Real-time, optimal, robust spacecraft reorientation using sequential convex programming

Master Thesis

Matthijs Diercks

Technische Universiteit Delft

TU Delft

OHB

This page was intentionally left blank.

Real-time, optimal, robust spacecraft reorientation using sequential convex programming

Master Thesis

by

Matthijs Diercks

in fulfilment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at Delft University of Technology,
to be defended publicly on Friday August 20, 2021 at 14:00.

Student number:	4362373	
Thesis committee:	Dr. Ir. E. Mooij	TU Delft (chair)
	Dr. Ir. E. van Kampen	TU Delft (examiner)
	Ir. R. Noomen	TU Delft (supervisor)
	Ir. J. Maene	OHB (supervisor)
Other supervisors:	Dr. Ir. F. de Bruijn	OHB

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

Cover picture source: `http://www.esa.int/ESA_Multimedia/Images/2017/09/Hurricane_Harvey_could_be_seen_easily_from_the_International_Space_Station`, Retrieved on 28-04-2020.

TU Delft

OHB

This page was intentionally left blank.

Preface

The report that lies in front of you marks the end of an incredibly exciting adventure over the last year. It has been a pleasure to work on such a cutting-edge field of research that, in my eyes, holds true promise for future onboard applications, both in the domain of spaceflight and elsewhere. As I did not stand alone in this endeavour, I hereby want to thank the people that enabled me to have this experience and supported me during it.

First of all, I would like to thank my supervisor, Ron Noomen, for his enthusiasm, interest in my work, and insightful views. Our weekly meetings with Jochim and Ferdi were stimulating and significantly contributed to the quality of this study. Perhaps as important, they were enjoyable and played a major role in motivating me to pursue our goals to the extent that we did.

Next, I would like to thank Jochim Maene and Ferdi de Bruijn for providing me with the unique opportunity to conduct relevant research at a leading player in the European space industry. I was deeply impressed by your sharp insights and commitment to this research project. From long discussions to late-night exchanges over Whatsapp and team lunches in Bremen, you were in every possible sense fundamental to the positive experience that this project was.

Then, I would like to thank the Mission Analysis team at OHB for creating the ideal atmosphere to work on this project. From great discussions to a shared quest for (too much) coffee, I feel lucky to have had such an experience, especially in times of Covid-19.

Finally, I would like to thank all friends and family that never failed to provide support, distraction when necessary, and who have made every leg of this journey worthwhile in the first place.

Matthijs Diercks
Delft, July 2021

This page was intentionally left blank.

Abstract

Reorientation manoeuvres represent a major aspect of space missions. Due to the scarceness of resources as time, energy and propellant in space, optimising these reorientation manoeuvres has received great research interest. With the unprecedented rise of CubeSats, agile satellites, and satellite constellations, solving these optimisation problems autonomously and onboard is considered to be a promising opportunity to achieve further performance improvements. However, the spacecraft reorientation problem is characterised by highly nonlinear dynamics, kinematics and constraints. Consequently, the nonlinear programming (NLP) methods currently used to solve this problem are too computationally expensive to be implemented onboard. Hence, developing a method to solve the optimal spacecraft reorientation problem in real time and onboard would be an important step towards future autonomous satellite applications. Recently, a novel class of so-called *sequential convex programming* (SCP) algorithms has shown promising performance regarding onboard optimisation. These methods convert a problem that is originally non-convex into a sequence of convex approximations that can be solved with state-of-the-art, highly efficient convex solving algorithms. However, these methods have only been applied to the spacecraft reorientation problem to a limited extent at this point in time.

Therefore, an SCP-based optimisation method to solve the spacecraft reorientation problem was developed in this study. Compared to earlier studies, this method was extended to include challenging problem elements: a minimum-time objective function, time-dependent boundary conditions and conical attitude constraints. Furthermore, several techniques were implemented to enhance the performance of the algorithm. Amongst others, a shape-based initialisation technique increased the computational efficiency by up to 40%. Moreover, a novel method to enforce constraints on inter-nodal segments was developed and found to effectively tackle the challenge of inter-nodal constraint violations. Finally, to bridge the gap from research to implementation, the convex solving algorithm was implemented directly from MATLAB, eliminating the need for time-consuming modelling software.

An extensive Monte Carlo campaign was conducted to study the performance of the SCP algorithm regarding six different formulations of the minimum-energy and minimum-time reorientation problems. In terms of speed, the algorithm showed a median computational efficiency that was a factor of 4.2 to 6.1 higher than a representative NLP benchmark algorithm. Furthermore, the algorithm delivered similar performance in terms of optimality as the NLP benchmark, which is widely regarded as the standard for obtaining optimal solutions for nonlinear optimisation problems. Finally, regarding robustness, 100% convergence rates were obtained for four of the six problem formulations. For the remaining two problem types, (conservative) convergence rates of 99.84% and 98.91% were observed. However, a strategy was identified that could potentially increase these rates in future research.

It should be noted that for final conclusions on the potential of SCP-based methods for onboard applications, tests will need to be executed on representative satellite hardware. Nevertheless, considering all results that were obtained, it can be concluded that SCP-based optimisation algorithms are one of the most promising candidates for onboard implementation, and could potentially pave the way for future autonomous applications.

This page was intentionally left blank.

Contents

This page was intentionally left blank.

Nomenclature

Acronyms and abbreviations

Abbreviation	Definition
2-D	Two-dimensional
3-D	Three-dimensional
4-D	Four-dimensional
CCS	Cartesian coordinate system
CDF	Cumulative distribution function
CGL	Chebyshev-Gauss-Lobatto
CMG	Control moment gyroscope
CP	Convex programming
CRP	Classical Rodrigues Parameter
DCM	Direction cosine matrix
DoF	Degree-of-freedom
ECI	Earth-centred inertial
EM	Euler method
ESA	European Space Agency
FOH	First-order-hold
FoV	Field of view
fRPM	Flipped Radau pseudospectral method
IPM	Interior point method
IQR	Interquartile range
JWST	James Webb Space Telescope
LP	Linear programming
LPM	Lobatto pseudospectral method
NLP	Nonlinear programming
MICP	Mixed-integer convex programming
MRP	Modified Rodrigues parameter
OCP	Optimal control problem
PS	Pseudospectral
PSO	Particle swarm optimisation
QP	Quadratic programming
RK4	Runge-Kutta fourth order
SC	Successive convexification
SCP	Sequential convex programming
SDP	Semidefinite programming
SLERP	Spherical linear interpolation
SOCP	Second-order cone programming
SQP	Sequential quadratic programming
TM	Trapezoidal method
TPBVP	Two-point boundary value problem
ZOH	Zero-order-hold

