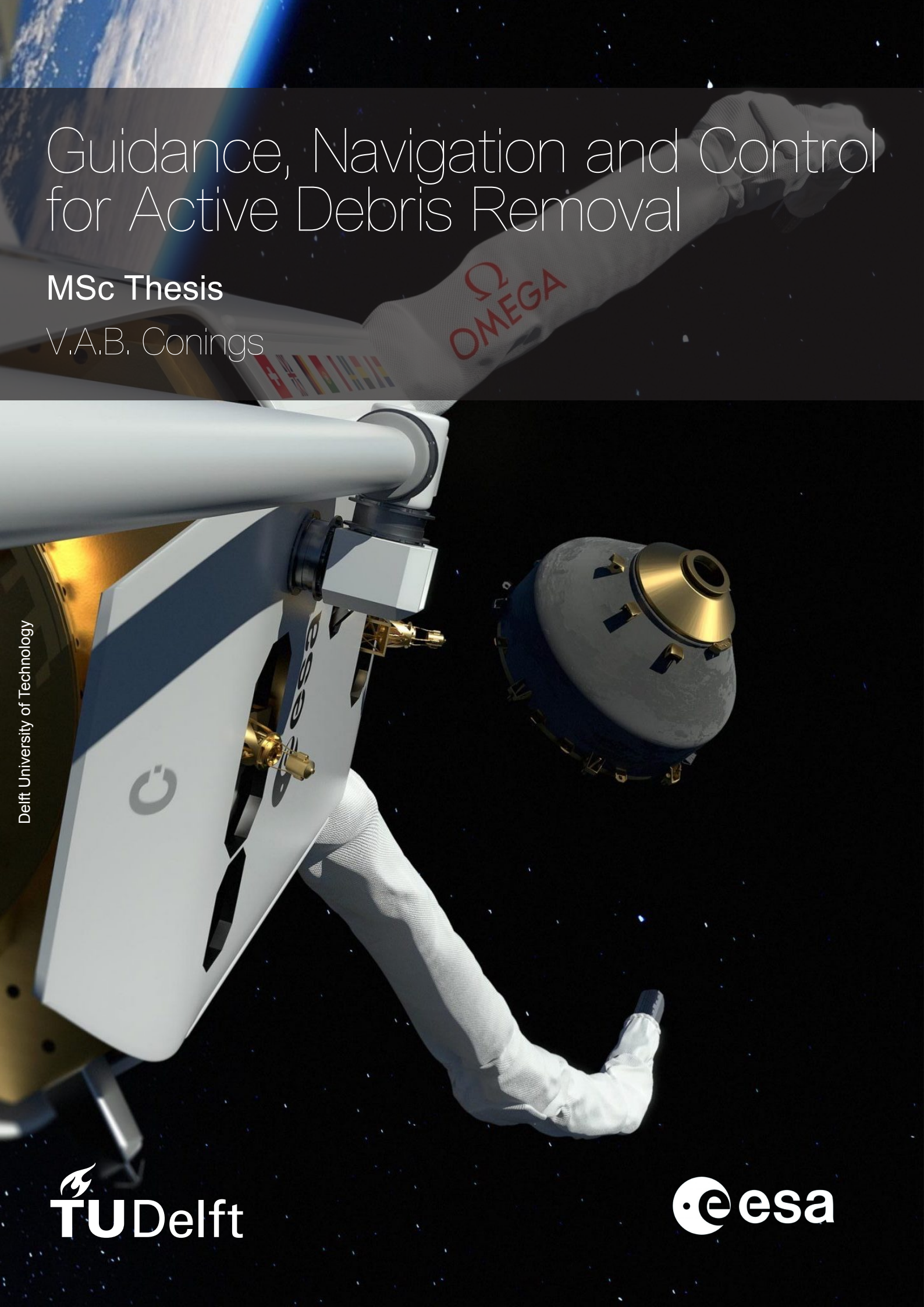
Guidance, Navigation and Control for Active Debris Removal

MSc Thesis

V.A.B. Conings

Delft University of Technology



TU Delft

esa

Guidance, Navigation and Control for Active Debris Removal

by

V.A.B. Conings

in partial fulfilment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on Friday, October 28, 2022 at 10 am.

Supervisor:	Dr. ir. E. Mooij	TU Delft	examiner
Thesis committee:	Dr. ir. W. van der Wal	TU Delft	chair
	Dr. ir. E. van Kampen	TU Delft	external examiner
	ir. I. Huertas Garcia	ESA	external member

An electronic version of this thesis is available at https://repository.tudelft.nl

Cover Image taken from ClearSpace SA: https://clearspace.today

TU Delft

esa

”*Houston, Tranquillity Base here. The Eagle has landed.*”
Neil Armstrong

Preface

With the launch of the first ever artificial satellite in 1957, the grips of humankind could finally reach into space and a new era filled with technological advancements was initiated. However, this day also marked a historic milestone because of quite different reasons. The environmental impact of humankind evolved from terrestrial to celestial, as the direct vicinity around Earth got progressively filled with man-made objects. Presently, the pollution of space has come so far that a saturation level is reached, where the number of space debris items will further grow due to collisional interaction between them. Therefore, the space industry has recognised that space debris and its mitigation is a pressing topic requiring drastic actions, such that humankind is not losing its access to space and the accompanying benefits that come from it. However, despite the clear urgency related to this matter, one cannot help but notice that the first actual debris removal mission still needs to be launched. Therefore, to contribute to the execution of active debris removal, the current thesis study is performed.

This thesis report represents the apotheosis of all my efforts in obtaining a Master's degree in Space Flight engineering from the faculty of Aerospace Engineering, at the Delft University of Technology. For this thesis, a study on the guidance, navigation and control system of an active debris removal mission is performed. The selected mission is presented by the ClearSpace-1 project, a pioneering mission set out to perform the first ever capture and deorbiting of a resident space object. Within the current thesis, the most noticeable development is presented by a novel guidance system, which applies the mathematical concepts of convex optimisation theory to the physical problem of orbital rendezvous and docking, with the purpose of obtaining an autonomous guidance system onboard the chaser spacecraft. Therefore I am pleased to announce that a contribution to the realisation of an actual debris removal mission could be provided, even when this contribution is ever so slightly, by analysing relevant models and integrating a convex guidance scheme with a realistic navigation and control system. Furthermore, I hope that, with the current thesis report, I can share my excitement for guidance, navigation and control systems of spacecraft in active debris removal missions.

To perform this thesis study, a number of people provided me with the necessary guidance and support. First and foremost, I would like to express my gratitude to my supervisor, Dr. Erwin Mooij, for presenting this exciting research opportunity, for sharing his work methodology and for keeping me on a trajectory towards success. Our vivid discussions taught me that there are no limitations to what can be achieved, as long as an open and positive attitude is adopted. Rather than seeing limitations, Dr. Mooij focuses on opportunities and because of this, I value his opinion above all others. Furthermore, I would like to thank Dr. Ralph Kahle and the entire Flight Dynamics department of the German Space Operations Centre in Oberpfaffenhofen. During my internship there, they stimulated my personal and academic growth and taught me invaluable lessons that could be applied either directly or indirectly during the execution of this thesis. Lastly, I would also like to thank my family and friends for their unconditional love and support, not only throughout this challenging process, but throughout the entire duration of my higher education.

With humans becoming prisoners of Earth when excessive space debris denies us our access to space, it is evident that actions have to be undertaken now, and a potential first step towards such an action could be provided by the current thesis. Therefore, sit back, relax and enjoy as a marvellous endeavour, that seeks to refrain this doom scenario from happening, is analysed.

V.A.B. Conings
Delft, October 2022

Abstract

To ensure long-term sustainable access to space, it is of paramount importance that humankind urgently acts to alleviate the space debris problem that is currently faced in the near-vicinity of Earth. From research it was found that the most effective mitigation strategy is provided in terms of an active debris removal mission, which is capable of removing one or more resident space objects from the most popular orbital regions around Earth. A potential first of such missions is presented by the ClearSpace-1 mission, which is tasked with deorbiting the upper part of a Vega payload adaptor. Note that this specific target is chosen, as it provides a relatively simple and sturdy geometry and is therefore used to gain relevant experience before continuing with more complex debris objects. The execution of such a complex mission does, however, come with distinct technological challenges, which are mainly related to the development of an advanced guidance, navigation and control system.

The current thesis provides a first step in the development of such a system, by providing a design for the ClearSpace-1 mission. The considered mission scenario is limited to the close-range rendezvous phase, where the chaser spacecraft approaches its target up to the capture position. To be able to achieve this, a prime focus is put on the development of an advanced guidance system, which combines a convex guidance algorithm with an reference tracking guidance system, to obtain a robust, autonomous and fuel-efficient execution of a close-range rendezvous scenario. The implemented convex guidance system is shown to find the fuel-optimal trajectory that connects the initial conditions with the final conditions, while adhering to the implemented path and state constraints given by keep-out spheres , approach cones and bounded thrust and velocity magnitudes. Furthermore, it is shown that the developed guidance algorithm is capable of executing all mission scenarios that can be anticipated for the ClearSpace-1 missions, as well as contingency scenarios, during which mission parameters are varied. Note however that, due to the sensitivity of the guidance system to the output accuracy of orbit control system, ideal thruster performance must be assumed. Therefore, to increase the realism of the presented system, a thruster calibration algorithm should be developed when combining the convex guidance system with a non-ideal thruster system. Moreover, as the implemented guidance system only controls the translational motion of the chaser spacecraft, an Incremental Nonlinear Dynamic Inversion controller is used to ensure attitude synchronisation between both spacecraft prior to capture. During a nominal and worst-case tumbling scenario of the target spacecraft, results indicated that the attitude controller is capable of providing effective synchronisation, while requiring limited control effort. A signal smoother at the start-up of the attitude control system should, however, still be implemented to improve the control performance by removing spikes from its input history. Lastly, an Unscented Kalman Filter is applied to obtain an inertial navigation solution, while an artificial Vision-based Navigation system is used to obtain a relative navigation solution. Nevertheless, both are capable of attaining high navigation accuracies by obtaining measurements from high performance navigation sensors.

Subsequent analysis on the integration of the developed guidance, navigation and control system indicated that a generic active debris removal mission can be executed by the proposed system design, and would result in a terminal position dispersion of approximately 0.1 m, while requiring a propellant mass of at most 2 kg. Therefore, results display the promising performance of the presented system design for a future implementation on the ClearSpace-1 removal mission. Moreover, as the proposed system provides straightforward scaling, the generic rendezvous scenario can further be applied to equivalent active debris removal missions of other rocket bodies, or even dissimilar missions to defunct satellites. Nevertheless, some further developments are necessary and highly encouraged.

Nomenclature

Abbreviations

Abbreviation	Definition
ADR	Active Debris Removal
ARCSS	Autonomous Rendezvous and Capture Sensor System
ATV	Automated Transfer Vehicle
AVGS	Advanced Video Guidance Sensor
DARPA	Defence Advanced Research Projects Agency
EKF	Extended Kalman Filter
ESA	European Space Agency
GDOP	Geometric Dilution of Precision
GNC	Guidance, Navigation and Control
GPS	Global Positioning System
GRADS	Generic Rendezvous and Docking Simulator
IADC	Inter-Agency Space Debris Coordination Committee
I/F	Interface
IMU	Inertial Measurement Unit
INDI	Incremental Nonlinear Dynamic Inversion
ISS	International Space Station
LIRIS	Laser and Infra-Red Imaging Sensors Demonstrator
LRF	Laser Rangefinder
MIMO	Multiple-Input Multiple-Output
MRP	Modified Rodrigues Parameters
NASA	National Aeronautics and Space Administration
NDI	Nonlinear Dynamic Inversion
NFOV	Narrow-Field-Of-View
PID	Proportional, Integral and Derivative
SGLS	Space-Ground Link System
SIGI	Space-Integrated GPS/INS
SMRP	Shadow Modified Rodrigues Parameters
SOCP	Second-Order Cone Programming
SRP	Solar Radiation Pressure
UKF	Unscented Kalman Filter
VBN	Vision-based Navigation
VESPA	Vega Secondary Payload Adapter

Calligraphic Symbols

Symbol	Definition
$\mathcal{F}$	Reference frame
$\mathcal{F}_{B,C}$	Body-Centred-Body-Fixed frame of chaser
$\mathcal{F}_{B,T}$	Body-Centred-Body-Fixed frame of target
$\mathcal{F}_{H}$	Hill frame
$\mathcal{F}_{I}$	Earth-Centred inertial frame
$\mathcal{F}_{L}$	Local-Vertical-Local-Horizontal body-centred frame
$\mathcal{O}()$	Order of

Greek Symbols

Symbol	Definition	Unit
α	Approach cone half-angle	[rad]
α	Spread parameter of sigma points in UKF	[-]
$\Delta..$	Error	[-]
$\delta..$	Difference	[-]
$\delta\mathbf{u}$	Incremental control command	[-]
$\delta\boldsymbol{\sigma}$	MRP error vector	[-]
$\delta\mathbf{q}$	Quaternion error vector	[-]
ε	Azimuth angle measured by radar	[rad]
ε	Scaling parameter	[-]
ζ	Damping ratio	[-]
$\boldsymbol{\eta}$	Outward unit normal of rotating-hyperplane	[-]
κ	Secondary scaling parameter in UKF	[-]
λ	Scaling parameter in UKF	[-]
μ	Gravitational parameter of Earth	[m^3/s^2]
$\mathbf{v}$	White noise vector	[-]
ρ	Atmospheric density	[kg/m^3]
ρ	Range	[m]
$\dot{\rho}$	Range rate	[m/s]
$\boldsymbol{\sigma}$	MRP vector	[-]
$\boldsymbol{\sigma}^S$	SMRP vector	[-]
σ	Magnitude of MRP vector	[-]
σ	Standard deviation	[-]
τ	Time constant	[s]
$\boldsymbol{\Upsilon}$	Propagated sigma-point vector	[-]
Φ	Rotation angle over Euler eigenaxis	[rad]
$\boldsymbol{\chi}$	Sigma-point vector	[-]
$\boldsymbol{\omega}$	Angular velocity vector	[rad/s]
$\dot{\boldsymbol{\omega}}$	Angular acceleration vector	[rad/s^2]
ω_n	Cut-off frequency	[Hz]

Roman Symbols

Symbol	Definition	Unit
$\mathbf{0}_{N\times N}$	N-by-N matrix containing zeroes	[-]
$\mathbf{1}_N$	N-dimensional column vector containing ones	[-]
$\mathbf{a}$	Acceleration vector	[m/s^2]
$\mathbf{b}$	Bias vector	[-]
$\mathbf{C}$	Transformation matrix	[-]
c	Speed of light	[m/s]
$c()$	Cosine function	[-]
d	Drift	[m/s]
$\hat{\mathbf{e}}$	Euler eigenaxis	[-]
$\mathbf{e}_d$	Docking-axis	[-]
$\mathbf{F}$	Force vector	[N]
$f(), g(), h()$	Generic function representations	[-]
f_x, f_y, f_z	Non-gravitational forces acting on spacecraft per unit mass	[N/kg]
$\mathbf{g}$	Gravitational acceleration vector	[m/s^2]
$H(s)$	Transfer function	[-]
h	Step size	[s]

Symbol	Definition	Unit
$\mathbf{I}_{N\times N}$	N-by-N identity matrix	[-]
I	Electrical current	[A]
$\mathbf{J}$	Inertia tensor	[kgm^2]
J_f	Objective function of optimisation problem	[-]
$\mathbf{K}$	Kalman gain matrix	[-]
K_p, K_i, K_d	Proportional, integral and differential gains	[-]
k_c, k_v	Coulomb and viscous friction coefficients	[-]
k_m	Motor constant	[Nm/A]
$\mathbf{M}$	Moment vector	[Nm]
m	Instantaneous mass of the body	[kg]
m	Misalignment term in navigation filter	[-]
n	Mean motion of target satellite in orbit	[rad/s]
$\mathbf{P}$	Covariance matrix	[-]
$\bar{\mathbf{P}}$	Covariance matrix of propagated state	[-]
$\hat{\mathbf{P}}$	Covariance matrix of measurement-corrected state	[-]
$\mathbf{P}_{xy}$	Cross-covariance matrix of state and measurement errors	[-]
$\mathbf{Q}$	Process noise covariance matrix	[-]
$\mathbf{Q}$	Auxiliary matrix of quaternion kinematics	[-]
$\mathbf{R}$	Observation noise covariance matrix	[-]
$\mathbf{r}$	Cartesian position vector	[m]
r_{KOS}	Keep out sphere radius	[m]
$\mathbf{S}()$	Anti-symmetric matrix	[-]
s	Laplace domain variable	[-]
s	Scaling term in navigation filter	[-]
$s()$	Sine function	[-]
$\mathbf{T}$	Thrust vector	[N]
T_{nom}	Nominal thrust magnitude	[N]
$\mathbf{t}_{act}$	Activation time vector	[s]
t_f	Time of flight	[s]
$\mathbf{u}$	Control input vector	[-]
$\mathbf{V}$	Cartesian velocity vector	[m/s]
$\mathbf{v}$	Virtual control input vector	[-]
$\mathbf{v}$	Process noise vector	[-]
$\mathbf{w}$	Observation noise vector	[-]
$\mathbf{x}$	State vector	[-]
$\bar{\mathbf{x}}$	Propagated state vector	[-]
$\hat{\mathbf{x}}$	Measurement-corrected state vector	[-]
X, Y, Z	X-, Y- and Z-axes	[-]
$\mathbf{y}$	System output vector	[-]
$\bar{\mathbf{y}}$	Observation vector estimate	[-]
z	Discrete time Laplace domain variable	[-]

Subscripts

Subscript	Definition
0	Initial
$1, 2, 3, 4$	Vector components
att	Attitude
b	Bias
c	Chaser spacecraft
cm	Centre of mass
cmd	Commanded

Subscript	Definition
des	Design
e	Estimated
elec	Electrical
f	Final
in	Innerloop
max	Maximum
meas	Measured
out	Outerloop
pos	Position
rate	Angular rate
RCT	Reaction control thruster
ref	Reference
rel	Relative
rot	Rotational
RW	Reaction wheel
SS	Star sensor
sat	Satellite
t	Target spacecraft
trans	Translational
VBN	Vision-based navigation
velo	Velocity
w	Wheel

Contents

1 Introduction

With the start of the space flight era in 1957, humankind has continuously filled up the orbital regions in close proximity to Earth with space debris (Forshaw et al., 2017). Given that these man-made objects have an operational lifetime that spans at most a few decades, a continuous replenishment of new satellites was required. This meant launching new satellites into orbit, without removing the defunct ones. Consequently, the far majority of objects around Earth is currently composed of non-functional debris object consisting of: nonoperational satellites, burnt rocket stages and artificial satellite fragmentations. Moreover, the object density in the direct vicinity around Earth is increased with every new satellite that is being launched, which in turn increases the collision probability between artificial satellites, and therefore further increases the accumulation of space debris. Currently, NASA's Space Surveillance Network estimates that there are close to 23,000 pieces of space debris with a diameter of 10 cm or larger, about half a million pieces of space debris with a diameter of 1 cm or larger and approximately 100 million pieces of debris with a diameter of 1 mm or larger (Garcia, 2021). As this number greatly surpasses the number of operational satellites in orbit, currently estimated to be around 3400 (UCS, 2021), the urgency associated with controlling the orbital debris problem becomes apparent.

When a specific orbital region around Earth becomes saturated, *i.e.*, the population density exceeds a given threshold, a chain reaction of collisions will be initiated that results in an unbounded growth of debris items. This is formally known as the Kessler-effect (Kessler and Cour-Palais, 1978) and research has shown that the current spatial environment around Earth has already reached this saturation point, where the Kessler-effect has been initiated by two major collisions, which occurred in 2007 and 2009 (Liou and Johnson, 2008). Furthermore, on 15 November 2021, an anti-satellite test was executed by Russia, during which it destroyed one of its defunct satellites, causing yet another major fragmentation in this already delicate orbital region. During this anti-satellite test, over 1500 debris objects with a diameter of 10 cm or larger were generated, and hundreds of thousands smaller pieces, as estimated by The United States Space Command (Raju, 2021). Hence, to ensure that the orbital regions in proximity of Earth remain available for long-term space operations, mankind is compelled to take an active approach in cleaning this spatial junkyard.

Currently, the most effective solution strategy for larger debris objects is provided in terms of an Active Debris Removal (ADR) mission (Liou and Johnson, 2009). During such a mission, an active chaser spacecraft is sent up to space with the purpose of capturing and removing a passive debris item from orbit. A potential first mission of this type is presented by the ClearSpace-1 mission, which is planned to be launched by 2025. Note that, in contrast to all previously flown ADR missions, ClearSpace-1 is tasked with deorbiting a preexisting debris object, rather than a dummy spacecraft carried onboard. Evidently, this comes will substantial technological challenges and therefore provides the reference mission for the current thesis study.

During the execution of the ClearSpace-1 mission, the active chaser spacecraft is tasked with approaching, capturing and deorbiting the Vega Secondary Payload Adapter, a burnt rocket stage that was left in low Earth orbit during the second flight of the Vega rocket in 2013. Throughout the entire mission duration, only the chaser spacecraft can act, given that its target is truly passive. Moreover, as passive debris item, it has partially unknown mass and inertia properties and is likely to be tumbling. Additionally, the target is uncooperative throughout the entire mission duration and not designed for rendezvous and docking with its active chaser spacecraft (Bonnal et al., 2013). Nevertheless, capture of such a resident space object can be achieved when the chaser spacecraft is equipped with an advanced guidance, navigation and control (GNC) system, motivating the research of the present thesis study.

The current report is aimed at providing a detailed discussion on all the work that is performed during the thesis project. To this extent, an analysis is performed that focuses on the development of a convex guidance algorithm and its incorporation into a dedicated GNC system for an ADR mission. Therefore, an answer on the following research question is ought to be found.

- *How can a real-time guidance, navigation and control system be applied robustly, safely and autonomously to execute a fuel-efficient final rendezvous between an active chaser spacecraft and a passive debris item?*

From this research question, a number of related subquestions can be derived and are provided below. Given that the type of guidance methodology has already been selected by ESA in (European Space Research and Technology Centre, 2020), where a convex guidance algorithm is defined for the ClearSpace-1 mission, it becomes interesting to investigate why this specific type of guidance system is requested. To do so, the performance benefits that are obtainable from this specific guidance methodology are analysed in terms of its robustness, safety, autonomy and fuel-efficiency.

- *What performance in terms of robustness, safety, autonomy and fuel-efficiency can be obtained when employing a convex guidance system?*

Furthermore, given that previous thesis projects, which were executed at the Delft University of Technology and were concerned with the development of a convex guidance system, solely developed the convex guidance methodology or its combination with a simple control system, it is required for the current thesis project to take this research one step further. Therefore, it is necessary to analyse the effect that the integration of the convex guidance system with a realistic control and navigation system has on its performance.

- *What is the effect on the GNC system performance when implementing the convex guidance system with a realistic navigation and control system?*

Related to this, it is required to analyse what establishes the minimum sensor and actuator suites that can provide successfull implementation of the convex guidance system in an integrated GNC system. This is because the selection of the respective control and navigation algorithms readily follows from the considered reference scenario, where a UKF estimation filter has been chosen due to its derivative-free nature, and an INDI controller for its application of linear control techniques to a nonlinear system without requiring an accurate system model. Therefore, instead of carrying out a detailed analysis on the performance of numerous navigation and control systems in combination with the convex guidance system, a study is performed on the minimum sensor and actuator performance that is needed to achieve successfull integration of the convex guidance system with the previously selected navigation and control systems. Hence, the performance requirements of the sensor suite and control actuator setup of the ClearSpace-1 satellite are sought.

- *What minimum sensor suite is required to allow successfull operation of the GNC system?*
- *What performance is required from the navigation sensors to allow successfull operation of the GNC system?*
- *What minimum actuator setup is required to allow successfull operation of the GNC system?*
- *What performance is required from the control actuators to allow successfull operation of the GNC system?*

Answers to these questions are provided in the remainder of this report, which is structured as follows. First of all, Chapter 2 is concerned with providing the research context of the current thesis study, by motivating the need for active debris removal missions. Moreover, this chapter provides the state of the art for GNC systems from past and current rendezvousing missions and selects the ClearSpace-1 mission as reference scenario for the current thesis study. The chapter is concluded with a listing of the relevant system and mission requirements that flow down from the baseline mission selection, together with an outline of the presented thesis study. To define, model and analyse the motion of the target and chaser spacecraft during the considered mission scenario, the definition of reference frames and state representations is needed. A definition of these is provided in Chapter 3, together with the orbital mechanics that are used to simulate the ClearSpace-1 mission. Additionally, the chapter defines

the implemented motion simulator by selecting the relevant environmental conditions, and verifies the derived acceleration and torque models.

A discussion on the GNC system of the ClearSpace-1 satellite is subdivided into three chapters, where each subsystem is separately detailed. First, the guidance system is provided in Chapter 4. In this chapter, the different segments of the mission are presented, along with a discussion on the specific guidance methodologies that are developed for each of these mission segments. Second, Chapter 5 details the navigation system of the ClearSpace-1 satellite by providing its inertial and relative navigation algorithms, as well as the respective navigation sensors that are employed by these algorithms. Third, an elaboration on the control system is presented in Chapter 6, where the developed control algorithms are discussed jointly with the control actuators that are selected to execute their control commands. Note that, for each of these three chapters, first the theory behind the algorithms is provided, after which a development of the respective subsystems is performed to fit the specifics of the ClearSpace-1 mission. Ultimately, the three chapters are concluded with a verification of the subsystems' implementation.

Furthermore, to combine the previously developed subsystems into one integrated GNC system, a functional simulator is developed. A discussion of this simulator is provided in Chapter 7, where the software architecture is provided, as well as the integration tests that are performed to ensure the correct functionality of the combined GNC system. Subsequently, Chapter 8 presents the results that can be obtained from the integrated GNC system when applying it to the expected mission scenarios of the ClearSpace-1 mission. Moreover, this chapter also discusses some preliminary robustness analyses of the developed system, where its flexibility is assessed to varying mission conditions.

Lastly, Chapter 9 provides the conclusions and recommendations that follow from the current thesis study. It represents a culmination of all the work that is carried out during the thesis project, providing an answer to the research questions, as given above, along with recommendations for future work.

2
Research Context

The current chapter identifies the research context within which this thesis study can be positioned. An outline of the present situation in the vicinity of Earth is provided, along with a discussion on the space debris that is currently intrinsic to this region, in Section 2.1. Furthermore, without providing a detailed analysis on the accumulation of space debris around Earth, this section also identifies the clear need for active debris removal missions and the distinct technological challenges that are related to such endeavours. As the GNC system is a key technology enabler, its study in the current paper is justified. Subsequently, relevant previous missions are analysed in a survey emphasised on their GNC system in Section 2.2. From this survey, a potential first actual active debris removal mission is selected in Section 2.3 and used as the baseline mission scenario for the current research. Furthermore, a listing of relevant mission and system requirements that flown that from this mission selection is provided in Section 2.4. Lastly, an outline of the current thesis study is provided in Section 2.5, where a discussion is provided on the topics that are currently omitted from the research.

2.1. Space Debris and Its Removal

It has widely been recognised that human pollution of space is one of the most challenging problems the space industry is currently facing. This is because the accumulation of debris will jeopardise the use of popular operational orbits, such as geostationary orbits and the low-Earth orbital region. In these densely populated areas, there is an elevated risk of collision, resulting in an inevitable increase in the growth of space debris, as given by the Kessler-effect (Kessler and Cour-Palais, 1978). The current spatial environment in the vicinity of Earth has reached the saturation point where, even when no additional objects are launched, the growth of space debris will further increase due to collisional interaction between the debris objects (Liou and Johnson, 2008). Therefore, to limit the consequences resulting from the initiation of the Kessler-effect, there is a general incentive within the space industry to control and mitigate space debris. For instance, one of such incentives is given by the Clean Space initiative of the European Space Agency (Innocenti et al., 2013).

To limit the accumulation of additional space debris, there are international regulations instituted by the Inter-Agency Space Debris Coordination Committee (IADC). These regulations relate to the end-of-life scenario of spacecraft in low-Earth and geostationary orbits (United Nations Office For Outer Space Affairs, 2010). However, these regulations only apply to recently launched spacecraft and are thus intended to reduce future growth of space debris, but cannot nullify this. Therefore, to stabilise the growth of space debris, the space industry has to resort to active debris removal systems capable of disposing large chunks of recent and less recent junk. It has been shown by Liou and Johnson (2009) that active disposal of existing bulky and heavy debris results in an efficient manner to significantly reduce the growth of space debris over time, as can be seen in Figure 2.1.

In this figure, the number of debris objects larger than 10 cm, in a low Earth orbit between 200 and 2,000 km, is simulated over time for four different scenarios; the first one being business as usual without active debris removal and the other three implement an active removal of five, ten and fifteen objects per year starting from 2020. From this figure, it can be seen that there is a strong near-exponential growth in case no-mitigation is applied, whereas this growth is significantly reduced when as few as five objects per year are actively removed. Moreover, it should be noted that the starting year of active debris removal is flexible to the extent that it should remain within the linear growth segment of the curve, such that the future behaviour shifts accordingly. Postponing the starting date to the exponential segment of the curve is not possible, as that will significantly decrease the efficiency of active debris removal actions (Liou and Johnson, 2009).

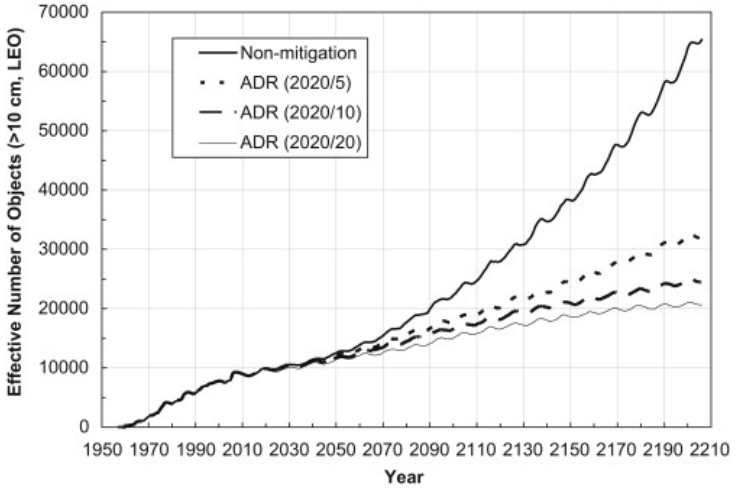


Figure 2.1: Simulated LEO debris populations between 1957 and 2006 (historical), and between 2007 and 2206 (future projection) as provided by Liou and Johnson (2009)

Based on the discussion above, it can be concluded that there is a clear need for ADR to get control over the growth of space debris around Earth. Nevertheless, this active remediation strategy for past and current satellite missions has to be combined with mitigation strategies for future satellite missions, *i.e.*, adhering to the guidelines set out by the IADC. Moreover, to perform this active remediation strategy as efficiently as possible, simulations identified critical regions around Earth where the risk for collision is highest. These regions are specified by Sellmaier et al. (2010) as circular low Earth orbits, with an altitude between 800 and 1,000 km and a high inclination angle, such that a specific removal mission can be designed to dispose a particular debris object.

Despite the pressing nature of the space debris issue, the first ever actual ADR mission is still to be launched. Hence, it is interesting to investigate why such missions have not been developed in the past, even though a need for them is evident. On one hand, the urgency for active debris removal missions only became apparent over time, as space debris accumulated, whereas on the other hand, the design of such an active debris removal mission brings distinct technological challenges that first have to be overcome. One of the most complex challenges is provided by the rendezvous of the chaser spacecraft with the target debris, as this debris object can have partially unknown mass and inertia properties, can be tumbling, is not cooperating nor designed for rendezvous and docking with the chaser (Bonnal et al., 2013). Note that this is in very strong contrast with all missions to date that have been capable of performing successfull rendezvous between two spacecraft, as the attitude of both actors was controlled, such that no tumbling of one spacecraft with respect to the other occurred. Furthermore, the target spacecraft was cooperating with its chaser through sensors or reflectors, aiding the approach and rendezvous, and both spacecraft were specifically designed to established a rigid connection upon rendezvous, by use of dedicated docking ports. This therefore shows the advancements that are required in terms of the GNC system to overcome the tumbling and non-cooperative nature of the target spacecraft, and provides a motivation as to why the GNC design for an active debris removal mission is the topic of the current research.

2.2. Mission Heritage

To gain insight in the physics behind the approach and docking of two satellites in orbit, the rendezvous mission heritage that exists from past and current missions is analysed. Note that these missions are similar, but not identical to a rendezvous mission with a non-cooperative satellite. Instead, these missions provide a sound preliminary insight into the general problem of rendezvousing two objects in space and represent a concise summary on rendezvousing missions that where identified as relevant by the author. Note that an emphasis is put on missions containing some degree of autonomy in the rendezvous and docking phase, as an increased level of autonomy is required to obtain a low decision time for active collision avoidance during an ADR mission (Peters et al., 2013).

2.2.1. Engineering Test Satellite ETS-VII

The first autonomous rendezvous and docking was performed by the Japanese Engineering Test Satellite ETS-VII in 1998 (ESA, 2021b). The mission consisted of a dual-satellite configuration containing a large chaser satellite equipped with a 6 degree-of-freedom robotic arm and a smaller target satellite. The purpose of this mission was to demonstrate an autonomous rendezvous and docking capability between two spacecraft, as well as to demonstrate the use of robotic manipulators in space. During the rendezvous and docking experiments, the target satellite was undocked from its chaser such that both spacecraft drifted apart, after which the autonomous rendezvous and docking sequence was initiated and both systems were reattached. In total three different experiments have been conducted in flight: a detachment and formation flight with a relative separation distance of 2 m, a detachment and subsequent separation of 12 km and a test where the rendezvous and docking between both spacecraft, as well as a collision avoidance manoeuvre, was remotely controlled by ground operators. After successfull completion of all these experiments, the spacecraft operational lifetime was extended and an additional experiment was performed in which the robotic manipulator was used to capture the free floating target satellite in a fully autonomous manner. Note that this was a groundbreaking revolution for spacecraft proximity operations, as it was the first time in history that autonomous capture and docking between two spacecraft occurred without interference from ground operators. The spacecraft were finally decommissioned in 2002 after an operational lifetime of nearly five years.

To ensure that the autonomous rendezvous and docking occurred in a safe manner, the designers opted for low impact docking over hard impact docking (Kawano et al., 1999). This resulted in a low closing velocity of only 1 cm/s, whereas hard impact docking is generally achieved with approach velocities between 5 and 10 cm/s. Additionally, low impact docking was preferred over hard, as it allowed for easier collision avoidance manoeuvres during contingency operations. To enable the autonomous rendezvous and docking experiments, the chaser spacecraft was equipped with the following hardware: the guidance computer, GPS receivers, a rendezvous radar, a camera-type proximity sensor, an accelerometer and the docking mechanism, whereas the target spacecraft was equipped with: GPS receivers, passive rendezvous radar reflectors, a proximity sensor reflector and handles for the docking mechanism (Kawano et al., 1999). Based on this distribution of hardware, it is evident that the chaser spacecraft performed the rendezvous and docking actively, in contrast to the target, which operated pseudo-passively, given that it only performed attitude control. Furthermore, the chaser spacecraft contained navigation software capable of determining relative positioning between both spacecraft. To do so, three navigation algorithms based on an Extended Kalman filter approach were available, with each their specified operational regime. When performing far range rendezvous, for relative distances between 10 km to 500 m, navigation was performed using GPS data. When performing near range rendezvous, for relative distances between 500 m and 2 m, navigation was performed using rendezvous radar data. Finally, during the terminal phase within 2m of the target, navigation was performed using proximity sensor data and provided a key technology demonstration, as this was capable of calculating the relative position and attitude using images processing algorithms.

2.2.2. XSS-10 and XSS-11

The Experimental Spacecraft Systems 10 and 11, labelled as the XSS-10 and XSS-11 missions, were an American microsatellite series providing technology demonstrations for autonomous proximity operations with a resident space object (ESA, 2021f). The first of these two satellites, XSS-10, was launched in 2003 onboard a Delta-2 launch vehicle. After reaching orbit, the XSS-10 satellite separated from the Delta-2 second stage and used this as the resident space object. As a technology demonstration, the microsatellite autonomously navigated to a number of pre-planned navigation points around the Delta-2 second stage and showcased an autonomous operation through specially developed GNC algorithms. Subsequently, the XSS-11 mission was launched in 2005 with the purpose of demonstrating technologies for extended proximity operations with resident space objects. The technology for the XSS-11 satellite was, evidently, a further development of the technology included in the XSS-10 satellite. This further development was required because the XSS-11 mission had a planned duration of 1 year, in contrast to the 24 hours of operational lifetime of the XSS-10 mission. Furthermore, the XSS-11 satellite was designed for proximity operations of multiple resident space objects, in contrast to the single object visited by the XSS-10 satellite. During its operational lifetime, the XSS-11 satellite was designed to autonomously plan and execute rendezvous with a number of pre-selected American resident space objects within reach of its operational orbit. Upon rendezvous with the selected space

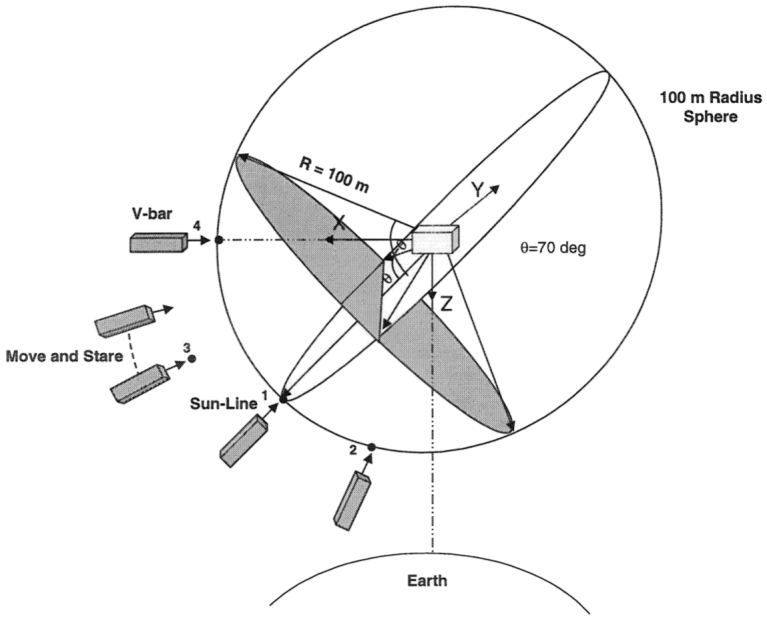


Figure 2.2: Mission design for inspection manoeuvres to be performed by the XSS-10 satellite mission as provided by Davis and Melanson (2004)

object, the spacecraft autonomously performed circumnavigation of these objects as proximity operations. However, not much is further known on the execution of this second mission, due to a lack in publications of the corresponding technical reports.

When considering the XSS-10 satellite mission, it can be determined that its sensory hardware consisted of a visible camera system and star sensor combination for attitude determination, an IMU sensor for state propagation and a mini SGLS communication system for two-way communication with ground operators (Davis and Melanson, 2004). Furthermore, the satellite consisted of four main lateral thrusters to control its position and eight reaction control thrusters to control its attitude. This sensory and actuator hardware was required to execute the pre-planned mission sequence, consisting of a separation from the Delta-2 second stage, manoeuvring to four primary inspections points, manoeuvring to one extra credit station point and finally decommissioning into a graveyard orbit. A graphical representation of the manoeuvring between the four primary inspection points is provided in Figure 2.2. From this figure it can be observed that for the first three inspection points, the XSS-10 spacecraft maintained a relative distance of approximately 100 m with respect to the Delta-2 second stage. Furthermore, at the third inspection point, it performed a move and stare manoeuvre where the satellite effectively maintained attitude with respect to its target while manoeuvring around it. Lastly, at the fourth and final inspection point, the satellite performed a V-bar manoeuvre to close in on the resident space object along a radial path, to within a distance of approximately 50 m. Hereafter, the extra credit station manoeuvre was performed and consisted of separating radially outwards from the Delta-2 second stage to a distance of 1 km and subsequently re-approaching it to within 200 m. This manoeuvre was, however, prematurely halted as ground operators could not determine whether the satellite had successfully re-acquired the resident space object with its visual camera system (Davis and Melanson, 2004). Therefore after this final failed manoeuvre, the satellite was decommissioned and moved to its graveyard orbit.

To guide the satellite through its mission phases, semi-autonomous relative state navigation was used. Note that the navigation was semi-autonomous, given that ground operators had to upload commands to initiate each new mission phase, as given by Davis and Melanson (2004). The GNC software expressed the relatives states using a Local-Vertical-Local-Horizontal reference frame centred at the Delta-II second stage and employed the Clohessy-Wiltshire equations to propagate the relative motion of both spacecraft. This propagation started from an initial vehicle position and velocity, obtained from ground operators, after which the satellite subsequently integrated the states based on measurements of the IMU. To correct for errors present in the state propagations, the GNC software updated the relative states by combining it with imagery data of the visual camera system. In case zero error would

be present, the images put the Delta-2 second stage in the centre of the field of view of its camera, whereas offsets resulted in corrections of the relative states.

2.2.3. DART

The DART mission, acronym for Demonstration for Autonomous Rendezvous Technology, was an American technology demonstrator mission with the purpose of validating fully autonomous rendezvous and proximity operations between an active chaser spacecraft and a passive target object (ESA, 2021a). The mission, launched in 2005, was designed such that the DART satellite was to rendezvous with the MUBLCOM satellite, an experimental communications microsatellite of the Defence Advanced Research Projects Agency, referred to as DARPA. During an arc of 24 hours, the DART satellite was to approach and rendezvous with the MUBLCOM satellite and subsequently perform proximity operations, such as collision avoidance manoeuvres and circumnavigation, in a fully autonomously manner. In reality, however, the DART satellite was only capable of successfully initiating the approach of the MUBLCOM satellite and during proximity operations, anomalies in the navigation system caused a soft impact between both spacecraft, resulting in a premature retreat of DART. Since the DART spacecraft was designed for autonomous operations, no corrections from ground operators could be uploaded. This lead to a premature decommissioning of the mission after only 11 hours of operations. Note that during the soft impact, which was caused by a faulty GPS receiver leading to an incorrect navigation solution, luckily none of the two spacecraft had been damaged. Therefore, the collision did not further contribute to the accumulation of fragmented space debris.

When investigating the spacecraft specifications provided by NASA (2006), it can be established that the satellite employed an Advanced Video Guidance Sensor (AVGS) as the primary navigation sensor during proximity operations. This primary sensor was further complemented with two GPS receivers onboard the DART spacecraft and one GPS receiver onboard the MUBLCOM spacecraft. During far range operations, data obtained from the GPS receivers was used to determine the relative position and velocity between both spacecraft. When in close range, between 200 and 500 m, the GPS receivers were complemented by the AVGS sensor providing bearing estimates, *i.e.*, relative azimuth and elevation angle, to determine the correct approach direction. When the relative distance was below 200 m, the AVGS sensor was switched on to further provide relative range and attitude estimates.

2.2.4. Orbital Express

Orbital Express is an American satellite mission commissioned by DARPA to provide in-orbit technology demonstration and validation for robotic servicing of satellites, using autonomous rendezvous and proximity operations (ESA, 2021e). The Orbital Express mission consisted of two satellites, ASTRO and NextSat, where ASTRO was designed to be the autonomous servicing satellite, whereas NextSat was designed as the satellite that is to be serviced by ASTRO. The most important mission objective entailed the capture and berth of the NextSat satellite by the robotic arm of ASTRO, as an auxiliary operation preceding the docking manoeuvre. Furthermore, it was planned to perform propellant transfer between both spacecraft, as well as the replacement of a battery onboard of NextSat. During the mission lifetime, the satellites were able to successfully transfer fuel and batteries between both spacecraft multiple times, as well as autonomously perform numerous rendezvous and docking manoeuvres.

Spacecraft specifications provided by Dennehy and Carpenter (2011) state that the autonomous navigation system onboard the Orbital Express consisted of an inertial navigation system and a rendezvous and capture sensor system. The purpose of the inertial system, containing a star tracker and a Space Integrated GPS/INS (SIGI) sensor, was to provide inertial attitude, position and velocity measurements. The autonomous rendezvous and capture sensor system (ARCSS) consisted of five different optical sensors with the purpose of providing relative range, bearing and attitude measurements. Three of these sensors were imaging sensors: a narrow-field-of-view (NFoV) visible acquisition and tracking sensor, Mid-to short-range side field-of-view (WFoV) visible tracking sensor and an infrared (IR) sensor to complement the above two in case of bad lighting conditions. In addition to these imaging sensors, a precision laser rangefinder (LRF) for mid-range target tracking and an advanced video guidance sensor (AVGS) laser-based tracking system were used. The purpose of implementing this multitude of sensors was to create an effective ARCSS system providing redundant coverage during approach, as can be seen in Figure 2.3.

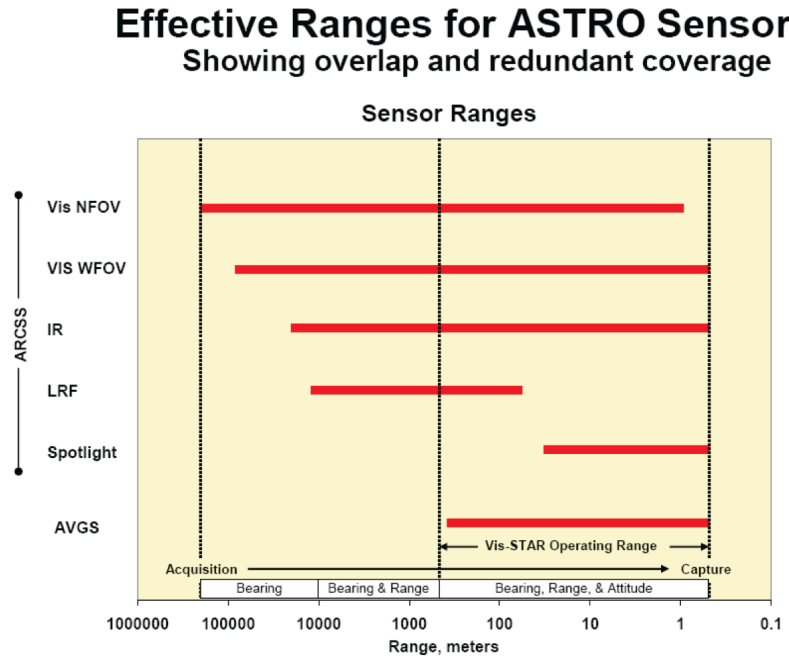


Figure 2.3: Operational ranges of the sensors constituting the autonomous rendezvous and capture sensor system of the Orbital Express mission as provided by Dennehy and Carpenter (2011)

2.2.5. ATV

The ATV missions, short for Automated Transfer Vehicle, were a set of European resupply missions to the International Space Station, of which five were launched between 2008 and 2014 (ESA, 2021c). With these vehicles being unmanned, a strong level of autonomy was required and resulted in autonomous rendezvous and docking capabilities. During the nominal missions of ATV, autonomous rendezvous and docking was performed between two cooperating spacecraft. At long range, relative navigation was provided by GPS sensors onboard the ATV and the ISS, whereas at short range, the relative navigation was provided by optical sensors. These optical sensors comprised laser rangers associated with target reflectors on the ISS for range and attitude measurements, and telegoniometers for bearing measurements. However, during the approach of the fifth and final ATV mission, an experiment was performed to demonstrate the technology required to perform a rendezvous with a non-cooperative target object (Cavrois et al., 2015). During this experiment, called LIRIS, laser and infrared imaging sensors were validated as rendezvous sensors in combination with a newly developed GNC algorithm, that was capable of providing non-cooperative rendezvous and docking. An important lesson uncovered by this experiment is the need for a GNC system design that can implement image processing from visible and infrared cameras for far-range rendezvous, and from lidar sensors for close-range rendezvous.

2.2.6. RemoveDEBRIS

RemoveDEBRIS is a European research project with the aim of providing in-orbit validation of key technologies for future active debris removal missions (ESA, 2021d). The mission objective is to demonstrate some of the key mission phases and technologies required for a real ADR mission, rather than to provide an end-to-end illustration of such a mission. The mission is comprised of four satellites, one main satellite called RemoveSAT and three auxiliary satellites, called DebrisSats 1 to 3. Once in orbit, RemoveSAT will first release DebrisSat 1 at a low relative velocity, after which it proceeds to inflate a balloon with the dual purpose of: increasing the target area allowing for easier capture by RemoveSAT, as well as acting as an drag augmentation device to enhance deorbitation. Once inflated, RemoveSAT will capture the balloon by ejecting a net as an active debris capture system, after which the DebrisSat 1 is left to deorbit. Once this has been completed, a second experiment is performed where the second DebrisSat is released at a low relative velocity and proceeds to deploy its reference panels with the same purpose as the balloon inflation for the DebrisSat 1 satellite. Once deployed, RemoveSAT will

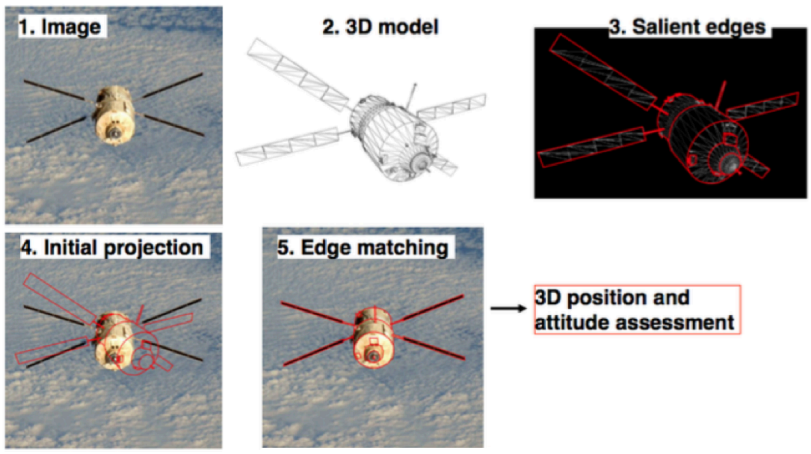


Figure 2.4: Model-based tracking approach applied to generic image and model of the ATV spacecraft as provided by Marchand et al. (2019)

capture the DebrisSat 2 by ejecting a harpoon as an active debris capture system and DebrisSat 2 is left to deorbit. The third and final experiment is aimed at validating a vision-based navigation system by ejecting the final DebrisSat 3 and using optical, infrared and lidar cameras onboard of RemoveSAT to conduct a number of manoeuvres. After these three experiments, RemoveSAT will deploy an own dragsail and is left to deorbit

Subsequent to its launch in April 2018, the mission provided successfull in-orbit demonstrations of the net and harpoon technology for active debris capture, as well as the capability of reconstructing the dynamics of a space object using the vision-based navigation system. Thereby, providing proof that the basic technology needed for ADR is viable in a real space environment.

When considering the vision-based navigation (VBN) experiment, as discussed by Marchand et al. (2019), it is given that the goal of the experiment was to localise DebrisSat 3 and determine its relative dynamics with respect to RemoveSAT. To perform this experiment, RemoveSAT was equipped with a camera operating in the visible light, in combination with a flash-imaging lidar. During the VBN experiment, a template matching approach was employed to detect the object in the images taken by the camera. This detection was used as initialisation to the subsequent tracking over the different images. During this image tracking, a model-based methodology was employed that utilises a 3D model of the target to determine its three dimensional orientation and position in the pictures. To enhance computational efficiency, the three dimensional model was simplified to the salient edges visible in the image, reducing the time required for model matching. A graphical representation of this process is provided in Figure 2.4, where it should be noted that the orientation and position of the spacecraft is obtained by minimising the error between the object in the image and the corresponding projection of the model on the image. After comparison of the experimental data with ground truth data, the VBN algorithm was capable of providing position estimates with an RMS error smaller than 1 cm, and attitude estimates with an RMS error smaller than 2 degrees. Note, however, that the VBN experiment consisted of ground-based processing of the images taken during flight. Hence a logical next step would be to perform vision-based navigation onboard a spacecraft. Nevertheless, the experiment proved the potential of this type of navigation for future ADR missions.

2.2.7. ELSA-d

The ELSA-d mission, short for End-of-Life Service by Astroscale-demonstration mission, is a first commercial demonstration mission providing in orbit validation of active debris removal technologies (Blackerby et al., 2018). The mission consists of two satellites, a chaser spacecraft, representing the servicing satellite, and the target spacecraft, representing the debris object that is to be removed from orbit (ESA, 2022). The satellite duo is joined together during launch and subsequently separates once commissioned in orbit. During mission operations, the target and chaser spacecraft repeatedly approach, capture and separate to demonstrate key technologies, such as relative navigation algorithms, capture of non-tumbling and tumbling objects and capture through use of a magnetic capture system.

To execute this mission, the sensor suite of the chaser spacecraft consists of star sensors, an IMU system, magnetometers, sun sensors and a GPS receiver, as provided by ESA (2022). This sensor suite is further extended with relative navigation sensors consisting of a visual camera system, a rendezvous laser and a rendezvous radar. Furthermore, to actuate the spacecraft during mission execution, the attitude control system consists of four reaction wheels in pyramid configuration, alongside three magnetorquers, whereas the orbit control system consists of eight chemically propelled thrusters. When considering the configuration of the target spacecraft, it must be noted that it contains a GPS receiver, as well as a three-axis attitude control system and optical markers for attitude identification. This implies that the target spacecraft is cooperative and at most pseudo-passive during proximity operations. This is also evident from the two capture phases performed during the ELSA-d mission, where the chaser uses its magnetic capture system to repeatedly capture and release the target. A non-tumbling capture is performed by commanding the target spacecraft to hold a predefined attitude, whereas during tumbling capture, the target is commanded to perform a natural tumbling motion. During non-tumbling capture, the chaser spacecraft can utilise the docking plate mounted on the target to autonomously calculate guidance commands. This is in contrast to the tumbling capture phase, where images of the tumbling target are transmitted to the ground segment and used to extract the target attitude. Subsequently, ground operators use this attitude estimate to calculate the reference trajectory for the chaser spacecraft during rendezvous and capture. Therefore, it is evident that the current mission performs some key technology demonstrations, but does not yet execute the debris removal of a true resident space object that is both non-cooperative, as well as fully passive. This therefore remains a topic of active research.

2.2.8. e.Deorbit

In 2012, ESA was set out to implement its own active debris removal mission, called e.Deorbit. The goal of this mission was to remove a large ESA-owned debris object from the critical zone in low-Earth orbit, for which the ENVISAT satellite was selected as target. This specific satellite was chosen, as it represents the most critical debris object in the low-Earth orbital region due to its significant size and mass characteristics (Wiedemann et al., 2012). To execute e.Deorbit's mission goal, ESA engineers developed the following strategy. After transfering to and phasing with ENVISAT's orbit, the chaser would rendezvous with ENVISAT, capture it and deorbit together (Biesbroek et al., (2017)). During rendezvous, the GNC subsystem of the active chaser would be required to calculate the relative position to the debris item and subsequently hop to a hold point close to the target. At this hold point, the chaser would synchronise with its target, approach it and capture it. The capture is established through a rigid connection with a robotic arm. After this, ENVISAT would be detumbled and the combined stack of target and chaser spacecraft would be steered into Earth's atmosphere for deorbitation.

Since 2012, many analyses on the e.Deorbit mission have been performed. Most noticeable for the current thesis is the study performed by Bhagat (2016), as this was focused on the development of a convex guidance algorithm for e.Deorbit. It was found that the implemented strategy, based on lossless convexification and a subsequent method of successive approximations, proved to be unsuitable due to its lack in robustness and autonomy. Moreover, a consortium of European space actors led by ESA performed a feasibility study of the e.Deorbit mission, of which some results are provided by Branco et al. (2021). Most noteworthy for the current thesis study is the developed mission scenario and approach strategy. In this strategy, the chaser spacecraft first performs station keeping on the V-bar axis of the target's Hill frame, maintaining a relative distance of 50 m. Second, the chaser steadily approaches the target over the V-bar axis to a relative separation distance of 30 m. Third, the chaser spacecraft transfers from the V-bar axis of the Hill frame to its H-bar axis, at an equal radial distance, and subsequently approaches the chaser within a distance of 6.5 m. Fourth, the chaser synchronises its attitude with the target while maintaining its previous hold position on the H-bar axis, subsequently transfers from the H-bar axis to the docking-axis of the tumbling debris item and approaches it within a distance that allows capture. This approach strategy is not only considered feasible when capturing a tumbling debris item, but also safest, given that ENVISAT is likely to be in a flat spin, *i.e.*, minimum kinetic energy for constant angular momentum (Branco et al., 2021). Nevertheless, it should be noted that, as of 2018, the e.Deorbit mission is no longer specified as an ADR mission. Rather, ESA reassigned it to an in-orbit servicing mission that is to be developed further.

Summary

All missions, as discussed above, illustrated the technological possibilities for rendezvousing, approaching and docking of two satellites in orbit. A summary table containing relevant design characteristics of these missions is provided in Table 2.2. Note that the discussion above had the purpose of providing an insight into the heritage of missions containing rendezvous and docking capabilities, but does not provide an exhaustive list containing all flown missions. Furthermore note that these missions are all concerned with rendezvousing two satellites that are cooperative to a certain extent. Therefore, the logical subsequent step for an active debris removal system is to design a mission capable of capturing a non-cooperative object in space. A potential first mission to undertake such an endeavour is the ClearSpace-1 mission, which will serve as the reference for the current thesis study.

The knowledge provided by the mission heritage can be applied to the GNC system design of the ClearSpace-1 satellite. It can be established from the ETS-VII mission that low impact velocity will be required to ensure safe and autonomous rendezvous and docking. This is required as it simplifies collision avoidance manoeuvres during contingency operations and can easily be applied by enforcing a maximum approach velocity constraint. Furthermore, the need for a vision-based navigation system has been established by the ETS-VII, ATV and RemoveDEBRIS missions. During vision-based navigation, the chaser is required to estimate its relative position, velocity, attitude and angular rate with respect to the target using image processing technology. To this extent, ATV has validated the use of image processing from visible and infrared cameras for far-range rendezvous and from lidar sensors for close-range rendezvous (Cavrois et al., 2015). Moreover, nearly all rendezvousing missions implemented a rendezvous radar as part of their sensor suite. Hence it can be established that the sensory hardware onboard the ClearSpace-1 satellite will have to contain, as a minimum, a radar, a visual camera and a lidar. Moreover, based on the vision-based navigation experiment performed by the RemoveDEBRIS mission, it can be established that a realistic target RMS error can be smaller than 1 cm for position estimates, and smaller than 2 degrees for attitude estimates (Marchand et al., 2019). Additionally, it can be concluded from the ELSA-d mission that a ground-calculated reference trajectory can be flown during cooperative capture of a tumbling debris-item. However, for the ClearSpace-1 mission, an increased level of autonomy is required due to safety concerns (Peters et al., 2013). This, therefore, further motivates the study of this mission and its corresponding technological challenges, as no endeavour similar to this has previously been performed. This is because, in contrast to all other ADR missions that have already been flown, ClearSpace-1 will deorbit an actual resident space object rather than a dummy spacecraft carried onboard during launch.

From the study on the e.Deorbit mission performed by Bhagat (2016), it was shown that the implemented convex guidance system based on a method of successive approximations will be unsuitable in its presented form. This is due to its lack in robustness and autonomy. Therefore, it is required to investigate different convexification strategies for the ClearSpace-1 mission. Moreover, due to the beneficial safety characteristics of the approach strategy provided by Branco et al. (2021), it is decided to further implement this as baseline strategy for the ClearSpace-1 mission.

2.3. Reference Mission

The research performed in the current thesis study is centred around the ClearSpace-1 mission, established by the Swiss start-up company ClearSpace SA and funded by ESA. The mission was selected out of more than 12 contenders to provide a potential first ever space mission that actively removes a debris item from orbit. As a result, ClearSpace-1 is set to launch in 2025 as an ADR mission with the objective of deorbiting the upper part of the Vega Secondary Payload Adapter, called VESPA. This debris object is chosen as the target for the first ADR mission, as it provides a relatively simple and sturdy geometry before continuing with more complex and fragile geometries. With a mass of 112 kg, the adapter provides a target that closely resembles a small satellite in size and mass. The adapter was part of the second flight of the Vega rocket, launched in 2013.

After separation from its payload, VESPA was left in an 801 km by 664 km disposal orbit. Therefore, for the chaser spacecraft, an initial parking orbit is envisaged as a circular orbit at 500 km altitude. From this orbit, the satellite will subsequently transfer and actively approach, capture and deorbit the uncooperative VESPA adapter. Once the VESPA adapter has been captured using four robotic tentacles onboard the ClearSpace-1 satellite, the combined pair will stabilise its rotation and deorbit together to burn up in the atmosphere. An illustration of this planned mission scenario is provided in Figure 2.5.

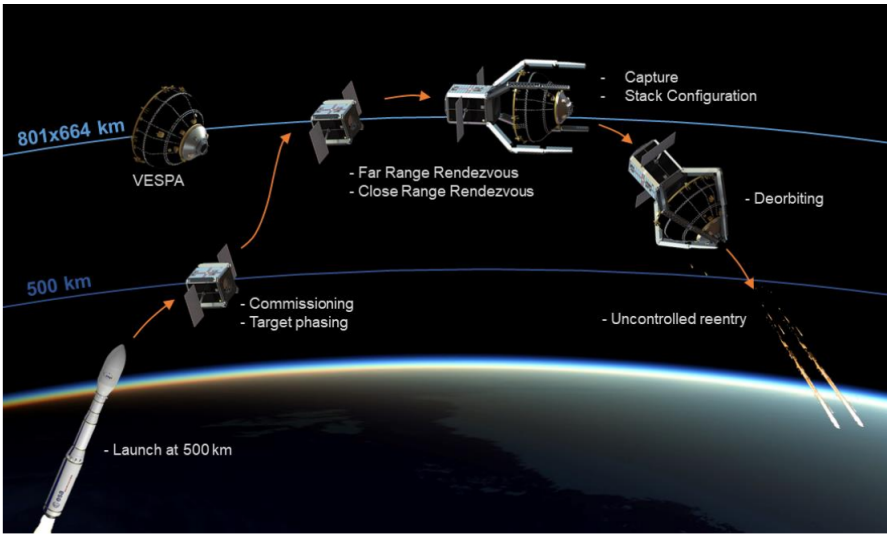


Figure 2.5: Mission scenario of ClearSpace-1 satellite to approach, capture and deorbit the VESPA launch vehicle adapter Source: ClearSpace SA https://elecnor-deimos.com/clearspace/ [Cited 17 June 2021]

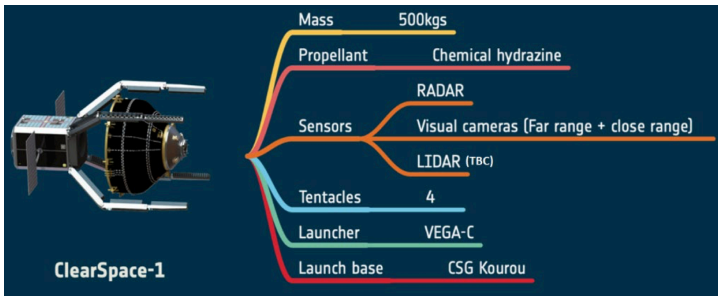


Figure 2.6: Overview of characteristics of the ClearSpace-1 satellite as provided by Biesbroek et al. (2021)

It should be noted that the ClearSpace-1 mission is initiated by ESA as part of a larger framework, which extends further into in-orbit servicing missions. This is because ADR and in-orbit servicing are based on essentially the same key technologies, such as cutting-edge GNC systems paired with vision-based navigation. These technologies are necessary to allow the chaser satellite to close in on its target in a safe and autonomous manner. Therefore it has been decided to center the present thesis study around the design of a GNC system that can successfully be implemented for the ClearSpace-1 mission. To this extent, the considered reference mission is further detailed here, where Subsection 2.3.1 provides a description on the ClearSpace-1 chaser satellite, whereas Subsection 2.3.2 provides a description on the VESPA target satellite.

2.3.1. ClearSpace-1 Satellite

The active chaser spacecraft that will rendezvous with the target is called the ClearSpace-1 satellite. This 500 kg medium-sized spacecraft will be equipped with four robotic arms that clamp around the target to allow for a hard capture. Furthermore, the spacecraft will be equipped with sensors for visual navigation: visual cameras for both far and close range navigation, a radar and a lidar (Biesbroek et al., 2021). An overview of the characteristics of this chaser spacecraft is provided in Figure 2.6. Note that not many details are currently available, as the satellite is still under development by ClearSpace SA.

2.3.2. VESPA Adapter

The non-cooperative target is represented by the upper part of the Vega Secondary Payload Adapter. This system has been designed to increase the versatility of the Vega launcher, by allowing to launch a main passenger in its upper position simultaneously with a one or more secondary passengers in

Table 2.1: Orbital elements of the VESPA adapter as obtained from NORAD two-line element tracking on 24th of May 2021
Source: CelesTrak https://celestrak.com/satcat/search-results.php [cited 24 May 2021]

Norad ID	39162
Orbital Parameter	**Parameter value**
Inclination [deg]	98.7928
RAAN [deg]	17.6301
Eccentricity [-]	0.0094276
Argument of Periapsis [deg]	224.9993
Mean Anomaly [deg]	134.3533
Mean Motion [revs/day]	14.48362939
Epoch [JD]	2459358.58946
Revolution number at epoch [revs]	42503

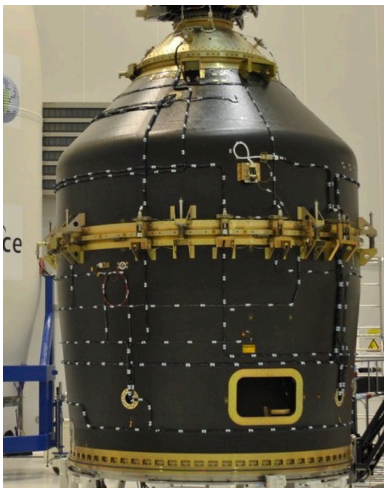


Figure 2.7: VESPA configuration as provided by Biesbroek et al. (2021)

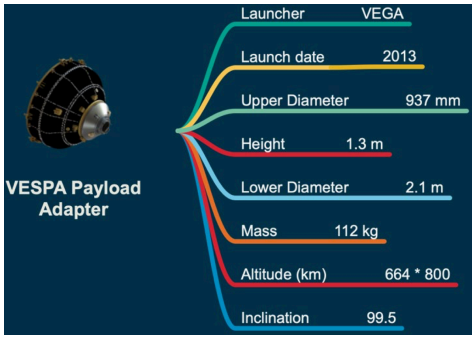


Figure 2.8: Overview of characteristics of the VESPA upper part as provided by Biesbroek et al. (2021)

its lower cavity. VESPA was first launched onboard the second Vega rocket in 2013, and the payload onboard this flight consisted of the main PROBA-V satellite of ESA, together with two additional smaller satellites, being the Vietnamese VNREDSat-1 and the Estonian ESTCube-1. After launch and separation, VESPA has been left in an approximately 801 km by 664 km disposal orbit with orbital parameters as summarised in Table 2.1. An illustration of the VESPA configuration is provided in Figure 2.7, where it should be noted that the target of the ClearSpace-1 mission is limited to the upper part of the VESPA adapter, *i.e.*, the conical section above the golden ring. From this figure it can further be seen that there are a number of bright coloured segments on the VESPA upper section which will aid in its identification by a vision-based navigation system (Biesbroek et al., 2021). An overview of the characteristics of the upper part of the VESPA adapter is provided in Figure 2.8.

2.4. Mission and System Requirements

Combining the reference mission with the insights obtained from the discussion on mission heritage, the following mission requirements can be established. These requirements define the boundaries of the current research and provide insight into the problem that is ought to be solved. Note that henceforth, the ClearSpace-1 satellite will be referred to as the chaser spacecraft, whereas the VESPA adapter is referred to as the target spacecraft. Furthermore, mission requirements are designated by **REQ_M**, system requirements of the chaser by **REQ_S_C** and system requirements of the target by **REQ_S_T**.

Mission Requirements

REQ_M_1 The chaser shall move in an 801 by 664 km near-circular orbit.

REQ_M_2 The target shall move in an 801 by 664 km near-circular orbit.

REQ_M_3 The reference mission shall cover the terminal approach phase of the target, up to but not including the capture phase.

REQ_M_4 The reference mission shall start with the chaser located at a hold point 20 m behind the target.

REQ_M_4.1 This hold point shall be located on the negative V-bar of the target's Hill frame.
REQ_M_4.2 The chaser shall be located at this hold point with zero relative velocity.
REQ_M_4.3 The target shall be assumed to have undetermined initial attitude.

REQ_M_5 The chaser shall synchronise its attitude with the uncooperative target before final approach.

REQ_M_6 The chaser shall actively approach the uncooperative target.

REQ_M_6.1 The chaser shall approach the uncooperative target within an approach cone with a half angle of at least 30 degrees.
REQ_M_6.2 The centreline of the approach cone shall be 5 m long.
REQ_M_6.3 The centreline of the approach cone shall be directed along the instantaneous docking-axis of the tumbling target.
REQ_M_6.4 The instantaneous docking-axis of the tumbling target shall coincide with the negative X-axis of the target's Body-Centred-Body-Fixed reference frame.
REQ_M_6.5 The chaser shall approach the target with a maximum approach velocity of 1 m/s to ensure low impact docking.
REQ_M_6.6 The chaser shall remain outside a keep-out sphere with radius of 1.35 m centred at the target.

REQ_M_7 The reference mission shall end with the chaser located at a hold point 1.35 m behind the target.

REQ_M_7.1 This hold point shall be located on the instantaneous docking-axis of the tumbling target.
REQ_M_7.2 This hold point shall represent the relative geometry that will be tracked in a subsequent capture phase.

REQ_M_8 The reference mission shall be launched by 2025.

System Requirements
The above mission requirements correspond to the following system requirements for the chaser.

REQ_S_C_1 The chaser shall be considered controllable over its six degrees-of-freedom.

REQ_S_C_2 The chaser shall weigh 500 kg.

REQ_S_C_2.1 The mass properties of the chaser shall be assumed known throughout the terminal approach phase.
REQ_S_C_2.2 The mass properties of the chaser shall be assumed constant throughout the terminal approach phase.

REQ_S_C_3 The inertia tensor of the chaser shall equal $\begin{bmatrix} 40.849 & 0 & 0 \\ 0 & 70.861 & 0 \\ 0 & 0 & 70.861 \end{bmatrix}$ kgm^2 throughout the terminal approach phase.

REQ_S_C_3.1 The inertia properties of the chaser shall be assumed known throughout the terminal approach phase.
REQ_S_C_3.2 The inertia properties of the chaser shall be assumed constant throughout the terminal approach phase.

REQ_S_C_4 The centre of mass location of the chaser shall be on the longitudinal symmetry axis, 0.55 m from its base.

REQ_S_C_4.1 The centre of mass location of the chaser shall be assumed known throughout the terminal approach phase.

REQ_S_C_4.2 The centre of mass location of the chaser shall be assumed constant throughout the terminal approach phase.

REQ_S_C_5 The chaser shall be represented by a rigid parallelepiped body with dimensions of 1.1 m $\times$ 2.5 m $\times$ 0.7 m.

REQ_S_C_6 The chaser shall have a drag coefficient of 2.25.

REQ_S_C_7 The chaser shall have a specular reflection coefficient of 0.32.

REQ_S_C_8 The chaser shall have a diffuse reflection coefficient of 0.36.

REQ_S_C_9 The chaser shall estimate its position, velocity, attitude and angular rate inertially.

REQ_S_C_10 The chaser shall estimate the relative position, velocity, attitude and angular rate of the target using vision-based navigation.

REQ_S_C_11 The inertial navigation sensors onboard the chaser shall contain as a minimum: an IMU, a GPS receiver and two star sensors.

REQ_S_C_12 The relative navigation sensors onboard the chaser shall contain as a minimum: a radar, a far range visual camera, a close range visual camera and a lidar.

REQ_S_C_13 The GNC actuators onboard the chaser shall contain as a minimum: three reaction wheels and 24 reaction control thrusters.

REQ_S_C_13.1 The reaction control thruster system shall be used as only orbit control system during the terminal approach phase.
REQ_S_C_13.2 The maximum thrust a reaction control thruster shall provide equals 55 N.
REQ_S_C_13.3 The reaction wheel system shall be used as only attitude control system during the terminal approach phase.
REQ_S_C_13.4 The maximum torque a reaction control wheel shall provide equals 50 Nm.

REQ_S_C_14 The GNC algorithm onboard the chaser shall provide real-time onboard optimisation of the terminal rendezvous phase.

REQ_S_C_15 The GNC algorithm onboard the chaser shall execute the terminal rendezvous phase autonomously.

REQ_S_C_16 The navigation accuracy shall equal 1% of the relative distance, velocity, attitude and angular velocity to the target.

REQ_S_C_16.1 The final navigation accuracy shall equal 2 cm in position.
REQ_S_C_16.2 The final navigation accuracy shall equal 2 mm/s in velocity.
REQ_S_C_16.3 The final navigation accuracy shall equal 2 deg in attitude.
REQ_S_C_16.4 The final navigation accuracy shall equal 0.2 deg/s in angular velocity.

REQ_S_C_17 The chaser shall be equipped with four robotic tentacles to capture the target.

Furthermore, these mission requirements accumulate to the following system requirements for the target.

REQ_S_T_1 The target shall be considered uncooperative and passive.

REQ_S_T_2 The target shall weigh 112 kg.

REQ_S_T_2.1 The mass properties of the target shall be assumed known throughout the terminal approach phase.
REQ_S_T_2.2 The mass properties of the target shall be assumed constant throughout the terminal approach phase.

REQ_S_T_3 The inertia tensor of the target shall equal $\begin{bmatrix} 47.505 & 0 & 0 \\ 0 & 36.342 & 0 \\ 0 & 0 & 36.342 \end{bmatrix}$ kgm^2 throughout the terminal approach phase.

REQ_S_T_3.1 The inertia properties of the target shall be assumed known throughout the terminal approach phase.
REQ_S_T_3.2 The inertia properties of the target shall be assumed constant throughout the terminal approach phase.

REQ_S_T_4 The centre of mass location of the target shall be on the longitudinal symmetry axis, 0.5 m from its base.

REQ_S_T_4.1 The centre of mass location of the target shall be assumed known throughout the terminal approach phase.
REQ_S_T_4.1 The centre of mass location of the target shall be assumed constant throughout the terminal approach phase.

REQ_S_T_5 The target shall be represented by a rigid conical body with a lower diameter of 2.1 m, an upper diameter of 0.937 m and a height of 1.3 m.

REQ_S_T_6 The target shall have a drag coefficient of 2.25.

REQ_S_T_7 The target shall have a specular reflection coefficient of 0.32.

REQ_S_T_8 The target shall have a diffuse reflection coefficient of 0.36.

REQ_S_T_9 The target shall be tumbling while traversing its orbit.

REQ_S_T_9.1 The target shall spin at a nominal rate of 3.5 deg/s over all three body axes.
REQ_S_T_9.2 The target shall spin at a worst-case rate of 5 deg/s over its X and Y-axis.

2.5. Scope of Thesis Study

To conclude this chapter on the research context, an outline of the project scope is provided. As previously established, the current thesis study provides a GNC system design for an active debris removal mission. In this respect, the prime focus is put on the guidance subsystem, in terms of a convex guidance system, followed by a focus on the control system, in terms of an attitude synchronisation controller, and a final focus on the navigation system, in terms of the inertial navigation system of the chaser. Nevertheless, the goal of the project is to provide a full system integration for the considered mission scenario. Therefore, to achieve this integrated system design within the time constraints of the current project, the following restrictions are applied.

Mission Scenario

- The mission scenario does not cover the launch, commissioning, target phasing and far-range rendezvous phases of the reference mission.
- The mission scenario does not cover the capture, detumbling, deorbit and reentry phases of the reference mission.
- The mission scenario does not cover the identification phase of the mass and inertia properties of the target.
- The mission scenario does not cover the identification phase of the mass and inertia properties of the chaser.
- The mission scenario does not cover the modelling of plume impingement, as generated by the reaction control thrusters of the chaser, on the translational and rotational dynamics of the target.
- The mission scenario does not cover the interaction between the reaction control thrusters and the reaction wheel system for momentum dumping.

Spacecraft

- The target maintains constant mass and inertia properties throughout the implemented mission scenario.
- The chaser maintains constant mass and inertia properties throughout the implemented mission scenario.

Guidance System

- The guidance system does not implement collision avoidance manoeuvres.
- The guidance system does not command rotational motion of chaser spacecraft.

Control System

- No formal optimisation of control system gains is provided.

- The control system separates attitude control from orbit control.
- The attitude control system does not perform in-orbit calibration of reaction wheel system.
- The reaction wheel system does not contain scaling or misalignment errors.
- The reaction wheel system omits time delays between control commands and actuator outputs.
- The reaction wheel system output measurements are continuously available.
- The reaction wheel system output measurements do not contain any time delay.
- The orbit control system does not perform in-orbit calibration of reaction control thruster system
- The reaction control thruster system implements ideal thrusters.
- The reaction control thruster system omits time delays between control commands and actuator outputs.
- The reaction control thruster system provides omnidirectional thrust.

Navigation System

- The navigation system separates inertial state estimation from relative state estimation.
- The inertial navigation measurements do not contain any time delay.
- The inertial navigation measurements are continuously available, *i.e.*, no angular velocity limitations for star sensors, always at least 4 satellites in view of GPS receiver and no signal blockage by target.
- The inertial navigation system does not perform in-orbit calibration of its sensors.
- The relative navigation system is artificially implemented.
- The relative navigation measurements do not contain any time delay.
- The relative navigation measurements are continuously available, *i.e.*, no consideration of lighting conditions for visual sensors.
- The relative navigation system does not perform in-orbit calibration of its sensors.

System Analysis

- The system analysis does not cover verification procedures of the basic GRADS library, only extensions made to it are covered.
- The system analysis does not cover lighting conditions checks, nor communication conditions checks for the implemented the mission scenario.
- The system analysis does not cover variations to the mass and inertia properties of chaser and target spacecraft.
- The system analysis does not cover variations to the error levels of the navigation sensors and control actuators.
- The system analysis does not cover the simultaneous variation of more than one error source at a time.

Note that, from these restrictions, it is evident that different levels of focus are put on the different subsystems of the GNC system, where less focused systems are characterised by a more restricted design space.

Table 2.2: Summary table containing rendezvous characteristics of mission heritage

Mission / Program	Target Orbit	Target Altitude [km]	Target Type	Degree of Autonomy	Sensors	Guidance Law	References
ETS-VII	Circular	550	Cooperative and pseudo-passive	Autonomous	Rendezvous radar, GPS receiver, Camera-type proximity sensor, Accelerometer	CW equations for V-bar approach during terminal phase	Kawano et al. (1999)
XSS-10	Circular	800	Cooperative and passive	Semi-autonomous with phase initiations commanded by ground operators	IMU, Star sensor, Visible camera	CW equations during terminal phase	Davis and Melanson (2004)
DART	Near-Circular	723 by 730	Cooperative and passive	Autonomous	GPS receiver, AVGS sensor	Information not provided	NASA (2006)
Orbital Express	Circular	492	Cooperative and passive	Autonomous	Star sensor, SIGI sensor, Vis NFOV, VIS WFOV and IR cameras LRF and AVGS sensors	Information not provided	Dennehy and Carpenter (2011)
ATV	Circular	408	Cooperative and active	Semi-autonomous with phase initiations commanded by ground operators	GPS receivers, Laser rangers, Telegoniometers	CW equations during terminal phase	Cavrois et al. (2015)
RemoveDEBRIS	Circular	405	Psuedo-uncooperative and passive	Autonomous	Visible camera, Flash-imaging lidar	Information not provided	Marchand et al. (2019)
ELSA-d	Circular	550	Cooperative and pseudo-passive	Reference trajectory calculated by ground operators	Star sensor, IMU, Magnetometer, Sun sensor, GPS receiver, Visual camera system, Rendezvous laser, Rendezvous radar	Information not provided	ESA (2022)
e.Deorbit	Circular	800	Uncooperative and passive	Autonomous	Star sensor, Sun sensor, IMU, Lidar GPS receiver, Far field and near field cameras	Convex Guidance	Bhagat (2016)

3

Astrodynamics and Space Environment

This chapter details the orbital mechanics and space environment that will be instituted in the motion simulator developed for the current thesis study. The reference frames used to define the motion of the spacecraft are discussed in Section 3.1 and the transformations required to convert between different reference frames are provided in Section 3.2. After this, the different state representations used by the simulator are defined in Section 3.3 and the specifics of the motion simulator are detailed in Section 3.4. Lastly, the verification procedures of the self-developed environmental models implemented in this simulator are provided in Section 3.5.

