat is within position and velocity margins for the interval 1, it is considered that the deployment of the cubesat is successful under interval 1 criteria. This fact also explains why the success rate for each of the intervals is not the lower value between the position and velocity success rate, since not every velocity success case is also a position success one.

It must be noted that if the success rates reported in Table 8.5 are compared to the success rates that were obtained during the  $t_{firing}$  tuning in Section 8.3 a notable difference is seen. First of all, the overall success rate of the  $a_{SSTO} = 5$  km trajectory increases from the roughly 50% reported at the beginning of Section 8.3.2, to a 75%. As already explained, this is not enough for a real operational scenario but it can be said that there is a noticeable improvement on the results and it is obtained exclusively by tuning the stochastic firing times. However, the 75% success rate reported hereby is not as high as the success rate shown in Figure 8.16 for the chosen  $t_{firing_3}$  value, which gets close to 85%. This difference is caused by a couple of facts:

- The study carried out in Section 8.3 treats each impulsive manoeuvre as independent in the sense that its effect on the trajectory is analysed only as a function of its own firing time. The reality is that stochastic firing conditions, especially the second and third one, are strongly coupled, and their effect should have been studied as such. For the case at hand, it is possible that the close to 85 % success rate shown in Figure 8.16 could only be that promising for  $t_{firing_2}$  values that are simply not the specific ones that were chosen for it in that particular campaign.
- The resolution used in the study of Section 8.3 was set according to an estimation of the number of points and firing time intervals that yielded the more realistic and noise-free plots. Although the improvements expected from a "mesh" refinement are small, it is definitely a source of error that could be reduced. It must also be noted that an increment in the resolution would increase the computational requirements of each Monte Carlo campaign, since the amount of points per analysis time interval must be kept in the same order of magnitude. This was a limiting factor during this MSc thesis due to time and computational constraints.

If this operational scenario was to be analysed further in future design iterations, a reassessment of the stochastic firing times taking these two facts into account is recommended.

However, the success rate could also be improved by identifying the main causes of deployment failure and trying to dampen their effects. For this reason, the following pages will present the injection conditions as a function of the dynamical uncertainties, command errors and dispersions that the spacecraft encounters. To do so the trajectory targeting the SSTO with  $a_{SSTO} = 5$  km is chosen, which is also identified as the most successful one and thus, the most promising for future iterations.

### 8.5.1. UNCERTAINTIES

In this section, the injections conditions will be studied as a function of the dynamical uncertainties. All the nominal values for the parameters have been given in Section 7.1.4. In Figure 8.25, the effect on the deployment of the deviations from the nominal Didymoon conditions are shown.

Note that the normalised deviation, used to group the effects coming from Didymoon, was calculated as:

$$\Delta \bar{x}_{Moon,m} = \frac{|a_{Moon} - a_{Moon,m}|}{a_{Moon}} + e_{Moon,m} + \frac{|\mu_{Moon} - \mu_{Moon,m}|}{\mu_{Moon}} + \theta_{Moon,m}(t_0)/10\pi \quad (8.1)$$

where the subscript  $m$  indicates a Monte Carlo variable, whose calculation was explained in Section 7.1.4.

From what is seen in Figure 8.25, the condition of Didymoon has little effect on the final success of the mission. However, a noticeable effect can be seen on both velocity difference angles, with changes of up to 20% for the in-plane angle. From this plot it can be concluded that by fixing the Didymoon conditions with which the reference trajectories are generated, the deployment conditions could be slightly improved. The prime meridian initial condition for both bodies were kept out of the analysis due to their negligible effects.

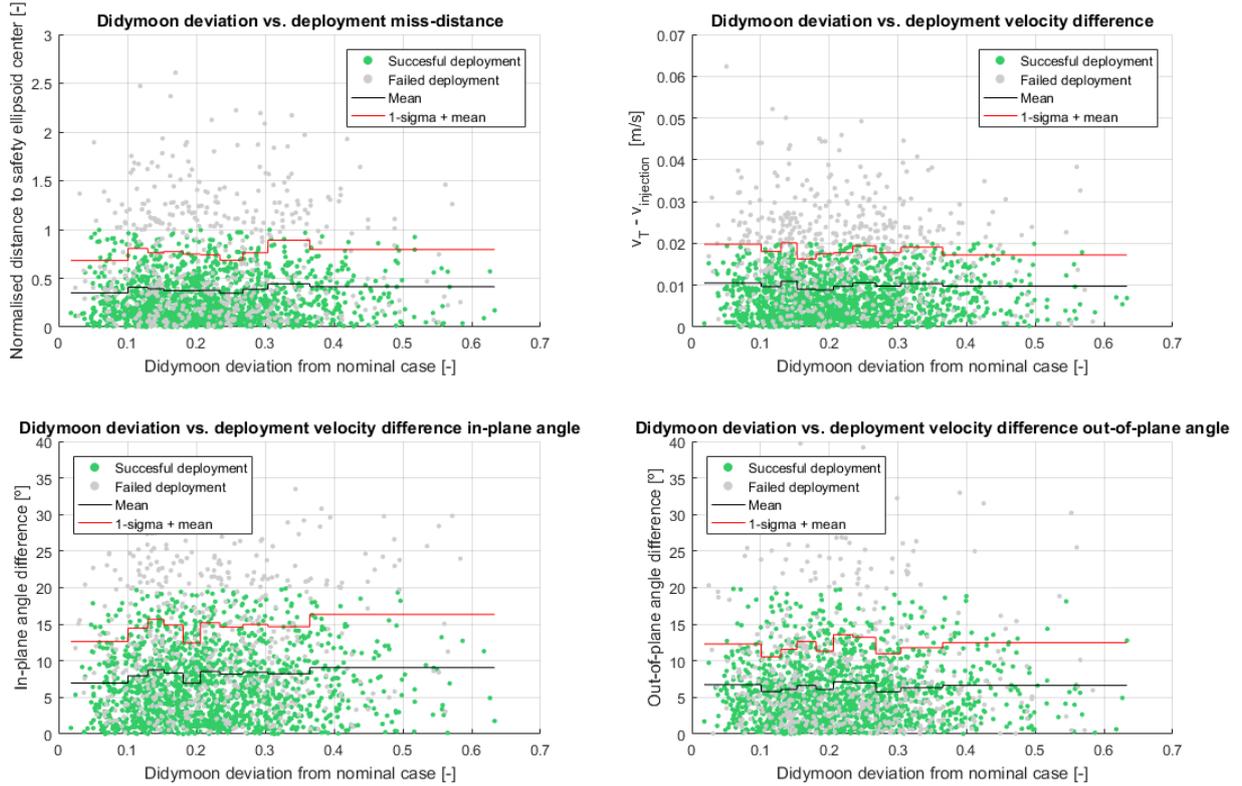


Figure 8.25: Didymos normalised deviation from the nominal status effect on deployment conditions.

It must be noted that the effect that the dynamical uncertainties have on the end-of-mission conditions should be understood as the effect that these uncertainties have on the stochastic corrections given, being these corrections what end up driving the deployment conditions. In this case, the deviations from the nominal moon status induce mainly perturbation forces that slightly move the center of gravity of the system, which deflects the trajectories. The guidance corrections are small in magnitude (small impact on the velocity magnitude difference), but have rather perpendicular directions to the velocity, which induces the changes in direction observed for the deployment velocities.

In Figure 8.26 the effect of the  $SRP$  coefficient  $C_R/B$  can be seen. Along the first arc, the  $SRP$  impulses the spacecraft in the longitudinal direction, which increases the end velocity difference magnitude and changes the in-plane angle slightly. During the second arc, the  $SRP$  acts perpendicularly to the spacecraft motion, but as seen on the figure, this induces negligible changes in the out-of-plane angle.