Latin symbols

Symbol	Description	Unit
A	Partial derivative of the time-normalised dynamics w.r.t. the state	[-]
B	Partial derivative of the time-normalised dynamics w.r.t. the control	[-]
$\boldsymbol{C}$	Partial derivative of the time-normalised dynamics w.r.t. the time dilation factor	[-]
$\boldsymbol{C}$	Direction Cosine Matrix	[-]
$\boldsymbol{c}_u$	Trapezoidal objective quadrature	[-]
D	Pseudospectral differentiation matrix	[-]
$\boldsymbol{E}$	Orthonormal basis	[-]
$\boldsymbol{e}$	Coordinate axis	[-]
F	Time-normalised dynamics	[-]
F_B	Body-fixed reference frame	[-]
F_I	Inertial reference frame	[-]
f	Problem dynamics	[s^{-1}] or [rad s^{-2}]
$\boldsymbol{h}$	Reaction wheel angular momentum	[kg m^2 s^{-1}]
$\boldsymbol{I}$	Inertia matrix	[kg m^2]
I	Rigid-body principal moment of inertia	[kg m^2]
J	Objective function	[-]
K	Number of grid nodes	[-]
K_{disc}	Number of nodes for the discretisation step	[-]
M	Number of pseudospectral collocation nodes	[-]
N	Number of pseudospectral discretisation nodes	[-]
$n_{\text{it,lim}}$	Iteration limit	[-]
$n_{\text{tr,adap,thres}}$	Threshold for the adaptive trust-region technique	[-]
n_u	Number of control parameters	[-]
n_x	Number of state parameters	[-]
n_z	Number of optimisation parameters	[-]
$\boldsymbol{p}$	Time-dependent boundary condition	[-] or [rad s^{-1}]
$\boldsymbol{p}_c$	Polynomial coefficient for a boundary condition	[-]
$\boldsymbol{q}$	Quaternion	[-]
$\boldsymbol{q}_0$	Initial quaternion boundary condition	[-]
$\boldsymbol{q}_f$	Final quaternion boundary condition	[-]
S	Scaling factor	[-]
s	Scaling offset	[-]
s_a	Slack variable (general)	[-]
s_{tr}	Slack variable for the trust-region objective term	[-]
s_u	Slack variable for the minimum-energy objective	[-]
t	Real time	[s]
t_0	Initial time	[s]
t_f	Final time, manoeuvre duration	[s]
$t_{f,\text{guess}}$	Initial guess for the manoeuvre duration	[s]
$\boldsymbol{u}$	Control vector	[Nm]
$\boldsymbol{u}_{\text{ellipse}}$	Ellipsoidal control constraint	[Nm]
w_t	Weight factor for the minimum-time objective term	[-]
w_{tr}	Weight factor for the trust-region objective term	[-]
w_u	Weight factor for the minimum-energy objective term	[-]
w_{vc}	Weight factor for the virtual-control objective term	[-]
w_η	Weight factor for the minimum-time objective term	[-]
$\boldsymbol{x}$	State parameter	[-] or [rad s^{-1}]
$\boldsymbol{x}_0$	Initial state boundary condition	[-] or [rad s^{-1}]
$\boldsymbol{x}_B$	Instrument boresight vector (in the F_B frame)	[-]
$\boldsymbol{x}_e$	Vector of ECOS optimisation parameters	[-]
$\boldsymbol{x}_f$	Final state boundary condition	[-] or [rad s^{-1}]
$\boldsymbol{y}_I$	Central vector of a keep-out or keep-in cone	[-]
$\boldsymbol{z}$	Vector of optimisation parameters	[-] or [rad s^{-1}] or [s]

Greek symbols

Symbol	Description	Unit
$\alpha_{\text{tr,adap}}$	Update factor for the adaptive trust-region term	[-]
$\alpha_{\text{vc,adap}}$	Update factor for the adaptive virtual-control term	[-]
δ_η	Hard trust region for the time dilation factor	[%]
ϵ	Convergence criterion	[-]
ϵ_{vc}	Convergence criterion for the virtual control use	[-]
ϵ_x	Convergence criterion for the state deviation	[-]
ϵ_η	Convergence criterion for the time dilation factor	[-]
η	Time dilation factor	[-]
η	Pseudo time	[-]
$\theta_{\text{sep,in}}$	Attitude keep-in maximum separation angle	[rad]
$\theta_{\text{sep,out}}$	Attitude keep-out minimum separation angle	[rad]
$\boldsymbol{\lambda}$	Trust region slack variable	[-]
$\boldsymbol{\mu}$	Trust region slack variable	[-]
$\boldsymbol{\nu}$	Virtual control parameter	[-]
τ	Normalised trajectory time	[-]
Φ	State transition matrix	[-]
Φ	Mayer objective term	[-]
$\boldsymbol{\phi}$	Euler angle	[rad]
Ψ	Lagrange objective term	[-]
$\boldsymbol{\omega}$	Angular rate vector	[rad s^{-1}]
$\boldsymbol{\omega}_0$	Initial angular rate boundary condition	[rad s^{-1}]
$\boldsymbol{\omega}_{\text{box}}$	Box angular rate constraint	[rad s^{-1}]
$\boldsymbol{\omega}_f$	Final angular rate boundary condition	[rad s^{-1}]

This page was intentionally left blank.

I

Theory and Background

1 Introduction

In this chapter, the topic of this research project is introduced. First, in Section 1.1, the problem around which this study revolves is identified. Then, Section 1.2 presents the development that led to this research project. Afterwards, in Section 1.3, the research (sub-)questions are introduced. Finally, Section 1.4 provides a short overview of the structure of this report.

1.1. Problem and relevance

Reorientation manoeuvres are an important element of most space missions. These manoeuvres can be required for a large variety of purposes, ranging from establishing a data link with a ground station to observing a target on Earth or tracking a passing comet for scientific reasons. However, the resources that are required for these manoeuvres, such as time, propellant and energy, are extremely scarce in space. Therefore, optimising these attitude manoeuvres with respect to one or more of these variables has received a lot of attention over the past decades.

In recent years, the rise of agile satellites, satellite constellations and small satellites has created a push towards higher performance of space missions, resulting in a need for satellites to achieve further efficiency improvements (Kjellberg, 2014). Performing optimised attitude manoeuvres autonomously would be a major step towards answering this need (Aucoin and Zee, 2019), as this would have many advantages. For instance, the workload on ground infrastructure could be reduced through a decreased need for contact regarding manoeuvre planning purposes. Furthermore, this capability would enable spacecraft to be more adaptive to certain mission elements, for instance, changing weather conditions (which can influence the collected data), the need for responsive delivery of military information (Boyarko et al., 2011), or the unpredictable nature of certain scientific phenomena (Preda et al., 2021). Moreover, it could increase the amount of useful work (e.g., image collection capacity) a spacecraft could perform in a given period of time (observation window) compared to the sub-optimal methods that are currently used for onboard applications (Ventura et al., 2015).

However, the optimal spacecraft reorientation problem is characterised by complex, nonlinear dynamics and constraints. The pseudospectral nonlinear programming methods that are typically used to solve this optimisation problem show good performance in terms of robustness and optimality but suffer from low computational efficiency, rendering these methods unsuitable for onboard applications (Boyarko et al., 2011). As a result, the corresponding optimisation process currently has to be performed using high-performance computers located on Earth. Furthermore, previous methods proposed to solve the spacecraft reorientation problem in real time suffer from penalties with respect to robustness or optimality, and typically rely on certain problem simplifications (Section 3.8). Consequently, there exists great interest in the development of novel methods that could reliably solve the optimal spacecraft reorientation problem onboard and in real time, thereby bringing the space community one step closer to autonomous applications (Ventura et al., 2015).