3.1. Reference Frames

A reference frame represents a coordinate system or set of axes within which the properties of an object can be defined. Therefore, to describe the relative motion of both satellites in orbit, at least one frame is required. In case one wants to apply Newtonian mechanics to obtain the equations of motion, the reference frame needs to be an inertial one. However, it is often more convenient to derive certain terms of the equations of motion in a different reference frame before transforming it to the inertial frame. Furthermore, certain quantities are more conveniently expressed in a frame different to the inertial frame. Therefore, the current section will be concerned with defining the relevant reference frames, denoted by $\mathcal{F}$, for the mission scenario provided in Chapter 2. Note that for all presented reference frames, a right-handed coordinate system is utilised such that the right-hand rule is satisfied.

3.1.1. Earth-Centred Inertial Frame

The inertial reference frame, with respect to which the relevant equations of motion will be derived, is given by the Earth-Centred inertial frame, $\mathcal{F}_\mathrm{I}$. A graphical representation of this reference frame is provided in Figure 3.1, where it can be seen that the inertial frame is centred at Earth but not fixed to Earth, *i.e.*, it translates but does not rotate along with Earth. The $\mathcal{F}_\mathrm{I}$ frame is defined by a Z-axis pointing along Earth's rotation axis towards the north pole, an X-axis laying within Earth's equatorial plane and pointing towards a fixed point in inertial space, and a Y-axis positioned in Earth's equatorial plane to complement the right-handed coordinate system. For the current research, the general definition of the Earth's Mean Equator and Equinox frame, commonly referred to as the J2000 or EME2000 frame, is employed, which positions this fixed point at the mean vernal equinox established on the 1st of January 2000 at 12:00 Terrestrial Time.

3.1.2. Local-Vertical-Local-Horizontal Frame

To conveniently express the relative position of a chaser spacecraft with respect to its target, the Local-Vertical-Local-Horizontal frame, $\mathcal{F}_\mathrm{L}$, is defined. A graphical depiction of this frame is provided in Figure 3.2, where a body-fixed frame is presented that positions its origin at the centre of mass of the target spacecraft. Furthermore, its X-axis points along the radial direction, its Y-axis points along the in-flight direction and its Z-axis complements the right-handed frame by pointing perpendicular to the orbital plane. However, when using this frame to describe the relative motion between two spacecraft, it is also commonly referred to as the Hill frame, $\mathcal{F}_\mathrm{H}$, where its Z-axis is defined as the H-bar direction, its Y-axis as the V-bar direction and its X-axis as the R-bar direction. Nevertheless, both frames are identical and their names will be used interchangeably throughout the subsequent discussions.

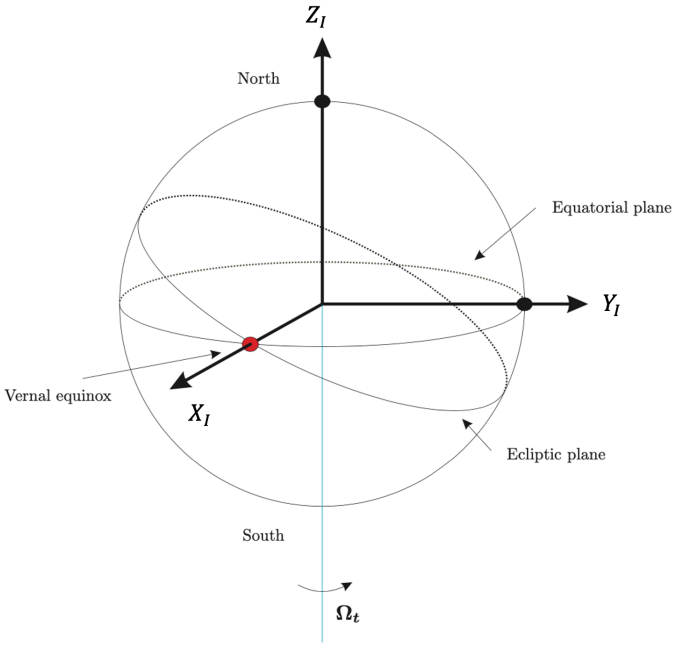


Figure 3.1: J2000/EME2000 Earth-Centred inertial reference frame $\mathcal{F}_I$, adapted from Mulder et al. (2013). Note that in this figure Ω_t represents Earth's angular velocity

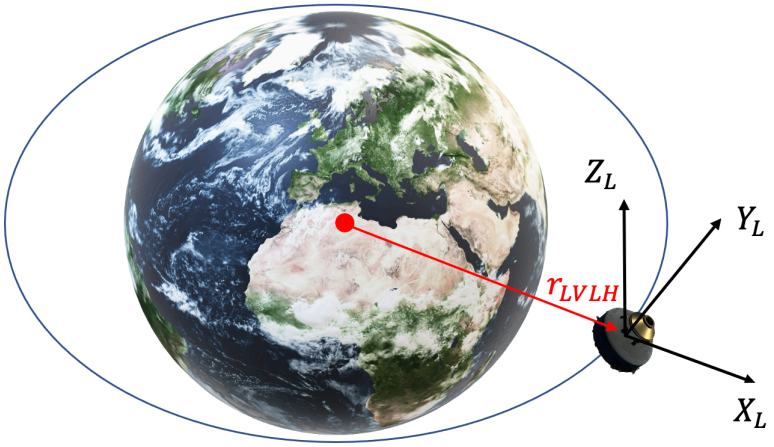


Figure 3.2: Local-Vertical-Local-Horizontal frame $\mathcal{F}_L$ or Hill frame $\mathcal{F}_H$. Note that in this figure r_{LVLH} represents the vector from the centre of mass of the Earth to the centre of mass of the target spacecraft.
Source: Adapted from Microsoft [cited 21 July 2021]

3.1.3. Body-Centred-Body-Fixed Frame

The Body-Centred-Body-Fixed reference frame, $\mathcal{F}_B$, is used to provide the attitude of an object. The origin of the frame is located at the centre of mass of the spacecraft body, with its Z-axis pointing along the nadir vector towards the centre of Earth, its X-axis pointing forward along the plane of symmetry of the spacecraft body and its Y-axis orthogonal to both X- and Z-axes to complement the right-handed coordinate frame. Note that the exact orientation of the X-axis along the plane of symmetry is still undetermined and its definition depends on the vehicle itself. When considering the body-fixed frame of the ClearSpace-1 satellite, given by $\mathcal{F}_{B,C}$ in Figure 3.3, the X-axis is defined along the intersection of the horizontal and vertical symmetry plane and points forward. This also implies that the Y-axis is positioned within the horizontal symmetry plane and points towards the right. When considering the body-fixed frame of the VESPA target, given by $\mathcal{F}_{B,T}$ in Figure 3.3, the X-axis is defined as the symmetry axis of this body of revolution and the Y-axis complements the right handed frame by pointing towards the right.

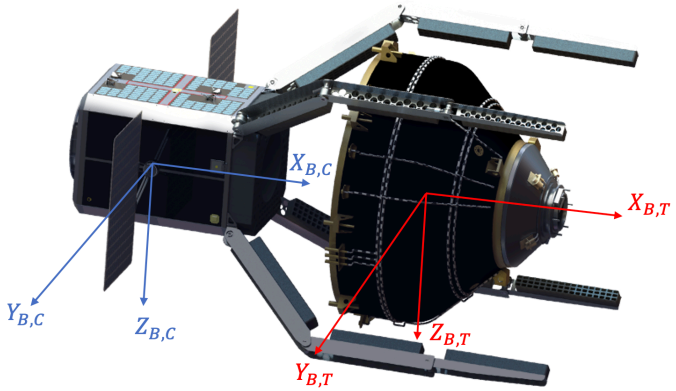


Figure 3.3: Body-Centred-Body-Fixed reference frames $\mathcal{F}_{B,C}$ and $\mathcal{F}_{B,T}$
Source: Adapted from ESA https://www.air-cosmos.com/article/premire-mission-de-nettoyage-orbital-23925 [cited 15 June 2021]

3.2. Frame Transformations

The previously derived reference frames are used to express the necessary properties of the spacecraft in a convenient manner. However, for the governing equations of motion, it is necessary that all these properties are expressed with respect to an inertial reference frame. This is obtained when applying a frame transformation from the current frame, in which the property is defined, to the inertial frame. Furthermore, once the motion of the spacecraft is obtained in the inertial frame, the frame transformations can be used to re-transform it to other reference frames, such as for example the Local-Vertical-Local-Horizontal frame.

The following two subsections will discuss the frame transformations that are applied throughout the current thesis study, where Subsection 3.2.1 discusses the conversion between the inertial frame and the Local-Vertical-Local-Horizontal frame and Subsection 3.2.2 the conversion between the inertial frame and the Body-Centred-Body-Fixed frames.

3.2.1. $\mathcal{F}_I$ and $\mathcal{F}_L$ Frames

The Local-Vertical-Local-Horizontal frame is used to describe the translational motion of the chaser spacecraft with respect to its target. Therefore, the current discussion will describe how to transform the translational states of both spacecraft from the inertial frame, as obtained from the motion simulator, to a relative state in the Local-Vertical-Local-Horizontal frame, as used during rendezvous and docking studies.

First, to be able to transform the Cartesian position vector of the chaser, the transformation matrix from the Earth-Centred inertial to the Local-Vertical-Local-Horizontal frame is defined. A characterisation of this matrix can be obtained from three unit vectors: $\hat{\mathbf{e}}_R$, $\hat{\mathbf{e}}_N$ and $\hat{\mathbf{e}}_T$, which are defined in Equations (3.1) to (3.3). Note for these equations that $\mathbf{r}_T^{\mathcal{F}_I}$ and $\mathbf{V}_T^{\mathcal{F}_I}$ represents the Cartesian position and velocity vectors of the target spacecraft and are expressed in the inertial $\mathcal{F}_I$ frame.

$$\hat{\mathbf{e}}_R = \frac{\mathbf{r}_T^{\mathcal{F}_I}}{||\mathbf{r}_T^{\mathcal{F}_I}||} \tag{3.1}$$

$$\hat{\mathbf{e}}_N = \frac{\mathbf{r}_T^{\mathcal{F}_I} \times \mathbf{V}_T^{\mathcal{F}_I}}{||\mathbf{r}_T^{\mathcal{F}_I} \times \mathbf{V}_T^{\mathcal{F}_I}||} \tag{3.2}$$

$$\hat{\mathbf{e}}_T = \frac{\hat{\mathbf{e}}_R \times \hat{\mathbf{e}}_N}{||\hat{\mathbf{e}}_R \times \hat{\mathbf{e}}_N||} \tag{3.3}$$

Using these unit vector definitions, the transformation matrix from the $\mathcal{F}_I$ frame to the $\mathcal{F}_L$ frame can be described, and is provided in Equation (3.4). Furthermore, with this transformation matrix, the relative Hill position of the chaser spacecraft with respect to its target can be calculated using Equation (3.5). Note that in this equation, $\mathbf{r}_C^{\mathcal{F}_L}$ and $\mathbf{r}_C^{\mathcal{F}_I}$ represent the Cartesian position vectors of the chaser

spacecraft, as expressed in the Local-Vertical-Local-Horizontal $\mathcal{F}_L$ frame and the inertial $\mathcal{F}_I$ frame, respectively.

$$\mathbf{C_{L/I}} = \begin{bmatrix} \hat{\mathbf{e}}_R & \hat{\mathbf{e}}_T & \hat{\mathbf{e}}_N \end{bmatrix}^T \tag{3.4}$$

$$\mathbf{r}_C^{\mathcal{F}_L} = \mathbf{C_{L/I}} \cdot \left(\mathbf{r}_C^{\mathcal{F}_I} - \mathbf{r}_T^{\mathcal{F}_I} \right) \tag{3.5}$$

Second, to transform the Cartesian velocity vector of the chaser spacecraft from its inertial representation to its relative Hill representation, a derivative of the previously provided transformation matrix is required. To characterise this second transformation matrix, the derivatives of the previously defined unit vectors are employed, as given by Equations (3.6) to (3.8).

$$\dot{\hat{\mathbf{e}}}_R = \frac{1}{\|\mathbf{r}_T^{\mathcal{F}_I}\|} \left(\mathbf{V}_T^{\mathcal{F}_I} - \left(\hat{\mathbf{e}}_R \cdot \mathbf{V}_T^{\mathcal{F}_I} \right) \hat{\mathbf{e}}_R \right) \tag{3.6}$$

$$\dot{\hat{\mathbf{e}}}_N = 0 \tag{3.7}$$

$$\dot{\hat{\mathbf{e}}}_T = \dot{\hat{\mathbf{e}}}_R \times \hat{\mathbf{e}}_N \tag{3.8}$$

This results in the derivative of the transformation matrix from the Earth-Centred inertial frame to the Local-Vertical-Local-Horizontal frame, as given in Equation (3.9). Furthermore, with this transformation matrix, the relative Hill velocity of the chaser spacecraft with respect to its target can be calculated using Equation (3.10). Note that in this equation, $\mathbf{V}_C^{\mathcal{F}_L}$ and $\mathbf{V}_C^{\mathcal{F}_I}$ represent the Cartesian velocity vectors of the chaser spacecraft, as expressed in the Local-Vertical-Local-Horizontal $\mathcal{F}_L$ frame and the inertial $\mathcal{F}_I$ frame, respectively.

$$\dot{\mathbf{C}}_{\mathbf{L/I}} = \begin{bmatrix} \dot{\hat{\mathbf{e}}}_R & \dot{\hat{\mathbf{e}}}_T & \dot{\hat{\mathbf{e}}}_N \end{bmatrix}^T \tag{3.9}$$

$$\mathbf{V}_C^{\mathcal{F}_L} = \dot{\mathbf{C}}_{\mathbf{L/I}} \cdot \left(\mathbf{r}_C^{\mathcal{F}_I} - \mathbf{r}_T^{\mathcal{F}_I} \right) + \mathbf{C_{L/I}} \cdot \left(\mathbf{V}_C^{\mathcal{F}_I} - \mathbf{V}_T^{\mathcal{F}_I} \right) \tag{3.10}$$

An inverse transformation of the translational Hill state to an inertial position and velocity vector for the chaser is currently not applied in the thesis study. Therefore, this transformation is currently omitted. Nevertheless, the thrust commands calculated by the guidance system will be transformed from the Hill frame to the inertial frame. However, to centralise the discussion, this transformation will be provided together with the guidance system in Chapter 4.

3.2.2. $\mathcal{F}_I$ and $\mathcal{F}_B$ Frames

The transformation from the Earth-Centred inertial frame to the Body-Centred-Body-Fixed frame is characterised by a yaw angle Ψ, a pitch angle θ and a roll angle φ of the $\mathcal{F}_B$ frame with respect to the $\mathcal{F}_I$ frame. During this transformation from the $\mathcal{F}_I$ frame to the $\mathcal{F}_B$ frame, one first rotates a yaw angle Ψ over the Z_I-axis, subsequently one rotates a pitch angle θ over the $Y_{I'}$-axis and finally one rotates a roll angle φ over the $X_{I''}$-axis. A mathematical description of this transformation matrix is provided in Equation (3.11), where it should be noted that c() and s() are used to represent the cosine and sine function in a compact manner. Note that this transformation matrix can be defined for rotations to both the chaser-fixed frame and target-fixed frame, by using the Ψ, θ and φ angles that correspond with the $\mathcal{F}_{B,C}$ and $\mathcal{F}_{B,T}$ frames, respectively.

$$\mathbf{C_{B/I}} = \mathbf{C_x}(\varphi)\mathbf{C_y}(\theta)\mathbf{C_z}(\Psi) = \begin{bmatrix} c\theta c\Psi & c\theta s\Psi & -s\theta \\ -c\varphi s\Psi + s\varphi s\theta c\Psi & c\varphi c\Psi + s\varphi s\theta s\Psi & s\varphi c\theta \\ s\varphi s\Psi + c\varphi s\theta c\Psi & -s\varphi c\Psi + c\varphi s\theta s\Psi & c\varphi c\theta \end{bmatrix} \tag{3.11}$$

The corresponding inverse transformation, *i.e.*, from the Body-Centred-Body-Fixed frame to the Earth-centred inertial frame, is provided by Equation (3.12). Note that this inverse relation can be established by taking the transpose of the transformation matrix provided in Equation (3.11).

$$\mathbf{C_{I/B}} = \mathbf{C}_{\mathbf{B/I}}^T \tag{3.12}$$

3.3. State Representations

A state representation provides the mathematical description used to describe the properties of a dynamical system. For the rendezvous and capture of a tumbling debris object in space, it is evident that a six degrees of freedom simulation of both the target and chaser spacecraft will be required. This is because both the position and attitude of the chaser with respect to its target have to be controlled to ensure successfull mission execution. In addition to this, it is assumed throughout the current thesis that both spacecraft consist of rigid bodies, such that body deformations can safely be neglected as part of the state, as given by requirements **REQ_S_T_5** and **REQ_S_C_5**. Moreover, the masses of both can be assumed constant, as given by **REQ_S_T_2** and **REQ_S_C_2**. Therefore, terms related to mass dynamics can be omitted.

A discussion on the state representations that are used in the current thesis study is provided in the following subsections, where Subsection 3.3.1 describes the translational state representation and Subsection 3.3.2 the rotational state representation.

3.3.1. Translational State

For the translational state, a Cartesian representation is provided, as it results in a straightforward formulation of the equations of motion and is directly obtainable from the Cowell propagator used in the motion simulator. The translational state of a vehicle can be expressed in Cartesian form by decomposing its position and velocity vectors along the reference axes of a Cartesian frame. A mathematical description of the translation state of a vehicle described in Cartesian elements, $\mathbf{x}_{\text{trans}}$, is provided in Equation (3.13), where x, y and z represent the Cartesian position components, whereas $\dot{x}$, $\dot{y}$ and $\dot{z}$ represent the Cartesian velocity components. Note that the Cartesian state representation can be used to provide either the absolute state of the target and chaser spacecraft, *e.g.*, in the $\mathcal{F}_I$ frame, or the relative state between the target and chaser spacecraft, *e.g.*, in the $\mathcal{F}_L$ frame. To calculate the relative translation state between two vehicles, a simple elemental subtraction of the cartesian state vectors can be performed. Note, however, that this subtraction is only valid when both state vectors are referenced with respect to the same frame of reference.

$$\mathbf{x}_{\text{trans}} = \left(x, y, z, \dot{x}, \dot{y}, \dot{z}\right)^T \tag{3.13}$$

3.3.2. Rotational State

For the rotational state, two alternative state representations were considered. Initially, the Modified Rodrigues Parameter (MRP) representation was selected for the motion simulator, as discussed in Section 3.4. However, practical considerations related to the attitude controller gave rise to changing this MRP representation to the well-known quaternion representation. Further details are omitted here and the interested reader is referred to Appendix B for more information related to this attitude change.

Quaternions represent a four dimensional hyper-complex number consisting of three imaginary numbers and one real number (Kuipers, 2002). These four parameters represent a single rotation around the Euler eigenaxis of an object. The first three elements of the quaternion vector, q_1, q_2 and q_3, provide the orientation of the Euler eigenaxis, $\hat{\mathbf{e}}$, and are therefore commonly referred to as the vectorial part of the quaternion. This is in contrast to the last element of the quaternion vector, q_4, which provides the rotation angle over this Euler eigenaxis, Φ, and is referred to as the scalar part of the quaternion. A mathematical definition of the quaternion vector, $\mathbf{q}$, in terms of the Euler eigenaxis and the corresponding rotation angle is provided in Equation (3.14). Note that in this equation, $\hat{e}_1$, $\hat{e}_2$ and $\hat{e}_3$ represent the three components of the Euler eigenaxis vector, $\hat{\mathbf{e}}$.

$$\mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = \begin{pmatrix} \hat{e}_1 \sin(\Phi/2) \\ \hat{e}_2 \sin(\Phi/2) \\ \hat{e}_3 \sin(\Phi/2) \\ \cos(\Phi/2) \end{pmatrix} \tag{3.14}$$

In case quaternions are employed to represent the attitude of a vehicle, it is required to impose a unit length constraint, as the system otherwise becomes overdetermined with four parameters defining three degrees of freedom. This means that the parameters of the quaternion vector are not independent of each other, as can be seen in Equation (3.15). Furthermore, this implies that after each calculation

step, a normalisation has to be applied to remove numerical inaccuracies that violate the unit length constraint. Nevertheless, as the quaternion representation is not minimal, no singularity is present.

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \tag{3.15}$$

A mathematical description of the rotational state of a vehicle described using attitude quaternions, $\mathbf{x}_{rot}$, is provided in Equation (3.16). Note that, to complement the quaternion representation with an angular velocity parameterisation, Equation (3.16) defines an angular velocity vector, $\boldsymbol{\omega}$, that is decomposed along the three reference axes of the vehicle's body frame, as given by ω_1, ω_2 and ω_3.

$$\mathbf{x}_{rot} = \left(q_1, q_2, q_3, q_4, \omega_1, \omega_2, \omega_3\right)^T \tag{3.16}$$

Note that the provided rotational state representation can be used to represent either the absolute state of the target and chaser spacecraft, or the relative state between the target and chaser spacecraft. To calculate this relative state, an elemental subtraction of the angular velocities can be performed, given that both are expressed in the same frame of reference. This is, however, in contrast to the attitude error in terms of quaternions, which has to be calculated using a so-called error quaternion, $\delta\mathbf{q}$. This error quaternion, defined by the quaternion product of the second attitude and the inverse quaternion of the first attitude, provides the relative rotation from the first body to the second body. A mathematical description of this quaternion product can be found in Equation (3.17), with $\mathbf{q}_2$ representing the second attitude quaternion, whereas $\overline{\mathbf{q}}_1$ represents the inverse quaternion of the first attitude. Note that in this equation, q_{2_i} and $\overline{q}_{1_i}$, with i equal to 1, 2, 3 or 4, represent the quaternion elements of $\mathbf{q}_2$ and $\overline{\mathbf{q}}_1$, respectively.

$$\delta\mathbf{q} = \mathbf{q}_2 \otimes \overline{\mathbf{q}}_1 = \begin{bmatrix} q_{2_4} & q_{2_3} & -q_{2_2} & q_{2_1} \\ -q_{2_3} & q_{2_4} & q_{2_1} & q_{2_2} \\ q_{2_2} & -q_{2_1} & q_{2_4} & q_{2_3} \\ -q_{2_1} & -q_{2_2} & -q_{2_3} & q_{2_4} \end{bmatrix} \begin{pmatrix} \overline{q}_{1_1} \\ \overline{q}_{1_2} \\ \overline{q}_{1_3} \\ \overline{q}_{1_4} \end{pmatrix} \tag{3.17}$$

Moreover, Equation (3.18) describes how to calculate the inverse quaternion of the first attitude, $\overline{\mathbf{q}}_1$, from its nominal attitude quaternion $\mathbf{q}_1$, where q_{1_1}, q_{1_2}, q_{1_3} and q_{1_4} represent the quaternion components of $\mathbf{q}_1$.

$$\overline{\mathbf{q}}_1 = \begin{pmatrix} -q_{1_1} \\ -q_{1_2} \\ -q_{1_3} \\ q_{1_4} \end{pmatrix} \tag{3.18}$$

3.4. Motion Simulator

The motion simulator is used to model the dynamics of the target and chaser spacecraft under the influence of the space environment, as if it were the motion of both spacecraft during the actual mission. The output of this simulator will be used as input to the navigation system to simulate the measurements of the sensors. Note that, as given above for the state representation in Section 3.3, the simulated states of both spacecraft will consists of components related to their translational motion, as well as components related to their rotational motion. In Subsection 3.4.1, the equations of motion that are implemented in the motion simulator are discussed. Subsequently, the setup of the numerical simulations that are used to assess the relevance of the disturbance models for the considered reference mission is provided in Subsection 3.4.2. Lastly, with the setup of the simulation defined, the relevant accelerations models can be selected in Subsection 3.4.3, whereas Subsection 3.4.4 selects the implemented torque models.

3.4.1. Equations of Motion

The equations of translational motion are most conveniently expressed using Cartesian coordinates. This implies that the translational state vector consists of three Cartesian position components and three Cartesian velocity components. The general equations of translational motion are provided in Equation (3.19). Note that this equation represents a vector equation with **a** representing the vector of

Cartesian acceleration components on the body. Moreover, $\mathbf{F}_{net}$ represents the vector containing the three net external force components acting on the body and m represents its instantaneous mass. A discussion on the forces that make up $\mathbf{F}_{net}$ will, however, be provided in Subsection 3.4.3.

$$\mathbf{a} = \frac{\mathbf{F}_{net}}{m} \tag{3.19}$$

The kinematic equation for translational motion is given in Equation (3.20), where $\dot{\mathbf{x}}$ represents the time derivative of the position vector, whereas $\mathbf{V}$ represents the velocity vector.

$$\dot{\mathbf{x}} = \mathbf{V} \tag{3.20}$$

When considering the equations of rotational motion, it must be noted that the motion simulator initially implemented an attitude representation in terms of Modified Rodrigues Parameter. This characterisation was chosen, as it provided a minimum attitude parameterisation that, when combined with a corresponding shadow set, is capable of avoiding singularities for all rotation angles over the Euler eigenaxis. In addition to this, the MRP attitude set was chosen as it provides computationally efficient attitude kinematics for spacecraft applications, for example as provided by Tsiotras (1996) and Schaub and Junkins (2003). However, it was found from preliminary analyses on the INDI controller, which was tasked with performing attitude synchronisation with the target spacecraft, an undesirable control input was calculated when using the MRP attitude parameterisation. This is caused by the switching nature of the MRP and SMRP sets, generating a discontinuous reference profile. As a consequence of this discontinuous reference profile, the attitude controller can result in a shortly oscillating output at the switch to or from the MRP sets. This is undesirable behaviour for the durability of the control actuators and a nonphysical phenomenon that is caused by the particular attitude representation, rather than by the underlying attitude dynamics of the spacecraft. Therefore, to enhance controllability over the chaser, it was decided to change the attitude representation to attitude quaternions.

The general equations of rotational motion that result from this change in attitude parameterisation are provided in Equation (3.21). Note that this equation constitutes a vector equation, with $\dot{\boldsymbol{\omega}}$ representing the angular acceleration vector of the body. Moreover, $\mathbf{J}$ represents the inertia tensor of the body, $\boldsymbol{\omega}$ the angular velocity vector and $\mathbf{M}_{net}$ the vector containing the net external moment acting on the body. A discussion on the moments that make up $\mathbf{M}_{net}$ will, however, be provided Subsection 3.4.4.

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}\left(\mathbf{M}_{net} - \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}\right) \tag{3.21}$$

The kinematic equation for rotational motion is given in Equation (3.22), where $\dot{\mathbf{q}}$ represents the time derivative of the quaternion vector, $\boldsymbol{\omega}$ the angular velocity vector and $\mathbf{Q}$ an auxiliary matrix defined by Equation (3.23).

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{Q}\boldsymbol{\omega} \tag{3.22}$$

$$\mathbf{Q} = \begin{bmatrix} q_4 & -q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix} \tag{3.23}$$

3.4.2. Numerical Simulation

With the generic form of the equations of motion established, it is required to define the force models that constitute $\mathbf{F}_{net}$ in Equation (3.19) and the torque models that constitute $\mathbf{M}_{net}$ in Equation (3.21). To this end, numerical simulations of the VESPA target and the ClearSpace-1 chaser are performed. The relevant simulation parameters are summarised in Table 3.1. Using these parameters, which have been established either from the list of system requirements in Section 2.4 or from analogy with the design values of the e.Deorbit mission, the orbital dynamics are simulated a total of ten times. Initially, a Keplerian orbit is simulated as baseline solution using a central gravity attraction from Earth. Subsequently, the acceleration and torque generated by Earth's spherical harmonics is simulated using a gravity fields of degree and order two up to degree and order five. To assess the perturbing accelerations of third bodies, the central gravity field of Earth is simulated in combination with either a point mass

attraction from the Sun or a point mass attraction from the Moon. Lastly, the effect of surface forces on the accelerations and moments of both spacecraft is modelled in terms of atmospheric drag and solar radiation pressure, as well as modelling the disturbance moment generated by Earth's magnetic field.

Table 3.1: Simulation parameters used during acceleration and torque model analysis of motion simulator

General Parameters			
epoch		24/05/2025 00:00:00.0	
simulation time [hr]		6	
VESPA Parameters			
initial position [m]	$(-4.975540, -0.782642, -4.920253)^T \cdot 10^6$	upper radius [m]	0.4685
initial velocity [m/s]	$(4.8467, 2.3976, -5.2825)^T \cdot 10^3$	height [m]	1.3
initial attitude quaternion [-]	$(0.3542, -0.6909, -0.5991, 0.1958)^T$	drag coefficient [-]	2.25
initial angular velocity [deg/s]	$(0, 0, 0)^T$	specular reflection coefficient [-]	0.32
mass [kg]	112	diffuse reflection coefficient [-]	0.36
inertia tensor [kgm^2]	$\begin{bmatrix} 47.505 & 0 & 0 \\ 0 & 36.342 & 0 \\ 0 & 0 & 36.342 \end{bmatrix}$	residual magnetic dipole [Am^2]	$(0, 0, 0)^T$
lower radius [m]	1.05		
ClearSpace-1 Parameters			
initial position [m]	$(-4.975528, -0.782636, -4.920267)^T \cdot 10^6$	width [m]	2.5
initial velocity [m/s]	$(4.8467, 2.3976, -5.2825)^T \cdot 10^3$	height [m]	0.7
initial attitude quaternion [-]	$(-0.6074, -0.3696, -0.1963, 0.6752)^T$	drag coefficient [-]	2.25
initial angular velocity [rad/s]	$(0, 0, 0)^T$	specular reflection coefficient [-]	0.32
mass [kg]	500	diffuse reflection coefficient [-]	0.36
inertia tensor [kgm^2]	$\begin{bmatrix} 40.849 & 0 & 0 \\ 0 & 70.861 & 0 \\ 0 & 0 & 70.861 \end{bmatrix}$	residual magnetic dipole [Am^2]	$(0.1, 0.1, 0.1)^T$
length [m]	1.1		

3.4.3. Force Models

A comprehensive summary of the acceleration models is provided in Figure 3.5. In this figure, the relative position and velocity errors generated by the acceleration models is compared to the final navigation accuracy requirements, as given in **REQ_S_C_16.1** and **REQ_S_C_16.2**. Note, however, that these relative errors pertain to the position and velocity errors between the target and chaser spacecraft during the rendezvous scenario. This means that the errors represent the difference between the relative state of the chaser spacecraft in the Hill frame of the target for the baseline Keplerian simulation, and the relative state of the chaser spacecraft in the Hill frame of the target for the simulations with the various acceleration models. Furthermore note that, in Figure 3.5, the relative errors caused by the spherical gravity fields are obtained through comparison with a gravity field of one degree and order less, with the purpose of isolating the effect of the added spherical harmonic terms. For example, the relative errors of spherical harmonics degree and order three are obtained by subtracting these results from the relative position and velocity errors obtained when modelling a spherical harmonics gravity field of degree and order two. All other errors are obtained by comparing with the relative quantities from the baseline Kepler orbit, as explained above.

From Figure 3.5 it can be seen that the relevant acceleration models are:

- Gravity acceleration due to a spherical harmonic Earth with degree and order 2
- Atmospheric drag acceleration
- Solar radiation pressure acceleration

3.4.4. Moment Models

A comprehensive summary of the torque models is provided in Figure 3.6. In this figure, the relative attitude and angular velocity errors generated by the torque models are compared to the final navigation accuracy requirements, as given in **REQ_S_C_16.3** and **REQ_S_C_16.4**. Note again that these relative errors pertain to the attitude and angular velocity errors between the target and chaser spacecraft during the rendezvous scenario. This means that the errors represent the difference between the relative rotational state of the chaser spacecraft with respect to the target spacecraft for the baseline

Keplerian simulation, and relative rotational state of the chaser spacecraft with respect to the target spacecraft for the simulations with the various torque models. Furthermore note again that the relative errors of the spherical gravity field are obtained by comparing with a gravity field of one degree and order less.

From Figure 3.6 it can be seen that the relevant torque models are:

- Gravity torque due to a spherical harmonic Earth with degree and order 2
- Atmospheric drag torque
- Solar radiation pressure torque
- Magnetic dipole torque

As a check of the numerically simulated results, the selection of relevant torque and acceleration models is explained from analysing the underlying physics. Both satellites start in an equal orbit with a relative separation distance of 20 meter, as given by the mission scenario in **REQ_M_4**. This means that the magnitude of the disturbance torques and accelerations that only depend on the distance to the object, such as gravitational accelerations and torques generated by the primary body or by third bodies, are practically identical. However, given that the mass distribution, and thus the inertia of both spacecraft are dissimilar, these forces will have a different influence on their dynamics. Consequently, relative state errors are generated. Nevertheless, for these influences to be relevant, the underlying torques and accelerations need to have sufficient magnitude. Only a spherical harmonic gravity field of degree and order 2 possesses sufficient influence over the spacecraft dynamics to accumulate a relative state error that exceeds the provided navigation accuracies in the performed simulations. Furthermore note that, as the most prominent third bodies in low-Earth orbit do not provide sufficient influence on the relative errors between both spacecraft, also other third bodies in the Solar System can safely be omitted from the analysis. When considering the pressure forces and corresponding moments, it is evident that due to the large discrepancy in size, shape and mass of the two satellites, relative errors in the translational and rotational state will be generated that exceed the navigation accuracy requirement. Therefore, it is clear that the disturbance torques and accelerations generated by the solar radiation pressure and the atmospheric drag have to be included in the motion simulator. Lastly, considering the magnetic dipole torque, it is evident that a relative error in the rotational state of both spacecraft will be generated, as the defunct VESPA target does not contain a residual dipole moment and thus does not generate a magnetic dipole torque, whereas the operational ClearSpace-1 chaser does. By generating this additional disturbance torque on the chaser spacecraft, the relative rotational error accumulates to beyond the navigation accuracy requirements, implying the inclusion of the magnetic dipole moment in the motion simulator. Hence from this discussion it can be concluded that the selection of relevant disturbance models for the motion simulator makes complete sense.

Nevertheless, it should be noted that, in addition to these disturbance models, the acceleration model of the chaser spacecraft is extended to include its thrusting capabilities. This is required as the active chaser has an onboard propulsive system to manipulate its translational motion, whereas the passive target does not and therefore cannot provide a thrust acceleration. Furthermore, also the torque model of the chaser spacecraft is extended to implement its control moment, as the active chaser has onboard control actuators to manipulate its attitude, whereas the passive target does not.

3.5. Verification

The verification procedure of the astrodynamics and space environment is limited to assessing the functionality of the self-developed disturbance models. This is because the basic frame transformations, state conversions and motion simulator employed in the current thesis study have already been verified as part of the GRADS library by Mooij (2021b). Additional reference frames, frame transformations, state representations and state conversions that have been developed for the current thesis study have been verified through straightforward unit-testing. However, for brevity purposes, these results are currently omitted. Nevertheless, the results that are provided in Chapter 8 could not have been obtained without correct implementation of these complementary models, providing indirect proof of their functionality and justifying the omission of their verification procedures. The following discussion therefore only provides the verification process of the solar radiation pressure model in Subsection 3.5.1

Table 3.2: Simulation parameters used during solar radiation pressure model acceleration analysis

General Parameters		**Parallelepiped Parameters**	
epoch	23/09/2001 00:00:00.0	length [m]	1
simulation time [sec]	6027.14	height [m]	1
initial position [m]	$(7.158137, 0, 0)^T \cdot 10^6$	width [m]	1
initial velocity [m/s]	$(0, 7.4622, 0)^T \cdot 10^3$	**Cone Parameters**	
initial attitude quaternion [-]	$(0, 0, 0, 1)^T$	lower radius [m]	0.6
initial angular velocity [rad/s]	$(0, 0, 0)^T$	upper radius [m]	0.517
mass [kg]	100	height [m]	1
inertia tensor [kgm^2]	$\begin{bmatrix} 80 & 0 & 0 \\ 0 & 82 & 0 \\ 0 & 0 & 4 \end{bmatrix}$		
specular reflection coefficient [-]	0.32		
diffuse reflection coefficient [-]	0.36		

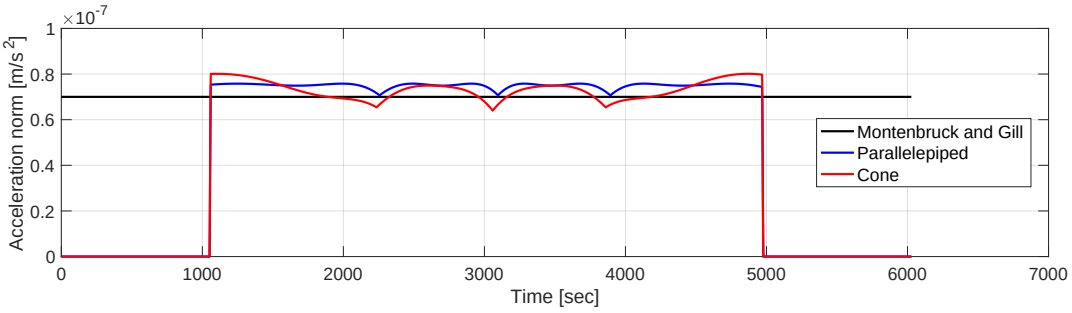


Figure 3.4: Time series of solar radiation pressure acceleration norm and reference magnitude as provided by Montenbruck and Gill (2000)

and of the atmospheric drag model in Subsection 3.5.2. The interested reader is referred to Appendix A for more details on these disturbance models.

3.5.1. Solar Radiation Pressure Model

A correct implementation of the solar radiation pressure disturbance model can be assessed from a two-step verification process. First, the sign and magnitude of the solar radiation pressure force are compared with literature provided by Montenbruck and Gill (2000) and Schöttle (2000), after which the sign and magnitude of the solar radiation pressure moment are compared with Larson and Wertz (1999). As this perturbation model is relatively complex, and given the lack of representative examples, it becomes difficult to compare the simulation results with predefined benchmarks. Therefore, it should be noted that the current verification is not exhaustive and solely meant to enhance the confidence in a correct implementation.

A summary of the simulation parameters used for the force model testing, unless specifically stated otherwise, are provided in Table 3.2. Note that the simulation epoch in Table 3.2 is chosen purposely as September 23 2001, as it is known that at this instant, the Sun was positioned approximately on the negative X-axis of the inertial $\mathcal{F}_I$ frame. This implies that the solar radiation pressure forces are therefore acting along the positive X-axis of the inertial $\mathcal{F}_I$ frame, allowing to verify the correct direction of the solar radiation pressure disturbance models.

The resulting magnitude of the solar radiation pressure acceleration on a conical and a parallelepiped body can be seen in Figure 3.4. Note that this figure also provides the estimate of the solar radiation pressure acceleration as determined by Montenbruck and Gill (2000), for a spacecraft with a representative area-to-mass ratio of 0.01 m^2/kg. Given that only a constant is provided, it should solely be used to assess the magnitude of the solar radiation pressure accelerations. For a verification of the correct direction, the acceleration components provided in Figure 3.7 can be used.

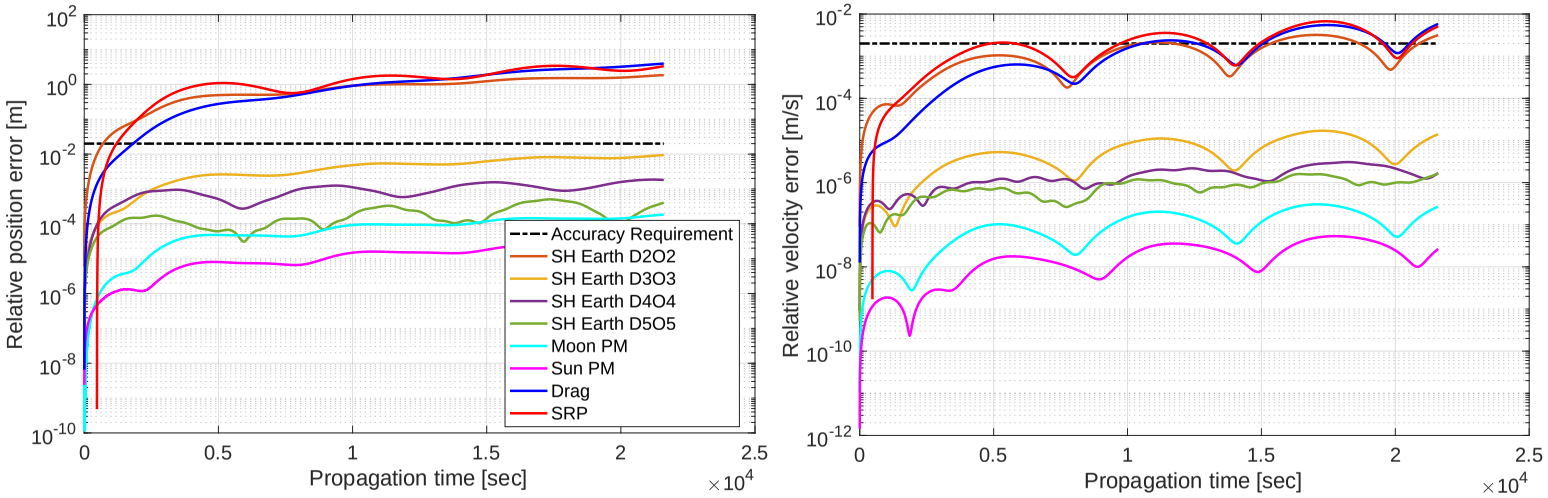


Figure 3.5: Time series of relative position and velocity errors of perturbing acceleration models, obtained using fixed-step ode3 integrator in Simulink environment with a time step of 1 second

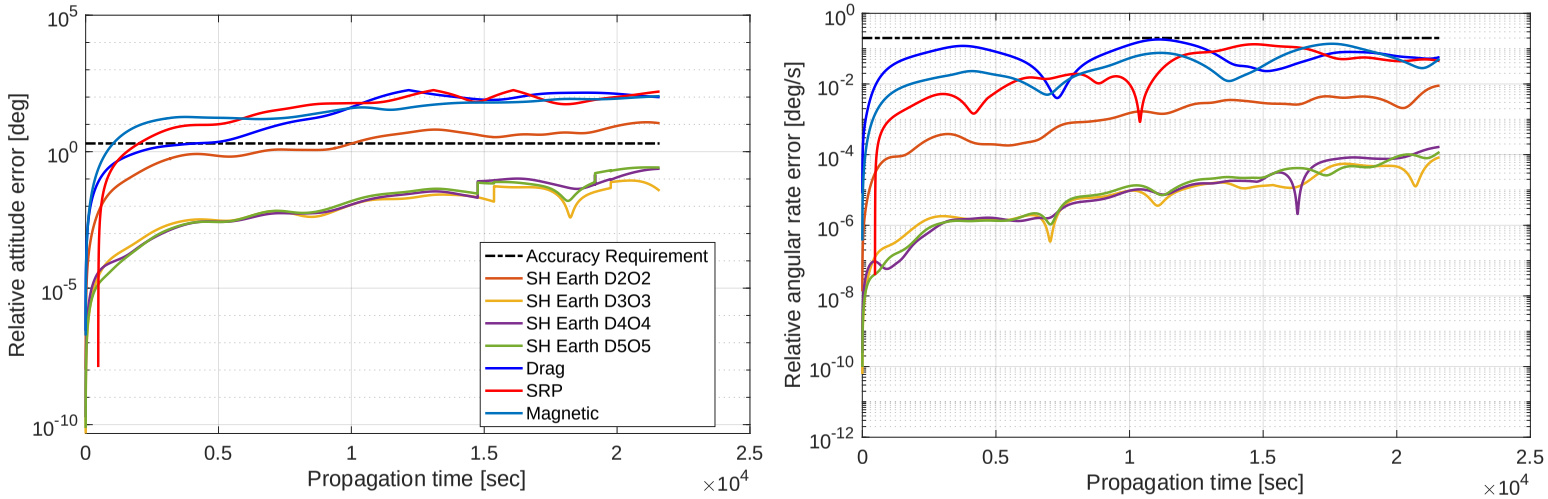


Figure 3.6: Time series of relative attitude and angular velocity errors of disturbance torque models, obtained using fixed-step ode3 integrator in Simulink environment with a time step of 1 second

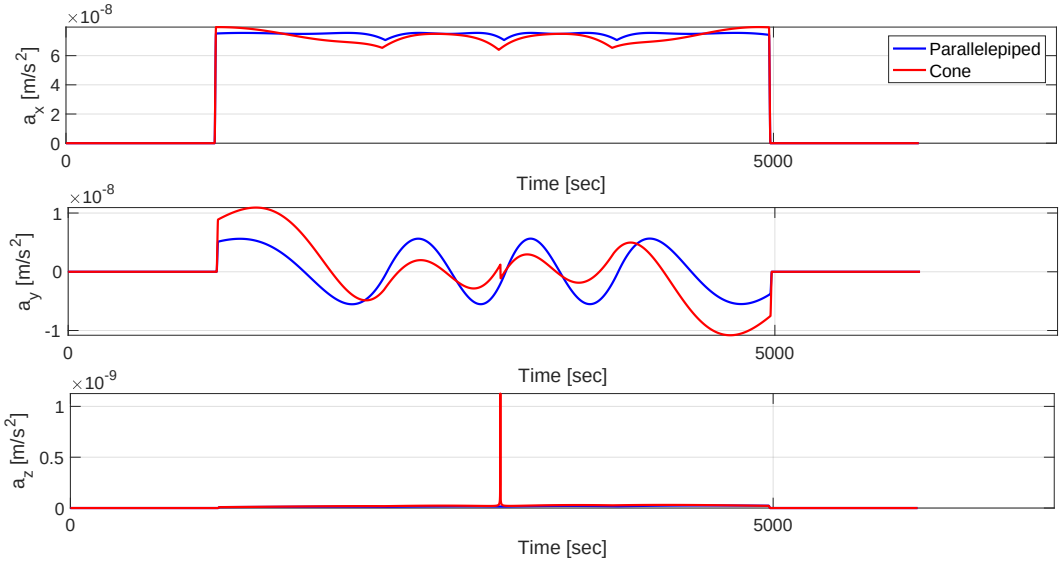


Figure 3.7: Time series of solar radiation pressure acceleration components

When inspecting Figures 3.4 and 3.7, it can be observed that, at the start and the end of the simulation, the solar radiation pressure accelerations are zero. This is caused by the fact that the spacecraft is in eclipse during those portions of the orbit, such that no acceleration can be generated. Furthermore, it can be seen from Figure 3.4 that for both the conical and parallelepiped bodies, the acceleration magnitude corresponds well with the $7 \cdot 10^{-8}$ m/s^2 reference value provided by Montenbruck and Gill (2000). Moreover, from Figure 3.7 it can be seen that for both the conical and parallelepiped bodies, the main acceleration component acts along the positive X-axis of the inertial $\mathcal{F}_I$ frame, as expected from the position of the Sun at the simulation epoch.

Additionally, it is interesting to assess the effect of the solar radiation disturbance on the eccentricity of a geostationary orbit. An analytical estimate of this change in eccentricity is provided by Schöttle (2000) and given below in Equation (3.24). Note that in this equation, e(t) represents the change in eccentricity of a geostationary orbit as a function of time, ε the reflectivity of the surface material, $\frac{A}{m}$ the area-to-mass ratio of the spacecraft and $\dot{\Lambda}_E$ the angular velocity of the Earth around the Sun.

$$e(t) = 0.022(1+\varepsilon)\frac{A}{m}\left|\sin\left(\frac{1}{2}\dot{\Lambda}_E t\right)\right| \tag{3.24}$$

When simulating geostationary spacecraft with conical and parallelepiped bodies, the eccentricity changes provided in Table 3.4 result. These eccentricity changes are obtained by simulating the two spacecraft using the simulation parameters established in Table 3.3. Moreover, Table 3.4 also contains the corresponding analytical changes in eccentricity, which are calculated from Equation 3.24. When comparing the numerical and analytical results, it can be identified that both values correspond well. Hence from this discussion, it can be concluded that the models of the solar radiation pressure force on a conical and parallelepiped body are implemented correctly.

When considering the solar radiation pressure moment on a parallelepiped body, it should be noted that the implemented model locates the centre of mass at the geometric centre of the body. This implies that, given that the forces also act on the geometric centre of their respective panel, the body will not generate a perturbing moment. This is because the symmetry of the problem makes the components of the different faces cancel each other out, resulting in zero perturbing moment. This can mathematically be shown from the moment equations provided in Appendix A, by adding the respective moment components of each of the illuminated faces. A comparison of the analytically and numerically calculated solar radiation pressure moment both resulted in zero moment generation. Hence it can be concluded that both the orientation and magnitude of the solar radiation pressure moment on a parallelepiped body are implemented correctly.

Table 3.3: Simulation parameters used during analysis of eccentricity perturbation caused by solar radiation pressure

General Parameters		**Parallelepiped Parameters**	
epoch	$23/09/200100:00:00.0$	length [m]	1
simulation time [sec]	86400	height [m]	1
initial position [m]	$(42.164148, 0, 0)^T \cdot 10^6$	width [m]	1
initial velocity [m/s]	$(0, 3.0747, 0)^T \cdot 10^3$	**Cone Parameters**	
initial attitude quaternion [-]	$(0, 0, 0, 1)^T$	lower radius [m]	0.6
initial angular velocity [rad/s]	$(0, 0, 0)^T$	upper radius [m]	0.517
mass [kg]	100	height [m]	1
inertia tensor [kgm^2]	$\begin{bmatrix} 80 & 0 & 0 \\ 0 & 82 & 0 \\ 0 & 0 & 4 \end{bmatrix}$	**Analytical Parameters**	
specular reflection coefficient [-]	0.32	ε [-]	0.55
diffuse reflection coefficient [-]	0.36	$\dot{\Lambda}_E$ [rad/s]	$\frac{\pi}{365}$
		$\frac{A}{m}$ [m^2/kg]	0.01

Table 3.4: Change in eccentricity after one orbital period in geostationary orbit due to solar radiation pressure disturbance

Eccentricity Change	**Analytical**	**Numerical**
Parallelepiped Body [-]	$2.9350 \cdot 10^{-6}$	$3.0308 \cdot 10^{-6}$
Conical Body [-]	$2.9350 \cdot 10^{-6}$	$2.9834 \cdot 10^{-6}$

Table 3.5: Simulation parameters used during solar radiation pressure moment analysis

General Parameters		**Parallelepiped Parameters**	
epoch	$23/09/200100:00:00.0$	length [m]	1
simulation time [sec]	6027.14	height [m]	1
initial position [m]	$(-7.1781, 0, 0)^T \cdot 10^6$	width [m]	1
initial velocity [m/s]	$(0, -7.4518, 0)^T \cdot 10^3$	**Cone Parameters**	
initial angular velocity [rad/s]	$(0, 0, 0)^T$	lower radius [m]	1.05
mass [kg]	100	upper radius [m]	0.4685
inertia tensor [kgm^2]	$\begin{bmatrix} 80 & 0 & 0 \\ 0 & 82 & 0 \\ 0 & 0 & 4 \end{bmatrix}$	height [m]	1.3
specular reflection coefficient [-]	0.32		
diffuse reflection coefficient [-]	0.36		

When considering the solar radiation pressure moment on a conical body, it should be noted that the implemented model also locates the centre of mass at the geometric centre. However, for this body, the generated moment components will generally not cancel each other out, resulting in a solar radiation pressure moment generation. To analyse the orientation of the generated solar radiation pressure moments, the general simulation parameters as provided in Table 3.5 are employed together with the initial orientations as provided by Table 3.6, given that these locate the Sun on top of the $\pm$X, $\pm$Y and $\pm$Z-axes of the body, respectively. Note that, to ensure the Sun position is exactly on the body axes of the conical spacecraft, its location is artificially fixed to be on the negative X-axis of the inertial $\mathcal{F}_I$ frame. This is because the chosen epoch only positions the Sun approximately on this axis, and a higher accuracy is required for the provided verification tests. Furthermore, for brevity purposes, only the details of two test cases are discussed, as the remaining test cases provide similar results and are therefore provided in Appendix A.

When the Sun is located on the positive X-axis of the body, both the upper flat surface and the slanted curved surface are illuminated. However, the flat surface will not generate a solar radiation pressure moment given that the line of action of its resulting force acts through the centre of mass of the spacecraft. Furthermore, the slanted side of the body will also not generate a resulting moment

Table 3.6: Initial body orientations used during solar radiation pressure moment analysis

Sun position	Initial quaternion [-]	Sun position	Initial quaternion [-]
$+X_b$	$(0, 0, -1, 0)^T$	$-Y_b$	$(0, 0, -0.7071, 0.7071)^T$
$-X_b$	$(0, 0, 0, 1)^T$	$+Z_b$	$(0, -0.7071, 0, 0.7071)^T$
$+Y_b$	$(0, 0, 0.7071, 0.7071)^T$	$-Z_b$	$(0, 0.7071, 0, 0.7071)^T$

Table 3.7: Solar radiation pressure moment generated on a conical body with the Sun positioned on its $+X_B$-axis

β [deg]	0	Moment on curved surface [Nm]	$(0, 0, 0)^T$
α [deg]	180	Moment on flat surface [Nm]	$(0, 0, 0)^T$

Table 3.8: Solar radiation pressure moment generated on a conical body with the Sun positioned on its $-X_B$-axis

β [deg]	180	Moment on curved surface [Nm]	$(0, 0, 0)^T$
α [deg]	180	Moment on flat surface [Nm]	$(0, 0, 0)^T$

Table 3.9: Magnitudes of solar radiation pressure moment of a conical body for different test cases, compared to reference value provided by Larson and Wertz (1999)

Test Case	Solar Radiation Pressure Torque [Nm]
$+Y_b$	$0.6726 \cdot 10^{-6}$
$-Y_b$	$0.6726 \cdot 10^{-6}$
$+Z_b$	$0.6726 \cdot 10^{-6}$
$-Z_b$	$0.6726 \cdot 10^{-6}$
Reference value	10^{-6}

as the entire curved surface is illuminated and thus the symmetry around the X-axis causes that all components cancel each other out. The simulation results for this test case are provided in Table 3.7.

When the Sun is subsequently located on the negative X-axis of the body, only the lower flat surface is illuminated and no solar radiation pressure moment is generated, given that the line of action of its resulting force acts through the centre of mass of the spacecraft. The simulation results for this test case are provided in Table 3.8.

Note that similar results are obtained for the verification tests that position the Sun on the ±Y-axes and the ±Z-axes of the body frame and are provided in Appendix A. Hence from this discussion, it can be concluded that the orientation of the moment components follow from expectations. Furthermore, when considering the magnitude of the solar radiation pressure torques, it is given by Larson and Wertz (1999) that, for low-Earth orbiting spacecraft, this torque amounts to a value of approximately 10^{-6} Nm. When comparing the magnitudes of the solar radiation pressure torques of the provided test cases with this reference value, as can be seen in Table 3.9, it can be concluded that sufficient correspondence can be found with the applicable test cases $+Y_b$ to $-Z_b$. This correspondence is assumed sufficient when considering that the magnitude in (Larson and Wertz, 1999) is a reference value for all low-Earth orbiting spacecraft, without considering their respective surface properties or geometry. Moreover, the accuracy is deemed sufficient given the small time scale of the considered mission scenario during which these perturbations can act. Lastly, for the current thesis study, the emphasis is put on the relative perturbations between a target and chaser spacecraft during a rendezvous scenario, rather than their absolute perturbations. Therefore, the accuracy of the inertial perturbations is only of secondary importance, as long as the relative perturbation between both spacecraft is sufficiently accurate. Hence it can be concluded that the developed moment model for the solar radiation pressure on a conical body is indeed implemented correctly.

3.5.2. Atmospheric Drag Model

To assess the correct implementation of the atmospheric drag models, a two-step verification process is followed. First, the sign and magnitude of the drag forces are compared with Montenbruck and Gill (2000) and Larson and Wertz (1999), after which the sign and magnitude of the drag moments are com-

Table 3.10: Simulation parameters used during atmospheric drag acceleration analysis

General Parameters		Parallelepiped Parameters	
epoch	01/01/2001 00 : 00 : 00.0	length [m]	1
simulation time [sec]	6027.14	height [m]	1
initial position [m]	$(7.158137, 0, 0)^T \cdot 10^6$	width [m]	1
initial velocity [m/s]	$(0, 7.4622, 0)^T \cdot 10^3$	**Cone Parameters**	
initial attitude quaternion [-]	$(0, 0, 0, 1)^T$	lower radius [m]	0.6
initial angular velocity [rad/s]	$(0, 0, 0)^T$	upper radius [m]	0.517
mass [kg]	100	height [m]	1
inertia tensor [kgm^2]	$\begin{bmatrix} 80 & 0 & 0 \\ 0 & 82 & 0 \\ 0 & 0 & 4 \end{bmatrix}$		
drag coefficient [-]	2		

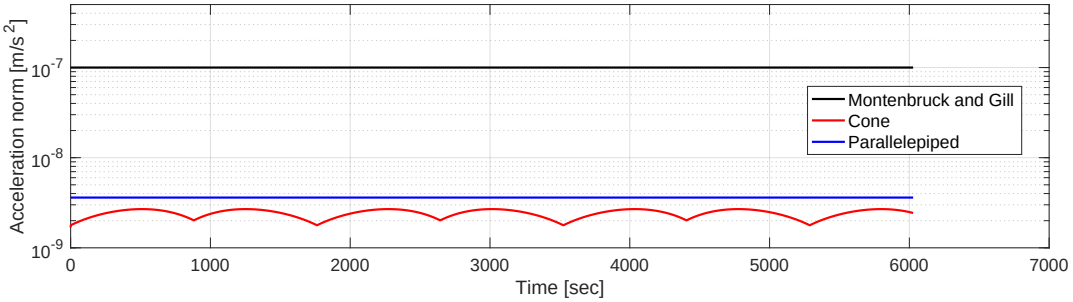


Figure 3.8: Time series of atmospheric drag acceleration norm and reference magnitude as provided by Montenbruck and Gill (2000)

pared with Larson and Wertz (1999). As previously mentioned for the solar radiation pressure model, also this perturbation model is relatively complex and few representative examples are available. This makes it difficult to compare the simulation results with predefined benchmarks. Therefore, it should be noted that the current verification is not exhaustive and solely meant to enhance the confidence in a correct implementation of the perturbation model.

A summary of the simulation parameters that were used during the force model testing are, unless specifically stated otherwise, provided in Table 3.10.

The resulting magnitude of the atmospheric drag accelerations acting on a conical and a parallelepiped spacecraft can be seen in Figure 3.8. Note that this figure also provides the estimate of the drag acceleration as determined by Montenbruck and Gill (2000) for a spacecraft with a representative area-to-mass ratio of 0.01 m^2/kg. Furthermore, the corresponding components of the drag accelerations can be found in Figure 3.9, which can be used to assess whether the direction of the acceleration models are implemented correctly.

When inspecting Figure 3.8, it can be observed that the magnitudes of the drag accelerations calculated by the disturbance models do not correspond well with the reference value provided by Montenbruck and Gill (2000). The calculated accelerations are considerably lower, despite providing a reference during minimum density conditions. However, as the magnitude of the accelerations are directly proportional to the atmospheric density, a small discrepancy in atmosphere model can cause huge differences. Therefore, it is assumed that the difference in acceleration magnitudes are caused by the simple atmosphere model used as external input to disturbance models. To verify this, a second test will subsequently be provided to assess the correct magnitude of the drag accelerations.

When inspecting Figure 3.9, it can be identified that at the initial epoch, the accelerations have a negative component along the Y-axis and the components along the X and Z-axes are zero. This result can be explained from the initial velocity provided in Table 3.10, as the atmospheric drag acts along the direction opposite to the velocity direction. Furthermore, as the simulated orbit is an equatorial orbit, the acceleration component along the Z-axis must remain zero, which can be seen in Figure 3.9, where the Z-component of the drag acceleration of the cone is practically zero to within machine

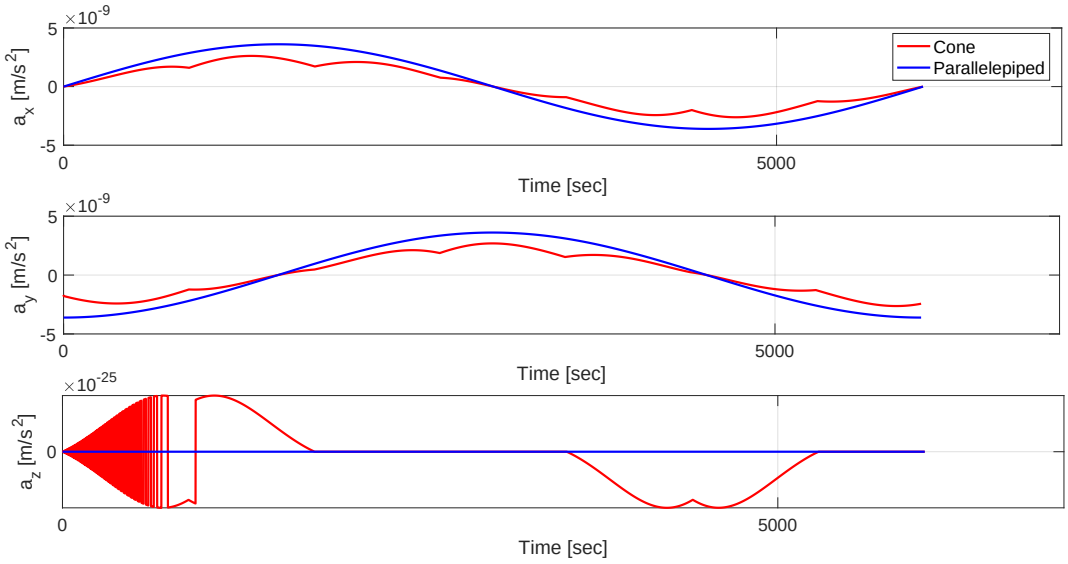


Figure 3.9: Time series of atmospheric drag acceleration components

Table 3.11: Change in semi-major axis after one orbital period in near-circular orbit due to atmospheric drag disturbance

Semi-major Axis Change	Analytical	Numerical
Parallelepiped Body [m]	−0.0241	−0.0417
Conical Body [m]	−0.0241	−0.0278

precision. Furthermore, during one full orbit, the acceleration components vary along the X and Y-axes in a sinusoidal fashion, where the magnitude along the X-direction is maximum when the magnitude along the Y-direction is zero and vice versa. This behaviour corresponds well with the results provided in Figure 3.9. Hence based on this discussion it can be concluded that at least the direction of the accelerations models is already implemented correctly.

To further verify the acceleration magnitudes, an analytic approximation for the change in semi-major axis caused by atmospheric drag is employed. Note that this analytic approximation is only valid for spacecraft in near-circular orbits and is taken from Larson and Wertz (1999). Furthermore, this relation is given below in Equation (3.25), where Δa represents the reduction in semi-major axis caused by atmospheric drag during one orbit, C_D the drag coefficient of the spacecraft, $\frac{A}{m}$ the area-to-mass ratio of the spacecraft, ρ the atmospheric density at the altitude of the spacecraft and a the semi-major axis of the spacecraft.

$$\Delta a = -2\pi C_D \frac{A}{m} \rho a^2 \tag{3.25}$$

When simulating a conical and parallelepiped spacecraft, the changes in semi-major axis as provided in Table 3.11 result. These semi-major axis changes are obtained by simulating the two spacecraft during one orbit, using the simulation parameters as established in Table 3.10. Moreover, Table 3.11 also contains the corresponding analytical changes in semi-major axis, which are calculated from Equation 3.25 using an area-to-mass ratio of 0.01 m^2/kg and atmospheric density of $3.7470 \cdot 10^{-15}$ kg/m^3. When comparing the numerical and analytical changes in semi-major axis, it can be seen that both values correspond well. Furthermore, when simulating multiple subsequent orbits, it can be observed that the semi-major axis continues to decrease. This causes the magnitude of the drag acceleration to increase, as the spacecraft dives deeper into the atmosphere. Hence from this discussion, it can be concluded that also the acceleration magnitude of the atmospheric drag models is implemented correctly.

When considering the atmospheric drag moment on a parallelepiped body, it should be noted that the same discussion as provided for the solar radiation pressure moment is applicable. This means that, as the implemented model locates the centre of mass at the geometric centre of the body and given that

Table 3.12: Initial body orientations used during atmospheric drag moment analysis

Relative Velocity	Initial quaternion [-]	Relative Velocity	Initial quaternion [-]
$+X_b$	$(0, 0, -0.7071, 0.7071)^T$	$-Y_b$	$(0, 0, 0, 1)^T$
$-X_b$	$(0, 0, 0.7071, 0.7071)^T$	$+Z_b$	$(0.7071, 0, 0, 0.7071)^T$
$+Y_b$	$(0, 0, -1, 0)^T$	$-Z_b$	$(-0.7071, 0, 0, 0.7071)^T$

Table 3.13: Atmospheric drag moment generated on a conical body with the velocity vector relative to the atmosphere positioned on its $+X_B$-axis

β [deg]	0	Moment on curved surface [Nm]	$(0, 0, 0)^T$
α [deg]	180	Moment on flat surface [Nm]	$(0, 0, 0)^T$

Table 3.14: Atmospheric drag moment generated on a conical body with the velocity vector relative to the atmosphere positioned on its $-X_B$-axis

β [deg]	180	Moment on curved surface [Nm]	$(0, 0, 0)^T$
α [deg]	0	Moment on flat surface [Nm]	$(0, 0, 0)^T$

the drag forces act through the geometric centre of their respective panel, the body will not generate a perturbing moment. The symmetry of the problem makes the components of the different faces cancel each other out, resulting in zero disturbance moment moment. This can be shown mathematically from the moment equations provided in Appendix A and both analytical and numerical results provide zero moment generation. Hence it can be concluded that both the orientation and magnitude of the drag moment on a parallelepiped body are implemented correctly.

When subsequently considering the drag moment on a conical body, it should be noted that also this model locates the centre of mass at the geometric centre of the body. However, the moment components will generally not cancel each other out, resulting in a drag moment generation. To analyse the orientation of the generated moments, the simulation parameters as previously provided in Table 3.5 are employed. Furthermore, the simulations are performed using the initial orientations provided in Table 3.12. These initial orientations are implemented as they locate the velocity vector relative to the atmosphere, $\hat{\mathbf{v}}$, on the ±X, ±Y and ±Z-axes of the body, respectively.

The current discussion only provides the results obtained for relative velocity vectors located on the ±X-axes of the body, as the remaining test cases provide similar results and are therefore presented in Appendix A. When the velocity vector relative to the atmosphere is located on the positive X-axis of the body, both the upper surface and the slanted surface are impinged. However, the flat surface will not generate a drag moment, given that the line of action of its resulting force acts through the centre of mass of the spacecraft. Furthermore, also the slanted side of the body will not generate a resulting moment, as its entire curved surface is illuminated. The simulation results for this test case are provided in Table 3.13.

When the velocity vector relative to the atmosphere is located exactly on the negative X-axis of the body, only the lower flat surface is impinged and no atmospheric drag moment is generated. This is because the line of action of its resulting force acts through the centre of mass of the spacecraft. The simulation results for this test case are provided in Table 3.14.

Similar results are obtained for the verification tests that position the relative velocity vector on the ±Y-axes and the ±Z-axes of the body frame. However, to centralise the discussion, these results are provided in Appendix A. From the discussion above, it can be concluded that the orientation of the moment components follow the expectations and can therefore be assumed to be implemented correctly. Furthermore, when considering the magnitude of the atmospheric drag moments, it is given by Larson and Wertz (1999) that, for low-Earth orbiting spacecraft, this torque amounts to a value of approximately 10^{-7} Nm. When comparing the magnitudes of the drag torques calculated during the verification tests with this reference number, as provided in in Table 3.15, it can be observed that a sufficiently close correspondence can be found for the applicable test cases $+Y_b$ to $-Z_b$. This correspondence is assumed sufficient when considering that the magnitude in (Larson and Wertz, 1999) is a reference value for all low-Earth orbiting spacecraft, without considering its surface properties or geometry. Moreover, given the small time scale of the implemented mission scenario, this perturbation can only act over a short

Table 3.15: Magnitudes of atmospheric drag moment of a conical body for different test cases, compared to reference value provided by Larson and Wertz (1999)

Test Case	Atmospheric Drag Torque [Nm]
$+Y_b$	$0.2296 \cdot 10^{-7}$
$-Y_b$	$0.2296 \cdot 10^{-7}$
$+Z_b$	$0.2296 \cdot 10^{-7}$
$-Z_b$	$0.2296 \cdot 10^{-7}$
Reference value	10^{-7}

time span, for which the provided accuracy is deemed sufficient. Lastly, the focus of the current thesis is on the relative perturbations between a target and chaser spacecraft, rather than their absolute perturbations. Therefore, this accuracy is only of secondary importance, such that it can be concluded that the developed moment model for a conical body is implemented correctly.

4 Guidance

The GNC system of a spacecraft is composed of three subsystems: the guidance system, the navigation system and lastly the control system. The task of the guidance system is to calculate a desired reference trajectory or tracking command for the chaser spacecraft, that provides it with a safe and successfull final approach of the target. Note that this reference trajectory, as generated by the guidance subsystem, thus provides a time series of the nominal state of the spacecraft. This is subsequently compared with an estimate of the actual state of the chaser, as obtained from the navigation system. To obtain this estimate, the navigation system combines information from an onboard model with sensory measurements in a filtering technique, resulting in a best estimate of the actual chaser state. Lastly, the control system determines the control commands, executed by the control actuators, that are required to remove any discrepancy between the actual state, as obtained from the navigation system, and the desired state, as obtained from the guidance system. A thorough discussion on these three systems will be provided in the following three chapters, where Chapter 4 discusses the guidance system, Chapter 5 the navigation system and Chapter 6 the control system.

Within the current chapter, Section 4.1 discusses the different segments of the terminal rendezvous phase and Section 4.2 discusses the selected guidance strategies corresponding to each of these mission segments. Section 4.3 discusses the backbone of the guidance system, *i.e.*, the guidance propagator. The section provides a trade-off of state representation that can be used during the propagations, as it is required to be accurate and computationally efficient. In addition to this, it discusses the implications of the selected propagator for the remaining guidance system. Furthermore, Section 4.4 elaborates the mathematical principles of convex optimisation in general terms, given that the guidance system will calculate its reference trajectory through use of this type of optimisation. Subsequently, Section 4.5 applies this convex guidance methodology to the mission segments of the terminal rendezvous phase. Moreover, Section 4.6 discusses the tracking guidance that is combined with the convex guidance system, whereas Section 4.7 provides details on the golden section search algorithm that is used to find the optimal time of flight for each mission segments. Lastly, Section 4.8 provides the verification procedures that are followed to ensure a correct implementation of the guidance algorithm.

4.1. Mission Scenario

When considering the mission scenario studied in the current research, as provided in Figures 4.1 to 4.5, it can be seen that it contains a total of five segments. Note that this sequence of mission segments is inspired by ESA's previous work on the e.Deorbit study, as given by Branco et al. (2021).

The first mission segment of the rendezvous phase deals with the transfer from hold point S_1, 20 m behind the target on its negative V-bar axis, to hold point S_2, 20 m above the target on its positive H-bar axis. A graphical representation of this phase can be found in Figure 4.1. This approach direction is considered as safest, given that the VESPA stage is likely to be in a flat spin, *i.e.*, minimum kinetic energy for constant angular momentum. This is a reasonable assumption when considering that internal dissipation will always transform arbitrary rotations to a flat spin within a relatively short time span, *i.e.*, approximately one year (Efimov et al., 2018). Subsequently, the chaser approaches the target along its H-bar axis from hold point S_2 to hold point S_3, which is located 5 m above the target as depicted in Figure 4.2. Note that it is mandatory to get quite close to the VESPA target prior to synchronising the position of the chaser with the motion of the tumbling docking-axis, as the chaser has limited thrusting capabilities to compensate the centrifugal forces encountered during this synchronisation manoeuvre. Evidently, the closer the chaser is to the target, the lower these compensating thrust forces must be. Therefore, a standoff distance of 5 m is employed, as it lies within the thruster capabilities of the chaser

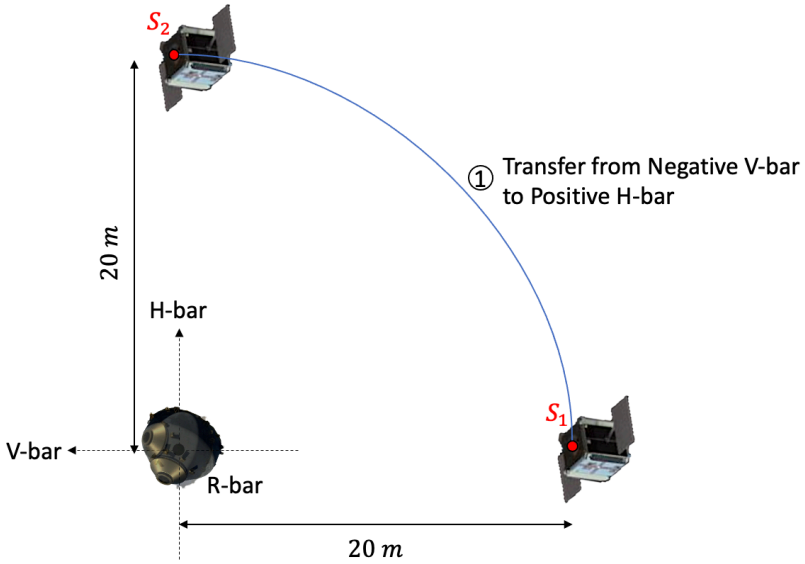


Figure 4.1: Transfer phase from negative V-bar to positive H-bar expressed in Hill frame $\mathcal{F}_H$
Source: Adapted from ClearSpace SA https://clearspace.today and Elecnor Deimos https://elecnor-deimos.com/clearspace/ [cited 13 July 2022]

spacecraft, while ensuring sufficient separation for collision avoidance.

During the third mission segment, as provided in Figure 4.3, the chaser spacecraft performs attitude synchronisation with the tumbling VESPA target, while maintaining its position on hold point S_3. Note however that for the considered reference mission, the rotational motion guidance will be decoupled from the translational motion guidance, given that the tumbling rates of the target spacecraft are rather limited, as provided by **REQ_S_T_9**. Moreover, it is assumed that omnidirectional thrust is available, such that a simplified guidance algorithm results (Leomanni et al., 2022). Consequently, attitude synchronisation will be maintained as an attitude control problem and treated separately from translational motion guidance. Furthermore, given the short time required for attitude synchronisation, up to a few minutes, no dedicated guidance algorithm is developed for this mission segment. Rather, a tracking guidance will be used to maintain proximity with hold point S_3. Furthermore note that from the third mission segment onward, attitude synchronisation is maintained. Then prior to approaching the VESPA target along its docking-axis in the final approach phase, the chaser spacecraft has to synchronise its position. This is executed during the fourth mission segment, as provided in Figure 4.4, where the chaser transfers from hold point S_3, on the positive H-bar of the target, to hold point S_4, 5 m separated from the target along the direction of the instantaneous docking-axis. Subsequently, the final approach phase between hold points S_4 and S_5 is initiated and the inter-satellite distance is reduced, while simultaneously maintaining position synchronisation. Hold point S_5 is located 1.35 m from the target along the direction of the instantaneous docking-axis, as given in Figure 4.5. Consequently, an out-of-plane closing fly-around manoeuvre will be performed during this phase and is necessary prior to capture, such that the target spacecraft becomes within reach of the robotic tentacles of the chaser.

Hence, the implemented mission scenario is currently limited to the close-range rendezvous phase, up to but not including the capture of the debris item. This is because the capture phase is defined as a synchronised flight of predetermined duration at fixed relative geometry with respect to the target, *i.e.*, tracking the path that is traced out by hold point S_5. This trajectory is needed for the robotic tentacles of the ClearSpace-1 chaser to deploy and provide rigid clamping around VESPA. However, the currently developed guidance system is not well suited for this mission phase, given that this predefined reference trajectory does not allow for any liberty in the solution of the convex system. Furthermore, due to the strict safety requirements and the complexity of the considered reference trajectory, it is expected that the developed linear tracking guidance system will not provide sufficient performance. Rather, a more advanced guidance system based on for example an adaptive controller should be developed, as given in (Zhu and Wang, 2020). However, due to the time constraints of the thesis project, the implementation of the capture phase and its corresponding guidance system have been left for future work.

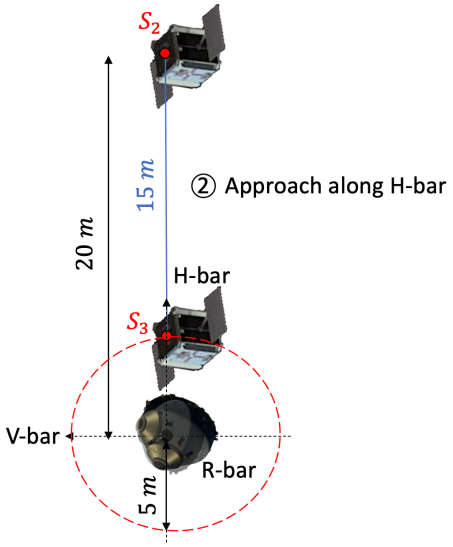


Figure 4.2: Approach along positive H-bar expressed in Hill frame $\mathcal{F}_H$
Source: Adapted from ClearSpace SA https://clearspace.today and Elecnor Deimos https://elecnor-deimos.com/clearspace/ [cited 13 July 2022]

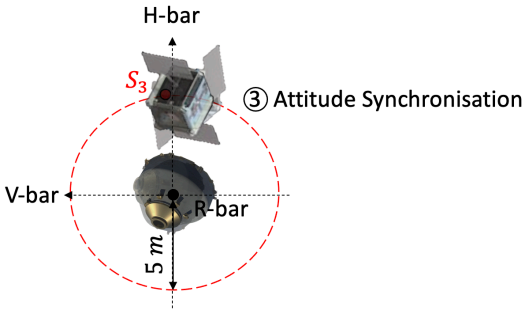


Figure 4.3: Attitude synchronisation with tumbling target expressed in Hill frame $\mathcal{F}_H$
Source: Adapted from ClearSpace SA https://clearspace.today and Elecnor Deimos https://elecnor-deimos.com/clearspace/ [cited 13 July 2022]

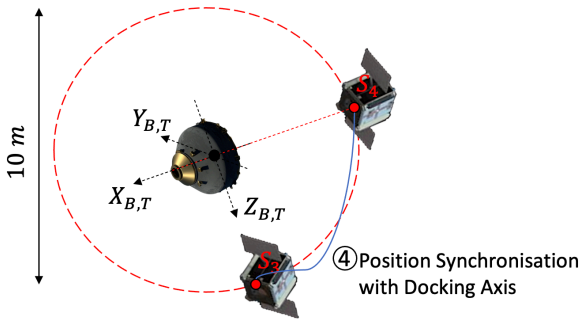


Figure 4.4: Position synchronisation with tumbling docking-axis of target expressed in Body-Centred-Body-Fixed frame $\mathcal{F}_{B,T}$
Source: Adapted from ClearSpace SA https://clearspace.today and Elecnor Deimos https://elecnor-deimos.com/clearspace/ [cited 13 July 2022]

4.2. Guidance Algorithms

A guidance law onboard a spacecraft has the designated task of calculating the most optimal trajectory to reach its objective. For rendezvous trajectories, this most optimal trajectory often consists of the most fuel efficient path, most time efficient path, the path that minimises the thruster plume impingement or the path that combines any of these previously stated objectives. Furthermore, this reference trajectory is also subject to a number of trajectory constraints such as, for example, keep-out spheres or approach

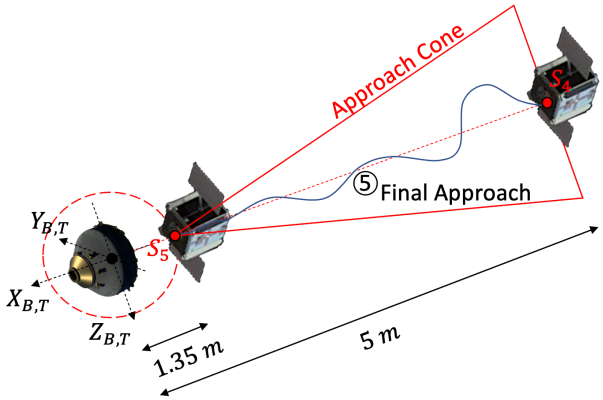


Figure 4.5: Final approach along docking-axis of target expressed in Body-Centred-Body-Fixed frame $\mathcal{F}_{B,T}$
Source: Adapted from ClearSpace SA https://clearspace.today and Elecnor Deimos https://elecnor-deimos.com/clearspace/ [cited 13 July 2022]

cones. However, for the given reference mission concerned with deorbiting a passive debris item, the plume impingement constraint is considered less important when compared to more conventional rendezvous missions. This is caused by the rotating nature of the target object with which the chaser has to synchronise. Due to this position synchronisation of chaser and rotating docking-axis of the target, the commanded thrust vector will be directed away from the target to counteract the experienced centrifugal forces (Leomanni et al., 2022) and therefore it is evident that the trajectory is more effectively optimised for fuel consumption and/or rendezvous time.