In Figure 8.27 the effect of the  $\mu_{Didymos}$  on the deployment conditions can be seen. For increasing values of the gravitational parameter the spacecraft velocity slightly starts pointing towards the barycentre of the system, which induces velocity differences with bigger in-plane angles. This also induces a slightly less accurate end position. Interesting to note is that lower values of the gravitational parameter seem to not be as "bad" for the deployment conditions as higher ones, which seems to indicate that higher forces tend to create systems more difficult to control.

In Figure 8.28 the effect on the deployment of the knowledge on Didymos gravitational parameter, expressed as a deviation with respect to the actual value  $\mu_{Didymos,m}$ , is seen. It is concluded that the effect of this knowledge on the final deployment conditions is small if not negligible, only showing a slight worsening of the conditions for high knowledge errors. This is probably due to the fact that the guidance linearised estimations already introduce bigger errors than the ones caused by this knowledge.

In Figure 8.29 the effect the navigation errors on the deployment conditions can be seen. Note that here cumulative means that the magnitudes of the errors are added for every stochastic firing epoch. Surprisingly, only a small worsening of all the final conditions can be appreciated with increasing position knowledge error values. This can be explained by the robustness of the guidance system to errors with sizes as shown. It

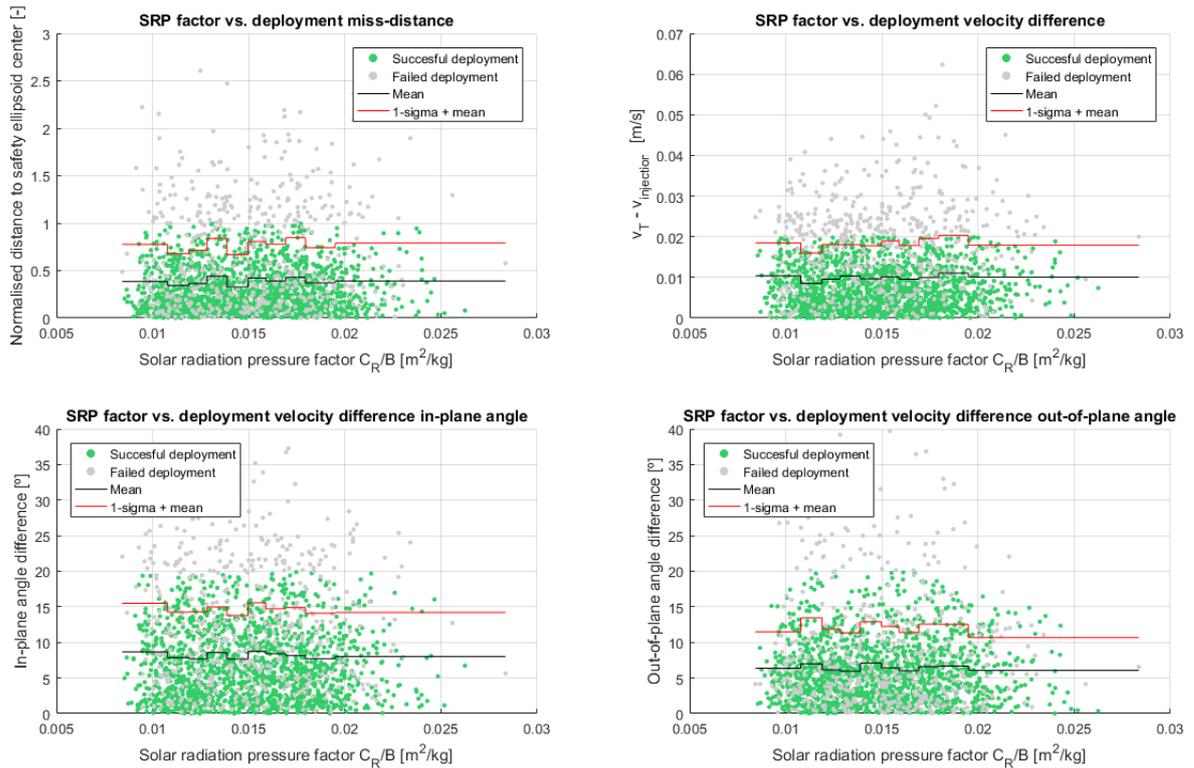


Figure 8.26: SRP coefficient  $C_R/B$  effect on deployment conditions.

must be noted however, an extremely important relation of this robustness with the actual dispersion with respect to the reference trajectory, as depicted in Figure 8.30. The system shows robustness to considerably big knowledge errors (around 100 m) because the actual deviations from the reference trajectory are so big, that in comparison to the estimation errors introduced by the guidance system, the navigation error contributions can be considered almost negligible. If the dispersions from the reference trajectory were the same size as the navigation errors, then the guidance system corrections would become unstable.

In Figure 8.30 it is made clear the strong relation between the position dispersions from the reference trajectory and the velocity-related deployment conditions. As already explained, the guidance system robustly corrects for position errors, but at the cost of commanding increasingly high  $\Delta v$ 's that compromise the successful end-of-mission velocity conditions.

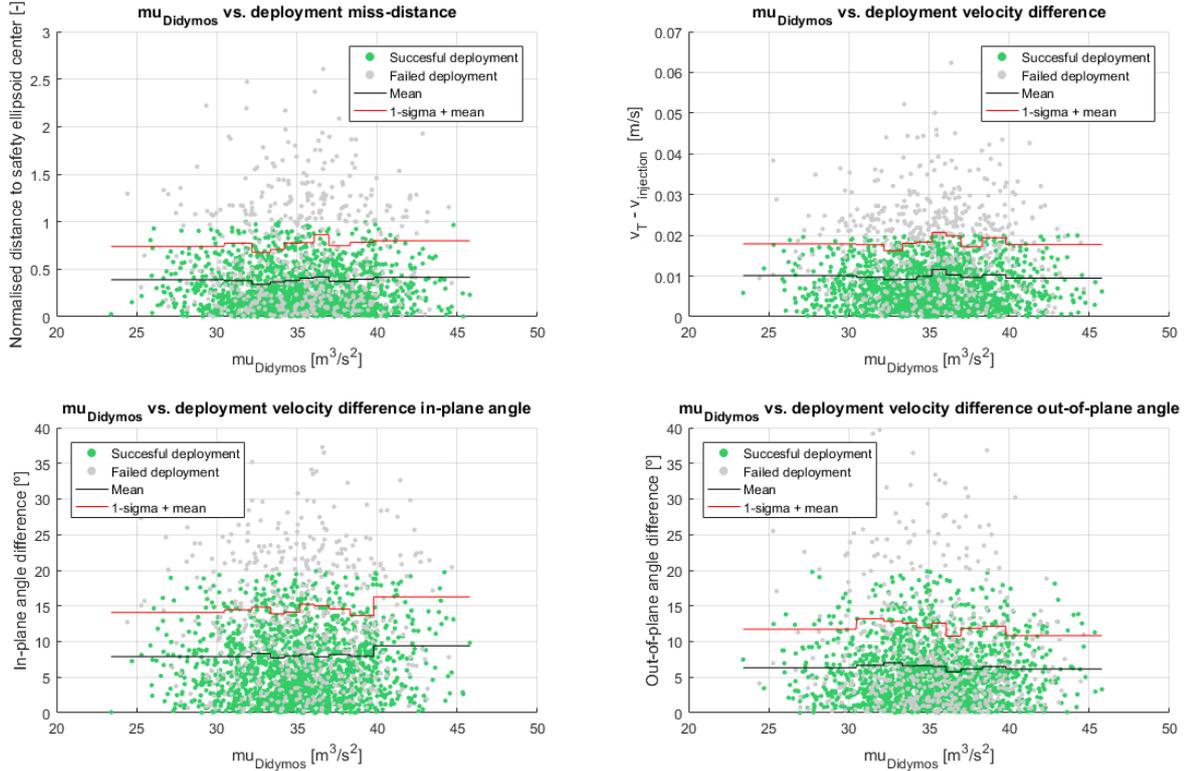
The relation of the system with the velocity knowledge errors is quite different, as depicted in Figures 8.31 and 8.32. The velocity knowledge strongly affects the end-of-mission position accuracy. This is explained by the fact that the guidance scheme used only tries to correct dispersions in position, which makes it robust to any value of those. However, the velocity knowledge is used to calculate the required  $\Delta v$ , as seen in Equation (5.14), which introduces a source of uncertainty that is never corrected for and ultimately drives the position accuracy. The velocity-related deployment conditions also get worse by this effect.