1.2. Research trigger

Recent advances in the field of optimal control have spurred a wave of research into the development of onboard guidance algorithms for a variety of space-related problems. A class of algorithms that shows particularly promising results, is composed of so-called convexification-based optimisation methods (Liu et al., 2017). These methods convert an originally non-convex optimisation problem into either a single convex problem (*lossless convexification*) or a sequence of convex sub-problems (*sequential convex programming* or *successive convexification*), which can be solved extremely efficiently using state-of-the-art convex solving algorithms. Successful applications of this framework can be found in a wide range of space-related research fields, such as powered descent, orbital rendezvous, and hypersonic re-entry. More recently, a limited number of studies identified the sequential convex programming (SCP) framework as a promising option for efficiently solving the optimal spacecraft reorientation problem. However, the effectiveness of SCP-based optimisation methods regarding this problem has, at this point in time, only been investigated to a limited extent.

1.3. Research question

The purpose of this study is to obtain a thorough understanding of the capabilities of a sequential convex programming-based optimisation algorithm in terms of solving the spacecraft reorientation problem onboard. In this section, the research question and sub-questions are presented that were investigated in this study. These research (sub-)questions were formulated in close consultation with the research partner OHB. The research question that is to be answered in this study is:

Could a sequential convex programming-based optimisation method be a candidate for solving the optimal spacecraft reorientation problem in onboard applications?

Answering this question will directly contribute to the search for a solution to the problem identified in Section 1.1. This research question was further split up into four sub-questions:

1. **Could SCP-based optimisation methods be used to solve the complex minimum-time reorientation problem with time-dependent boundary conditions and conical attitude constraints?**

At the moment of writing, only a limited number of studies have focused on the application of the SCP framework to the optimal spacecraft reorientation problem. These studies have considered reorientation problems that were relatively simple in terms of formulation and enforced constraints. To be a promising candidate for many autonomous applications, it is important to understand which problem types the SCP framework could successfully solve. To that end, extending the application of SCP algorithms to include complex problem elements, specifically a free-final-time formulation, time-dependent boundary conditions and conical attitude constraints, is a focus of this research project.

2. **How could the performance of current SCP-based optimisation algorithms be improved?**

In order to meet the rigorous demands for an onboard implementation, it would be interesting to identify methods to improve the performance of SCP-based optimisation algorithms, compared to the algorithms that have currently been proposed in literature. These methods can either be obtained from SCP-related studies on other optimisation problems (e.g., the powered descent problem) or developed in this study from scratch.

3. **How could the first steps towards an onboard implementation of an SCP algorithm be executed?**

Current studies on SCP-based optimisation algorithms typically rely on software such as a numerical computing environment (e.g., MATLAB) and an interfacing tool (e.g., CVX) which greatly simplify the development of these optimisation methods. However, these tools are not available (from a software perspective) or too computationally inefficient for onboard use. In order to bridge the gap between research and actual implementation, it would be interesting to explore how these tools could be eliminated and to which extent this influences the performance of the algorithm.

4. **What are the performance characteristics of an SCP-based optimisation algorithm in terms of computational efficiency, optimality, and robustness?**

To assess the usability of SCP-based optimisation methods for onboard applications, it is crucial to obtain a thorough understanding of their performance. To this end, the three most important criteria for onboard implementation were identified and extensively studied. In Section 3.7, these three performance criteria (computational efficiency, optimality and robustness) are introduced and defined.

1.4. Report structure

In this section, a brief overview of the structure of this report is provided. First, in Chapter 2, a theoretical foundation is provided which is required to understand the problem at hand. Then, in Chapter 3, the optimal spacecraft reorientation problem is introduced and discussed. Afterwards, Chapter 4 outlines the main design choices and theory behind the sequential convex optimisation algorithm developed in this study. Chapter 5 discusses the way in which this algorithm and other relevant analysis tools were implemented for this research project. Subsequently, in Chapter 6, the set-up of the Monte Carlo experiments that were conducted is covered. Afterwards, Chapter 7 studies the performance of the SCP algorithm on a single-case level. In Chapter 8, the influence of several important design choices and elements on the performance of the SCP algorithm is analysed. Next, in Chapter 9, various Monte Carlo analyses are presented and discussed to draw conclusions about the performance characteristics of the SCP algorithm. Finally, Chapter 10 presents the main conclusions of this study and recommendations for further research.

This page was intentionally left blank.

2

Theory

This chapter provides the theoretical background information which is required to understand the research that was performed during this MSc thesis project and its context. In Section 2.1, the two reference frames that are used in this study are introduced. Then, in Section 2.2, the quaternion attitude parameterisation is introduced. Afterwards, a brief overview of optimal control theory is provided in Section 2.3. Finally, in Section 2.4, an introduction is provided to convex optimisation and the convexification-based optimisation methods that form the focus of this study.

2.1. Reference frames

Before spacecraft reorientation manoeuvres can be studied, first the reference frames that are used to describe the attitude and motion of a spacecraft, have to be defined. A reference frame is typically defined as a set of three orthonormal physical basis vectors and an origin (Wertz, 1978). In space-related research, a wide variety of different reference frames are encountered that can be used to describe complex problems. However, the optimal spacecraft reorientation problem considered in this study only requires two relatively basic and intuitive reference frames to fully describe the problem. This results from the fact that this study only considers the attitude of a spacecraft and not its translational position and motion in space. The two reference frames that are used in this study are the *inertial* reference frame F_I (Subsection 2.1.1), in which the boundary conditions and attitude constraints of the manoeuvres are usually specified, and the *body-fixed* reference frame F_B (Subsection 2.1.2), which is required to describe the attitude and the rotational dynamics of a spacecraft.

2.1.1. Inertial frame

The inertial reference frame F_I is inertially fixed, which means that the axes of this frame do not rotate or change relative to distant stars (Wertz, 1978). For this study on the optimal spacecraft reorientation problem, the location of the origin of this reference frame is irrelevant as the physical position of the spacecraft is not considered and does not influence the problem formulation and dynamics. In a study of Virgili-Llop et al. (2019), this reference frame is simply referred to as the *inertial Cartesian coordinate system (CCS)* F_I. For potential extensions of this study, where the origin of the inertial reference frame *does* become relevant, it would be a logical choice to select the Earth-Centred Inertial (ECI) reference frame (Reynolds et al., 2021; Tam and Glenn Lightsey, 2016). This frame could, for instance, be used to describe and specify the location of observation targets relative to the spacecraft.

2.1.2. Body-fixed frame

The body-fixed reference frame F_B is used to describe the orientation of a studied body, in this case, a spacecraft, often with respect to an inertial reference frame. The centre of mass of the spacecraft is often taken as the origin of this reference frame, and its orientation is defined with respect to the geometrical features of the spacecraft. A visual overview of both the inertial and body-fixed reference frames is provided in Figure 2.1.