In general, there are three types of guidance methodologies that can be followed: fly-by-wire guidance, where one tracks a reference trajectory that was precalculated by ground operators before launch, explicit guidance, where one tracks a reference trajectory analytically calculated onboard the spacecraft after applying simplifying assumptions, and numerical guidance, where one tracks a reference trajectory that has been numerically optimised onboard the spacecraft. To select one of these guidance methodologies, it is important to consider their robustness to off-nominal conditions. The fly-by-wire methodology is not able to react to large trajectory dispersions and cannot provide a contingency plan, as the spacecraft can only fly one of the precalculated trajectories. This methodology thus provides only very low robustness. Furthermore, with the advent of more onboard computational power, the spacecraft itself is able to calculate its reference trajectory either analytically or numerically. This has the clear benefit of being adaptable to changing conditions and therefore lends itself well for the rendezvous problem. It enhances the robustness of the system, as required for autonomous execution of the terminal rendezvous phase. Consequently, it is decided to further omit the fly-by-wire guidance methodology, and only focus on the explicit or numerical guidance methodologies. This implies that the guidance subsystem onboard the chaser will have to propagate the state of the spacecraft to calculate a desired reference trajectory, which can provide safe and successfull mission execution. However, for the considered reference mission, it is required by ESA that the guidance system onboard the chaser spacecraft is capable of providing a real-time onboard optimisation of the terminal rendezvous phase, as given by **REQ_S_C_14**. In particular, it is given in ESA's procurement document (European Space Research and Technology Centre 2020) that a convex trajectory optimisation has to be performed by the terminal guidance system. This type of guidance methodology is evidently a numerical guidance system and will applied to all mission segments elaborated in Section 4.1, with an exception of the attitude synchronisation phase, as this will be treated by the tracking guidance algorithm.

4.3. Guidance Propagator

The propagator of the guidance system has the designated task of propagating the dynamics of the chaser spacecraft with respect to its target based on the calculated guidance commands, to assess the viability and optimality of the generated reference trajectory. For this propagator, it is important to assess whether the states of both spacecraft will be propagated inertially, or whether the propagator will directly calculate their relative motion. Since the current research is concerned with the terminal rendezvous phase between the VESPA target and the ClearSpace-1 chaser, the guidance system is

in principle only concerned with relative motion between both spacecraft. There are numerous simplified models that describe relative orbital motion and can provide closed form analytical solutions. The most common of these models is the linear Hill model (Hill, 1878), which assumes that the distance between the target and chaser is small compared to the distance between the target and the central body. Moreover, it assumes that the target's orbit is near-circular. The fundamental advantage this form of propagation solution provides, is given by the fact that one solves the relative motion between the chaser and target in the relative frame more efficiently and swiftly than solving the exact nonlinear equations of motion in the inertial frame. Note that high computational efficiency is of paramount importance for the current reference mission, as the guidance system shall provide real-time onboard optimisation of the terminal rendezvous phase, as given by **REQ_S_C_14**. Nevertheless, these Hill equations were derived after applying some simplifying assumptions, resulting in a calculated relative motion that is less accurate than propagating the exact nonlinear equations. When considering the applicability of the Hill equations for the reference scenario, as analysed in the current research, it can be established that, given the low-Earth orbit within which both spacecraft move, the relative distance between the central body and the target is much larger than the 20 m of relative distance between target and chaser at the start of the planned mission scenario. Moreover, the 801 by 664 km target orbit is near-circular, implying that the reference scenario satisfies all simplifying assumptions as required by the Hill equations. Therefore, the trade-off boils down to assessing the reduction in computational effort with respect to the reduction in accuracy that results when propagating the Hill equations rather than the exact nonlinear equations of motion.

When considering the deficit in numerical accuracy resulting from the Hill equations, it must be noted that numerous extensions to this model have been developed, which are capable of taking the primary perturbations in low-Earth orbit into account. Examples of such linearised models are, among others, given by Silva (2008) and Carter and Humi (2002), which include the perturbation of aerodynamic drag, or Arya et al. (2016), which includes the J_2 perturbation of Earth's gravity field. By taking these primary perturbation into account, a considerable increase in accuracy can be achieved for the Hill equations. Moreover, any inaccuracy in the propagated state of the guidance system can, in principle, be corrected for by the navigation subsystem, given that sufficiently accurate measurements of the spacecraft state are available. In reality, however, measurements will always be non-ideal and contain some kind of inaccuracy. Yet, it can safely be assumed that, given the sensors used onboard the ClearSpace-1 satellite, a navigation accuracy at cm level in terms of position and degree level in attitude is feasible, as given by the VBN experiment of the RemoveDEBRIS mission (Marchand et al., 2019). Therefore, it can be concluded that the loss of accuracy can largely be compensated for by the navigation sensor measurements and, if need be, the propagator accuracy can be increased by extending the Hill equations to include, among others, perturbing accelerations such as atmospheric drag or J_2. Hence, the benefit of the reduced computational effort, as required for the Hill equations, easily outweighs the relatively minor loss in accuracy, when combined with the navigation subsystem. Note that the outcome of this rudimentary trade-off should not be surprising given the frequent application of this model in the guidance laws, as provided in Table 2.2. Hence it can be concluded that the propagation of the guidance system designed for the current reference mission will incorporate the linearised Hill equations.

A mathematical description of this system of linearized differential equations is taken from (Curtis, 2013) and provided in Equation (4.1). Note that in these equations, $\ddot{x}$, $\ddot{y}$ and $\ddot{z}$ represent the three Cartesian accelerations components, $\dot{x}$, $\dot{y}$ and $\dot{z}$ the three Cartesian velocity components and x, y and z the three Cartesian position components of the chaser spacecraft. These Cartesian components are all referenced with respect to the $\mathcal{F}_L$ frame centred at the target spacecraft. Moreover, n represents the mean motion of the target satellite in its orbit around Earth and f_x, f_y and f_z represent the components of the force per unit mass acting on the spacecraft external to the Newtonian gravity of the main body. Note that these equations generally assume only a central body gravitational attraction and the additional forces are often modelled as the thrust force of the spacecraft. An extension of these models can, however, be developed to contain perturbations such as drag or an extended central body gravity field next to the thrust force, but this is currently omitted and will only be provided, if proven to be required to increase the accuracy of the onboard propagator.

$$\begin{cases} \ddot{x} - 2n\dot{y} - 3n^2x = f_x \\ \ddot{y} + 2n\dot{x} = f_y \\ \ddot{z} + n^2z = f_z \end{cases} \tag{4.1}$$

Hence the state vector of the translational guidance propagator can be expressed as $\mathbf{x} = (\mathbf{r}, \mathbf{V})^T = \left(x, y, z, \dot{x}, \dot{y}, \dot{z}\right)^T$ such that the equations of motion can be expressed by Equations (4.2) and (4.3) with $\mathbf{T}$ representing the spacecraft's thrust vector and m representing its mass.

$$\dot{\mathbf{r}}(t) = \mathbf{V}(t) \tag{4.2}$$

$$\dot{\mathbf{V}}(t) = \begin{bmatrix} 0 & 2n & 0 \\ -2n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}(t) + \begin{bmatrix} 3n^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -n^2 \end{bmatrix} \mathbf{r}(t) + \frac{\mathbf{T}(t)}{m} \tag{4.3}$$

Evidently, the selection of the propagator directly impacts the optimisation problem solved by the guidance system. The solution to this optimisation problem consists of an optimal reference trajectory for by chaser spacecraft, along with the commanded thrust history necessary to fly said reference trajectory, both referenced with respect to the Hill frame of the target. However, to simulate the actual motion of the chaser spacecraft, as performed in the motion simulator, it is required that the calculated thrust commands are transformed from the Hill frame to the inertial frame. Furthermore, when actuator models are employed, it is required that these thrust commands are further transformed to the Body-Centred-Body-Fixed frame of the chaser spacecraft. The transformation of the thrust commands between the Hill and inertial reference frames is, however, not as straightforward as those provided in Chapter 3. This is given by the fact that, effectively, the thrust commands are acceleration components that have to be transformed between two observer frames, *i.e.*, the thrust commands need to be transformed such that these can be propagated in the inertial frame, rather than the Hill frame. This is further complicated by the fact that both frames translate and rotate with respect to each other. Therefore, this transformations will be described in detail here and its derivation is based on the information provided in (Curtis, 2013). Note that this is in contrast to the transformation of the thrust commands from the inertial frame to the body frame, given that it constitutes a simple transformation of the presenter frame, *i.e.*, the frame in which the thrust vector is presented changes, rather than the frame in which it is observed. This is because the inertial frame remains the observer with respect to which the motion of the chaser is referenced and therefore greatly simplifies the transformation.

First the transformation from the Hill to the inertial frame is performed, after which the commands are subsequently transformed from the inertial to the Body-Centred-Body-Fixed frame of the chaser spacecraft.

Thrust Command Transformations

The thrust commands calculated in the Hill frame of the target, as performed by the guidance system, represent relative accelerations of the chaser with respect to a moving target. Evidently, to transform these accelerations to an inertial frame, the relative motion of the Hill frame with respect to the inertial frame has to be accounted for. To complicate matters, this movement consists of both a rotation and an acceleration. Therefore, the most generic frame transformation has to be applied. Using the information provided in (Curtis, 2013), the inertial acceleration of the chaser can be described using Equation (4.4). In this equation, $\mathbf{a}_C^{\mathcal{F}_I}$ and $\mathbf{a}_T^{\mathcal{F}_I}$ represent the acceleration vector of chaser and target expressed in the inertial frame, respectively. Moreover, $\mathbf{a}_{rel}^{\mathcal{F}_I}$, $\mathbf{V}_{rel}^{\mathcal{F}_I}$ and $\mathbf{r}_{rel}^{\mathcal{F}_I}$ represent the relative acceleration, velocity and position vector of the chaser with respect to the target, expressed in the inertial frame. Lastly, Ω and $\dot{\Omega}$ represent the angular velocity and angular acceleration vectors of the Hill frame with respect to the inertial frame.

$$\mathbf{a}_C^{\mathcal{F}_I} = \mathbf{a}_{rel}^{\mathcal{F}_I} + \mathbf{a}_T^{\mathcal{F}_I} + \dot{\Omega} \times \mathbf{r}_{rel}^{\mathcal{F}_I} + \Omega \times \left(\Omega \times \mathbf{r}_{rel}^{\mathcal{F}_I}\right) + 2\Omega \times \mathbf{V}_{rel}^{\mathcal{F}_I} \tag{4.4}$$

The relative position, velocity and acceleration vectors of the chaser with respect to the target, expressed in the inertial frame, can be calculated using Equations (4.5) to (4.7), respectively. In Equations

(4.5) and (4.6), $\mathbf{r}_{C}^{\mathcal{F}_I}$ and $\mathbf{r}_{T}^{\mathcal{F}_I}$ represent the inertial position vectors of the chaser and target spacecraft, whereas $\mathbf{V}_{C}^{\mathcal{F}_I}$ and $\mathbf{V}_{T}^{\mathcal{F}_I}$ represent their respective inertial velocity vectors. Furthermore in Equation (4.7), $\mathbf{C_{I/L}}$ represents the transformation matrix from the Hill or Local-Vertical-Local-Horizontal frame to the inertial frame and can be obtained from the transformations provided in Chapter 3. Moreover, this equation contains the relative dynamics in the Hill frame and therefore follows the definitions as previously provided for Equation (4.3). It should, however, be noted that this equation requires the relative position and velocity vector of the chaser spacecraft in the Hill frame, as given by $\mathbf{r}_{rel}^{\mathcal{F}_H}$ and $\mathbf{V}_{rel}^{\mathcal{F}_H}$, and can be obtained from the reference trajectory calculation provided by the guidance system. Furthermore, in this equation, $\mathbf{T}_{C}^{\mathcal{F}_H}$ represents the thrust commands of the guidance system, as calculated in the Hill frame, and m represents the mass of the chaser.

$$\mathbf{r}_{rel}^{\mathcal{F}_I} = \mathbf{r}_{C}^{\mathcal{F}_I} - \mathbf{r}_{T}^{\mathcal{F}_I} \tag{4.5}$$

$$\mathbf{V}_{rel}^{\mathcal{F}_I} = \mathbf{V}_{C}^{\mathcal{F}_I} - \mathbf{V}_{T}^{\mathcal{F}_I} - \Omega \times \mathbf{r}_{rel}^{\mathcal{F}_I} \tag{4.6}$$

$$\mathbf{a}_{rel}^{\mathcal{F}_I} = \mathbf{C_{I/L}} \left(\begin{bmatrix} 0 & 2n & 0 \\ -2n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{V}_{rel}^{\mathcal{F}_H} + \begin{bmatrix} 3n^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{r}_{rel}^{\mathcal{F}_H} + \frac{\mathbf{T}_{C}^{\mathcal{F}_H}}{m} \right) \tag{4.7}$$

The angular velocity and angular acceleration vector of the Hill frame with respect to the inertial frame are defined in Equations (4.8) and (4.9).

$$\Omega = \frac{\mathbf{r}_{T}^{\mathcal{F}_I} \times \mathbf{V}_{T}^{\mathcal{F}_I}}{||\mathbf{r}_{T}^{\mathcal{F}_I}||^2} \tag{4.8}$$

$$\dot{\Omega} = -2 \frac{\mathbf{V}_{T}^{\mathcal{F}_I} \cdot \mathbf{r}_{T}^{\mathcal{F}_I}}{||\mathbf{r}_{T}^{\mathcal{F}_I}||^2} \Omega \tag{4.9}$$

Lastly, the inertial acceleration acting on the target spacecraft can be defined using Equation (4.10), where μ represents the gravitational parameter of Earth. Note that, as the linearised Hill equations only model a central gravity field, the considered acceleration of the target spacecraft, as used during reference frame transformation, must also be limited to this simple gravity field model. Consequently, the inertial acceleration acting on the chaser spacecraft can be obtained through a similar formulation, as defined in Equation (4.11). Note that for the chaser spacecraft, the central gravity acceleration is extended with the thrust force acceleration, which is defined in the inertial frame as $\frac{\mathbf{T}_{C}^{\mathcal{F}_I}}{m}$. Combining this equation with Equation (4.4) results in Equation (4.12), defining the thrust commands of the guidance system in the inertial frame.

$$\mathbf{a}_{T}^{\mathcal{F}_I} = \frac{-\mu}{||\mathbf{r}_{T}^{\mathcal{F}_I}||^3} \mathbf{r}_{T}^{\mathcal{F}_I} \tag{4.10}$$

$$\mathbf{a}_{C}^{\mathcal{F}_I} = \frac{-\mu}{||\mathbf{r}_{C}^{\mathcal{F}_I}||^3} \mathbf{r}_{C}^{\mathcal{F}_I} + \frac{\mathbf{T}_{C}^{\mathcal{F}_I}}{m} \tag{4.11}$$

$$\mathbf{T}_{C}^{\mathcal{F}_I} = m \left(\mathbf{a}_{C}^{\mathcal{F}_I} + \frac{\mu}{||\mathbf{r}_{C}^{\mathcal{F}_I}||^3} \mathbf{r}_{C}^{\mathcal{F}_I} \right) \tag{4.12}$$

To subsequently transform the thrust commands from the inertial frame, $\mathcal{F}_I$, to the Body-Centred-Body-Fixed of the chaser spacecraft, $\mathcal{F}_{B,C}$, the straightforward relation provided in Equation (4.13) can be used. Note that in this equation, $\mathbf{T}_{C}^{\mathcal{F}_{B,C}}$ represents the thrust vector expressed in the body frame, $\mathbf{T}_{C}^{\mathcal{F}_I}$ the thrust vector expressed in the inertial frame and $\mathbf{C_{B/I}}$ the frame transformation from the inertial to the body frame, as can be obtained from Equation (3.11).

$$\mathbf{T}_{C}^{\mathcal{F}_{B,C}} = \mathbf{C_{B/I}} \mathbf{T}_{C}^{\mathcal{F}_I} \tag{4.13}$$

4.4. Convex Optimisation

The final approach phase of an active debris removal mission constitutes a highly-constrained and complex trajectory planning problem for the guidance system. To handle this complexity, a number of guidance systems can be devised. However, one promising candidate is presented in the form of convex guidance, due to its beneficial characteristics in terms of robustness, autonomy and computational efficiency (Malyuta et al., 2021). In particular, convex guidance is characterised by favourable convergence properties, even while handling complex path constraints (Virgili-Llop et al., 2019). This implies that it can provide real-time continuous optimisation of a constrained trajectory. Moreover, it is mathematically guaranteed that convex guidance will find an optimal solution within a fixed number of iterations in case a feasible solution is contained within the optimisation space, or alternatively provides a certitude of infeasibility when no solution can be found (Mao et al., 2018). This is in contrast to nonlinear programming, as alternative optimisation strategy to convex guidance, as it cannot guarantee to provide a solution, and even when a solution is found, it cannot be guaranteed to be the global optimal (Li and Sun 2006). Hence, it can be concluded that convex optimisation is ideally suited to be implemented as part of the guidance system of an ADR mission. Therefore, it has been selected by ESA to employ a real-time convex guidance system for the terminal rendezvous phase of the reference mission considered in the current thesis research (European Space Research and Technology Centre 2020).

In recent years, the use of convex guidance for aerospace applications has seen an increased level of interest. Most noteworthy are the results achieved by applying convex guidance to powered descents on planetary bodies, which have been used by SpaceX to land the booster rockets of their Falcon 9 and Falcon Heavy rockets back on a platform on Earth (Blackmore, 2016). This is in contrast to the application of convex guidance for rendezvous and docking missions, as they have currently only been studied within a theoretical framework. Examples are provided by Virgili-Llop et al. (2019), who used convex guidance to simultaneously control the motion of the chaser spacecraft and its robotic manipulator during capture of a tumbling debris object, Lu and Liu (2013), who implemented convex guidance for autonomous rendezvous between two spacecraft, and Malyuta et al. (2020), who applied a convex guidance strategy to re-optimise the transposition and docking manoeuvre of the Apollo missions to obtain a 90% more propellant efficient solution with respect to their design objective. This therefore motivates the reason why the current thesis researches convex guidance for rendezvous and docking scenarios in an ADR mission, as it provides an additional step in lifting the technology from the theoretical framework to its practical counterpart.

In the following subsections, the basics on convex optimisation are discussed in preparation of its application onboard the guidance system. Firstly, Subsection 4.4.1 discusses what exactly constitutes a convex optimisation problem. Secondly, Subsection 4.4.2 details how to solve a convex optimisation problem and thirdly Subsection 4.4.3 provides a solution strategy on how to transform a non-convex optimisation problem into a convex equivalent.

4.4.1. Convex Problem Formulation

Before being able to apply convex optimisation theory to generate reference trajectories for the guidance system, it has to be established what exactly constitutes a convex problem and how to optimise it. A general mathematical optimisation problem can be characterised by Equation (4.14), where it should be noted that the optimum refers to the minimum. In this equations, $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$ represents the objective function, *i.e.*, the function to be minimised, whereas the functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ represent the constraint functions, *i.e.*, the functions that constrain the domain within which the minimum is to be found. Furthermore, $\mathbf{x}$ represents the vector containing the optimisation variables of the problem with $\mathbf{x} = [x_1, \ldots, x_n]$, which are the parameters that can be varied to find the minimum solution. Furthermore, the constants b_i for $i = 1, \ldots, m$ represent the bounds of these constraints. A vector $\mathbf{x}^*$ is called a solution of the problem, or optimal, if it results in the smallest function value of the objective function f_0 among the solution domain, while satisfying all constraints (Boyd and Vandenberghe, 2004).

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimise}} \quad & f_0(\mathbf{x}) \\ \text{subject to} \quad & f_i(\mathbf{x}) \leq b_i, \; i = 1, \ldots, m \end{aligned} \tag{4.14}$$

In general, different classes of optimisation problems are characterised by the type of objective and constraint functions that are applied. For convex optimisation, one seeks to find the minimum of a

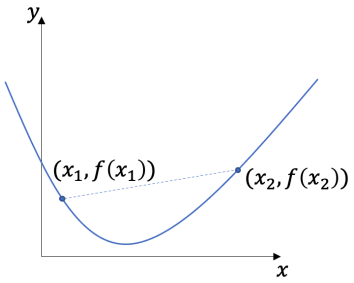


Figure 4.6: Convex function f() for which any line segment connecting two points on the function will lie above the function's curve

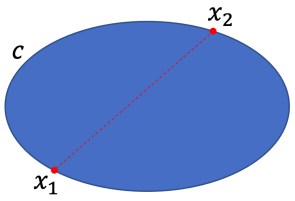


Figure 4.7: Convex set $\mathcal{C}$ for which every point can be seen by every other point along an unobstructed straight path between them

convex function over a convex domain of function variables. Hence for the class of convex optimisation problems, the objective and constraint functions are all convex which means these must satisfy the inequality, as given in Equation (4.15), for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and all $\alpha, \beta \in \mathbb{R}$ subject to $\alpha + \beta = 1$, $\alpha > 0$ and $\beta > 0$ (Boyd and Vandenberghe, 2004). This inequality can be explained graphically using Figures 4.6 and 4.7. For a function $f : \mathbb{R} \rightarrow \mathbb{R}$ to be convex, any line segment between two points must lie above the graph of the function itself, as given by Figure 4.6, and the domain of the function must constitute a convex set. This also further applies to higher order functions, but as it does not provide a very intuitive geometrical representation, it is not provided in Figure 4.6. Furthermore, a convex set $\mathcal{C}$ is provided in Figure 4.7 and defined as a set within which every point can be seen by every other point, along an unobstructed straight path between them (Boyd and Vandenberghe, 2004).

$$f_i(\alpha\mathbf{x} + \beta\mathbf{y}) \leq \alpha f_i(\mathbf{x}) + \beta f_i(\mathbf{y}) \tag{4.15}$$

4.4.2. Convex Solver

Real-time onboard implementation of convex optimisation-based algorithms has recently been made possible due to advances in their respective solution strategies. These improvements considerably reduced the computational burden and resulted in optimal solutions that are obtainable within polynomial time. To achieve this, recent developments exploited either generic Second-Order Cone Programming (SOCP) methods or a specific solver customised to the particular problem at hand (Mao et al., 2018). Given that the current thesis study is focused on the design of a GNC system for an ADR mission, rather than the development of a new mathematical solver, it is decided to use a generic SOCP solver.

For the selection of the solver, the choice is limited to open-source and/or academically available algorithms, for which a number of alternatives are provided. Among others, this list contains MOSEK, Gurobi, SDPT3, SeDuMi and ECOS, and details related to each of these solvers can be found on their respective websites. Nevertheless, it should be noted that the selection of a specific convex solver from this list will only influence the required solution time, not the obtainable solution. This is because, for a well-defined convex problem, the optimal solution represents a global optimal and is accessible for any SOCP within a limited number of iterations. Therefore, within the scope of the current thesis study, it is assumed that all solvers in the previously provided list are equally capable of obtaining this global optimal solution. For a real-time implementation of the developed guidance algorithm, however, it is of utmost importance to select the most efficient SOCP solver. Despite this, the current thesis study chooses to select one that facilitates implementation in the guidance system, whereas computational time has only been given secondary importance. To achieve this, the decision has been made to use an interfacing software in Matlab capable of wrapping around a SOCP solver. Even though some computational efficiency might be lost, a great amount of implementation convenience can be gained.

To this extent, two alternative wrapping software are considered. The first interfacing software is named CVX and is developed as a modelling language dedicated to convex optimisation problems (Liberti and Maculan, 2006). The second interfacing software is called Yalmip and is developed as a more generic modelling language, not solely restricted to convex optimisation problems (Löfberg, 2004). Both software, however, operate in a similar fashion by transforming the objective and constraint function definitions of the optimisation problem from Matlab syntax to the specific syntax necessary for the dedicated SOCP solvers. Therefore both software require a comparable implementation and a

selection is made based on their solution time. Note that for this selection, no detailed analysis on the solution performance is performed as it is not deemed necessary, given that the current study provides only a first analysis of the developed GNC system. For future design iterations, it is recommended to select a dedicated solver optimal at solving the problem at hand. Nonetheless, from preliminary analyses on the final approach guidance problem, it is uncovered that on average, the Yalmip solver is twice as fast in finding a solution, when compared to CVX. This difference in solution time is, without stepping into too much detail, caused by the way in which each software compiles the SOCP models from the Matlab syntax, given that the subsequent convex solvers are identical for both wrappers. Hence from this observation, it is decided to employ Yalmip, as the interfacing software, in combination with Mosek, as the SOCP solver. Note that the selection for Mosek ensues from the work performed by Lu and Liu (2013) and Bhagat (2016), which selected Mosek as convex solver and therefore provide proof of its functionality as part of the convex guidance system in an ADR mission.

4.4.3. Convexification Strategy

For a problem to be of convex form, it has to satisfy strict conditions on its objective and constraint functions. Therefore, most real-world problems are non-convex. Still, due to the computational efficiency in finding the global optimal solution, effort is put in the conversion of non-convex problems into convex ones. This process is referred to as convexification and, in general, two procedures exist that maintain optimality and feasibility transferable to the original non-convex problem. These two convexification methods are known as lossless convexification, which applies a convex relaxation of the original non-convex problem through the use of a slack variable, and successive convexification, which applies an iterative procedure that solves a sequence of convex subproblems (Mao et al., 2018). Based on the type of non-convex problem at hand, one or a combination of these transformations will be applied to convert a generic optimisation problem into a convex one. However, sometimes it is possible to combine insight into the problem with a smart redefinition of the objective and/or constraint functions to generate a convexification strategy that effectively avoids lossless and/or successive convexification. This can be achieved when using an alternative mathematical formulation that is not exactly equivalent to the non-convex formulation, but in all practical cases will result in a equivalent or nearly-equivalent solution.

Through application of clever mathematical tricks, the convexification strategy followed in the current thesis study has been performed in this manner. The benefits of this strategy are provided by the fact that the feasibility and transferability of the obtained solution to the original design space do not have to be checked. This is in contrast to lossless convexification, for which detailed proof must be provided that the convex relaxation transfers the globally optimal solution of the lifted problem to, not only a feasible solution of the non-lifted problem, but also to the solution that corresponds to its global optimum. This can be provided when a number of complex mathematical theorems and conditions are satisfied and is therefore preferably avoided. Furthermore, also no iterative procedure needs to be applied, as is performed during successive convexification, and therefore the algorithm will require less time to converge to the optimal solution. Moreover, the algorithm is guaranteed to provide the optimal solution, when available within the design space, in contrast to successive convexification, which might converge to a local optimal within the linearisation region. Therefore, the author is convinced that a convexification strategy that can avoid these two generic methodologies is optimal and should be sought for the current thesis study.

4.5. Convex Guidance

The convex guidance system contains four distinct guidance algorithms that are each applied to one of the corresponding mission segments, as previously discussed in Section 4.1. However, given that the optimisation problems for these four algorithms are similar, but not identical, the current section will first define a general optimisation problem that is applicable to all mission segments. Note that this general optimisation problem is therefore incomplete and the additional constraint definitions are discussed per mission segment in Subsections 4.5.1 to 4.5.4.

The general optimisation problem solved by the guidance system is provided in Problem 1. Note that, as discussed above, the guidance system is only concerned with the translational motion of the chaser spacecraft and thus the optimisation problem contains three degrees of freedom. The objective function, given by J_f, is provided in Equation (4.16), where the magnitude of the cumulative thrust

force, given by $||\mathbf{T}(t)||$, is minimised over the time of flight of the mission segment, as indicated by t_f. It is chosen to minimise the integrated thrust value, given that it provides a direct measure of the propellant usage. Furthermore, next to the objective function, the optimisation problem is also characterised by constraint functions, which are provided in Equations (4.17) to (4.23). Chiefly, the problem is constrained by the equations of translational motion given in Equations (4.17) and (4.18). Definitions for the variables provided in these equations have previously been discussed in Section 4.3 on the guidance propagator. Furthermore, as a common safety precaution for the considered mission segments, the reference trajectory of the chaser spacecraft must remain outside a keep-out sphere with a radius r_{KOS}, as given by Equation (4.19). Note that this requirement is instituted to ensure that the chaser does not accidentally collide with its target during proximity operations. Moreover, a maximum transfer velocity is instituted by the upper limit V_{max}, as represented in Equation (4.20), to ensure that, in case a potential impact occurs during contingency operations, the impact will remain low. Additionally, Equation (4.21) constrains the thrust vector history to the performance characteristics of the onboard engines, given by lower bound T_{min} and upper bound T_{max}. Lastly, the problem is also constrained to satisfy the initial conditions provided in Equation (4.22) and the final conditions provided in Equation (4.23). It should be noted that, as **REQ_S_C_2** states that the chaser's mass is assumed to be known and constant throughout the considered mission scenario, no mass dynamics are included in the guidance system. Therefore, there is also no constraint implemented on maximum propellant consumption. This simplifies the guidance system and is considered reasonable, given that the propellant consumption of the considered mission segments are expected to remain low, *i.e.*, only a fraction of the total mass of the spacecraft and are therefore insignificant.

The guidance problems thus consist of finding the time histories of the thrusts vector, given by $\mathbf{T}(t)$, along with the reference trajectory, given by $\mathbf{r}(t)$ and $\mathbf{V}(t)$, that minimise the objective function J_f while satisfying all path constraints, as provided in Problem 1. This problem is repeatedly solved onboard the chaser spacecraft during the different mission segments, using the current best navigation estimate as initial condition.

Problem 1:

$$\underset{\mathbf{T}(t),\mathbf{r}(t),\mathbf{V}(t)}{\text{minimise}} \quad J_f = \int_0^{t_f} ||\mathbf{T}(t)||dt \quad \text{subject to} \tag{4.16}$$

$$\dot{\mathbf{r}}(t) = \mathbf{V}(t) \tag{4.17}$$

$$\dot{\mathbf{V}}(t) = \begin{bmatrix} 0 & 2n & 0 \\ -2n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \mathbf{V}(t) + \begin{bmatrix} 3n^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -n^2 \end{bmatrix} \cdot \mathbf{r}(t) + \frac{\mathbf{T}(t)}{m} \tag{4.18}$$

$$||\mathbf{r}(t)|| \geq r_{KOS} \tag{4.19}$$

$$||\mathbf{V}(t)|| \leq V_{max} \tag{4.20}$$

$$T_{min} \leq ||\mathbf{T}(t)|| \leq T_{max} \tag{4.21}$$

$$\mathbf{r}(0) = \mathbf{r}_0 \quad \mathbf{V}(0) = \mathbf{V}_0 \tag{4.22}$$

$$\mathbf{r}(t_f) = \mathbf{r}_f \quad \mathbf{V}(t_f) = \mathbf{V}_f \tag{4.23}$$

When inspecting Problem 1, it can be established that the objective function in Equation (4.16) is convex, given that it contains the Euclidean norm of the thrust vector, which is a strictly convex function on its respective argument. This is, however, in contrast to Equation (4.21), which contains a lower- and upper-limit constrain on the Euclidean norm of the thrust vector. Per definition, this constitutes a non-convex set as not every point can be seen by every other point along an unobstructed straight path between them. Therefore, Equation (4.21) represents a non-convex constraint. Moreover, this also applies to Equation (4.19), due to the lower-limit constrain implemented on the Euclidean norm of the position vector. Therefore, also Equation (4.19) represents a non-convex constraint. All other constraint functions are, however, convex. The upper-limit constrained Euclidean norm of the velocity vector, as given in Equation (4.20), represents a convex set and therefore constitutes a convex constraint. Furthermore, the equations of translational motion, as given by Equations (4.17) and (4.18), are linear time-invariant equations which are affine and therefore convex. Lastly, also the equality constraints providing the boundary conditions of the mission segment, as given by Equations (4.22) and (4.23), constitute affine constraints and are thus convex. Hence, overall, Problem 1 is defined as a non-convex

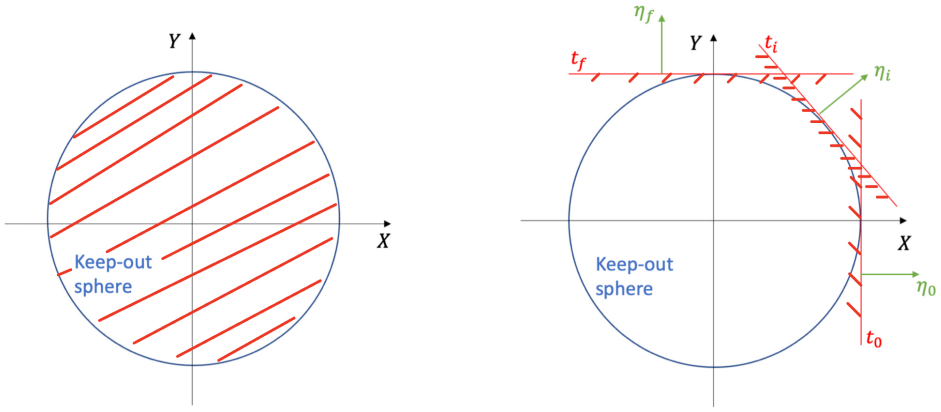


Figure 4.8: Conventional keep-out sphere constraint represented by hatched area (left) and rotating-hyperplane constraint represented by hatched half-spaces (right)

optimisation problem. To solve this problem efficiently, the non-convexities will have to be convexified, after which the problem can be discretised with respect to time and subsequently solved using the SOCP solver.

To convexify Problem 1, both Equations (4.19) and (4.21) have to be adjusted. First, a convexification of the keep-out sphere constraint, as provided by Equation (4.19), is performed using the rotating-hyperplane method, as proposed by Leomanni et al. (2022). In this method, instead of enforcing a spherical keep-out zone during reference trajectory calculation, a rotating plane is proposed that confines the design space of the guidance system to a moving half-space defined at each point along the reference trajectory. A graphical representation of this rotating-hyperplane methodology is provided in Figure 4.8, where on the left side, a conventional keep-out sphere constraint is provided, whereas the right side depicts the rotating-hyperplane method. Note that for intuitive interpretation, the rotating-hyperplane method is applied to a two-dimensional reference scenario, but the methodology is evidently not limited to two-dimensional problems. Furthermore note that in this figure, three instances of the rotating-hyperplane are provided and each one is activated separately at its corresponding node in the time array, rather than jointly. However, for the rotating-hyperplane method to mimic a keep-out sphere constraint, the dividing plane must rotate along with the position of the spacecraft in its reference trajectory. Nevertheless, the hyperplane rotation has to be defined prior to the guidance system calculating a reference trajectory. Therefore, it will directly steer the reference trajectory that can be obtained by the spacecraft.

The rotating-hyperplane method can mathematically be described using the convex constraint definition provided in Equation (4.24). In this equation, $\boldsymbol{\eta}(t)$ represents the time history of the outward unit normal of the separating plane that defines the rotating half-space. However, for each mission segment, the unit normal time history is defined differently, as to optimally steer the reference trajectory based on the characteristics of each mission segment. Therefore, more details on the convexified keep-out sphere constraint will be provided in Subsections 4.5.1 to 4.5.4.

$$\boldsymbol{\eta}^{T}(t)\mathbf{r}(t) \geq r_{KOS} \tag{4.24}$$

Second, to convexify the thrust magnitude constraint, it must be noted that, because omnidirectional thrust is available for the chaser spacecraft, the vector constraint given in Equation (4.21) can be decomposed into three scalar constraints. These scalar constraints are given in Equations (4.25) to (4.27) and, due to the scalar nature of these equations, the Euclidean norm vector of the original constraint has been replaced by the absolute sign operator. Furthermore, it should be noted that this transformation has been made possible by the particular configuration of the reaction control thruster system, as given in Figure 6.1. This is because the X, Y- and Z-components of the thrust vector history calculated by the guidance system will, per definition, each have four reaction control thrusters available for the execution of their commands. Therefore, the three scalar components of the thrust vector can have a magnitude that lies between 0 and four times the nominal operational thrust level of the reaction control thrusters, denoted by T_{nom}. Thus, the constraints provided in Equations (4.25) to (4.27) can be applied and as they each represent a linear set, the corresponding redefinition is convex.

$$T_{min} \leq |\mathbf{T}_x(t)| \leq T_{max} \tag{4.25}$$

$$T_{min} \leq |\mathbf{T}_y(t)| \leq T_{max} \tag{4.26}$$

$$T_{min} \leq |\mathbf{T}_z(t)| \leq T_{max} \tag{4.27}$$

With the objective and constraint functions redefined in convex form, the convex optimisation problem, as provided by Problem 2, can subsequently be discretised with respect to time and solved by the SOCP algorithm. Note that more details on the time discretisation are provided at the end of this section. Furthermore note that in Problem 2, Equation (4.31) provides the rotating-hyperplane definition of the keep-out sphere constraint, for which details will be provided below on a case-by-case basis. Moreover, Equations (4.33) to (4.35) provide the scalar redefinitions of the non-convex thrust norm constraint. Lastly, all other equations are identical to their definitions provided above in Problem 1.

Problem 2:

$$\underset{\mathbf{T}(t),\mathbf{r}(t),\mathbf{V}(t)}{\text{minimise}} \quad J_f = \int_0^{t_f} ||\mathbf{T}(t)||dt \quad \text{subject to} \tag{4.28}$$

$$\dot{\mathbf{r}}(t) = \mathbf{V}(t) \tag{4.29}$$

$$\dot{\mathbf{V}}(t) = \begin{bmatrix} 0 & 2n & 0 \\ -2n & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \mathbf{V}(t) + \begin{bmatrix} 3n^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -n^2 \end{bmatrix} \cdot \mathbf{r}(t) + \frac{\mathbf{T}(t)}{m} \tag{4.30}$$

$$\boldsymbol{\eta}^T(t)\mathbf{r}(t) \geq r_{KOS} \tag{4.31}$$

$$||\mathbf{V}(t)|| \leq V_{max} \tag{4.32}$$

$$0 \leq |\mathbf{T}_x(t)| \leq 4T_{nom} \tag{4.33}$$

$$0 \leq |\mathbf{T}_y(t)| \leq 4T_{nom} \tag{4.34}$$

$$0 \leq |\mathbf{T}_z(t)| \leq 4T_{nom} \tag{4.35}$$

$$\mathbf{r}(0) = \mathbf{r}_0 \quad \mathbf{V}(0) = \mathbf{V}_0 \tag{4.36}$$

$$\mathbf{r}(t_f) = \mathbf{r}_f \quad \mathbf{V}(t_f) = \mathbf{V}_f \tag{4.37}$$

4.5.1. Transfer Guidance

For the transfer guidance system, dealing with the transfer of the chaser spacecraft from the negative V-bar direction to the positive H-bar direction, the time history of the outward normal defining the rotating-hyperplane can be obtained from a spline interpolation of the normals at the initial and final position, as given by $\boldsymbol{\eta}_0$ and $\boldsymbol{\eta}_f$ in Equations (4.38) and (4.39). To obtain a smooth rotation that favours the transition from the initial to the final position, a spline interpolation of uniformly spaced nodes is performed using Matlab's embedded cubic spline function (Matlab, 2022a). Note that the interpolation variable is the node in the time array of the considered mission segment, whereas the interpolated variable is the outward normal vector of the intersecting hyperplane. A mathematical representation of this spline interpolation is provided in Equation (4.40), where spline() represents the embedded cubic spline function of Matlab. Hence by extending the general problem definition, provided in Problem 2, with Equations (4.38) to (4.40), the convex algorithm developed for the transfer phase can be fully defined.

$$\boldsymbol{\eta}_0 = \frac{\mathbf{r}_0}{||\mathbf{r}_0||} \tag{4.38}$$

$$\boldsymbol{\eta}_f = \frac{\mathbf{r}_f}{||\mathbf{r}_f||} \tag{4.39}$$

$$\boldsymbol{\eta}(t) = \text{spline}\left([0, t_f], [\boldsymbol{\eta}_0, \boldsymbol{\eta}_f], t\right) \tag{4.40}$$

4.5.2. Approach Guidance

For the approach guidance system, dealing with the approach phase along the H-bar direction, the time history of the outward normal vector can be defined using Equation (4.41). In this equation, the $\boldsymbol{\eta}$ vector is defined as a constant vector pointing along the positive H-bar. This is because the current mission segment performs an straight-line forced-motion over the H-bar, such that no rotation is needed for the hyperplane. Rather, the separating plane can be defined as stationary. Hence, when extending the general problem definition with Equation (4.41), the convex algorithm developed for the approach phase can be fully characterised.

$$\boldsymbol{\eta}(t) = (0, 0, 1)^T \tag{4.41}$$

4.5.3. Synchronisation Guidance

For the synchronisation guidance system, dealing with the phase that ensures position synchronisation with the docking-axis of the target spacecraft, the time history of the outward normals can also be obtained from a spline interpolation of the normals at the initial and final position, as previously discussed for the transfer phase. Therefore, a mathematical representation of this outward normal vector history is not repeated, and can be found in Equations (4.38) to (4.40). Note that this constraint definition is identical to the transfer phase, given that the current mission segment also favours a smooth rotation from the initial to the final position. Therefore, the exact same basic guidance problem is solved, and follows from expectations when considering the similar trajectory requirements that are imposed on both phases, where the former performs a transfer from negative V-bar to positive H-bar, whereas the latter performs a transfer from the positive H-bar to the instantaneous docking-axis.

4.5.4. Final Approach Guidance

For the final approach guidance system, which deals with the phase that approaches the target spacecraft along its docking-axis up to the capture position, the time history of the outward normal vector can be described using Equation (4.42). In this equation, $\mathbf{C}_{\mathbf{L/B,T}}(t)$ represents the time-varying transformation matrix from the $\mathcal{F}_{B,T}$ frame, in which the docking-axis of the target is defined, to the $\mathcal{F}_L$ frame, in which the guidance problem is defined. Note that this transformation is time-varying given that the target spacecraft is tumbling while traversing its orbit. Furthermore in this equation, $\mathbf{e}_d$ represents a unit vector along the docking-axis of the target spacecraft, which is expressed in the $\mathcal{F}_{B,T}$ frame, and defines the approach direction and final capture position. Note that this specific definition for the outward normal vector is chosen because, during the final approach phase, the tumbling motion of the target docking-axis must be used to steer the direction of the reference trajectory. Therefore, effectively, this rotating-hyperplane definition ensures that the trajectory crudely follows the docking-axis while simultaneously enforcing a keep-out constraint.

$$\boldsymbol{\eta}(t) = \frac{\mathbf{C}_{\mathbf{L/B,T}}(t) \cdot \mathbf{e}_d}{\|\mathbf{C}_{\mathbf{L/B,T}}(t) \cdot \mathbf{e}_d\|} \tag{4.42}$$

However, to ensure a more accurate tracking of the docking-axis, an additional path constraint is imposed on the final approach guidance problem. This additional constraint is given in terms of an approach cone constraint and follows from requirement **REQ_M_6.1**, stating that the reference trajectory of the chaser is confined to be within the approach cone with predefined half angle. A mathematical description of this approach cone constraint is provided in Equation (4.43), where α represents the half-angle defining the cone geometry. Note that for this constraint equation, the approach cone represents a second-order convex constraint and therefore does not require further convexification.

$$\|\mathbf{r}(t)\| \cos\alpha \leq \left(\mathbf{C}_{\mathbf{L/B,T}}(t) \cdot \mathbf{e}_d\right)^T \cdot \mathbf{r}(t) \tag{4.43}$$

Furthermore, an additional keep-in sphere constraint is imposed by the final approach guidance system to limit the magnitude of the position vector, as given by Equation (4.44). In this equation, the maximum allowed radial position of the calculated trajectory is provided by r_{max}. This keep-in sphere constraint is defined for this terminal phase to ensure a forced-motion manoeuvre towards the docking position, such that the maximum radial position is constrained by the magnitude of the initial position, $\|\mathbf{r}_0\|$. This ensures a reference trajectory that is directly along the direction of the capture

position, instead of a combined outward-inward manoeuvre. However, in case a navigation and/or control system error causes the initial position to lie within the keep-out sphere region, the maximum allowed position magnitude is modified to 5 m. This is done to allow the spacecraft to leave the keep-out sphere and increase the inter-satellite distance, but not further than the base plane of the approach cone, as given by **REQ_M_6.2**. Therefore, a check on the magnitude of the initial position vector is performed using a simple if-condition, as given in Equation (4.45). Note, however, that identical to the maximum velocity constraint provided in Problem 2, this new constraint is readily provided in convex form. Hence a full characterisation of the final approach guidance system can be obtained by extending the general problem definition, provided in Problem 2, with Equations (4.42) to (4.45).

$$||\mathbf{r}(t)|| \leq r_{max} \tag{4.44}$$

$$\begin{cases} r_{max} = ||\mathbf{r}_0|| & \text{if} \quad ||\mathbf{r}_0|| > r_{KOS} \\ r_{max} = 5 & \text{if} \quad ||\mathbf{r}_0|| \leq r_{KOS} \end{cases} \tag{4.45}$$

The mission parameters that define the previously discussed optimisation problems are given in Table 4.1. Note that this summary table is not limited to the problem parameters of the final approach phase. Rather, it contains the necessary details to all mathematical optimisations provided in Subsections 4.5.1 to 4.5.4. In this table, the general mission parameters related to mean anomaly and mass are taken from mission and system requirements, **REQ_M_2** and **REQ_S_C_2**, respectively. Moreover, the maximum approach velocity requirement is imposed based on **REQ_M_6.5** and the nominal thrust level is defined by the performance parameters of the reaction control thrusters, as provided in **REQ_S_C_13.2**. Lastly, the operational frequency of the guidance algorithm is set at 1 Hz, as common for guidance systems. This frequency is chosen as it preforms best in a preliminary trade-off assessing controllability over the chaser's reference trajectory and required computational effort. However, a detailed analysis on the selection of the optimal operational frequency is left for a subsequent study. Nevertheless, it means that every second, a new command is generated by the guidance system and therefore the navigation and control systems should run at an equal or higher operational frequency.

Furthermore, when considering the phase-specific mission parameters, it can be seen that most settings follow from the defined mission scenario, as elaborated in Section 4.1. For the transfer, approach and synchronisation phases, a keep-out sphere with a radius of 5 m is applied, whereas for the final approach phase, this radius is reduced to 1.35 m. The former value follows from safety requirements, given that higher thrust values are expected during far-range rendezvous, when compared to close-range rendezvous. Therefore, a more conservative safety region is applied. This is in contrast to the last phase, during which close-range rendezvous is established. For this terminal phase, the safety region is reduced and follows from the capture position with respect to the target, as given in **REQ_M_7**. Moreover, this final phase requires a definition for the docking-axis and the half-angle of the approach cone, which are provided by **REQ_M_6.1** to **REQ_M_6.3**, respectively. Lastly, it can be observed that for all mission segments, a definition of final conditions is provided. These definitions are taken from the mission scenario elaborated in Section 4.1. However, for the first phase, which is the transfer phase, also initial conditions are predefined and follow from the mission scenario in Section 4.1. This is in contrast to the subsequent mission segments, for which no initial conditions are defined as they follow from the previous mission segment. In an ideal scenario, the initial conditions are identical to the final conditions of the preceding phase. In reality, however, these values are given by the current best estimate of the navigation system and therefore might differ due to errors in the GNC system of the chaser spacecraft.

The set of optimisation problems provided until now all represent continuous time problems, which cannot be solved numerically by a computer. Instead, the problems must be discretised with respect to time. To find the solution of Problem 2, the thrust vector $\mathbf{T}(t)$, position vector $\mathbf{r}(t)$ and velocity vector $\mathbf{V}(t)$ of the chaser spacecraft must be discretised using the definitions provided in Equations (4.46) to (4.48). In these equations, i represents the index of the epochs in the discretised time array and N represents the total number of epochs in this time array.

$$\mathbf{T}_i = \mathbf{T}(t_i) \quad \text{for} \quad i = 1, \ldots, N \tag{4.46}$$

Table 4.1: Summary table containing mission parameters for guidance system

General Parameters			
n [deg/s]	0.063	V_{max} [m/s]	1
m [kg]	500	T_{nom} [N]	55
Transfer Parameters		operational frequency [Hz]	1
r_{KOS} [m]	5	**Approach Parameters**	
$(\mathbf{r}_0, \mathbf{V}_0)$ [m,m/s] $(\mathbf{r}_f, \mathbf{V}_f)$ [m,m/s]	$(0,-20,0,0,0,0)^T$ $(0,0,20,0,0,0)^T$	r_{KOS} [m]	5
Final Approach Parameters		$(\mathbf{r}_f, \mathbf{V}_f)$ [m,m/s]	$(0,0,5,0,0,0)^T$
r_{KOS} [m]	1.35	**Synchronisation Parameters**	
$\mathbf{e}_d$ [−]	$(-1,0,0)^T$	r_{KOS} [m]	5
α [deg]	30		
$(\mathbf{r}_f, \mathbf{V}_f)$ [m,m/s]	$(r_{KOS}\mathbf{C}_{\mathbf{L/B,T}}(t_f)\mathbf{e}_d, 0,0,0)^T$	$(\mathbf{r}_f, \mathbf{V}_f)$ [m,m/s]	$(r_{KOS}\mathbf{C}_{\mathbf{L/B,T}}(t_f)\mathbf{e}_d, 0,0,0)^T$

$$\mathbf{r}_i = \mathbf{r}(t_i) \quad \text{for} \quad i = 1, ..., N \tag{4.47}$$

$$\mathbf{V}_i = \mathbf{V}(t_i) \quad \text{for} \quad i = 1, ..., N \tag{4.48}$$

Subsequently, also the equations of translational motion implemented in the guidance problem need to be discretised with respect to time. To do so, a fourth order Runge-Kutta scheme is employed, which represents a single-step explicit numerical integration scheme. Furthermore, when considering the thrust vector, a zero-order hold discretisation is applied. Consequently, throughout one time step of the discretised dynamics, the value of the thrust vector remains constant. This implies that the objective function can be rewritten to discretised form as provided in Equation (4.49). In this equation, the objective function is minimised by manipulating the optimisation vectors $\mathbf{T}(i)$, $\mathbf{r}(i)$ and $\mathbf{V}(i)$, which represent the discretised thrust, position and velocity vectors, respectively.

$$\underset{\mathbf{T}(i),\mathbf{r}(i),\mathbf{V}(i)}{\text{minimise}} \quad J_f = \sum_{i=1}^{N} ||\mathbf{T}(i)|| \tag{4.49}$$

By using this discretisation scheme, a sparse optimisation problem can be defined, which benefits the solution time of the guidance system (Bhagat, 2016).

4.6. Tracking Guidance

As the optimal reference trajectory is calculated by the convex guidance system, which applies the simplifying assumptions of the Hill equations in its propagator, a discrepancy will persist between the calculated trajectory and the actual trajectory flown by the spacecraft. Furthermore, dispersions between this guidance trajectory and the one known onboard the spacecraft are also caused by non-ideal actuators and navigation system errors. Therefore, a trajectory error will exist between the reference of the guidance system and the path measured by the navigation system. Hence to guide the spacecraft back to the required trajectory, a simple tracking guidance system is developed. By doing so, the errors between the desired state and the estimated state can effectively be eliminated. Following the work performed by Bhagat (2016), who developed a convex guidance algorithm for the e.Deorbit mission, it was found that a simple PID controller with manually tuned gains can provide sufficient path-tracking during an ADR scenario. Therefore, it has been decided to develop a similar tracking controller.

The PID controller represents a straightforward feedback controller that calculates the required control action based on the value of the current state error, the integrated state error and the state error derivative. A mathematical description of this concept is provided in Equation (4.50), where $\mathbf{T}(t)$ represents the thrust command needed to remove the trajectory dispersion, K_p the proportional gain corresponding to the trajectory error $\mathbf{e}(t)$, K_i the integral gain corresponding to the integrated trajectory error $\int \mathbf{e}(t)dt$ and K_d the derivative gain corresponding to the trajectory error derivative $\dot{\mathbf{e}}(t)$. To calculate this trajectory error, the discrepancy between the ideal trajectory of the guidance system and the current position estimate of the navigation system is used.

Table 4.2: PID tracking guidance settings

Parameter	Value
K_p [-]	50
K_i [-]	0
K_d [-]	200
operational frequency [Hz]	1

$$\mathbf{T}(t) = K_p\mathbf{e}(t) + K_i \int \mathbf{e}(t)dt + K_d\dot{\mathbf{e}}(t) \tag{4.50}$$

It should be noted that, due to the simplicity of the control law provided in Equation (4.50), it cannot be guaranteed that optimal control is achieved. Nevertheless, the controller is found to be sufficiently effective to guide the chaser spacecraft back to its reference trajectory for the e.Deorbit mission in (Bhagat, 2016) and is therefore expected to do the same for the ClearSpace-1 mission. Its performance parameters are as defined in Table 4.2, from which it can be seen that a considerable proportional gain is used. This is done because a sufficiently high gain is needed to calculate the necessary burns that can track the reference trajectory in between feasible recalculations of the convex guidance system, as well as calculating corrective burns when the convex guidance system fails to find a solution. This is in contrast to the integral gain, which is set to zero because, when using a variable reference signal, the integral controller will have difficulties in removing the steady-state error, thereby saturating the control actuators. Furthermore, a high differential gain is implemented and used to remove oscillations in the system response. This is required as preliminary analyses on the guidance system indicated that it is very sensitive to changes in the thrust vector. Lastly, it can be seen that the operational frequency of the tracking guidance system is set at 1 Hz, given that the control allocation and pulse-width modulation algorithm of the reaction control thrusters cannot process commands faster than this.

With the gains and frequency defined in Table 4.2, the chaser spacecraft can provide efficient trajectory control while simultaneously requiring limited control effort. Note, however, that due to time constraints, the provided controller gains are not optimally tuned and their respective values are established from preliminary analysis on the time response of the system to a step input. Nevertheless, it is found that the selected gains result in sufficient performance for the first analyses performed in the current thesis study.

4.7. Golden Section Search

All convex guidance problems provided in Section 4.5 represent finite horizon optimisation problems. This implies that the convex solver will look for the optimal trajectory that reaches the terminal conditions at the terminal time, while adhering to all path constraints. Therefore, for different flight times, the resulting trajectories will be different and also their corresponding optimality will be different. Moreover, for some flight times it even occurs that no feasible solution is available. Hence, to obtain overall optimality, it is clear that the optimal terminal time will have to be sought, where in this case optimal is used to indicate the value for which the resulting trajectory is feasible and requires minimum propellant usage.

To find this optimal time of flight, the golden section search methodology is employed as proposed by Malyuta et al. (2021), given that it is known to work well for objective functions that are unimodal with respect to time. This search procedure is a numerical optimisation technique that locates the optimal function value within a predefined interval, through progressive reduction of the considered interval width in the direction of the optimum. Therefore, when applying it to the considered reference mission, a minimum and maximum time of flight need to be provided and the optimal time of flight needs to be located within these boundaries. However, due to time constraints, an own golden section search algorithm has not been developed fur the current thesis study. Rather, the intrinsic Matlab function called fminbnd() has been used. This function can be described mathematically using Equation (4.51), where t_f represents the time of flight, J_f the objective function, t_{min} the lower bound of the optimisation interval and t_{max} the upper bound of the optimisation interval.

$$\underset{t_f}{\text{minimise}} \quad J_f = f(t_f) \quad \text{such that} \quad t_{min} \leq t_f \leq t_{max} \tag{4.51}$$

Table 4.3: Golden section search algorithm settings

Parameter	Value
t_{min} [sec]	0
t_{max} [sec]	300
ε [sec]	0.5

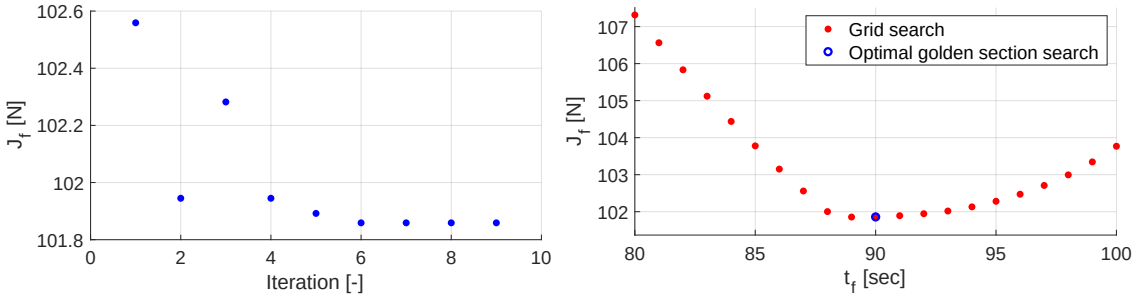


Figure 4.9: Objective value J_f as a function of the iteration performed by the golden section search algorithm (left) and objective value J_f as a function of time of flight t_f performed by a grid search algorithm (right) for final approach phase, defined on the interval $t_f \in [80, 100]$

Furthermore, to define the convergence criterion that terminates the execution of the golden section algorithm, a threshold on the minimum step size needs to be defined. When the solver tries to reduce its step size below the threshold value, represented by the parameter ε, the golden section search terminates. For all guidance problems of the current reference scenario, the algorithm definition is provided in Table 4.3. In this table, the minimum time of flight is set at 0 seconds and the maximum time of flight is set at 300 seconds. This initial interval width is chosen as it is sufficiently wide to contain the optimal time of flight for a large number of configurations of each guidance problem, as well as not being too wide to still allow for sufficiently rapid convergence. Moreover, the convergence threshold is set at 0.5 seconds, given that the optimal time of flight obtained from the algorithm will get rounded to the nearest second, for convenience in subsequent use of this parameter in the guidance system. For brevity purposes, however, no further details on the algorithm are provided and the reader is referred to (Forsythe et al., 1976) for more information concerning the golden section search algorithm in general and (Matlab, 2022b) for its Matlab implementation.

Applying the golden section algorithm to find the optimal time of flight for the final approach phase is provided in Figure 4.9. Note that for computational efficiency, the search domain used in the current figure has been reduced with respect to Table 4.3, and is defined by $t_{min} = 80$ seconds and $t_{max} = 100$ seconds. Moreover, the value of ε remains unchanged. When inspecting the left side of Figure 4.9, the iterations performed by the search algorithm are presented and for each iteration, the corresponding objective value is provided. It can be seen from this figure that, from approximately the sixth iteration, the algorithm converges to a solution within the search accuracy. For the current thesis study, this search accuracy is given by possible step size in time of flight and is set at 1 second, as defined above. Furthermore, the right side of Figure 4.9 provides the objective value as a function of the time of flight. These values are obtained from a grid search on the final approach phase and its corresponding values are provided in red. Moreover, this figure also provides the optimal time of flight obtained from a golden section search in blue and it can be observed that the algorithm converges to one of the two near-equivalent optimal objective values, given at $t_f = 89$ and $t_f = 90$. Hence, it can be concluded that the real optimal time of flight is located in between these objective values but falls outside the employed grid. Therefore, it can be assumed that the golden section algorithm is able to provide an optimal flight time within the accuracy specified for the current mission.

4.8. Verification

With the mission parameters defined in Table 4.1, the convex guidance system calculates the reference trajectory given in Figure 4.10, where the left side of the figure represents the trajectory in the Hill frame of the target, whereas the right side represents it in the body frame of the target. To obtain this reference

trajectory, the simulation parameters are taken from Table 3.1, with the only difference that the angular velocity of the target is set to $(3.5, 3.5, 3.5)^T$ deg/s. Furthermore, during initialisation, no disturbance forces and/or torques are implemented, and the parameters that define these disturbances for both spacecraft are consequently set to zero. Note that, due to the aspect ratio of the plots, the keep-out sphere constraints can look a bit morphed. Moreover, their projection on the base planes of the figures is omitted as it might create the illusion of a violation of the keep-out sphere constraints when provided in a two-dimensional representation. Hence, this should be avoided as the constraints are nevertheless accurately imposed in three dimensions. Furthermore, it should be noted that similar to the mission scenario discussion provided in Section 4.1, the transfer and approach phases are conveniently expressed in the Hill frame, whereas the synchronisation and final approach phases are best expressed with respect to the body frame of the target. This is in contrast to the attitude synchronisation phase, which is provided in both reference frames to indicate the transition between both. Moreover, it shows that a stationary point in Hill frame does not correspond to a stationary point in the body frame and vice versa. Lastly, it should be noted that as the approach cone is defined in the body frame of the target spacecraft, and as this frame rotates with respect to the Hill frame, no clear image of the approach cone constraint can be provided on the left side of the figure and is therefore omitted.

When inspecting Figure 4.10, it can be seen that a successfull reference trajectory is provided that transfer from the negative V-bar to the positive H-bar, approaches along the H-bar up to the first keep-out sphere constraint and maintains this hold-point for attitude synchronisation. Subsequently, the trajectory synchronises its position with the docking-axis of the target, represented by the dotted black line, after which it approaches the target within the approach cone up to the second keep-out sphere. To ensure the calculated trajectory does not violate any path constraints, a summary graph containing the position and velocity vector constraints is provided in Figure 4.11. On the left of the figure, the position vector constraints are provided in terms of the keep-out sphere and keep-in sphere constraints. Note that, a keep-out sphere is imposed during the entire time of flight whereas a keep-in sphere is only imposed during the final approach phase. Furthermore note that the dotted lines in this figure indicate a switch of the respective constraint value. From this figure it is evident that the entire reference trajectory calculated by the guidance system adheres to all position vector constraints that are imposed on it. Moreover, the right side of Figure 4.11 provides the constraint on the maximum velocity magnitude that is enforced throughout the entire trajectory. From this figure it can be determined that the velocity of chaser remains well within this limit and that, at the end of each mission segment, the velocity magnitude reduces back to zero as imposed by the boundary conditions of Table 4.1.

The thrust vectors that are calculated by the guidance system to fly these trajectories are provided in Figure 4.12, where the magnitude is given for the three components of the thrust vector along with their accompanying constraint. From this figure it can be concluded that the thrust magnitude adheres to the constraint applied on it throughout the entire duration of the reference mission. Furthermore, it can be seen that for the first two manoeuvres, the thrust commands are impulsive. This can be explained from the relative orbital motion in the Hill frame, for which the most fuel efficient manoeuvres are provided in terms of impulsive ones, given that these two initial phases represent the transfer phase to the H-bar and the approach phase along the H-bar, respectively. Moreover, it can be seen that for the transfer phase, most impulse is provided along the V- and H-bar direction and this impulse is equally split over both directions, whereas for the approach phase only an impulse is provided along the H-bar and both observations are to be expected when considering the performed manoeuvres. For the subsequent attitude synchronisation phase, where a hold-point along the H-bar is tracked for a short time duration, no thrust is commanded by the convex guidance system, as this phase is executed by the tracking guidance system instead. Therefore, it is omitted in the current analysis. Nevertheless it is assumed that hold position tracking is executed accurately by maintaining a fixed position on the H-bar during this mission phase. Lastly, when considering the final two phases, which deal with tracking the tumbling docking-axis, a more complex thrust history of low thrust values is expected. This can also be seen from the asymmetric nature of these phases, in contrast to the symmetric nature observed for the previous phases. Moreover, no impulsive manoeuvres are expected for these mission segments. Hence, it can be concluded that the guidance system provides a predictable command history for the thrust actuators. Additionally, when assuming a thruster system with common characteristics, *i.e.*, a specific impulse of I_{sp} =270 seconds and gravitational acceleration of g_0 =9.807 m/s^2, a consumed propellant mass of 0.56 kg results. On a total vehicle mass of 500 kg, this value is only marginal and the validity of omitting the mass dynamics throughout the mission scenario is confirmed.

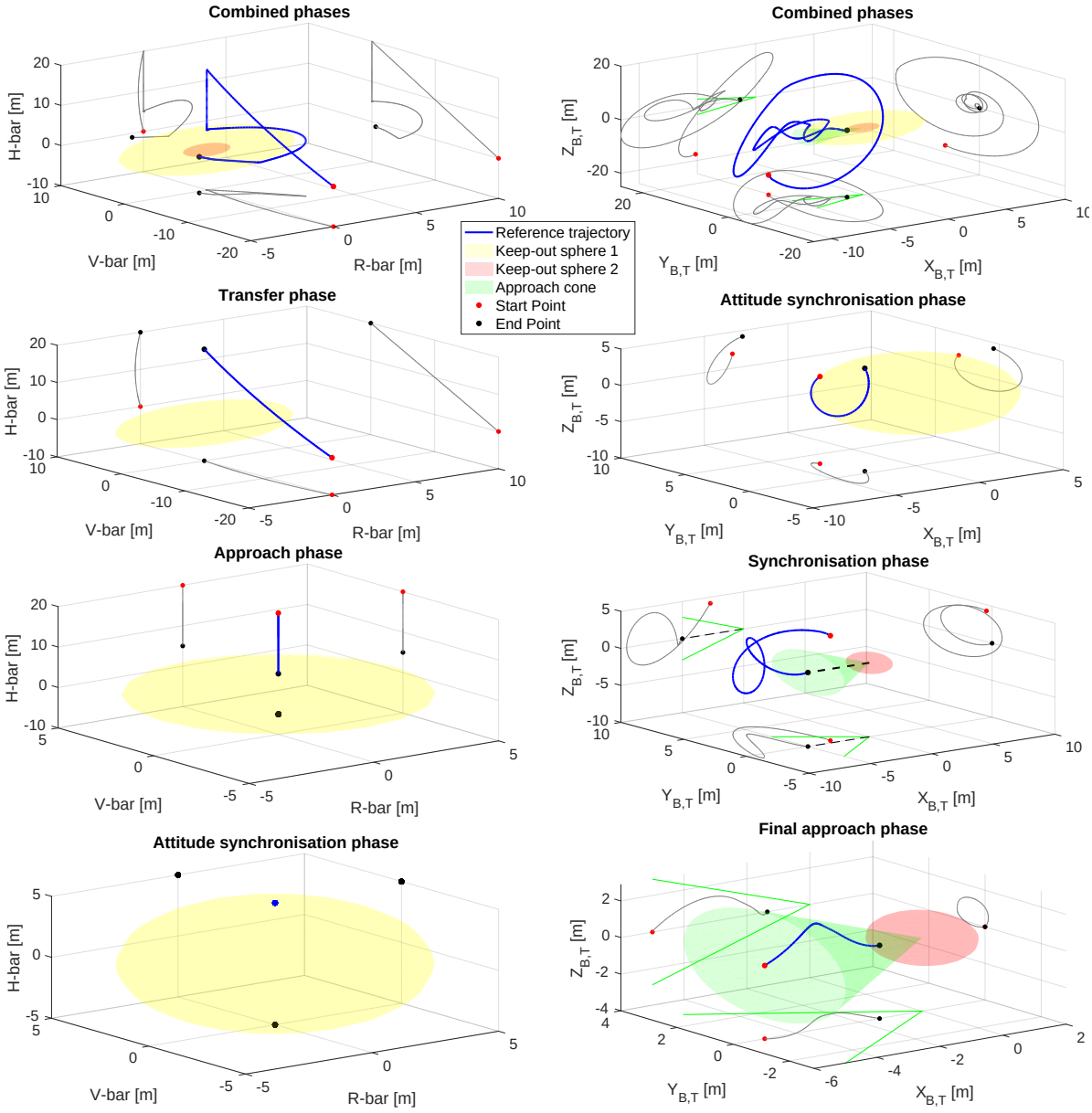


Figure 4.10: Reference trajectory provided by the guidance system, as expressed in the Hill frame of the target spacecraft (left) and its Body-Centred-Body-Fixed frame (right)

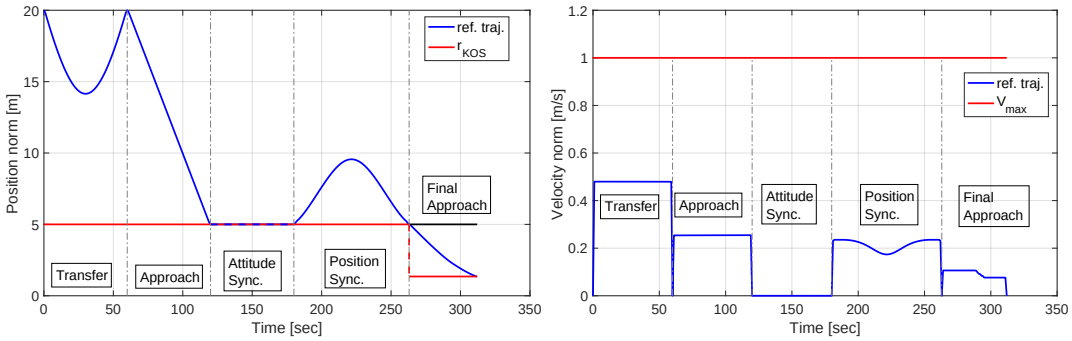


Figure 4.11: Position vector constraints (left) and velocity vector constraints (right) imposed on the reference trajectory calculated by the guidance system and referenced with respect to the Hill frame of the target spacecraft

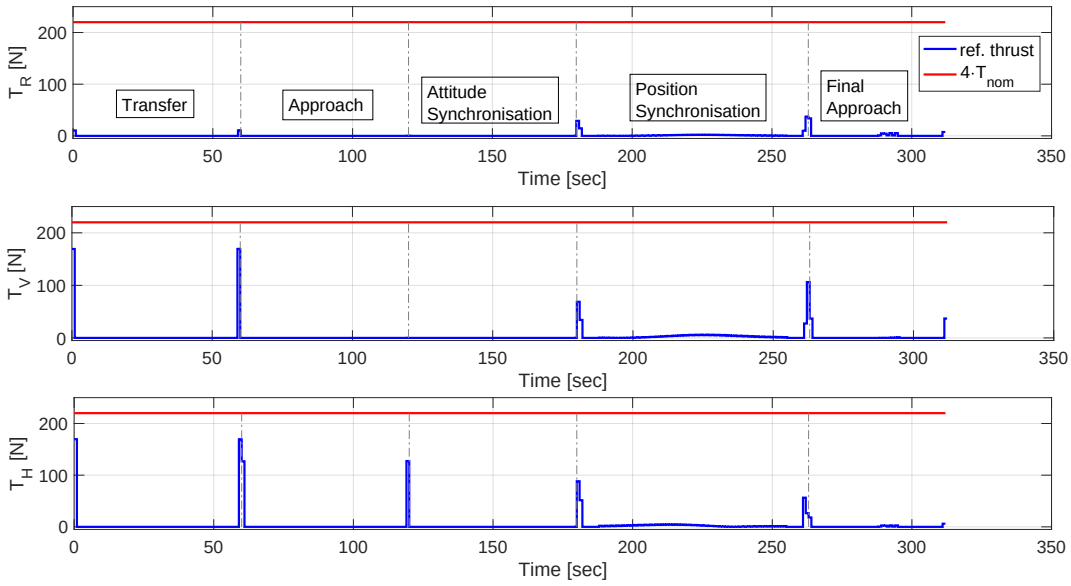


Figure 4.12: Absolute value of thrust vector components calculated by the guidance system with accompanying constraint, referenced with respect to the Hill frame of the target spacecraft

To verify the optimality of the reference trajectory provided by the guidance algorithm, a Monte Carlo analysis is performed in which the thrust command profile of the convex guidance system is varied. For all solutions that can reach the terminal conditions, the propellant consumption is compared with the guidance solution. Note that the propellant system characteristic as discussed above are reused. Furthermore note that no further path constraints are imposed on these variation trajectories and that the terminal position is allowed to be within a radius of 0.5 m to the capture position, as to obtain a conservative check on the optimality of the guidance result. This is ensured, given that some trajectories will actually be infeasible results for the physical guidance problem, but are still included in the Monte Carlo analysis. To perform this analysis, 5000 random samples are drawn for each thrust component in the thrust vector from a normally distributed probability function. To sample this normal distribution, Matlab's randn() function is used. The procedure is iterated using four different seeds to define the random number generator, which are 17, 6, 19 and 99, and are used to obtain conclusive results. Moreover, also four different variations on the normal distribution are employed, where each distribution is defined with a zero mean and a standard deviation of 0.01, 0.05, 0.1 and 0.25 N, respectively. These deviations are low, given that it was uncovered during preliminary analyses of the guidance system that it is very sensitive to changes in the thrust vector. Even the slightest error in thrust magnitude or direction might render the trajectory infeasible. Therefore, the standard deviations as provided above are implemented.

The results of the Monte Carlo analysis for seed 17 are provided in Figure 4.13. Note that for other seeds, no results are provided given that comparable results are achieved. In the figure, the horizontal axis presents the number of feasible solutions, whereas the vertical axis provides the corresponding fuel consumption for each feasible solution. It can be observed that the optimal solution, *i.e.*, the reference trajectory with lowest propellant consumption, is indeed presented by the result of the convex guidance system, given by the single blue dot in the lower left corner. Furthermore, it can be identified that, with increasing standard deviation of the thrust variation, the number of feasible trajectories reduces and the propellant consumption increases. Hence, it can be concluded that with standard deviations larger than those implemented, a higher fuel consumption will result. Therefore, it can be concluded from this discussion that the convex guidance algorithm is indeed implemented correctly.

Furthermore, also the tracking guidance system and its integration with the convex guidance system are verified. To achieve this, first the time response of the system to a step input is analysed, as provided in Figure 4.14. The left side of the figure presents the system response for three combinations of proportional and derivative gains, K_p and K_d, while the right side of the figure provides their corresponding control input. Therefore, this verification step also serves as a check on the gains selected in Table 4.2. Note that the step input is defined as a repositioning manoeuvre along the R-bar, where

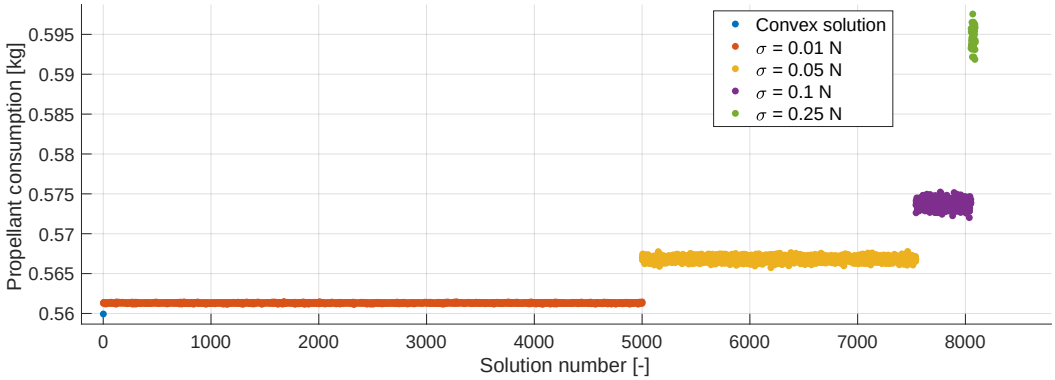


Figure 4.13: Monte Carlo analysis of solution optimality provided in terms of propellant consumption for reference trajectory calculation performed by convex guidance system. Note that the Monte Carlo analysis is performed using 5000 samples and a seed equal to 17

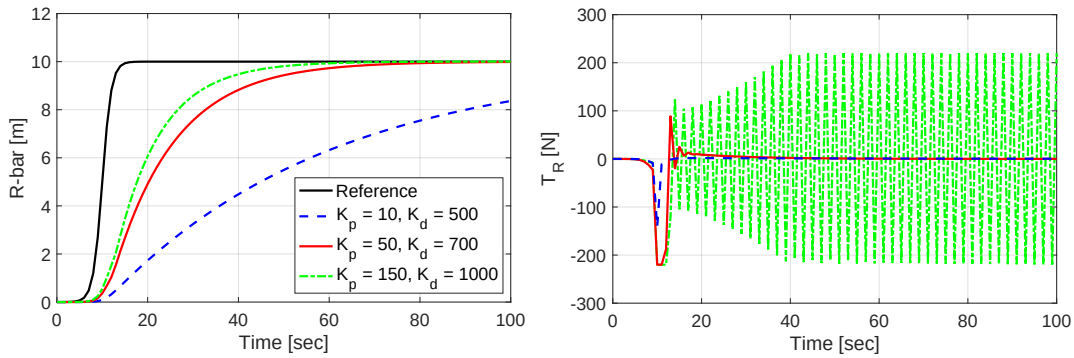


Figure 4.14: Sigmoid response of tracking guidance system for a reorientation manoeuvre along the R-bar direction (left) and corresponding control commands (right) for three different gain combinations, referenced with respect to the Hill frame of the target spacecraft

the chaser spacecraft has to increase its relative position from 0 m to 10 m. Therefore both the system response as well as the control command are provided along the R-bar direction of the Hill frame. Furthermore, note that instead of a true step input, a sigmoid function is used as it represents a more realistic control input when compared to the non-smooth step function.

From Figure 4.14 it can be seen that, for the gains selected in Table 4.2 and as represented by the red curve, an adequate system response is achieved using a reasonable control input. Nevertheless, the system is rather slow, requiring a settling time of approximately 70 seconds. However, increasing the gains to speed up this system response, as represented by the green curve, results in a very unstable control input requirement. To remove the oscillations in the control signal, one would generally further increase the derivative gain. However, doing so would subsequently reduce the speed of the response, thereby defying the purpose of the entire alteration. Moreover, it is found that, even when increasing the derivative gain further, the required control input remains unstable, which indicates that the high proportional gain is saturating the control system. Additionally, also a short actuator saturation can be seen in the initial response of the red curve. However, removing this saturation by reducing both gains, as represented by the blue curve, results in a system that is too slow for effective control. Despite saturation being undesirable, the actuator that will be dealing with this control command is represented by the non-throttled reaction control thruster system. Hence, it can provide either zero thrust or maximum thrust and should therefore have no problems during the execution of this command. Accordingly, the author is confident that the system characteristics provided in Table 4.2 result in a good balance between trajectory control and control effort. Moreover, the provided discussion indicates that the tracking controller is capable of effectively changing the position of the chaser spacecraft, such that it can be assumed to be implemented correctly.

Moreover, the integration of the tracking guidance system with the convex system is verified. To do so, a simulation of the combined guidance system is performed using the simulation parameters

given in Table 3.1. Note that, as before, the angular velocity of the target is altered with respect to Table 3.1 and set as $(3.5, 3.5, 3.5)^T$ deg/s. However, in contrast to the initialisation of the convex guidance system, the current test includes the influence of disturbances during its simulation. These disturbances are incorporated to induce trajectory dispersions that can be corrected for by the tracking system. Therefore, for both spacecraft, the parameters defining disturbance forces and moments are taken from Table 3.1.

The results of the analysis are presented in Figure 4.15, where the left side presents a comparison between the reference trajectory of the initial solution, indicated by the green curve, and the simulation trajectories with and without tracking guidance, indicated by the blue and red curves, respectively. Furthermore, the right side of Figure 4.15 provides the corresponding trajectory errors of the simulation trajectories with respect to the reference trajectory. Note that in this figure, the trajectories and trajectory errors are referenced with respect to the Hill frame of the target. From this figure it can be seen that the simulation trajectory including tracking guidance is capable of closely following the reference, resulting in a near-zero trajectory error throughout the entire mission duration, whereas the trajectory without tracking guidance starts to disperse from the reference during the attitude synchronisation phase. This is because disturbances cause the chaser to drift away from hold point S_3, rendering the convex guidance system incapable of calculating the commands during the subsequent position synchronisation and final approach phases.

Moreover, it should be noted that for both simulation trajectories, the convex guidance system performs frequent recalculations of its reference trajectory, using the current Hill state of the chaser as initial condition. These recalculations are performed at an update rate of 0.2 Hz and, in case an infeasible solution is encountered, the convex guidance system will no longer send thrust commands to its actuators. Instead, it will simply output the reference trajectory obtained from the last feasible recalculation step. This is done such that the tracking system can guide the chaser back to the reference trajectory, after which a subsequent recalculation of the convex system should be capable of finding a new feasible solution.