These results highlight the importance of the navigation errors in the cubesat deployment success. From a position accuracy point of view, it would be desirable to improve the knowledge available on the velocity of the main spacecraft, since so far it has been proved to be the main driver for this value.

It must be noted that navigation error and guidance dynamical parameter knowledge effects are treated under dynamical uncertainties although they effectively translate into "command errors".

### 8.5.2. COMMAND ERRORS

In this section, the errors associated with the attitude knowledge and the thruster behaviour are analysed. These two sources of errors were characterised in Chapter 2.

Figure 8.27:  $\mu_{Didymos}$  effect on deployment conditions.

In Figure 8.33 the effect of these two errors on the final deployment conditions is studied. They were calculated as the sum of the errors of every commanded manoeuvre throughout the whole trajectory, including both stochastic and deterministic ones. These errors have a strong influence on the end-of-mission velocity difference angles. In particular, it can be seen that the out-of-plane angle is affected the most, with an increase of around a 50% for the biggest command error magnitudes. This difference between the effect on the in-plane and out-of-plane angles is explained by the fact that the overall amount of factors influencing on the latter one is more limited and furthermore, the magnitude of these forces acting on the out-of-orbit plane direction is smaller in general. This makes a direct effect on the spacecraft state vector as the commanded errors to have a considerably big influence.

The modelling given to the source of these errors in this thesis has been oversimplified. This model is thought to be conservative in the sense that it overestimates the errors associated to these sources. For future studies it would be recommendable to apply actual thruster and attitude navigation models. This would most likely increase the representativeness and the success rate of the mission at hand.

The contribution of the CRS was assessed to be negligible and kept out of the analysis. This also points out that AIM attitude accuracy and knowledge, which are expected to be lower than the CRS injection errors would play a negligible role in the deployment.

### 8.5.3. INITIAL DISPERSIONS

In this section the effect of the initial dispersions associated with the initial condition taken from the DCP trajectory file is studied. This dispersion was characterised in Section 5.2.

The initial position dispersion presented in Figure 8.34 has a strong effect on the end-of-mission velocity conditions. As already explained many times throughout the report, the position is corrected by the guidance system, which manages to consistently reduce any initial dispersion encountered by the spacecraft. However, the end-of-mission velocity conditions are extremely influenced by this dispersion since for increasing values of it, the commanded  $\Delta v$ s get bigger and deviate the velocity enough to make the deployment infeasible.

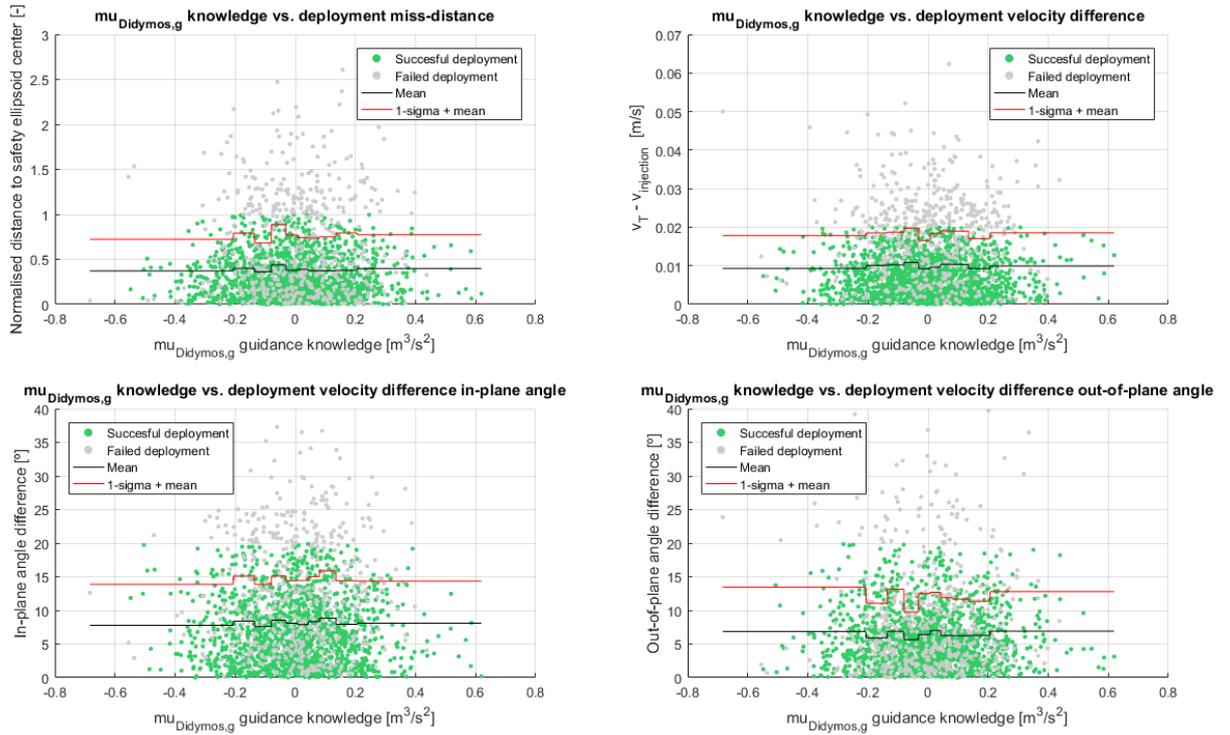
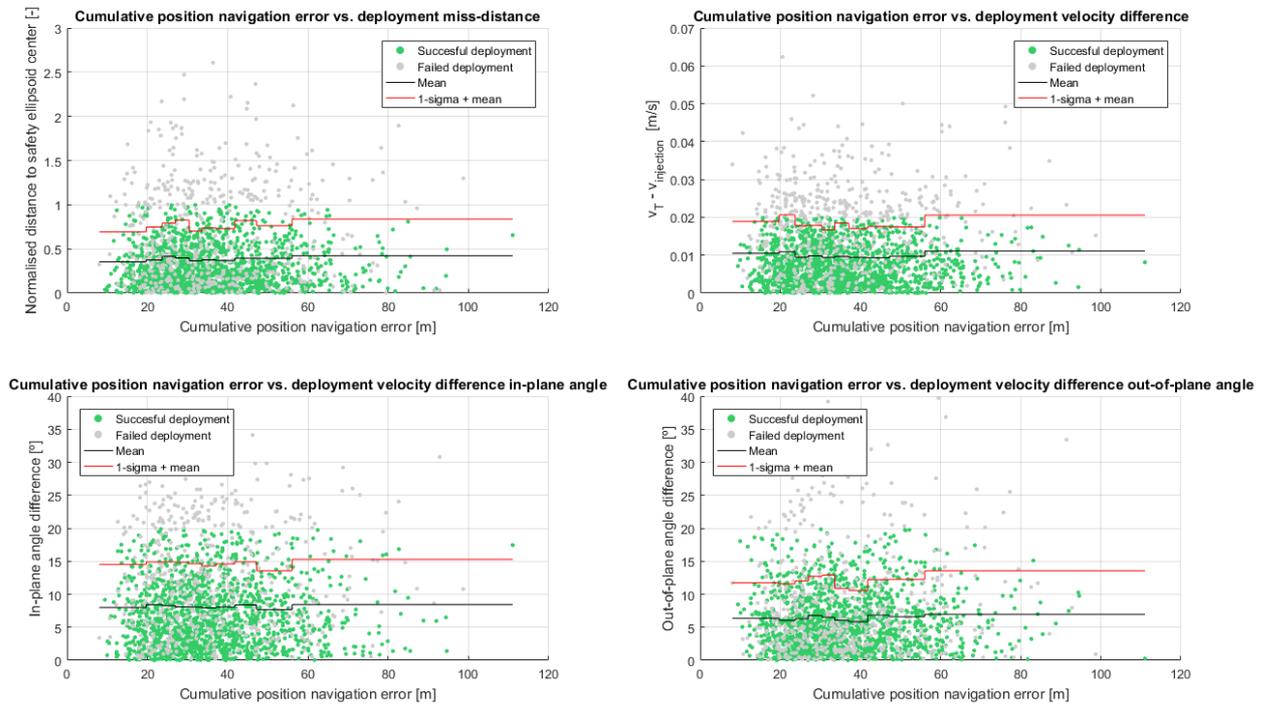
Figure 8.28: Guidance knowledge on  $\mu_{Didymos}$  effect on deployment conditions.