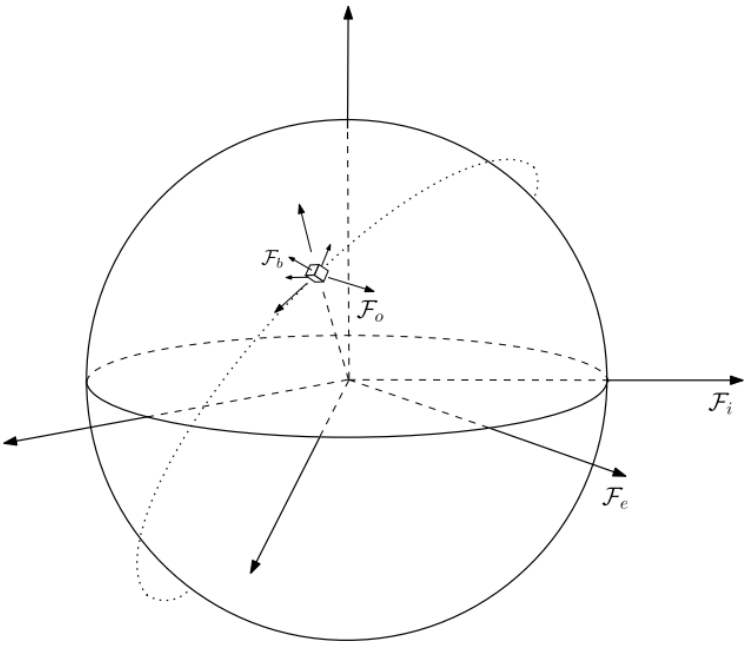


Figure 2.1: Overview of various reference frames, expressed in a generic way (Aucoin and Zee, 2019). The (Earth-centered) inertial frame F_i and the body-fixed frame F_b are relevant for the spacecraft reorientation problem considered in this study, while the Earth-fixed frame F_e and orbital frame F_o are not relevant for this study and, therefore, not introduced.

2.2. Attitude parameterisation

For the reorientation problem at hand, it is required to find a way to describe the attitude of the spacecraft. This attitude represents the orientation of the body-fixed reference frame F_B relative to the inertial reference frame F_I, which is used to define the boundary conditions and path constraints of the problem. Different methods exist that can be used to describe the orientation of one reference frame relative to another reference frame. These methods for expressing the attitude of a spacecraft are often referred to as *attitude parameterisation* methods (Diebel, 2006). In this section, the direction cosine matrix (DCM) and quaternion attitude parameterisations, which were used in this study, are introduced in Subsection 2.2.1 and Subsection 2.2.2, respectively. Subsection 2.2.3 outlines the main reasons for choosing the quaternion parameterisation for the sequential convex programming (SCP) algorithm developed in this study.

2.2.1. Direction cosine matrix

The attitude of a reference frame F_B relative to another reference frame F_I can be fully described by a direction cosine matrix (DCM) $\boldsymbol{C}_{BI} \in SO(3)$, where $SO(3) := \{\boldsymbol{C} \in \mathbb{R}^{3\times3}, \boldsymbol{C}\boldsymbol{C}^\top = \mathbf{1}, \det(\boldsymbol{C}) = 1\}$ (Diebel, 2006). $SO(3)$ is known as the matrix Lie group (Walsh et al., 2018). A DCM $\boldsymbol{C}$ represents a matrix that rotates the orthonormal basis of one reference frame to the orthonormal basis of another reference frame, under the assumption that the origins of both frames are identical. This rotation can be represented as

$$\boldsymbol{E}_B = \boldsymbol{C}_{BI}\boldsymbol{E}_I \tag{2.1}$$

where $\boldsymbol{E}_B$ and $\boldsymbol{E}_I$ denote, respectively, the orthonormal bases of reference frames F_B and F_I. The matrix $\boldsymbol{C}_{BI}$ is referred to as a DCM as each of its elements represents the angle between two coordinate axes from both orthonormal bases ($\boldsymbol{C}_{BI,ij} = \cos(\alpha_{ij})$, where α_{ij} is the angle between the coordinate axes $\boldsymbol{e}_{I,i}$ and $\boldsymbol{e}_{B,j}$) (Shuster, 1993). This relation can be formulated as (Bhagat, 2016)

$$\boldsymbol{C}_{BI} = \begin{bmatrix} \cos\alpha_{11} & \cos\alpha_{12} & \cos\alpha_{13} \\ \cos\alpha_{21} & \cos\alpha_{22} & \cos\alpha_{23} \\ \cos\alpha_{31} & \cos\alpha_{32} & \cos\alpha_{33} \end{bmatrix} \tag{2.2}$$

A useful property of using the DCM attitude parameterisation is that it is straightforward to obtain the backward transformation:

$$\boldsymbol{E}_I = \boldsymbol{C}_{BI}^{-1}\boldsymbol{E}_B = \boldsymbol{C}_{BI}^{T}\boldsymbol{E}_B \tag{2.3}$$

The main disadvantage of this attitude parameterisation is that it uses nine parameters to describe the relative orientation of two reference frames, rendering it computationally expensive to use for numerical optimisation problems (Bhagat, 2016). Nevertheless, the DCM representation is arguably one of the most intuitive attitude parameterisations, as a simple matrix-vector multiplication can be used to express any vector in any frame or to rotate a vector over any angle (Terzakis et al., 2018).

2.2.2. Quaternions

The quaternion attitude parameterisation is one of the most popular methods to parameterise the attitude. Conceptually, this attitude parameterisation is based on Euler's theorem, which states that any rotation from one reference frame to another reference frame can be described by a single rotation with a so-called *Euler angle* ϕ, around an axis which is known as the *Euler axis* (or *eigenaxis*) $\boldsymbol{e}$. This rotation is visualised in Figure 2.2.

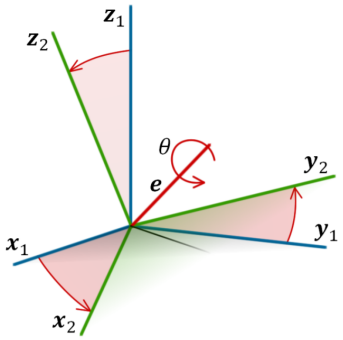


Figure 2.2: Transformation between two reference frames with the Euler angle θ around the Euler axis $\boldsymbol{e}$ (Ridder, 2016).

A quaternion consists of four variables that are functions of the Euler angle ϕ and eigenaxis $\boldsymbol{e}$.

$$\boldsymbol{q} = \begin{bmatrix} \overline{\boldsymbol{q}} \\ q_4 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{2.4}$$

where

$$\overline{\boldsymbol{q}} = \boldsymbol{e} \sin \frac{\phi}{2} \tag{2.5}$$

$$q_4 = \cos \frac{\phi}{2} \tag{2.6}$$

where $\overline{\boldsymbol{q}} = [q_1, q_2, q_3]^T$ and q_4 are referred to, respectively, as the *vector* and *scalar* components of the quaternion (Shuster, 1993). This formulation, where q_4 represents the scalar component, is used in this study and often referred to as the *scalar-last* convention. Other studies adopt a *scalar-first* convention, where the scalar component is represented by the q_1 parameter (Tam and Glenn Lightsey, 2016). A key property of a quaternion is that its L_2-norm is equal to 1, which can be represented as

$$\|\boldsymbol{q}\|_2 = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} = 1 \tag{2.7}$$

This property is often referred to as the quaternion *unit-norm*. The DCM $\boldsymbol{C}_{BI}$, which describes the attitude transformation from F_I to F_B, can be obtained from the quaternion parameters through (Wertz, 1978; Wie, 2008)

$$\boldsymbol{C}_{BI}(\boldsymbol{q}) = \begin{bmatrix} 1-2\left(q_2^2+q_3^2\right) & 2\left(q_1q_2+q_3q_4\right) & 2\left(q_1q_3-q_2q_4\right) \\ 2\left(q_2q_1-q_3q_4\right) & 1-2\left(q_1^2+q_3^2\right) & 2\left(q_2q_3+q_1q_4\right) \\ 2\left(q_3q_1+q_2q_4\right) & 2\left(q_3q_2-q_1q_4\right) & 1-2\left(q_1^2+q_2^2\right) \end{bmatrix} \tag{2.8}$$

One important advantage of the use of unit quaternions to parameterise the attitude is that the corresponding kinematic equations do not contain any singularities (Andrle and Crassidis, 2013). A minor disadvantage is the lack of an intuitive physical interpretation. Another disadvantage of the quaternion parameterisation

is the fact that it is non-unique (Walsh et al., 2018). This property results from vector geometry, and states that the rotations $(\boldsymbol{e}, \phi)$ and $(-\boldsymbol{e}, 2\pi - \phi)$ are physically equivalent. Put differently, both $\boldsymbol{q}$ and $-\boldsymbol{q}$ produce the identical rotation mapping (Lysandrou, 2019). When this problematic phenomenon is encountered during an attitude manoeuvre, and an attitude controller commands the larger rotation instead of the (desired) smaller rotation, it is spoken of the *unwinding* phenomenon. Strictly spoken, as this study only considers attitude *guidance*, the unwinding phenomenon is not relevant for this work. However, for the SCP algorithm presented in Chapter 4, the non-uniqueness of quaternions is taken into account to ensure that the obtained guidance trajectories represent the shortest path.