For the simulation without tracking guidance, as provided by the red curve in Figure 4.15, the convex guidance system is unable to find feasible solutions during the position synchronisation and final approach phases. As explained before, this is caused by excessive trajectory dispersions in the preceding attitude synchronisation phase. Therefore, the trajectories flown in these last three phases can be characterised as free drift trajectories starting from a hold point on the H-bar. Note that, during the simulation epoch of these free drift trajectories, the chaser spacecraft is positioned within Earth's shadow, such that the dominant perturbing forces are given by aerodynamic drag and the spherical harmonics of Earth. The free drift dispersion along the R-bar direction can be explained by the effect of aerodynamic drag, as it operates approximately along the negative V-bar direction of the chaser spacecraft. By doing so, the orbital velocity of the chaser gets reduced and with that, its altitude is reduced because the original orbit can no longer be sustained. Therefore, a dispersion along the negative R-bar can be observed. Note that due to the short time scale of Figure 4.15, only a small reduction in V-bar position can be observed from the reduction of the orbital velocity. However, when increasing the simulation time, it can be seen that, at a lower orbital altitude, the chaser will catch up again with its target along the V-bar direction, as its lower orbit is characterised by a larger mean motion with respect to the target orbit. Furthermore, the Hill equations also indicate the decoupled nature of the motion along the H-bar direction. For the current simulation, this H-bar motion is induced by the disturbance force of Earth's spherical harmonics. Nevertheless, despite not including formal collision avoidance manoeuvres, the free drift trajectories disperse away from the target, rather than converge towards it.

The thrust commands of the combined guidance system can be found in Figure 4.16. When comparing this command history to Figure 4.12, it can be seen that during all phases, various additional burns are executed. Furthermore, it can be seen that this combined control history mostly provides additional burns during the position synchronisation and final approach phases, which is to be expected due to the complex nature of their respective trajectories. However, given that the complete guidance system combines recalculations of the the convex guidance system with the tracking guidance system, the additional burns cannot solely be attributed to intermediate corrections calculated by the tracking system. Nevertheless, the control input remains bounded, has no extended periods of actuator saturation and follows the expectations, as explained for the thrust vector history of the convex guidance initialisation. Furthermore the updated propellant mass consumption equals 0.827 kg, which maintains its insignificant status with respect to the total vehicle mass. Hence, it can be concluded that the track-

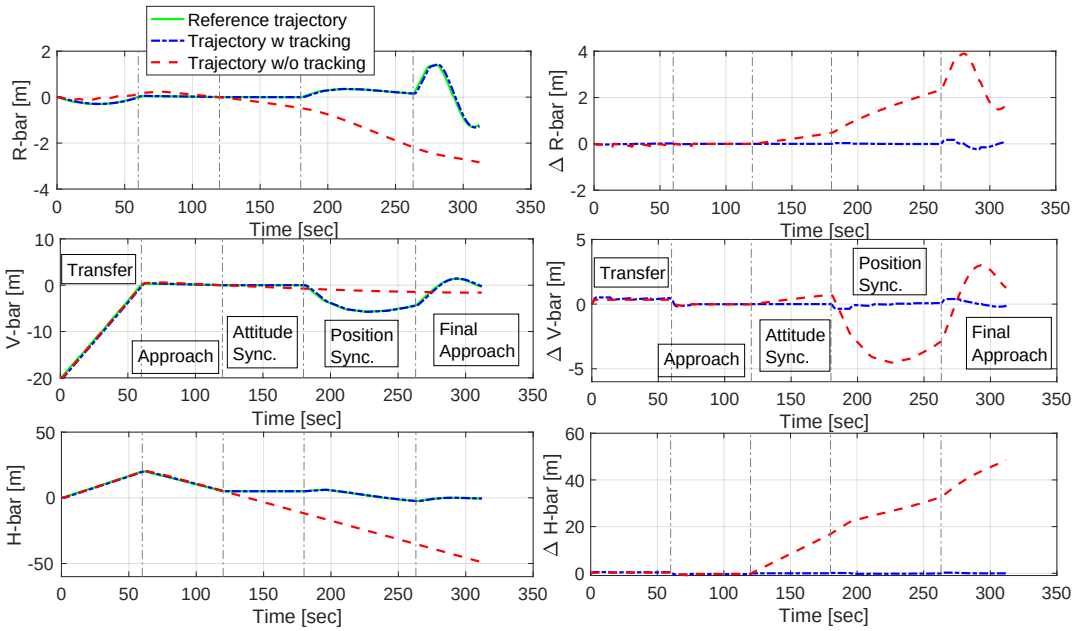


Figure 4.15: Comparison of the reference trajectory calculated during initialisation of the convex guidance system with simulation trajectories with and without tracking guidance system (left) and corresponding trajectory errors of the simulation trajectories with respect to the reference trajectory (right), referenced with respect to the Hill frame of the target spacecraft

ing guidance system is correctly implemented to perform position tracking at hold point S_3, as well as providing intermediate corrective burns that maintain proximity with the reference trajectory of the convex guidance system. Furthermore, it can be concluded that the integrated guidance system is verified and ready for implementation in the GNC system.

Note, however, that the presented discussion does not provide a conclusive analysis on the gain tuning of the tracking guidance system. Rather, it provides a first-order feeling for the response that can be expected by the system. Nevertheless, the gains obtained through this analysis are used, without further optimisation, as good results could be obtained when using an isolated tracking guidance system, as shown in Figure 4.14, as well as when using it in combination with the convex guidance system, as shown in Figure 4.15. It is, however, recommended for a future design iteration to perform proper gain tuning of this tracking system.

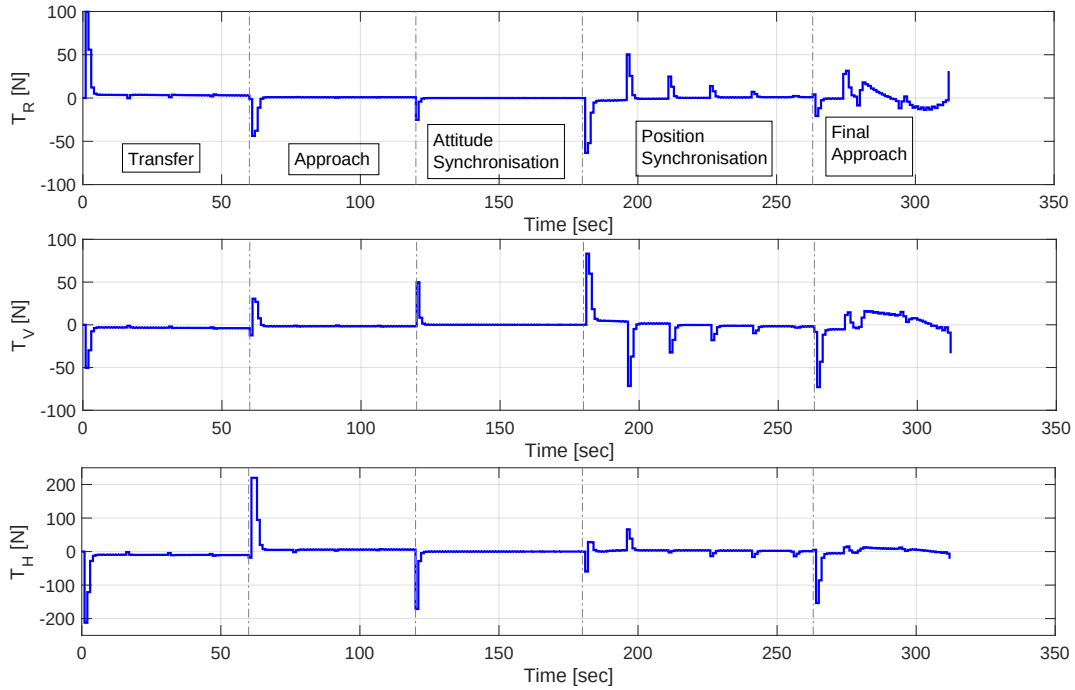


Figure 4.16: Thrust command history of the combined guidance system, referenced with respect to the Hill frame of the target spacecraft

5
Navigation

The current chapter provides information regarding the navigation function of the GNC subsystem. For the ClearSpace-1 mission, this navigation system is required to perform two simultaneous tasks. On one hand, the inertial state of the ClearSpace-1 spacecraft is to be estimated using an inertial navigation system, as discussed in Section 5.1. On the other hand, the relative state of the ClearSpace-1 spacecraft with respect to the VESPA target is to be estimated using a vision-based navigation system, as discussed in Section 5.2. An implementation of these two navigation systems for the considered mission is provided in Section 5.3, whereas Section 5.4 discusses the verification procedures followed to ensure their correct functionality.

5.1. Inertial Navigation

To obtain information about the inertial state of the chaser spacecraft, sensory measurements are taken. However, the raw data of these measurements are likely to contain measurement errors and sensor noise. Furthermore, various sensors might provide contradicting measurements of the same quantity. Therefore, to effectively process these sensory measurements, a state estimation algorithm is implemented in the inertial navigation system. Details concerning the implemented inertial navigation system are provided below, where Subsection 5.1.1 provides information on the implemented inertial navigation sensors, whereas Subsection 5.1.2 details the implemented state estimation algorithm.

5.1.1. Inertial Navigation Sensors

The inertial navigation sensors onboard the ClearSpace-1 satellite consist, as a minimum, of: an IMU, a GPS receiver and two star sensors. The following section will provide a short elaboration on the working principle and type of measurements that are provided by these sensors. Note that, for convenience, the measurement components of these sensors are referenced with respect to the inertial $\mathcal{F}_I$ frame.

Inertial Measurement Unit

An Inertial Measurement Unit (IMU) consists of three accelerometers, capable of measuring the non-gravitational acceleration of the body, and three gyroscopes, capable of measuring the angular rates of the body. Note, however, that the accelerometers cannot measure the entire acceleration acting on the body. Therefore, the gravitational acceleration has to be calculated and artificially added in the state estimation algorithm. Nevertheless, by measuring the angular rates and translational accelerations acting on the spacecraft, the IMU can provide attitude and position information in three dimensions. When starting from an initial state, integration of the gyroscopic and accelerometer measurements allow to obtain the current position, velocity and attitude of the spacecraft. A mathematical description of the state measured by the IMU sensor, $\mathbf{x}_{IMU}$, is provided in Equation (5.1), where it should be noted that $\ddot{x}$, $\ddot{y}$ and $\ddot{z}$ represent the Cartesian acceleration components whereas ω_1, ω_2 and ω_3 represent the angular velocity components around the body-axes of the spacecraft, as measured by the IMU sensor.

$$\mathbf{x}_{IMU} = \left(\ddot{x}, \ddot{y}, \ddot{z}, \omega_1, \omega_2, \omega_3\right)^T \tag{5.1}$$

The IMU sensor model implemented in the GRADS library employs a simplified representation of the accelerometers and gyroscopes, which should suffice for all practical purposes of the navigation filter considered in the current thesis study. This simplified representation assumes that biases, misalignments and scaling factors are the main sources of measurement errors. Hence the accelerations provided by the IMU, $\ddot{x}_{IMU}$, $\ddot{y}_{IMU}$ and $\ddot{z}_{IMU}$, can be modelled as given in Equation (5.2). In this equation,

the b-terms represent the biases, the s-terms represent the scaling factors and the m-terms represent the misalignment factors of the accelerometer measurements. Furthermore, **a** represents the true acceleration vector acting on the chaser spacecraft, whereas **g** represents the gravitational acceleration vector on the chaser spacecraft. Lastly, v_x, v_y and v_z represent random white noise parameters in the accelerometer measurements. Note that a random white noise parameter is defined to have a Gaussian normal distribution with zero mean and a specific standard deviation. Furthermore note that this definition of white noise is valid for all other inertial navigation sensors, unless specifically stated otherwise.

$$\begin{pmatrix} \ddot{x}_{IMU} \\ \ddot{y}_{IMU} \\ \ddot{z}_{IMU} \end{pmatrix} = \begin{pmatrix} b_{IMU,x} \\ b_{IMU,y} \\ b_{IMU,z} \end{pmatrix} + \begin{bmatrix} 1+s_{IMU,x} & m_{IMU,xy} & m_{IMU,xz} \\ m_{IMU,yx} & 1+s_{IMU,y} & m_{IMU,yz} \\ m_{IMU,zx} & m_{IMU,zy} & 1+s_{IMU,z} \end{bmatrix} \cdot (\mathbf{a}-\mathbf{g}) + \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} \tag{5.2}$$

The bias factors have a given magnitude when the IMU sensor is turned on. However, throughout operation of the IMU sensor, the bias factors also tend to follow a random walk. To model the random walk process as a differential equation, Equation (5.3) can be used. In this equation $v_{b,x}$, $v_{b,y}$ and $v_{b,z}$ are normally distributed white noise parameters, indicating that the trajectory consists of sequential random steps, *i.e.*, a random walk.

$$\begin{pmatrix} \dot{b}_{IMU,x} \\ \dot{b}_{IMU,y} \\ \dot{b}_{IMU,z} \end{pmatrix} = \begin{pmatrix} v_{b,x} \\ v_{b,y} \\ v_{b,z} \end{pmatrix} \tag{5.3}$$

Similarly, the angular rate components provided by the IMU, $\omega_{IMU,1}$, $\omega_{IMU,2}$ and $\omega_{IMU,3}$, can be modelled as given in Equation (5.4). In this equation, the b-terms represent the biases, the s-terms represent the scaling factors and the m-terms represent the misalignment factors of the gyroscopic measurements. Furthermore, $\boldsymbol{\omega}$ represents the true angular velocity vector around the body-fixed axes of the chaser spacecraft. Lastly, v_{ω_1}, v_{ω_2} and v_{ω_3} represent random white noise parameters contained in the gyroscope measurements.

$$\begin{pmatrix} \omega_{IMU,1} \\ \omega_{IMU,2} \\ \omega_{IMU,3} \end{pmatrix} = \begin{pmatrix} b_{IMU,\omega_1} \\ b_{IMU,\omega_2} \\ b_{IMU,\omega_3} \end{pmatrix} + \begin{bmatrix} 1+s_{IMU,\omega_1} & m_{IMU,\omega_1\omega_2} & m_{IMU,\omega_1\omega_3} \\ m_{IMU,\omega_2\omega_1} & 1+s_{IMU,\omega_2} & m_{IMU,\omega_2\omega_3} \\ m_{IMU,\omega_3\omega_1} & m_{IMU,\omega_3\omega_2} & 1+s_{IMU,\omega_3} \end{bmatrix} \cdot \boldsymbol{\omega} + \begin{pmatrix} v_{\omega_1} \\ v_{\omega_2} \\ v_{\omega_3} \end{pmatrix} \tag{5.4}$$

The bias factors of the gyroscope are modelled once more using a random walk process, as given in Equation (5.5). In this equation v_{b,ω_1}, v_{b,ω_2} and v_{b,ω_3} are normally distributed white noise parameters.

$$\begin{pmatrix} \dot{b}_{IMU,\omega_1} \\ \dot{b}_{IMU,\omega_2} \\ \dot{b}_{IMU,\omega_3} \end{pmatrix} = \begin{pmatrix} v_{b,\omega_1} \\ v_{b,\omega_2} \\ v_{b,\omega_3} \end{pmatrix} \tag{5.5}$$

Global Positioning System

A GPS receiver uses the Global Positioning System (GPS) Network developed by the U.S. military to provide navigation measurements. This network consists of 24 to 32 satellites flying in a medium Earth orbit constellation, which continuously emit signals containing their location and time at the instant of transmission. The GPS receiver receives these signals and from the time difference between the transmission and reception of the signal, the receiver can figure out its range with respect to the GPS satellite. When signals of at least three GPS satellites can be combined, the position of the receiver can be triangulated. However, this generally does not result in accurate position determination, as the clock onboard the receiver is not very accurate. This is a huge issue when considering that the positioning system is solely based on measured time differences. Hence, to increase the accuracy, the signals of at least four GPS satellites are used to simultaneously estimate the receiver's clock error. Furthermore, the velocity of the receiver can be determined from the Doppler shift present in the GPS signals. Note that since the ClearSpace-1 mission contains a spaceborne GPS receiver, signal attenuation caused by the atmosphere is not applicable. Therefore, the receiver clock error is the only considerable error model that has to be taken into account, in addition to white noise errors. A mathematical description of the state measured by the GPS receiver, $\mathbf{x}_{GPS}$, is provided in Equation (5.6). In this equation ρ_1, ρ_2, ρ_3

and ρ_4 represent the four range measurements, whereas $\dot{\rho}_1$, $\dot{\rho}_2$, $\dot{\rho}_3$ and $\dot{\rho}_4$ represent the four range rate measurements, as provided by the GPS receiver. Note that in case a satellite is equipped with multiple GPS receivers, the difference in phase angle of the received signal can also be used to determine the attitude of the receiving spacecraft. This is, however, currently omitted for the ClearSpace-1 spacecraft.

$$\mathbf{x}_{GPS} = \left(\rho_1, \rho_2, \rho_3, \rho_4, \dot{\rho}_1, \dot{\rho}_2, \dot{\rho}_3, \dot{\rho}_4\right)^T \tag{5.6}$$

A mathematical description of the range measurement model is provided in Equation (5.7). In this equation, $\mathbf{r}_{GPS,i}$ represents the Cartesian position vector of the i^{th} GPS satellite, $\mathbf{r}_C$ the Cartesian position vector of the chaser spacecraft, c the speed of light, $b_{GPS,clock}$ the bias of the receiver's clock and v_{ρ_i} the white noise in the range measurement. When range measurements to four satellites are available, the position of the receiver as well as it's clock bias can be determined. However, note that the Cartesian position of the i^{th} GPS satellite, $\mathbf{r}_{GPS,i}$, is referenced with respect to the Earth-Centred-Earth-Fixed frame. Therefore, to estimate the inertial position of the receiver from range measurements, it is necessary that the GPS satellite position vectors are provided in the inertial $\mathcal{F}_I$ frame. This is however a straightforward transformation and is therefore omitted from the current thesis report.

$$\rho_i = \sqrt{\left(\mathbf{r_{GPS,i}} - \mathbf{r}_C\right)^T \left(\mathbf{r_{GPS,i}} - \mathbf{r}_C\right)} + c \cdot b_{GPS,clock} + v_{\rho_i} \tag{5.7}$$

Furthermore, range rate measurements can be modelled using Equation (5.8), where the receiver clock drift, as well as some white measurement noise are considered. In this equation, $\mathbf{V}_{GPS,i}$ represents the Cartesian velocity vector of the i^{th} satellite, $\mathbf{V}_C$ the Cartesian velocity vector of the chaser spacecraft, c the speed of light, $d_{GPS,clock}$ the drift of the receiver's clock and $v_{\dot{\rho}_i}$ the signal's white noise. Note also here that the position and velocity of the i^{th} GPS satellite, $\mathbf{r}_{GPS,i}$ and $\mathbf{V}_{GPS,i}$, are referenced with respect to the Earth-Centred-Earth-Fixed frame. Therefore, to estimate the inertial velocity of the receiver from range rate measurements, it is necessary that the GPS satellite position and velocity vectors are transformed to the inertial $\mathcal{F}_I$ frame. However, for brevity purposes, this transformation is currently omitted.

$$\dot{\rho}_i = \frac{(\mathbf{r}_{GPS,i} - \mathbf{r}_C)^T (\mathbf{V}_{GPS,i} - \mathbf{V}_C)}{\sqrt{(\mathbf{r}_{GPS,i} - \mathbf{r}_C)^T (\mathbf{r}_{GPS,i} - \mathbf{r}_C)}} + c \cdot d_{GPS,clock} + v_{\dot{\rho}_i} \tag{5.8}$$

It should be noted that the GPS receiver model implemented in the GRADS library performs a simultaneous simulation of all GPS satellites in their respective orbits. Therefore, more than four satellites may be in view of the GPS receiver. When this happens, the receiver model can select a combination of GPS satellites that minimises the Geometric Dilution of Precision (GDOP) in the measurements. Note that this GDOP value quantifies how errors in the measurements affect the precision of a subsequent state estimation. In the particular case of GPS measurements, GDOP refers to the effect the distribution of GPS satellites has on the positioning accuracy of the receiver. A graphical representation of this phenomenon is provided in Figure 5.1, from which it can be seen that the denser the GPS satellites are distributed over space, the greater the uncertainty is in the positioning accuracy of the receiver, as represented by the grey area. Therefore, when more than four satellites are simultaneously in view of the receiver, the GPS sensor of the ClearSpace-1 mission automatically selects the combination of four satellites that minimises the GDOP value and thus maximises the state estimation accuracy. Details on how the model calculates the GDOP value can be found in (Mooij, 2021a).

Star Sensor

A star sensor performs autonomous recognition of stars and star patterns in the sky and subsequently compares the direction vectors to these recognised star patterns with a star catalog onboard the sensor's computer. Through this procedure, the star sensor can obtain an estimate of its orientation with respect to inertial space. Therefore it is common practise for star sensors to output the spacecraft orientation using attitude quaternions referenced with respect to an inertial frame. A mathematical description of the state measured by the star sensor, $\mathbf{x}_{SS}$, is provided in Equation (5.9), where it should be noted that q_1 , q_2, q_3 and q_4 represent the quaternion components measured by the star sensor.

$$\mathbf{x}_{SS} = \left(q_1, q_2, q_3, q_4\right)^T \tag{5.9}$$

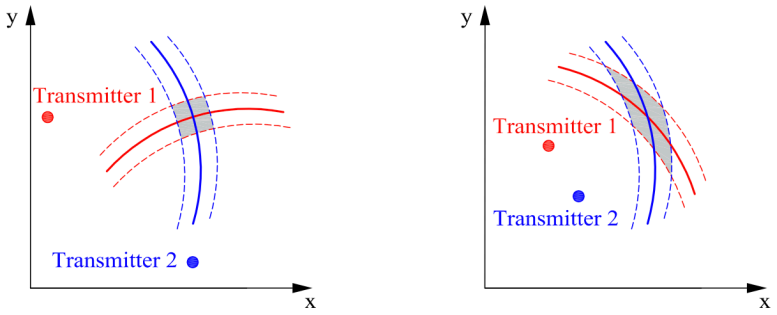


Figure 5.1: Geometric Dilution of Precision: effect of inter-satellite positioning of GPS satellites on position accuracy of a GPS receiver as indicated by the grey area, taken from (Mooij, 2021a)

The star sensor model selected from the GRADS library only takes white measurement noise into account. Hence the quaternion vector provided by the star sensor, $\mathbf{q}_{SS}$, can be modelled as given in Equation (5.10). Note that in this equation, $\mathbf{q}$ represents the actual orientation of the spacecraft with respect to the inertial frame. Furthermore, $\mathbf{v}_q$ represents the random white noise vector that appears in the star sensor measurements. It should be noted that to satisfy the unit-length constraint imposed on attitude quaternions, the summation of the true attitude quaternion and the white noise vector is divided by its norm. Furthermore, it should be noted that the white noise vector, $\mathbf{v}_q$, contains a distinct white noise value for each component in the quaternion vector, as given by Equation (5.11). Lastly, note that for star sensors operating at high angular velocities, the star patterns captured by the images become streaks and are therefore difficult to recognise. Evidently, this degrades the accuracy of the attitude measurements or might even render them impossible. However, due to advances in the optical sensors used for star camera systems, patterns can be recognised at spin rates up to 20 deg/s (Katake et al., 2008). Therefore, for the current mission scenario, it can be assumed that attitude measurements remain available from the star sensor, given the limited angular velocity of the target spacecraft with which the chaser spacecraft has to synchronise.

$$\mathbf{q}_{SS} = \frac{\mathbf{q} + \mathbf{v}_q}{||\mathbf{q} + \mathbf{v}_q||} \tag{5.10}$$

$$\mathbf{v}_q = \begin{pmatrix} v_{q_1} \\ v_{q_2} \\ v_{q_3} \\ v_{q_4} \end{pmatrix} \tag{5.11}$$

5.1.2. Unscented Kalman Filter

The sensors provided in the previous section could be prone to large measurement errors, as given in their sensor models. Hence, to improve the performance of the navigation system, data fusion is applied that combines the measurements of various sensors with a mathematical model of the systems' dynamics in a filtering algorithm. Due to the intrinsic nonlinear nature of orbital dynamics and spacecraft attitude dynamics, linear filtering techniques will be ineffective. Therefore, a nonlinear filter will be employed. Moreover, as it was initially decided in Chapters 3 to propagate attitude kinematics in terms of MRP components, their resulting time series will be discontinuous each time a conversion between MRP and SMRP parameters is made. This implies that these singular points cannot be estimated by more conventional filtering techniques that linearise the system dynamics based on Taylor-series expansions, like performed in the Extended Kalman Filter. For this type of partial-derivative dependent filters, the local derivative will be undefined at the singular points, rendering them ineffective. Therefore the derivative-free Unscented Kalman Filter (UKF) is selected for implementation onboard the ClearSpace-1 mission and detailed in the current section. It should be noted that, to avoid excessive referencing, general information on the UKF is taken from (Haykin, 2001). Furthermore, it should be noted that the UKF was implemented before the switch from MRP to quaternion representation was established. Therefore, even though a partial-derivative filter could have been used, the UKF filter was simply adjusted to quaternion representation and resulted in an equally effective estimation algorithm.

The UKF is part of a new class of nonlinear estimation filters called Sigma-Point Kalman Filters that do not linearise the system dynamics before propagating the state and covariance (Ahmadi et al., 2012). Instead, they propagate a deterministically selected cloud of weighted points around the mean through the nonlinear system equations, from which a new state and covariance can be estimated. The UKF applies a specific sampling technique, referred to as the unscented transform, which picks a minimal set of sample points around the mean, the so-called sigma points, and assigns their respective weights before propagating these through the nonlinear system equations (Ahmadi et al., 2012). This implies that no computationally intense partial derivatives have to be calculated nor that the filter results in poor performance when the modelled system is highly nonlinear. Moreover, the filter can estimate states for singular points in the system dynamics where the partial derivatives are not defined. Lastly, it is noted by Julier and Uhlmann (2004) that the basic UKF requires a computational cost that is of the same order of magnitude as the Extended Kalman Filter. Hence, from this argumentation it is clear why the UKF is selected for the considered reference mission.

The state estimation principle of the UKF applies an a-priori state and covariance estimate obtained through propagation of the system model, known onboard the navigation filter. Subsequently, the UKF corrects these state and covariance estimates based on performed measurements. By applying these prediction and correction steps sequentially, a more accurate a-posteriori estimate of the spacecraft state can be obtained. The framework of the UKF is centred around the estimation of a state for a discrete-time nonlinear system, as given by Equations (5.12) and (5.13). In Equation (5.12), $\mathbf{x}_{k+1}$ represents the state vector of the system at time step $k+1$, $\mathbf{x}_k$ the state vector of the system at time k, $\mathbf{u}_k$ the control input vector at time k and $\mathbf{v}_k$ the process noise vector at time k. Furthermore, $f()$ represents the nonlinear vector function mapping the state, control input and process noise vectors at time k to the future state vector at time $k+1$. In Equation (5.13), $\mathbf{y}_{k+1}$ represents the measurement vector at time $k+1$, $\mathbf{x}_{k+1}$ is once again the state vector of the system at time $k+1$ and $\mathbf{w}_{k+1}$ the observation noise vector of the system at time $k+1$. Furthermore, $g()$ represents the nonlinear vector function mapping the state and observation noise vectors at time $k+1$ to the measurement vector at time $k+1$.

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k) \tag{5.12}$$

$$\mathbf{y}_{k+1} = g(\mathbf{x}_{k+1}, \mathbf{w}_{k+1}) \tag{5.13}$$

To propagate the state and covariance estimates, a deterministically selected cloud of weighted points around the mean are propagated through the nonlinear system equations. Subsequently, from this propagated cloud of sigma points, a new state and covariance estimate is calculated. Note that these sigma points are carefully selected through use of the unscented transform, which is a method to calculate the statistics of a random variable undergoing a nonlinear transformation (Julier and Uhlmann, 1997). The $2L+1$ sigma-point vectors, where L represents the number of states in the state vector $\mathbf{x}$, can be calculated using Equations (5.14) to (5.16).

$$\boldsymbol{\chi}_0^k = \hat{\mathbf{x}}_k \tag{5.14}$$

$$\boldsymbol{\chi}_i^k = \hat{\mathbf{x}}_k + \left(\sqrt{(L+\lambda)}\sqrt{\hat{\mathbf{P}}_{x_i}^k}\right) \qquad \text{for } i = 1, ..., L \tag{5.15}$$

$$\boldsymbol{\chi}_i^k = \hat{\mathbf{x}}_k - \left(\sqrt{(L+\lambda)}\sqrt{\hat{\mathbf{P}}_{x_{i-L}}^k}\right) \qquad \text{for } i = L, ..., 2L \tag{5.16}$$

Note that in these equations, the sigma-point vectors $\boldsymbol{\chi}_i^k$ are calculated based on the current best estimate of state vector mean, given by $\hat{\mathbf{x}}_k$, and the i^{th} row of the matrix square root of the current best estimate for the state covariance matrix, given by $\sqrt{\hat{\mathbf{P}}_{x_i}^k}$. With these sigma-point vectors, a $L \times (2L+1)$ matrix $\boldsymbol{\chi}$ can be formed with corresponding weight factors, $W_i{}^m$ and $W_i{}^c$, as given by Equations (5.17) to (5.19).

$$W_0{}^m = \frac{\lambda}{L+\lambda} \tag{5.17}$$

$$W_0{}^c = \frac{\lambda}{L+\lambda} + \left(1 - \alpha^2 + \beta\right) \tag{5.18}$$

$$W_i{}^m = W_i{}^c = \frac{1}{2\,(L+\lambda)} \qquad \text{for } i = 1, \ldots, 2L \tag{5.19}$$

In the equations provided above, λ is a scaling parameter defined by Equation (5.20). Furthermore, α is a parameter that determines the spread of the sigma points around the mean and is usually set to a small value, $\mathcal{O}\left(10^{-3}\right)$, κ a secondary scaling parameter which is generally set to 0 and β a parameter used to incorporate knowledge on the distribution of the to-be-estimated parameter $\mathbf{x}$. For Gaussian distributions, an optimal value for β is set at 2.

$$\lambda = \alpha^2(L + \kappa) - L \tag{5.20}$$

With the sigma-point vectors established, a propagation through the nonlinear system dynamics can be performed to obtain an estimate of the sigma-point vectors at the next epoch. The propagation is provided in Equation (5.21), from which an a-priori mean and state covariance matrix estimate, $\bar{\mathbf{x}}_{k+1}$ and $\bar{\mathbf{P}}_x^{k+1}$, respectively, can be calculated using Equations (5.22) and (5.23).

$$\boldsymbol{\chi}_i^{k+1} = f(\boldsymbol{\chi}_i^k, \mathbf{u}_k, \mathbf{v}_k) \qquad \text{for } i = 0, \ldots, 2L \tag{5.21}$$

$$\bar{\mathbf{x}}_{k+1} = \sum_{i=0}^{2L} \left(W_i{}^m \boldsymbol{\chi}_i^{k+1}\right) \tag{5.22}$$

$$\bar{\mathbf{P}}_x^{k+1} = \sum_{i=0}^{2L} W_i{}^c \left(\boldsymbol{\chi}_i^{k+1} - \bar{\mathbf{x}}_{k+1}\right)\left(\boldsymbol{\chi}_i^{k+1} - \bar{\mathbf{x}}_{k+1}\right)^T + \mathbf{Q} \tag{5.23}$$

Note that these equations make use of the weight factors as previously established in Equations (5.17) to (5.19). Moreover, Equation (5.23) implements the constant process noise covariance matrix $\mathbf{Q}$, which generally contains unmodelled system dynamics as well as parameter uncertainties of the dynamic model used by the UKF. Furthermore, the propagated sigma-point vectors are also used to obtain a-priori estimates of the measurement vector and its corresponding covariance matrix, $\bar{\mathbf{y}}_{k+1}$ and $\bar{\mathbf{P}}_y^{k+1}$, respectively. To obtain these parameters, the propagated sigma-point vectors must first be transformed trough the nonlinear measurement equation, as given in Equation (5.24).

$$\boldsymbol{\Upsilon}_i^{k+1} = g\left(\boldsymbol{\chi}_i^{k+1}, \mathbf{w}_{k+1}\right) \qquad \text{for } i = 0, \ldots, 2L \tag{5.24}$$

Subsequently, the a-priori estimates of the measurement vector and its corresponding covariance matrix can be calculated using Equations (5.25) and (5.26), where once again use is made of the previously established weight factors. Moreover, Equation (5.26) implements the constant measurement noise covariance matrix $\mathbf{R}$. Note that the measurement noise covariance matrix generally contains the noise in the sensory measurements used by the UKF.

$$\bar{\mathbf{y}}_{k+1} = \sum_{i=0}^{2L} \left(W_i{}^m \boldsymbol{\Upsilon}_i^{k+1}\right) \tag{5.25}$$

$$\bar{\mathbf{P}}_y^{k+1} = \sum_{i=0}^{2L} W_i{}^c \left(\boldsymbol{\Upsilon}_i^{k+1} - \bar{\mathbf{y}}_{k+1}\right)\left(\boldsymbol{\Upsilon}_i^{k+1} - \bar{\mathbf{y}}_{k+1}\right)^T + \mathbf{R} \tag{5.26}$$

An estimate of the cross covariance matrix can be calculated from these obtained results, as given in Equation (5.27).

$$\mathbf{P}_{xy}^{k+1} = \sum_{i=0}^{2L} W_i{}^c \left(\boldsymbol{\chi}_i^{k+1} - \bar{\mathbf{x}}_{k+1}\right)\left(\boldsymbol{\Upsilon}_i^{k+1} - \bar{\mathbf{y}}_{k+1}\right)^T \tag{5.27}$$

Lastly, the a-priori estimates can be updated through corrections from the performed measurements, given by the true measurement vector $\mathbf{y}_{k+1}$. This step is identical to the correction step performed in a classical Kalman filter. Therefore, the Kalman gain can straightforwardly be calculated from Equation (5.28) and the a-posteriori estimate of the state mean and its corresponding covariance matrix can be calculated from Equations (5.29) and (5.30).

$$\mathbf{K}_{k+1} = \mathbf{P}_{xy}^{k+1} \left(\overline{\mathbf{P}}_{y}^{k+1} \right)^{-1} \tag{5.28}$$

$$\hat{\mathbf{x}}_{k+1} = \overline{\mathbf{x}}_{k+1} + \mathbf{K}_{k+1} \left(\mathbf{y}_{k+1} - \overline{\mathbf{y}}_{k+1} \right) \tag{5.29}$$

$$\hat{\mathbf{P}}_{x}^{k+1} = \overline{\mathbf{P}}_{x}^{k+1} - \mathbf{K}_{k+1} \overline{\mathbf{P}}_{y}^{k+1} \left(\mathbf{K}_{k+1} \right)^{T} \tag{5.30}$$

Using the equations provided above, an estimate of a dynamic system state can be obtained from noisy measurements. These equations constitute the UKF algorithm and a summary of the UKF filtering procedure can be found in Algorithm 1. Note that no partial-derivatives are required for this algorithm and the overall computational complexity is of the same order as an Extended Kalman Filter (Julier and Uhlmann, 2004). Furthermore note that, due to time constraints, the current inertial navigation system implements Matlab's UKF algorithm, as detailed in (Matlab, 2022c). Nevertheless, this implementation follows the exact same theory as provided above, and results in a navigation solution that is identical to that of a self-developed UKF. Therefore, to make efficient use of resources, it was decided to follow Matlab's implementation, rather than developing an own algorithm that is equivalent to it.

Data: Initialise with $\hat{\mathbf{x}}_0$, $\hat{\mathbf{P}}_x^0$, $\mathbf{Q}$, $\mathbf{R}$, α, β and κ;
Calculate the weighting parameters W_i^m and W_i^c using Equations (5.17) and (5.19);
Result: Obtain system state $\hat{\mathbf{x}}_{k+1}$ from noisy measurements $\mathbf{y}_{k+1}$
for *Next Epoch* **do**
  Calculate $2L + 1$ sigma-point vectors $\boldsymbol{\chi}_i^k$ using Equations (5.14) to (5.16);
  Propagate the sigma-point vectors through the nonlinear system dynamics to obtain $\boldsymbol{\chi}_i^{k+1}$ from Equation (5.21);
  Calculate a-priori state and covariance matrix estimates $\overline{\mathbf{x}}_{k+1}$ and $\overline{\mathbf{P}}_x^{k+1}$ using Equations (5.22) and (5.23);
  Propagate the sigma-point vectors through the nonlinear measurement equation to obtain $\boldsymbol{\Upsilon}_i^{k+1}$ from Equation (5.24);
  Calculate measurement and covariance matrix estimates $\overline{\mathbf{y}}_{k+1}$ and $\overline{\mathbf{P}}_y^{k+1}$ using Equations (5.25) and (5.26);
  Calculate cross-covariance matrix estimate $\mathbf{P}_{xy}^{k+1}$ using Equation (5.27);
  Calculate Kalman gain matrix $\mathbf{K}_{k+1}$ using Equation (5.28);
end
Next Epoch becomes Current Epoch
for *Current Epoch* **do**
  Obtain a-posteriori state estimate $\hat{\mathbf{x}}_{k+1}$ using Equation (5.29);
  Obtain a-posteriori covariance estimate $\hat{\mathbf{P}}_x^{k+1}$ using Equation (5.30);
end
Corrected state vector becomes best estimate of state at current epoch
if *Final Epoch Reached* **then**
  stop algorithm;
else
  look at Next Epoch;
end

Algorithm 1: Unscented Kalman filter (UKF)

5.2. Relative Navigation

As discussed in Chapter 2, it has been shown by experiments of the Engineering Test Satellite ETS-VII, the XSS-10 satellite, the DART and Orbital Express spacecraft, the final ATV mission and the RemoveDEBRIS mission that a vision-based navigation system is of fundamental importance when performing autonomous rendezvous and docking between two spacecraft. Therefore, it was decided for the ClearSpace-1 mission that such a vision-based navigation system is to be used to estimate the relative position, velocity, attitude and angular rate between both spacecraft, as provided by **REQ_S_C_10**.

During vision-based navigation, visual cameras, possibly extended with other sensors, such as lidars and radars, are utilised to detect and track an object in the images taken by the camera. To track the position and attitude of the target spacecraft, a 3D model of the target is matched with its images, as was provided in Figure 2.4. To subsequently track the velocity and angular rate of the target, the changes in its orientation and position over consecutive images are calculated. As given by Marchand et al. (2019), a model-based tracking of the spacecraft is basically a local nonlinear optimisation problem in which one tries to determine the orientation and position that minimises the error between the object in the image and the projection of its 3D model onto the image. The procedure is performed in an iterative manner until the error between the image and the projected model is below a specified threshold.

It should, however, be noted that preceding the tracking of an object from image to image, the target first needs to be detected in the initial sequence of images. This detection is performed by comparing the object in the image to a database consisting of numerous attitude and position representations and selecting the position and attitude combination in the database that most closely approximates the current image. After this detection is performed, the subsequent tracking will provide a faster and more accurate result as only small adjustments over two consecutive images are required.

Hence, to obtain information about the relative state of the target spacecraft with respect to the chaser spacecraft, sensory measurements are taken using a visual camera system. Details concerning the implemented relative navigation system are provided below, where Subsection 5.2.1 provides information on the implemented sensor suite, whereas Subsection 5.2.2 details the implemented relative navigation algorithm. Note however that, to obtain the relative position, velocity, attitude and angular rate between both spacecraft, complex image processing has to be performed that is capable of first detecting visual reference points in the images and subsequently matching these with the corresponding reference points on the 3D model. It has been decided by the author, however, that this vision-based navigation falls outside the scope of the current thesis study, as it can only be done justice when performing a research solely based on VBN. This is because the author has no knowledge nor experience concerning such image processing algorithms and is thus deemed unfeasible given the time constraints and general complexity of the research. Therefore, it is assumed that the output of VBN algorithm is readily available in the navigation system in terms of relative state estimates.

This assumption has implications on the performance of the overall navigation system. As relative state estimates are provided artificially, the inertial and relative state estimators must act as separate systems. This means that, to obtain an estimate of the inertial state of the target, the outputs of both navigation systems are added. This, therefore, implies that also the resulting navigation error is a simple addition of the navigation errors of the separate estimator systems. However, due to the strong coupling between the inertial and relative states, the navigation error could be reduced when a simultaneous estimation of both could be performed. This is because, rather than simply adding the respective navigation errors, the integrated navigation filter would reduce their combined error. Therefore, it is expected that this integrated procedure will result in an improved navigation accuracy and, as such, is recommended for a subsequent design iteration of the GNC system.

5.2.1. Relative Navigation Sensors

To implement a vision-based navigation system onboard the ClearSpace-1 spacecraft, it is equipped with optical navigation sensors as given by **REQ_S_C_12**. These optical sensors consist, as a minimum, of a radar, a lidar and a far- and close-range visual camera system. The following section will therefore provide a short elaboration on the working principle and type of measurements that are provided by these sensors.

Radar

The radar sensor operates by transmitting electromagnetic waves and detecting the corresponding echo of these waves, as reflected by an external object. The nature of these reflected echo waves

provides information on the object: the time between emitting and receiving the reflected signal indicates the relative distance to the object and the shift in frequency of the signal due to the Doppler effect indicates the relative velocity with respect to the object. Alternatively, the relative velocity can be determined by analysing the variations of the range measurements over time. Furthermore, when using a directive antenna as part of the radar, the relative angular position of the object with respect to the radar can be determined. When applying this to the current reference mission, the radar onboard the chaser spacecraft can determine the relative distance, through the time of flight of the signal, along the line of sight of the radar. By inspecting the Doppler shift in frequency of the received signal with respect to the emitted signal, or by analysing the variations of the range measurements, the relative radial velocity, *i.e.*, the velocity along the line of sight of the radar, can also be determined. Finally, from the angular position measurements of the directional antenna, the relative azimuth angle of the target with respect to the line of sight of the radar can be determined. A mathematical description of the state measured by the radar, $\mathbf{x}_{\dot{R}}$, is provided in Equation (5.31), where it should be noted that ρ represents the range measurement, $\dot{\rho}$ the range-rate measurement and ε the azimuth angle measurement performed by the radar.

$$\mathbf{x}_R = \left(\rho, \dot{\rho}, \varepsilon\right)^T \tag{5.31}$$

Note that no sensor model is developed for the current and future relative navigation sensors, given that the output of an artificial VBN system will be modelled directly from the actual relative state between both spacecraft. Therefore, no sensory measurements are involved and thus no sensor models must be developed.

Lidar

The lidar sensor operates by emitting pulsed light waves and detecting the corresponding return of these waves, as reflected by an external object. Similarly to the radar, the distance to the object can be determined through measuring the time between emission and reception of the reflected signal. By doing so, the lidar provides a cloud of points in three dimensional space, with each point defined by an own set of Cartesian position components that orient the cloud of points with respect to the sensor. When applying this to the current reference mission, the lidar onboard the chaser spacecraft can determine the relative position and orientation with respect to the target. A mathematical description of the state measured by the lidar, $\mathbf{x}_L$, is provided in Equation (5.32), where it should be noted that $\mathbf{x}_l$, $\mathbf{y}_l$ and $\mathbf{z}_l$ represent the vectors containing the X, Y and Z-coordinates of the points detected by the lidar, respectively.

$$\mathbf{x}_L = \left(\mathbf{x}_l, \mathbf{y}_l, \mathbf{z}_l\right)^T \tag{5.32}$$

Visual Camera

The far- and close-range visual cameras will simply provide images of the target spacecraft in the visible spectrum. A 3D model of the target is matched with these images to obtain estimates on its relative translational and rotational state with respect to the chaser spacecraft. When combining these measurements with the lidar and radar measurements in the computer vision algorithm, a pose estimation of the object captured in the images can be performed. Furthermore, the velocity and angular rate of the object captured in the images is estimated from the changes in its orientation and position over consecutive images. The output generated by the corresponding computer vision algorithm is thus a three dimensional position and attitude estimate of the target with respect to the chaser. A mathematical description of this state output, $\mathbf{x}_{VC}$, is provided in Equation (5.33). In this equation, x_{VC}, y_{VC} and z_{VC} represent the relative cartesian position components of the target with respect to the chaser, $\dot{x}_{VC}$, $\dot{y}_{VC}$ and $\dot{z}_{VC}$ the relative cartesian velocity components, $\delta q_{1,VC}$, $\delta q_{2,VC}$, $\delta q_{3,VC}$ and $\delta q_{4,VC}$ the relative attitude components in terms of the error quaternion and $\omega_{1,VC}$, $\omega_{2,VC}$ and $\omega_{3,VC}$ the relative angular velocity components as obtained from the computer vision algorithm.

$$\mathbf{x}_{VC} = \left(x_{VC}, y_{VC}, z_{VC}, \dot{x}_{VC}, \dot{y}_{VC}, \dot{z}_{VC}, \delta q_{1,VC}, \delta q_{2,VC}, \delta q_{3,VC}, \delta q_{4,VC}, \omega_{1,VC}, \omega_{2,VC}, \omega_{3,VC}\right)^T \tag{5.33}$$

5.2.2. Vision-based Navigation

In the vision based navigation system, the pose estimation from the visual camera system is combined with radar and lidar data in a state estimation algorithm to increase the accuracy of the relative state estimate. However, due to its complexity, the current thesis study will not develop the actual pose estimation procedure, nor model the generation of radar and lidar data. Rather, the output generated by the relative navigation system will artificially be simulated. To provide the output of the relative navigation filter, a simplified model is developed that takes the actual relative state between both spacecraft as input and provides an estimate of this relative state as output, as would be obtained from an actual navigation filter. It should be noted that, for simplicity reasons, the output is obtained by adding white noise to the actual relative state of the spacecraft, rather than coloured noise. This is because the imposed time constraints did not allow to investigate the necessary correlations between the noise parameters to obtain a representative estimate of the coloured noise values in the output of a true VBN system. Furthermore, other sources of measurement errors are omitted, as these are assumed to be corrected for by the artificial relative navigation filter. Hence a mathematical model for the vision-based navigation system can be described by Equation (5.34). In this equation, $\mathbf{x}_{\text{VBN}}$ represents the relative state estimate that is outputted by the artificial navigation filter, $\mathbf{x}_{\text{rel}}$ the true relative state between both spacecraft and $\mathbf{v}_{\text{VBN}}$ a random white noise vector representing the estimation errors.

$$\mathbf{x}_{\text{VBN}} = \mathbf{x}_{\text{rel}} + \mathbf{v}_{\text{VBN}} \tag{5.34}$$

Note that the output of the vision-based navigation system, $\mathbf{x}_{\text{VBN}}$, contains the relative translational and rotational state components, similar to what is provided by the computer vision algorithm of the visual camera system in Equation 5.33. Moreover, also the true relative state, $\mathbf{x}_{\text{rel}}$, contains these translational and rotational state components. Lastly, the white noise vector, $\mathbf{v}_{\text{VBN}}$, contains a distinct white noise value for each component in the state vector. Based on **REQ_S_C_16**, the relative navigation accuracy can be taken as 1% of the actual relative state. Therefore, the value of the standard deviation used to sample the white noise vector is implemented as a varying parameter, whose magnitude is a predefined percentage of the true relative state. A mathematical description of this white noise vector is provided in Equation (5.35), where $\mathbf{v}_{\text{VBN}}$ is sampled from a normal distribution N(), which is characterised by a zero mean and a standard deviation equal to a predefined percentage $p_{\%}$ of the magnitude of the actual relative state vector $||\mathbf{x}_{\text{rel}}||$.

$$\mathbf{v}_{\text{VBN}} \sim \text{N}(0, p_{\%}||\mathbf{x}_{\text{rel}}||) \tag{5.35}$$

In reality, the output of the vision-based navigation system would be provided in the body frame of the chaser spacecraft and subsequently be transformed to its equivalent representation in the inertial frame. This is necessary as the relative translational state expressed in the body frame of the chaser is of no interest to the GNC system. Rather, the inertial translational state of both spacecraft is necessary to obtain the relative Hill state of the chaser. This can only be obtained from the absolute inertial state of chaser, provided by the inertial navigation system, and the relative inertial state of the target, obtained from the relative navigation system. Therefore, as the current study artificially models the output of the VBN, it can directly be provided in the inertial frame of the motion simulator. This allows the transformations to and from the body frame of the chaser to be omitted. Furthermore, for the attitude controller, the relative rotational state can be provided both in the body frame of the chaser or the inertial frame. However, for consistency with the translational state, it is chosen to provide also the relative rotational state with respect to the inertial frame.

To obtain the Hill state of the chaser spacecraft with respect to its target, as is used by the guidance system in Chapter 4, it is required to first obtain the inertial state of both before they can be transformed into a relative Hill state. As the VBN system readily provides the relative translational state of the target in the inertial frame, it can be combined with the inertial state estimate of the chaser to obtain the absolute state of the target, as given in Equation (5.36). In this equation, $\mathbf{x}_{\text{trans}_\text{T}}$ represents the absolute translational state of the target, $\mathbf{x}_{\text{VBN}_{\text{trans}}}$ the relative translational state of the chaser with respect to the target, as obtained from the relative navigation system, and $\mathbf{x}_{\text{trans}_\text{C}}$ the absolute translational state of the chaser, as obtained from the inertial navigation system. Note that as the translational states contain only Cartesian components, Equation (5.36) is a simple addition of the respective position and velocity components and is therefore not further elaborated.

$$\mathbf{x}_{\text{trans}_T} = -\mathbf{x}_{\text{VBN}_{\text{trans}}} + \mathbf{x}_{\text{trans}_C} \tag{5.36}$$

Subsequently, the translational state vector of the chaser spacecraft in the Hill frame of its target can be calculated from $\mathbf{x}_{\text{trans}_T}$ and $\mathbf{x}_{\text{trans}_C}$, using the transformation procedure provided in Chapter 3.

To obtain the absolute rotational state of the target, as is used by the control system in Chapter 6, the relative state estimate can be combined with the inertial state estimate of the chaser, as can seen in Equation (5.37).

$$\mathbf{x}_{\text{rot}_T} = -\mathbf{x}_{\text{VBN}_{\text{rot}}} + \mathbf{x}_{\text{rot}_C} \tag{5.37}$$

In this equation, $\mathbf{x}_{\text{rot}_T}$ represents the absolute rotational state of the target, $\mathbf{x}_{\text{VBN}_{\text{rot}}}$ the relative rotational state of the chaser with respect to the target, as obtained from the relative navigation system, and $\mathbf{x}_{\text{rot}_C}$ the absolute rotational state of the chaser, as obtained from the inertial navigation system. Note that the rotational states contain both attitude quaternions and Cartesian angular velocity components. Therefore, Equation (5.37) represents a simple addition for the respective angular rate components, and is therefore not further elaborated. This is in contrast to the quaternion representation, for which the relative state is provided by the VBN system in terms of an error quaternion. To obtain the inertial attitude quaternion of the target spacecraft, $\mathbf{q}_T$, the inertial quaternion of the chaser, $\mathbf{q}_C$, is combined with the relative error quaternion, $\delta\mathbf{q}$, as performed Equation (5.38). Note that this equation represents the inverse relation of what is provided in Equation (3.17).

$$\mathbf{q}_T = \begin{bmatrix} q_{4_C} & -q_{3_C} & q_{2_C} & q_{1_C} \\ q_{3_C} & q_{4_C} & -q_{1_C} & q_{2_C} \\ -q_{2_C} & q_{1_C} & q_{4_C} & q_{3_C} \\ -q_{1_C} & -q_{2_C} & -q_{3_C} & q_{4_C} \end{bmatrix} \delta\mathbf{q} \tag{5.38}$$

Lastly, it should be noted that as the output of the relative navigation system is artificially generated, its accuracy can be tweaked to fit the requirements of the current reference scenario. However, to maintain realism, the performance of the implemented navigation system will be modelled following the 1% navigation accuracy requirement, as given by **REQ_S_C_16**.

5.3. Navigation System Design

The current section details the navigation system design for the ClearSpace-1 mission. Subsection 5.3.1 presents the specifics of the employed sensor models, Subsection 5.3.2 contains the details concerning the UKF as inertial navigation filter and Subsection 5.3.3 elaborates the vision-based navigation system as artificial relative navigation system.

5.3.1. Sensor Models

The sensor models used for the ClearSpace-1 spacecraft are all part of the GRADS library and have previously been elaborated in Section 5.1. The interested reader is, however, referred to (Mooij, 2021a) for more details.

Inertial Measurement Unit

A summary of the settings defining the measurements provided by the accelerometers and gyroscopes of the IMU is provided in Table 5.1. Note that in this table, the σ_{v_a} and σ_{v_ω} parameters indicate the 1-σ standard deviations used to sample the zero-mean Gaussian white-noise vector present in the accelerometer and gyroscope measurements, respectively. In this table, the values for the bias factors, scaling factors and misalignment factors are taken from (Mooij and Chu, 2001), as well as the standard deviation magnitudes used for the white noise vectors. Moreover, the sampling rate is set at 100 Hz, making the navigation system operate at an equal frequency to that of the control system. This is done such that each control command can be calculated using a newly obtained navigation solution. Furthermore, reducing the step size of the integration performed during the prediction step of the navigation filter will improve its accuracy and, due to the computational efficiency of the UKF algorithm, will not require excessive computation time. Therefore, all subsequent navigation sensors and navigation algorithms, either inertial or relative, will operate at this frequency of 100 Hz.

Table 5.1: Inertial Measurement Unit model settings as provided by Mooij and Chu (2001)

Accelerometer Error	Value	Gyroscope Error	Value
$b_{IMU,x}$ [m/s^2]	$3 \cdot 10^{-4}$	b_{IMU,ω_1} [rad/s]	$5 \cdot 10^{-6}$
$b_{IMU,y}$ [m/s^2]	$-3.5 \cdot 10^{-4}$	b_{IMU,ω_2} [rad/s]	$-5 \cdot 10^{-6}$
$b_{IMU,z}$ [m/s^2]	$3 \cdot 10^{-4}$	b_{IMU,ω_3} [rad/s]	$5 \cdot 10^{-6}$
$s_{IMU,x}$ [-]	$2 \cdot 10^{-4}$	s_{IMU,ω_1} [-]	$2 \cdot 10^{-4}$
$s_{IMU,y}$ [-]	$-1.7 \cdot 10^{-4}$	s_{IMU,ω_2} [-]	$-1.7 \cdot 10^{-4}$
$s_{IMU,z}$ [-]	$2.3 \cdot 10^{-4}$	s_{IMU,ω_3} [-]	$2.3 \cdot 10^{-4}$
$m_{IMU,xy}$ [-]	10^{-6}	$m_{IMU,\omega_1\omega_2}$ [-]	10^{-6}
$m_{IMU,xz}$ [-]	-10^{-6}	$m_{IMU,\omega_1\omega_3}$ [-]	-10^{-6}
$m_{IMU,yx}$ [-]	-10^{-6}	$m_{IMU,\omega_2\omega_1}$ [-]	-10^{-6}
$m_{IMU,yz}$ [-]	$2 \cdot 10^{-6}$	$m_{IMU,\omega_2\omega_3}$ [-]	$2 \cdot 10^{-6}$
$m_{IMU,zx}$ [-]	10^{-6}	$m_{IMU,\omega_3\omega_1}$ [-]	10^{-6}
$m_{IMU,zy}$ [-]	$2 \cdot 10^{-6}$	$m_{IMU,\omega_3\omega_2}$ [-]	$2 \cdot 10^{-6}$
σ_{v_a} [m/s^2]	10^{-5}	σ_{v_ω} [rad/s]	10^{-6}
seed [-]	12345	seed [-]	54321
sample rate [Hz]	100	sample rate [Hz]	100

Table 5.2: GPS receiver model settings

GPS Receiver Error	Value
$c \cdot b_{GPS,clock}$ [m]	1
$c \cdot d_{GPS,clock}$ [m/s]	0.1
σ_{v_ρ} [m]	10^{-3}
$\sigma_{v_{\dot{\rho}}}$ [m/s]	10^{-4}
seed [-]	1234
mask angle [deg]	10
sample rate [Hz]	100

Global Positioning System

A summary of the implemented GPS receiver settings is provided in Table 5.2, where the σ_{v_ρ} and $\sigma_{v_{\dot{\rho}}}$ parameters indicate the 1-σ standard deviation used to sample the zero-mean Gaussian white-noise vector present in the range and range rate measurements, respectively. Note that the values for the clock bias and clock drift errors are taken as 1 m and 0.1 m/s, respectively. However, their exact numerical value is not of great importance given that these error parameters can accurately be estimated by the UKF and therefore will not really influence the accuracy of the navigation solution. What is of high importance to the accuracy of the navigation solution, is the standard deviation values for the white-noise errors in the range and range rate measurements. These have been set at 10^{-3} m and 10^{-4} m/s, respectively, where their magnitudes have been determined based on the navigation accuracy requirement as established by **REQ_S_C_16**. Lastly, the mask angle, defining the minimum elevation angle permitted between the GPS satellites and the GPS receiver, is set at a representative value of 10 degrees and the sampling rate is set at 100 Hz.

Star Sensor

A summary of the implemented star sensor settings is provided in Table 5.3. Note that in this table, the σ_{v_q} parameter indicates the standard deviation of the attitude error, and it corresponding value is taken from (Canuto et al., 2018). Moreover, the sampling rate of star sensor output is set to 100 Hz. Note that two star sensors are implemented onboard the chaser spacecraft to ensure sufficiently accurate angular velocity estimates can be obtained from the derivative of two independent star sensor quaternions. Therefore, the model settings as provided in Table 5.3 apply to both sensors onboard the spacecraft. Nevertheless, to simulate independent measurements, the noise in the first sensor is

Table 5.3: Star sensor model settings as provided by Canuto et al. (2018)

Star Sensor Error	Value
σ_{v_q} [deg]	$2.5 \cdot 10^{-6}$
sample rate [Hz]	100

initialised using a seed equal to 17061999, whereas the second sensor is initialised using a seed equal to 20101998.

5.3.2. Inertial Navigation Filter

To apply the UKF algorithm as the inertial navigation system of the ClearSpace-1 satellite, a number of variable definitions are required. A mathematical description of the state vector of the chaser spacecraft, $\mathbf{x}_C$, estimated by the UKF filter is provided in Equation (5.39).

$$\begin{aligned}\mathbf{x}_C = (&x_c, y_c, z_c, V_{x,c}, V_{y,c}, V_{z,c}, b_{IMU,x}, b_{IMU,y}, b_{IMU,z}, b_{GPS,clock}, d_{GPS,clock},\\ &q_{1,c}, q_{2,c}, q_{3,c}, q_{4,c}, \omega_{1,c}, \omega_{2,c}, \omega_{3,c})^T\end{aligned} \tag{5.39}$$

In this equation, x_c, y_c and z_c represent the Cartesian position components, $V_{x,c}$, $V_{y,c}$ and $V_{z,c}$ the Cartesian velocity components, $q_{1,c}$, $q_{2,c}$, $q_{3,c}$ and $q_{4,c}$ the quaternion attitude components and $\omega_{1,c}$, $\omega_{2,c}$ and $\omega_{3,c}$ the angular velocity components. Further part of this state vector are $b_{IMU,x}$, $b_{IMU,y}$ and $b_{IMU,z}$, which represent the bias factors of the accelerometers in the IMU, $b_{GPS,clock}$, which represents the bias of the GPS receiver's clock, and $d_{GPS,clock}$, which represents the drift of the GPS receiver's clock. Moreover, it should be noted that other error parameters of the sensor models, such as scaling factors or misalignment factors, cannot accurately be estimated from the obtainable measurements. Therefore, to increase computational efficiency of the state estimation algorithm, an estimation of these parameters will be omitted. Consequently, they are not part of the to-be-estimated state vector. Furthermore, the measurements taken by the IMU are used for the control input vector, $\mathbf{u}_C$, in the navigation filter. This is described in Equation (5.40).

$$\mathbf{u}_C = (\ddot{x}_{IMU}, \ddot{y}_{IMU}, \ddot{z}_{IMU}, \omega_{IMU,1}, \omega_{IMU,2}, \omega_{IMU,3})^T \tag{5.40}$$

The remaining measurements, *i.e.*, the measurements taken by the GPS sensor and star sensors, constitute the observation vector $\mathbf{y}_C$. A mathematical description of this observation vector is provided in Equation (5.41).

$$\mathbf{y}_C = \left(\rho_1, \rho_2, \rho_3, \rho_4, \dot{\rho}_1, \dot{\rho}_2, \dot{\rho}_3, \dot{\rho}_4, q_{1_1}, q_{1_2}, q_{1_3}, q_{1_4}, q_{2_1}, q_{2_2}, q_{2_3}, q_{2_4}\right)^T \tag{5.41}$$

However, as the IMU measurements are used as control input vector in the navigation filter, no observations are available for the corrective step of the angular velocity estimates in the UKF algorithm. This is in contrast to the translational velocity estimates of the chaser, as these are corrected through range rate measurements of the GPS receiver. Therefore, to improve the accuracy of the angular velocity estimation, a pseudo-measurement is calculated using the derivative of consecutive star sensor measurements. This significantly improves the estimation accuracy of the angular velocity components, given that the measurements of the star sensor are accurate and provided at a sufficiently high sampling rate. By inverting the kinematic attitude equation, as previously provided in Equation (3.22), a pseudo-measurement of the angular velocity can be obtained from the quaternion derivative, as given in Equation (5.42). In this equation, the transpose of the auxiliary matrix $\mathbf{Q}$, as defined in Equation (3.23), is used. Furthermore, this equation makes use of the quaternion derivative $\dot{\mathbf{q}}$, which can be calculated using Equation (5.43).

$$\boldsymbol{\omega} = 2\,(\mathbf{Q})^T\,\dot{\mathbf{q}} \tag{5.42}$$

$$\dot{\mathbf{q}} = \frac{\mathbf{q}_k - \mathbf{q}_{k-1}}{\Delta t} \tag{5.43}$$

Equation (5.43) uses two consecutive measurements of the star sensor, $\mathbf{q}_k$ at epoch k and $\mathbf{q}_{k-1}$ at epoch $k-1$, and is therefore subject to noise. Moreover, it is generally known that taking the derivative of noisy measurements will always decrease the signal to noise ratio and therefore increase errors. One way to deal with this is by first passing the noisy star sensor measurements through a low pass filter, effectively smoothing part of the noise from the signal, before taking its derivative. However, given the low level of noise present in the star sensor signal, this was deemed redundant as sufficient navigation accuracy could be obtained when taking taking a direct derivative of the unfiltered signal. Nevertheless, in case a hardware implementation of the current UKF algorithm does require the signal to be filtered, a low pass filter can straightforwardly be applied. Note that a low pass filter is needed rather than a high pass filter, as the high frequency oscillations associated with noise need to be removed from the measurements. Hence the observation vector $\mathbf{y}_C$ can be extended with these pseudo-measurements, given by ω_{1_i} and ω_{2_i} for the first and second star sensor pseudo-measurement components, respectively, and results in Equation (5.44).

$$\begin{aligned}\mathbf{y}_C = (&\rho_1, \rho_2, \rho_3, \rho_4, \dot{\rho}_1, \dot{\rho}_2, \dot{\rho}_3, \dot{\rho}_4, q_{1_1}, q_{1_2}, q_{1_3}, q_{1_4},\\ &q_{2_1}, q_{2_2}, q_{2_3}, q_{2_4}, \omega_{1_1}, \omega_{1_2}, \omega_{1_3}, \omega_{2_1}, \omega_{2_2}, \omega_{2_3})^T\end{aligned} \tag{5.44}$$

With these three vectors defined, the functions f() and g() that govern the model employed by the navigation filter, as given in Equations (5.12) and (5.13), can be described. A mathematical description of f() is provided in Equation (5.45), whereas Equation (5.49) provides a description of g().

Equation (5.45) combines the differential equations of motion with a simple Euler integration scheme to estimate the inertial state vector of the chaser spacecraft at the next epoch, using information of the current epoch. It is assumed that this Euler integration scheme is sufficiently accurate given the small time step of the navigation system. Furthermore, any errors made in this propagation step can subsequently be corrected for during the update step of the filter. In this equation, h represents the time step, $\mathbf{V}_C$ the velocity vector of the chaser spacecraft given by $\mathbf{V}_C = \left(V_{x,c}, V_{y,c}, V_{z,c}\right)^T$, $\mathbf{a}_e$ the estimated non-gravitational acceleration vector given by Equation (5.46) and $\mathbf{g}$ the gravitational acceleration vector given by $\mathbf{g} = \left(g_x, g_y, g_z\right)^T$. Furthermore, $\mathbf{0}_{3\times1}$ represents a column vector containing three zeroes, as the bias factors of the accelerometer in the IMU do not contain any predictable dynamics, similarly to the clock drift of the GPS receiver. Moreover, the auxiliary quaternion matrix, $\mathbf{Q}$, is identical to what is previously defined in Chapter 3, $\boldsymbol{\omega}_e$ represents the estimated angular velocity vector given by Equation (5.48) and $\mathbf{J}$ represents the inertia tensor of the chaser spacecraft. It should however be noted that, after propagating the attitude in terms of quaternions with Equation (5.45), the norm of the propagated attitude vector must be normalised to satisfy the unit-length constraint for attitude quaternions.

$$f(\mathbf{x}_C, \mathbf{u}_C, \mathbf{v}) = \mathbf{x} + h\begin{pmatrix} \mathbf{V}_C \\ \mathbf{a}_e + \mathbf{g} \\ \mathbf{0}_{3\times1} \\ d_{GPS,clock} \\ 0 \\ \frac{1}{2}\mathbf{Q}\boldsymbol{\omega}_e \\ -\mathbf{J}^{-1}\left(\boldsymbol{\omega}_e \times \mathbf{J}\boldsymbol{\omega}_e\right) \end{pmatrix} \tag{5.45}$$

The estimated non-gravitational acceleration vector, $\mathbf{a}_e$, is defined in Equation (5.46), where it should be noted that this estimate is based on the measurements taken by the accelerometers onboard the IMU and corrected through the current best estimate of its bias factors.

$$\mathbf{a}_e = \begin{pmatrix} \ddot{x}_{IMU} \\ \ddot{y}_{IMU} \\ \ddot{z}_{IMU} \end{pmatrix} - \begin{pmatrix} b_{IMU,x} \\ b_{IMU,y} \\ b_{IMU,z} \end{pmatrix} \tag{5.46}$$

Note that it is assumed that these measurements are referenced with respect to the inertial frame. This implies that also the gravitational acceleration vector, $\mathbf{g}$, must be referenced with respect to the inertial frame. This gravitational acceleration vector is calculated onboard with Equation (5.47), where

μ represents the gravitational parameter of Earth given by $\mu = 3.9860 \cdot 10^{14}$ m^3/s^2 and $\mathbf{r}$ the inertial position vector of the chaser spacecraft given by $\mathbf{r}_C = (x_c, y_c, z_c)^T$. The current gravity model implemented onboard the navigation system contains a point mass representation of Earth, as it is found that sufficient navigation accuracy can be obtained while requiring minimum computational effort. Nevertheless, in case a hardware implementation of the current UKF algorithm requires the onboard gravity model to improve, a spherical harmonic representation of Earth can be implemented, where the degree and order of the model is dependent on the required model improvement.

$$\mathbf{g} = \frac{-\mu \mathbf{r}_C}{||\mathbf{r}_C||^3} \tag{5.47}$$

The estimated angular velocity vector, $\boldsymbol{\omega}_e$, is defined in Equation (5.48), where it should be noted that this estimate is solely based on the measurements taken by the gyroscopes onboard the IMU.

$$\boldsymbol{\omega}_e = \begin{pmatrix} \omega_{IMU,1} \\ \omega_{IMU,2} \\ \omega_{IMU,3} \end{pmatrix} \tag{5.48}$$

Equation (5.49) provides the mathematical description for the vector function g(). The first row in Equation (5.49) represents the four range equations to calculate the distance from the transmitting satellites to the receiver. In this equation, $\mathbf{r}_{GPS,i}$ represents the Cartesian position vector of the ith satellite given by $\mathbf{r}_{GPS,i} = (x_{GPS,i}, y_{GPS,i}, z_{GPS,i})^T$, $\mathbf{r}_C$ the Cartesian position vector of the chaser, c the speed of light and $b_{GPS,clock}$ the clock bias of the receiver. Furthermore, the second row in Equation (5.49) represents the four range rate equations that calculate the velocity of the receiver with respect to the transmitting satellites. In this equation, $\mathbf{V}_{GPS,i}$ represents the Cartesian velocity vector of the ith satellite given by $\mathbf{V}_{GPS,i} = \left(V_{x,GPS_i}, V_{y,GPS_i}, V_{z,GPS_i}]\right)^T$, $\mathbf{V}_C$ the Cartesian velocity vector of the chaser, c the speed of light and $d_{GPS,clock}$ the clock drift of the receiver. Lastly, the bottom rows of Equation (5.49) represent the inertial attitude quaternion of the chaser spacecraft, $\mathbf{q}_C$, as measured by each of the two star sensors along with the inertial angular velocity vector of the chaser spacecraft, $\boldsymbol{\omega}_C$, as calculated by the previously established derivative pseudo-measurements.

$$g(\mathbf{x}_C, \mathbf{w}) = \begin{pmatrix} \sqrt{(\mathbf{r}_{GPS,i} - \mathbf{r}_C)(\mathbf{r}_{GPS,i} - \mathbf{r}_C)^T} + c \cdot b_{GPS,clock} & \text{for } i = 1, ..., 4 \\ \frac{(\mathbf{r}_{GPS,i} - \mathbf{r}_C)^T (\mathbf{V}_{GPS,i} - \mathbf{V}_C)}{\sqrt{(\mathbf{r}_{GPS,i} - \mathbf{r}_C)^T (\mathbf{r}_{GPS,i} - \mathbf{r}_C)}} + c \cdot d_{GPS,clock} & \text{for } i = 1, ..., 4 \\ \mathbf{q}_C \\ \mathbf{q}_C \\ \boldsymbol{\omega}_C \\ \boldsymbol{\omega}_C \end{pmatrix} \tag{5.49}$$

To initialise the UKF algorithm, it is assumed that an accurate initial state estimate for $\mathbf{x}_C$ is available through either ground tracking data, or from the navigation filter employed during the far-range rendezvous phase. Therefore, a small error is added to the actual inertial state of the chaser spacecraft and used as initial guess $\hat{\mathbf{x}}_0$. Note that the imposed error is consistent with requirement **REQ_S_C_16**, stating that the navigation accuracy equals 1% of the relative state to the target. Moreover, the state vector of the navigation system also contains error parameters of the navigation sensors, *i.e.*, the bias vector of the IMU and the clock drift and clock bias of the GPS receiver, and these parameters are initialised by zero, due to a lack of better estimates. Furthermore, the initial state error covariance matrix $\hat{\mathbf{P}}_x^0$ must be estimated based on the corresponding uncertainty in the initial state. Note that for simplicity reasons, this and other covariance matrices are assumed to be diagonal. Moreover, the process noise covariance matrix, $\mathbf{Q}$, and the observation noise covariance matrix, $\mathbf{R}$, have to be estimated. An educated guess for $\mathbf{Q}$ can be obtained when comparing the model onboard the navigation filter with the best known model of the system dynamics. The matrix $\mathbf{R}$, on the other hand, is estimated based on the accuracy of the navigation sensors. Nevertheless, the value of these three initial matrices is subsequently tuned to establish a correct functioning of the UKF, by ensuring that the estimation error made by the filter is bounded by the standard deviations of the error covariance matrices. More information on this tuning is, however, provided in Section 5.4. Furthermore, a value for the α, β and κ parameters must be selected. Their magnitude was selected from the discussion provided by Haykin

Table 5.4: Unscented Kalman filter initialisation

UKF Parameter	Value	UKF Parameter	Value
$\hat{\mathbf{x}}_0$: $\mathbf{r}$ [m]	$\begin{pmatrix} -4975526 \\ -782634 \\ -4920265 \end{pmatrix}$	$\hat{\mathbf{P}}_x^0$ [-]	$\mathbf{I}_{18\times18}$
$\mathbf{V}$ [m/s]	$\begin{pmatrix} 4847 \\ 2398 \\ 5282 \end{pmatrix}$	$\mathbf{Q}$ [-]	$\text{diag}([4\cdot10^{-5}, 4\cdot10^{-5}, 4\cdot10^{-5}, \ldots$ $\frac{1}{4}\cdot10^{-5}, \frac{1}{4}\cdot10^{-5}, \frac{1}{4}\cdot10^{-5}, \ldots$ $10^{-6}, 10^{-6}, 10^{-6}, 10^{-10}, 10^{-10}, \ldots$ $0.0071, 0.0084, 0.0094, 0.0154 \ldots$ $0.0006, 0.0006, 0.0006])$
$\mathbf{b}$ [m/s^2]	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	$\mathbf{R}$ [-]	$\text{diag}([10^{-5}, 10^{-5}, 10^{-5}, 10^{-5}, \ldots$ $10^{-7}, 10^{-7}, 10^{-7}, 10^{-7}, \ldots$ $10^{-14}, 10^{-14}, 10^{-14}, 10^{-14}, \ldots$ $10^{-14}, 10^{-14}, 10^{-14}, 10^{-14}, \ldots$ $10^{-9}, 10^{-9}, 10^{-9}, \ldots$ $10^{-9}, 10^{-9}, 10^{-9}])$
$b_{GPS,clock}$ [m] $d_{GPS,clock}$ [m/s]	$\begin{matrix} 0 \\ 0 \end{matrix}$	α [-]	10^{-3}
$\mathbf{q}$ [-]	$\begin{pmatrix} -0.5 \\ -0.3 \\ -0.1 \\ 0.8 \end{pmatrix}$	β [-]	2
$\boldsymbol{\omega}$ [deg/s]	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	κ [-]	0

(2001), with the purpose of obtaining an optimal spread of sigma points for a Gaussian distributions. With all these parameters defined, the UKF algorithm as provided in Algorithm 1 can be initialised and recursively estimated states of the chaser spacecraft can be obtained. A summary of the implemented UKF settings is provided in Table 5.4

5.3.3. Relative Navigation Filter

Due to the simple nature of the artificially implemented vision-based navigation system, the only system parameters that need to be provided are the 1-σ standard deviation values of the white-noise errors, as well as the sampling rate at which the relative navigation filter provides an output. A summary of the implemented VBN settings is provided in Table 5.5. Note that in this table, the $\sigma_{v_{pos}}$ parameter indicates the standard deviation error that is applied to each component of the relative position vector, $\sigma_{v_{velo}}$ to the components of the relative velocity vector, $\sigma_{v_{att}}$ to the components of the relative attitude vector and $\sigma_{v_{rate}}$ to the components of the relative angular velocity vector. The specific values for these parameters are selected based on the relative navigation accuracy requirement that has been established in **REQ_S_C_16**. This results in the noise values, as provided in Table 5.5, such that, when the error components are combined, the resulting navigation error for the position vector, velocity vector, attitude vector and angular velocity vector fall within 1% relative navigation accuracy. This means that for the position vector, velocity vector and angular velocity vector, each component contributes to $\frac{1}{\sqrt{3}}$ of the total error, whereas for the quaternion error each component contributes $\frac{1}{\sqrt{4}}$ of the total error, and that the 3-σ bound of this total error needs to be within the relative navigation accuracy of **REQ_S_C_16**. Therefore, the individual error bounds of $\frac{\sqrt{3}}{9}$ and $\frac{1}{6}$ result. Lastly, the sampling rate of the VBN system is, like the inertial navigation system, set to 100 Hz to be conform the sampling rate of the control system.

Table 5.5: Vision-based navigation system settings

VBN Error	Value	VBN Error	Value
$\sigma_{v_{pos}}$ [%]	$\frac{\sqrt{3}}{9}$	position seed	4720
$\sigma_{v_{velo}}$ [%]	$\frac{\sqrt{3}}{9}$	velocity seed	4721
$\sigma_{v_{att}}$ [%]	$\frac{1}{6}$	attitude seed	4722
$\sigma_{v_{rate}}$ [%]	$\frac{\sqrt{3}}{9}$	angular rate seed	4723
sample rate [Hz]	100		

Table 5.6: Unscented Kalman filter initialisation for SWARM A position estimation

UKF Parameter	Value
$\hat{\mathbf{x}}_0$:	
$\mathbf{r}$ [m]	$(5.8717, -1.5724, -3.0642)^T \cdot 10^6$
$\mathbf{V}$ [m/s]	$(-3.7035, 0.7281, -7.3315)^T \cdot 10^3$
$b_{GPS,clock}$ [m]	0
$d_{GPS,clock}$ [m/s]	0
$\hat{\mathbf{P}}_x^0$ [-]	$\mathbf{I}_{8\times8}$
$\mathbf{Q}$ [-]	$\text{diag}(250, 250, 250, 0.025, 0.25, 0.025, 10, 10)$
$\mathbf{R}$ [-]	$\text{diag}(0.0004, 0.0004, 0.0004, 0.0004, 0.0004, 0.0004)$
α [-]	10^{-3}
β [-]	2
κ [-]	0

5.4. Verification

Given that the implemented navigation system is split into two basic functions, also its verification procedure will performed in a two-step process. For the verification of the inertial navigation function, it must be noted that the employed sensor models have already been verified by Mooij (2021b) and Cattani (2022). Therefore, a dedicated verification of models can be omitted. What will be verified, however, is a correct implementation of the UKF. First, the UKF is employed to determine the position of the SWARM A satellite from GPS pseudorange measurements. As input, the precise GPS orbits and clock corrections as provided by the International GNSS Service are used, alongside with the SWARM A GPS pseudorange measurements. To verify the validity of the results calculated by the UKF algorithm, the position estimates are compared with a precise orbit calculation provided by van den IJsel et al. (2015). A comparison between the precise orbit calculation, the navigation solution obtained by the UKF and a navigation solution obtained by a representative Extended Kalman filter are provided in Figure 5.2. Note that the Extended Kalman filter is added to the figure with the purpose of providing a benchmark solution of what can be obtained with the provided pseudorange measurements. A performance comparison between the UKF and Extended Kalman filter falls outside of the scope of the current discussion, as the UKF has been selected for previously elaborated reasons, *i.e.*, its derivative-free nature. Furthermore note that, to obtain this navigation solution, the input parameters as provided in Table 5.6 are used. Note that the initial state in Table 5.6 differs from the state vector definition provided in Equation (5.39), given that only the position of the SWARM A satellite is to be estimated. Therefore, the state vector contains the Cartesian position and velocity components, as well as the receiver clock bias and receiver clock drift present in GPS measurements. From visual inspection of Figure 5.2 it can be concluded that the UKF algorithm is capable of obtaining position estimates for the SWARM A satellite with a comparable accuracy as achievable from an equivalent Extended Kalman filter. Furthermore, when inspecting the navigation accuracies as provided in Table 5.7, it can be found that the UKF obtains a navigation error that is of the same order of magnitude as the Extended Kalman Filter. Hence based on this discussion, it can be concluded that the basic UKF algorithm is implemented correctly.

Moreover, the integration of the UKF algorithm with the sensor models is verified by comparing the estimated inertial state of the chaser spacecraft, as obtained from the inertial navigation system, with

Table 5.7: Maximum navigation errors obtained by Unscented and Extended Kalman Filters during position estimation of the SWARM A satellite compared to precise orbit determination given in (van den IJsel et al., 2015)

Max. Navigation Error	Value
UKF [m]	0.024
EKF [m]	0.015

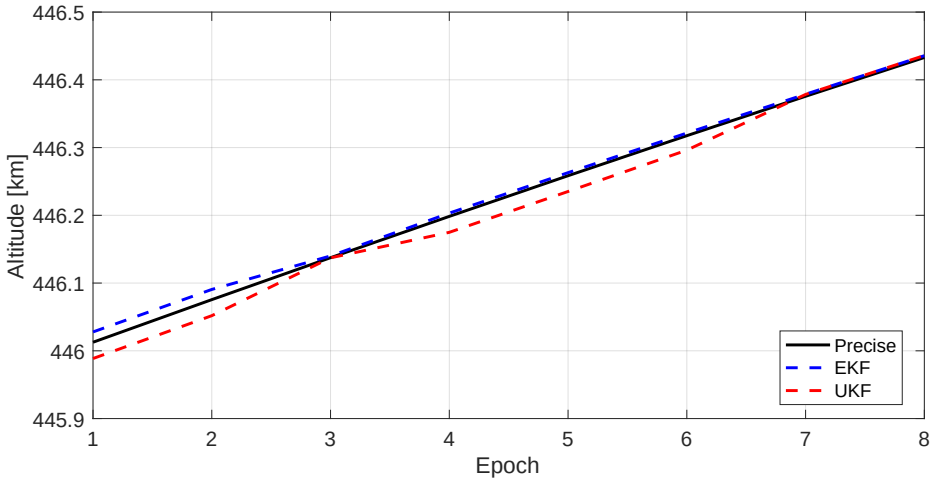


Figure 5.2: Comparison of inertial navigation solution provided by Unscented Kalman filter, equivalent Extended Kalman filter and precise orbit determination given in (van den IJsel et al., 2015) for the SWARM A satellite

its true inertial state, as obtained from the motion simulator. In this way, the functionality of the complete inertial navigation system can be verified. A visual representation of the inertial state estimation errors is provided in Figures 5.3 to 5.4 and Figures 5.5 to 5.6, where the navigation error per state component is provided. Moreover, the figures also contain the corresponding bounding curves based on the standard deviations of the state error covariance matrices. Note that these covariance curves are used to tune the measurement and process noise covariance matrices, $\mathbf{R}$ and $\mathbf{Q}$, as well as the initial state error covariance matrix $\hat{\mathbf{P}}_x^0$, such that the true estimation error is of the same magnitude or smaller as the expected error. Furthermore note that the results provided in these figures are obtained by employing the sensor model parameters, as provided in Tables 5.1 and 5.2, and the filter parameters, as provided in Table 5.4. Lastly, the simulation parameters used to obtain these results are the ClearSpace-1 parameters defined in Table 3.1.