Figure 8.29: Cumulative position navigation error at stochastic firing epochs effect on deployment conditions.

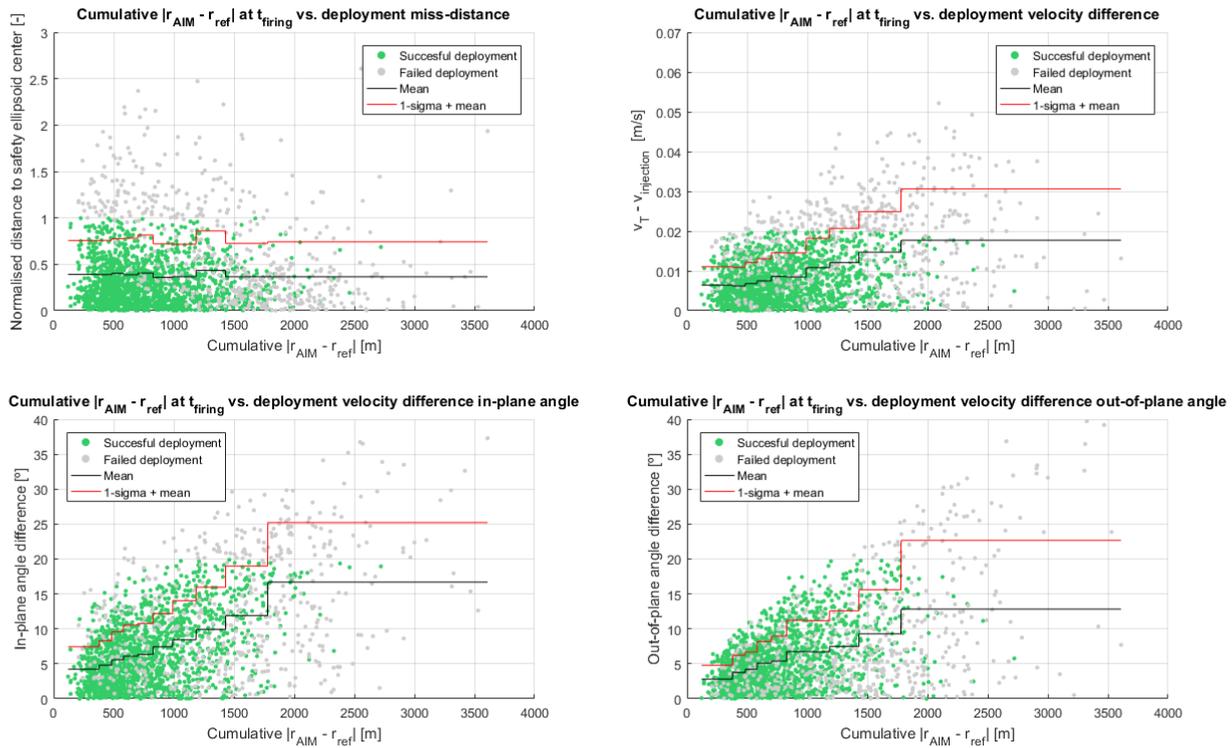


Figure 8.30: Cumulative position deviation from the reference trajectory at stochastic firing epochs effect on deployment conditions.

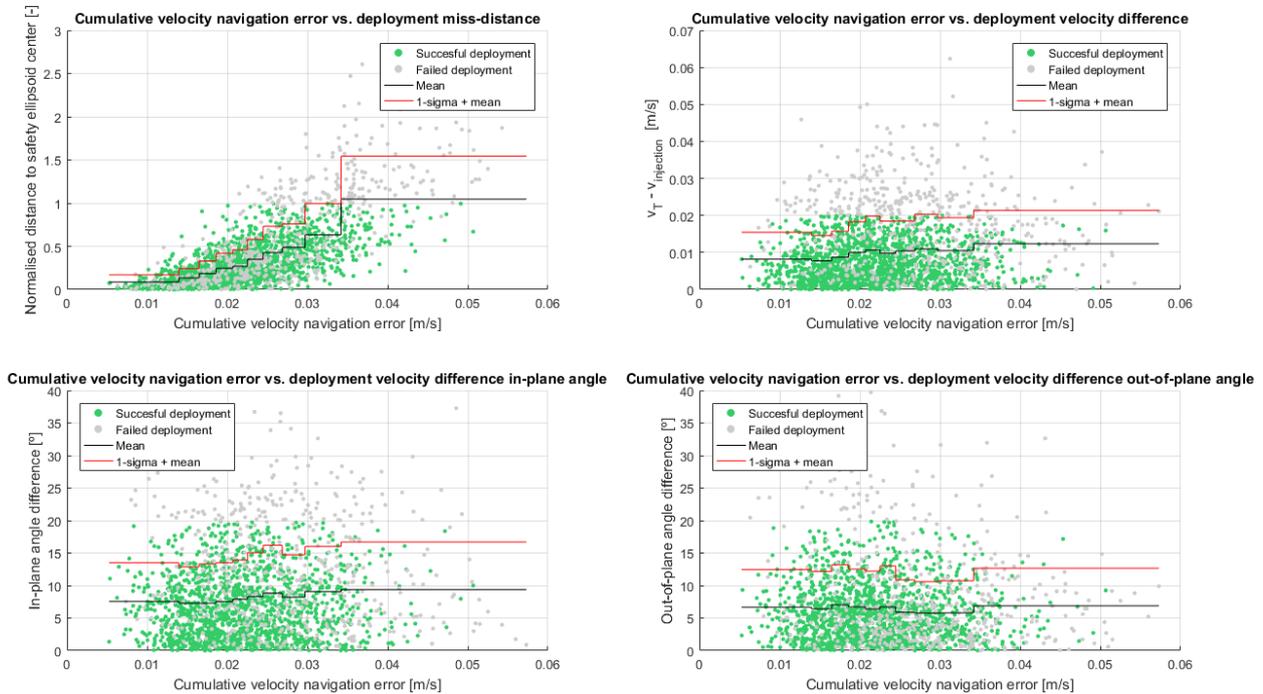


Figure 8.31: Cumulative velocity navigation error at stochastic firing epochs effect on deployment conditions.