A third problematic characteristic is the unit-norm constraint of the quaternion formulation. Due to the fact that this is a quadratic equality constraint, several optimisation frameworks, for instance, the convex frameworks considered in this study, cannot directly enforce it (Diebel, 2006). If an iterative optimisation method is considered, one strategy to deal with this problem is to perform a re-normalisation of the unit quaternions following each step. This approach, however, does unfortunately not work for the direct optimisation methods that are considered in this study.

The reader is referred to the work of Kuipers (1999) for more information on the quaternion parameterisation, for instance regarding its mathematical properties.

2.2.3. Selection of the quaternion parameterisation

In the literature study that was part of this research project (Diercks, 2021), an extensive analysis of and comparison between the properties of different attitude parameterisations were performed. This subsection contains a brief overview of the conclusions and reasons for selecting the quaternion attitude parameterisation for solving the reorientation problem using an SCP-based optimisation method.

A range of methods is available to parameterise the attitude of a spacecraft, such as Euler angles, (Modified) Rodrigues Parameters, and rotation vectors. For the sequential convex programming framework studied in this project, it was concluded that using the quaternion attitude parameterisation was the most logical choice. There are several reasons for this. To begin with, the quaternion parameterisation does not face the challenge of having singularities. Although solutions have been proposed to deal with this challenge (e.g., reorienting the body-fixed frame (Virgili-Llop et al., 2018) to avoid the regions that have singularities), not having to deal with this challenge simplifies the optimisation algorithm and could (potentially) improve its robustness. Furthermore, its bilinear kinematics (Section 3.2) are only nonlinear in a limited sense. This is helpful as the sequential convex programming framework relies heavily on linearisation techniques. In addition, a convex formulation of the important conical attitude constraints is available for this parameterisation (Section 3.4) (Kim and Mesbahi, 2004), which is not the case for other attitude parameterisations. Finally, earlier studies that apply the convex programming framework to the spacecraft reorientation problem showed that the quaternion-norm constraint can be enforced to a sufficient extent through the linearised norm-preserving kinematic differential equations (Jerez et al., 2017; McDonald et al., 2020; Reynolds et al., 2021).

2.3. Optimal control

Optimal control can be defined as the search for solutions that maximise or minimise certain criteria while satisfying differential constraints on dynamics (the equations of motion) and algebraic constraints on the state and control variables of the problem (Sagliano, 2018). This section introduces the general formulation of an optimal control problem and the methods that can be used to solve such problems.

2.3.1. Problem formulation

The *Bolza* problem is a general representation of an optimal control problem (OCP), and can be formulated as (Sagliano, 2018)

$$\begin{aligned} \text{minimise} \quad & J = \Phi\left[t_f, \boldsymbol{x}\left(t_f\right), \boldsymbol{u}\left(t_f\right)\right] + \int_{t_0}^{t_f} \Psi[\boldsymbol{x}(t), \boldsymbol{u}(t)]\mathrm{d}t \\ \text{subject to} \quad & \dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}, \boldsymbol{u}) \\ & \boldsymbol{g}_L \leq \boldsymbol{g}(t, \boldsymbol{x}, \boldsymbol{u}) \leq \boldsymbol{g}_U \\ & \boldsymbol{x}_L \leq \boldsymbol{x}(t) \leq \boldsymbol{x}_U \\ & \boldsymbol{u}_L \leq \boldsymbol{u}(t) \leq \boldsymbol{u}_U \end{aligned} \tag{2.9}$$

where J is the objective function (known as the cost function or performance index), which consists of the *Mayer* term Φ and an integral term Ψ, which is referred to as the *Lagrange* term. The Lagrange term can be used to optimise certain model elements over the entire problem duration. Moreover, t_0 and t_f represent, respectively, the initial and final time, $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ the state parameters and $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$ the control parameters. $(\cdot)_L$ and $(\cdot)_U$ refer to lower and upper bounds on certain constraints. The control variable $\boldsymbol{u}(t)$ is commonly referred to as the *decision variable*, which is defined as the variable that can be changed in order to arrive at the optimal solution that is desired. As can be seen in Equation 2.9, the problem can be subject to both state and control path constraints $\boldsymbol{g}$.

2.3.2. Numerical solving methods

Due to its complex, nonlinear dynamics, there are no analytic solutions to the optimal spacecraft reorientation problem. Therefore, one has to resort to numerical solving methods to find optimal solutions. The two main classes within this category of numerical solving methods for optimal control problems are *indirect methods* ('optimisation then discretisation') and *direct methods* ('discretisation then optimisation') (Chai et al., 2019). An overview of both of these classes and the main numerical solution methods that they consist of is provided in Figure 2.3.

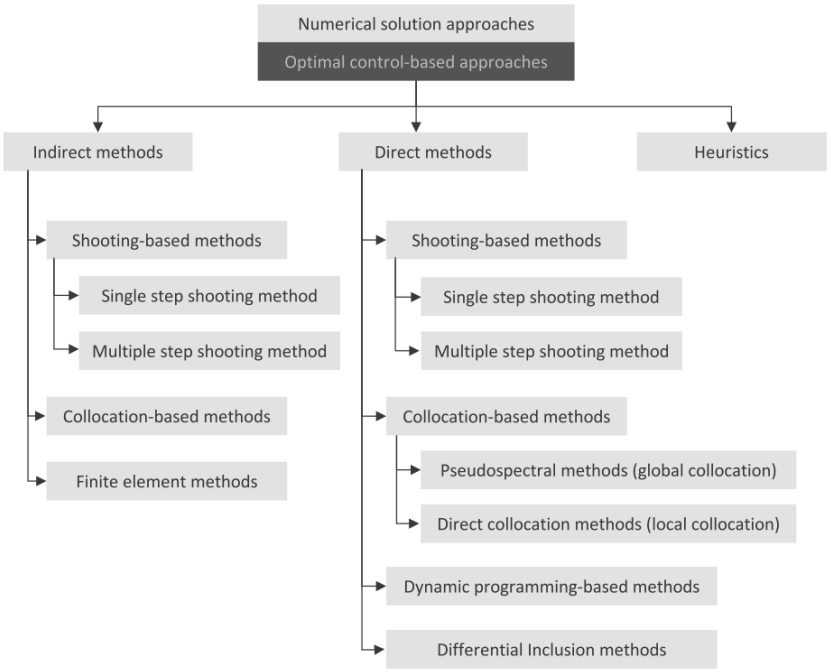


Figure 2.3: Common numerical solution approaches for optimal control problems (Chai et al., 2019).