When considering the navigation errors in the inertial position estimate, as presented in Figure 5.3, it can be seen that the estimation error remains bounded by the error covariance curves, provided by the black dotted lines, for all three position components. Furthermore, the error magnitudes remains well within the 2 cm accuracy level required by **REQ_S_C_16.1**. When subsequently considering the navigation errors of the inertial velocity estimation, as provided in Figure 5.4, it can again be seen that the estimation error of all three components remain nicely bounded by the error covariance curves. Moreover, also their magnitudes remain below the required 2 mm/s accuracy of **REQ_S_C_16.2**. This high accuracy in the inertial orbit determination system is mostly due to the high performance GPS receiver, characterised by the low noise values for the range and range rate measurements. These noise values are, however, necessary to achieve the strict terminal navigation accuracy requirement established in **REQ_S_C_16**. This is in contrast to the values of the clock bias and clock drift errors in the GPS measurements, given that their specific magnitude is not so important for the obtainable navigation accuracy. This is because the UKF algorithm can achieve a truthful estimation of these error parameters. However, figures are omitted for their estimation, as results comparable to Figures 5.3 and 5.4 are obtained, where the error value remains bounded by the respective covariance curves. Therefore, it does not provide sufficient addition to the current discussion.

Note that the erratic behaviour of the covariance curves of the inertial position estimate is likely to be caused by the noise in the GPS range measurements, whereas this is not observable for the inertial velocity covariances. This is because, during velocity estimation, the GPS data is combined

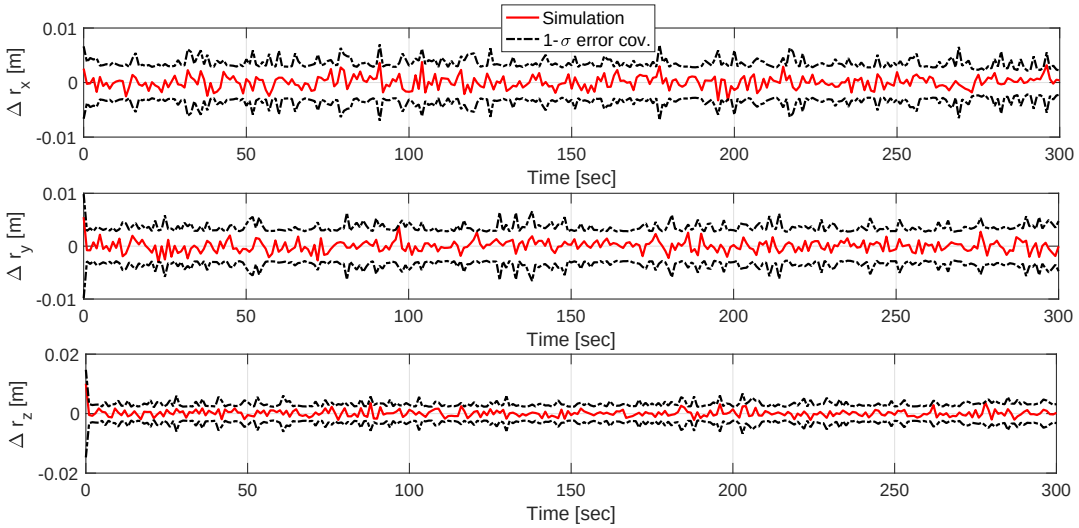


Figure 5.3: Navigation error in inertial position estimate provided by Unscented Kalman filter, when compared to the true simulated inertial position

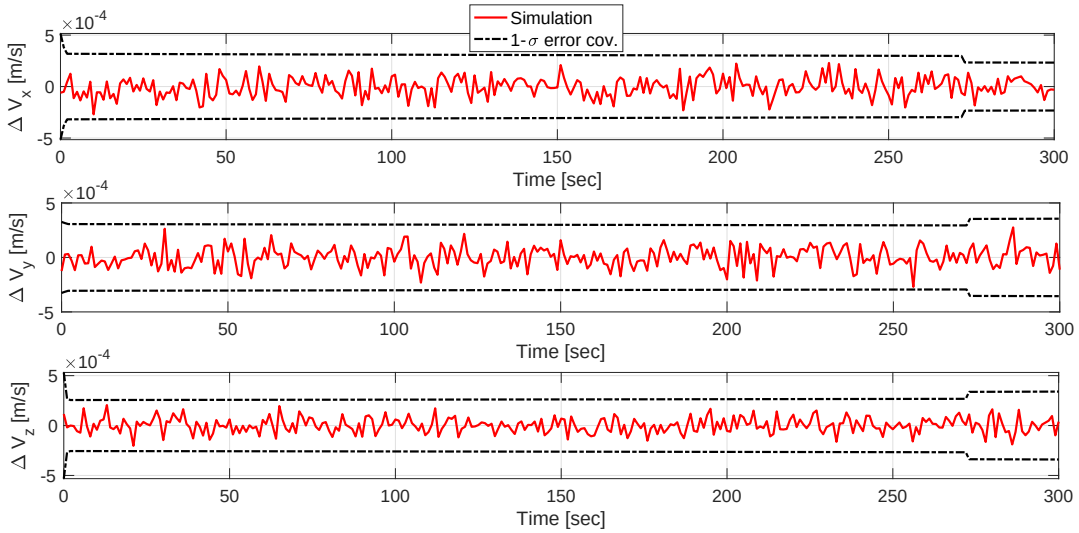


Figure 5.4: Navigation error in inertial velocity estimate provided by Unscented Kalman filter, when compared to the true simulated inertial velocity

with the accelerometer measurements, reducing the estimation errors and smoothing the covariance curves. However, further analysis should be performed to be able to make a conclusive statement and provide a potential fix. Nevertheless, due to time constraints, the position estimation is implemented as is, given that sufficient navigation accuracy can be obtained throughout the entire simulation duration.

Moreover, discrete jumps in the covariance curves of the inertial velocity estimation are observed. These jumps can, however, be correlated to the GDOP of the constellation in view of the GPS receiver, where a change in this parameter results in a corresponding change of the estimation error covariances. This is because, with a changing GDOP, the obtainable GPS measurements converge or diverge more from one consolidated translational state estimate, such that the UKF algorithm has more, respectively less, confidence in the obtainable navigation solution. Equivalently, the magnitudes of the respective velocity covariance curves change. Effectively, this provides a kick to the filter, ensuring that the Kalman gain maintains a sufficiently large magnitude, such that the calculated solution does not converge to incorrect values. Furthermore, when closely inspecting the covariance curves of the inertial position estimation, similar jumps can be observed and follow from the same rationale as provided for the velocity estimation.

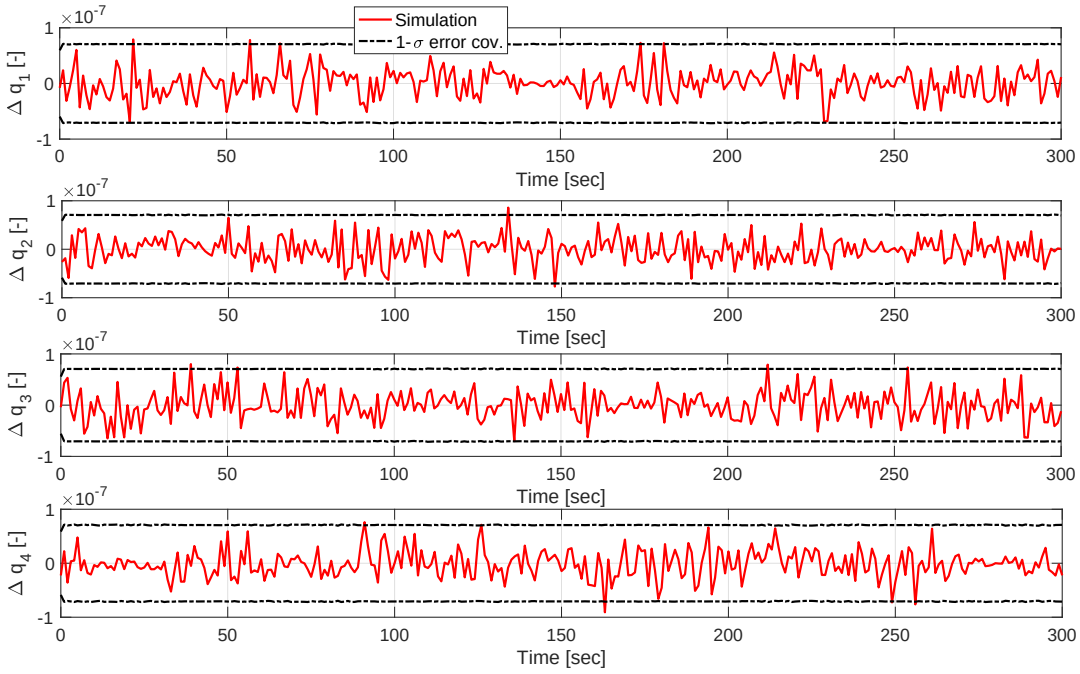


Figure 5.5: Navigation error in inertial attitude estimate provided by Unscented Kalman filter, when compared to the true simulated inertial attitude

Table 5.8: Maximum inertial navigation errors

Max. Navigation Error	Value	Max. Navigation Error	Value
position [m]	0.012	attitude [deg]	$2 \cdot 10^{-5}$
velocity [m/s]	$5 \cdot 10^{-4}$	angular velocity [deg/s]	0.003

When subsequently investigating the navigation errors of the inertial attitude estimation, as presented in Figure 5.5, it can be observed that they remains bounded by the error covariance curves for all four attitude components. Furthermore, although not directly visible as the attitude is provided in terms of quaternion components, it can be seen that the error remains far below the 2 deg accuracy requirement of **REQ_S_C_16.3**. This is because the implemented star trackers have an extremely low noise value, as given in Table 5.3. Furthermore, the navigation errors of the inertial angular velocity estimate, as provided in Figure 5.6, also remain approximately within the bounds imposed by the error covariances. Moreover, their error components remain far beneath the required 2 deg/s accuracy of **REQ_S_C_16.4**, and are achieved by the addition of pseudo-measurements of the chaser's inertial angular velocity vector. Obtaining this low noise level in the pseudo-measurements based on quaternion derivatives is, however, only possible due to the high accuracy present in the star tracker measurements.

A summary of the maximum navigation errors for the inertial state estimation of the chaser spacecraft is presented in Table 5.8. From this table it can be determined that the inertial navigation system is capable of obtaining a navigation solution that satisfies the terminal accuracy requirements provided in **REQ_S_C_16.1** to **REQ_S_C_16.4**. This therefore indicates that the design parameters of the inertial navigation senors and UKF algorithm are appropriately chosen, as they result in obtaining a navigation solution up to the required precision. Hence based on this discussion, it can be concluded that the inertial navigation system correctly integrates the UKF algorithm with the navigation sensors and it can be assumed that the inertial navigation system is verified.

Second, a verification of the implemented relative navigation system must be performed. Note again that no sensor models are currently implemented, nor that an actual filter algorithm is built. Instead, the relative navigation system is artificially implemented by adding percentage-dependent zero-mean white noise parameters to the true relative state. To verify this percentage dependency of the estimation error, a straightforward test is performed where a generic relative X-position estimate is obtained from the VBN system when providing a sequenced step input and a ramp input. Results of this verification step

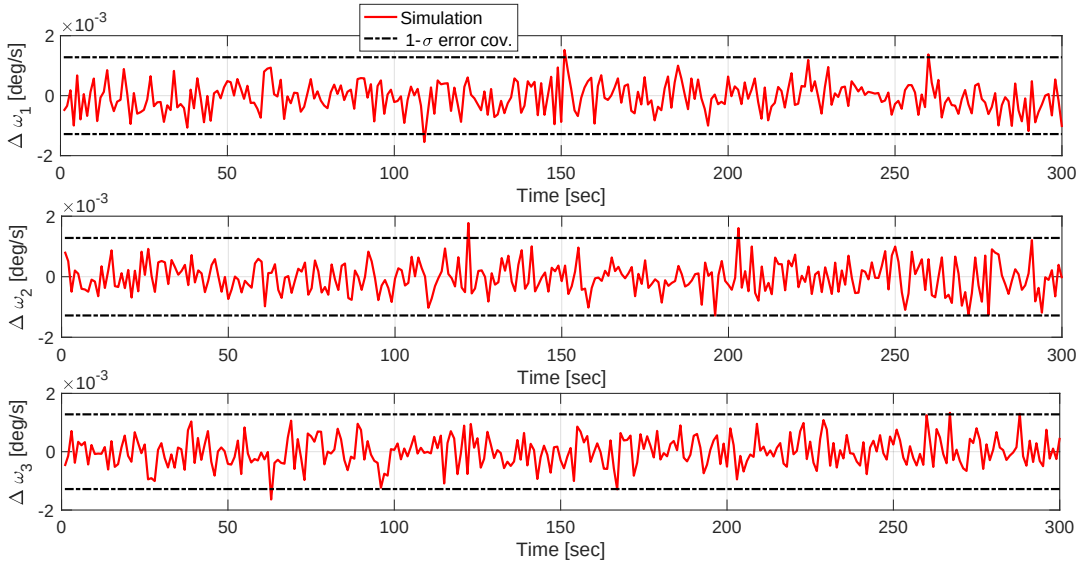


Figure 5.6: Navigation error in inertial angular velocity estimate provided by Unscented Kalman filter, when compared to the true simulated inertial angular velocity

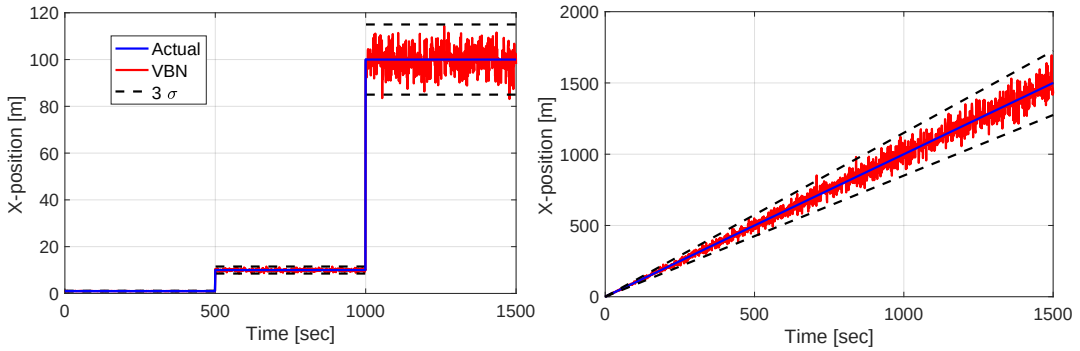


Figure 5.7: Generic X-position estimate provided by vision-based navigation system with corresponding error bounds, for a step signal in X-position (left) and for a ramp signal in X-position (right). Note that for reproducibility reasons, a seed equal to 12345 is employed.

are provided in Figure 5.7, for which it can be seen that the left side provides the step input, whereas the right side presents the ramp input. This plot provides the true input signal to the VBN system, indicated by the blue curves, the estimated relative state, represented by the red curve, and the corresponding error bounds on the state estimation, provided by the black dotted lines. For the error calculation, a normal distribution with a zero mean and a standard deviation equal to $p_{\%} = 5\%$ of the input value is used. This implies that 99.7% of all errors must be located within plus and minus three standard deviations of the input value. Therefore, error bounds on the 3-σ level are provided in Figure 5.7. For both the sequenced step input and the ramp input, the error in the X-position estimate is proportional to the provided input value, as well as bounded by the 3-σ curves. Therefore, it can be concluded that the percentage-dependency of the VBN system is implemented correctly.

Subsequently the VBN system is applied to estimate the relative translational and rotational state of the chaser spacecraft with respect to its target. To do so, a simulation is performed where the VBN parameters of Table 5.5 are used, along with the simulation parameters defined in Table 3.1. Note, however, that one difference with respect to this table is given by the angular velocity of the target spacecraft, which is defined as 3.5 deg/s over all three axes, rather than zero. This results in free-drifting trajectories for both target and chaser spacecraft, for which the simulation indicates that the relative navigation system is capable of providing a state estimate within the 3-σ bounds of the predefined error percentage. However, the obtained results are qualitatively equivalent to those provided in Figure 5.7. Therefore, for brevity purposes, it has been decided to omit this figure. Instead, a quantification of the

Table 5.9: Maximum relative navigation errors

Max. Relative State	**Value**	**Max. Relative State**	**Value**
position [m]	20.011	attitude [deg]	278
velocity [m/s]	0.0215	angular velocity [deg/s]	6.1
Max. Navigation Error	**Value**	**Max. Navigation Error**	**Value**
position [m]	0.128	attitude [deg]	0.4575
velocity [m/s]	$2 \cdot 10^{-4}$	angular velocity [deg/s]	0.042

Table 5.10: Maximum integrated navigation system errors

Max. Hill Error Chaser	**Value**	**Max. Inertial Error Target**	**Value**
position [m]	0.009	attitude [deg]	0.3254
velocity [m/s]	$1 \cdot 10^{-4}$	angular velocity [deg/s]	0.031

maximum relative navigation errors is presented in Table 5.9. From the magnitude of the error levels it can be concluded that the VBN system is capable of obtaining a navigation solution within 1% of the true relative state, as prescribed by **REQ_S_C_16**. Therefore, the design parameters of the relative navigation system are appropriately chosen and it can be assumed that the overall VBN system is verified.

Lastly, the integration of the relative navigation system with the inertial navigation system is verified. This integration is tested by analysing the navigation errors of the Hill vector estimate for the chaser spacecraft and the inertial rotational state estimation error for the target spacecraft. To obtain the Hill vector of the chaser, the inertial state of the chaser and target spacecraft are needed, where the former is directly obtainable from the UKF algorithm, whereas the latter is obtained when combining the inertial translational state of the chaser, from the UKF, with the inertial relative state of the target, from the VBN. Furthermore, to obtain the inertial rotational state of the target, the inertial rotational state of the chaser needs to be combined with the relative state of the target. Therefore, these two test cases asses the integration of the relative navigation system with the inertial navigation system. For this analysis, a simulation is performed, where the implemented inertial and relative navigation system parameters are as defined above. Furthermore, the simulation parameters of Table 3.1 are used, but the initial angular velocity of the target spacecraft is altered to $(3.5, 3.5, 3.5)^T$ deg/s and the initial position of the chaser spacecraft is altered to $(4975539, -782642, -4920254)^T$ m. This implies that the chaser is relocated to the negative V-bar, 1.35 m behind the target. This is done because this new position is similar to the capture position but located on the V-bar, as it provides a stable hold point. Therefore, for simplicity reasons, it is possible to assess the terminal navigation accuracy, without having to provide an orbit control system that can track this capture point tumbling along with the target spacecraft. Nevertheless, even though the chaser is positioned closer to its target, the resulting navigation error histories are qualitatively identical to the results provided for the separate navigation system verifications. Therefore, additional plots are omitted and only a summary on the maximum terminal navigation errors is provided, as given in Table 5.10. From this table it can be determined that the translational Hill state and the inertial rotational state satisfy the terminal navigation accuracy requirements given by **REQ_S_C_16.1** to **REQ_S_C_16.4**. Hence, it can be concluded that the integrated navigation system is verified and ready for implementation in the GNC system.

6
Control

The current chapter provides information regarding the control function of the GNC system. Section 6.1 discusses the need for a control system onboard the ClearSpace-1 satellite and elaborates the functions it will perform. To provide control over the chaser spacecraft, it is equipped with a number of control actuators, which are discussed in Section 6.2. Furthermore, the control algorithm that will command these actuators is elaborated in general terms in Section 6.3, after which its implementation onboard the ClearSpace-1 spacecraft is discussed in Section 6.4. Lastly, Section 6.5 provides the verification procedures that were followed to ensure a correct functioning of the developed control system.

6.1. Control Algorithm

The control system of the ClearSpace-1 satellite is required to calculate the commands needed to maintain attitude synchronisation between target and chaser spacecraft during the close-range rendezvous, as established by **REQ_M_5**. This attitude synchronisation is ensured by removing discrepancies between the estimated attitudes of the chaser and target spacecraft, both obtained from the navigation system. The control system instituted onboard the ClearSpace-1 satellite will thus only have to control the rotational motion of the chaser spacecraft, whereas the translational motion is controlled by executing the commands provided by the guidance system, as previously established in Chapter 4. This implies that the translational and rotational motion of the chaser can, to a certain extent, be decoupled, which reduces both the complexity of the guidance system and the control system. However, to ensure this decoupling, an omnidirectional thrust system is required, as will be discussed in the next section on control actuators.

6.2. Control Actuators

To control the state of the chaser spacecraft, the ClearSpace-1 satellite is equipped with two types of control actuators. Control over the translational state, *i.e.*, execution of the control commands calculated by the guidance system, is performed by a reaction control thruster system. Control over the rotational state, *i.e.*, execution of the control commands calculated by the attitude synchronisation system, is performed by a reaction wheel system. An elaboration on each of these systems is provided below in Subsection 6.2.1 and 6.2.2, respectively.

6.2.1. Reaction Control Thrusters

The reaction control thrusters of the ClearSpace-1 satellite have the sole purpose of controlling its translational motion. To ensure that this decoupling of the translational and rotational control can be established, it is necessary that the chaser spacecraft is equipped with a reaction control system that can provide omnidirectional thrust, effectively implying that the commanded thrust can be provided independent of the current attitude of the spacecraft. Therefore, it is decided to use a configuration with 24 reaction control thrusters as provided in Figure 6.1. This specific thruster configuration is taken from the design of the chaser spacecraft that was established during the e.Deorbit study, which provided the predecessor to the current ClearSpace-1 study.

When considering the thruster model for the reaction control thrusters, it must be noted that these engines are non-throttled, implying that the thruster can either provide zero or maximum thrust, and no other value within this range. It is chosen to use non-throttled thrusters as it is more realistic for an actual reaction control system. However, having a non-throttled thruster limits the controllability of the chaser spacecraft. Therefore, to increase control effectiveness, an algorithm is developed that modulates the

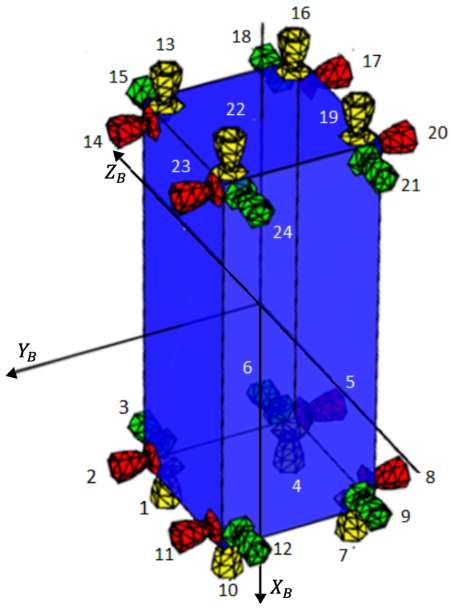


Figure 6.1: Reaction control thruster system configuration, adapted from (Habets, 2015)

pulse-width during which the reaction control thrusters are fired, to be able to provide a wider thrust range. Furthermore, to increase the efficiency of the control system onboard the ClearSpace-1 satellite, the pulse-width modulation algorithm will also allocate the commanded control actions over the different reaction control thrusters. This allocation is formulated as an optimisation problem that minimises the discrepancy between the commanded and provided control actions. Details concerning this pulse-width modulation algorithm are, however, provided in Section 6.4.

A thruster model for the attitude control thrusters is provided in Equation (6.1).

$$T_{RCT} = b_{RCT} + T_{nom} + v_{RCT} \tag{6.1}$$

In this equation, T_{RCT} represents the output thrust delivered by a single reaction control thruster, whereas b_{RCT} represents the bias factor, T_{nom} the nominal thrust of the engine and v_{RCT} the noise in the thruster output. Note that it is assumed that the thrust noise is made up of random white noise parameters, which have a Gaussian normal distribution with zero mean and a specific standard deviation. Furthermore note that, for simplicity reasons, the current thruster model does not incorporate a time delay in the thruster response. This implies that the control command is instantly provided by the thruster and any latency is omitted. Moreover, note that the model provided in Equation (6.1) is applied to each of the 24 thrusters in the reaction control system.

6.2.2. Reaction Wheels

Reaction wheels, also referred to as momentum wheels, constitute the actuators of the attitude control system onboard the ClearSpace-1 spacecraft. They operate as simple flywheels that store rotational energy when spinning. As the angular momentum of the entire satellite system, *i.e.*, the satellite body and the reaction wheel assembly, must be conserved, the spinning wheel causes the spacecraft to rotate in a direction opposite to that of the wheel. By using a minimum of three reaction wheels that are positioned at right angles with respect to one another, the attitude of the spacecraft can effectively be controlled. However, for redundancy purposes, four reaction wheels are generally arranged in a pyramid configuration that can maintain control over the spacecraft, even when one reaction wheel breaks. The angular momentum of a reaction wheel can be changed by changing its rotational velocity. By doing so, a momentum exchange between the reaction wheel and the satellite body will occur. This momentum exchange can be explained when considering Euler's equation of rotational motion, as given by Equation (6.2). In this equation, $\mathbf{M}_{ext}$ represents the external moment acting on the satellite, $\dot{\mathbf{h}}_{cm}$ the angular momentum of the entire satellite system around its centre of mass and $\boldsymbol{\omega}_{sat}$ the angular velocity vector of the satellite body.

$$\mathbf{M}_{ext} = \dot{\mathbf{h}}_{cm} + \boldsymbol{\omega}_{sat} \times \mathbf{h}_{cm} \tag{6.2}$$

The angular momentum of the entire satellite system, $\mathbf{h}_{cm}$, is composed of the angular momentum of the satellite body plus the angular momentum of the reaction wheel assembly, as can be seen in Equation (6.3). In this equation, the angular momentum of the satellite body is given by the product of inertia tensor of the satellite, $\mathbf{I}_{sat}$, and the angular velocity vector of the satellite, $\boldsymbol{\omega}_{sat}$. Furthermore, $\mathbf{h}_w$ represents the angular momentum vector of the entire reaction wheel assembly.

$$\mathbf{h}_{cm} = \mathbf{I}_{sat}\boldsymbol{\omega}_{sat} + \mathbf{h}_w \tag{6.3}$$

Combining Equations (6.2) and (6.3) results in Equation (6.4), from which it can be seen that a control moment can be created on the satellite by changing the angular momentum of the reaction wheel assembly.

$$\mathbf{M}_{ext} = \mathbf{I}_{sat}\dot{\boldsymbol{\omega}}_{sat} + \dot{\mathbf{h}}_w + \boldsymbol{\omega}_{sat} \times (\mathbf{I}_{sat}\boldsymbol{\omega}_{sat} + \mathbf{h}_w) \tag{6.4}$$

As the angular momentum vector of the complete reaction wheel assembly is given by Equation (6.5), it can be observed that this angular momentum vector can be changed by varying the angular velocity of the reaction wheel assembly, given by $\boldsymbol{\omega}_w$.

$$\mathbf{h}_w = \mathbf{I}_w\boldsymbol{\omega}_w \tag{6.5}$$

When using three orthogonally positioned reaction wheels, the angular velocity vector of the complete reaction wheel assembly is simply given by Equation (6.6), where ω_{RW_x}, ω_{RW_y} and ω_{RW_z} represent the angular velocities of the reaction wheels in X, Y and Z-direction, respectively. In case a pyramid configuration is considered, the angular velocities of the four reaction wheels must be mapped to the three orthogonal axes of the satellite's body frame based on their respective location within this body frame. This mapping is, however, currently omitted given that it depends on the specific geometry of the reaction wheel system.

$$\boldsymbol{\omega}_w = \begin{pmatrix} \omega_{RW_x} \\ \omega_{RW_y} \\ \omega_{RW_z} \end{pmatrix} \tag{6.6}$$

A model for the dynamics of a each reaction wheel in the assembly is provided by Markley and Crassidis (2014) and given below in Equation (6.7). Note that in this equation, $\boldsymbol{\omega}_{RW}$ represents the rotational velocity vector of a single reaction wheel, $\mathbf{I}_{RW}$ its inertia tensor and $\mathbf{M}_{RW}$ the momentum vector generated by the reaction wheel.

$$\boldsymbol{\omega}_{RW}(t) = \int_{t_0}^{t} \mathbf{I}_{RW}\mathbf{M}_{RW}\,dt + \boldsymbol{\omega}_{RW}(t_0) \tag{6.7}$$

Furthermore, an actuator model relating the moment provided by the reaction wheel, given by $\mathbf{M}_{RW}$, and the commanded moment, given by $\mathbf{M}_{cmd}$, is provided below in Equation 6.8. Note that this actuator model, as developed by Carrara and Kuga (2013), implements a misalignment matrix for the reaction wheel system, containing scaling factors given by the s-terms and the misalignment factors given by the m-terms, as well as two friction components. The first friction term, viscous friction, is dependent on the angular velocity of the reaction wheel. Therefore, it is given by $k_v\boldsymbol{\omega}_{RW}$, with k_v being the viscous friction coefficient. The second friction term, Coulomb friction, is a constant friction in the reaction wheel which only depends on the direction of rotation. Therefore, it is given by $k_c\text{sign}(\boldsymbol{\omega}_{RW})$, with k_c being the Coulomb friction coefficient and sign() being the signum function.

$$\mathbf{M}_{RW} = \begin{bmatrix} 1 + s_{RW,x} & m_{RW,xy} & m_{RW,xz} \\ m_{RW,yx} & 1 + s_{RW,y} & m_{RW,yz} \\ m_{RW,zx} & m_{RW,zy} & 1 + s_{RW,z} \end{bmatrix} (\mathbf{M}_{cmd} - k_v\boldsymbol{\omega}_{RW} + k_c\text{sign}(\boldsymbol{\omega}_{RW})) \tag{6.8}$$

To provide the commanded moment, $\mathbf{M}_{cmd}$, the reaction wheel will obtain a certain angular velocity, provided by the electric motor within the reaction wheel. Hence to make the motor spin, the reaction wheel draws a current from the electric power system and a direct relation between the drawn current and delivered moment is provided in Equation (6.9), where $\mathbf{k}_m$ represents the motor constant vector containing the motor constants of all reaction wheels and $\mathbf{I}_{elec}$ the current vector containing the electric current drawn by all reaction wheels.

$$\mathbf{M}_{cmd} = \mathbf{k}_m \mathbf{I}_{elec} \tag{6.9}$$

Combining Equations (6.8) and (6.9) result in Equation (6.10), which represents the implemented actuator model for the reaction wheel system.

$$\mathbf{M}_{RW} = \begin{bmatrix} 1 + s_{RW,x} & m_{RW,xy} & m_{RW,xz} \\ m_{RW,yx} & 1 + s_{RW,y} & m_{RW,yz} \\ m_{RW,zx} & m_{RW,zy} & 1 + s_{RW,z} \end{bmatrix} \left(\mathbf{k}_m \mathbf{I}_{elec} - k_v \boldsymbol{\omega}_{RW} + k_c \text{sign}(\boldsymbol{\omega}_{RW}) \right) \tag{6.10}$$

Note that the previously discussed reaction control thrusters can also, and in fact are frequently used as attitude control system onboard spacecraft. However, benefits of using reaction wheels as attitude control actuators are given by the fact that these can provide the required torques internally and do not require external resources such as propellant. Moreover, when the rotational velocity of the reaction wheels can be controlled accurately, the attitude of the spacecraft can be controlled to very high precision. Such accurate control over the chaser's attitude is then again of paramount importance for any ADR mission requiring attitude synchronisation with its target prior to rendezvous and docking. Therefore, the current study chooses to use a simple reaction wheel assembly consisting of three orthogonal wheels as the complete attitude control system, as can be seen in Figure 6.2. Note that a pyramid scheme would be more realistic for an actual mission when considering the robustness of the satellite, but is currently omitted due time constraints. Nevertheless, changing the reaction wheel assembly from an orthogonal to a pyramid scheme is expected to change the equations of the implemented model, without changing much about the performance of the reaction control system, when appropriate scaling is applied to the reaction wheels. Therefore it justifies the use of three orthogonal wheels in the current study. Lastly, it should be noted that there can be some interaction between the reaction control thrusters and reaction wheel assembly, when the thrusters perform momentum unloading of the reaction wheels. However, for simplicity reasons, this interaction is currently omitted.

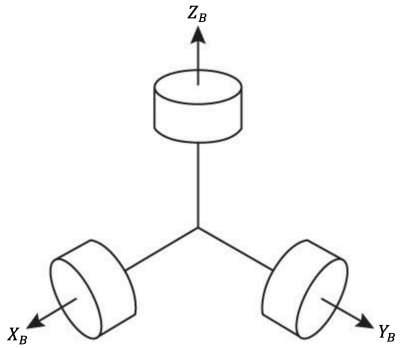


Figure 6.2: Reaction wheel system configuration, adapted from (Narkiewicz et al., 2020)

6.3. INDI Control

As spacecraft rotational dynamics are an intrinsic nonlinear problem, especially when large attitude changes are commanded, linear control techniques will not suffice to control the ClearSpace-1 spacecraft. Therefore, a nonlinear approach must be adopted. The current study implements Incremental Nonlinear Dynamic Inversion (INDI) control as part of the GNC system of the chaser satellite. This INDI controller is a modified version of the more conventional Nonlinear Dynamic Inversion (NDI) controller, with increased robustness when compared to the baseline NDI controller.

The conventional NDI controller applies a system inversion based on a state feedback loop which cancels out all nonlinearities in the system model, such that the resulting closed loop system becomes

linear (Acquatella et al., 2012). This implies that one can effectively apply straightforward linear control techniques to a nonlinear problem. To perform such a feedback linearisation, full knowledge of the entire system is required, *i.e.*, the real model of the system has to be known exactly. If this is not the case, some nonlinearities remain in the system after inversion and one will not be able to apply a linear control technique. To decrease the dependency of the controller on model accuracy, the INDI controller was developed, which applies a sensor-based methodology in contrast to the model-based methodology applied in the NDI controller. The INDI controller obtains detailed knowledge on the system through measurements of the vehicle's state and control actuator output, which can be used in its feedback loop to cancel out the nonlinearities in the uncertain model. This means that a simplified model onboard the spacecraft is sufficient for adequate control, given that accurate measurements are present at a sufficiently high sampling rate. This significantly increases the robustness of the controller when compared to conventional NDI and allows for the application of linear control techniques to real nonlinear systems. Therefore, it is expected that the INDI controller will outperform other nonlinear control techniques due to its combination of high robustness with straightforward application of linear designing methodologies. Furthermore, the author has previous experience with implementing an INDI controller to spacecraft attitude control systems. Hence, it can safely be concluded that INDI will be an adequate control technique for the considered reference mission.

The discussion on the governing principles behind INDI controllers is based on the information provided by Acquatella et al. (2012). A general representation of the nonlinear MIMO-system to which the INDI controller will be applied is given in Equations (6.11) and (6.12).

$$\dot{\mathbf{x}} = \mathrm{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \tag{6.11}$$

$$\mathbf{y} = \mathrm{h}(\mathbf{x}) \tag{6.12}$$

Note that in these equations, $\mathbf{x}$ represents the state vector, $\mathbf{u}$ the control input vector, $\mathbf{y}$ the system output vector, f() and h() smooth vector fields and $\mathbf{G}()$ a matrix whose columns are smooth vector fields. To be able to apply the INDI controller to this general MIMO-system, it is assumed that the number of inputs in $\mathbf{u}$ is equal to the number of outputs in $\mathbf{y}$ to avoid control allocation problems. Additionally, it is assumed that the system is affine in input such that no nonlinear solvers are required to solve for this input. Moreover, it is assumed that the system contains state output such that $\mathbf{y}$ simply equals $\mathbf{x}$, as given in Equation (6.13).

$$\mathbf{y} = \mathbf{x} \tag{6.13}$$

This implies that the dynamics of the system can be rewritten to the form provided in Equation (6.14).

$$\dot{\mathbf{y}} = \dot{\mathbf{x}} = \mathrm{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \tag{6.14}$$

Note that this system provides the dynamics that will have to be inverted to cancel out all non-linearities in the model of the controller. However, in an INDI controller, the system dynamics are approximated using a first-order Taylor series expansion evaluated in the neighbourhood of the state $\mathbf{x}_0$ and input $\mathbf{u}_0$, as can be seen in Equation (6.15).

$$\dot{\mathbf{y}} \approx \mathrm{f}(\mathbf{x}_0) + \mathbf{G}(\mathbf{x}_0)\mathbf{u}_0 + \frac{\partial}{\partial \mathrm{x}}\left[\mathrm{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}\right]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}}(\mathbf{x}-\mathbf{x}_0) + \frac{\partial}{\partial \mathrm{u}}\left[\mathrm{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}\right]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}}(\mathbf{u}-\mathbf{u}_0)$$

$$\dot{\mathbf{y}} \approx \mathrm{f}(\mathbf{x}_0) + \mathbf{G}(\mathbf{x}_0)\mathbf{u}_0 + \frac{\partial}{\partial \mathrm{x}}\left[\mathrm{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}\right]\Big|_{\substack{\mathbf{x}=\mathbf{x}_0 \\ \mathbf{u}=\mathbf{u}_0}}(\mathbf{x}-\mathbf{x}_0) + \mathbf{G}(\mathbf{x}_0)(\mathbf{u}-\mathbf{u}_0) \tag{6.15}$$

By using the definition of the state derivative provided in Equation (6.11), the corresponding state rate at the current instant can be identified as provided by Equation (6.16).

$$\dot{\mathbf{x}}_0 = \mathrm{f}(\mathbf{x}_0) + \mathbf{G}(\mathbf{x}_0)\mathbf{u}_0 \tag{6.16}$$

Furthermore, two auxiliary matrices $\mathbf{A}_0$ and $\mathbf{B}_0$ are defined and given in Equations (6.17) and (6.18), respectively.

$$\mathbf{A}_0 = \frac{\partial}{\partial \mathbf{x}} \left[\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \right] \Big|_{\substack{\mathbf{x} = \mathbf{x}_0 \\ \mathbf{u} = \mathbf{u}_0}} \tag{6.17}$$

$$\mathbf{B}_0 = \frac{\partial}{\partial \mathbf{u}} \left[\mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \right] \Big|_{\substack{\mathbf{x} = \mathbf{x}_0 \\ \mathbf{u} = \mathbf{u}_0}} = \mathbf{G}(\mathbf{x}_0) \tag{6.18}$$

Note that these auxiliary matrices contain the partial derivatives of Equation (6.15) and are defined to simplify the notation, as can be seen in Equation (6.19).

$$\dot{\mathbf{y}} \approx \dot{\mathbf{x}}_0 + \mathbf{A}_0(\mathbf{x} - \mathbf{x}_0) + \mathbf{B}_0(\mathbf{u} - \mathbf{u}_0) \tag{6.19}$$

Equivalently, this equation can be expressed using the incremental control command $\delta\mathbf{u} = (\mathbf{u} - \mathbf{u}_0)$ as can be seen in Equation (6.20).

$$\dot{\mathbf{y}} \approx \dot{\mathbf{x}}_0 + \mathbf{A}_0(\mathbf{x} - \mathbf{x}_0) + \mathbf{B}_0\delta\mathbf{u} \tag{6.20}$$

Note that Equation (6.20) provides an explicit relation between the output of the system, $\mathbf{y}$, and the incremental input to the system, $\delta\mathbf{u}$. Subsequently taking the inverse of this relation allows to obtain the feedback linearisation loop that linearises the system, as given in Equation (6.21). Note that, to be able to perform this system inversion, the required model knowledge is provided by $\mathbf{B}_0$ and is not allowed to be singular. The singularity of this matrix will, however, be assessed later, when applying the INDI methodology as attitude controller of the ClearSpace-1 satellite.

$$\delta\mathbf{u} = \mathbf{B}_0^{-1} \left[\dot{\mathbf{y}} - \left(\dot{\mathbf{x}}_0 + \mathbf{A}_0(\mathbf{x} - \mathbf{x}_0) \right) \right] \tag{6.21}$$

In case the virtual control input, $\mathbf{v}$, is defined as the rate of the system output, as given in Equation (6.22), the incremental control input that linearises the system can be obtained through Equation (6.23).

$$\dot{\mathbf{y}} = \mathbf{v} \tag{6.22}$$

$$\delta\mathbf{u} = \mathbf{B}_0^{-1} \left[\mathbf{v} - \left(\dot{\mathbf{x}}_0 + \mathbf{A}_0(\mathbf{x} - \mathbf{x}_0) \right) \right] \tag{6.23}$$

This input transforms the nonlinear system dynamics provided in Equation (6.15) to the linear system defined in Equation (6.24).

$$\dot{\mathbf{y}} = \dot{\mathbf{x}}_0 + \mathbf{A}_0(\mathbf{x} - \mathbf{x}_0) + \mathbf{B}_0 \left[\mathbf{B}_0^{-1} \left[\mathbf{v} - \left(\dot{\mathbf{x}}_0 + \mathbf{A}_0(\mathbf{x} - \mathbf{x}_0) \right) \right] \right] = \mathbf{v} \tag{6.24}$$

When the time step is taken sufficiently small and the control update rate is sufficiently high, the value of the current state $\mathbf{x}$ approaches the value of the state at the linearisation point $\mathbf{x}_0$, such that $\mathbf{A}_0(\mathbf{x} - \mathbf{x}_0) << \mathbf{B}_0\delta\mathbf{u}$ and can thus be neglected. This implies that the linearising control can be defined as given by Equation (6.25).

$$\delta\mathbf{u} = \mathbf{B}_0^{-1} \left[\mathbf{v} - \dot{\mathbf{x}}_0 \right] = \mathbf{G}(\mathbf{x}_0)^{-1} \left[\mathbf{v} - \dot{\mathbf{x}}_0 \right] \tag{6.25}$$

This linearising control law results in incremental control commands, *i.e.*, the incremental command must be added to the reference command of the linearisation point to obtain the full control command, as can be seen in Equation (6.26).

$$\mathbf{u} = \mathbf{u}_0 + \delta\mathbf{u} \tag{6.26}$$

This basically means that $\delta\mathbf{u}$ is the change in control command required to generate the virtual control input $\mathbf{v}$ and obtain a linearised system. Furthermore note that $\dot{\mathbf{x}}_0$ corresponds to the derivative of the state vector at the linearisation point and thus corresponds to the reference command $\mathbf{u}_0$. Effectively, the controller has been split into two loops, where the outer loop contains a linearised system controlled by virtual control input $\mathbf{v}$, and the innerloop is controlled by the control input $\mathbf{u}$ chosen such

that the outerloop system becomes linear. The values for the reference command, $\mathbf{u}_0$, and reference state derivative, $\dot{\mathbf{x}}_0$, are obtained from direct measurements of the control output and state derivative, respectively. This makes the control law more dependent on sensory aspects such as noise, bias or misalignment in the measurements. However, when these measurements can be provided at a sufficiently high rate, the control law becomes less dependent on the exact system dynamics, as $\mathbf{A}_0$ and $f(\mathbf{x})$ are no longer required to linearise the system. Nevertheless, the exact system dynamics will still be reflected in the control law through measurements of $\dot{\mathbf{x}}_0$. This is exactly what makes the INDI controller such a robust system.

6.4. Control System Design

The current section details the control system design for the ClearSpace-1 mission. First, Subsection 6.4.1 presents the specifics of the developed control algorithm. Subsequently, Subsection 6.4.2 contains the details concerning the developed pulse-width modulation and control allocation algorithm. Lastly, Subsections 6.4.3 and 6.4.4 elaborate the control actuator models and the sensor models used by the control system, respectively.

6.4.1. Attitude Controller

The control algorithm developed for the ClearSpace-1 mission consists actually of two consecutive controllers. This choice has been made based on the work provided by Singh et al. (2019), which states that the use of a full state attitude controller for synchronisation with a tumbling target will be inefficient. This is because an instability is caused by the fact that for complete synchronisation, both the angular velocity and instantaneous attitude must be matched with the debris item. However, errors in the instantaneous attitude synchronisation will lead the controller to corrective actions, causing the chaser to diverge from rate synchronisation. Moreover, errors in rate synchronisation will lead the controller to other corrective actions, causing the chaser to diverge from instantaneous attitude synchronisation. Therefore, simultaneous attitude and angular velocity synchronisation cannot effectively be achieved with a full state controller. Hence, the solution strategy as provided by Singh et al. (2019) is followed, where a straightforward PID controller is used to achieve rate synchronisation within a given threshold, after which full state control is used to perform simultaneous attitude and angular velocity synchronisation. A description of these two control algorithms is provided below.

PID Controller

A linear PID controller is selected as the rate synchronisation controller, to remove the angular velocity error with respect to the tumbling debris item. Note that due to the simplicity of the problem, a linear control technique suffices. This is in contrast with full state control for attitude synchronisation, which constitutes a nonlinear control problem when large changes in attitude are commanded. Consequently, a nonlinear INDI controller is instituted for that.

A basic definition of a PID controller has already been established in Equation (4.50). However, when implementing the controller for angular rate synchronisation, it is required that the thrust command is replaced by a control moment, $\mathbf{M}(t)$, and that the trajectory error, $\mathbf{e}(t)$, is replaced by the angular velocity error. This angular velocity error is given as the difference between the rate of the tumbling debris item, represented by $\boldsymbol{\omega}_T$, and the rate of the chaser spacecraft, represented by $\boldsymbol{\omega}_C$. Therefore, the PID controller calculates the control moment using Equation (6.27). Note that in this equation, the angular velocity vectors of the target and chaser spacecraft are the inertial angular velocity vector estimates obtained from the navigation system.

$$\mathbf{M}(t) = K_p\left(\boldsymbol{\omega}_T(t) - \boldsymbol{\omega}_C(t)\right) + K_i \int \boldsymbol{\omega}_T(t) - \boldsymbol{\omega}_C(t)dt + K_d\left(\dot{\boldsymbol{\omega}}_T(t) - \dot{\boldsymbol{\omega}}_C(t)\right) \tag{6.27}$$

To characterise the control law, the respective gains in Table 6.1 are used. From initial analyses it was found that a simple proportional controller could already provide the necessary control effectiveness for rate synchronisation. The integral gain is omitted, as the threshold value effectively allows a small steady-state error in the output signal. Moreover, the differential gain is also omitted because some initial overshoot is desirable in the output signal to quickly achieve rate synchronisation within the error threshold. It should, however, be noted that due to time constraints, the values of the gains are established from preliminary analysis on the time response of the system to a step input. Therefore, these are by no means optimised, but show sufficient performance for the first analyses performed in

Table 6.1: PID control algorithm settings

Parameter	Value
K_p [-]	50
K_i [-]	0
K_d [-]	0
threshold [deg/s]	0.6
operational frequency [Hz]	100

the current thesis study. Moreover, Table 6.1 provides the threshold level that defines the switching point from PID control to full state INDI control. The first time the magnitude of the relative rate error subceeds this threshold, the control actions of the full state INDI control algorithm are followed. For the magnitude of this threshold, a low value of 0.6 deg/s is selected, as to ensure that the angular velocities are nearly synchronised. Lastly, it can be seen that the operational frequency of the controller is set at 100 Hz. This is done to have a unified operational frequency for the entire control system, given that the outputs provided by the PID controller and INDI controller will be merged in the Simulink environment. Consequently, both must have an equal sampling time driven by the high operational frequency of the INDI controller. Nevertheless, from analyses on the isolated rate controller, it is found that an equivalent response can be obtained using a more realistic operational frequency of 4 Hz. However, due to time constraints, this could currently not be implemented in the functional simulator and is therefore left for future work.

INDI Controller

When applying the INDI technique as full state attitude controller of the chaser, it is given that synchronisation needs to be established with the target. The kinematic and dynamic equations of rotational motion in terms of quaternions have already been provided in Chapter 3, when discussing the motion simulator. However, using a kinematic inversion based on Equation (3.22) in the INDI controller will result in a discontinuity when the rotation angle of the body exceeds 180 degrees. Therefore, the normal quaternion representation provides an ineffective system for the attitude controller and quaternion error kinematics are applied instead, as proposed by Bacon (2012). A mathematical description of these error kinematics is provided in Equation (6.28), where $\delta\mathbf{q}$ and $\otimes$ represent the error quaternion and the quaternion product operator, as previously defined in Equation (3.17). Furthermore, $\boldsymbol{\omega}_{cmd}$ represents the commanded angular velocity vector, whereas $\boldsymbol{\omega}$ represents the actual angular velocity vector of the body. Note that for attitude synchronisation with a tumbling debris item, this commanded angular velocity vector simply equals the estimated angular velocity of the target spacecraft.

$$\dot{\delta\mathbf{q}} = \frac{1}{2}\left(\delta\mathbf{q}\otimes\boldsymbol{\omega}_{cmd} - \boldsymbol{\omega}\otimes\delta\mathbf{q}\right) \tag{6.28}$$

Equivalently to Equation (6.28), the quaternion error kinematics can also be expressed using Equation (6.29), where the matrices $\mathbf{Q}_1$ and $\mathbf{Q}_2$ represent two auxiliary matrices that are defined by Equation (6.30) and (6.31), respectively. Note that this alternative representation is provided, given that it will simplify the system inversion required for the outerloop attitude controller, as will be discussed below.

$$\dot{\delta\mathbf{q}} = \frac{1}{2}\left(\mathbf{Q}_1\boldsymbol{\omega}_{cmd} - \mathbf{Q}_2\boldsymbol{\omega}\right) \tag{6.29}$$

$$\mathbf{Q}_1 = \begin{bmatrix} \delta q_4 & -\delta q_3 & \delta q_2 \\ \delta q_3 & \delta q_4 & -\delta q_1 \\ -\delta q_2 & \delta q_1 & \delta q_4 \\ -\delta q_1 & -\delta q_2 & -\delta q_3 \end{bmatrix} \tag{6.30}$$

$$\mathbf{Q}_2 = \begin{bmatrix} \delta q_4 & \delta q_3 & -\delta q_2 \\ -\delta q_3 & \delta q_4 & \delta q_1 \\ \delta q_2 & -\delta q_1 & \delta q_4 \\ -\delta q_1 & -\delta q_2 & -\delta q_3 \end{bmatrix} \tag{6.31}$$

To allow for more rapid control, the Time Scale Separation principle is applied during the dynamic inversion, as given by Singh et al. (2019). Note that the time scale separation principle can be applied when the system contains slow and fast dynamics, which is the case for the spacecraft attitude dynamics. The attitude control consists of the slow attitude kinematics, whereas the rate control consists of the rapid attitude dynamics. The purpose of applying this time scale separation principle is then to separate the governing equations into two systems, one containing dynamic equations and one containing kinematic equations. Consequently, the system inversion is applied twice, once to each of these two systems, and the two systems are coupled through the input for the outerloop controller, which becomes the reference signal that the innerloop controller must follow. This results in a more effective controller for the spacecraft attitude tracking problem.

When considering the innerloop rate control, its objective is to track a desired angular velocity reference profile, given by $\boldsymbol{\omega}_{\text{des}}$, obtained from the outerloop attitude controller. Hence this innerloop is concerned with the attitude dynamics given by Equation (3.21) and it is evident that the controlled variables equal the angular velocities. Differentiating the output $\mathbf{y}_{\text{in}}$ with respect to time allows to calculate an explicit relation between the output of the system and the incremental input to the system needed to generate the feedback linearisation loop. By setting the rate of the system output equal to the virtual control input, $\mathbf{v}_{\text{in}}$, the incremental control input, $\delta\mathbf{u}_{\text{in}}$, that linearises the system can be obtained and is provided in Equation (6.32). Note that for the system inversion of the innerloop controller, the $\mathbf{G}(\mathbf{x}_{\text{in},0})^{-1}$ matrix simply equals the inertia tensor and can therefore not be singular.

$$\delta\mathbf{u}_{\text{in}} = \mathbf{G}(\mathbf{x}_{\text{in},0})^{-1}\left(\mathbf{v}_{\text{in}} - \dot{\mathbf{x}}_{\text{in},0}\right) = \mathbf{J}\left(\mathbf{v}_{\text{in}} - \dot{\boldsymbol{\omega}}_{\text{meas}}\right) \tag{6.32}$$

This is added to the reference control input, $\mathbf{u}_{\text{in},0}$, to obtain the complete control input, $\mathbf{u}_{\text{in}}$, given in Equation (6.33). Note that for this INDI controller, it is assumed that the time step is taken sufficiently small and the control update rate is sufficiently high such that the definition of the incremental control input, $\delta\mathbf{u}$, can be employed. Furthermore note that the value of the reference input, $\mathbf{u}_{\text{in},0}$, is the measured output of the control actuators at the instance of the angular acceleration measurements, $\dot{\boldsymbol{\omega}}_{\text{meas}}$.

$$\mathbf{u}_{\text{in}} = \mathbf{u}_{\text{in},0} + \delta\mathbf{u}_{\text{in}} \tag{6.33}$$

Lastly, the virtual control law, provided by Equation (6.34), is designed using a simple proportional controller, as used by Acquatella et al. (2012) and Singh et al. (2019). A mathematical description of this control law is provided in Equation (6.35), where the proportional controller with gain $K_{p,\text{in}}$ is used to remove any error between the desired angular velocity, $\boldsymbol{\omega}_{\text{des}}$, and the actual angular velocity, $\boldsymbol{\omega}$.

$$\dot{\mathbf{y}}_{\text{in}} = \dot{\boldsymbol{\omega}} = \mathbf{v}_{\text{in}} \tag{6.34}$$

$$\mathbf{v}_{\text{in}} = K_{p,\text{in}}(\boldsymbol{\omega}_{\text{des}} - \boldsymbol{\omega}) \tag{6.35}$$

When subsequently considering the outerloop attitude controller, its objective is twofold: it generates the required angular velocity profile for the innerloop, $\boldsymbol{\omega}_{\text{des}}$, and tracks the attitude of the target spacecraft to maintain synchronisation. Hence this outerloop is concerned with the quaternion error kinematics given by Equation (6.29) and it is evident that the controlled variable equals the error quaternion vector, $\delta\mathbf{q}$. Differentiating the output, $\mathbf{y}_{\text{out}}$, with respect to time allows to calculate an explicit relation between the output of the system and the input to the system needed to generate the feedback linearisation loop. Note that as the kinematic attitude equation is a given geometrical relation, it does not contain any uncertainties and can therefore effectively be controlled using an NDI controller, rather than an INDI controller. This can be achieved by setting the rate of the system output equal to the virtual control input, $\mathbf{v}_{\text{out}}$, to obtain the control input, $\mathbf{u}_{\text{out}}$, that exactly linearises the system in Equation (6.36). For the system inversion of the outerloop controller, the $\mathbf{G}(\mathbf{x}_{\text{in},0})^{-1}$ matrix equals the inverse of the auxiliary matrix $\mathbf{Q}_2$. However, as this matrix is orthonormal, its inverse is identical to its transpose and is therefore never singular. Furthermore, the value of the $\mathbf{G}(\mathbf{x}_{\text{in},0})^{-1}$ matrix is calculated using the current error quaternion, which is obtained from the navigation system estimates for the attitude quaternions of the chaser and target spacecraft.

Table 6.2: INDI control algorithm settings

Parameter	Value
$K_{p,in}$ [-]	5
$K_{p,out}$ [-]	0.1
operational frequency [Hz]	100

$$\begin{aligned}\mathbf{u}_{out} &= \mathbf{G}(\mathbf{x}_{out,0})^{-1}\left(\mathbf{v}_{out} - f(\mathbf{x}_{out})\right) = -2\left(\mathbf{Q}_2\right)^{-1}\left(\mathbf{v}_{out} - \frac{1}{2}\mathbf{Q}_1\boldsymbol{\omega}_{cmd}\right) \\ &= -\left(\mathbf{Q}_2\right)^{T}\left(2\mathbf{v}_{out} - \mathbf{Q}_1\boldsymbol{\omega}_{cmd}\right)\end{aligned} \tag{6.36}$$

Note again that the commanded angular velocity vector, $\boldsymbol{\omega}_{cmd}$, in Equation (6.36) simply equals the estimated angular velocity vector of the tumbling target. Furthermore, note that this control input of the outerloop attitude controller is used as the design angular velocity profile for the innerloop, $\boldsymbol{\omega}_{des}$, as given in Equation (6.37).

$$\boldsymbol{\omega}_{des} = \mathbf{u}_{out} \tag{6.37}$$

Lastly, the virtual control law provided by Equation (6.38) is designed using a straightforward proportional controller as employed by Singh et al. (2019). A mathematical depiction of this control law is provided in Equation (6.39), where the proportional controller with gain $K_{p,out}$ is used in combination with the quaternion error vector to synchronise the attitude of the chaser spacecraft with the attitude of its target. Note that the orthogonal projection, established by premultiplying with $\mathbf{Q}_1\mathbf{Q}_1^T$, ensures that the control action always remains feasible (Bacon, 2012). Furthermore note that this equation makes use of the parameter $\Delta\mathbf{q}$, which is defined based on the error quaternion, given by $\delta\mathbf{q}$, in Equation (6.40). The definition of $\Delta\mathbf{q}$ depends on the sign of the scalar part of the quaternion error vector, such that the outerloop controller always rotates along the shortest way that reaches the desired attitude (Bacon, 2012).

$$\dot{\mathbf{y}}_{out} = \dot{\delta\mathbf{q}} = \mathbf{v}_{out} \tag{6.38}$$

$$\mathbf{v}_{out} = \mathbf{Q}_1\mathbf{Q}_1^T K_{p,out}\left(\Delta\mathbf{q}\right) \tag{6.39}$$

$$\Delta\mathbf{q} = \begin{cases} \begin{pmatrix}0 & 0 & 0 & 1\end{pmatrix}^T - \delta\mathbf{q} & \text{if } \delta q_4 \geqslant 0 \\ \begin{pmatrix}0 & 0 & 0 & -1\end{pmatrix}^T + \delta\mathbf{q} & \text{if } \delta q_4 < 0 \end{cases} \tag{6.40}$$

Hence to characterise the control law, the inner- and outerloop gains as provided in Table 6.2 are used. Note also here that, due to time constraints, the values for the control gains are established from preliminary analysis on the time response of the system to a step input and are therefore not claimed to be optimal. Furthermore, the operational frequency is provided as 100 Hz. This frequency, as commonly chosen for INDI systems, is necessary to allow for effective linearisation of the system dynamics after inversion. Therefore, the unified operational frequency of the entire control system is driven by the need for a high update rate in the INDI controller.

6.4.2. Pulse-width Modulation and Control Allocation

To achieve omnidirectional thrust, the ClearSpace-1 satellite is equipped with redundant reaction control thrusters, as previously shown in Figure 6.1. However, a redundancy in actuators will give rise to control allocation problems. Therefore, details on the allocation algorithm that is developed for the considered reference mission are provided in the current section. Moreover, as the reaction control thrusters fitted on the ClearSpace-1 satellite will likely be non-throttled, the algorithm will simultaneously apply a pulse-width modulation to their control commands. This pulse-width modulation is needed to increase control effectiveness by widening the thrust range that can be provided by the spacecraft. Hence the allocation procedure is formulated as an optimisation problem that minimises the discrepancy

between the commanded and provided control action, to within the accuracy that can be provided by the modulated thrusters. This can in general terms be described by Equation (6.41), where J_f represents the objective function that is to be minimised, $\mathbf{V}$ the configuration matrix of the control actuators, $\mathbf{u}$ the control input vector provided by the actuators and $\mathbf{u}_{cmd}$ the commanded control vector obtained from the control law, or in this case guidance function (Bodson and Frost, 2011). Moreover, the control input vector that can be provided by the actuators is commonly bounded by some constraints as given in Equation (6.42).

$$\underset{\mathbf{u}}{\text{minimise}} \quad J_f = ||\mathbf{V}\mathbf{u} - \mathbf{u}_{cmd}|| \tag{6.41}$$

$$\mathbf{u}_{min} \leqslant \mathbf{u} \leqslant \mathbf{u}_{max} \tag{6.42}$$

From Equation (6.41), it must be noted that in case $J = 0$, the provided control input exactly matches the commanded control action. However, for redundant control system, this exact solution might not be unique. Therefore, Bodson and Frost (2011) propose to extend the objective function with a second objective that minimises the required control effort, as given in Equation (6.43). Note that in this equation, ε represents a scaling parameter that provides the relative importance of the control effort objective with respect to the control error objective. Therefore, ε is set to a low value, $\mathcal{O}(10^{-3})$, in case the error minimisation has clear dominance over the effort minimisation (Bodson and Frost, 2011). Note that by extending the objective function, a unique solution can be ensured even when a redundant control system is used, given that the solution with minimum control effort is selected from the list of equivalent minimum error solutions.

$$\underset{\mathbf{u}}{\text{minimise}} \quad J_f = ||\mathbf{V}\mathbf{u} - \mathbf{u}_{cmd}|| + \varepsilon||\mathbf{u}|| \tag{6.43}$$

For the considered reference mission, however, this general control allocation algorithm is adjusted to take pulse-width modulation of the non-throttled thrusters into account. Therefore, rather than directly finding the control input vector $\mathbf{u}$, an alternative vector is calculated that contains the activation times of the different reaction control thrusters. This activation vector is represented by $\mathbf{t}_{act}$ and contains an equal amount of entries as there are reaction control thrusters. Furthermore, the activation vector indicates the pulse duration during which the thrusters are activated at maximum thrust level to achieve a control output as close as possible to the commanded control vector. Subsequently, this activation vector is mapped to a control input vector and used in the objective function of the optimisation problem, as previously described for Equation (6.43). Therefore, the optimisation problem provided in Problem 1 is solved for each thrust command obtained from the guidance system.

Problem 1:

$$\underset{\mathbf{t}_{act}}{\text{minimise}} \quad J_f = ||\mathbf{V}\mathbf{u} - \mathbf{u}_{cmd}|| + \varepsilon||\mathbf{u}|| \quad \text{subject to} \tag{6.44}$$

$$0 \leqslant t_{act_i} \leqslant h \quad \forall i \in \{1, ..., N\} \tag{6.45}$$

$$\mathbf{u} = \text{diag}\left(\mathbf{t}_{act}\right) T_{nom} \mathbf{1}_N^T \tag{6.46}$$

In Problem 1, Equation (6.44) represents the previously discussed objective function. Furthermore, Equation (6.45) represents a constraint on the activation vector. This equation states that for each thruster of the N thrusters in the reaction control system, the activation time lies between 0 and the time step of the guidance function, given by h. This is because at the next time step, a new command will be generated by the guidance function and subsequently sent to the control allocation algorithm. Therefore, no entry of the activation vector can exceed the magnitude of this time step. Lastly, Equation (6.46) represents how the activation vector gets mapped into the control input vector. In this equation, a diagonal matrix is generated from the activation vector, given by $\text{diag}\left(\mathbf{t}_{act}\right)$, and subsequently multiplied with the nominal thrust level of the reaction control thrusters, given by T_{nom}, and a N-dimensional column vector containing ones, given by $\mathbf{1}_N^T$.

For the considered reference mission, the reaction control thruster system consists of 24 thrusters, which are positioned in groups of three at each vertex of the chaser body, as was shown previously in Figure 6.1. The thruster matrix or configuration matrix, $\mathbf{V}$, which maps the thrust forces of all individual reaction control thrusters to the forces and moments acting on the chaser spacecraft, is given by Equation (6.47). In this equation, $\mathbf{F}$ and $\mathbf{M}$ represent the generated thrust force and moment on the chaser spacecraft, respectively, and $\mathbf{u}$ the control input vector, or equivalently the thrust vector containing the thrust forces of each reaction control thruster. Combining the geometry of the spacecraft, as given by **REQ_S_C_5** and Figure 3.3, with the configuration of the reaction control thrusters given in Figure 6.1, the configuration matrix as given by Equation (6.48) can be obtained. Note that for clarity purposes, Equation (6.48) actually provides the transpose of the thrust matrix $\mathbf{V}$.

$$\begin{pmatrix} \mathbf{F} \\ \mathbf{M} \end{pmatrix} = \mathbf{V}\mathbf{u} = \mathbf{V}\text{diag}\left(\mathbf{t}_{\text{act}}\right) \mathrm{T}_{\text{nom}} \mathbf{1}_{24}^{\mathrm{T}} \tag{6.47}$$

$$\mathbf{V}^{\mathrm{T}} = \begin{bmatrix} -1 & 0 & 0 & 0 & -0.35 & 1.25 \\ 0 & -1 & 0 & 0.35 & 0 & -0.55 \\ 0 & 0 & -1 & -1.25 & 0.55 & 0 \\ -1 & 0 & 0 & 0 & -0.35 & -1.25 \\ 0 & 1 & 0 & -0.35 & 0 & 0.55 \\ 0 & 0 & -1 & 1.25 & 0.55 & 0 \\ -1 & 0 & 0 & 0 & 0.35 & -1.25 \\ 0 & 1 & 0 & 0.35 & 0 & 0.55 \\ 0 & 0 & 1 & -1.25 & -0.55 & 0 \\ -1 & 0 & 0 & 0 & 0.35 & 1.25 \\ 0 & -1 & 0 & -0.35 & 0 & -0.55 \\ 0 & 0 & 1 & 1.25 & -0.55 & 0 \\ 1 & 0 & 0 & 0 & 0.35 & -1.25 \\ 0 & -1 & 0 & 0.35 & 0 & 0.55 \\ 0 & 0 & -1 & -1.25 & -0.55 & 0 \\ 1 & 0 & 0 & 0 & 0.35 & 1.25 \\ 0 & 1 & 0 & -0.35 & 0 & -0.55 \\ 0 & 0 & -1 & 1.25 & -0.55 & 0 \\ 1 & 0 & 0 & 0 & -0.35 & 1.25 \\ 0 & 1 & 0 & 0.35 & 0 & -0.55 \\ 0 & 0 & 1 & -1.25 & 0.55 & 0 \\ 1 & 0 & 0 & 0 & -0.35 & -1.25 \\ 0 & -1 & 0 & -0.35 & 0 & 0.55 \\ 0 & 0 & 1 & 1.25 & 0.55 & 0 \end{bmatrix} \tag{6.48}$$

Furthermore, the commanded control action calculated by the guidance system will consist solely of a thrust force command to influence the translational motion of the chaser spacecraft. The rotational control commands are calculated by the attitude controller and handled by the reaction wheel system. Therefore, the thrust moment component will always be zero, as can be seen in Equation (6.49), where $\mathbf{F}_{\text{cmd}}$ represents the force command calculated by the guidance system and $\mathbf{M}_{\text{cmd}}$ the moment command, which equals the zero vector as no rotation should be induced. Furthermore, to fully characterise the control allocation algorithm, the nominal thrust force of the reaction control thrusters is set to $\mathrm{T}_{\text{nom}} = 55$ N, as given in **REQ_S_C_13.2**. Additionally, the scaling parameter ε in the objective function is set to 10^{-3}, as provided by Bodson and Frost (2011).

$$\mathbf{u}_{\text{cmd}} = \begin{pmatrix} \mathbf{F}_{\text{cmd}} \\ \mathbf{M}_{\text{cmd}} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_{\text{cmd}} \\ \mathbf{0} \end{pmatrix} \tag{6.49}$$

Lastly, it should be noted that the control allocation algorithm will provide an activation time vector with entries that can be anywhere within the interval from 0 to h, including both interval limits. This is, however, not realistic given that the reaction control system can only provide a thrusting time that is a multiple of its operational frequency. Therefore, to ensure that sufficient accuracy remains, the reaction control thruster system is designed to operate at a frequency of 100 Hz, whereas the pulse-width modulation algorithm operates 1 Hz, identical to the frequency of the guidance system. Consequently,

Table 6.3: Reaction control thruster model settings

Thruster Error	Value
b_{RCT} [N]	0
T_{nom} [N]	55
$\sigma_{v_{RCT}}$ [N]	0
seed [-]	111
operational frequency [Hz]	100

the activation times get rounded to a hundredth of a second and a 1% thrusting accuracy can be obtained.

6.4.3. Actuator Models

The actuator models used for the ClearSpace-1 spacecraft are all part of the GRADS library and have previously been elaborated in Section 6.2. The interested reader is, however, referred to (Mooij, 2021a) for more details. The purpose of the current section is to provide details on the settings implemented for the respective actuator models.

Reaction Control Thrusters

A summary of the settings defining the actuator output provided by the reaction control thrusters is provided in Table 6.3. Note that in this table, the $\sigma_{v_{RCT}}$ parameter indicates the 1-σ standard deviation used to sample the zero-mean Gaussian white-noise vector present in the actuator output. Furthermore, it should be noted that, for simplicity purposes, the bias and noise in the reaction control thruster output are assumed to equal zero. This is to ensure that the thruster exactly outputs its provided command. However, to maintain the connection with reality, this provided command is a modulated command. This ideal reaction control thruster system is implemented, given that preliminary analyses on the actuator performance indicated that the guidance system is very sensitive to the accuracy of the thruster output. This has already been discussed for the Monte Carlo analysis performed on the guidance system in Chapter 4, where the slightest error in thrust magnitude or direction could render the trajectory infeasible. This is because the thruster is not capable of performing the commands of the guidance system, rendering a subsequent recalculation of the reference trajectory ineffective. Therefore, the combined system is no longer capable of converging to the required trajectory and diverges away from the terminal conditions. A possible solution could be provided in terms of a in-orbit calibration of the reaction control thrusters, during which predefined manoeuvres are executed to estimate some error characteristics of the thruster system. Alternatively, an adaptive controller could be used to handle the actuator errors of the thruster system. However, due to the complexity of these proposed solutions and the limited time available for the thesis project, unfortunately both have not been investigated further and are left for future work.

Aside from that, it should be noted that the nominal thrust level of each reaction control thruster is set at a representative value of 55 N, as specified by **REQ_S_C_13.2**. Additionally, an operational frequency of 100 Hz is instituted for the system, as to correspond with the operational frequency required by the previously discussed pulse-width modulation and control allocation algorithm.

Reaction Wheels

A summary of the settings defining the actuator performance is provided in Table 6.4, from which it should be noted that the reaction wheels are assumed to be ideally aligned with the spacecraft, *i.e.*, the misalignment matrix is defined as an identity matrix. Nevertheless, errors are present in the generated control output caused by viscous and Coulomb friction. Representative values for these respective friction coefficient are taken from (Carrara and Kuga, 2013). Furthermore, to make the reaction wheel system sufficiently powerful to synchronise attitude with the tumbling debris item, each reaction wheel is assumed to have an inertia of 0.5 kgm^2, a motor constant of 0.5 Nm/A, a maximum rotational velocity of 5000 rpm and an electrical current limit of 100 A. Moreover, each reaction wheel can provide a maximum deliverable torque of 50 Nm, as specified by **REQ_S_C_13.4**. Lastly, it should be noted that the operational frequency of the reaction wheel system must be in accordance with the 100 Hz frequency of the INDI controller, such that an effective linearisation of the attitude dynamics can be achieved.

Table 6.4: Reaction wheel system model settings as provided by Carrara and Kuga (2013)

Reaction Wheel Error	Value	Performance Parameter	Value
Misalignment matrix [-]	$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$	wheel inertia [kgm^2]	0.5
		k_m [Nm/A]	0.5
		maximum rotational velocity [rpm]	5000
k_v [Nms]	$4.84 \cdot 10^{-6}$	maximum electrical current [A]	100
k_c [Nm]	$8.975 \cdot 10^{-4}$	maximum torque [Nm]	50

Table 6.5: Second-order derivative filter settings as provided by E. Smeurs (2018)

Filter setting	Value
ω_n [Hz]	50
ζ [-]	0.55

6.4.4. Sensor Models

To apply the INDI technique in the attitude controller of the chaser spacecraft, it is required that accurate measurements of the actuator output and angular acceleration are available at a sufficiently high sampling rate. Therefore, the current section details how these measurements are established.

Angular Acceleration Measurement

Dedicated sensors capable of directly measuring angular accelerations do exist, but are unlikely to be installed onboard the ClearSpace-1 satellite. Therefore, accurate estimates of the angular accelerations will have to be deduced from noisy measurements of other sensors. Generally, this is achieved by differencing angular velocity measurements taken by the IMU sensor. However, taking the derivative of noisy measurements will always decrease the signal to noise ratio and therefore increases errors. The noisy measurements can, however, first be passed through a low pass filter, effectively smoothing part of the noise from the signal, before taking its derivative. The second-order transfer function provided in Equation (6.50) can perform exactly this, as proposed by E. Smeurs (2018).

$$H(s) = \frac{\omega_n^2 s}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{6.50}$$

In this equation, ω_n represents the cut-off frequency of the low-pass filter and ζ its damping ratio. Values for these two parameters are as defined in Table 6.5, where their respective magnitudes are taken from E. Smeurs (2018). Note that, to improve the accuracy of the angular acceleration measurements, the inertial rate estimate obtained from the navigation system is used, instead of noisy IMU measurements. Moreover, as the navigation system operates at an equal frequency to the control system, *i.e.*, both at 100 Hz, the simulation environment can consider the input signal to the transfer function as continuous in time. Consequently, Equation (6.50) can be used. However, in case a future design iteration requires to transform this transfer function to discrete time, *e.g.*, when the control system is handling real signals onboard a spacecraft, Tustin's bilinear method can be applied, where the Laplace domain variable s is replaced by $\frac{2}{T_s}\frac{z-1}{z+1}$, with z the discrete time equivalent of s and T_s the sampling time. This is nevertheless left for future work.

Actuator Output Measurement

To estimate the control moment generated by a reaction wheel, a simple relation between the current drawn by the wheel and the generated control torque can be employed, as previously given by Equation (6.9). To obtain this control torque, an electrical current sensor is needed that can measure the current drawn by the reaction wheel system. For this sensor, a simple random noise value is added to the actual current. The sensor model is therefore similar to the mathematical model for the vision-based navigation system, given in Equation (5.34). The only difference is given by the fact that the measured output is a measured electrical current, $I_{elec,meas}$, which is calculated from the sum of the true electric current, I_{elec}, and a random white noise vector, v_{elec}. A description of this simple current sensor model, which is applied to each wheel of the reaction wheel system, is provided in Equation (6.51).

Table 6.6: Electrical current sensor settings as provided by Cheemi (2022)

Current Sensor Error	Value
$\sigma_{v_{elec}}$ [%]	0.5
seed [-]	5432
sample rate [Hz]	100

$$I_{elec,meas} = I_{elec} + v_{elec} \tag{6.51}$$

To specify the output established by this sensor model, the performance parameters given in Table 6.6 are used. The value for the 1-σ standard deviation of the zero-mean Gaussian white-noise, given by $\sigma_{v_{elec}}$, is based on the product sheet provided by Cheemi (2022), which states that high accuracy current sensors can achieve an accuracy of 0.5% of the input current. Lastly, note that an operational frequency of 100 Hz is applied to all parts of the control system, including this sensor.

However, to ensure that this actuator output measurement is synchronous with the angular acceleration measurement, the signal needs to be filtered using an equivalent second-order low-pass filter, as given by E. Smeurs (2018). A mathematical description of this filter is provided in Equation (6.52)

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{6.52}$$

Note that this second-order filter is similar to Equation 6.50, with the difference that the s parameters is removed from the numerator, to ensure that a purely filtered signal is obtained. This filtering of actuator outputs is on one hand needed to remove noise from the signal, whereas on the other hand, it is applied to ensure a delay is obtained in the actuator output measurements that is equal to the delay of the measured acceleration signal. Therefore, an equivalent second-order filter is employed, for which the performance parameters are taken from Table 6.5. Note again that a continuous time filter is currently implemented, as the actuator sensor operates at an equal frequency to the control system. However, in case it is required to transform the transfer function to discrete time, Tustin's bilinear method can once more be applied. Nevertheless, this is left for future work.

6.5. Verification

The verification procedure of the control system is provided in two distinct parts. First, the attitude control function is verified by analysing the PID rate controller, the full state INDI controller and the developed sensor models. Second, the orbit control system is verified by analysing the pulse-width modulation algorithm in combination with the reaction control thruster system. The employed actuator models, however, are already verified by Mooij (2021b), as they are part of the GRADS library. Therefore, formal verification processes of the reaction wheel system and the reaction control thruster system are omitted here. Nevertheless, the integration of these actuator models with the attitude control system and the orbit control system is performed during a system test of the respective controllers.

To start, a verification of the isolated PID rate controller and the isolated full state INDI controller is performed by analysing their respective time response to a sigmoid command. A sigmoid is used as input rather than a step function, given that it more realistically resembles the true input send to the respective control systems.

A graphical representation containing the time response of the PID rate controller is provided in Figure 6.3. The left side of this figure provides the input command sent to the three angular velocity components and represents a temporary rotation of the spacecraft over all three body axes. During this rotation, the angular velocity is temporarily increased from 0 deg/s to 3.5 deg/s over all three axes, as would occur during attitude synchronisation with the tumbling debris item. Furthermore, the simulation parameters of this verification test are taken from the definitions provided in Table 3.1 and the implemented PID controller is characterised by Table 6.1. The right side of Figure 6.3 provides the corresponding angular velocity error history for this temporary manoeuvre, along with the error threshold that defines the switch from PID rate control to full state INDI control in the attitude control system. Therefore, the PID controller only needs to reduce the angular velocity error below this threshold, rather than the entire error. From Figure 6.3 it can be seen that the controller quickly tracks the reference

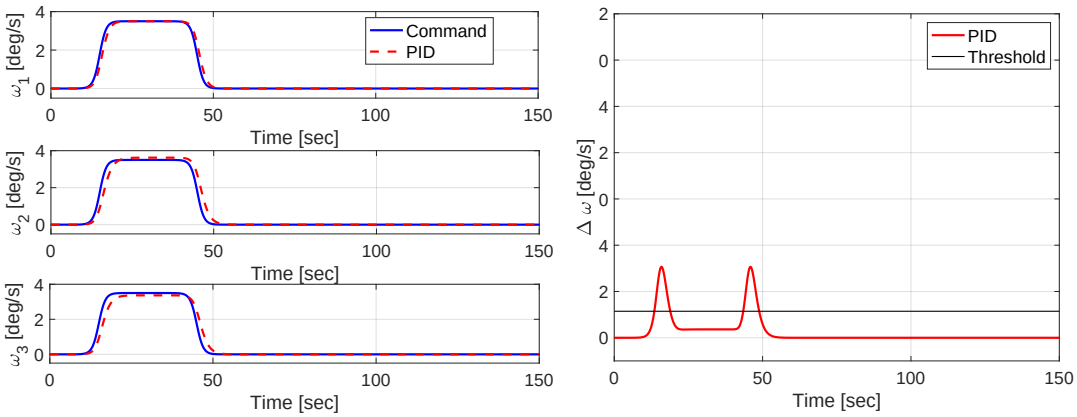


Figure 6.3: Sigmoid response of PID rate controller for an temporary angular velocity change (left) and corresponding angular velocity error with predefined error threshold (right)

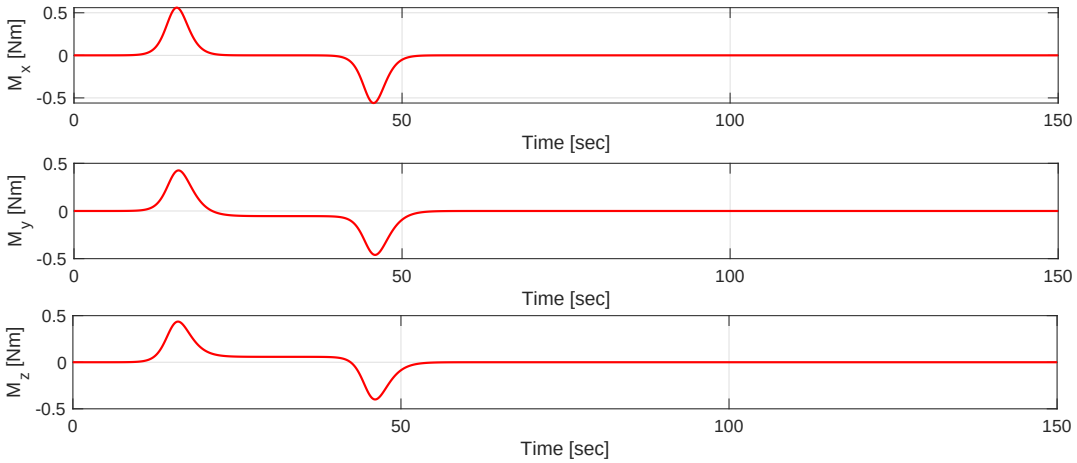


Figure 6.4: Control input history for temporary angular velocity change established by PID rate controller

command within the predefined error threshold, only taking approximately 3.5 seconds. Furthermore, a steady-state error persists, but is insignificant given that it falls below the predefined threshold.