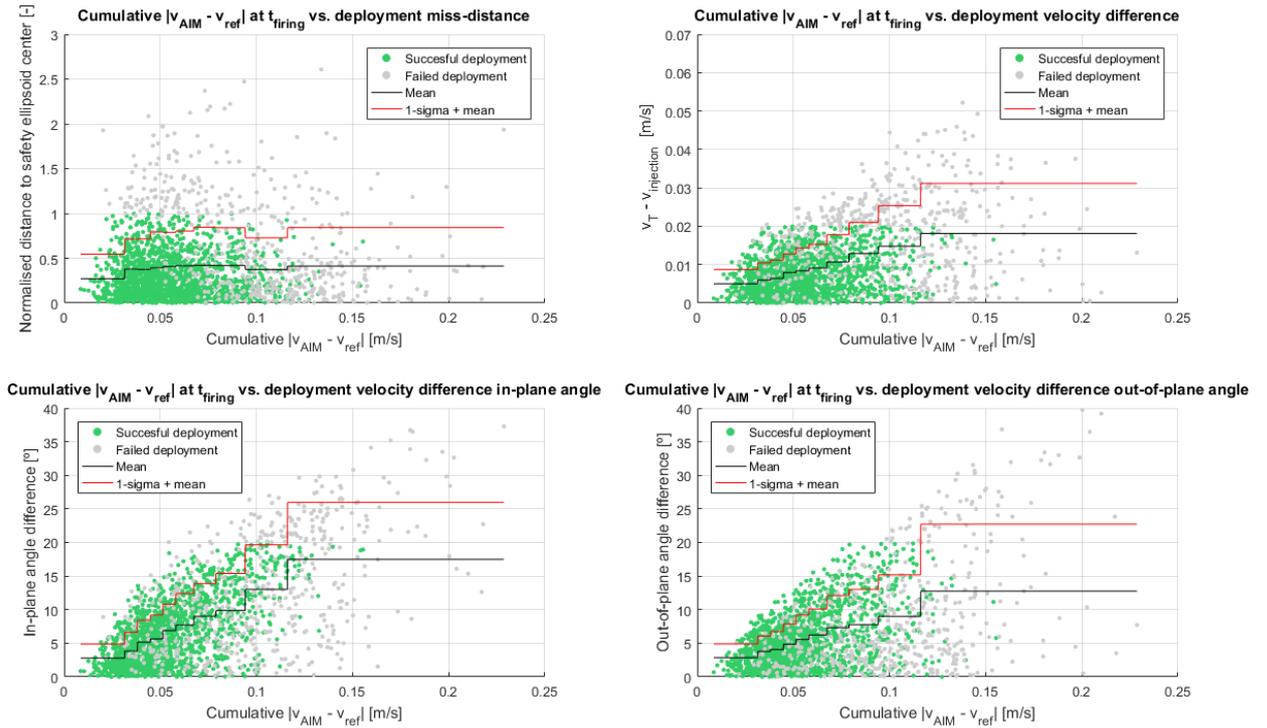


Figure 8.32: Cumulative velocity deviation from the reference trajectory at stochastic firing epochs effect on deployment conditions.

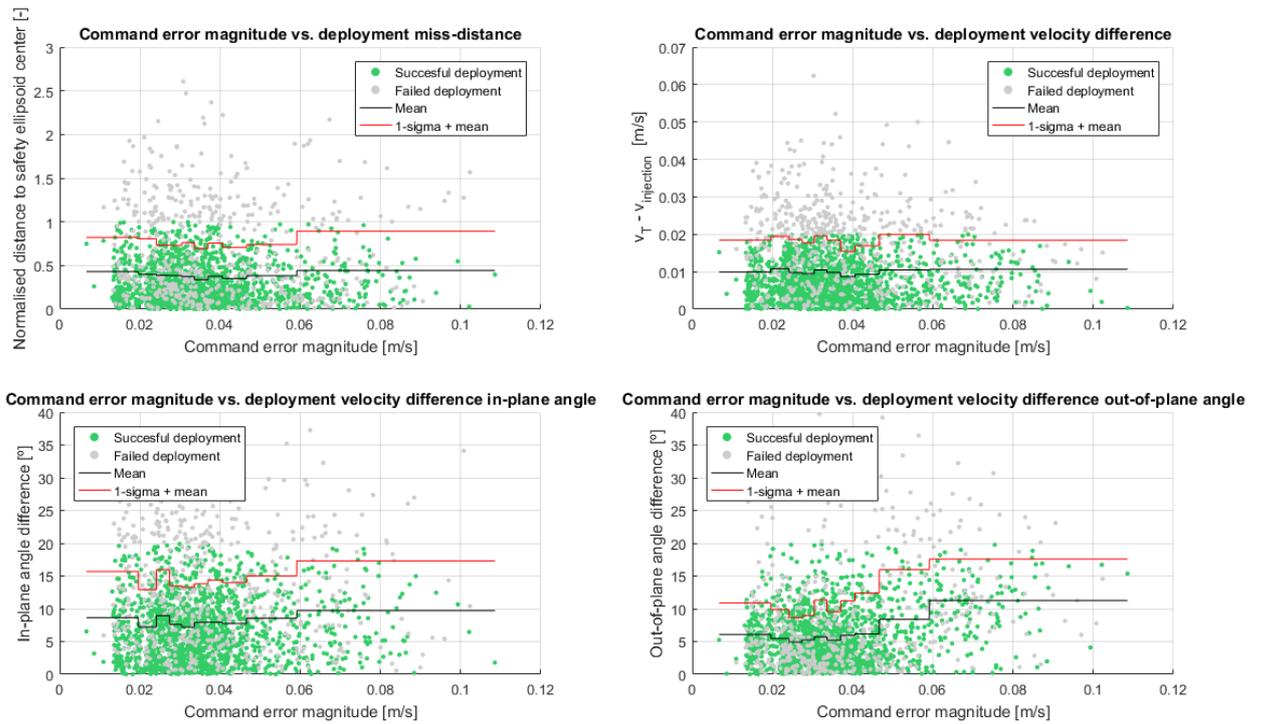


Figure 8.33: Cumulative commanded velocity error effect on deployment conditions.

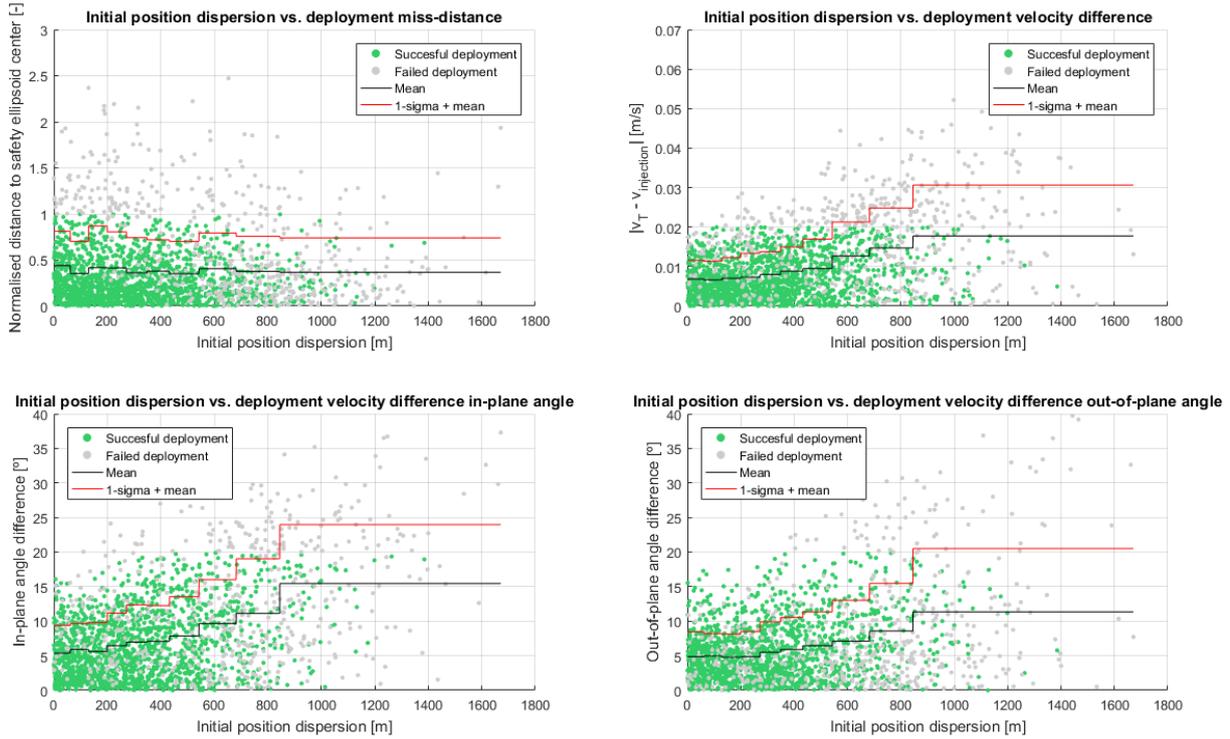


Figure 8.34: Initial position dispersion effect on deployment conditions.