Indirect methods

Indirect methods make use of elements of optimal control theory, primarily Pontryagin's Maximum Principle (Levskii, 2009; Liberzon, 2007; Pontryagin, 1987), and rely on the derivation of the necessary conditions of optimality, such as transversality conditions and adjoint equations (Lysandrou, 2019). Afterwards, a two-point boundary value problem (TPBVP) is solved. This approach can also be interpreted as trying to find the point

where the slope of the objective function is zero (Kelly, 2015). Indirect methods typically cannot be used to solve the optimal spacecraft reorientation problem considered in this study, due to the complicated, non-linear dynamics and constraints that are involved (Mao et al., 2018a). These cause the necessary conditions to be relatively complicated, which makes it difficult to determine the switch points where the constraints become active (Chai et al., 2019). Furthermore, the resulting TPBVP is known to be extremely sensitive to an initial guess of the solution, which can be difficult or impossible to find (Bonalli et al., 2019).

Direct methods

Direct methods are usually favoured because these methods do not require the derivation and consideration of the necessary conditions of optimality (Betts, 1998), but directly convert the continuous, infinite-dimensional optimisation problem into a discrete, finite-dimensional optimal control problem. This process is commonly referred to as the *transcription* or *discretisation* process (Section 4.5) (Sagliano, 2018). Direct methods parameterise either the control or the state and control with a set of basis functions. These basis functions can be, for instance, piecewise constant, linear, cubic or polynomial functions, and the coefficients of these functions are then obtained through the actual parameter optimisation process. A variety of numerical solving algorithms exists (f.i, nonlinear programming (NLP) methods) that can solve these discrete problems relatively efficiently (Thomson, 1962). As the complexity of an optimisation problem increases, direct methods are typically significantly easier to implement than indirect methods, making them the favoured approach in engineering practice (Chai et al., 2019). However, the NLP methods that are commonly used to solve spacecraft reorientation problems are typically too slow for real-time, onboard applications, as is further discussed in Section 3.8.

2.4. Convex optimisation

In this section, the topics of convex optimisation and convexification-based optimisation techniques are introduced. To this end, Subsection 2.4.1 covers the basics of convex optimisation, while Subsection 2.4.2 introduces a specific type of convex problem that has a special interest for this study. Subsequently, Subsections 2.4.3 and 2.4.4 introduce the two main approaches that are used to convexify non-convex problems.

2.4.1. Introduction

Convex optimisation problems form a class of mathematical optimisation problems that are characterised by convex objective functions, linear equality constraints, and inequality constraints that represent a convex admissible set (Boyd and Vandenberghe, 2004). Figure 2.4 shows examples of the various subclasses that convex programming (CP) consists of: linear programming (LP), quadratic programming (QP), second-order cone programming (SOCP), and semidefinite programming (SDP). Of these subclasses, SOCP holds special interest regarding the convexification-based optimisation method that is developed in this study, and is therefore independently discussed in Subsection 2.4.2.

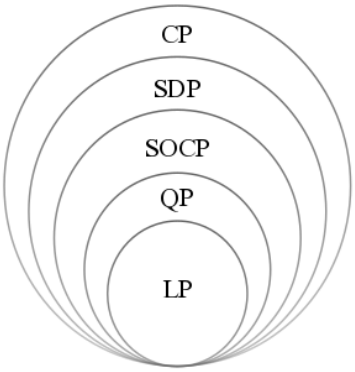


Figure 2.4: A hierarchy of convex optimisation subclasses.[1]

Earlier research has shown that for a variety of optimisation problems, in contrast to popular belief, the most important characteristic is the convexity of the problem, not its linearity (Boyd and Vandenberghe, 2004). The main advantages of convex programming problems in comparison to non-convex problems are:

[1] `https://en.wikipedia.org/wiki/Convex_optimization` (Cited: 23-11-2020).

1. State-of-the-art, very efficient solving algorithms exist that can be used to solve convex programming problems in real time (Wang and Grant, 2018b). The class of primal-dual interior-point methods (IPMs) forms an example of such methods (Mattingley and Boyd, 2012).
2. In contrast to many NLP methods, these convex solving algorithms do not require an initial guess to find a solution, if one exists.
3. Any minimum found by the convex solving algorithm is the global minimum by definition (Hindi, 2004; Sagliano, 2019).
4. On the other hand, if the problem is infeasible, this can be reported in polynomial time using duality theory (Boyd and Vandenberghe, 2004).

A general convex optimisation problem can be formulated as (Boyd and Vandenberghe, 2004; Lysandrou, 2019; Sagliano, 2018)

$$\begin{aligned} &\text{minimise} && f_0(\boldsymbol{x}) \\ &\text{subject to} && f_i(\boldsymbol{x}) \leq a_i, \quad i = 1, \ldots, m \end{aligned} \tag{2.10}$$

where $\boldsymbol{x} \in \mathbb{R}^{n_x}$ is the vector of optimisation parameters and the functions $f_0, \ldots, f_m$ are convex functions. In order to be convex, these functions must satisfy the following relationship (Sagliano, 2018)

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y), \quad \forall \quad \alpha, \beta \geq 0, \quad \alpha + \beta = 1, \quad \alpha, \beta \in \mathbb{R}, \quad i = 0, \ldots, m \tag{2.11}$$

Put differently, in convex optimisation problems, all (in)equality constraints must define a convex set. An intuitive definition of a convex set was provided by Boyd and Vandenberghe (2004), who defined a set as convex 'if every point in the set can be seen by every other point, along an unobstructed straight path between them'. In order to illustrate and visualise the concept of a convex set, some examples of both convex and non-convex sets are provided in Figure 2.5.

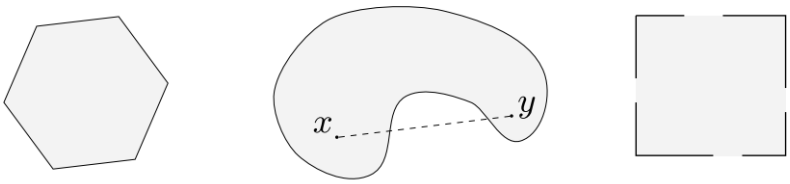


Figure 2.5: Examples of a convex set (left) and two non-convex sets (right) (Hindi, 2004). The thick lines on the sides of the square of the third set are part of the defined set.

Historically seen, convex programming has been out-of-scope for complex optimisation problems, such as the optimal spacecraft reorientation problem, which are characterised by nonlinear, non-convex dynamics and constraints (Liu et al., 2017). However, in recent years, methods and techniques were developed that can reformulate a variety of non-convex problems into a convex form, resulting in extremely fast optimisation algorithms that exploit the power of the available convex IPM solvers. These studies apply either lossless (Subsection 2.4.3) or sequential convex programming techniques (Subsection 2.4.4) to a wide range of (space-related) problems that have challenging sources of non-convexity.