When subsequently inspecting the control input required to track this sigmoid command, as provided in Figure 6.4, it can be observed that a small control effort is calculated. Furthermore, a smooth control input is provided, which is beneficial for the durability of the control actuators. Hence, it can be concluded that the PID rate controller is capable of efficiently changing the angular velocity of the chaser spacecraft to a predefined accuracy requirement. Therefore, it can be assumed that the controller is implemented correctly and that the system characteristics provided in Table 6.1 result in a good balance between controllability and control effort, even though these values are not formally optimised. Note however that, to improve the performance of the controller, it is recommended to perform formal gain tuning per reference axis. This is because it can be observed from the system responses in Figure 6.3, that the controller behaviour varies over the different axes and is caused by a different moments of inertia of the body. Nevertheless, due to time constraints, gain tuning per axis is currently omitted and left for future work.

Furthermore, the time response of the full state INDI controller is provided in Figure 6.5, where the left side of the figure provides the reference command for the quaternion vector, the middle represents the reference command for the angular velocity vector and the right side specifies the components of the control input vector. Note that, to avoid having to employ the PID rate controller for angular velocity synchronisation, the reference command is initialised as presynchronised, *i.e.*, both the reference command and the spacecraft are started from zero angular velocity. This is done to achieve a verification test of the isolated INDI controller performance, implying that the provided time response represents a temporary reorientation of the spacecraft to a fixed attitude, as defined by the quaternion command.

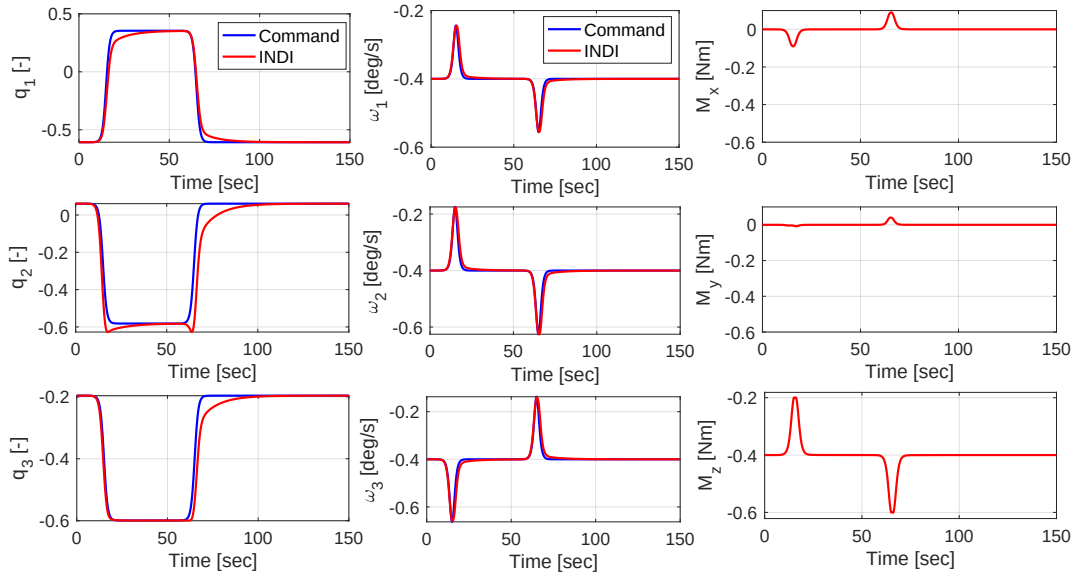


Figure 6.5: Quaternion response of full state INDI controller to a sigmoid attitude change (left), angular velocity response corresponding to the sigmoid attitude change (middle) and resulting control input (right)

Accordingly, the provided angular velocity command follows from this attitude reorientation manoeuvre. Furthermore, to define the verification test, the simulation parameters are taken from the definitions provided in Table 3.1 and the INDI controller is characterised by the values in Table 6.2. Moreover, the actuator output measurements and angular acceleration measurement provided to the INDI controller are taken as ideal for the current verification test, to achieve an isolated assessment of the INDI's performance.

From Figure 6.5 it can be seen that the controller is capable of providing accurate and rapid tracking of the reference command, with a settling time of approximately 35 seconds. When subsequently inspecting the corresponding control input of this reorientation manoeuvre, it should be noted that a limited control effort is required and that this control effort is distributed over a smooth control profile. Hence, it can be concluded that the full state INDI attitude controller can effectively change the attitude of the chaser spacecraft, while simultaneously requiring low control effort. Therefore, it can be assumed that resulting INDI controller is implemented correctly and that the system characteristics provided in Table 6.2 provide sufficient controllability, despite not being formally optimised. Nevertheless, differences in system responses are observed for the different reference axes in Figure 6.5. These differences are caused by different moment of inertia values and can be resolved by performing formal gain tuning per axis. However, due to time constraints, this could not be performed and is therefore left for future work.

To further verify the functionality of the INDI controller, its implementation with the angular acceleration sensor and actuator output sensor needs to be verified. However, first the individual sensor performance of the electrical current sensor is verified. To test this sensor, a sample input signal is provided and the corresponding output is analysed. The percentage-dependency of the error value on the input magnitude is assessed in the exact same manner as performed in Figure 5.7 for the VBN system. For both a sequenced step input and a ramp input, it can be seen that the error is proportional to the provided input value, as well as bounded by the 3-σ curves. Therefore, it can be concluded that the overall functionality of the current sensor is correct. Furthermore, for brevity purposes, also the verification tests of the low-pass filters are currently omitted. Nevertheless, the cut-off frequencies of these filters are verified by inspection of their respective bode diagrams.

Subsequently, an integration test of the full state INDI algorithm with the previously discussed sensor models is performed. However, these results are not provided. Instead, the results of a full system test of the attitude control system are presented. This test does not only analyse the integration of the full state INDI attitude controller with the developed sensor models, but also the further integration of the PID rate controller and the implemented actuator models. Therefore, a successfull system test renders the intermediate integration test superfluous.

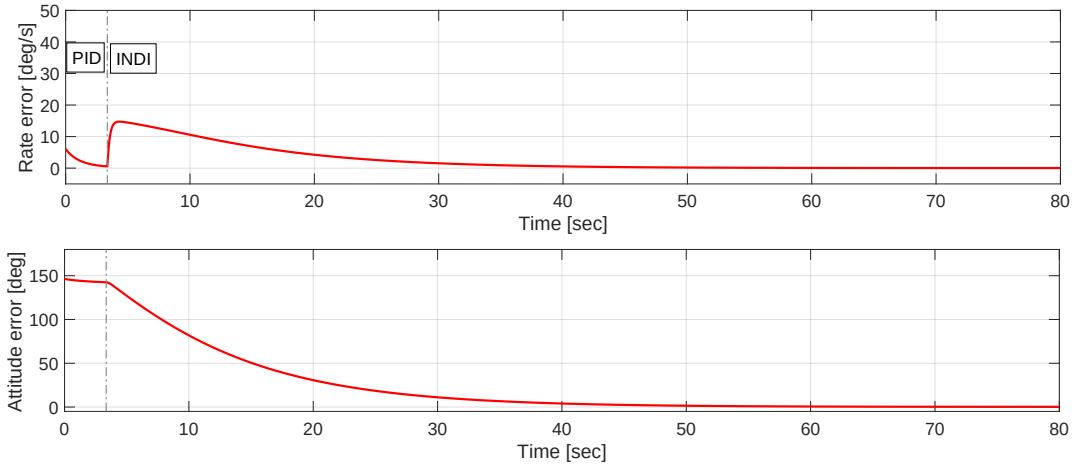


Figure 6.6: Angular rate and attitude error during synchronisation with tumbling debris item performed by integrated attitude control system

To perform the system test, a simulation is run using the simulation parameters defined in Table 3.1, with the sole difference that the angular velocity of the target spacecraft is altered to 3.5 deg/s over all three body axes, as it replicates a realistic synchronisation manoeuvre. Furthermore, the control algorithm and control sensor parameters are as defined for the previous verification tests. Lastly, for the control actuators, the reaction wheel system characteristics provided in Table 6.4 are employed. With the simulation defined, the attitude control system is applied to perform attitude synchronisation with the tumbling debris item, identical to the true mission scenario.

Results for this attitude controller test are provided in Figures 6.6 and 6.7, where the former presents the angular velocity error and attitude error during synchronisation, whereas the latter provides the corresponding control inputs. From these figures it can be identified that first the PID rate controller is applied to reduce the angular velocity error to below the threshold, which can be achieved within approximately 3.5 seconds, after which the full state INDI controller is used to achieve attitude synchronisation, which can be achieved within approximately 50 seconds. Furthermore, from the control input history it is visible that the rate controller requires significantly less control effort when compared to the INDI controller. This is because the rate controller has to perform a control action identical to the sigmoid response in Figure 6.3, whereas the INDI controller now has to provide simultaneous rate and attitude control, in contrast to the sigmoid response in Figure 6.5. Therefore, larger control inputs are expected for the full state INDI controller when compared to Figure 6.5. Nevertheless, it can be observed that the control effort remains bounded over the entire synchronisation manoeuvre.

Note however that, due to time constraints, the current control system implements a discrete transition from PID control to full state INDI control, which causes the strong spikes in control input histories. In a future design iteration, a signal transition should be performed where the step change between the PID controller and INDI controller is smoothed. This can for example be provided by a sigmoid function and will ensure that the undesirable behaviour can be removed from the control input histories. Therefore, it can be concluded that the entire attitude control system correctly integrates the interaction between the control algorithms, the control actuators and control sensors. Hence the system is ready to be part of the overarching GNC system.

To verify the functionality of the orbit control system, a test on the output generated by the pulse-width modulation and control allocation algorithm is performed. Note that the corresponding control actuators, in this case reaction control thrusters, are not verified as this procedure has already been performed by Mooij (2021b). Nevertheless, their integration with the pulse-width modulation algorithm is assessed and provided in Figure 6.8. To obtain this figure, the reaction control thruster characteristics as provided in Table 6.3 are used. Note that, in the current test, the correct conversion of a guidance command into a corresponding modulated thrust output is analysed. For this, a simulation of the orbit control system is performed, where the command provided on top of Figure 6.8 is required as thrust force along the positive X-direction of the chaser's body frame. Furthermore, to make this test as representative as possible, the thrust commands are given by discrete thrust levels sampled at a rate

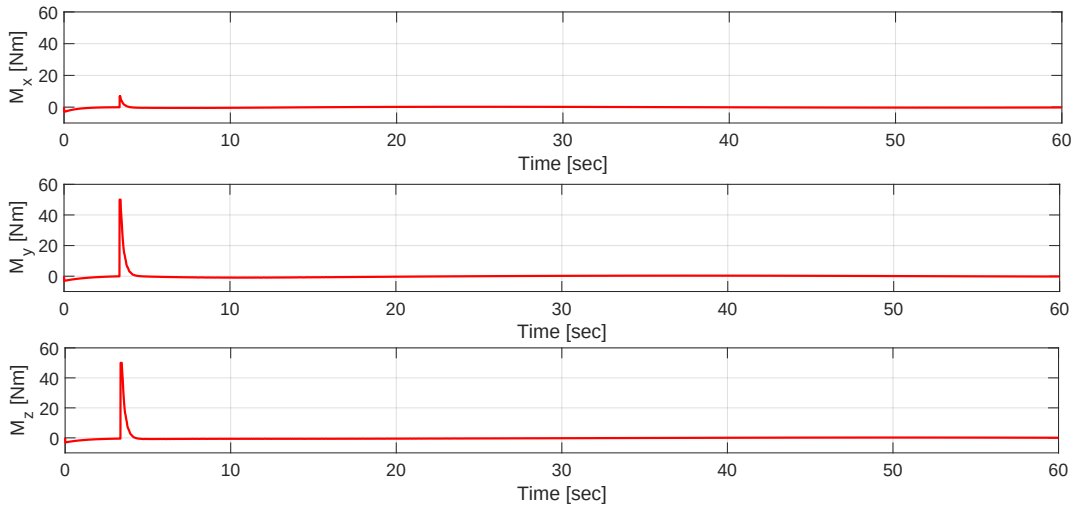


Figure 6.7: Control input history during synchronisation with tumbling debris item performed by integrated attitude control system

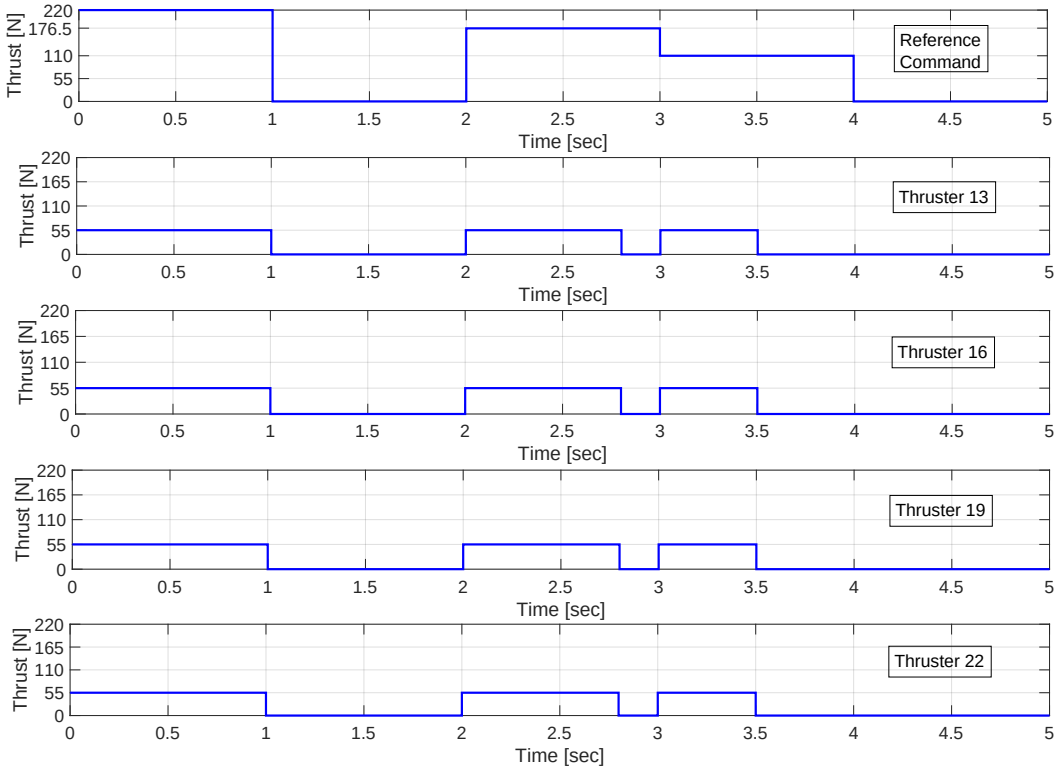


Figure 6.8: Thrust command along positive X-axis of chaser's Body-Centered-Body-Fixed frame (top) and corresponding modulated reaction control thruster output for thrusters 13, 16, 19 and 22 (bottom)

of 1 Hz, similar to the output of the guidance system. Moreover, no torque commands around any body axis are provided and, for clarity purposes, also no other thrust force directions are commanded. This is done such that, when considering the configuration of the reaction control thruster system in Figure 6.1, it can be seen that the active thrusters are represented by numbers 13, 16, 19 and 22 and only their respective output needs to be inspected. Therefore, the modulated thrust outputs of these four thrusters are provided in the remaining graphs of Figure 6.8.

When inspecting the commanded thrust signal, it can be seen that during the first second, a thrust level of 220 N is required. This results in all four thrusters to be activated for the full duration of the second, as each thruster operates at a nominal level of 55 N. For the subsequent second, no thrust is commanded and all modulated thrusters are switched off. During the third second, 176.5 N is commanded, and its modulation is achieved by activating all four thrusters during 0.80 seconds and subsequently switching them off for the remaining 0.20 seconds. This provides a thrust value of only 176 N, rather than the commanded 176.5 N. Nevertheless, this modulated thrust level falls within the accuracy of what can be achieved with the true reaction control thrusters, as the calculated activation times are always rounded down. Therefore, it can be concluded to be correct. Subsequently, during the fourth second, 110 N is requested, and all thrusters are activated for half a second, until finally the last second requires 0 N, during which all reaction control thrusters remain switched off. Note that during the entire duration of the thrust command, the remaining 18 reaction control thrusters maintain an inactive status, as expected. Furthermore, note that identical results are obtained when providing thrust commands along the other directions of the chaser's body frame, as well as when commanding combined thrust commands. However, for brevity purposes, these results are omitted here, as they do not add much value to the discussion. Hence, from the results provided in Figure 6.8, it can be concluded that the pulse-width modulation and control allocation algorithm is verified and correctly implemented with the reaction control thruster system. Therefore, the orbit control system is ready to be implemented in the GNC system.

7 Software Development and Testing

The current chapter is concerned with the development of the functional GNC simulator in MATLAB, as given in ESA's procurement document (European Space Research and Technology Centre 2020). A software architecture of this simulator is provided in Section 7.1 and contains the software components used to model the performance of the guidance, navigation and control systems developed in the current thesis study. However, to ensure that feasible results can be obtained, a verification of the simulator's functionality is required. Given that the separate subsystems of the GNC system have previously been verified in Chapters 4 to 6, the only thing that remains is a system test of the integrated system. This is provided in Section 7.2, where first an integration test of the guidance and control system is performed, after which a system test of the overall GNC system is presented.

7.1. Simulator Architecture

An overview of the integrated software architecture, incorporating the functionalities discussed in the previous chapters, is provided in Figure 7.1. From this figure, it can be seen that the simulator can basically be segregated into four main components: the motion simulator represented by the orange block, the navigation system represented by the yellow blocks, the guidance system represented by the blue block and the control system represented by the green blocks.

At the heart of the software diagram lies the motion simulator, which contains the translational and rotational dynamics of the VESPA target and ClearSpace-1 chaser. Moreover, the motion simulator also incorporates the spatial environment to which both spacecraft are subject. Starting from an initial state, the translational and rotational dynamics are propagated under the influence of environmental accelerations and torques. Moreover, the motion of the chaser spacecraft is further subject to control forces and moments, as generated by the control actuators. After propagation, the states of the chaser and target are fed back to the environment models, which are represented by a extended gravity model, an atmosphere model, a solar radiation pressure model and a magnetic field model to calculate their magnitude and direction. The calculation of these environmental accelerations and torques is provided by conical body interfaces of the target and the parallelepiped body interfaces of the chaser.

When subsequently considering the navigation system, the propagated state of the chaser spacecraft is fed to the sensor suite of the inertial navigation system, whereas the propagated states of both spacecraft are needed to provide measurements of the pseudo-sensors in the relative navigation system. These pseudo-sensors basically operate by calculating the true relative inertial state of the chaser with respect to its target. With these measurements, the Unscented Kalman Filter of the inertial navigation system estimates the inertial state of the chaser, and the artificial vision-based navigation system provides a pseudo-estimate of the relative inertial state. As discussed in Chapter 5, these estimates are subsequently combined to obtain a inertial estimate for the target spacecraft, which in combination with the inertial state of the chaser, allow to calculate the Hill vector. This vector is needed, as it provides the initial condition for the guidance system.

The guidance system is split into two subsystems, a convex guidance system calculating a reference trajectory for the chaser, along with a thrust command history needed to fly this reference trajectory, and a tracking guidance system that calculates intermediate corrective burns to eliminate trajectory dispersions. The thrust commands of both subsystems are combined and, as these are provided in the Hill frame of the target spacecraft, converted to the inertial frame for subsequent handling.

The orbit control system of the chaser spacecraft receives this converted thrust command, runs it through a pulse-width modulation and control allocation algorithm and passes it on to the reaction control thruster system. Moreover, the control function of the chaser spacecraft also implements an atti-

tude control function. To do so, a PID rate controller is combined with a full state INDI attitude controller to establish attitude synchronisation with the tumbling target. The rate controller uses the estimated angular velocity of the chaser and target spacecraft as input, whereas the INDI controller obtains the quaternion profiles of both spacecraft along with their angular velocity profiles. Note that, to make the INDI controller independent of an accurate system model, actuator outputs of the reaction wheel system are provided from electrical current measurements. These measurements are first multiplied by the motor constant of the electrical motors in the reaction wheels to obtain a torque measurement, and subsequently processed by a second-order low-pass filter, before they are provided as input to the INDI controller. Furthermore, the inertial angular velocity estimate of the chaser spacecraft is processed by a second-order derivative filter to obtain an angular acceleration estimate. With the inputs defined, a torque command output can be provided by the attitude control system, which is subsequently fed to the reaction wheel system. Lastly, to close the loop of the functional GNC simulator, the forces and moments generated by the control actuators are provided to the plant dynamics of the chaser and used to manipulate its translational and rotational motion.

Note that for the verification tests performed in Chapters 4 to 6, isolated blocks of Figure 7.1 are used. To be able to do so, real states of the motion simulator are connected to the guidance and control systems, replacing any state estimates provided by the navigation system. Furthermore, the orbit control commands calculated by the guidance system are directly applied to the motion simulator, rather than provided by the reaction control thruster system, and the trajectories of the control and navigation system tests consist of freely drifting motion, without the inclusion of a guidance system.

7.2. Integration Testing

To verify the functionality of the developed GNC simulator, unit tests have previously been performed on the components that constitute the GNC system in Chapters 4 to 6. Therefore, what is left for verification is an integration test of the various subsystems and a subsequent system test of the overall simulator. During the integration tests, first the performance of the integrated guidance and control system is verified, after which the functionality of the navigation system integration with the combined guidance and control system is assessed. Note, however, that the latter integration test is thus equivalent to a system test of the overall GNC system. Hence the current section will be limited to providing the outcome of these two tests, and to avoid repetition, the results of both are provided in one go. To obtain the results, a simulation of the mission scenario described in Chapter 4 is performed using the general parameters given in Table 3.1, where only the angular velocity of the target is altered to $(3.5, 3.5, 3.5)^T$ deg/s. Furthermore, the respective settings for the guidance, navigation and control system are as defined in Chapters 4 to 6.

A comparison of the flown trajectories that result from these integration tests is provided in Figure 7.2, for which it should be noted that, due to the aspect ratio of the plot, the keep-out sphere constraints can look a bit morphed. Furthermore, it should be noted that, for all three trajectory calculations, identical environmental disturbances are implemented. Therefore, trajectory dispersions are inevitable and recalculations of the guidance solution will be provided. This implies that differences in the flown trajectories, although partly caused by errors in the control and/or navigation systems, cannot solely be quantified as trajectory errors. This is because, with updates of the reference trajectory provided by the convex guidance system, the path to the capture position is reconstructed by the onboard computer on the fly, using its current state as initial condition. Consequently, performing a direct trajectory comparison is difficult. Nevertheless, it is visible from Figure 7.2 that similar trajectories are flown when the chaser is equipped with an isolated guidance system, an integrated guidance and control system and an integrated GNC system.

Furthermore, it is required for all three trajectories to satisfy the path and input constraints defined in Chapter 4. To analyse the path constraints, Figure 4.11 has been remade with the results obtained for the isolated guidance system, integrated guidance and control system and integrated GNC system. To avoid repetition of the graph, however, the current discussion is limited to providing the differences that are observed with respect to Figure 4.11.

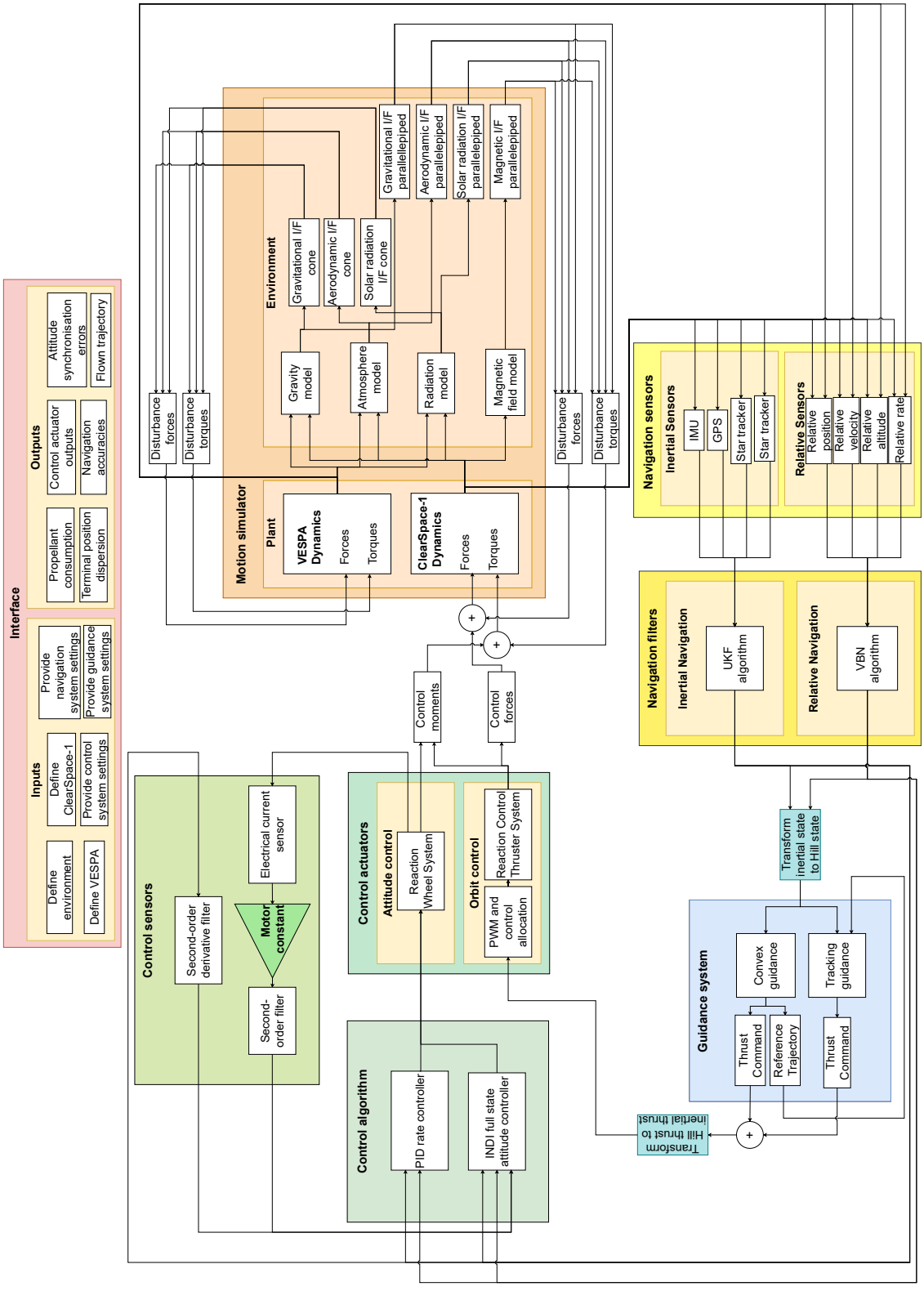


Figure 7.1: Software architecture of functional GNC simulator

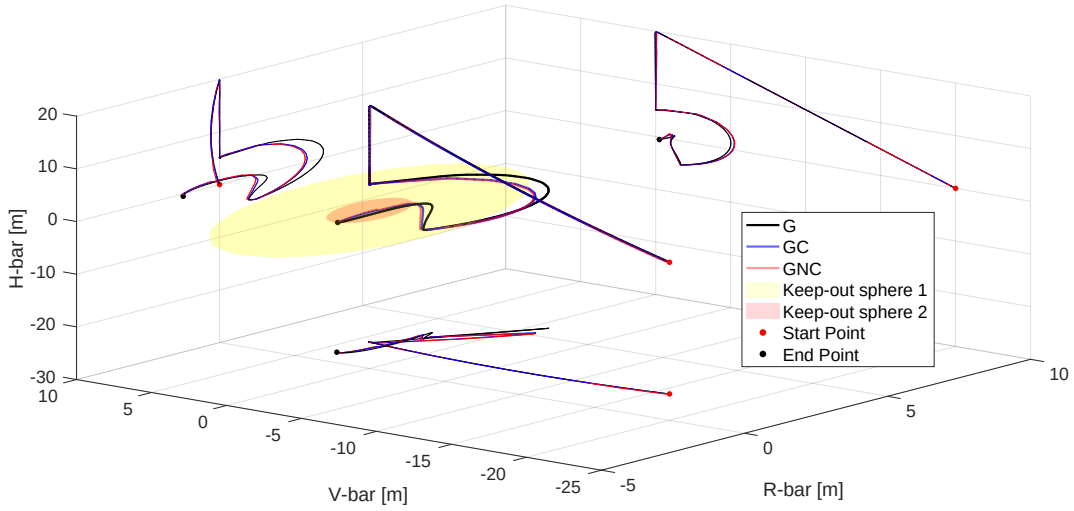


Figure 7.2: Trajectories flown by isolated guidance system (G), integrated guidance and control system (GC) and integrated GNC system (GNC), as expressed in the Hill frame of the target spacecraft

When inspecting the position constraints, it can be observed that all three trajectories easily satisfy the keep-in sphere constraint of the final approach phase in a manner equivalent to what is provided in Figure 4.11. However, in contrast to the results of this figure, the trajectories that result from the integrated guidance and control system and the integrated GNC system temporarily dip into the keep-out sphere constraint during the attitude synchronisation phase and the final approach phase. This is because the current implementation of the GNC system does not contain any collision avoidance manoeuvres, such that violations of the keep-out sphere constraint can persist due to control and/or navigation errors. Nevertheless, it is found that the tracking guidance system rapidly corrects these violations by leaving the keep-out sphere, after which a recalculation of the convex guidance system is performed. Furthermore, note that the chaser never dips so deep into the keep-out sphere that a collision actually fires, even though it would not result in disastrous consequences for the mission, as the chaser ensures low impact velocity throughout all three trajectories.

This low velocity magnitude is visible when comparing the velocity norms to the maximum velocity constraint, similar to what is performed in Figure 4.11. To only discrepancy with respect to the figure can be found in the fact that, for the integrated guidance and control system and the integrated GNC system, the velocity magnitudes do not reduce back to absolute zero at the end of each mission segment. However, as subsequent mission segments can also successfully be initiated with a non-zero starting velocity, these terminal velocity discrepancies are not classified as problematic state violations.

Lastly, the only remaining path constraint that needs to be assessed is the approach cone constraint during the final approach phase. A comparison of this constraint's performance during the three trajectories is provided in Figure 7.3, where the green curve represents the right hand side of Equation (4.43) and the other curves represent the left hand side of this equation for their respective trajectories. Therefore, when the distance with respect to the centreline of the approach cone is below the green curve, the approach cone constraint is successfully imposed. When inspecting Figure 7.3, it can be seen that all three trajectories remain within the approach cone. However, the trajectory obtained from the isolated guidance system touches the approach cone for a brief moment. This is probably because it results in the most fuel-efficient trajectory to the capture position, and is not visible for the integrated guidance and control system and the integrated GNC system trajectories, as control and/or navigation errors during the preceding phases cause trajectory dispersions that initiated the definition of a new trajectory.

Nevertheless, to achieve these seemingly analogous trajectories, three rather different thrust input histories are calculated, as can be seen from Figure 7.4. The figure indicates that all trajectories are flown while satisfy the imposed maximum thrust constraint, where the magnitude of each thrust component is limited to 220 N. Moreover, the figure also shows the difference in the respective thrust commands, where it can be seen that, with the addition of realism into the system in the form of non-ideal control and navigation systems, the discrepancy in thrust command increases. This follows from

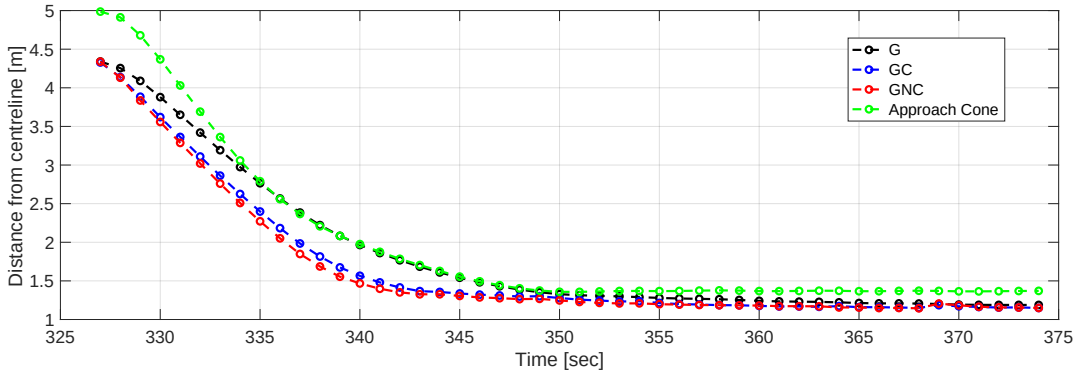


Figure 7.3: Approach cone constraint imposed during final approach phase, where the green curve represents the right side of Equation (4.43), whereas the remaining curves represent the left side of Equation (4.43) for their respective trajectories

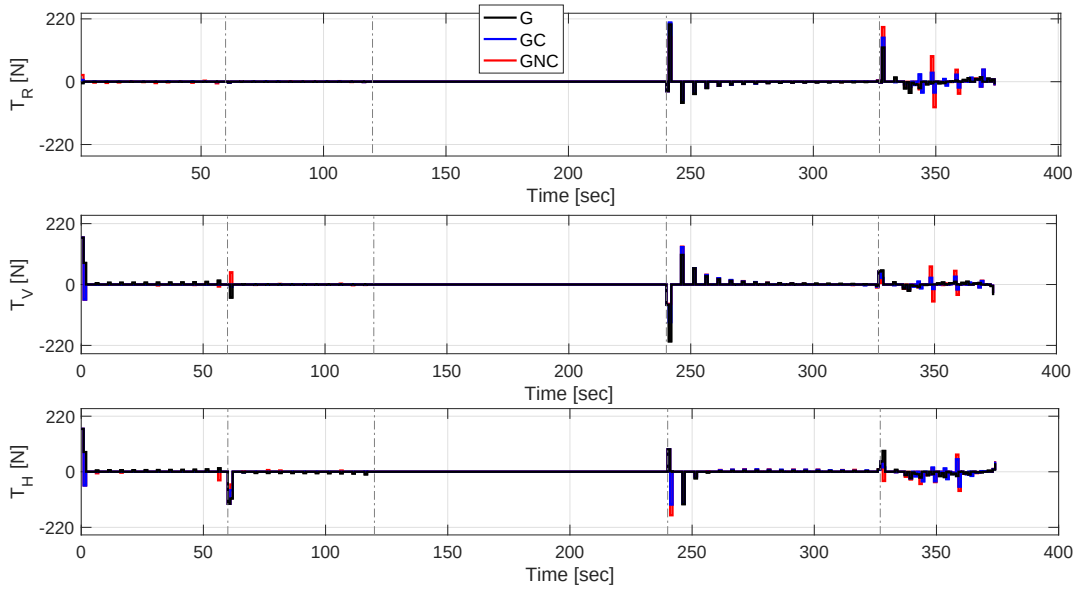


Figure 7.4: Thrust vector components calculated by the guidance system referenced with respect to the Hill frame of the target spacecraft

expectations when considering that these non-ideal systems cause trajectory dispersions, resulting in more intermediate corrective burns from the tracking guidance systems, as well as a change in reference trajectory provided by the convex guidance system. Therefore, the calculation of the thrust commands functions as expected.

For the isolated guidance system, these thrust commands are implemented as is in the motion simulator. However, for the integrated guidance and control system and the integrated GNC system, these commands are first allocated over the reaction control thrusters and subsequently modulated. To ensure that this procedure is implemented correctly, the thrust outputs provided by the first, second and third thruster of the reaction control system, as given in Figure 6.1, are provided in Figure 7.5. Note that Figure 6.1 indicates that these thrusters are directly along the negative X, Y and Z-axes of the chaser's body frame such that the components of Figure 7.5 result. Furthermore, from this figure it can be seen that the thrusters either provide no thrust, or short pulses of thrust at maximum capacity, indicating a correctly functioning modulation and allocation algorithm. Moreover, it can be observed from the trajectories in Figure 7.2 and the previously provided discussion that the modulation and allocation algorithm provides sufficient controllability over the translational state of the chaser spacecraft.

Lastly note that, for the remaining thrusters of the orbit control system, similar results to Figure 7.5 are obtained. However, for brevity purposes, these results are currently omitted.

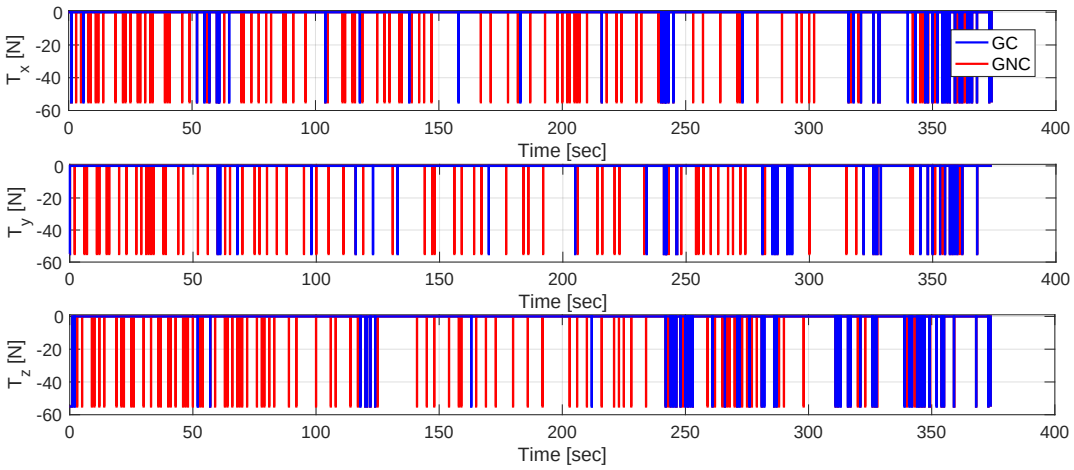


Figure 7.5: Reaction control thruster output of thruster 1 (top), thruster two (middle) and thruster three (bottom), referenced with respect to the Body-Centred-Body-Fixed frame of the chaser spacecraft

Additionally, the integrated guidance and control system and the integrated GNC system provide attitude synchronisation with the tumbling debris item. To compare their respective results, Figure 7.6 can be employed. In this figure, the angular velocity error and the attitude error during the entire mission duration are provided. With the start of the attitude synchronisation phase, the controllers are switched on and attitude synchronisation is initiated. The system response of both systems is near-identical, even though the integrated guidance and control system uses ideal navigation measurements as input, whereas the integrated GNC system employs the state estimates of the navigation system. This difference does, however, not seem to affect the system responses and is due to the imposed navigation accuracy requirements that result in a high performance navigation system. Therefore, both attitude control system are capable of providing attitude synchronisation.

Nevertheless, despite the system responses being near-identical, the calculated control inputs of both system are rather different, as can be seen from Figure 7.7. In this figure, the control moment components provided by the reaction wheel system are presented. It can be identified that a larger moment is required for the integrated GNC system and that the time history of this control moment is less smooth. This is caused by the errors presents in the navigation estimates and follows from expectations. Still, the calculated control input remains bounded over the entire manoeuvre and, after the initial spikes at the instant the control systems are turned on, the input history rapidly converges to a smooth tracking profile.

Until now, the guidance and control performances of the combined guidance and control system and the integrated GNC system have been discussed. However, for the latter system, it is also necessary to verify the performance of its navigation system, when combined with the remaining parts of the GNC system. To this extent, the navigation results of the previous test case can be employed. Note that, in contrast to the navigation system verification performed in Chapter 5, the current trajectory is not a free-drift trajectory on the negative V-bar of the target. Instead, a close-range rendezvous of the target is flown where, on top of the environmental disturbances, control forces and moments are imposed on the chaser spacecraft. Therefore, a reduction in navigation accuracy is to be expected.

The resulting inertial state estimation errors are provided in Figures 7.8 to 7.11, where the inertial navigation error is given per state component, along with the corresponding standard deviations of the error covariance matrices. When comparing the estimation errors of the inertial position estimate with the navigation system verification tests, as given by Figures 7.8 and 5.3, an increase in both the navigation errors and the estimation error covariance can be seen. This is caused by the discrete thrust pulses commanded by the guidance system, which, by rapidly changing the position of the chaser, result in a reduced measurement accuracy of the GPS receiver. Consequently, a poorer performance of the inertial navigation system results. Nevertheless, the respective navigation errors remain low, *i.e.*, at centimeter level. Furthermore, when comparing the inertial velocity errors, as given by Figures 7.9 and 5.4, it can be seen that the navigation accuracies are near-identical. This is because, even though the thrust pulses rapidly change the velocity of the chaser and thereby reduce the measurement accuracy

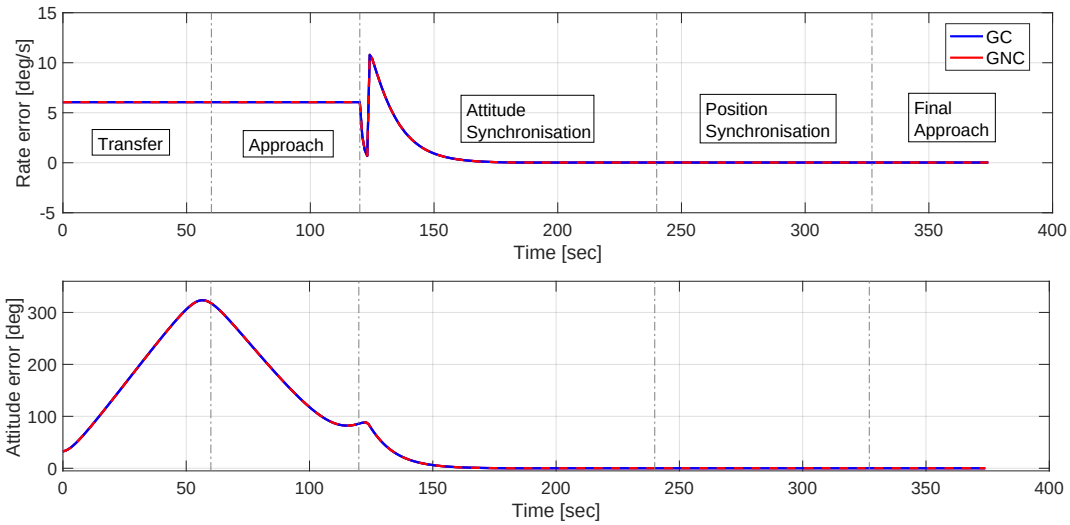


Figure 7.6: Angular rate error and attitude error during synchronisation with tumbling debris item performed by integrated guidance and control system and integrated GNC system

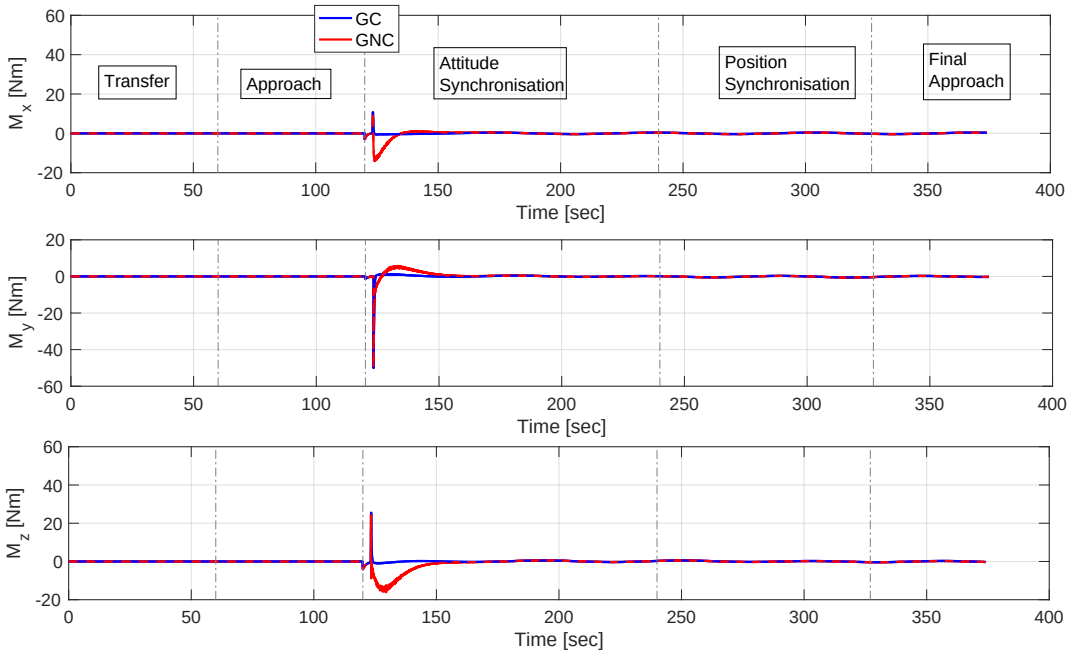


Figure 7.7: Control input history during synchronisation with tumbling debris item performed by integrated guidance and control system and integrated GNC system

of the GPS receiver, the data is fused with accelerometer measurements and a similar navigation accuracy can be obtained.

When considering the estimation errors of the inertial attitude vector, as given by Figures 7.10 and 5.5, it can be seen that the time histories of the quaternion errors are practically identical and an equal navigation accuracy can be achieved. This is due to the high measurement accuracy of the star sensors, as described in Chapter 5. However, when comparing the inertial angular velocity errors, as given in Figures 7.11 and 5.6, a large spike in the angular rate error can be identified for the integrated GNC system. When correlating this spike to the control input history of the attitude controller, as provided in Figure 7.7, it can be concluded that this increased error is obtained at the instant that the full state INDI controller is turned on. At that moment, a near-discrete jump in the control input history is commanded and the angular velocity of the chaser spacecraft changes rapidly. Consequently, the accuracy of the quaternion derivatives used for the pseudo-measurements of the spacecraft's angular velocity, as

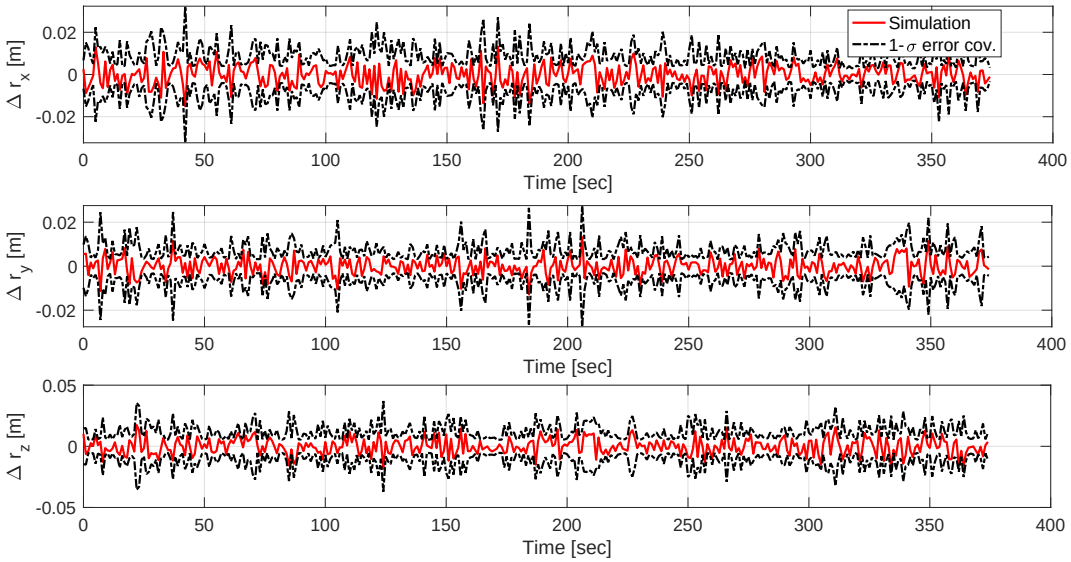


Figure 7.8: Navigation error history for inertial position estimate provided by integrated GNC system

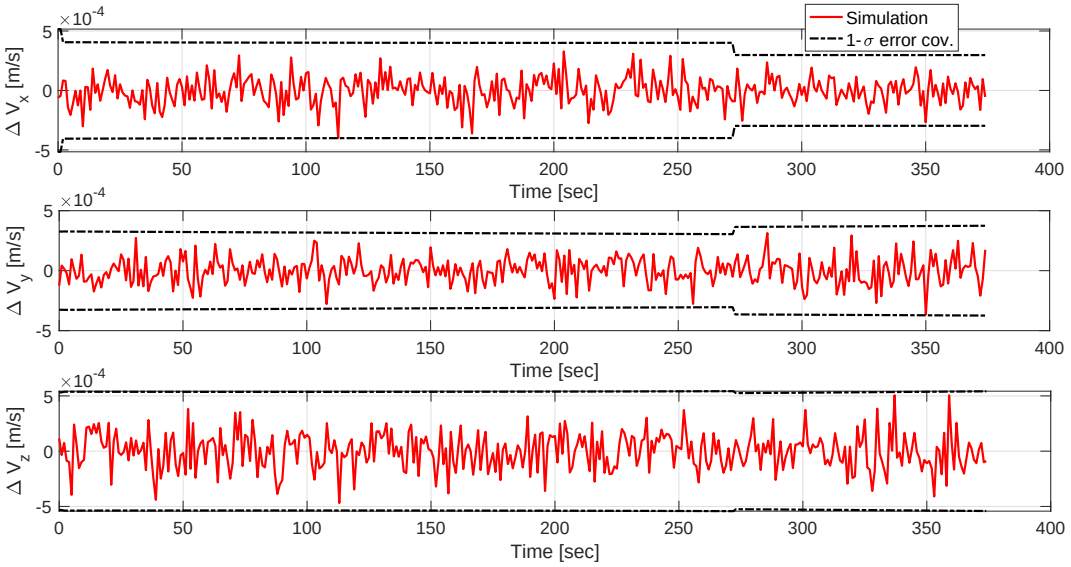


Figure 7.9: Navigation error history for inertial velocity estimate provided by integrated GNC system

given in Equations (5.42) and (5.43), deteriorates and a navigation error beyond the estimation error covariance results. Note however that, given that this behaviour is caused by the discrete switch to full state INDI controller, it will not occur during a true mission execution, as this transition will be performed more gradually. Furthermore, given that the navigation error subsequently converges back within the covariance bounds, it is currently not flagged as problematic behaviour and a future design iteration of the presented GNC system should no longer display it. Nevertheless, it is recommended to investigate the implementation of a dedicated angular velocity sensor to replace the pseudo-measurements based on star sensor derivatives. Still, it can be seen that the errors before and after this spike remain low and bounded by the covariance curves. Hence, it can be concluded that all inertial navigation error histories behave as expected and remain sufficiently accurate.

In addition to the inertial navigation error plots, also the terminal navigation accuracy is assessed, as given in Table 7.1. In this table, the terminal navigation accuracies for the inertial state estimation of the chaser is provided, along with the accuracies of the integrated navigation system for the Hill state of the chaser, as well as the inertial state of the target. For all terminal navigation errors, it can be observed that they satisfy the respective accuracy requirements of 2 cm in position, 2 mm/s in velocity, 2 deg in

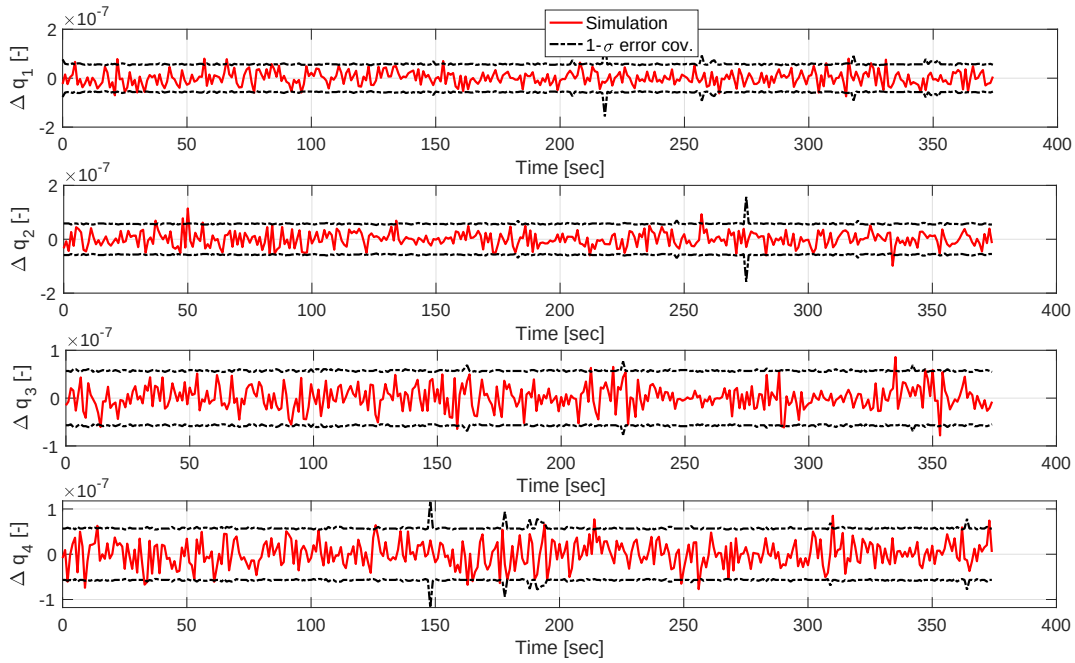


Figure 7.10: Navigation error history for inertial attitude estimate provided by integrated GNC system

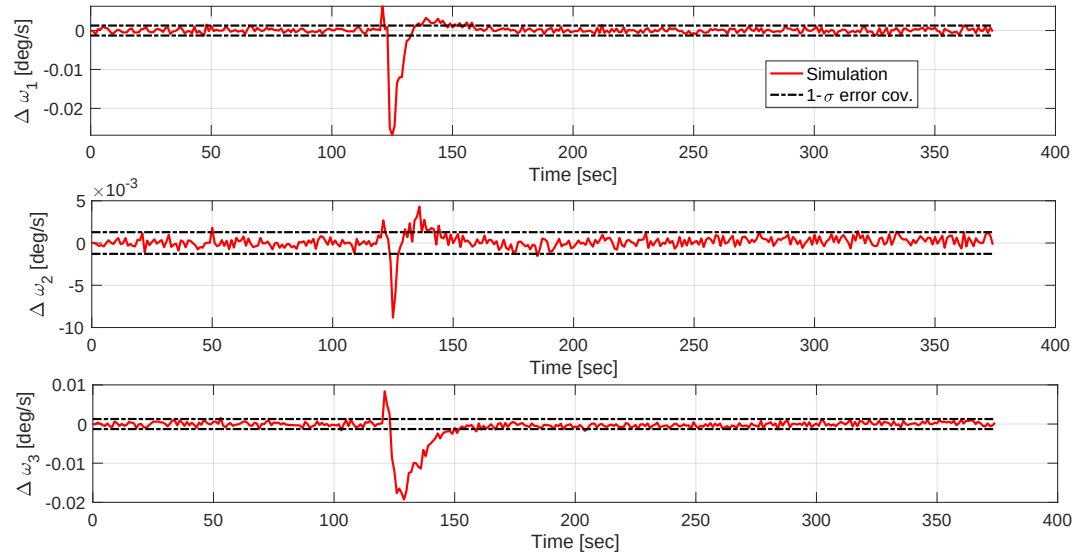


Figure 7.11: Navigation error history for inertial angular velocity estimate provided by integrated GNC system

attitude and 0.2 deg/s in angular velocity, as established in **REQ_S_C_16.1** to **REQ_S_C_16.4**. Hence, from the provided analysis, it can be concluded that the navigation system is correctly combined with the guidance and control system in the integrated GNC system.

Lastly, to assess the overall mission performance of the integrated guidance and control system and the performance of the integrated GNC system, the terminal-position error with respect to the capture position is calculated. Furthermore, also the corresponding fuel consumptions are obtained when assuming a thruster system with a specific impulse of $I_{sp} = 270$ seconds and a gravitational acceleration of $g_0 = 9.807$ m/s^2. Note that the terminal velocity dispersion is not assessed, as a quantification of this error is less relevant. This is because the chaser spacecraft will have to maintain position synchronisation with the tumbling docking-axis during the subsequent capture phase, such that the terminal velocity magnitude is not constrained to zero. A summary of the system performances is provided in Table 7.2, together with a benchmark from the isolated guidance system.

Table 7.1: Terminal navigation errors of integrated GNC system

Terminal Inertial Error Chaser	**Value**	**Terminal Inertial Error Chaser**	**Value**
position [m]	0.01	attitude [deg]	$1.7 \cdot 10^{-6}$
velocity [m/s]	$2 \cdot 10^{-4}$	angular velocity [deg/s]	$4.6 \cdot 10^{-4}$
Terminal Hill Error Chaser	**Value**	**Terminal Inertial Error Target**	**Value**
Hill position [m]	0.01	attitude [deg]	$7 \cdot 10^{-4}$
Hill velocity [m/s]	$9 \cdot 10^{-4}$	angular velocity [deg/s]	$5.3 \cdot 10^{-4}$

From this table it can be seen that the isolated guidance system results in a non-zero terminal-position error, as well as a fuel consumption that is higher than the 0.56 kg provided by the initial guidance solution in Chapter 4. This implies that the effect of external disturbances and errors in the guidance system, such as linearisation errors in the Hill equations, result in a non-recoverable terminal-position error of approximately 7 cm. This means that, for the integrated guidance and control system, and the integrated GNC system, the best achievable terminal position accuracy is thus equal to this 7 cm. When considering the combined guidance and control system and the integrated GNC system, it can be observed that an error of 8 cm results. One would however expect that, with increasing trajectory errors due to the integration of a non-ideal navigation system, the terminal-position error would increase beyond that of the combined guidance and control system. Nevertheless, this behaviour does not occur due to the recalculation power of the guidance system. Despite increased trajectory errors, the guidance system is capable of providing a new reference trajectory that, starting from a dispersed position, generates an alternative path to the capture position. Furthermore, the differences between these three position errors are limited to 1 cm and considered negligible, especially when comparing them the scale of the entire mission scenario. Therefore, it can be concluded that the provided position errors, although seeming unexpected at first, do follow from the GNC system design provided in the current thesis.

To obtain these practically identical final positions, the non-ideal systems consistently require additional propellant mass. This directly follows from expectations when considering that the decrease in control and navigation accuracy will require more intermediate trajectory corrections from the tracking guidance system, as well as larger trajectory changes commanded by the convex guidance system. Hence, it makes complete sense that increased propellant masses are required to maintain the terminal position accuracy. The inclusion of a real control system adds 243 grams to the propellant mass budget, whereas the subsequent inclusion of a real navigation system requires an additional 134 grams. Once more, this highlights the recalculation power of the convex guidance system, where a new optimal path is calculated when trajectory dispersions render the previous trajectory infeasible. Evidently, this comes at the expense of additional propellant consumption. Nevertheless, as the integrated GNC system requires only 1.2 kg of propellant, it results in 0.24% of the total mass of the ClearSpace-1 spacecraft and therefore justifies the omission of mass dynamics in the current thesis study.

Furthermore, from the analyses provided above, it can be concluded that the effect of implementing a true navigation system is limited. This can for example be seen by the similar trajectories in Figure 7.2 or the equivalence in synchronisation errors in Figure 7.6. This low impact is caused by the high performance of the navigation system and the corresponding assumptions that have been made to achieve this. Two major assumptions are given by the low range and range rate noises of the GPS receiver and the 1% relative navigation accuracy of the VBN system. Therefore, for a future work, it is recommended to analyse the realism of these two assumptions, as well as to develop an alternative solution that can potentially avoid these assumptions.

To conclude, it should be noted that the provided terminal position dispersions are not expected to jeopardise the mission, given that the chaser is equipped with robotic tentacles to ensure hard capture. It is given by Gavrilovich et al. (2013) that this docking mechanism can operate correctly, even when the chaser is not fully aligned with its target. Moreover, these robotic tentacles allow dispersions that are directly proportional to their length. However, as the characteristics of the ClearSpace-1 chaser are currently not available, an exact terminal-position error threshold cannot be provided. Nevertheless, it is expected that, given the size of the VESPA target, a dispersion of 50 cm must be feasible. This implies that the dispersions of Table 7.2 will easily fall within this threshold and it can be assumed that the integration of the overall GNC system is performed correctly. Therefore, this functional GNC

Table 7.2: Final trajectory dispersion and fuel consumption for trajectories flown by isolated guidance system, integrated guidance and control system and integrated GNC system

System Definition	Position Error [cm]	Propellant mass [kg]
Guidance	7.0	0.827
Guidance and Control	8.0	1.070
Guidance, Navigation and Control	8.0	1.204

simulator can subsequently be used to analyse the reference scenarios of the ClearSpace-1 mission in Chapter 8.

8

Integrated Simulator Results

The current chapter provides the results that can be obtained from the GNC system design presented in the current thesis study. To this extent, Section 8.1 provides an analysis on the system performance during the mission scenarios that are expected for the considered reference mission. Furthermore, Section 8.2 applies variations to this mission scenario and assesses the flexibility of the developed GNC system. Note that, similar to Chapter 7, the performance is assessed based on the terminal-position error with respect to the capture position, as well as the propellant mass consumption. To calculate these masses, a thruster system with a specific impulse of $I_{sp} = 270$ seconds and gravitational acceleration of $g_0 = 9.807$ m/s^2 is used. Furthermore, the current analyses do not consider the terminal velocity error, following the same rationale as provided in Chapter 7, nor will it display any trajectory plots, given that these would qualitatively be equivalent to Figure 4.10. Therefore, to avoid repetition, the results are provided using alternative presentations. Lastly, note that some of the resulting trajectories in the subsequent analyses slightly enter the keep out-sphere during final approach, because no formal collision avoidance manoeuvres are implemented. However, it does not result in disastrous consequences for the remainder of the active debris removal mission, as the chaser is not colliding with its target due to sufficient margin provided by the bounding surface of the keep-out sphere.

8.1. Mission Scenario Analysis

The basic mission scenario, as implemented in all preceding verification tests, consisted of a target spacecraft rotating at an initial angular velocity of $(3.5,\ 3.5,\ 3.5)^T$ deg/s. However, as indicated by **REQ_S_T_9**, this tumbling rate only represents the expected nominal spin of the target, whereas a worst-case spin of 5 deg/s over the X and Y-axis is also anticipated. Therefore, numerical simulations are performed, using the basic parameters of Table 3.1, with the only difference given by the initial angular velocity vector of VESPA, which is varied following the definitions provided in Table 8.1. Note that, on top of the nominal and worst-case spin rate, the analysis also implements two spin scenarios that fall beyond what is expected for the ClearSpace-1 mission. On one hand, a lower initial angular velocity of 0 deg/s is simulated, whereas on the other hand, a higher initial angular velocity of 20 deg/s over the Y-axis is used.

From Table 8.1, it can be seen that, with increasing magnitudes of the target's angular velocity vector, both the terminal-position error and required fuel consumption grow. This emerges from the fact that a higher spin rate will result in a more complex trajectory for the guidance system, requiring additional propellant to track the rapidly tumbling docking-axis. Furthermore, given the more complex nature of these trajectories, GNC errors will accumulate in larger trajectory dispersions, increasing the terminal-position error, as well as requiring additional intermediate trajectory corrections, increasing the propellant consumption. Hence, it can be concluded that the results provided in Table 8.1 do follow from expectations. Furthermore, it can be observed that all terminal-position errors remain low when compared to the allowed final dispersion of 50 cm, while requiring limited propellant, *i.e.*, below 1.5 kg.

However, the previously provided results only ensure that the translational motion of the chaser spacecraft adheres to the expected mission scenarios. In addition to this, also the corresponding rotational motion must be considered, given that attitude synchronisation needs to be maintained. Therefore, an analysis on the performance of the attitude controller is presented for the nominal and worst-case spin scenarios. The time histories of the angular velocity error and attitude error during both spin scenarios are provided in Figure 8.1. Furthermore, the accompanying control inputs are provided in Figure 8.2. It can be seen that, for the two spin scenarios, the attitude synchronisation manoeuvre is initiated after 120 seconds of simulation time. At this instant, the PID rate controller is switched on and

Table 8.1: Final trajectory dispersion and fuel consumption for trajectories flown by integrated GNC system during spin rate analysis of the VESPA target

Initial Angular Velocity VESPA [deg/s]	Position Error [cm]	Propellant mass [kg]
$(0, 0, 0)^T$	4.2	0.943
$(3.5, 3.5, 3.5)^T$	8.0	1.204
$(5, 5, 0)^T$	9.7	1.319
$(0, 20, 0)^T$	27.9	1.425

reduces the angular velocity error to the predefined threshold value, after which the controller switches to the full state INDI algorithm. Note that for both controllers, more time is required to achieve synchronisation during the worst-case spin scenario, when compared to the nominal scenario. Evidently, this is because a larger change is commanded for the rotational state of the chaser. Nevertheless, the control algorithm is in both cases capable of performing synchronisation with the target.

Moreover, when inspecting the control input histories of Figure 8.2, it can be observed that, when increasing the magnitude of the target's spin rate, also the magnitude of the provided control moment increases. This is again evident when considering that the synchronisation manoeuvre is directly proportional to the angular velocity vector of VESPA. However, for both scenarios, a similar input history results, where a small spike is observed during initialisation of the PID rate controller, which further evolves into a larger spike when turning on the full state INDI controller. Nevertheless, this behaviour can be explained by the fact that the initialisation of the PID rate controller and the full state INDI controller are currently implemented as discrete steps, such that spikes are observed in the control input histories. However, this is not deemed problematic when considering that a future design iteration will smoothen these signal transitions, for example by use of a sigmoid function. Therefore, it can be concluded that the control system is capable of providing attitude synchronisation for all considered spin rate scenarios of the VESPA target.

Lastly, note that the attitude synchronisation phase during the two limit cases provided in Table 8.1 have also been analysed. From this analysis, the same conclusions follow as discussed above, where the magnitudes of the control moments and the settling times of the controller responses are proportional to the magnitude of the spin velocity. Nevertheless, the attitude controller is capable of providing attitude synchronisation during these limit cases, while avoiding actuator saturation. The results are, however, omitted from Figures 8.1 and 8.2, with the dual purpose of centralising the discussion on the spin scenarios that are to be expected for the ClearSpace-1 mission, as well as avoiding clutter in the graph.

The previously provided analysis considered the performance of the integrated GNC system, *i.e.*, through the resulting terminal-position error and the propellant consumption, and the performance of the attitude control system. For the other subsystems, *i.e.*, the guidance system, the navigation system and the orbit control system, the obtained results are practically identical to those in Chapter 7. Therefore, their detailed analysis is currently omitted. Consequently, for the remaining analyses performed in this chapter, the presented results will be limited to the performance of the integrated GNC system, as it provides one figure of merit containing the performance of all subsystems, apart from the attitude control system. For the latter it should be noted that all performed simulations are capable of performing synchronisation with the target, obtaining qualitatively equivalent results to Figures 8.1 and 8.2. Therefore, any subsequent discussion on the attitude control system can be omitted.

Furthermore, it is indicated by **REQ_M_4.3** that the initial attitude of the target is undetermined. However, all simulations up to now have assumed a constant initial orientation for VESPA, which was set equal to the one provided in Table 3.1. Evidently, this has a large impact on the resulting trajectories, given that the last two mission phases, represented by the synchronisation phase and final approach phase, are dependent on the instantaneous orientation of the target. With a different initial orientation, the time history of VESPA's attitude will change, even when equal tumbling rates are applied. Therefore, to analyse the effect of the initial orientation on the performance of the GNC system, a Monte Carlo analysis is performed. In this analysis, the initial attitude of the target is randomly sampled from a normal distribution. To do so, the basic simulation parameters of Table 3.1 are used, with the initial spin rate of VESPA set to either nominal or worst-case and the initial attitude obtained from the random sampling. In total, 150 samples are drawn from a standard normal distribution using Matlab's randn()

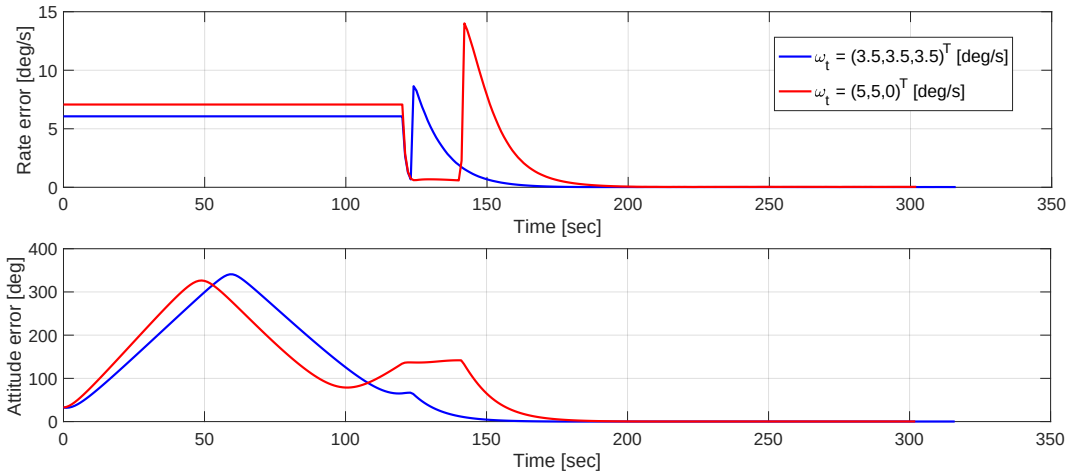


Figure 8.1: Angular rate error and attitude error of trajectories flown by integrated GNC system during spin rate analysis of the VESPA target

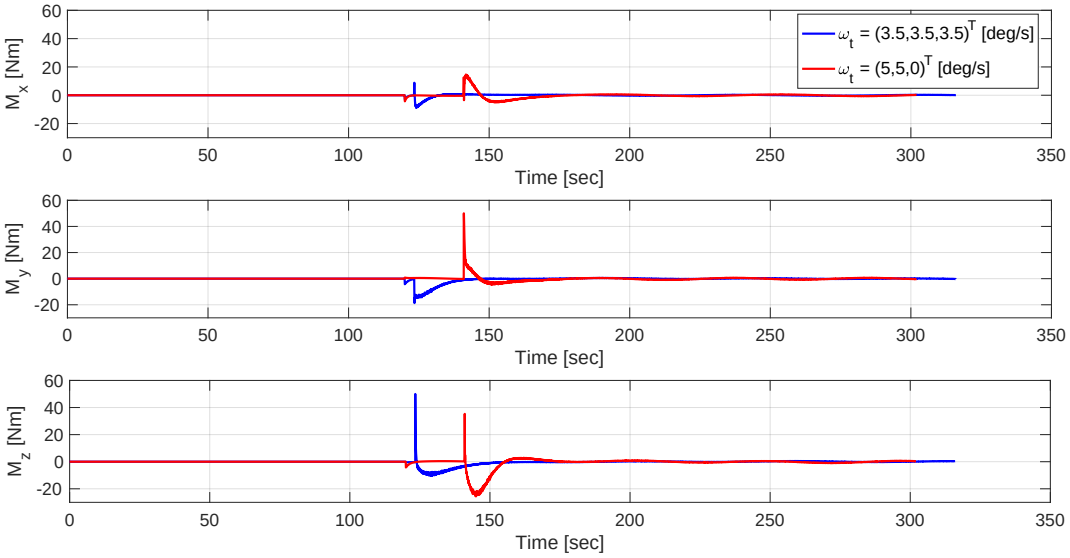


Figure 8.2: Control input history of trajectories flown by integrated GNC system during spin rate analysis of the VESPA target

function for each component in the quaternion vector. Subsequently, the samples are normalised to obtain attitude quaternions of unit length. The entire procedure is iterated using four different seeds, which are given by 12345, 20, 10 and 98, such that conclusive results can be provided. The outcome of the Monte Carlo analysis for seed 12345 is provided in Figure 8.3, where the top row presents the results for the nominal spin scenario and the bottom row for the worst-case spin scenario . Note that for other seeds, no results are provided given that comparable results to Figure 8.3 are achieved.

When inspecting Figure 8.3, it can be seen that the resulting terminal-position errors and propellant masses do not follow a true normal distribution, as given by the discrepancies in the provided kernel distribution indicated in red and the reference normal distribution indicated in green. This is because the current analysis only adopts a rather limited number of samples, as each simulation requires a relatively long computation time, clocking in at approximately 5 minutes. Furthermore, it is also not guaranteed that a normal distribution follows from a Monte Carlo analysis when the inputs are sampled from a normal distribution. Nevertheless, it can be identified from the kernel distributions that, for both spin scenarios, the terminal-position error resembles a negatively skewed normal distribution, whereas the propellant mass has a much closer match with the normal distributions. Hence it can be concluded that, to obtain a nearly identical position error, *i.e.*, within the peak of the skewed normal distribution, a wider variation of propellant masses are needed, *i.e.*, a more normally distributed propellant consump-

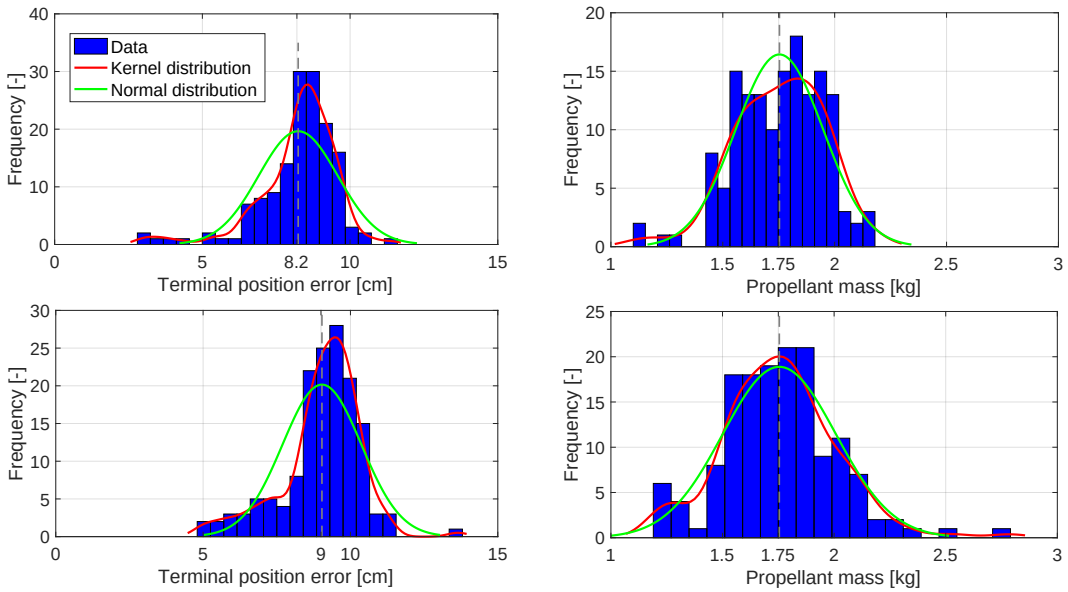


Figure 8.3: Histograms of terminal-position error and propellant consumption, obtained from Monte Carlo analysis on initial attitude of VESPA for nominal spin rate (top) and worst-case spin rate (bottom). The red curve provides a kernel distribution fitted to the provided data and the green curve a reference normal distribution with the grey dotted line as its average. Note that the Monte Carlo analysis is performed using 150 samples and a seed equal to 12345

tion is obtained. These variations in propellant masses result from the different initial orientations of the target, which define the motion of the tumbling docking-axis during the synchronisation and final approach phases. Consequently, as the convex guidance system provides a trajectory that is optimal based on the current problem definition, a variation in initial orientation of the target will cause a variation in propellant consumption. Nevertheless, the terminal-position errors are nearly identical, which showcases the power of the guidance system over the trajectory of the chaser.

Furthermore, it can be observed from Figure 8.3 that, for both spin rate scenarios, the terminal-position error remains below 15 cm. Yet, a slightly higher average is observed for the worst-case spinning VESPA, when compared to the nominally spinning VESPA. Additionally, when considering the propellant consumptions, it can be identified that the averages of both distributions are equal. However, for a nominally spinning VESPA, some 'outliers' are found below the average propellant consumption. Conversely, for the worst-case spinning VESPA, an 'outlier' can be found above this average consumption. Nevertheless, all propellant masses remain well below 3 kg. Therefore, it can be concluded that these obtained results correspond well with the angular velocity analysis of Table 8.1, where an increased terminal-position error and propellant consumption are expected for a worst-case spin scenario. However, both scenarios provide bounded results, indicating that the GNC system is capable of executing a close-range rendezvous with VESPA, independent of the 150 initial orientations that have been simulated. Consequently, it can be assumed that the developed GNC system is capable of executing all nominal mission scenarios that are to be expected during the close-range rendezvous phase of the ClearSpace-1 mission.

8.2. Robustness Analysis

In addition to analysing the performance of the GNC system during the nominal mission scenario, its robustness is examined using varying mission conditions. To this extent, two analyses are performed, where the orientation of the docking-axis in the body frame of the target is altered, as well as the initial position of the chaser used at the start of the simulation. The former analysis is performed to ensure flexibility of the developed GNC system to an in-orbit redefinition of the mission execution, when a visual inspection of the target prior to initiation of the close-range rendezvous phase uncovers that an alternative approach direction should be employed. Reasons to redefine this approach strategy can be given by varying lighting conditions, communication availability or contingencies in the mechanical state of the rocket body, as these might be difficult to assess from ground. Therefore, different docking-

axis definitions are analysed. Moreover, a second analysis is performed to ensure flexibility of the developed GNC system to position dispersion that accumulate during the far-range rendezvous phase. As it is inevitable that the phases preceding the close-range rendezvous will contain trajectory errors, it is necessary that the developed GNC system is capable of handling dispersion in its initial position. Therefore, different initial positions are analysed.

During the docking-axis analysis, 14 independent simulations are run using the basic simulation parameters defined in Table 3.1, in combination with the nominal and worst-case spin scenarios. A constant initial orientation is applied for the target spacecraft over the different simulations, such that the isolated effect of the docking-axis orientation on the overall GNC performance can be measured. During this analysis, the docking-axis orientations constitute a symmetrical distribution with respect to the conical body of the chaser, with the aim of providing a most general conclusion. A graphical description of these docking-axes is provided in Figures 8.4 and 8.5, for the nominal and worst-case spin scenario, respectively. Additionally, these figures also indicate the terminal-position error and the propellant consumption that result from approaching the target along the particular docking-axes.

When inspecting both figures, it must be noted that it is difficult to obtain a direct pattern between the docking-axis definition and the performance of the GNC system. Some axis orientations result in an increased terminal-position error, but do not necessarily transfer from one spin scenario to another. Furthermore, when inspecting the variations in terminal-position error, it can be seen that this is limited to 1 cm for both scenarios. When putting this into perspective with the full of the scale close-range rendezvous phase, it can be concluded that this variation is negligible. Therefore, it is evident that no patter is observable.

Furthermore, some axis orientations result in an increased propellant mass consumption, but do not per definition result in an increased terminal-position error, nor transfer over the two spin scenarios. However, a larger variation in propellant consumption can be observed, when compared to the position error variation. This is because the definition of the docking-axis defines the two most complex mission segments of the close-range rendezvous phase, given by the synchronisation segment and final approach segment. Therefore, a change in the docking-axis definition will completely alter the last part of the flown trajectories, resulting, as expected, in big propellant mass variations. Nevertheless, as the terminal-position errors hardly change, it showcases how powerful the recalculation and correction actions of the guidance system are. Furthermore, it can be observed that the largest position error is limited to 10 cm, whereas the maximum propellant consumption equals 2 kg. Hence, the developed GNC system is capable of flying the close-range rendezvous phase with a freely chosen docking-axis, as robust outcomes are obtained for the 14 different orientations. A real choice of an optimal docking-axis definition is, however, outside the scope of the current thesis work, as it is dependent on lighting and communication conditions, which are currently omitted. Nevertheless, a future design iteration can employ Figures 8.4 and 8.5 to provide an indication on a preferred approach direction.