Another fact that also contributes to the observed results is that for increasing deviations of the spacecraft from the reference trajectory, the effect of the dynamical uncertainties gets amplified, since the forcefield deviates further from the nominal one. This is one example of the coupling between the different factors that has been pointed out many times throughout the report.

It is interesting to realise that the effects shown in Figure 8.30 are simply a particular case of what is shown in Figure 8.34, but applied to the firing epochs. It can be seen that the correlation found in Figure 8.30 is much higher. This is due to the fact that the deployment velocity differences are strongly related to the magnitude of the stochastic corrections, which are directly proportional to the gap between the reference trajectory and the actual spacecraft position at the firing epochs. It must be noted however that the initial dispersion shown in Figure 8.34, is a "design" parameter that can be directly tuned by slightly changing the concept of operations of the second cycle of the DCP manoeuvre, or instead by scheduling a TAP.

Initial velocity dispersions shown in Figure 8.35 do not show correlation with the deployment conditions because these initial values are quite low with respect to the navigation and command errors introduced along the trajectory, as it is seen from Figures 8.31 and 8.33. Even though it is not because of its initial value, this velocity dispersion builds up over time and has a big influence on the final results, as it was shown in Figure 8.32.

The mission scenario described in this thesis is a complex one, with plenty of interdependencies that complicate even further the interpretation of the results. To draw clear and precise conclusions about the relations between the different factors that play a role in the deployment conditions, a chain of facts drawn from this section's results is given below.

1. The initial position dispersion is a big driver of the magnitude of the impulsive commands given along the trajectory. The bigger the initial dispersion is, the bigger these commands will be. Since the dispersions from the DCP are used, they are considerably big.
2. The guidance scheme used is robust as far as end-of-mission position accuracy goes. This robustness comes at the price of a somewhat poor velocity accuracy.

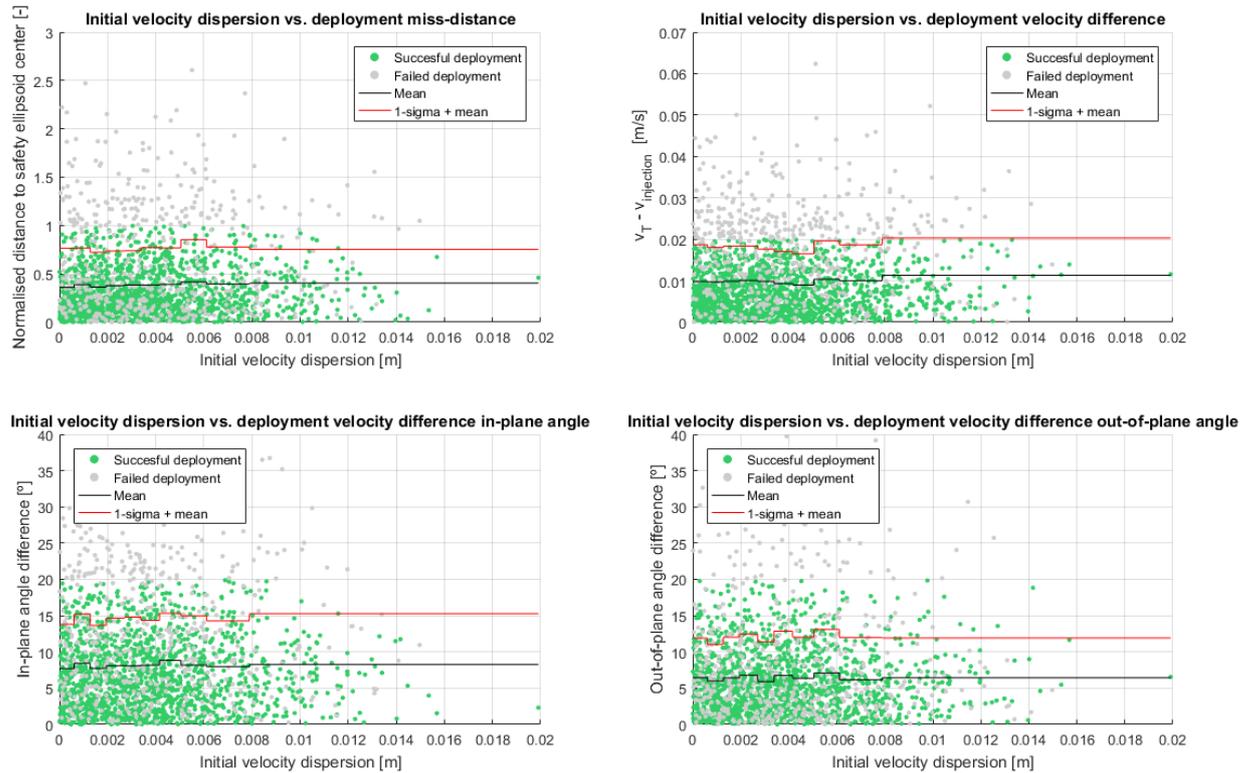


Figure 8.35: Initial velocity dispersion effect on deployment conditions.

3. The big impulses given, the command errors and the guidance estimation inaccuracies due to the velocity knowledge, cause big dispersions in the velocity, which build up over time.
4. These velocity dispersions, which are not corrected by the guidance system, cause residual position dispersions. Besides, the different dynamical uncertainties of the system also enlarge the position dispersions as the spacecraft advances along the trajectory.
5. These errors are corrected with stochastic impulses, and the loop goes back to step number 4.
6. Steps 4 and 5 are repeated for each stochastic manoeuvre.

From the chain of facts explained above, it becomes clear that the initial dispersions, taken from the **DCP** ground navigation assessment found in Section 5.2, are the main driver of the deployment success rates presented in Table 8.5. They do not only directly contribute to the deployment conditions, but also enable and increase the effect of most of the other contributions.

In this sense it must be noted that one of the main goals of the **TAP**, which was part of the past baseline for **AIM**, was to reduce the dispersions at the beginning of the **PDP**, since this latter phase involved dangerous autonomous operations as close as 500 m from the asteroid system.

Reducing the initial dispersion is thought to be the easiest way of increasing the deployment success rate. This could be done by the use of a **TAP** or simply by ensuring that the dispersions present in the trajectory of the second **DCP** cycle from which **AIM** is injected into the **PDP** are low enough.

It must be noted that the analysis hereby presented only tries to grasp which factors drive the deployment conditions, and thus its success. In this study, only those factors who have a strong enough influence on these conditions as to not be "covered" by the noise produced by the variation of the other parameters are pointed out. This is a clear limitation of the method used for the study, since it does not directly handle the strong and numerous couplings that are present in the system. However, and as the results presented show, the method is consistent enough as to highlight the main drivers of the deployment success and allows to identify the recommended future line of work.

# 9

## CONCLUSIONS AND RECOMMENDATIONS

The objective of the thesis was to answer the research question and its sub-questions. They read as follows:

*How can we design a transfer trajectory between two points in the close-proximity operations range of a binary asteroid while satisfying payload injection accuracy and mission safety requirements?*

with the following sub-questions:

- *What is the nominal trajectory that best fits the [AIM](#) requirements for the deployment of [ASPECT](#)?*
- *What are the guidance firing conditions that lead to the best deployment conditions?*
- *How do uncertainties in the strongly perturbed environment affect the deployment conditions?*
- *How do guidance command errors affect the deployment conditions?*
- *How do uncertainties and dispersions in the spacecraft state vector affect the deployment conditions?*

In this chapter, the conclusions drawn from the thesis work in relation to the research goals will be presented. Then, recommendations for the future work will be given.