2.4.2. Second-order cone programming

As outlined in the previous subsection, a variety of subclasses of convex programming exist, of which second-order cone programming (SOCP) is of special interest for the convexification-based optimisation methods that are considered in this research project (Açikmeşe and Ploen, 2007; Liu et al., 2017). SOCP problems are characterised by a linear objective function and can be subject to both linear equality and conical inequality constraints (Liu et al., 2015). SOCP problems can be formulated as (Sagliano, 2018)

$$\begin{aligned} &\text{minimise} && \boldsymbol{c}_0^{\mathrm{T}} \boldsymbol{x} \\ &\text{subject to} && A_0 \boldsymbol{x} = \boldsymbol{b}_0 \\ & && \| A_i \boldsymbol{x} + \boldsymbol{b}_i \|_2 \leq \boldsymbol{c}_i^{\mathrm{T}} \boldsymbol{x} + d_i, \quad i = 1, 2, \ldots, p \end{aligned} \tag{2.12}$$

where $\boldsymbol{x}, \boldsymbol{c} \in \mathbb{R}^{n_x}$ represent, respectively, the optimisation variables and the vector that defines the (linear) cost. In addition, $A_0 \in \mathbb{R}^{m \times n_x}$, and $\boldsymbol{b}_0 \in \mathbb{R}^m$ describe a linear system of m equations. Furthermore, $A_i \in \mathbb{R}^{m_i \times n_x}, \boldsymbol{b}_i \in \mathbb{R}^{m_i}, \boldsymbol{c}_i \in \mathbb{R}^{n_x}$ and $d_i \in \mathbb{R}$ represent a conic constraint of order m_i. Figure 2.6 presents a visual example of such a conic constraint with order $m_i = 2$.

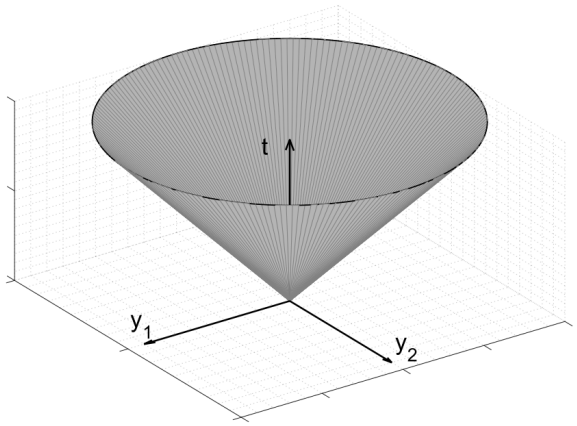


Figure 2.6: Example of 3-D cone. The volume of the cone satisfies the condition $\|A_i \boldsymbol{x} + \boldsymbol{b}_i\|_2 \leq \boldsymbol{c}_i^{\mathrm{T}} \boldsymbol{x} + d_i$ for $m_i = 2$ (Sagliano, 2018).

In line with Figure 2.4, the relation between LP, SOCP, and SDP, which represent three subclasses of CP that are frequently applied to aerospace-related optimisation problems, can be expressed as (Liu et al., 2017)

$$\mathrm{LP} \subseteq \mathrm{SOCP} \subseteq \mathrm{SDP} \tag{2.13}$$

As mentioned, SOCP is of particular interest for real-time onboard guidance applications (Wang and Lu, 2020). The reason for this is that this subclass provides a balance between computational efficiency and modelling power. To illustrate, while LP is typically too inaccurate to model complex guidance problems, the algorithms currently available to solve SDP problems are often not computationally efficient enough for onboard applications. In contrast, SOCP can model more complex problems than LP and can solve problems faster than SDP methods, through the use of state-of-the-art IPMs (Song et al., 2020). An important property of SOCP, which is used extensively in this study, is that convex quadratic constraints can be converted into an SOCP formulation (Boyd and Vandenberghe, 2004; Lobo et al., 1998). This transformation is in accordance with the relation provided in Equation 2.13.

2.4.3. Lossless convexification

The method of lossless convexification was the first method developed to convexify problems that were non-convex in their original form. A lossless convexification method applies one or more convexification techniques in order to produce a fully convex problem from a problem that is non-convex in its original form (Lysandrou, 2019). The convexification process is referred to as *lossless* if it can be proven that the optimal solution of the convexified problem is identical to the optimal solution of the original, non-convex problem (Açikmeşe and Blackmore, 2011). As a result, little or no approximation error is introduced through the convexification procedure, which is a major advantage of this method. Ideally, in the lossless convexification framework, the equivalence of the original and the convexified problem is theoretically proven through the application of Pontryagin's maximum principle (Reynolds et al., 2020b). However, establishing such a proof requires rigorous analysis and is, most likely, not possible for all problem scenarios (Foust et al., 2020). The lossless convexification framework was originally developed to solve the minimum-propellant, planetary soft-landing problem, which is one of the benchmark problems of optimal control theory, in real time (Açikmeşe and Ploen, 2007). In recent years, several guidance algorithms for the powered-descent problem that are based on lossless convexification, have been experimentally validated in flight experiments (Açikmeşe et al., 2013; Scharf et al., 2014, 2017).

However, it is frequently stated in literature that lossless convexification is not applicable to problems with highly nonlinear dynamics and path constraints. To illustrate, Wang et al. (2019a) argued that this is the case for the nonlinear hypersonic entry problem. Several studies that applied convexification-based optimisation algorithms to the spacecraft attitude reorientation problem confirmed the expectation that the lossless

convexification framework cannot be applied to the spacecraft reorientation problem (Jerez et al., 2017; McDonald et al., 2020; Reynolds et al., 2021).

2.4.4. Sequential convex programming

The *sequential convex programming* (SCP) framework, also known as the *successive convexification* (SC) framework, was developed with the aim to rapidly solve optimal control problems with nonlinear dynamics and non-convex constraints that render a lossless convexification into a single convex problem impossible (Lysandrou, 2019). Instead, SCP-based optimisation methods transform a single, non-convex problem into a sequence of multiple convex sub-problems (Wang et al., 2019a). Each of these sub-problems is a convex approximation of the original, non-convex problem. This approximation is always based on a reference solution, which is in practice the solution that was obtained during a previous iteration of the sequential scheme (Szmuk and Açıkmeşe, 2018). As a result, using a poor first guess that is sub-optimal or even dynamically infeasible, an SCP-based optimisation algorithm approaches an optimal solution in a step-by-step manner, as is shown in Figure 2.7. Consequently, the SCP framework can be described as an iterative, local optimisation framework (Wang and Lu, 2020). Although it cannot be guaranteed that its guidance solution is globally optimal, fast local methods are often preferred over slow global methods for a variety of real-time, autonomous applications of optimal guidance algorithms (Lysandrou, 2019).

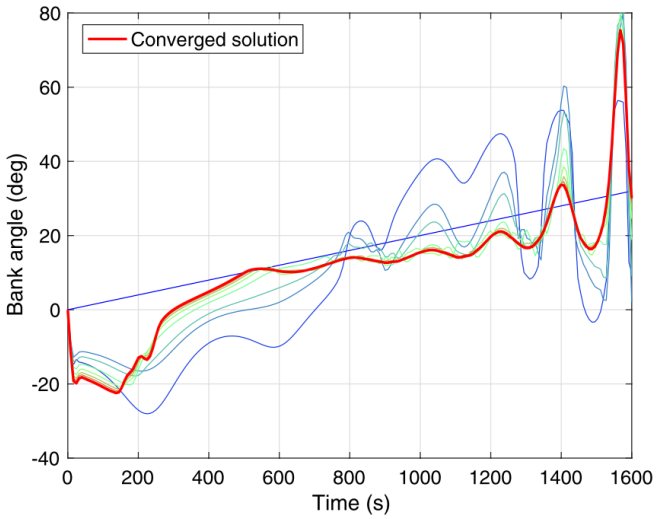


Figure 2.7: Example of the convergence process of an SCP algorithm. The initial reference solution (straight blue line) is iteratively updated (progressively changing colour from blue to red) until a converged, locally optimal solution (red line) is obtained. In this example, the figure shows a control (bank angle) profile as a function of time for the hypersonic entry problem (Wang and Lu, 2020).