To ensure the flexibility of the developed GNC system to position dispersions, the initial position is varied along a circle with a 20 m radius in the combined R-bar V-bar plane of the target's Hill frame. Dispersions along the H-bar are omitted, given that the first mission segment always transfers to the positive H-bar, as this is considered the safest approach direction. Therefore, it is expected that errors along this direction do not form any issues for the flexibility of the guidance system, justifying their omission. Furthermore, by employing this 20 m radius, the simulated initial dispersions largely exceed the terminal-position errors that are expected from the far-range rendezvous phase, which result in a magnitude of only 2 m when assuming a 1% relative navigation accuracy. Consequently, a conservative analysis can be obtained and a conclusive verdict can be formed. To do so, 24 independent simulations are performed, using the basic simulation parameters defined in Table 3.1, and the initial position of the chaser varied along a circular path around its target. Note that the simulations are performed using a constant initial orientation, such that the effect of initial position dispersions on the overall GNC system performance can be isolated. Furthermore, note that the current analysis is not provided as an optimisation of the starting configuration, even though the presented GNC system design provides the necessary tool to do so. This is because radically changing the initial position of the chaser is currently not considered as relevant, given that only a hold point on the V-bar direction can be considered as stable, *i.e.*, does not require the expulsion of propellant to maintain position. Therefore, this initial configuration provides the ideal starting position preceding the close-range rendezvous phase, as it allows for initialisation procedures to be performed, such as a final go/no-go decision to be uploaded from ground control, at the expense of zero propellant mass.

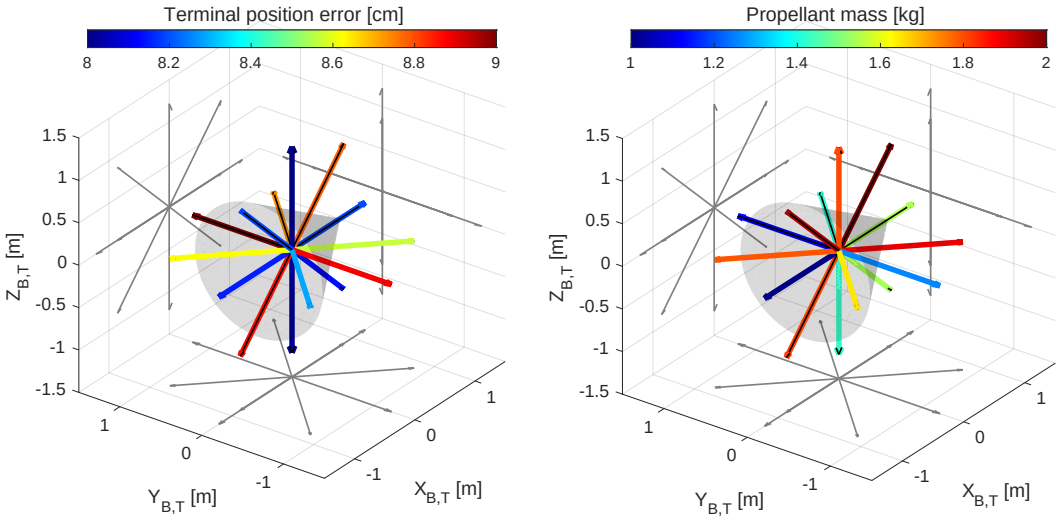


Figure 8.4: Terminal-position error (left) and propellant consumption (right) obtained from docking-axis analysis at nominal spin rate. Note that the grey cone represents the conical body of the VESPA target and the arrows indicate the respective definitions of the docking-axes in the Body-Centred-Body-Fixed frame of the target.

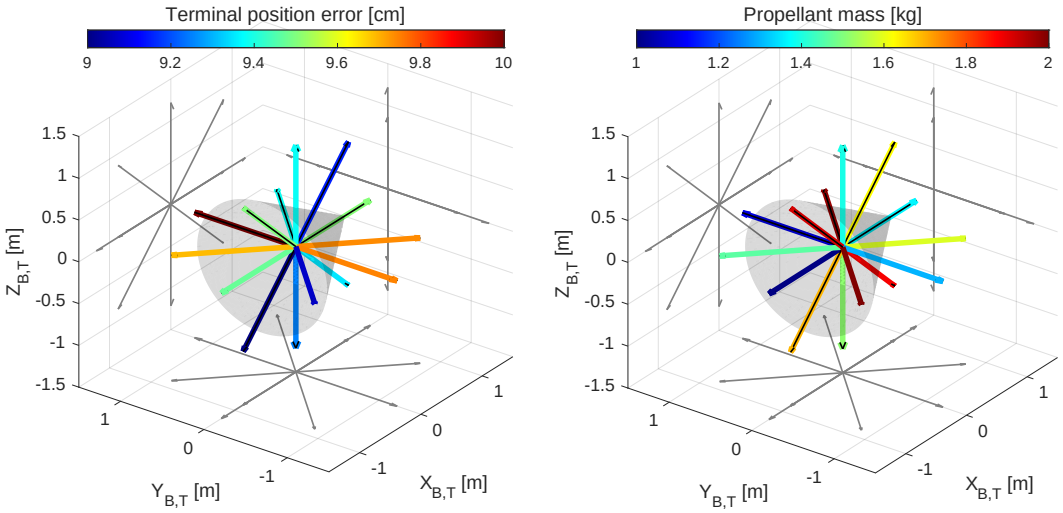


Figure 8.5: Terminal-position error (left) and propellant consumption (right) obtained from docking-axis analysis at worst-case spin rate. Note that the grey cone represents the conical body of the VESPA target and the arrows indicate the respective definitions of the docking-axes in the Body-Centred-Body-Fixed frame of the target.

The results that are obtained from this initial position analysis during a nominal spin scenario are provided in Figure 8.6. A graphical representation for the worst-case spin scenario is omitted, as results equivalent to this figure are obtained. When inspecting the Figure 8.6, it can be concluded that no direct pattern between the chaser's initial position and its GNC system performance can be obtained. As for the docking-axis analysis, some initial positions result in increased terminal-position errors, but these differences are not per definition applicable to both spin scenarios, nor large in magnitude, especially when considering the scale of the initial position differences. Therefore, it makes sense that no direct pattern in observable. However, in contrast to the docking-axis analysis, the variation in propellant consumption is much lower and can be explained by the fact that the all trajectories will move towards the H-bar direction during the transfer phase, and from there, fly identical trajectories to the capture position. Hence, only the initial segment is different, resulting in the low variations. This is because the transfer phase represents a straightforward trajectory segment that can be achieved using near-impulsive manoeuvres.

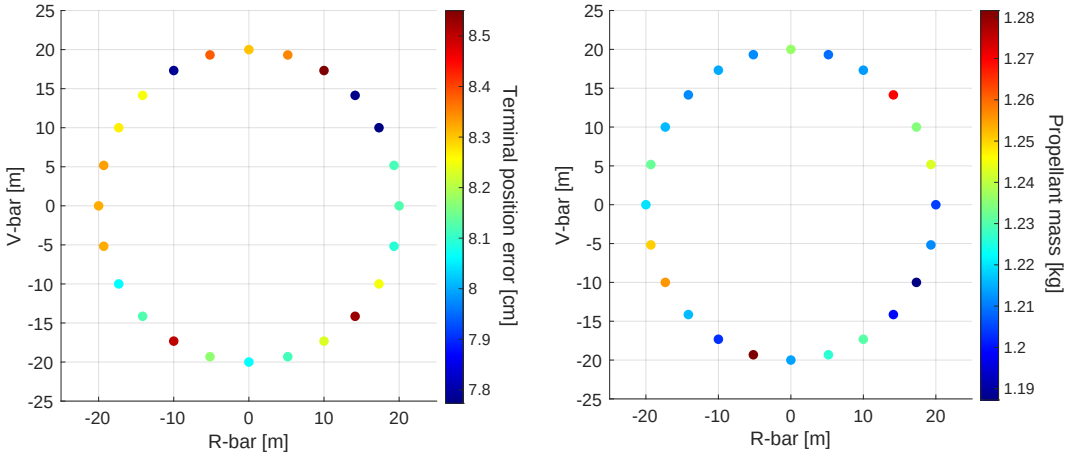


Figure 8.6: Terminal-position error (left) and propellant consumption (right) obtained from initial position analysis of ClearSpace-1 chaser during nominal spin rate of VESPA

Moreover, to increase the confidence in robustness to initial state dispersions, a Monte Carlo analysis is performed which varies the initial state of the spacecraft using realistic state errors derived from the far-range rendezvous phase. Therefore, the initial state error of the chaser spacecraft is randomly sampled from a normal distribution, with zero mean and a 2 m standard deviation for each position component, and a 0.5 m/s standard deviation for each velocity component. To do so, the basic simulation parameters of Table 3.1 are used together with a nominal or worst-case spin rate for VESPA. Once more, 150 samples are drawn from a standard normal distribution using Matlab's randn() function and the entire procedure is iterated using four different seeds. For brevity purposes, however, a detailed analysis of these results is omitted, given that, for both spin scenarios, the terminal-position errors are limited to 15 cm and the propellant consumption is limited to 2 kg, similar to the analyses provided above. Furthermore, the variations are near-identical to what is provided in Figure 8.6. Therefore, the Monte Carlo analysis indicates that the GNC system is sufficiently flexible to any position dispersion that can be expected from the far-range rendezvous phase.

Hence, from the analyses performed in the current chapter, it can be concluded that the developed GNC system is capable of executing the mission scenario of the ClearSpace-1 mission, even when certain mission parameters vary. To achieve a terminal position accuracy in the order of $\mathcal{O}(0.1)$ m, a maximum of 2 kg of propellant is needed. As this propellant mass results in only 0.4% of the total vehicle mass, an omission of the mass dynamics in the current thesis study is therefore once more justified. However, to increase the confidence in the performance of the GNC system, it is recommended to perform further robustness analyses. During these, the flexibility of the GNC system to varying mass and inertia properties must be assessed. In addition to this, the GNC system performance must be analysed under changing error conditions for the navigation sensors, along with changing error magnitudes for the control actuators. Moreover, a subsequent analysis must be performed that combines all the aforementioned errors. However, due to the time constraints, this must be left for future work.

9 Conclusions and Recommendations

The current chapter signifies the end of the presented thesis study. In this chapter, conclusions that can be drawn from the GNC system development are provided in Section 9.1, where its status quo is assessed through answering the research questions that were prompted at the beginning of the report. Furthermore, to move this status quo forward, recommendations for future work are provided in Section 9.2. These recommendations will need to be tackled in a subsequent research to further develop the GNC system for an implementation onboard an actual active debris removal mission.

9.1. Conclusions

Throughout the current thesis report, the capabilities of a guidance, navigation and control system for an active debris removal mission have been addressed. The GNC system was designed with the goal of obtaining a robust, safe, autonomous and fuel-efficient execution of a final rendezvous phase between an active chaser spacecraft and a passive debris item, as given by the initial research question.

- *How can a real-time guidance, navigation and control system be applied to robustly, safely and autonomously execute a fuel-efficient final rendezvous between an active chaser spacecraft and a passive space debris?*

This main question has, however, been broken down into multiple subquestions, which were used to guide the overall design of the presented GNC system. Therefore, answers to each of these lower level questions are provided in the discussion below and culminate into an answer of the main research question given above.

- *What performance can be obtained in terms of robustness, safety, autonomy and fuel-efficiency when employing a convex guidance system?*

When considering the robustness of the developed guidance system, it must be noted that the results of Chapter 8 indicate that the system is capable of executing the ClearSpace-1 mission for all anticipated missions scenarios, *i.e.*, using an unknown initial orientation for the target spacecraft combined with either a nominal spin scenario or a worst-case spin scenario. Furthermore, variations to these mission scenarios were performed in terms of different docking axes definitions, as well as initial state dispersions. Nevertheless, during all simulations, the GNC system was capable of executing the mission by achieving a terminal position accuracy in the order of $\mathcal{O}(0.1)$ m, while using a maximum of 2 kg of propellant. Moreover, even when simulating limit cases that fall beyond the scope of the ClearSpace-1 mission, the GNC system managed to obtain feasible results, both in terms of terminal-position error, as well as in terms of propellant consumption. This high level of robustness is, however, achieved through the recalculation and correction power of the guidance algorithm, and indicates its high level of autonomy.

All provided simulations have been run from initial conditions to terminal conditions without any human interference. Dispersions in the trajectory are allowed, as intermediate corrections from the tracking system guided the spacecraft back to its reference, such that a convex recalculation can occur. Crucial here is the impact of the tracking guidance system, which is needed to ensure robustness and autonomy by keeping the chaser spacecraft close to its reference trajectory. If this is omitted, the resulting guidance performance rapidly deteriorates, as the convex system would not be able to converge to a feasible solution and results in a freely drifting chaser spacecraft. Conversely, when the

chaser remains close to this reference trajectory, no infeasible recalculations are obtained from the convex guidance system, thanks to the lack of convexification strategy needed for the developed algorithm. This is in contrast to lossless convexification, where infeasible trajectories can result when the lifted solution does not transfer back to the feasible design space of the original problem. Alternatively, when applying successive convexification, infeasibility can result due to the lack of convergence in the successive approximations or from linearisation errors. Hence, by effectively avoiding these shortcomings in the convexification strategies, a robust and autonomous guidance system could be created.

Moreover, when assessing the safety of the developed guidance system, it must be noted that, due to time constraints, no formal collision avoidance manoeuvres have been included. Yet, from the performed analyses it was found that, although the chaser spacecraft might slightly enter the keep-out sphere during final approach, no collisions actually fired due to sufficient margin between the bounding surface of the keep-out sphere and the body of the target. Furthermore, the implementation of the tracking guidance system ensures that the chaser is never freely drifting, reducing the probability of an accidental collision. Therefore, it can be considered that the implemented system is pseudo-safe. Nevertheless, it is highly recommended for future work that a formal collision avoidance algorithm is implemented, such that no accidental collisions fire would the entire guidance system fail.

Furthermore, when considering the fuel-efficiency, it was provided by the verification procedures of Chapter 4 that the implemented convex guidance system is capable of acquiring the fuel-optimal solution of its provided rendezvous problem. Furthermore, from the various propellant masses presented in the current thesis report, it can be seen that a maximum of 2 kg of fuel is used. Although this propellant consumption is also partly achieved by the tracking guidance system, its value remains sufficiently low to conclude that, not only the convex guidance system, but the overall combined guidance system is capable of finding a fuel-efficient solution to all considered mission scenarios. Hence it can be assumed that the developed guidance system is indeed robust, safe, autonomous and fuel-efficient. Nevertheless, to formally quantify these benefits, a comparative analysis with a second guidance system should be performed, but falls outside the scope of the current thesis project. Instead, it is recommended for a future design iteration, as it is expected that the presented guidance algorithm will easily outperform any more conventional guidance system it is compared to.

Lastly note that, when considering the real-time implementation of the developed GNC system, as required by the main research question, it is stated in Chapter 8 that a single simulation run of the complete system takes approximately 5 minutes. This corresponds with the approximate 300 seconds of mission time, as provided in the different time histories of the current report. Therefore, a real-time implementation is considered as achievable, especially when considering that no speed-oriented code has currently been implemented. Note, however, that this real-time potential has been made possible due to the rapid convergence properties intrinsic to the convex guidance methodology.

- *What is the effect on the GNC system performance when implementing the convex guidance system with an actual navigation and control system?*

An assessment on the effect of integrating the developed convex guidance system with a true control and navigation system has been performed during the integration analyses of Chapter 7. From this analysis it was found that the increase in terminal-position error is marginal when integrating the convex guidance system with real control and/or navigation systems. Nevertheless, an increase in propellant consumption results and follows from a decrease in control and navigation accuracy. With these accuracy reductions, larger changes are commanded from trajectory recalculations, as well as more intermediate trajectory corrections are required. Evidently, an increase in propellant consumption results, although it remains bounded. This is because the pulse-width modulation algorithm of the reaction control thrusters can operate at a sufficiently high frequency, limiting the thrusting inaccuracy, and because a high navigation accuracy can be achieved with the developed navigation system.

Nonetheless it should be noted that, for the developed convex guidance system, sufficiently accurate orbit control thrusters are required. This is because, in case non-ideal thrusters are used, the performance of the convex guidance system quickly degenerates and the recalculation power over the reference trajectory is lost. Calculated commands are not accurately executed by the thruster system, causing larger dispersions with respect to the reference trajectory until the convex guidance system is eventually incapable of calculating a feasible solution. Moreover, when implementing non-ideal thrusters, also the current tracking guidance system will not be able to guide the chaser back to its reference trajectory, resulting in a loss of the previously discussed autonomy and robustness.

Hence it can be concluded that the performance of the convex guidance system is very sensitive to the orbit control accuracy. Therefore, it is recommended for future work to implement an in-orbit thruster calibration algorithm when using non-ideal reaction control thrusters and/or an adaptive controller that can handle these thrusting errors.

When considering the effect of the navigation system on the performance of the convex guidance algorithm, it is concluded that its impact is low. This is because a highly accurate navigation system could be implemented by making two major assumptions. Firstly, to obtain accurate inertial position estimates, a high performance GPS receiver has been modelled. Secondly, a vision-based navigation system has artificially been implemented with a 1% relative navigation accuracy. These assumptions might not be state of the art and will therefore have to be developed for the mission. More details are, however, provided for the next subquestion related to the navigation system.

- *What minimum sensor suite is required to allow successfull operation of the GNC system?*
- *What performance is required from the navigation sensors to allow successfull operation of the GNC system?*

To ensure successfull operation of the developed GNC system, the minimum sensor suite consists of one IMU, one GPS receiver and two star sensors. These double star senors are required to achieve sufficient navigation accuracy for the chaser's angular velocity estimation, by fusing pseudo-measurements of the angular velocity vector from two distinct quaternion derivatives. Still, even when implementing two star sensors, the obtainable navigation accuracy reduces considerably when the chaser starts to encounter near-discrete changes in its angular velocity profile, which occurs during the switch from PID rate controller to full state INDI control. Therefore, it is recommended for future work to implement a signal smoother that removes the discrete nature of this switch, improving the control input history of the attitude controller and the estimation accuracy of the inertial angular velocity vector. Moreover, it is also recommended to investigate a dedicated sensor that can obtain accurate angular velocity measurements, to replace the pseudo-measurements based on quaternion derivatives, by considering either a second IMU or a dedicated gyroscope.

Furthermore, when analysing the measurement accuracy that is required from the proposed minimum sensor suite, it should be noted that the error models for the IMU and star sensors are taken from representative values in literature, *i.e.*, those sensors could represent current state of the art. However, for the GPS receiver, representative error parameters are chosen for all but the range and range rate noise parameters. Instead, it was required to set these latter two parameters to low values, *i.e.*, 10^{-3} m and 10^{-4} m/s respectively, to achieve the strict terminal navigation accuracy requirements on cm-level for position and mm/s-level for velocity. Therefore, for a true implementation of the developed GNC system, either a high-performance GPS receiver needs to be developed or an alternative approach must be considered. This is, however, outside the scope of the current study and left as future work. Furthermore, for the artificial vision-based navigation system, a 1% navigation accuracy was implemented. However, to assess the realism of this accuracy level, a more profound knowledge on computer vision systems is needed and an own algorithm should be developed for an ADR scenario. Evidently, also this is recommended for future work. Nevertheless, it is obtained from the RemoveDEBRIS mission that ground processing of a VBN system can result in centimetre-level position accuracy and degree-level attitude accuracy. However, if this is obtainable from an onboard implementation still remains to be investigated.

- *What minimum actuator setup is required to allow successfull operation of the GNC system?*
- *What performance is required from the control actuators to allow successfull operation of the GNC system?*

For successfull operation of the developed GNC system, the chaser spacecraft needs to be equipped with an omnidirectional orbit control system, together with an independent attitude control system. This is needed as the current guidance system cannot take the rotational motion of the chaser spacecraft into account. Therefore, 24 reaction control thrusters are implemented and grouped in pods of three on each vertex of the chaser's body. Furthermore, as established above, modulated thrust commands can be sent to these reaction control thrusters, but their provided output must be modelled as ideal. When non-ideal thrusters are used, either an in-orbit calibration algorithm must be developed, or an adaptive controller must be instituted to handle the actuator errors of the thruster system. Moreover,

when considering the attitude control system, three independent reaction wheels are needed to obtain full control over the chaser's attitude, where the implemented reaction wheel models are allowed to contain realistic error values but used an ideal alignment. Nevertheless, it is expected that the INDI attitude controller will also be able to handle further actuator errors, due to its high level of robustness obtained by applying a sensor-based methodology. Therefore, an in-orbit calibration of the reaction wheel system is not deemed essential and an adaptive attitude controller must not be developed.

In addition to providing direct answers to these research questions, the current thesis project also developed results that fall outside their direct scope, but were necessary to obtain the presented answers. Most important is the development of extensions to the GRADS software library, such as the solar radiation pressure and aerodynamic drag models, navigation filters and control algorithms. In addition to that, the current work also provided a first implementation of the GRADS library for a functional GNC system implementation and can therefore serve as a baseline during future studies on not only the ClearSpace-1 mission, but more generic active debris removal and in-orbit servicing missions. This is because the developed software blocks contain modularity, where for example the conical disturbance models can be used for truncated stage adaptors, by defining a distinct top and bottom radius, as well as for cylindrical rocket stages, by defining two identical radii. Moreover, the developed guidance algorithm does not include any specifics of the VESPA target in the formulation of its guidance problem, as could for example be seen from the possibility of simulating different initial orientations, angular velocities or docking-axis definitions. Therefore, it can be concluded that the developed GNC system is sufficiently generic to be used for any ADR mission with a rocket body, under the condition that some constraints of the convex guidance problems will have to be scaled to the actual dimensions of the considered body. Apart from that, sufficient generality is implemented such that the overall architecture of the GNC system can remain identical. Furthermore, the developed GNC system also allows to analyse ADR missions to defunct satellites, under the condition that a dedicated constraint is added to take care of their flexible appendages. Alternatively, the size of the final keep-out sphere can be increased such that it encompasses these appendages.

Hence in conclusion, it can be stated that a real-time GNC system can be applied to the final rendezvous phase of an ADR mission in a robust, safe, autonomous and fuel-efficient manner, by applying the presented GNC system, not only for the ClearSpace-1 mission but for any generic ADR mission.

9.2. Recommendations

To further develop the currently presented GNC system, a number of recommendations can be made. Note that no detailed description is provided for these recommendations, given that they are either self-explanatory or have previously been discussed jointly with the topic they relate to in this thesis report. Furthermore, some recommendations directly transfer from the applied research restrictions, as discussed in Chapter 2. Therefore, a simple listing of recommendations is made per subject.

Motion Simulator

- More extensive verification and validation procedures for self-developed solar radiation pressure and atmospheric drag models
- Implementation of plume impingement interaction between reaction control thruster of chaser and body of target
- Implementation of mass dynamics for chaser spacecraft

Guidance System

- Capture phase guidance based on an adaptive control, as for example given by Zhu and Wang (2020)
- Dedicated golden section search algorithm that speeds up entire initialisation process, as for example given by Leomanni et al. (2022)
- Dedicated SOCP solver that avoids interfacing software
- Formal tuning of control gains used in PID tracking guidance
- Implementation of collision avoidance manoeuvre that ensures chaser does not enter keep-out sphere, where thrusting commands are executed when a certain proximity along docking-axis is achieved

- Addition of plume impingement constraint
- Addition of appendage constraint for ADR with defunct satellites

Control System

- Formal tuning of the control gains per reference axis, as used in PID rate controller and full state INDI attitude controller
- Implementation of transition smoother during initialisation of PID rate controller and full state INDI attitude controller
- Implementation of non-ideal reaction control thruster system
- Implementation of adaptive controller to handle actuator errors of reaction control thruster system
- Implementation of non-ideal reaction wheel system
- Implementation of MRP-based INDI controller without oscillatory behaviour
- Implement a PID rate controller that can operate at a frequency of 4 Hz in combination with an INDI controller operating at 100 Hz
- Implementation of discretely sampled control commands with potential time delays
- Implementation of discretely sampled actuator outputs with potential time delays
- Conversion of second-order filters from continuous time to discrete time using Tustin's method

Navigation System

- Simultaneous in-orbit calibration of navigation sensors and control actuators
- Alternative inertial navigation system design, avoiding high accuracy requirements for GPS receiver and pseudo-measurements of angular velocity from star sensor quaternions
- Implementation of discretely sampled navigation sensors with potential time delays
- Implementation of navigation sensor outages, for example due to blockage of signal by target
- Implementation of a realistic vision-based navigation system
- Implementation of navigation system that simultaneously estimates the inertial and relative states

System Analysis

- Real-time implementation analysis using speed-oriented code, such as C or C++
- Comparative analysis of developed GNC system with conventional GNC system to quantify obtainable benefits in terms of robustness, safety, autonomy and fuel-efficiency
- Robustness analysis that varies mass and inertia properties of chaser and target spacecraft
- Robustness analysis that varies error magnitudes of navigation sensors and control actuators
- Robustness analysis that combines all aforementioned variations, including those performed in Chapter 8
- System analysis that assesses lighting conditions and communication opportunities during mission execution

References

Acquatella, P., Falkena, W., Van Kampen, E., & Chu, Q. Robust Nonlinear Spacecraft Attitude Control using Incremental Nonlinear Dynamic Inversion. In: *AIAA Guidance, Navigation, and Control Conference*. AIAA. 2012.

Adams, R. (1995). *Calculus: A Complete Course* (3rd ed.). Addison-Wesley Publishers Ltd.

Ahmadi, M., Khayatian, A., & Karimaghaee, P. (2012). Attitude estimation by divided difference filter in quaternion space. *Acta Astronautica*, *Vol. 75*, pp. 95–107.

Arya, B., Laila Beebi, M., & Johnson, Y. (2016). Satellite Tracking Control under J2 Perturbations. *International Journal of Scientific and Engineering Research*, *Vol. 7*(No. 4), pp. 622–627.

Bacon, B. Quaternion-Based Control Architecture for Determining Controllability/Maneuverability Limits. In: *Proceedings of the 2012 AIAA Guidance, Navigation, and Control Conference*. AIAA. 2012.

Beyer, W. (1987). *CRC Standard Mathematical Tables* (28th ed.). CRC Press.

Bhagat, M. (2016). *Convex Guidance for Envisat Rendezvous* (Master's thesis). Delft University of Technology. Delft, The Netherlands.

Biesbroek, R., Aziz, S., Wolahan, A., Cipolla, S., Richard-Noca, M., & Piguet, L. The ClearSpace-1 Mission: ESA and ClearSpace Team Up To Remove Debris. In: *8th European Conference on Space Debris*. ESA. 2021.

Biesbroek, R., Innocenti, L., Wolahan, A., & Serrano, S. e.Deorbit – ESA's Active Debris Removal Mission. In: *7th European Conference on Space Debris*. ESA Space Debris Office. 2017.

Blackerby, C., Okamoto, A., Fujimoto, K., & Okada, N. ELSA-D: An In-Orbit End-Of-Life Demonstration Mission. In: *69th International Astronautical Congress*. IAF. 2018.

Blackmore, L. (2016). Autonomous precision landing of space rockets. *The Bridge*, *Vol. 46*, pp. 15–20.

Bodson, M., & Frost, S. (2011). Load Balancing in Control Allocation. *Journal of Guidance, Control, and Dynamics*, *Vol. 34*, pp. 380–387.

Bonnal, C., Ruault, J., & Desjean, M. (2013). Active debris removal: Recent progress and current trends. *Acta Astronautica*, *Vol. 85*, pp. 51–60.

Boyd, S., & Vandenberghe, L. (2004). *Convex Optimisation*. Cambridge University Press.

Branco, J., Colmenarejo, P., Serra, P., Lourenço, P., Peters, T., & Mammarella, M. Critical GNC Aspects for ADR missions. In: *11th International ESA Conference on Guidance, Navigation & Control Systems*. ESA. 2021.

Canuto, E., Novara, C., Massotti, L., Perez Montenegro, C., & Carlucci, D. (2018). *Spacecraft dynamics and control: the Embedded Model Control approach*. Elsevier.

Carrara, V., & Kuga, H. (2013). Estimating Friction Parameters in Reaction Wheels for Attitude Control. *Mathematical Problems in Engineering*, *Vol. 2013*.

Carter, T., & Humi, M. (2002). Clohessy-Wiltshire Equations Modified to Include Quadratic Drag. *Journal of Guidance, Control and Dynamics*, *Vol. 25*(No. 6), pp. 1058–1063.

Cattani, B. (2022). *Enhancing Collision Avoidance: Software and Hardware Techniques to Support and Validate GNC Developments* (Master's thesis). Delft University of Technology. Delft, The Netherlands.

Cavrois, B., Vergnol, A., Donnard, A., Casiez, P., Southivong, U., Mongrard, O., Ankersen, F., Pezant, C., Bretécher, P., Kolb, F., & Windmüller, M. LIRIS Demonstrator on ATV5: A Step Beyond for European Non Cooperative Navigation System. In: *AIAA Guidance, Navigation, and Control Conference*. AIAA. 2015.

Cheemi. (2022). *Current Transducer CHK_BS15D4H*. Retrieved September 6, 2022, from http://www.cheemi-tech.com/product-current-transducer-CHK_BS12D4H.html

Curtis, H. (2013). *Orbital Mechanics for Engineering Students* (3rd ed.). Elsevier.

Davis, T., & Melanson, D. XSS-10 microsatellite flight demonstration program results. In: *SPIE Spacecraft Platforms and Infrastructure Conference*. SPIE. 2004.

Dennehy, C., & Carpenter, J. (2011). *A summary of the Rendezvous, Proximity Operations, Docking and Undocking (RPODU) Lessons Learned from the Defense Advanced Research Project Agency (DARPA) Orbital Express (OE) Demonstration System Mission* (tech. rep.). NASA.

E. Smeurs. (2018). *Incremental Control of Hybrid Micro Air Vehicles* (Doctoral dissertation). Delft University of Technology.

Efimov, S., Pritykin, D., & Sidorenko, V. (2018). Defunct Satellites in Nearly Polar Orbits: Long-term Evolution of Attitude Motion. *Baltic astronomy an international journal*, *Vol. 27*, pp. 264–277.

ESA. (2021a). *DART: Rendezvous*. Retrieved May 30, 2021, from https://eoportal.org/web/eoportal/satellite-missions/d/dart-rendezvous

ESA. (2021b). *ETS-VII*. Retrieved May 30, 2021, from https://eoportal.org/web/eoportal/satellite-missions/e/ets-vii

ESA. (2021c). *ISS services: ATV-5*. Retrieved May 30, 2021, from https://directory.eoportal.org/web/eoportal/satellite-missions/i/iss-atv-5

ESA. (2021d). *RemoveDebris*. Retrieved May 30, 2021, from https://directory.eoportal.org/web/eoportal/satellite-missions/r/removedebris

ESA. (2021e). *STP-1*. Retrieved May 30, 2021, from https://earth.esa.int/web/eoportal/satellite-missions/s/stp-1

ESA. (2021f). *XSS*. Retrieved May 30, 2021, from https://eoportal.org/web/eoportal/satellite-missions/v-w-x-y-z/xss

ESA. (2022). *ELSA-d*. Retrieved July 27, 2022, from https://directory.eoportal.org/web/eoportal/satellite-missions/content/-/article/elsa-d

European Space Research and Technology Centre. (2020). ESA Express Procurement: On-board Constrained Optimisation of Final Rendezvous [Internal project document].

Forshaw, J., Aglietti, G., Salmon, T., Retat, I., Burgess, C., Chabot, T., Pisseloup, A., Phipps, A., Bernal, C., Chaumette, F., Pollini, A., & Steyn, W. The RemoveDebris ADR Mission: Preparing for an International Space Station Launch. In: *7th European Conference on Space Debris*. ESA Space Debris Office. 2017.

Forsythe, G., Malcolm, M., & Moler, C. (1976). *Computer Methods for Mathematical Computations*. Prentice Hall.

Garcia, M. (2021). *Space Debris and Human Spacecraft*. Retrieved January 28, 2022, from https://www.nasa.gov/mission_pages/station/news/orbital_debris.html

Gavrilovich, I., Shea, H., & Richard, M. (2013). *CleanSpace One Gripper Report* (tech. rep.). Swiss Space Center EPFL.

Habets, J. (2015). *Evolving Systems Approach to the Attitude Control of a Large-Space-Debris Removal Spacecraft* (Master's thesis). Delft University of Technology. Delft, The Netherlands.

Haykin, S. (2001). *Kalman Filtering and Neural Networks*. John Wiley & Sons.

Hill, G. (1878). Researches in the Lunar Theory. *American Journal of Mathematics*, *Vol. 1*(No. 1), pp. 5–26.

Innocenti, L., Soares, T., & Delaval, J. ESA Clean Space Initiative. In: *6th European Conference on Space Debris*. ESA. 2013.

Julier, S., & Uhlmann, J. (1997). A New Extension of the Kalman Filter to Nonlinear Systems. *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, *Vol. 3068*, pp. 182–193.

Julier, S., & Uhlmann, J. (2004). Unscented Filtering and Nonlinear Estimation. *Proceedings of the IEEE*, *Vol. 92*(No. 3), pp. 401–422.

Katake, A., Ochoa, J., Zbranek, J., Day, B., Bruccoleri, C., & Goodsell, D. Development and Testing of the StarCam SG100: A Stellar Gyroscope. In: *AIAA Guidance, Navigation and Control Conference Exhibit*. AIAA. 2008.

Kawano, I., Mokuno, M., Kasai, T., & Suzuki, T. Result and Evaluation of Autonomous Rendezvous Docking Experiments of ETS-VII. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. AIAA. 1999.

Kessler, D., & Cour-Palais, B. (1978). Collision frequency of artificial satellites: The creation of a debris belt. *Journal of Geophysical Research*, *Vol. 83*, pp. 2637–2646.

Kuipers, J. (2002). *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton University Press.

Larson, W., & Wertz, J. (1999). *Space Mission Analysis and Design* (3rd). Kluwer Academic Publishers.

Leomanni, M., Quartullo, R., Bianchini, G., Garulli, A., & Giannitrapani, A. (2022). Variable-Horizon Guidance for Autonomous Rendezvous and Docking to a Tumbling Target. *Journal of Guidance, Control and Dynamics*, *Vol. 45*(No. 5), pp. 846–858.

Li, D., & Sun, X. (2006). *Nonlinear Integer Programming*. Springer.

Liberti, L., & Maculan, N. (2006). *Global Optimization: From Theory to Implementation*. Springer.

Liou, J., & Johnson, N. (2008). Instability of the present LEO satellite populations. *Advances in Space Research*, *Vol. 41*(No. 7), pp. 1046–1053.

Liou, J., & Johnson, N. (2009). A sensitivity study of the effectiveness of active debris removal in LEO. *Acta Astronautica*, *Vol. 64*(No. 2-3), pp. 236–243.

Löfberg, J. A toolbox for modeling and optimization in MATLAB. In: *Proceedings of the CACSD Conference*. IEEE. 2004.

Lu, P., & Liu, X. (2013). Autonomous Trajectory Planning for Rendezvous and Proximity Operations by Conic Optimization. *Journal of Guidance, Control and Dynamics*, *Vol. 36*(No. 2), pp. 375–389.

Malyuta, D., Reynolds, T., Szmuk, M., Acikmese, B., & Mesbahi, M. Fast Trajectory Optimisation via Successive Convexification for Spacecraft Rendezvous with Integer Contraints. In: *AIAA Scitech 2020 Forum*. AIAA. 2020.

Malyuta, D., Reynolds, T., Szmuk, M., Lew, T., Bonalli, R., Pavone, M., & Açikmese, B. (2021). Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently.

Mao, Y., Szmuk, M., & Acikmese, B. A Tutorial on Real-time Convex Optimization Based Guidance and Control for Aerospace Applications. In: *2018 Annual American Control Conference (ACC)*. IEEE. 2018.

Marchand, E., Chaumette, F., Chabot, T., Kanani, K., & Pollini, A. RemoveDEBRIS Vision-Based Navigation preliminary results. In: *70th International Astronautical Congress*. IAC. 2019.

Markley, L., & Crassidis, J. (2014). *Fundamentals of Spacecraft Attitude Determination and Control*. Springer.

Matlab. (2022a). *Cubic spline data interpolation*. Retrieved September 13, 2022, from https://nl.mathworks.com/help/matlab/ref/spline.html

Matlab. (2022b). *Find minimum of single-variable function on fixed interval*. Retrieved September 14, 2022, from https://nl.mathworks.com/help/matlab/ref/fminbnd.html

Matlab. (2022c). *Nonlinear State Estimation Using Unscented Kalman Filter and Particle Filter*. Retrieved October 14, 2022, from https://nl.mathworks.com/help/ident/ug/nonlinear-state-estimation-using-unscented-kalman-filter.html

Montenbruck, O., & Gill, E. (2000). *Satellite Orbits: Models, Methods and Applications*. Springer.

Mooij, E. (2021a). On-Board Constrained Optimisation of Final Rendezvous Technical Note # 2: Functional Simulator Design [Internal project document].

Mooij, E. (2021b). On-Board Constrained Optimisation of Final Rendezvous Technical Note # 3: Library Blocks Development and Verification [Internal project document].

Mooij, E., & Chu, Q. IMU/GPS Integrated Navigation System for a Winged Re-entry Vehicle. In: *AIAA Guidance, Navigation, and Control Conference*. AIAA. 2001.

Mulder, J., van Staveren, W., van der Vaart, J., de Weerdt, E., de Visser, C., in 't Veld, A., & Mooij, E. (2013). *Lecture Notes AE3202 Flight Dynamics*. Delft University of Technology.

Narkiewicz, J., Sochacki, M., & Zakrzewski, B. (2020). Generic Model of a Satellite Attitude Control System. *International Journal of Aerospace Engineering*, pp. 1–17.

NASA. (2006). *Overview of the dart mishap investigation results*. Retrieved June 11, 2021, from https://www.nasa.gov/pdf/148072main_DART_mishap_overview.pdf

Peters, S., Fiedler, H., Mai, W., & Forstner, R. Research Issues and Challenges in Autonomous Active Space Debris Removal. In: *64th International Astronautical Congress*. IAC. 2013.

Raju, N. (2021). *Russia's anti-satellite test should lead to a multilateral ban*. Retrieved January 28, 2022, from https://www.sipri.org/commentary/essay/2021/russias-anti-satellite-test-should-lead-multilateral-ban

Schaub, H., & Junkins, J. (1996). Stereographic Orientation Parameters for Attitude Dynamics: A Generalization of the Rodrigues Parameters. *Journal of the Astronautical Sciences*, *Vol. 44*(No. 1), pp. 1–29.

Schaub, H., & Junkins, J. (2003). *Analytical Mechanics of Space Systems* (1st ed.). AIAA.

Schöttle, U. (2000). *Lecture Notes University of Stuttgart: Lage- und Bahnregelung*. University of Stuttgart.

Sellmaier, F., Spurmann, J., & Boge, T. On-Orbit Servicing Missions at DLR / GSOC. In: *61st International Astronautical Congress*. IAC. 2010.

Shuster, M. (1993). A Survey of Attitude Representations. *The Journal of the Astronautical Sciences*, *Vol. 41*(No. 4), pp. 439–517.

Silva, E. A Formulation of the Clohessy-Wiltshire Equations to Include Dynamic Atmospheric Drag. In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. AIAA/AAS. 2008.

Singh, S., Mooij, E., & Gransden, D. Multibody Approach to the Controlled Removal of Large Space Debris with Flexible Appendages. In: *AIAA Scitech 2019 Forum*. AIAA. 2019.

Stuelpnagel, J. (1964). On the Parametrization of the Three-Dimensional Rotation Group. *SIAM Review*, *Vol. 6*(No. 4), pp. 422–430.

Sun, L., & Huo, W. (2015). Robust adaptive relative position tracking and attitude synchronization for spacecraft rendezvous. *Aerospace Science and Technology*, *Vol. 41*, pp. 28–35.

Szauer, G. (2017). *Game Physics Cookbook*. Packt Publishing.

Tsiotras, P. (1996). Stabilisation and Optimality Results for the Attitude Control Problem. *Journal of Guidance, Control and Dynamics*, *Vol. 19*(No. 4), pp. 772–779.

UCS. (2021). *Satellite database*. Retrieved May 24, 2021, from https://www.ucsusa.org/resources/satellite-database

United Nations Office For Outer Space Affairs. *Space debris mitigation guidelines of the committee on the peaceful uses of outer space*. United Nations publication. 2010.

van den IJsel, J., Encarnação, J., Doornbos, E., & Visser, P. (2015). Precise science orbits for the Swarm satellite constellation. *Advances in Space Research*, *Vol. 56*(No. 6), pp. 1042–1055.

Virgili-Llop, J., Zagaris, C., Zappulla II, R., Bradstreet, A., & Romano, M. (2019). A convex-programming-based guidance algorithm to capture a tumbling object on orbit using a spacecraft equipped with a robotic manipulator. *The International Journal of Robotics Research*, *Vol. 38*(No. 1), pp. 40–72.

Vittaldev, V., Mooij, E., & Naeije, M. (2012). Unified State Model theory and application in Astrodynamics. *Celestial Mechanics and Dynamical Astronomy*, *Vol. 112*, pp. 253–282.

Wertz, J. (1978). *Spacecraft Attitude Determination and Control*. Kluwer Academic Publishers.

Wiedemann, C., Flegel, S., Möckel, M., Gelhaus, J., Braun, V., Kebschull, C., Kreisel, J., Metz, M., & Vörsmann, P. Cost estimation of Active Debris Removal. In: *63rd International Astronautical Congress 2012*. IAC. 2012.

Zhu, L., & Wang, S. (2020). Rotating Object Specific Tracking Based on Orbit-Attitude Coordinated Adaptive Control. *Journal of Guidance, Control, and Dynamics*, *Vol. 44*, pp. 266–282.

A

Self-Developed Disturbance Models

The current appendix provides a description of the disturbance models that have been derived for this research. In particular, the equations for the solar radiation pressure disturbance are provided in Section A.1, as well as the equations for the aerodynamic drag disturbance in Section A.2. Note, however, that this appendix is limited to the description of the models that were developed as an extension to the GRADS simulation software. Therefore, the interested reader is referred to (Mooij, 2021a) for more information concerning the models intrinsic to GRADS.

A.1. Solar Radiation Pressure Models

When the surface of a spacecraft is exposed to solar radiation, a small force is generated by the interaction of the incoming solar rays with the surface of the spacecraft. This interaction is caused by photons carrying a finite energy and momentum. When these photons impinge the exposed surface, they are either absorbed or reflected, resulting in a momentum exchange. Consequently, this momentum exchange generates a force on the surface, which is referred to as the solar radiation pressure, P_s. A mathematical description of this quantity is provided in Equation (A.1) and taken from (Curtis, 2013), where Φ represents the solar flux and c the speed of light.

$$P_s = \frac{\Phi}{c} \tag{A.1}$$

When assuming that the exposed surface can either absorb the incoming solar photons, reflect them specularly or reflect them diffusely, the elemental solar radiation pressure forces that result from these actions can be described using Equations (A.2) to (A.4), as provided by Wertz (1978). Note that in these equations, $\hat{\mathbf{r}}_{ss}$ represents a unit vector from the spacecraft to the Sun and $\hat{\mathbf{n}}$ a unit vector along the outward normal of the exposed surface with elemental area dA, all referenced with respect to the Body-Centred-Body-Fixed frame of the body undergoing the disturbance. Furthermore, in these equations, the variables C_a, C_s and C_d represent the absorption coefficient, the specular reflection coefficient and the diffuse reflection coefficient, respectively.

$$d\mathbf{f}_{absorbed} = -P_s C_a \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) \hat{\mathbf{r}}_{ss} \, dA \tag{A.2}$$

$$d\mathbf{f}_{specular} = -2P_s C_s \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right)^2 \hat{\mathbf{n}} \, dA \tag{A.3}$$

$$d\mathbf{f}_{diffuse} = -P_s C_d \left(\left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) \hat{\mathbf{r}}_{ss} + \frac{2}{3} \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) \hat{\mathbf{n}} \right) dA \tag{A.4}$$

Following the aforementioned assumption, where the incoming solar photons are either reflected specularly, reflected diffusely or absorbed, these coefficients are consequently coupled using Equation (A.5).

$$C_a + C_s + C_d = 1 \tag{A.5}$$

Using the equations provided above, the total elemental solar radiation pressure force, $d\mathbf{f}_{total}$, can be described using Equation (A.6), as given by Wertz (1978). Furthermore, the solar radiation pressure moment, $\mathbf{M}_s$, can be defined using Equation (A.7). Note that in this equation, $\mathbf{r}$ represents the position vector from the center of mass of the spacecraft to the elemental area dA in the Body-Centred-Body-Fixed frame of the body undergoing the disturbance.

$$d\mathbf{f}_{total} = -P_s \int (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) \left((1 - C_s)\hat{\mathbf{r}}_{ss} + 2 \left(C_s (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) + \frac{1}{3} C_d \right) \hat{\mathbf{n}} \right) dA \tag{A.6}$$

$$\mathbf{M}_s = \int \mathbf{r} \times d\mathbf{f}_{total} \tag{A.7}$$

These two equations will subsequently be applied to determine the solar radiation pressure forces and moments on a parallelepiped body and a conical body, to model the disturbances on the ClearSpace-1 chaser and VESPA target.

A.1.1. Parallelepiped Body

For a parallelepiped body, the disturbance forces and moments due to solar radiation pressure can be calculated in six parts, following the six exposed surfaces of the body, as can be seen in Figure A.1. For each panel of the parallelepiped body, the resulting solar radiation pressure force is calculated using Equation (A.6). This results in Equations (A.8) to (A.19), where the integration over the elemental area has readily been performed. Note that in these equations, the dot product $(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}})$ needs to be positive for the surface to be exposed to the incoming solar radiation. Otherwise, the surface is in the shadow and thus no solar radiation pressure force can be generated.

- Force on flat surface with normal along the positive X-axis:

$$\mathbf{f}_{+x} = \begin{cases} -P_s (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) \left[\left((1 - C_s)\hat{\mathbf{r}}_{ss} + 2 \left(C_s (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) + \frac{1}{3} C_d \right) \hat{\mathbf{n}} \right) \right] bh & \text{for } (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) > 0 \\ (0,0,0)^T & \text{for } (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) < 0 \end{cases} \tag{A.8}$$

  using:

$$\hat{\mathbf{n}} = (1,0,0)^T \tag{A.9}$$

- Force on flat surface with normal along the negative X-axis:

$$\mathbf{f}_{-x} = \begin{cases} -P_s (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) \left[\left((1 - C_s)\hat{\mathbf{r}}_{ss} + 2 \left(C_s (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) + \frac{1}{3} C_d \right) \hat{\mathbf{n}} \right) \right] bh & \text{for } (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) > 0 \\ (0,0,0)^T & \text{for } (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) < 0 \end{cases} \tag{A.10}$$

  using:

$$\hat{\mathbf{n}} = (-1,0,0)^T \tag{A.11}$$

- Force on flat surface with normal along the positive Y-axis:

$$\mathbf{f}_{+y} = \begin{cases} -P_s (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) \left[\left((1 - C_s)\hat{\mathbf{r}}_{ss} + 2 \left(C_s (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) + \frac{1}{3} C_d \right) \hat{\mathbf{n}} \right) \right] Lh & \text{for } (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) > 0 \\ (0,0,0)^T & \text{for } (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) < 0 \end{cases} \tag{A.12}$$

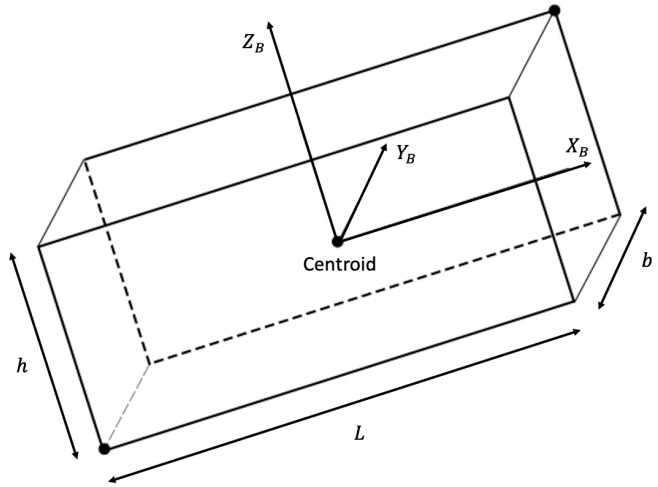


Figure A.1: Parallelepiped body characterised by length L, height h and width b, adapted from (Szauer, 2017)

using:

$$\hat{\mathbf{n}} = (0, 1, 0)^{\mathrm{T}} \tag{A.13}$$

- Force on flat surface with normal along the negative Y-axis:

$$\mathbf{f}_{-y} = \begin{cases} -P_s \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) \left[\left((1 - C_s)\hat{\mathbf{r}}_{ss} + 2 \left(C_s \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) + \frac{1}{3} C_d \right) \hat{\mathbf{n}} \right) \right] Lh & \text{for } \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) > 0 \\ (0, 0, 0)^{\mathrm{T}} & \text{for } \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) < 0 \end{cases} \tag{A.14}$$

using:

$$\hat{\mathbf{n}} = (0, -1, 0)^{\mathrm{T}} \tag{A.15}$$

- Force on flat surface with normal along the positive Z-axis:

$$\mathbf{f}_{+z} = \begin{cases} -P_s \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) \left[\left((1 - C_s)\hat{\mathbf{r}}_{ss} + 2 \left(C_s \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) + \frac{1}{3} C_d \right) \hat{\mathbf{n}} \right) \right] Lb & \text{for } \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) > 0 \\ (0, 0, 0)^{\mathrm{T}} & \text{for } \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) < 0 \end{cases} \tag{A.16}$$

using:

$$\hat{\mathbf{n}} = (0, 0, 1)^{\mathrm{T}} \tag{A.17}$$

- Force on flat surface with normal along the negative Z-axis:

$$\mathbf{f}_{-z} = \begin{cases} -P_s \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) \left[\left((1 - C_s)\hat{\mathbf{r}}_{ss} + 2 \left(C_s \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) + \frac{1}{3} C_d \right) \hat{\mathbf{n}} \right) \right] Lb & \text{for } \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) > 0 \\ (0, 0, 0)^{\mathrm{T}} & \text{for } \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) < 0 \end{cases} \tag{A.18}$$

using:

$$\hat{\mathbf{n}} = (0, 0, -1)^{\mathrm{T}} \tag{A.19}$$

In the equations provided above, $\hat{\mathbf{r}}_{ss}$ is defined by Equation (A.20). Note that in this equation, $\mathbf{C_{B/I}}$ represents the rotation matrix from the Earth-Centred inertial frame to the Body-Centred-Body-Fixed frame and the vectors $\mathbf{r}_{sun}$ and $\mathbf{r}_{sat}$ represent the inertial position vector of the Sun and the parallelepiped body, respectively. A graphical representation of the $\hat{\mathbf{r}}_{ss}$ vector in the Body-Centred-Body-Fixed frame of the parallelepiped body can be seen in Figure A.2.

$$\hat{\mathbf{r}}_{ss} = \mathbf{C_{B/I}} \left(\frac{\mathbf{r}_{sun} - \mathbf{r}_{sat}}{\|\mathbf{r}_{sun} - \mathbf{r}_{sat}\|} \right) \tag{A.20}$$

To obtain the total solar radiation pressure force on the parallelepiped body, a summation of the contributions from the different surfaces is performed, as can be seen in Equation (A.21).

$$\mathbf{f}_{total} = \mathbf{f}_{+x} + \mathbf{f}_{-x} + \mathbf{f}_{+y} + \mathbf{f}_{-y} + \mathbf{f}_{+z} + \mathbf{f}_{-z} \tag{A.21}$$

For each panel of the parallelepiped body, the resulting solar radiation pressure moment can be calculated using Equation (A.7). This results in Equations (A.22) to (A.27). Note, however, that these equations calculate the solar radiation pressure moment around the centroid of the body, as indicated in Figures A.1 and A.2.

- Moment due to flat surface with normal along the positive X-axis:

$$\mathbf{M}_{+x} = \begin{pmatrix} \frac{L}{2} \\ 0 \\ 0 \end{pmatrix} \times \mathbf{f}_{+x} \tag{A.22}$$

- Moment due to flat surface with normal along the negative X-axis:

$$\mathbf{M}_{-x} = \begin{pmatrix} \frac{-L}{2} \\ 0 \\ 0 \end{pmatrix} \times \mathbf{f}_{-x} \tag{A.23}$$

- Moment due to flat surface with normal along the positive Y-axis:

$$\mathbf{M}_{+y} = \begin{pmatrix} 0 \\ \frac{b}{2} \\ 0 \end{pmatrix} \times \mathbf{f}_{+y} \tag{A.24}$$

- Moment due to flat surface with normal along the negative Y-axis:

$$\mathbf{M}_{-y} = \begin{pmatrix} 0 \\ \frac{-b}{2} \\ 0 \end{pmatrix} \times \mathbf{f}_{-y} \tag{A.25}$$

- Moment due to flat surface with normal along the positive Z-axis:

$$\mathbf{M}_{+z} = \begin{pmatrix} 0 \\ 0 \\ \frac{h}{2} \end{pmatrix} \times \mathbf{f}_{+z} \tag{A.26}$$

- Moment due to flat surface with normal along the negative Z-axis:

$$\mathbf{M}_{-z} = \begin{pmatrix} 0 \\ 0 \\ \frac{-h}{2} \end{pmatrix} \times \mathbf{f}_{-z} \tag{A.27}$$

To obtain the total solar radiation pressure moment on the parallelepiped body, a summation of the contributions from the different surfaces is performed, as can be seen in Equation (A.28). Note that, due to the symmetry of the problem, the net perturbing moment equates to zero as the moment components of the different panels cancel each other out.

$$\mathbf{M}_{\text{total}} = \mathbf{M}_{+x} + \mathbf{M}_{-x} + \mathbf{M}_{+y} + \mathbf{M}_{-y} + \mathbf{M}_{+z} + \mathbf{M}_{-z} = 0 \tag{A.28}$$

A.1.2. Conical Body

For a conical body, the disturbance forces and moments due to solar radiation pressure can be calculated in two parts: one part is due to the flat surfaces exposed on the top and bottom of the conical body and the other part is due to the curved wall in between, as can be seen in Figure A.3. For both surfaces, the resulting solar radiation pressure force is calculated using Equation (A.6). For the flat

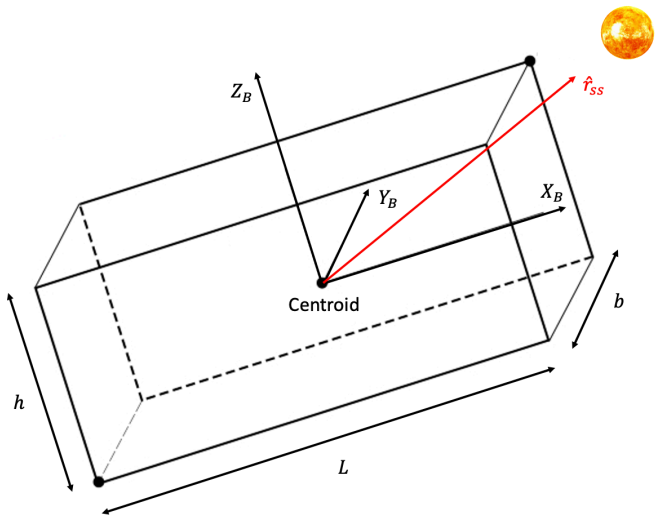


Figure A.2: Definition of $\hat{\mathbf{r}}_{ss}$ vector in parallelepiped Body-Centred-Body-Fixed frame, adapted from (Szauer, 2017)

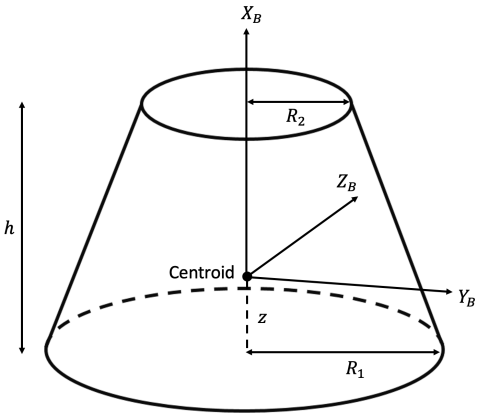


Figure A.3: Conical body characterised by lower radius R_1, upper radius R_2 with $R_2 < R_1$, and height h

surfaces of the cone, however, the disturbance force is calculated following the exact same procedure as given for the parallelepiped body above. This results in Equations (A.29) to (A.32), where the integration over the elemental area has readily been performed. Note again that in these equation, the dot product $(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}})$ needs to be positive for the surface to be exposed to the incoming solar radiation. Furthermore, note that also in these equations, $\hat{\mathbf{r}}_{ss}$ is defined by Equation (A.20).

- Force on upper flat surface with normal along the positive X-axis:

$$\mathbf{f}_{\text{flat}_{+x}} = \begin{cases} -P_s \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) \left[\left((1 - C_s)\hat{\mathbf{r}}_{ss} + 2 \left(C_s \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) + \frac{1}{3} C_d \right) \hat{\mathbf{n}} \right) \right] \pi R_2^2 & \text{for } \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) > 0 \\ (0,0,0)^T & \text{for } \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) < 0 \end{cases} \tag{A.29}$$

  using:

$$\hat{\mathbf{n}} = (1,0,0)^T \tag{A.30}$$

- Force on lower flat surface with normal along the negative X-axis:

$$\mathbf{f}_{\text{flat}_{-x}} = \begin{cases} -P_s \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) \left[\left((1 - C_s)\hat{\mathbf{r}}_{ss} + 2 \left(C_s \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) + \frac{1}{3} C_d \right) \hat{\mathbf{n}} \right) \right] \pi R_1^2 & \text{for } \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) > 0 \\ (0,0,0)^T & \text{for } \left(\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}\right) < 0 \end{cases} \tag{A.31}$$

  using:

$$\hat{\mathbf{n}} = (-1,0,0)^T \tag{A.32}$$

For the curved surface of the cone, however, the integral over the differential area is not as straightforward as for the flat surfaces. Nevertheless, to ease its derivation, two auxiliary angles, α and β are defined to parameterise the direction of the $\hat{\mathbf{r}}_{ss}$ vector in the Body-Centred-Body-Fixed frame, as can be seen in Figure A.4. Using these angles, the $\hat{\mathbf{r}}_{ss}$ vector can be defined as given in Equation (A.33).

$$\hat{\mathbf{r}}_{ss} = C_{B/I} \left(\frac{\mathbf{r}_{sun} - \mathbf{r}_{sat}}{||\mathbf{r}_{sun} - \mathbf{r}_{sat}||} \right) = \begin{pmatrix} \cos\beta \\ \sin\beta\cos\alpha \\ \sin\beta\sin\alpha \end{pmatrix} \tag{A.33}$$

To further ease the derivation, the magnitude of the solar radiation pressure forces are first calculated in the direction of the $\hat{\mathbf{r}}_{ss}$ vector projection in the YZ-plane of the body, $\hat{\mathbf{r}}_{ss_{yz}}$, and along the X-axis of the body, $\hat{\mathbf{r}}_{ss_x}$, as also given in Figure A.4. Subsequently, these calculated forces will be transformed into the Body-Centred-Body-Fixed frame of the truncated cone. To calculate the resulting solar radiation pressure force on the curved surface, the differential area, dA, has to be integrated over the height of the cone as well as over the curvature of the cone, as provided in Figure A.5. Note that in this figure, φ defines the direction of the local surface normal with respect to $\hat{\mathbf{r}}_{ss_{yz}}$ and is defined between -90 and 90 degrees, as generally half of the cone's curved surface will be in the shadow of the other half. This

is always valid apart from two special cases, when the $\hat{\mathbf{r}}_{ss}$ vector is either perfectly aligned with positive X-axis of the body, for which the entire curved surface is illuminated by the Sun, or perfectly aligned with the negative X-axis of the body, for which the entire curved surface is in shadow of the lower flat surface of the cone. These two boundary cases will, however, be treated later. Furthermore, θ defines the slant angle of the cone, for which the trigonometric relations, in Equations (A.34) to (A.36) can be derived.

$$\sin\theta = \frac{R_1 - R_2}{\sqrt{h^2 + (R_1 - R_2)^2}} \tag{A.34}$$

$$\cos\theta = \frac{h}{\sqrt{h^2 + (R_1 - R_2)^2}} \tag{A.35}$$

$$\tan\theta = \frac{R_1 - R_2}{h} \tag{A.36}$$

The components of the $\hat{\mathbf{r}}_{ss}$ and $\hat{\mathbf{n}}$ vectors in the direction of the body X-axis, along the projection in the body YZ-plane and perpendicular to the projection in the body YZ-plane are given in Equations (A.37) and (A.38).

$$\hat{\mathbf{r}}_{ss} = \begin{pmatrix} \cos\beta \\ \sin\beta \\ 0 \end{pmatrix} \tag{A.37}$$

$$\hat{\mathbf{n}} = \begin{pmatrix} \sin\theta \\ \cos\theta\cos\varphi \\ \cos\theta\sin\varphi \end{pmatrix} \tag{A.38}$$

Using these, the solar radiation pressure force on the curved surface in the direction of the $\hat{\mathbf{r}}_{ss}$ projection in the YZ-plane of the body, $\mathbf{f}_{r_{ss_{yz}}}$, and along the X-axis of the body, $\mathbf{f}_{r_{ss_x}}$, can be calculated with Equations (A.39) and (A.41), respectively. Note that the solar radiation pressure force on the curved surface perpendicular to the direction of the $\hat{\mathbf{r}}_{ss}$ projection in the YZ-plane of the body, $\mathbf{f}_{r_{ss_{perp}}}$, as given in Equation (A.40), is per definition zero and therefore does not have to be calculated.

- Force on curved surface along projection in YZ-plane

$$\mathbf{f}_{r_{ss_{yz}}} = -P_s \int\int (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) \left((1 - C_s)\sin\beta + 2\left(C_s (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) + \frac{1}{3} C_d \right) \cos\theta\cos\varphi \right) r\,d\varphi\,d\rho \tag{A.39}$$

- Force on curved surface perpendicular to projection in YZ-plane

$$\mathbf{f}_{r_{ss_{perp}}} = 0 \tag{A.40}$$

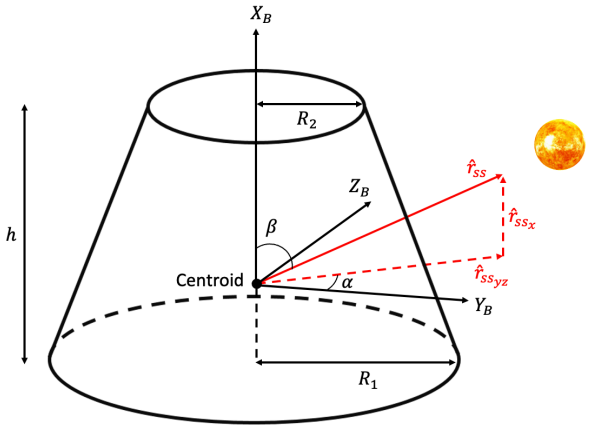


Figure A.4: Definition of $\hat{\mathbf{r}}_{ss}$ vector along body X-axis, $\hat{\mathbf{r}}_{ss_x}$, and projected in the body YZ-plane, $\hat{\mathbf{r}}_{ss_{yz}}$

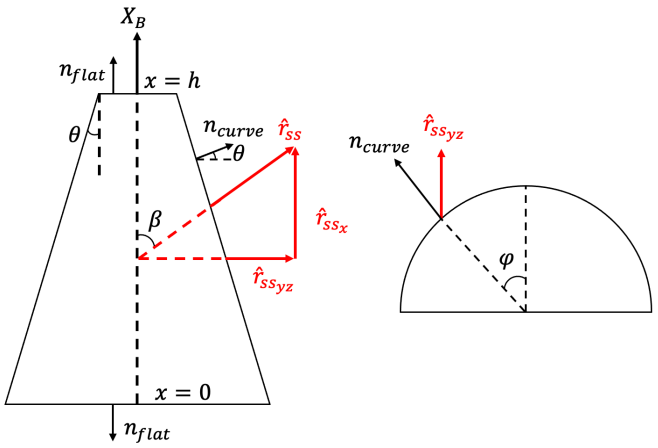


Figure A.5: Cross sections of the conical body defining the angles φ and θ

- Force on curved surface along X-axis

$$\mathbf{f}_{r_{ss_x}} = -P_s \int\int (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) \left((1 - C_s) \cos\beta + 2 \left(C_s (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) + \frac{1}{3} C_d \right) \sin\theta \right) r d\varphi d\rho \tag{A.41}$$

In these equations, the differential area, dA, of the conical surface has been replaced by rdφdρ, with r representing the radius of the cone and ρ the length of the slanted side of the cone (Adams, 1995). To eliminate the additional parameters r and ρ, however, the differential is replaced with an equivalent form as provided by Adams (1995) and given below in Equation (A.42).

$$dA = r d\varphi d\rho = \left(\frac{(R_1 - \tan\theta x)}{\cos\theta} \right) d\varphi dx = \left(R_1 - \left(\frac{R_1 - R_2}{h} \right) x \right) \left(\frac{\sqrt{h^2 + (R_1 - R_2)^2}}{h} \right) d\varphi dx \tag{A.42}$$

Hence the resulting integrals that have to be evaluated are provided in Equations (A.43) and (A.44). Note that in these equations, the vectors $\hat{\mathbf{r}}_{ss}$ and $\hat{\mathbf{n}}$ follow the definitions as provided in Equations (A.37) and (A.38), respectively.

$$\begin{aligned} \mathbf{f}_{r_{ss_{yz}}} = -P_s \int_0^h \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) \left((1 - C_s) \sin\beta + 2 \left(C_s (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) + \frac{1}{3} C_d \right) \cos\theta \cos\varphi \right) \\ \left(R_1 - \left(\frac{R_1 - R_2}{h} \right) x \right) \left(\frac{\sqrt{h^2 + (R_1 - R_2)^2}}{h} \right) d\varphi dx \end{aligned} \tag{A.43}$$

$$\begin{aligned} \mathbf{f}_{r_{ss_x}} = -P_s \int_0^h \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) \left((1 - C_s) \cos\beta + 2 \left(C_s (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) + \frac{1}{3} C_d \right) \sin\theta \right) \\ \left(R_1 - \left(\frac{R_1 - R_2}{h} \right) x \right) \left(\frac{\sqrt{h^2 + (R_1 - R_2)^2}}{h} \right) d\varphi dx \end{aligned} \tag{A.44}$$

Evaluating these integrals results in Equations (A.45) and (A.46), respectively. Note that, for brevity purposes, the intermediate steps have been omitted and only the final results are provided.

$$\mathbf{f}_{r_{ss_{yz}}} = -P_s\left(\frac{R_1+R_2}{2}\right)\left[\frac{2C_s h\left(\frac{4}{3}h^2\sin^2\beta+\pi h(R_1-R_2)\sin\beta\cos\beta+2(R_1-R_2)^2\cos^2\beta\right)}{h^2+(R_1-R_2)^2}\right.$$
$$\left.+\frac{\frac{1}{3}C_d h\left(\pi h\sin\beta+4(R_1-R_2)\cos\beta\right)}{\sqrt{h^2+(R_1-R_2)^2}}+(1-C_s)\sin\beta\left(2h\sin\beta+\pi(R_1-R_2)\cos\beta\right)\right] \tag{A.45}$$

$$\mathbf{f}_{r_{ss_x}} = -P_s\left(\frac{R_1+R_2}{2}\right)\left[\frac{C_s(R_1-R_2)\left(\pi\sin^2\beta h^2+8\sin\beta\cos\beta(R_1-R_2)h+2\pi\cos^2\beta(R_1-R_2)^2\right)}{h^2+(R_1-R_2)^2}\right.$$
$$\left.+\frac{\frac{2}{3}C_d(R_1-R_2)\left(2\sin\beta h+\pi\cos\beta(R_1-R_2)\right)}{\sqrt{h^2+(R_1-R_2)^2}}+(1-C_s)\cos\beta\left(2\sin\beta h+\pi\cos\beta(R_1-R_2)\right)\right] \tag{A.46}$$

Lastly, these force components can be transformed back to the Body-Centred-Body-Fixed frame of the truncated cone using Equations (A.47) to (A.49).

- Force on curved surface in X-direction

$$\mathbf{f}_{curve_x} = \mathbf{f}_{r_{ss_x}} \tag{A.47}$$

- Force on curved surface in Y-direction

$$\mathbf{f}_{curve_y} = \mathbf{f}_{r_{ss_{yz}}}\cos\alpha \tag{A.48}$$

- Force on curved surface in Z-direction

$$\mathbf{f}_{curve_z} = \mathbf{f}_{r_{ss_{yz}}}\sin\alpha \tag{A.49}$$

To obtain the total solar radiation pressure force on the conical body, a summation of the contributions from the different surfaces is performed, as can be seen in Equation (A.50).

$$\mathbf{f}_{total} = \mathbf{f}_{flat_{+x}} + \mathbf{f}_{flat_{-x}} + \mathbf{f}_{curve_x} + \mathbf{f}_{curve_y} + \mathbf{f}_{curve_z} \tag{A.50}$$

When considering the calculation of the solar radiation pressure moment due to the flat and curved surfaces of the conical body, Equation (A.7) is used. For the flat surfaces of the cone, however, the disturbance moment is calculated following the same procedure as provided above for the parallelepiped body. This results in Equations (A.51) and (A.52). Note that also these disturbance moments are calculated around the centroid of the conical body, as indicated in Figures A.3 and A.4.

- Moment due to upper flat surface with normal along the positive X-axis:

$$\mathbf{M}_{flat_{+x}} = \begin{pmatrix} h-z \\ 0 \\ 0 \end{pmatrix} \times \mathbf{f}_{flat_{+x}} \tag{A.51}$$

- Moment due to lower flat surface with normal along the negative X-axis:

$$\mathbf{M}_{flat_{-x}} = \begin{pmatrix} -z \\ 0 \\ 0 \end{pmatrix} \times \mathbf{f}_{flat_{-x}} \tag{A.52}$$

The equations above require the centroid height from the base plane of the cone, given by z. To calculate this parameters, use is made of Equation (A.53), which is obtained from (Beyer, 1987).

$$z = \frac{h\left(R_1^2 + 2R_1R_2 + 3R_2^2\right)}{4\left(R_1^2 + R_1R_2 + R_2^2\right)} \tag{A.53}$$

For the curved surface of the cone, however, the integral over the differential area is not as straightforward as for the flat surfaces, as can be seen in Equations (A.54). Note that this equation is the equivalent representation of Equation (A.7) for a conical body. Therefore, to ease the derivation, the same procedure as given for the disturbance forces on the cone is followed, where the calculation is split along the projection of the $\hat{\mathbf{r}}_{ss}$ vector in the YZ-plane of the body, perpendicular to this projection in the YZ-plane of the body and along the X-axis of the body. This implies that the vectors $\hat{\mathbf{r}}_{ss}$ and $\hat{\mathbf{n}}$ follow the definitions as provided in Equations (A.37) and (A.38), respectively. Furthermore, the position vector from the centroid of the body to the elemental area dA on the curved surface, $\mathbf{r}$, can be calculated with Equation (A.55).

$$\begin{aligned}\mathbf{M}_s = \int\int \mathbf{r} \times \left(-P_s\left(\hat{\mathbf{r}}_{ss}\cdot\hat{\mathbf{n}}\right)\left((1-C_s)\hat{\mathbf{r}}_{ss} + 2\left(C_s\left(\hat{\mathbf{r}}_{ss}\cdot\hat{\mathbf{n}}\right) + \frac{1}{3}C_d\right)\hat{\mathbf{n}}\right)\right)\cdot \\ \left(R_1 - \left(\frac{R_1 - R_2}{h}\right)x\right)\left(\frac{\sqrt{h^2 + (R_1 - R_2)^2}}{h}\right)d\varphi dx\end{aligned} \tag{A.54}$$

$$\mathbf{r} = \begin{bmatrix}\left(R_1 - \tan\theta(x+z)\right)\cos\varphi \\ \left(R_1 - \tan\theta(x+z)\right)\sin\varphi \\ x\end{bmatrix} \quad \text{with} \quad x \in [-z,\ h-z] \tag{A.55}$$

This results in three integral equations giving the solar radiation pressure moment along the projection of the $\hat{\mathbf{r}}_{ss}$ vector in the YZ-plane of the body, $\mathbf{M}_{r_{ss_{yz}}}$, perpendicular to this projection in the YZ-plane of the body, $\mathbf{M}_{r_{ss_{perp}}}$, and along the X-axis of the body, $\mathbf{M}_{r_{ss_x}}$, as given in Equations (A.56) to (A.58), respectively.

- Moment due to curved surface along projection in YZ-plane

$$\begin{aligned}\mathbf{M}_{r_{ss_{yz}}} = -\frac{P_s}{\cos\theta}\int_{-z}^{h-z}\int_{\frac{-\pi}{2}}^{\frac{\pi}{2}}\Big[\left(R_1 - \tan\theta(z+x)\right)\left(F\cos^2\varphi\sin\varphi + G\cos\varphi\sin\varphi + H\sin\varphi\right) \\ -x\left(A\cos^2\varphi\sin\varphi + B\sin\varphi\cos\varphi + D\sin\varphi\right)\Big]\left(R_1 - \tan\theta(x+z)\right)d\varphi dx\end{aligned} \tag{A.56}$$

- Moment due to curved surface perpendicular to projection in YZ-plane

$$\begin{aligned}\mathbf{M}_{r_{ss_{perp}}} = -\frac{P_s}{\cos\theta}\int_{-z}^{h-z}\int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} x\left(A\cos^3\varphi + B\cos^2\varphi + (C+D)\cos\varphi + E\right)\cdot \\ \left(R_1 - \tan\theta(z+x)\right) - \left(R_1 - \tan\theta(z+x)\right)^2\left(F\cos^3\varphi + G\cos^2\varphi + H\cos\varphi\right)d\varphi dx\end{aligned} \tag{A.57}$$

- Moment due to curved surface along X-axis

$$\mathbf{M}_{r_{ss_x}} = -\frac{P_s}{\cos\theta}\int_{-z}^{h-z}\int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} -\left(R_1 - \tan\theta(z+x)\right)^2\left(C\cos\varphi\sin\varphi + E\sin\varphi\right)d\varphi dx \tag{A.58}$$

Note that in the equations above, use is made of the auxiliary variables A to H, which are defined in Equations (A.59) to (A.66).

$$A = 2C_s \cos^3\theta \sin^2\beta \tag{A.59}$$

$$B = 4C_s \sin\theta \cos^2\theta \sin\beta \cos\beta + \frac{2}{3}C_d \cos^2\theta \sin\beta \tag{A.60}$$

$$C = (1 - C_s)\cos\theta \sin^2\beta \tag{A.61}$$

$$D = 2C_s \sin^2\theta \cos\theta \cos^2\beta + \frac{2}{3}C_d \sin\theta \cos\theta \cos\beta \tag{A.62}$$

$$E = (1 - C_s)\sin\theta \sin\beta \cos\beta \tag{A.63}$$

$$F = 2C_s \sin\theta \cos^2\theta \sin^2\beta \tag{A.64}$$

$$G = (1 - C_s)\cos\theta \sin\beta \cos\beta + 4C_s \sin^2\theta \cos\theta \sin\beta \cos\beta + \frac{2}{3}C_d \sin\theta \cos\theta \sin\beta \tag{A.65}$$

$$H = (1 - C_s)\sin\theta \cos^2\beta + 2C_s \sin^3\theta \cos^2\beta + \frac{2}{3}C_d \sin^2\theta \cos\beta \tag{A.66}$$

Evaluating these integrals results in Equations (A.67) to (A.69). Note however that, for brevity purposes, the intermediate steps have been omitted and only the final results are provided.

$$\mathbf{M}_{ss_{yz}} = 0 \tag{A.67}$$

$$\begin{aligned}\mathbf{M}_{r_{ss_{perp}}} =& \frac{-P_s}{\cos\theta}\left(\frac{4}{3}A + \frac{\pi}{2}B + 2(C + D) + \pi E\right)\left(\frac{h}{6}((3h\tan\theta - 6R_1)z - 2h^2\tan\theta + 3R_1 h)\right) \\ &- \left(\frac{4}{3}F + \frac{\pi}{2}G + 2H\right)\left(\frac{h}{3}(h^2\tan^2\theta - 3R_1 h\tan\theta + 3R_1^2)\right)\end{aligned} \tag{A.68}$$

$$\mathbf{M}_{r_{ss_x}} = 0 \tag{A.69}$$

Lastly, these moment components can be transformed back to the Body-Centred-Body-Fixed frame of the truncated cone using Equations (A.70) to (A.72).

- Moment due to curved surface in X-direction

$$\mathbf{M}_{curve_x} = \mathbf{M}_{rss_x} = 0 \tag{A.70}$$

- Moment due to curved surface in Y-direction

$$\mathbf{M}_{curve_y} = -\mathbf{M}_{rss_{perp}} \sin\alpha \tag{A.71}$$

- Moment due to curved surface in Z-direction

$$\mathbf{M}_{curve_z} = \mathbf{M}_{rss_{perp}} \cos\alpha \tag{A.72}$$

To obtain the total solar radiation pressure moment on the conical body, a summation of the contributions from the different surfaces is performed, as can be seen in Equation (A.73).

$$\mathbf{M}_{total} = \mathbf{M}_{flat_{+x}} + \mathbf{M}_{flat_{-x}} + \mathbf{M}_{curve_x} + \mathbf{M}_{curve_y} + \mathbf{M}_{curve_z} \tag{A.73}$$

However, in case the Sun is positioned on the positive X-axis of the conical body frame, *i.e.*, $\hat{\mathbf{r}}_{ss}$ is coincident with normal of the upper flat surface of the cone, both the upper flat surface of the cone and the entire curved surface of the cone are illuminated. The force and moment equations for the

flat surface remain identical to the ones previously provided in Equations (A.29) and (A.51), which is in contrast to the curved surfaces, for which Equations (A.47) to (A.49) and Equations (A.70) to (A.72) are no longer applicable. This is because these equations assume only a partial illumination of the curved surface. Therefore, new equations are derived here, following the same procedure as established above. As a simplification, this special case implies that the β angle equals 0 degrees, which simplifies Equation (A.37) into Equation (A.74).

$$\hat{\mathbf{r}}_{ss} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{A.74}$$

Furthermore, as the entire curved surface is illuminated, the φ angle is defined between -180 and 180 degrees. Applying these adjustments to Equations (A.39) to (A.41) results in Equations (A.75) to (A.77). Note that in these Equations, $\hat{\mathbf{n}}$ remains to be defined by Equation (A.38). Furthermore note that, by definition, $\mathbf{f}_{r_{ss_{yz}}}$ must also be zero as all the component of $\hat{\mathbf{r}}_{ss}$ are along the X-axis of the body.

- Force on curved surface along projection in YZ-plane

$$\mathbf{f}_{r_{ss_{yz}}} = 0 \tag{A.75}$$

- Force on curved surface perpendicular to projection in YZ-plane

$$\mathbf{f}_{r_{ss_{perp}}} = 0 \tag{A.76}$$

- Force on curved surface along X-axis

$$\mathbf{f}_{r_{ss_x}} = -P_s \int\int (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) \left((1 - C_s) + 2 \left(C_s (\hat{\mathbf{r}}_{ss} \cdot \hat{\mathbf{n}}) + \frac{1}{3} C_d \right) \sin\theta \right) r d\varphi d\rho \tag{A.77}$$

Transforming the differential with the procedure outlined in Equation (A.42) results in the integral provided by Equation (A.78).

$$\begin{aligned} \mathbf{f}_{r_{ss_x}} = &- P_s \int_0^h \int_{-\pi}^{\pi} \sin\theta \left((1 - C_s) + 2 \left(C_s \sin\theta + \frac{1}{3} C_d \right) \sin\theta \right) \\ &\left(R_1 - \left(\frac{R_1 - R_2}{h} \right) x \right) \left(\frac{\sqrt{h^2 + (R_1 - R_2)^2}}{h} \right) d\varphi dx \end{aligned} \tag{A.78}$$

Evaluating this integral results in Equation (A.79). Note that for brevity purposes, the intermediate steps have been omitted and only the final result is provided.

$$\mathbf{f}_{r_{ss_x}} = -P_s \left(R_1^2 - R_2^2 \right) \left(1 - C_s + 2 \left(C_s \sin\theta + \frac{1}{3} C_d \right) \sin\theta \right) \pi \tag{A.79}$$

Lastly, these force components can be transformed back to the Body-Centred-Body-Fixed frame of the truncated cone using Equations (A.80) to (A.82).