### 9.1. CONCLUSIONS

The conclusions drawn from the work carried out during this thesis are as follows:

- The best deployment points for the scenario in consideration are the pericentre of the [SSTO](#) target orbits. These points increase the margin of the spacecraft with respect to the escape velocity of the system and reduce the relative effect of injection errors.
- The target deployment point makes it impossible to achieve the cubesat injection using only one arc. The geometry of the problem with the phase angle constraint causes this.
- The higher the target deployment is with respect to Didymain, the more feasible reference trajectories are found. This is due to the minimum safety distance constraint, which plays a bigger role for lower trajectories.
- The margin to the escape velocity plays a big role in the trajectory design process. This optimisation constraint is active for all the reference trajectories found. This is caused by the need to reduce the difference between the spacecraft and the target orbit velocities as much as possible to allow the cubesat deployment with the limited  $\Delta v$  capabilities of the [CRS](#).
- A number of three stochastic firings are assessed to be the best choice, as they lead to around 50% reduced deployment conditions deviations and lower  $\Delta v$ . The firing conditions that yield the best deployment results are family-dependent. However, the trends found for all of them with respect to the firing times were almost identical, pointing out the similar behaviour against the guidance corrections.

- The safety of the reference trajectories under thruster failure conditions were tested, highlighting an actual danger of falling under the 2 cm/s margin over the escape velocity. However, this danger was caused by the combination of initial dispersions, dynamical uncertainties and command errors rather than by the thruster failures. The minimum distance to Didymain was only violated by the reference trajectory pointing at  $a_{SSTO} = 3$  km, which goes as low as 1500 metres.
- The uncertainties on the dynamical parameters of the system were assessed to have low impact on the deployment conditions. These cause much smaller deviations from the reference conditions than some of the other factors, which end up "hiding" the actual effect of these dynamical uncertainties.
- The navigation errors have proved to be one of the main sources of inaccuracy in the deployment conditions. This is especially true for the velocity navigation errors, which induce a velocity error component after each stochastic manoeuvre. The position knowledge errors are not so relevant in the thesis scenario, since the position dispersions in the system are usually much bigger than them.
- The velocity knowledge error coming from the navigation system is pointed out as the main source of deployment position inaccuracies.
- The combination of terminal position guidance and the velocity knowledge errors are a big driver of the deviations from the target deployment conditions, since velocity errors are never corrected for. The **PIG** is a robust guidance scheme for terminal position conditions. However, this robustness tends to cause big misalignments between the reference and the actual velocity of the controlled spacecraft due to the relatively big impulses commanded.
- The command errors, i.e. caused by the attitude knowledge and thruster misalignments, play a role in the deployment conditions. Their biggest contribution is to the in-plane and out-of-plane velocity difference angles.
- The initial condition dispersions are extremely relevant for the deployment conditions. The position dispersion is pointed out as the main source of misalignment between the target and actual spacecraft velocities at the deployment epoch due to the characteristics of the guidance scheme used (terminal position guidance).
- The success rate of the cubesat deployment manoeuvre has a peak of around 75% for the trajectory pointing at an orbit with  $a_{SSTO} = 5$  km. This maximum is reached for this trajectory because it presents the best balance between low objective function values and wide safety deployment intervals.

## 9.2. RECOMMENDATIONS

Even though the thesis has covered many of the aspects of the trajectory design in a thorough and consistent way, there are a few recommendations that can be made both as possible improvements for similar analysis and as a guideline for future work:

- The combination of different criteria for the safety deployment intervals considerably improves the results obtained. By further mapping the safe deployment conditions that are allowable, the study would be more representative of the real situation and the success rate results would improve.
- The safety margins imposed by the mission requirements are violated under the system uncertainties and dispersions. These margins should be reassessed to see what motivates them, and make them tighter if possible.
- If the reassessment of the safety margins does not yield any change, the spacecraft velocity margin to the escape velocity of the system could be consistently made bigger throughout the entire mission by equipping the **CRS** with a spring which is capable of giving a bigger  $\Delta v$ .
- Reducing the initial condition dispersions is considered as the alternative that would yield a higher success rate increase. This could be done by conducting a previous **TAP** with manoeuvres targeting reducing this dispersion, as it was for the previous **AIM** baseline.

- Another alternative to increase the success rate of the mission would be to improve the velocity navigation performance, although this is considered as a more difficult option. It must be noted that the navigation performance model used in this thesis is conservative in the values used to generate the navigation signals and thus, a refinement of these values could also yield higher success rates.
- The command errors were modelled in a oversimplified way that introduced high safety margins over the actual errors that would be expected from the attitude knowledge and thruster misalignments of a real spacecraft. A detailed modelling of these sources of errors could lead to more representative and higher estimated success rates, since they were identified as a relevant source of end-of-mission dispersions.
- Although relevant conclusions have been drawn from the thesis, no studies that take into account the actual interdependencies among more than two factors were carried out. The inclusion of such studies could increase the insight into the problem further.
- Monte Carlo sampling was used to characterised the system at hand. Other near-random sampling techniques like orthogonal sampling or Latin Hypercube sampling could yield better variable coverage for similar computational cost, which would increase the consistency of the analysis carried out at the cost of a more difficult implementation.



# A

## REFERENCE FRAMES TRANSFORMATIONS

In this appendix the reference frame transformation used during the thesis are presented. These transform a vector expressed in a reference frame [X] to a reference frame [Y]. Note that for the reverse transformation ([Y] -> [X]), the transpose of the matrix used for the [X] -> [Y] transformation can be used.

[I] TO [P]

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_P = \begin{pmatrix} -\sin \lambda & \cos \lambda & 0 \\ -\cos \lambda \sin \phi & -\sin \lambda \sin \phi & \cos \phi \\ \cos \lambda \cos \phi & \sin \lambda \cos \phi & \sin \phi \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}_I \quad (\text{A.1})$$

where  $\lambda$  and  $\phi$  are the ecliptic longitude and latitude of the Didymos pole with respect to the J2000 inertial ecliptic reference frame, respectively.

GENERIC FRAME ([L], [O], [D]) TO [I]

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_I = \begin{pmatrix} \hat{x}_G^1 & \hat{y}_G^1 & \hat{z}_G^1 \\ \hat{x}_G^2 & \hat{y}_G^2 & \hat{z}_G^2 \\ \hat{x}_G^3 & \hat{y}_G^3 & \hat{z}_G^3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}_G \quad (\text{A.2})$$

where the coordinates of the unit vectors defining the generic [G] reference frame expressed in [I] read as

$$\hat{\mathbf{x}}_G = \begin{pmatrix} \hat{x}_G^1 \\ \hat{x}_G^2 \\ \hat{x}_G^3 \end{pmatrix}_I, \hat{\mathbf{y}}_G = \begin{pmatrix} \hat{y}_G^1 \\ \hat{y}_G^2 \\ \hat{y}_G^3 \end{pmatrix}_I \text{ and } \hat{\mathbf{z}}_G = \begin{pmatrix} \hat{z}_G^1 \\ \hat{z}_G^2 \\ \hat{z}_G^3 \end{pmatrix}_I.$$

[P] TO [B1]

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{B1} = \begin{pmatrix} \cos \beta_{Main} & \sin \beta_{Main} & 0 \\ -\sin \beta_{Main} & \cos \beta_{Main} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}_P \quad (\text{A.3})$$

where  $\beta_{Main} = \dot{\beta}_{Main}(t - t_0) + \beta_{0,Main}$ . Here,  $\dot{\beta}_{Main}$  and  $\beta_{0,Main}$  denote the spin rate of the primary body and the initial condition of the prime meridian of Didymain with respect to  $\hat{\mathbf{x}}_P$ , respectively.

[P] TO [B2]

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{B2} = \begin{pmatrix} \cos \beta_{Moon} & \sin \beta_{Moon} & 0 \\ -\sin \beta_{Moon} & \cos \beta_{Moon} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}_P \quad (\text{A.4})$$

where  $\beta_{Moon} = \dot{\beta}_{Moon}(t - t_0) + \beta_{0,Moon}$ . Here,  $\dot{\beta}_{Moon}$  and  $\beta_{0,Moon}$  denote the spin rate of the secondary body and the initial condition of the prime meridian of Didymoon with respect to  $\hat{\mathbf{x}}_P$ , respectively.



## BIBLIOGRAPHY

- Ashino, R., Nagase, M., and Villancourt, R. (2000). Behind and beyond the matlab ode suite. *Computers and Mathematics with applications*, 40:491–512.
- Bayard, D. S. and Brugarolas, P. B. (2008). On-board vision-based spacecraft estimation algorithm for small body exploration. *IEEE*, 44(1).
- Bellerose, J. and Scheeres, D. (2005). The restricted hill full 4-body problem: Applications to spacecraft motion about binary asteroids. *Dynamical Systems*, 20:23–44.
- Berry, K., Sutter, B., Williams, A. M. K., Barbee, B. W., Beckman, M., and Williams, B. (2013). Osiris-rex touch-and-go (tag) mission design and analysis. In *36th Annual AAS Guidance And Control Conference, American Astronautical Society*.
- Conway, B. A. (2010). *Spacecraft Trajectory Optimization*. Cambridge University Press.
- Dell’Elce, L., Baresi, N., Naidu, S., Benner, L., and Scheeres, D. (2016). Numerical investigations of the dynamical environment of 65803 didymos. *Advances in Space Research*, 59:1304–1320.
- ESA-CDF (2014). *CDF Study Report AIM3P*. Available at: <http://gsp.esa.int/documents/10180/0/.../9755a7e0-73d5-41d2-95c9-dd2b32dbda87>.
- ESA-ESTEC-AD1 (2016). *[AD1] AIM System Requirements Document*. Internal communication, not publicly available.
- ESA-ESTEC-AD2 (2016). *[AD2] AIM Payload Interface Document*. Internal communication, not publicly available.
- ESA-ESTEC-RD27 (2016). *[RD27] AIM Proximity Operations Trajectories*. Internal communication, not publicly available.
- ESA-HQ-AD3 (2016). *[AD3] Asteroid Impact Mission: Didymos and Ejecta Reference Model*. Internal communication, not publicly available.
- Ferrari, F., Lavagna, M., and Carnelli, I. (2016). Coupling high fidelity body modeling with non-keplerian dynamics to design aim-mascot-2 landing trajectories on didymos binary asteroid. ICATT.
- GMV (2017). *[GMV-AIM-MA-ASPECT] GMV’s Mission Analysis on AIM ASPECT*. Internal communication, not publicly available.
- Hawkins, M., Guo, Y., and Wie, B. (2012). Spacecraft guidance algorithms for asteroid intercept and rendezvous missions. *International Journal of Aeronautical and Space Sciences*, 13:154–169.
- Hawkins, M., Pitz, A., Wie, B., and Gil-Fernandez, J. (2010). Terminal-phase guidance and control analysis of asteroid interceptors. In *AIAA GNC Conference, 2010-8348*.
- Izzo, D. (2010). *Global Optimization and Space Pruning for Spacecraft Trajectory Design, Spacecraft Trajectory Optimization*. Cambridge University Press.
- Näsilä, A., Viherkanto, K., Kestilä, A., Tikka, T., Kuhno, J., Kohout, T., and Granvik, M. (2016). Asteroid spectral imaging mission (aspect). Technical report, VTT Technical Research Centre of Finland Ltd.
- Pellacani, A., Kicman, P., Suatoni, M., Casasco, M., Gil, J., and Carnelli, I. (2017). Design and validation of a gnc system for missions to asteroids – the aim scenario. In *GNC 2017: 10th International ESA Conference on Guidance, Navigation & Control Systems*.

- Saboori, B., Bidgoli, A. M., and Saboori, B. (2013). Multiobjective optimization in repeating sun-synchronous orbits design for remote-sensing satellites. *Journal of Aerospace Engineering*.
- Scheeres, D. and Bellerose, J. (2008). Restricted full three-body problem: Application to binary system 1999 kw4. *Journal of Guidance, Control and Dynamics*, 31:160–171.
- Scheeres, D. and McMahon, J. (2013). Dynamic limits on planar libration-orbit coupling around an oblate primary. *Celestial Mechanics and Dynamical Astronomy*, 115:365–387.
- Scheeres, D. J. (2012). *Orbital Motion in Strongly Perturbed Environments: Applications to Asteroid, Comet and Planetary Satellite Orbiters*. Springer-Verlag, first edition.
- Scheeres, D. J. and McMahon, J. (2015). Improving space object catalog maintenance through advances in solar radiation pressure modeling. *Journal of Guidance, Control and Dynamics*, 38(8):1366–1381.
- Schlueter, M. (2017). Midaco software performance on interplanetary trajectory benchmarks. *Preprint submitted to Advances in Space Research*.
- Schlueter, M., Egea, J. A., Antelo, L. T., Alonso, A. A., and Banga, J. R. (2009). An extended ant colony optimization algorithm for integrated process and control system design. *Industrial & Engineering Chemistry Research*, 48:6723–6738.
- Schlueter, M. and Gerdtts, M. (2010). The oracle penalty method. *Journal of Global Optimization*, 47:293–325.
- Schlueter, M. and Munemoto, M. (2014). Parallelization for space trajectory optimization. In *Congress on Evolutionary Computation, IEEE*.
- Schlueter, M. and Munetomo, M. (2016). *MIDACO-SOLVER User Manual*.
- Schulte, P., Alegret, L., Arenillas, I., Arz, J. A., Barton, P. J., Bown, P. R., Bralower, T. J., Christeson, G. L., Claeys, P., Cockell, C. S., Collins, G. S., Deutsch, A., Goldin, T. J., Goto, K., Grajales-Nishimura, J. M., Grieve, R. A. E., Gulick, S. P. S., Johnson, K. R., Kiessling, W., Koeberl, C., Kring, D. A., MacLeod, K. G., Matsui, T., Melosh, J., Montanari, A., Morgan, J. V., Neal, C. R., Nichols, D. J., Norris, R. D., Pierazzo, E., Ravizza, G., Rebolledo-Vieyra, M., Reimold, W. U., Robin, E., Salge, T., Speijer, R. P., Sweet, A. R., Urrutia-Fucugauchi, J., Vajda, V., Whalen, M. T., and Willumsen, P. S. (2010). The chicxulub asteroid impact and mass extinction at the cretaceous-paleogene boundary. *Science*, 327(5970):1214–1218.
- Shampine, L. and Reichelt, M. (1997). The matlab ode suite. *SIAM Journal on Scientific Computing*, 18:1–22.
- Socha, K. and Dorigo, M. (2008). Ant colony optimization for continuous domains. *European Journal of Operational Research*, 185:1155–1173.
- Stryck, O. and Bulirsch, R. (1992). Direct and indirect methods for trajectory optimization. *J.C. Baltzer AG, Scientific Publishing Company*, 37:357–373.
- Tsuda, Y., Yoshikawa, M., Abe, H. M. M., and Nakazawa, S. (2013). System design of the hayabusa 2 - asteroid sample return mission to 1999 ju3. *Acta Astronautica*, 91:356–362.
- Vallado, D. A. (1997). *Fundamentals of Astrodynamics and Applications*. McGraw-Hill, SpaceTechnology Series.
- Wakker, K. F. (2015). *Fundamentals of Astrodynamics*. Delft University of Technology.
- Wertz, J. R. (2009). *Orbit & Constellation Design & Management*. Springer.