- Force on curved surface in X-direction

$$\mathbf{f}_{curve_x} = \mathbf{f}_{r_{ss_x}} \tag{A.80}$$

- Force on curved surface in Y-direction

$$\mathbf{f}_{curve_y} = 0 \tag{A.81}$$

- Force on curved surface in Z-direction

$$\mathbf{f}_{curve_z} = 0 \tag{A.82}$$

Table A.1: Solar radiation pressure moment generated on a conical body with the Sun positioned on its $+Y_B$-axis

β [deg]	90	Moment on curved surface [Nm]	$(0, 0, 0.6726)^T \cdot 10^{-6}$
α [deg]	0	Moment on flat surface [Nm]	$(0, 0, 0)^T$

Table A.2: Solar radiation pressure moment generated on a conical body with the Sun positioned on its $-Y_B$-axis

β [deg]	90	Moment on curved surface [Nm]	$(0, 0, -0.6726)^T \cdot 10^{-6}$
α [deg]	180	Moment on flat surface [Nm]	$(0, 0, 0)^T$

Hence the total solar radiation pressure force generated by the conical body is in this special case given by Equation (A.83).

$$\mathbf{f}_{\text{total}} = \mathbf{f}_{\text{flat}_{+x}} + \mathbf{f}_{\text{curve}_x} \tag{A.83}$$

When considering the disturbance moment generation, it can easily be seen from the symmetry of the problem that the resulting moment will be zero. The flat upper surface will not generate a solar radiation pressure moment, given that the line of action of its resulting force acts through the centre of mass of the body. Furthermore, the slanted side of the body will also not generate a resulting moment as the entire curved surface is illuminated and thus the symmetry around the X-axis of the body causes that all components cancel each other out. Therefore the resulting moment generated by the conical body is given by Equation (A.84).

$$\mathbf{M}_{\text{total}} = 0 \tag{A.84}$$

Furthermore, in case the Sun is positioned on the negative X-axis of the conical body frame, *i.e.*, $\hat{\mathbf{r}}_{\text{ss}}$ is coincident with the normal of the lower flat surface of the cone, all surfaces will be in the shadow of this lower flat plane. This implies that those surfaces do not generate disturbance forces or moments, meaning that the total force and moment on the conical body are given by Equations (A.85) and (A.86), respectively.

$$\mathbf{f}_{\text{total}} = \mathbf{f}_{\text{flat}_{-x}} \tag{A.85}$$

$$\mathbf{M}_{\text{total}} = \mathbf{M}_{\text{flat}_{-x}} = 0 \tag{A.86}$$

Note that in all other cases, Equations (A.50) and (A.73) remain valid.

A.1.3. Verification

To complete the verification process of the solar radiation pressure moments acting on a conical body, as provided in Chapter 3, the current section will present the remaining test cases that position the Sun on the ±Y and ±Z-axes of the body. Note that, to conceptualise how the generated radiation forces result in corresponding radiation moments, it must be considered that the centre of pressure of the curved surface is located below the centroid of the body.

When the Sun is located on the positive Y-axis, only the slanted curved surface is illuminated and a moment around the positive Z-axis is generated. The simulation results for this test case are provided in Table A.1 and correspond well with expectations. Conversely, when the Sun is located on the negative Y-axis of the body, also only the slanted curved surface is illuminated but a moment around the negative Z-axis is generated. The simulation results for this test case are provided in Table A.2. When comparing the results provided in Tables A.1 and A.2, it can be identified that the respective moment vectors have an identical magnitude but opposite direction, which follows from the rationale given above.

When the Sun is located on the positive Z-axis of the body, only the slanted curved surface is illuminated and a moment around the negative Y-axis is generated. The simulation results for this test case are provided in Table A.3 and are as expected. Oppositely, when the Sun is located on the negative Z-axis of the body, also only the slanted curved surface is illuminated but a moment around the positive Y-axis is generated. The simulation results for this test case are provided in Table A.4. Note that when comparing the moment components of Table A.3 with the components of Table A.4, it

Table A.3: Solar radiation pressure moment generated on a conical body with the Sun positioned on its $+Z_B$-axis

β [deg]	90	Moment on curved surface [Nm]	$(0, -0.6726, 0)^T \cdot 10^{-6}$
α [deg]	90	Moment on flat surface [Nm]	$(0, 0, 0)^T$

Table A.4: Solar radiation pressure moment generated on a conical body with the Sun positioned on its $-Z_B$-axis

β [deg]	90	Moment on curved surface [Nm]	$(0, +0.6726, 0)^T \cdot 10^{-6}$
α [deg]	−90	Moment on flat surface [Nm]	$(0, 0, 0)^T$

is evident that opposite moments are generated. Hence from this discussion it can be concluded that the direction of the solar radiation pressure moment on a conical body is indeed correct.

A.2. Aerodynamic Drag Models

When a spacecraft is exposed to the upper atmosphere around a planet, the particles present in the air stream will interact with its surface. This interaction is generally modelled as an elastic impact of the particles on the surface, *i.e.*, without reflection, such that all energy will be absorbed (Wertz, 1978). Consequently, a force will be generated on the exposed surface, which is referred to as the aerodynamic pressure, P_a. A mathematical description of this quantity is provided in Equation (A.87) and taken from (Wertz, 1978), where C_D represents the drag coefficient of the body, ρ the atmospheric density and V the magnitude of the spacecraft velocity. Furthermore, $\hat{\mathbf{n}}$ represents a unit vector along the outward normal of the exposed surface and $\hat{\mathbf{v}}$ a unit vector along the velocity vector relative to the atmosphere, both with respect to the Body-Centred-Body-Fixed frame.

$$P_a = -\frac{1}{2} C_D \rho (V)^2 (\hat{\mathbf{n}} \cdot \hat{\mathbf{v}}) \hat{\mathbf{v}} \tag{A.87}$$

From this, the aerodynamic force, $d\mathbf{f}_a$, that results from this pressure distribution on the exposed surface with elemental area dA can be calculated using Equation (A.88).

$$d\mathbf{f}_a = -\frac{1}{2} C_D \rho (V)^2 (\hat{\mathbf{n}} \cdot \hat{\mathbf{v}}) \hat{\mathbf{v}} dA \tag{A.88}$$

With the equations provided above, the total aerodynamic force, $d\mathbf{f}_{total}$, can be described using Equation (A.89), as given by Wertz (1978). Furthermore, the aerodynamic moment, $\mathbf{M}_a$ can be defined using Equation (A.90), where $\mathbf{r}$ represents the position vector from the center of mass of the spacecraft to the elemental area dA in the Body-Centred-Body-Fixed frame of the vehicle undergoing the disturbance.

$$d\mathbf{f}_{total} = -\frac{1}{2} C_D \rho (V)^2 \int (\hat{\mathbf{n}} \cdot \hat{\mathbf{v}}) \hat{\mathbf{v}} dA \tag{A.89}$$

$$\mathbf{M}_a = \int \mathbf{r} \times d\mathbf{f}_{total} \tag{A.90}$$

These last two equations will subsequently be applied to determine the aerodynamic forces and moments on a parallelepiped body and a conical body, to model the disturbances on the ClearSpace-1 chaser and VESPA target.

A.2.1. Parallelepiped Body

Identical to the solar radiation pressure calculations of the parallelepiped body provided above, the disturbance forces and moments due to aerodynamic drag can be calculated in six parts, following the six exposed surfaces of the body, as can be seen in Figure A.1. For each panel of the body, the resulting aerodynamic force is calculated using Equation (A.89). This results in Equations (A.91) to (A.102), where the integration over the elemental area has readily been performed. Note that in these equations, the dot product $(\hat{\mathbf{v}} \cdot \hat{\mathbf{n}})$ needs to be positive for the surface to be exposed to the incoming airflow. Otherwise, the surface is in the shadow and thus no aerodynamic force can be generated.

- Force on flat surface with normal along the positive X-axis:

$$\mathbf{f}_{+x} = \begin{cases} -\frac{1}{2} C_D \rho (V)^2 \hat{\mathbf{v}} (bh) & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) > 0 \\ (0,0,0)^T & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) < 0 \end{cases} \tag{A.91}$$

using:

$$\hat{\mathbf{n}} = (1,0,0)^T \tag{A.92}$$

- Force on flat surface with normal along the negative X-axis:

$$\mathbf{f}_{-x} = \begin{cases} -\frac{1}{2} C_D \rho (V)^2 \hat{\mathbf{v}} (bh) & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) > 0 \\ (0,0,0)^T & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) < 0 \end{cases} \tag{A.93}$$

using:

$$\hat{\mathbf{n}} = (-1,0,0)^T \tag{A.94}$$

- Force on flat surface with normal along the positive Y-axis:

$$\mathbf{f}_{+y} = \begin{cases} -\frac{1}{2} C_D \rho (V)^2 \hat{\mathbf{v}} (Lh) & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) > 0 \\ (0,0,0)^T & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) < 0 \end{cases} \tag{A.95}$$

using:

$$\hat{\mathbf{n}} = (0,1,0)^T \tag{A.96}$$

- Force on flat surface with normal along the negative Y-axis:

$$\mathbf{f}_{-y} = \begin{cases} -\frac{1}{2} C_D \rho (V)^2 \hat{\mathbf{v}} (Lh) & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) > 0 \\ (0,0,0)^T & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) < 0 \end{cases} \tag{A.97}$$

using:

$$\hat{\mathbf{n}} = (0,-1,0)^T \tag{A.98}$$

- Force on flat surface with normal along the positive Z-axis:

$$\mathbf{f}_{+z} = \begin{cases} -\frac{1}{2} C_D \rho (V)^2 \hat{\mathbf{v}} (Lb) & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) > 0 \\ (0,0,0)^T & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) < 0 \end{cases} \tag{A.99}$$

using:

$$\hat{\mathbf{n}} = (0,0,1)^T \tag{A.100}$$

- Force on flat surface with normal along the negative Z-axis:

$$\mathbf{f}_{-z} = \begin{cases} -\frac{1}{2} C_D \rho (V)^2 \hat{\mathbf{v}} (Lb) & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) > 0 \\ (0,0,0)^T & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) < 0 \end{cases} \tag{A.101}$$

using:

$$\hat{\mathbf{n}} = (0,0,-1)^T \tag{A.102}$$

In the equations provided above, $\hat{\mathbf{v}}$ is defined by Equation (A.103), where $\mathbf{V}_{rel}$ represents the velocity vector of the body relative to the incident stream of atmospheric particles and needs to be referenced with respect to the Body-Centred-Body-Fixed frame of the parallelepiped body. A graphical representation of the $\hat{\mathbf{v}}$ vector can be seen in Figure A.6.

$$\hat{\mathbf{v}} = \frac{\mathbf{V}_{rel}}{||\mathbf{V}_{rel}||} \tag{A.103}$$

To obtain the total aerodynamic force on the parallelepiped body, a summation of the contributions from the different surfaces is performed, as can be seen in Equation (A.104).

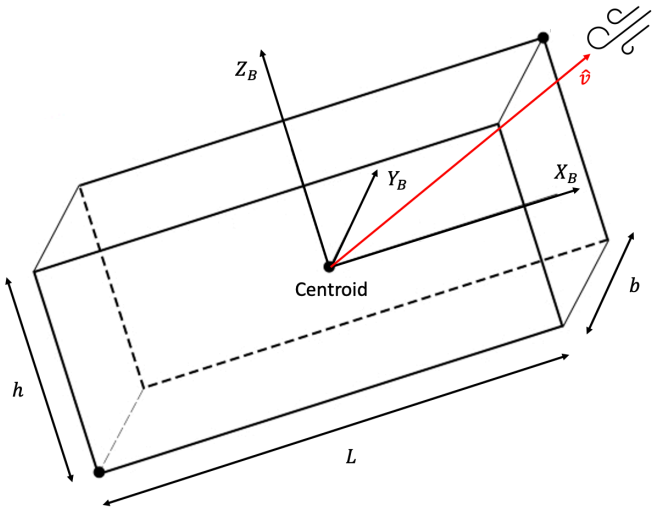


Figure A.6: Definition of $\hat{\mathbf{v}}$ vector in parallelepiped Body-Centred-Body-Fixed frame, adapted from (Szauer, 2017)

$$\mathbf{f}_{\text{total}} = \mathbf{f}_{+\text{x}} + \mathbf{f}_{-\text{x}} + \mathbf{f}_{+\text{y}} + \mathbf{f}_{-\text{y}} + \mathbf{f}_{+\text{z}} + \mathbf{f}_{-\text{z}} \tag{A.104}$$

When considering the calculation of the aerodynamic moment, Equation (A.90) is applied to each panel. This results in Equations (A.105) to (A.110). Note, however, that these equations calculate the aerodynamic moment around the centroid of the body, as indicated in Figure A.6.

- Moment due to flat surface with normal along the positive X-axis:

$$\mathbf{M}_{+\text{x}} = \begin{pmatrix} \frac{\text{L}}{2} \\ 0 \\ 0 \end{pmatrix} \times \mathbf{f}_{+\text{x}} \tag{A.105}$$

- Moment due to flat surface with normal along the negative X-axis:

$$\mathbf{M}_{-\text{x}} = \begin{pmatrix} \frac{-\text{L}}{2} \\ 0 \\ 0 \end{pmatrix} \times \mathbf{f}_{-\text{x}} \tag{A.106}$$

- Moment due to flat surface with normal along the positive Y-axis:

$$\mathbf{M}_{+\text{y}} = \begin{pmatrix} 0 \\ \frac{\text{b}}{2} \\ 0 \end{pmatrix} \times \mathbf{f}_{+\text{y}} \tag{A.107}$$

- Moment due to flat surface with normal along the negative Y-axis:

$$\mathbf{M}_{-\text{y}} = \begin{pmatrix} 0 \\ \frac{-\text{b}}{2} \\ 0 \end{pmatrix} \times \mathbf{f}_{-\text{y}} \tag{A.108}$$

- Moment due to flat surface with normal along the positive Z-axis:

$$\mathbf{M}_{+\text{z}} = \begin{pmatrix} 0 \\ 0 \\ \frac{\text{h}}{2} \end{pmatrix} \times \mathbf{f}_{+\text{z}} \tag{A.109}$$

- Moment due to flat surface with normal along the negative Z-axis:

$$\mathbf{M}_{-z} = \begin{pmatrix} 0 \\ 0 \\ \frac{-h}{2} \end{pmatrix} \times \mathbf{f}_{-z} \tag{A.110}$$

To obtain the total solar radiation pressure moment on the parallelepiped body, a summation of the contributions from the different surfaces is performed, as can be seen in Equation (A.111). Note that, due to the symmetry of the problem, the net perturbing moment equates to zero as the moment components of the different panels cancel each other out.

$$\mathbf{M}_{\text{total}} = \mathbf{M}_{+x} + \mathbf{M}_{-x} + \mathbf{M}_{+y} + \mathbf{M}_{-y} + \mathbf{M}_{+z} + \mathbf{M}_{-z} = 0 \tag{A.111}$$

A.2.2. Conical Body

For a conical body, the disturbance forces and moments are once again calculated in two parts: one part is due to the flat surfaces exposed on the top and bottom of the body and the other part is due to the curved wall in between. For both the flat and curved surfaces, the resulting aerodynamic force is calculated using Equation (A.89). However, for the flat surfaces, the disturbance force can be calculated using the procedure given for the parallelepiped body above. This results in Equations (A.112) to (A.115), where the integration over the elemental area has readily been performed. Note again that in these equation, the dot product $(\hat{\mathbf{v}} \cdot \hat{\mathbf{n}})$ needs to be positive for the surface to be exposed to the incoming stream of aerodynamic particles. Furthermore, note that also in these equations, $\hat{\mathbf{v}}$ is defined by Equation (A.103).

- Force on upper flat surface with normal along the positive X-axis:

$$\mathbf{f}_{\text{flat}_{+x}} = \begin{cases} -\frac{1}{2} C_D \rho (V)^2 \hat{\mathbf{v}} (\pi R_2^2) & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) > 0 \\ (0,0,0)^T & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) < 0 \end{cases} \tag{A.112}$$

  using:

$$\hat{\mathbf{n}} = (1,0,0)^T \tag{A.113}$$

- Force on lower flat surface with normal along the negative X-axis:

$$\mathbf{f}_{\text{flat}_{-x}} = \begin{cases} -\frac{1}{2} C_D \rho (V)^2 \hat{\mathbf{v}} (\pi R_1^2) & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) > 0 \\ (0,0,0)^T & \text{for } (\hat{\mathbf{v}} \cdot \hat{\mathbf{n}}) < 0 \end{cases} \tag{A.114}$$

  using:

$$\hat{\mathbf{n}} = (-1,0,0)^T \tag{A.115}$$

For the curved surface of the cone, however, the integral over the differential area is not as straightforward as for the flat surfaces. Nevertheless, to ease its derivation, two auxiliary angles, α and β are defined in a similar way to the solar radiation pressure disturbance, as can be seen in Figure A.7. Using these angles, the $\hat{\mathbf{v}}$ vector can be defined as given in Equation (A.116).

$$\hat{\mathbf{v}} = \frac{\mathbf{V}_{\text{rel}}}{||\mathbf{V}_{\text{rel}}||} = \begin{pmatrix} \cos\beta \\ \sin\beta\cos\alpha \\ \sin\beta\sin\alpha \end{pmatrix} \tag{A.116}$$

Following the same reasoning as for the solar radiation pressure disturbance on a conical body, the magnitude of the aerodynamic force is first calculated in the direction of the $\hat{\mathbf{v}}$ vector projection in the YZ-plane of the body, $\hat{\mathbf{v}}_{yz}$, and along the X-axis of the body, $\hat{\mathbf{v}}_x$, as presented in Figure A.7.

Subsequently, these calculated forces will be transformed into the Body-Centred-Body-Fixed frame of the truncated cone. To calculate the resulting aerodynamic force on the curved surface, the differential area, dA, has to be integrated over the height, as well as over the curvature of the cone, as provided in Figure A.8. Note that, φ defines the direction of the local surface normal with respect to $\hat{\mathbf{v}}_{yz}$ and is defined between -90 and 90 degrees, as generally half of the cone's curved surface will be in the shadow

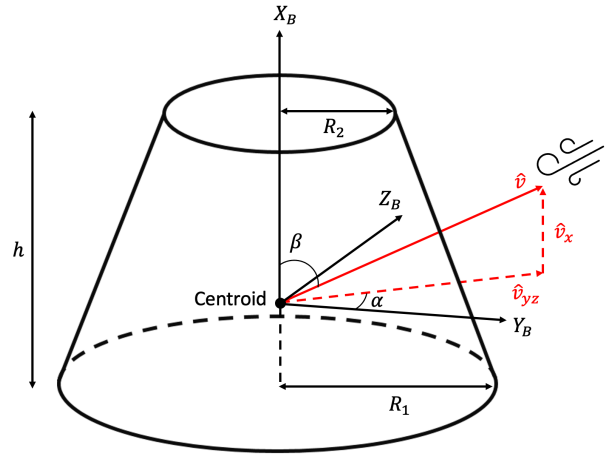


Figure A.7: Definition of $\hat{\mathbf{v}}$ vector along body X-axis, $\hat{\mathbf{v}}_x$, and projected in the body YZ-plane, $\hat{\mathbf{v}}_{yz}$

of the other half. This is always valid apart from two special cases, when the $\hat{\mathbf{v}}$ vector is either perfectly aligned with positive X-axis of the body, for which the entire curved surface encounters the incoming atmospheric flow, or perfectly aligned with the negative X-axis of the body, for which the entire curved surface is in shadow of the lower flat surface of the cone. These two boundary cases will, however, be treated later. Furthermore, θ defines the slant angle of the cone, for which the trigonometric relations, as previously provided in Equations (A.34) to (A.36) were derived. Lastly, the components of the $\hat{\mathbf{v}}$ and $\hat{\mathbf{n}}$ vectors in the direction of the body X-axis, along the projection of $\hat{\mathbf{v}}$ in the body YZ-plane and perpendicular to the projection of $\hat{\mathbf{v}}$ in the body YZ-plane are given in Equations (A.117) and (A.118).

$$\hat{\mathbf{v}} = \begin{pmatrix} \cos\beta \\ \sin\beta \\ 0 \end{pmatrix} \tag{A.117}$$

$$\hat{\mathbf{n}} = \begin{pmatrix} \sin\theta \\ \cos\theta\cos\varphi \\ \cos\theta\sin\varphi \end{pmatrix} \tag{A.118}$$

Using these, the aerodynamic force on the curved surface in the direction of the $\hat{\mathbf{v}}$ projection in the YZ-plane of the body, $\mathbf{f}_{v_{yz}}$, and along the X-axis of the body, $\mathbf{f}_{v_x}$, can be calculated with Equations (A.119) and (A.121), respectively. Note again that the aerodynamic force on the curved surface perpendicular to the direction of the $\hat{\mathbf{v}}$ projection in the YZ-plane of the body, $\mathbf{f}_{v_{perp}}$, as given in Equation (A.120), is per definition zero.

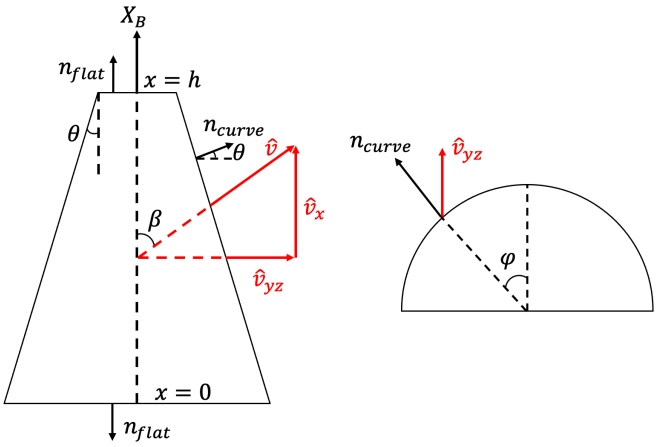


Figure A.8: Cross sections of the conical body defining the angles φ and θ

- Force on curved surface along projection in YZ-plane

$$\mathbf{f}_{v_{yz}} = -\frac{1}{2} C_D \rho (V)^2 \int \int \sin\beta \, (\hat{\mathbf{n}} \cdot \hat{\mathbf{v}}) \, r d\varphi d\rho \tag{A.119}$$

- Force on curved surface perpendicular to projection in YZ-plane

$$\mathbf{f}_{v_{perp}} = 0 \tag{A.120}$$

- Force on curved surface along X-axis

$$\mathbf{f}_{v_x} = -\frac{1}{2} C_D \rho (V)^2 \int \int \cos\beta \, (\hat{\mathbf{n}} \cdot \hat{\mathbf{v}}) \, r d\varphi d\rho \tag{A.121}$$

Note again that in these equations, the differential area of the conical surface has been replaced by $r d\varphi d\rho$, with r representing the radius of the cone and ρ the length of the slanted side of the cone (Adams, 1995). To eliminate these additional parameters, the differential is transformed into an equivalent form, as provided by Adams (1995), and given above in Equation (A.42). Hence the resulting integrals that have to be evaluated are provided in Equations (A.122) and (A.123). Note that in these equations, the vectors $\hat{\mathbf{v}}$ and $\hat{\mathbf{n}}$ follow the definitions as provided in Equations (A.117) and (A.118), respectively.

$$\mathbf{f}_{v_{yz}} = -\frac{1}{2} C_D \rho (V)^2 \int_0^h \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} \sin\beta \, (\hat{\mathbf{n}} \cdot \hat{\mathbf{v}}) \left(R_1 - \left(\frac{R_1 - R_2}{h} \right) x \right) \left(\frac{\sqrt{h^2 + (R_1 - R_2)^2}}{h} \right) d\varphi dx \tag{A.122}$$

$$\mathbf{f}_{v_x} = -\frac{1}{2} C_D \rho (V)^2 \int_0^h \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} \cos\beta \, (\hat{\mathbf{n}} \cdot \hat{\mathbf{v}}) \left(R_1 - \left(\frac{R_1 - R_2}{h} \right) x \right) \left(\frac{\sqrt{h^2 + (R_1 - R_2)^2}}{h} \right) d\varphi dx \tag{A.123}$$

Evaluating these integrals results in Equations (A.124) and (A.125). Note that for brevity purposes, the intermediate steps have been omitted and only the final results are provided.

$$\mathbf{f}_{v_{yz}} = -\frac{1}{2} C_D \rho (V)^2 \left(2h \sin^2\beta + \pi (R_1 - R_2) \sin\beta \cos\beta \right) \left(\frac{R_1 + R_2}{2} \right) \tag{A.124}$$

$$\mathbf{f}_{v_x} = -\frac{1}{2} C_D \rho (V)^2 \left(2h \sin\beta \cos\beta + \pi (R_1 - R_2) \cos^2\beta \right) \left(\frac{R_1 + R_2}{2} \right) \tag{A.125}$$

Lastly, these force components can be transformed back to the Body-Centred-Body-Fixed frame of the truncated cone using Equations (A.126) to (A.128).

- Force on curved surface in X-direction

$$\mathbf{f}_{curve_x} = \mathbf{f}_{v_x} \tag{A.126}$$

- Force on curved surface in Y-direction

$$\mathbf{f}_{curve_y} = \mathbf{f}_{v_{yz}} \cos\alpha \tag{A.127}$$

- Force on curved surface in Z-direction

$$\mathbf{f}_{curve_z} = \mathbf{f}_{v_{yz}} \sin\alpha \tag{A.128}$$

To obtain the total aerodynamic force on the conical body, a summation of the contributions from the different surfaces is performed, as can be seen in Equation (A.129).

$$\mathbf{f}_{total} = \mathbf{f}_{flat_{+x}} + \mathbf{f}_{flat_{-x}} + \mathbf{f}_{curve_x} + \mathbf{f}_{curve_y} + \mathbf{f}_{curve_z} \tag{A.129}$$

When considering the calculation of the aerodynamic moment due to the flat and curved surfaces of the conical body, Equation (A.90) is used. For the flat surfaces of the cone, however, the disturbance moment is calculated following the same procedure as provided for the parallelepiped body. This results in Equations (A.130) and (A.131). Note that also these disturbance moments are calculated around the centroid of the conical body, as indicated in Figure A.7. Furthermore note that these equations require the centroid height from the base plane of the cone, given by z. To calculate this parameters, use is made of Equation (A.53).

- Moment due to upper flat surface with normal along the positive X-axis:

$$\mathbf{M}_{\text{flat}_{+x}} = \begin{pmatrix} h - z \\ 0 \\ 0 \end{pmatrix} \times \mathbf{f}_{\text{flat}_{+x}} \tag{A.130}$$

- Moment due to lower flat surface with normal along the negative X-axis:

$$\mathbf{M}_{\text{flat}_{-x}} = \begin{pmatrix} -z \\ 0 \\ 0 \end{pmatrix} \times \mathbf{f}_{\text{flat}_{-x}} \tag{A.131}$$

For the curved surface of the cone, however, the integral over the differential area is not as straightforward as for the flat surfaces, as can be seen in Equations (A.132). Note that this equation is the equivalent representation of Equation (A.90) for a conical body. Therefore, to ease the derivation, the same procedure as given for the disturbance forces on the cone is followed, where the calculation is split along the projection of the $\hat{\mathbf{v}}$ vector in the YZ-plane of the body, perpendicular to this projection in the YZ-plane of the body and along the X-axis of the body. This implies that the vectors $\hat{\mathbf{v}}$ and $\hat{\mathbf{n}}$ follow the definitions as provided in Equations (A.117) and (A.118), respectively. Furthermore, the position vector from the centroid of the body to the elemental area dA on the curved surface, $\mathbf{r}$, is only dependent on geometry and can be calculated with Equation (A.55).

$$\mathbf{M}_a = \int\int \mathbf{r} \times \left(-\frac{1}{2} C_D \rho (V)^2 \left(\hat{\mathbf{n}} \cdot \hat{\mathbf{v}}\right) \hat{\mathbf{v}} \right) \left(R_1 - \left(\frac{R_1 - R_2}{h} \right) x \right) \left(\frac{\sqrt{h^2 + (R_1 - R_2)^2}}{h} \right) d\varphi dx \tag{A.132}$$

This results in three integral equations giving the aerodynamic moments along the projection of the $\hat{\mathbf{v}}$ vector in the YZ-plane of the body, $\mathbf{M}_{v_{yz}}$, perpendicular to this projection in the YZ-plane of the body, $\mathbf{M}_{v_{perp}}$, and along the X-axis of the body, $\mathbf{M}_{v_x}$, as given in Equations (A.133) to (A.135), respectively.

- Moment due to curved surface along projection in YZ-plane

$$\mathbf{M}_{v_{yz}} = -\frac{C_D \rho (V)^2}{2 \cos\theta} \int_{-z}^{h-z} \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} \left(R_1 - \tan\theta (z + x)\right)^2 \cos\beta \sin\varphi \left(\sin\beta \cos\theta \cos\varphi + \cos\beta \sin\theta\right) d\varphi dx \tag{A.133}$$

- Moment due to curved surface perpendicular to projection in YZ-plane

$$\begin{aligned} \mathbf{M}_{v_{perp}} = -\frac{C_D \rho (V)^2}{2 \cos\theta} \int_{-z}^{h-z} \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} & x \left(\sin^2\beta \cos\theta \cos\varphi + \sin\beta \cos\beta \sin\theta\right) \left(R_1 - \tan\theta (z + x)\right) \\ & - \left(R_1 - \tan\theta (z + x)\right)^2 \left(\sin\beta \cos\beta \cos\theta \cos^2\varphi + \cos^2\beta \sin\theta \cos\varphi\right) d\varphi dx \end{aligned} \tag{A.134}$$

- Moment due to curved surface along X-axis

$$\mathbf{M}_{v_x} = \frac{C_D \rho (V)^2}{2 \cos\theta} \int_{-z}^{h-z} \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} \left(R_1 - \tan\theta (z + x)\right)^2 \left(\sin^2\beta \cos\theta \sin\varphi + \sin\beta \cos\beta \sin\theta \sin\varphi\right) d\varphi dx \tag{A.135}$$

Evaluating these integrals results in Equations (A.136) to (A.138). Note however that, for brevity purposes, the intermediate steps have been omitted and only the final results are provided.

$$\mathbf{M}_{v_{yz}} = 0 \tag{A.136}$$

$$\begin{aligned}\mathbf{M}_{v_{perp}} = &- \frac{C_D \rho (V)^2}{2\cos\theta}\left(\frac{h}{6}\left((3h\tan\theta - 6R_1)z - 2h^2\tan\theta + 3R_1 h\right)\right)\left(2\sin^2\beta\cos\theta + \pi\sin\beta\cos\beta\sin\theta\right) \\ &- \left(\frac{h}{3}(h^2\tan^2\theta - 3R_1 h\tan\theta + 3R_1^2)\right)\left(\frac{\pi}{2}\sin\beta\cos\beta\cos\theta + 2\cos^2\beta\sin\theta\right)\end{aligned} \tag{A.137}$$

$$\mathbf{M}_{v_x} = 0 \tag{A.138}$$

Lastly, these moment components can be transformed back to the Body-Centred-Body-Fixed frame of the truncated cone using Equations (A.139) to (A.141).

- Moment due to curved surface in X-direction

$$\mathbf{M}_{curve_x} = \mathbf{M}_{v_x} = 0 \tag{A.139}$$

- Moment due to curved surface in Y-direction

$$\mathbf{M}_{curve_y} = -\mathbf{M}_{v_{perp}} \sin\alpha \tag{A.140}$$

- Moment due to curved surface in Z-direction

$$\mathbf{M}_{curve_z} = \mathbf{M}_{v_{perp}} \cos\alpha \tag{A.141}$$

To obtain the total aerodynamic moment on the conical body, a summation of the contributions from the different surfaces is performed, as can be seen in Equation (A.142).

$$\mathbf{M}_{total} = \mathbf{M}_{flat_{+x}} + \mathbf{M}_{flat_{-x}} + \mathbf{M}_{curve_x} + \mathbf{M}_{curve_y} + \mathbf{M}_{curve_z} \tag{A.142}$$

However, in case the cone orients the upper surface directly into the incoming stream of aerodynamic particles, *i.e.*, $\hat{\mathbf{v}}$ is coincident with normal of the upper flat surface, both this surface and the entire curved surface of the cone will be impinged by the incoming flow. The force and moment equations for the flat surface remain identical to the ones previously provided in Equations (A.112) and (A.130). This is in contrast to the curved surfaces, for which Equations (A.126) to (A.128) and Equations (A.139) to (A.141) are no longer applicable, as these equations assume only a partial impingement. Therefore, new equations are derived here, following the same procedure as established above. As a simplification, this special case implies that the β angle equals 0 degrees, which simplifies Equation (A.117) into Equation (A.143).

$$\hat{\mathbf{v}} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{A.143}$$

Furthermore, as the entire curved surface is impinged, the φ angle is defined between -180 and 180 degrees. Applying these adjustments to Equations (A.119) to (A.121) results in Equations (A.144) to (A.146). Note that in these Equations, $\hat{\mathbf{n}}$ remains to be defined by Equation (A.118). Furthermore note that, by definition, $\mathbf{f}_{v_{yz}}$ must also be zero as all the components of $\hat{\mathbf{v}}$ are only along the X-axis of the body.

- Force on curved surface along projection in YZ-plane

$$\mathbf{f}_{v_{yz}} = 0 \tag{A.144}$$

- Force on curved surface perpendicular to projection in YZ-plane

$$\mathbf{f}_{v_{perp}} = 0 \tag{A.145}$$

- Force on curved surface along X-axis

$$\mathbf{f}_{v_x} = -\frac{1}{2} C_D \rho (V)^2 \int \int \sin\theta r d\varphi d\rho \tag{A.146}$$

Transforming the differential with the procedure of Equation (A.42) results in the integral provided by Equation (A.147).

$$\mathbf{f}_{v_x} = -\frac{1}{2} C_D \rho (V)^2 \int_0^h \int_{-\pi}^{\pi} \sin\theta \left(R_1 - \left(\frac{R_1 - R_2}{h} \right) x \right) \left(\frac{\sqrt{h^2 + (R_1 - R_2)^2}}{h} \right) d\varphi dx \tag{A.147}$$

Evaluating this integral results in Equation (A.148). Note that for brevity purposes, the intermediate steps have been omitted and only the final result is provided.

$$\mathbf{f}_{v_x} = -\frac{1}{2} C_D \rho (V)^2 \left(R_1^2 - R_2^2 \right) \tag{A.148}$$

Lastly, these force components can be transformed back to the Body-Centred-Body-Fixed frame of the truncated cone using Equations (A.149) to (A.151).

- Force on curved surface in X-direction

$$\mathbf{f}_{curve_x} = \mathbf{f}_{v_x} \tag{A.149}$$

- Force on curved surface in Y-direction

$$\mathbf{f}_{curve_y} = 0 \tag{A.150}$$

- Force on curved surface in Z-direction

$$\mathbf{f}_{curve_z} = 0 \tag{A.151}$$

Hence the total atmospheric drag force generated by the conical body in this special case is given by Equation (A.152).

$$\mathbf{f}_{total} = \mathbf{f}_{flat_{+x}} + \mathbf{f}_{curve_x} \tag{A.152}$$

When considering the disturbance moment generation during this special case, it can easily be seen from symmetry that the resulting moment will be zero. The flat upper surface will not generate an atmospheric drag moment, given that the line of action of its resulting force acts through the centre of mass of the body. Furthermore, the slanted side of the body will also not generate a moment as the entire curved surface is impinged and thus the symmetry around the X-axis of the body causes all components to cancel each other out. Therefore, the resulting moment is given by Equation (A.153).

$$\mathbf{M}_{total} = 0 \tag{A.153}$$

Furthermore, in case the cone orients the bottom surface directly into the incoming stream of aerodynamic particles, *i.e.*, $\hat{\mathbf{v}}$ is coincident with the normal of this surface, all other surfaces will be in its shadow. Consequently, those surfaces do not generate disturbance forces or moments, meaning that the total force and moment on the conical body are given by Equations (A.154) and (A.155), respectively.

$$\mathbf{f}_{total} = \mathbf{f}_{flat_{-x}} \tag{A.154}$$

$$\mathbf{M}_{total} = \mathbf{M}_{flat_{-x}} = 0 \tag{A.155}$$

Note that in all other cases, Equations (A.129) and (A.142) remain valid.

Table A.5: Atmospheric drag moment generated on a conical body with the velocity vector relative to the atmosphere positioned on its $+Y_B$-axis

β [deg]	90	Moment on curved surface [Nm]	$(0, 0, -0.2296)^T \cdot 10^{-7}$
α [deg]	0	Moment on flat surface [Nm]	$(0, 0, 0)^T$

Table A.6: Atmospheric drag moment generated on a conical body with the velocity vector relative to the atmosphere positioned on its $-Y_B$-axis

β [deg]	90	Moment on curved surface [Nm]	$(0, 0, 0.2296)^T \cdot 10^{-7}$
α [deg]	180	Moment on flat surface [Nm]	$(0, 0, 0)^T$

Table A.7: Atmospheric drag moment generated on a conical body with the velocity vector relative to the atmosphere positioned on its $+Z_B$-axis

β [deg]	90	Moment on curved surface [Nm]	$(0, 0.2296, 0)^T \cdot 10^{-7}$
α [deg]	90	Moment on flat surface [Nm]	$(0, 0, 0)^T$

Table A.8: Atmospheric drag moment generated on a conical body with the velocity vector relative to the atmosphere positioned on its $-Z_B$-axis

β [deg]	90	Moment on curved surface [Nm]	$(0, -0.2296, 0)^T \cdot 10^{-7}$
α [deg]	−90	Moment on flat surface [Nm]	$(0, 0, 0)^T$

A.2.3. Verification

To complete the verification process of the atmospheric drag moments acting on a conical body, as provided in Chapter 3, the current section will present the remaining test cases that position the relative velocity vector on the ±Y and ±Z-axes of the body. Note that, to conceptualise how the generated drag forces result in a corresponding moment, it must be considered that the centre of pressure of the curved surface is located above the centroid of the body. This is in contrast to the centre of pressure location for the solar radiation pressure tests, and is caused by the difference in generated pressure distributions. Although the verification procedures manage to position both disturbances exactly on top of the body axes, an inverse position of the centre of pressure with respect to the centroid results. This is because the drag disturbance gets fully absorbed, whereas the solar radiation disturbance gets partly reflected. When defining the specular and diffuse reflection coefficients of the solar radiation model as 0, the disturbance also gets fully absorbed and an equivalent pressure distribution to the aerodynamic drag model can be obtained. Consequently, the moment components of Tables A.1 to A.4 would have the same sign as those in Tables A.5 to A.8. However, to make the simulations more realistic, non-zero reflections coefficients were used during the verification tests provided in Tables A.1 to A.4. Therefore, it is not unexpected that their results do not correspond with Tables A.5 to A.8.

When the velocity vector relative to the atmosphere is located on the positive Y-axis of the body, only the slanted surface is impinged and a moment is generated around the negative Z-axis. The simulation results for this test case are provided in Table A.5 and correspond well with the expectations. Conversely, when the velocity vector is located on the negative Y-axis of the body, only a moment around the positive Z-axis is generated and the simulation results for this test case are provided in Table A.6. When comparing, the results provided in Tables A.5 and A.6, it can be observed that the respective moment vectors have an identical magnitude but opposite direction, which follows from the orientations that have been provided for each test case.

Lastly, when the relative velocity vector is along the positive Z-axis of the body, only the slanted surface is impinged and a moment is generated around the positive Y-axis. The simulation results for this test case are provided in Table A.7 and correspond well with the expectations. Alternatively, when the velocity vector is along the negative Z-axis of the body, an opposite moment around the Y-axis is generated, as can be seen from the simulation results in Table A.8. Hence from this discussion it can be concluded that the direction of the atmospheric moment on a conical body is indeed correct.

B

Rotational State Representation for Attitude Control

The current appendix is focused on providing the analysis that led to changing the rotational state in the motion simulator from MRP representation to quaternion representation. To start, Section B.1 provides the state representation in terms of Modified Rodrigues Parameter, after which Section B.2 provides a short comparison between the characteristics of this state representation and the quaternion representation. Subsequently, Section B.3 describes the INDI controller that is part of the attitude controller, equivalent to the one described in Chapter 6, in terms of MRP elements as it was initially envisaged for the considered reference mission. Note that this MRP-based INDI controller would have been combined with the same PID rate controller as given in Chapter 6, but details related to this matter are currently omitted. This is because the current appendix solely focuses on the INDI aspect of the controller design. Lastly, Section B.4 provides a performance analysis of the MRP-based attitude controller and provides the motivation as to why the change to a quaternion-based attitude controller, and consequently a quaternion-based motion simulator, was performed.

B.1. MRP Representation

Modified Rodrigues Parameter represent a modification to the classical Rodrigues parameters, which are defined as a projection of the four dimensional unit quaternion sphere onto a 3-dimensional hyperplane (Vittaldev et al., 2012). This projection thus effectively encodes the orientation of the Euler eigenaxis and the magnitude of the rotation angle around this eigenaxis in a three parameter representation. However, as the MRP vector consist of only three parameters, this attitude representation must contain a singularity (Stuelpnagel, 1964). For MRPs, this singularity occurs when the rotation angle approaches ±360 degrees. However, to avoid this singularity, a set of Shadow Modified Rodrigues Parameters (SMRP) are used that express the same orientation, but for which the singularity occurs when the rotation angle approaches either 0 or 720 degrees. By switching between the MRP and SMRP set, the singularity can be avoided using a three parameter characterisation. A definition of the MRP vector, $\boldsymbol{\sigma}$, can be obtained using the Euler eigenaxis, $\hat{\mathbf{e}}$, and the rotation angle, Φ, over this Euler eigenaxis, as given by Equation (B.1). Note that in this equation, σ_1, σ_2 and σ_3 represent the three components of the MRP vector, $\boldsymbol{\sigma}$.

$$\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \sigma_3)^T = \hat{\mathbf{e}} \tan(\frac{\Phi}{4}) \tag{B.1}$$

It can be seen that Equation (B.1) indeed identifies a singularity at $\Phi = \pm 360$ degrees. To avoid this singularity, the SMRP set is created. To convert between the SMRP vector, $\boldsymbol{\sigma}^S$, and MRP vector, $\boldsymbol{\sigma}$, Equation (B.2) can be used, where σ represents the norm of $\boldsymbol{\sigma}$. Note that both the MRP vector and the SMRP vector describe the same attitude. Nevertheless, a switch is generally performed when the norm of the current set exceeds 1, which corresponds to a principal rotation angle of ±180 degrees. Thereby, it is ensured that both sets always remain far away from their respective singularities (Schaub and Junkins, 1996).

$$\boldsymbol{\sigma}^S = \frac{-\boldsymbol{\sigma}}{\sigma^2} \tag{B.2}$$

The corresponding rotational state of the vehicle in terms of Modified Rodrigues Parameters, $\mathbf{x}_{rot}$, is provided in Equation (B.3). In this equation, the same angular velocity vector definition is used as

for the rotational state vector based on attitude quaternions, where ω_1, ω_2 and ω_3 represent the three angular velocity components of the angular velocity vector, $\boldsymbol{\omega}$.

$$\mathbf{x}_{\text{rot}} = \left(\sigma_1, \sigma_2, \sigma_3, \omega_1, \omega_2, \omega_3\right)^{\mathsf{T}} \tag{B.3}$$

B.2. Comparison of Rotational State Representations

As previously explained, Modified Rodrigues Parameters provide a three component attitude parameterisation, which, by switching to an equivalent shadow set, can effectively avoid singularities in the attitude characterisation. This is in contrast to the quaternion representation, on which the Modified Rodrigues Parameters are based, as they provide a four component characterisation of the Euler eigenaxis and the corresponding rotation angle around it. This means that MRP propagation has to integrate one parameter less when compared to the quaternion propagation, improving the computational efficiency. Moreover, when quaternions are employed to represent the attitude of a vehicle, a unit-length constraint needs to be imposed, as otherwise the system becomes overdetermined. Hence, after each calculation step, the quaternion length needs to be normalised. When considering MRP propagation, it is also required to evaluate the norm of the MRP or SMRP vector and, in case this norm exceeds one, a switch between the different sets needs to be performed.

Furthermore, when comparing the kinematic attitude equations of both representations, it should be noted that the equation in terms of quaternion elements is linear, whereas the equation in terms of MRP elements contains nonlinear terms. Nevertheless, these nonlinear terms only contain quadratic nonlinearities and no transcendental functions, limiting their overall complexity. Therefore, it was initially decided not to use the quaternion propagation method, as it requires four parameters to be propagated, whereas MRP propagation requires only three. Hence resulting in a decrease of 25% in computational load, even though quadratic equations are propagated rather than linear ones. On top of that, it is given by Vittaldev et al. (2012) that the MRP representation is getting more popular for spacecraft attitude control, especially for (I)NDI control, as for example provided by Acquatella et al. (2012).

When it became apparent that the developed MRP-based attitude controller would be ineffective, it was decided to employ a quaternion-based approach, given the beneficial characteristics as previously discussed. Moreover, it is provided by Shuster (1993) that quaternions are ideal for simulations as the resulting kinematics are linear. Therefore, this representation has traditionally been selected as the preferred parametrisation for spacecraft attitude control systems. Hence it is evident that, as alternative set, the quaternion representation was selected. Especially when considering that the discontinuous nature of a changing MRP reference profile is the root cause that renders the MRP-based attitude controller ineffective. This is in contrast to the sine-like continuous nature, as established for a changing quaternion reference profile, resulting in an effective attitude controller. Consequently, it is not possible to integrate the spacecraft attitude dynamics using the MRP representation in the motion simulator and subsequently transform this into an attitude quaternion before the signal is fed to the INDI controller, as the transformation does not generate the necessary sine-like continuous quaternion profile. Rather, it generates a discontinuous profile where switches occur between the equivalent positive and negative quaternion representation, resulting in an equally ineffective attitude controller. Hence based on this discussion, it is evident that a change to quaternion representation in both the motion simulator and the attitude controller was absolutely necessary.

B.3. MRP-based Attitude Controller

When applying the INDI technique as full state attitude controller of the chaser, its task is to perform synchronisation with the target spacecraft. To achieve this, MRP error kinematics will be used rather than normal MRP kinematics. This follows from the exact same reasoning as provided for the quaternion-based INDI controller, where increased control effectiveness can be achieved when error kinematics are used. Therefore, the MRP error vector is defined in Equation (B.4), where $\delta\boldsymbol{\sigma}$ represents the MRP error vector, $\boldsymbol{\sigma}_{\mathrm{T}}$ the MRP vector of the target spacecraft, $\boldsymbol{\sigma}_{\mathrm{C}}$ the MRP vector of the chaser spacecraft and $\mathbf{S}()$ the anti-symmetric matrix defined in Equation (B.5).

$$\delta\boldsymbol{\sigma} = \frac{\boldsymbol{\sigma}_{\mathrm{T}}\left(\boldsymbol{\sigma}_{\mathrm{C}}^{\mathsf{T}}\boldsymbol{\sigma}_{\mathrm{C}} - 1\right) + \boldsymbol{\sigma}_{\mathrm{C}}\left(1 - \boldsymbol{\sigma}_{\mathrm{T}}^{\mathsf{T}}\boldsymbol{\sigma}_{\mathrm{T}}\right) - 2\mathbf{S}\left(\boldsymbol{\sigma}_{\mathrm{T}}\right)\boldsymbol{\sigma}_{\mathrm{C}}}{1 + \boldsymbol{\sigma}_{\mathrm{T}}^{\mathsf{T}}\boldsymbol{\sigma}_{\mathrm{T}}\boldsymbol{\sigma}_{\mathrm{C}}^{\mathsf{T}}\boldsymbol{\sigma}_{\mathrm{C}} + 2\boldsymbol{\sigma}_{\mathrm{T}}^{\mathsf{T}}\boldsymbol{\sigma}_{\mathrm{C}}} \tag{B.4}$$

$$\mathbf{S}(\mathbf{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{B.5}$$

The corresponding kinematic and dynamic equations expressing the spacecraft attitude motion in terms of the MRP error vector are given in Equations (B.6) and (B.7). In these equations, $\mathbf{J}$ represents the inertia tensor of the spacecraft, $\mathbf{M}$ the control moment vector, $\boldsymbol{\omega}$ the angular velocity vector and $\delta\boldsymbol{\omega}$ the error angular velocity vector, which is defined in Equation (B.8).

$$\dot{\delta\boldsymbol{\sigma}} = \frac{1}{4}\left(\left(1 - \delta\boldsymbol{\sigma}^{\mathsf{T}}\delta\boldsymbol{\sigma}\right)\mathbf{I}_{3\times3} + 2\mathbf{S}\left(\delta\boldsymbol{\sigma}\right) + 2\delta\boldsymbol{\sigma}\delta\boldsymbol{\sigma}^{\mathsf{T}}\right)\delta\boldsymbol{\omega} = \mathbf{D}\left(\delta\boldsymbol{\sigma}\right)\delta\boldsymbol{\omega} \tag{B.6}$$

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}\left(\mathbf{M} - \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}\right) \tag{B.7}$$

$$\delta\boldsymbol{\omega} = \boldsymbol{\omega} - \mathbf{C}_{\mathbf{B,C/B,T}}\boldsymbol{\omega}_{\mathrm{cmd}} \tag{B.8}$$

In Equation (B.8), the error angular velocity vector is calculated from the actual velocity vector of the spacecraft, denoted by $\boldsymbol{\omega}$, and the commanded angular velocity, denoted by $\boldsymbol{\omega}_{\mathrm{cmd}}$. Note that for attitude synchronisation with a tumbling debris item, this commanded angular velocity vector simply equals the estimated angular velocity vector of the target spacecraft. Moreover, this equation utilises the rotation matrix from the body frame of the target, $\mathcal{F}_{\mathrm{B,T}}$, to the body frame of the chaser, $\mathcal{F}_{\mathrm{B,C}}$, and can be defined by Equation (B.9) (L. Sun and Huo, 2015).

$$\mathbf{C}_{\mathbf{B,C/B,T}} = \mathbf{I}_{3\times3} - \frac{4\left(1 - \delta\boldsymbol{\sigma}^{\mathsf{T}}\delta\boldsymbol{\sigma}\right)}{\left(1 + \delta\boldsymbol{\sigma}^{\mathsf{T}}\delta\boldsymbol{\sigma}\right)^2}\mathbf{S}\left(\delta\boldsymbol{\sigma}\right) + \frac{8}{\left(1 + \delta\boldsymbol{\sigma}^{\mathsf{T}}\delta\boldsymbol{\sigma}\right)^2}\mathbf{S}\left(\delta\boldsymbol{\sigma}\right)\mathbf{S}\left(\delta\boldsymbol{\sigma}\right) \tag{B.9}$$

Note also here that, to allow for more rapid control, the Time Scale Separation principle is applied during the dynamic inversion. Hence when considering the innerloop rate control, its objective is to track a desired angular velocity reference profile, given by $\boldsymbol{\omega}_{\mathrm{des}}$, obtained from the outerloop attitude controller. Hence this innerloop is concerned with the attitude dynamics given by Equation (B.7) and it is evident that the controlled variables equal the angular velocities. The general nonlinear MIMO system given in Equation (6.11) and (6.12) can be expressed in terms of the innerloop rate controller, when using the definitions given in Equations (B.10) to (B.15).

$$\mathbf{x}_{\mathrm{in}} = \boldsymbol{\omega} = \left(\omega_1, \omega_2, \omega_3\right)^{\mathsf{T}} \tag{B.10}$$

$$\mathbf{u}_{\mathrm{in}} = \mathbf{M} = \left(M_x, M_y, M_z\right)^{\mathsf{T}} \tag{B.11}$$

$$\mathbf{y}_{\mathrm{in}} = \mathbf{x}_{\mathrm{in}} = \left(\omega_1, \omega_2, \omega_3\right)^{\mathsf{T}} \tag{B.12}$$

$$f(\mathbf{x}_{\mathrm{in}}) = -\mathbf{J}^{-1}\left(\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}\right) \tag{B.13}$$

$$\mathbf{G}(\mathbf{x}_{\mathrm{in}}) = \mathbf{J}^{-1} \tag{B.14}$$

$$h(\mathbf{x}_{\mathrm{in}}) = \mathbf{I}_{3\times3}\mathbf{x}_{\mathrm{in}} \tag{B.15}$$

Differentiating the output $\mathbf{y}_{\mathrm{in}}$ with respect to time allows the calculation of an explicit relation between the output of the system and the incremental input to the system needed to generate the feedback linearisation loop, as given in Equation (B.16).

$$\dot{\mathbf{y}}_{\mathrm{in}} = \dot{\mathbf{x}}_{\mathrm{in}} = f(\mathbf{x}_{\mathrm{in}}) + \mathbf{G}(\mathbf{x}_{\mathrm{in}})\mathbf{u}_{\mathrm{in}} \tag{B.16}$$

By setting the rate of the system output, $\dot{\mathbf{y}}_{\mathrm{in}}$, equal to the virtual control input, $\mathbf{v}_{\mathrm{in}}$, the incremental control input, $\delta\mathbf{u}_{\mathrm{in}}$, that linearises the system in Equation (B.17) can be obtained.

$$\delta \mathbf{u}_{in} = \mathbf{G}(\mathbf{x}_{in,0})^{-1} \left(\mathbf{v}_{in} - \dot{\mathbf{x}}_{in,0} \right) = \mathbf{J} \left(\mathbf{v}_{in} - \dot{\boldsymbol{\omega}}_{meas} \right) \tag{B.17}$$

This is added to the reference control input, $\mathbf{u}_{in,0}$, to obtain the complete control input, $\mathbf{u}_{in}$, given in Equation (B.18). Note that for this INDI controller, it is assumed that the time step is taken sufficiently small and the control update rate is sufficiently high such that the definition of the incremental control input, $\delta\mathbf{u}$, can be employed. Furthermore note that the value of the reference input, $\mathbf{u}_{in,0}$, is the measured output of the control actuators at the instance of the angular acceleration measurements, $\dot{\boldsymbol{\omega}}_{meas}$.

$$\mathbf{u}_{in} = \mathbf{u}_{in,0} + \delta \mathbf{u}_{in} \tag{B.18}$$

Lastly, the virtual control law provided by Equation (B.19), is designed using a simple proportional controller, as used by Acquatella et al. (2012) and Singh et al. (2019).

$$\dot{\mathbf{y}}_{in} = \dot{\boldsymbol{\omega}} = \mathbf{v}_{in} \tag{B.19}$$

A mathematical description of this control law is provided in Equation (B.20), where the proportional gain $K_{p,in}$ is used to remove any error between the desired angular velocity, $\boldsymbol{\omega}_{des}$, and the actual angular velocity of the spacecraft, $\boldsymbol{\omega}$.

$$\mathbf{v}_{in} = K_{p,in}(\boldsymbol{\omega}_{des} - \boldsymbol{\omega}) \tag{B.20}$$

When subsequently considering the outerloop attitude controller, its objective is twofold: it generates the required angular velocity profile for the innerloop, $\boldsymbol{\omega}_{des}$, and tracks the attitude of the target spacecraft to maintain synchronisation. Hence this outerloop is concerned with the MRP error kinematics given by Equation (B.6) and it is evident that the controlled variable equals the MRP error vector, $\delta\boldsymbol{\sigma}$. Hence the general nonlinear MIMO system given in Equation (6.11) and (6.12) can be expressed in terms of the outerloop attitude controller when using the definitions given in Equations (B.21) to (B.26).

$$\mathbf{x}_{out} = \delta\boldsymbol{\sigma} = \left(\delta\sigma_1, \delta\sigma_2, \delta\sigma_3 \right)^T \tag{B.21}$$

$$\mathbf{u}_{out} = \boldsymbol{\omega} = \left(\omega_1, \omega_2, \omega_3 \right)^T \tag{B.22}$$

$$\mathbf{y}_{out} = \delta\boldsymbol{\sigma} = \left(\delta\sigma_1, \delta\sigma_2, \delta\sigma_3 \right)^T \tag{B.23}$$

$$f(\mathbf{x}_{out}) = -\mathbf{D}\left(\delta\boldsymbol{\sigma}\right) \mathbf{C}_{\mathbf{B},\mathbf{C}/\mathbf{B},\mathbf{T}} \boldsymbol{\omega}_{cmd} \tag{B.24}$$

$$h(\mathbf{x}_{out}) = \mathbf{I}_{3\times3}\mathbf{x}_{out} \tag{B.25}$$

$$G(\mathbf{x}_{out}) = \mathbf{D}\left(\delta\boldsymbol{\sigma}\right) \tag{B.26}$$

Differentiating the output, $\mathbf{y}_{out}$, with respect to time allows to calculate an explicit relation between the output of the system and the input to the system needed to generate the feedback linearisation loop, as given in Equation (B.27).

$$\dot{\mathbf{y}}_{out} = \dot{\mathbf{x}}_{out} = f(\mathbf{x}_{out}) + \mathbf{G}(\mathbf{x}_{out})\mathbf{u}_{out} \tag{B.27}$$

Note also here that, as the relation provided in Equation (B.27) is a given geometrical relation, it does not contain any uncertainties and can therefore effectively be controlled using an NDI controller. This can be achieved by setting the rate of the system output, $\dot{\mathbf{y}}_{out}$, equal to the virtual control input, $\mathbf{v}_{out}$, to obtain the control input, $\mathbf{u}_{out}$, that exactly linearises the system in Equation (B.28).

$$\mathbf{u}_{out} = \mathbf{G}(\mathbf{x}_{out,0})^{-1} \left(\mathbf{v}_{out} - f(\mathbf{x}_{out}) \right) = \mathbf{D}\left(\delta\boldsymbol{\sigma}\right)^{-1} \left(\mathbf{v}_{out} + \mathbf{D}\left(\delta\boldsymbol{\sigma}\right) \mathbf{C}_{\mathbf{B},\mathbf{C}/\mathbf{B},\mathbf{T}} \boldsymbol{\omega}_{cmd} \right) \tag{B.28}$$

Note that this control input of the outerloop attitude controller is used as the design angular velocity profile for the innerloop, $\boldsymbol{\omega}_{des}$, as given in Equation (B.29). Furthermore note again that the commanded

angular velocity vector, $\boldsymbol{\omega}_{cmd}$, in Equation (B.28) simply equals the estimated angular velocity vector of the tumbling target.

$$\boldsymbol{\omega}_{des} = \mathbf{u}_{out} \tag{B.29}$$

Lastly, the virtual control law provided by Equation (B.30) is designed using a straightforward proportional controller as employed by Singh et al. (2019).

$$\dot{\mathbf{y}}_{out} = \delta\dot{\boldsymbol{\sigma}} = \mathbf{v}_{out} \tag{B.30}$$

A mathematical depiction of this control law is provided in Equation (B.31), where the proportional controller with gain $K_{p,out}$ is used in combination with the MRP error vector to synchronise the attitude of the chaser spacecraft with the attitude of its target.

$$\mathbf{v}_{out} = K_{p,out}\,(-\delta\boldsymbol{\sigma}) \tag{B.31}$$

To characterise the control law, the definitions for the inner- and outerloop gains and the operational frequency are provided in Table B.1. It can be seen that for a fair comparison, it is chosen to use the same gains as for the quaternion-based INDI controller. Nevertheless, some analysis was performed to ensure that the system response to a step input is comparable to the system response of the quaternion-based controller, but due to time constraints, no further optimisation could be performed.

Table B.1: INDI control algorithm settings

Parameter	Value
$K_{p,in}$ [-]	5
$K_{p,out}$ [-]	0.1
operational frequency [Hz]	100

B.4. Attitude Controller Analysis

To analyse the performance of the MRP-based INDI controller, first a simple system response to a sigmoid input is simulated. This is done, given that it provides a relatively simple test case with a potential true implementation as reorientation manoeuvre. As a benchmark, also the system response of the quaternion-based INDI controller is provided. For a fair comparison, both version of the INDI controller use ideal angular accelerations measurements as input, as well as ideal actuator outputs and corresponding output measurements. These simplifying assumptions are made to ensure that the performed analysis is solely centred around the control algorithm performances, rather than the accuracy of the accompanying sensors and/or actuators.

The time responses of two satellite systems tasked with an intermediate reorientation manoeuvre are provided in Figure B.1. For the left side of this figure, a satellite system using a quaternion-based control algorithm is simulated, whereas the right side of the figure is generated using Modified Rodrigues parameters. The simulated system responses are provided for the INDI controllers previously described in Chapter 6 and Appendix B, with their respective parameters as defined in Tables 6.2 and B.1. Note that Figure B.1 represents an equivalent manoeuvre for both the quaternion-based and MRP-based representations. Furthermore, note that in addition to the reference attitude command provided to the INDI controllers, also a reference angular velocity profile is needed. To this extent, the angular velocity commands are obtained by taking the derivative of the respective attitude commands, as provided in Figure B.2.

Although both attitude representations provide the same manoeuvre, numerical differences are obtained in the calculated angular velocity profiles due to the simple derivative procedure that has been applied. Nevertheless, when inspecting the attitude responses in Figure B.1, it can be observed that both controllers have a low settling time and are capable of removing any steady-state errors. Moreover, despite using different attitude parameterisations, the time responses of both systems are relatively similar. The slight differences that are visible, however, are caused by the discrepancies in the commanded angular velocity profiles. These difference in rate commands do not only change the

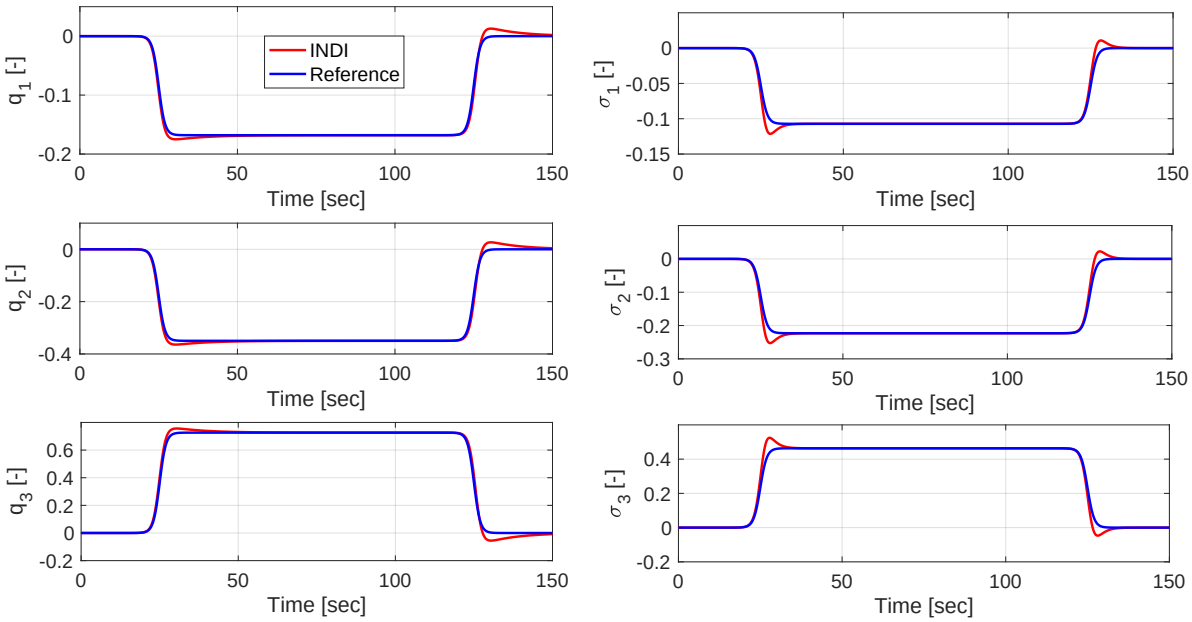


Figure B.1: Attitude profiles of quaternion-based (left) and MRP-based (right) INDI controllers during attitude reorientation manoeuvre

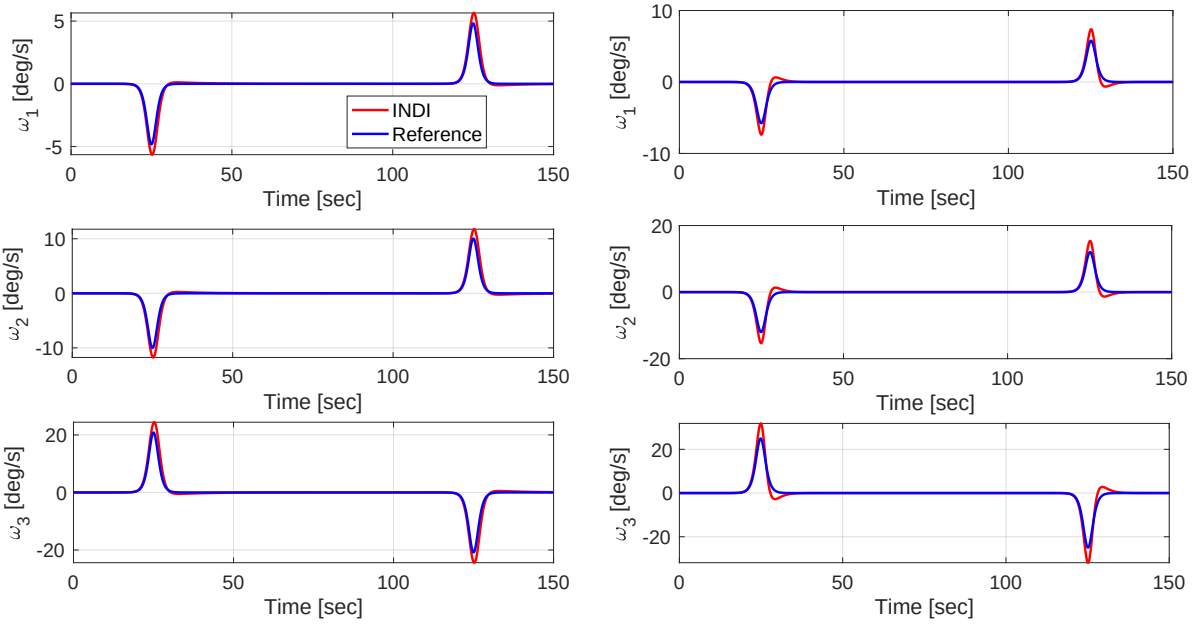


Figure B.2: Angular velocity profiles of quaternion-based (left) and MRP-based (right) INDI controllers during attitude reorientation manoeuvre

attitude response of the system, but also the corresponding commanded control moments, as can be seen in Figure B.3. The larger angular velocity command for the MRP-based controller evidently results in an increased corresponding control moment. Nonetheless, when inspecting the control moments of both reorientation manoeuvres, it is found that their magnitudes remain bounded and quickly damp to zero, indicating that both systems are stable. Therefore, sufficient proof is provided that the developed MRP-based controller is implemented correctly and can be used to control a spacecraft during a simple reorientation manoeuvre.

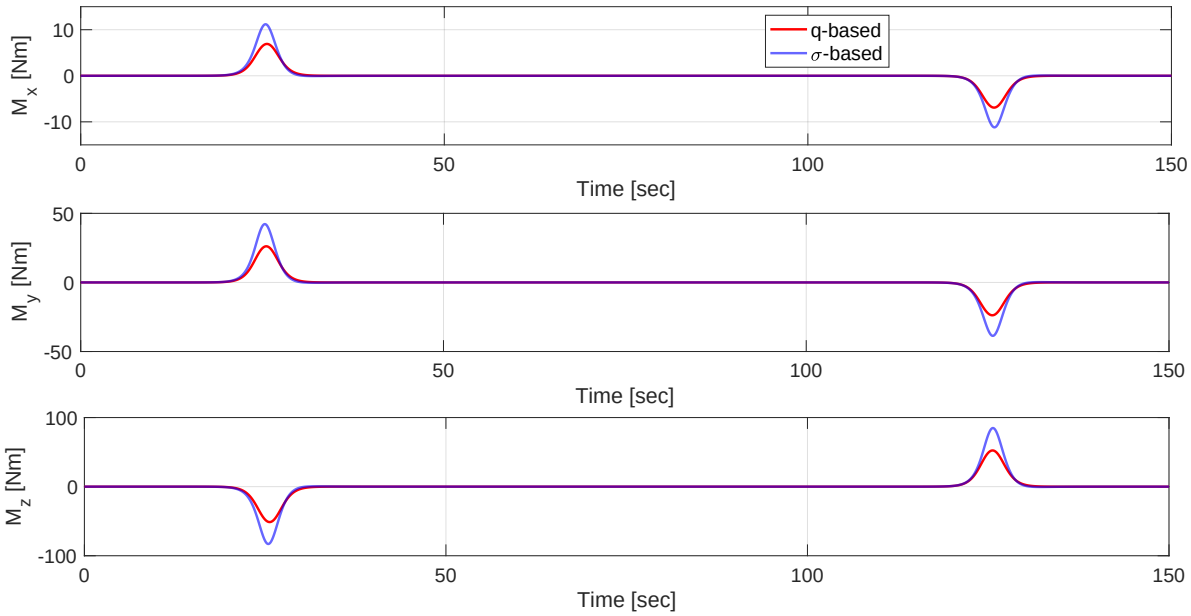


Figure B.3: Control moment profiles of quaternion-based and MRP-based INDI controllers during attitude reorientation manoeuvre

To further analyse the performance of the previously discussed controller, the system response is simulated during an attitude synchronisation manoeuvre with a tumbling debris item. This is done, as it exactly mimics what is required from the attitude control system during the actual mission scenario. Once again, for a fair and centralised comparison, both version of the INDI controllers are simulated with access to ideal angular accelerations measurements, as well as ideal actuators and ideal actuator output measurements. The basic simulation parameters used to define this analysis are taken from the definitions provided in Table 3.1, and the respective controller parameters are defined following the descriptions provided in Tables 6.2 and B.1. Note however that, for the considered simulation scenario, it is assumed that the angular velocity of the chaser and target spacecraft are already synchronised, such that an implementation of PID rate controller can be omitted. Moreover, a tumbling rate of 3.5 deg/s over all three body axes is employed at the instant that attitude synchronisation is initiated. Lastly, the initial attitude quaternion of the target spacecraft is set to $(-0.1680, -0.3497, 0.7262, 0.5675)^T$, whereas $(0,0,0,1)^T$ is employed for the chaser. Using these definitions, the time responses of the two controllers can be obtained, as provided in Figure B.4. The left side of this figure presents the quaternion-based results, whereas the right side presents the MRP-based results. Furthermore, as a complement to this figure, the corresponding angular velocity error and attitude error during the synchronisation manoeuvre are provided in Figure B.5.

From Figure B.4 it can be seen that both controllers are capable of rapidly obtaining and subsequently maintaining attitude synchronisation with the tumbling debris item. However, when inspecting the synchronisation errors in Figure B.5, a discrepancy in the response of the MRP controller is identified with respect to the quaternion controller after approximately 23 seconds. This epoch is not arbitrary and corresponds with the first discrete switch from MRP to SMRP components in Figure B.4. This behaviour can also be seen when inspecting the required control moments, as provided in Figure B.6. Short oscillations are encountered at this first discrete switch of the MRP-based controller, whereas the quaternion-based controller remains smooth throughout the entire simulation time. These oscillations are, however, non-physical and undesirable for the durability of the control actuators, given that they are caused by the particular attitude representation, rather than the underlying dynamics of the spacecraft. Due to the switching nature of the MRP and SMRP sets, a discontinuous reference profile is generated by the tumbling debris item, and as a consequence of this discontinuous reference profile, a short oscillation is obtained. This oscillation occurs when the reference profile of the target spacecraft has already made a switch to the shadow set, whereas the attitude profile of the chaser is lagging behind in the nominal set. To compensate for this, large control actions are calculated resulting in an

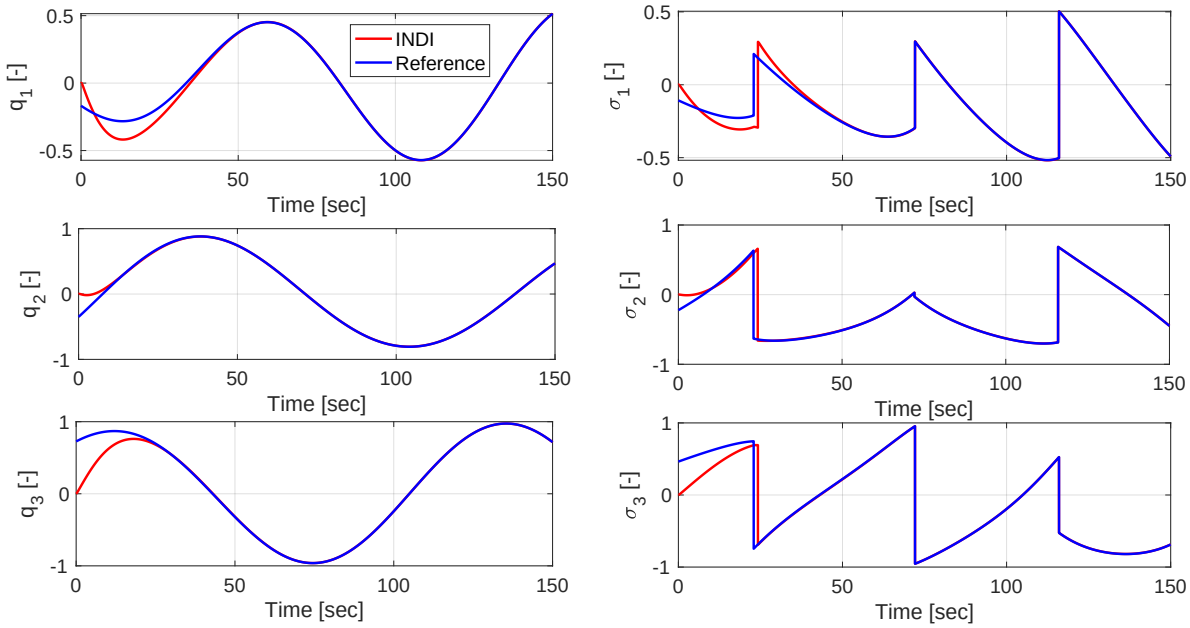


Figure B.4: Attitude profiles of quaternion-based (left) and MRP-based (right) INDI controllers during attitude synchronisation with debris item tumbling at 3.5 deg/s over all three body axes

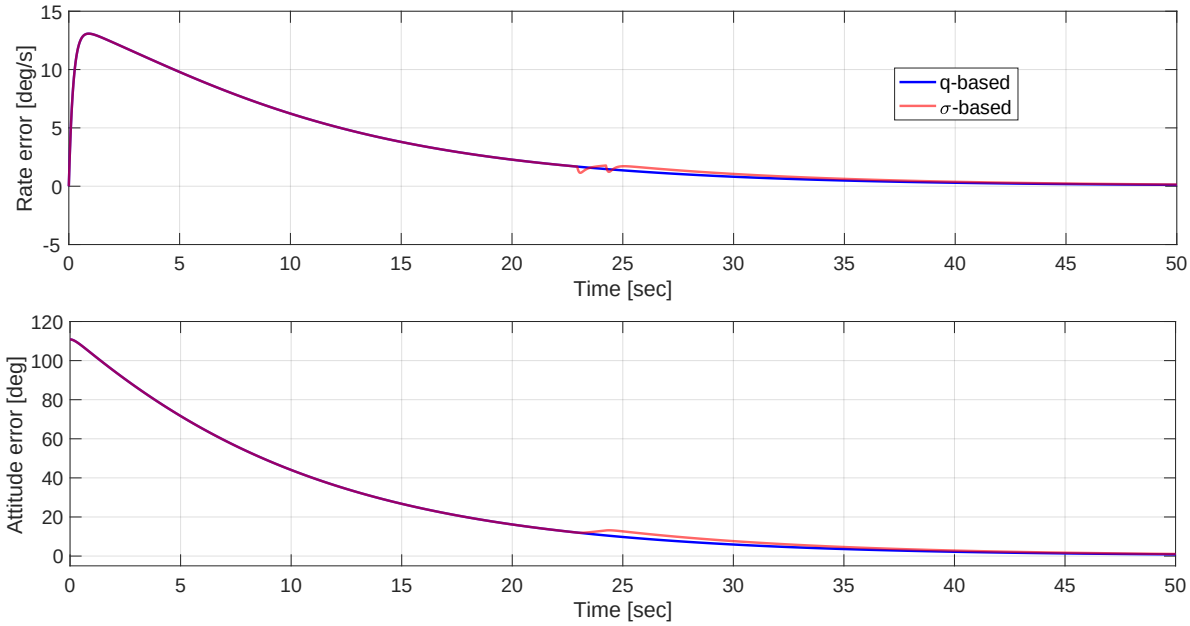


Figure B.5: Angular rate error and attitude error of quaternion-based and MRP-based INDI controllers during attitude synchronisation with debris item tumbling at 3.5 deg/s over all three body axes

angular velocity change that is used to induce a switch in the chaser's attitude. This change in angular rate can be seen from the short oscillation in rate error, as given in Figure B.5. However, once the attitude controller of the chaser has established sufficient synchronisation with its target, future switches of the reference profile will not induce oscillations. This is because both attitude profiles will perform these switches at the same instances. Therefore, in case attitude synchronisation can be established before a switch occurs, no oscillations will be present in the system. Nevertheless, this is not a realistic constraint that can be imposed on the system, nor desirable behaviour.

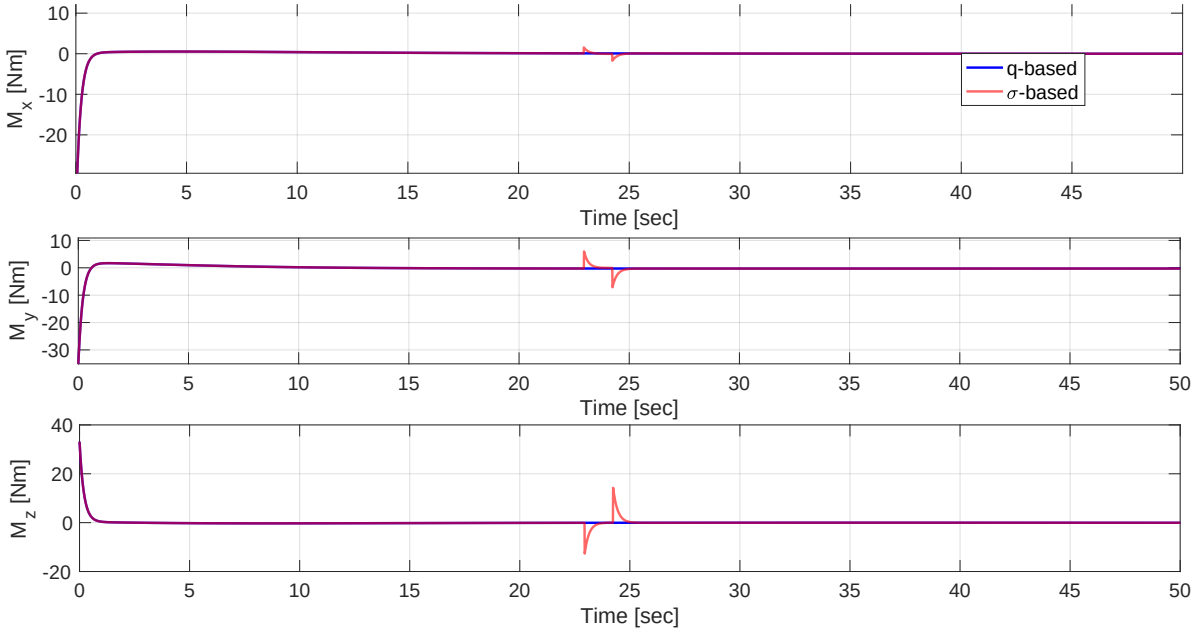


Figure B.6: Control moment profiles of quaternion-based and MRP-based INDI controllers during attitude synchronisation with debris item tumbling at 3.5 deg/s over all three body axes

From theory, one would expect that, by implementing an INDI controller based on the error MRP vector, as provided in Section B.3, such oscillations in the system response should not occur. When comparing the performance of the error MRP controller with an INDI controller implementing a standard MRP vector, it was found that most oscillations are indeed resolved by the error MRP representation, but not all. A potential work-around was sought by ensuring that the error MRP vector always represents the shortest relative rotation between both spacecraft, yet no performance improvement was obtained. Therefore, given the time constraints of the current thesis project, it was decided to omit further investigations on the MRP-based synchronisation controller. Instead, it was decided to change the attitude controller and motion simulator from MRP-based to quaternion-based, as it allows attitude synchronisation through a smooth and finite control actuator history.