The SCP framework was first developed in studies of Lu and Liu (2013) and Liu and Lu (2014). Since then, SCP-based optimisation methods have been frequently studied and have been reported to show promising performance in terms of computational efficiency (Sagliano, 2018). At the moment of writing, SCP-based optimisation methods have been applied to a wide range of optimisation problems in aerospace-related research fields. Collectively, the corresponding studies provide numerical evidence of the promising properties of this class of optimisation methods (Liu et al., 2017). As an example, in a study of Malyuta et al. (2019), convergence rates of 100% were obtained in a Monte Carlo analysis. In Figure 2.8, an illustration is provided of a typical convergence history of an SCP algorithm, in this case for the 6 degree-of-freedom (DoF) powered descent problem. In this figure, the dots represent the discretisation nodes of the numerical optimisation problem, and the lines represent trajectories that were obtained by propagating the original, nonlinear dynamics using the control history that was obtained for the corresponding iteration of the iterative scheme. It can be observed that the linearisation error decreases with subsequent iterations.

To further illustrate, SCP-based optimisation methods have, amongst others, been applied to the rendezvous problem (Benedikter et al., 2019b; Lu and Liu, 2013), low-thrust orbital transfer problem (Bergin et al., 2020; Wang and Grant, 2018b), the planetary entry problem (Liu et al., 2015; Wang and Grant, 2016; Wang and Lu, 2020; Zhou et al., 2020), the planetary soft-landing problem (Mao et al., 2017; Sagliano, 2019), the swarm

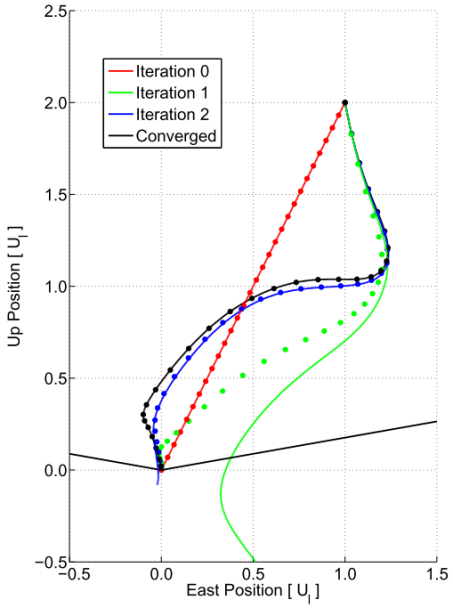


Figure 2.8: Convergence history of an SCP algorithm for a powered descent optimisation problem (Mao et al., 2018a). The black, straight lines represent a conical constraint above which the vehicle must remain during the descent manoeuvre.

trajectory optimisation problem (Morgan et al., 2016), and quadrotor motion planning problems (Szmuk et al., 2017, 2019; Wang et al., 2018). In these and other studies, a variety of different algorithm features and techniques have been developed to increase the performance of the algorithm, predominantly in terms of robustness and computational efficiency.

In recent years, the first steps have been taken to turn SCP-generated optimal trajectories into a guidance framework that can be implemented in real-world scenarios. For instance, an optimal feedback guidance law was designed to track an optimal reference trajectory that is generated online using SCP (Wang and Grant, 2018a), and the real-time capability of an SCP method was tested in quadrotor flight experiments in the study of Szmuk et al. (2017). The SCP framework has only been applied to the optimal spacecraft reorientation problem in a small number of studies. Characteristics and results of these studies are provided in Section 3.8.

One should not confuse the SCP framework with the more mature sequential quadratic programming (SQP) optimisation methods, which are used by many NLP solvers and typically require much more computational effort to find locally optimal solutions (Szmuk et al., 2020). The main cause of this is that SQP methods require second-order information when approximating the Hessian of the constraints that are imposed on the problem, which requires the use of computationally demanding techniques such as the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update (Boggs and Tolle, 1995).

3

Optimal Spacecraft Reorientation

In this chapter, the optimal spacecraft reorientation problem that is considered in this research project is introduced, formulated and discussed. In Sections 3.1 and 3.2, the equations of motions of the spacecraft reorientation problem are introduced. Afterwards, Sections 3.3, 3.4 and 3.5 introduce, respectively, the boundary conditions, constraints and objectives of the reorientation problem. In Section 3.6, an overview of two specific problems is provided, containing all elements from the previous sections. Then, Section 3.7 explores which requirements an (ideal) reorientation guidance algorithm should fulfil to be eligible for onboard implementation. Finally, in Section 3.8, a brief overview of the previous research into real-time optimisation of spacecraft reorientation manoeuvres is provided, including an analysis of studies that are based on the sequential convex programming framework.

3.1. Attitude dynamics

Euler's rotational equations of motion form the foundation for describing the physical behaviour of a spacecraft during a reorientation manoeuvre. These equations can be represented as (Wertz, 1978)

Dynamic equations

$$\boldsymbol{I}\,\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\,\boldsymbol{\omega} = \boldsymbol{u} \tag{3.1}$$

where $\boldsymbol{I} \in \mathbb{R}^{3\times3}$ represents the inertia matrix of the spacecraft, $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^T$ is the angular velocity vector (angular rate vector) of the body-fixed reference frame F_B relative to the inertial frame F_I, and $\boldsymbol{u} = [u_1, u_2, u_3]^T$ is the control (torque) vector. All problem elements that are introduced in this chapter that are included in the formulation of the reorientation problems that were studied in this research project (Section 3.6), are shown using a grey box for the sake of clarity.

Other formulations of the spacecraft dynamics are commonly used in literature. For instance, when the body-fixed frame is aligned with the principal moments of inertia of the spacecraft, the attitude dynamics can be represented in a simplified form as (Wertz, 1978)

$$\begin{aligned} I_1\dot{\omega}_1 - (I_2 - I_3)\,\omega_2\omega_3 &= u_1 \\ I_2\dot{\omega}_2 - (I_3 - I_1)\,\omega_3\omega_1 &= u_2 \\ I_3\dot{\omega}_3 - (I_1 - I_2)\,\omega_1\omega_2 &= u_3 \end{aligned} \tag{3.2}$$

where I_i is the rigid-body principal moment of inertia around axis i, for $i = 1,2,3$. It must be emphasised that $\boldsymbol{\omega}$ describes the angular motion of the rotating, body-fixed reference frame F_B. Therefore, one cannot describe the orientation of the spacecraft in the inertial frame F_I using only these rotational dynamic equations. In Section 3.2, the kinematic relations are provided that relate these body-relative rotations to attitude changes with respect to the inertial frame (Thomson, 1962). Another formulation of the spacecraft dynamics considers the situation where a spacecraft is equipped with reaction or momentum wheels. Such a body is in reality not a rigid body, as Equation 3.1 assumes. As a result, Equation 3.1 is only valid in this situation with a

minor modification (Wertz, 1978), which can be formulated as

$$\boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega} + \boldsymbol{h}) + \dot{\boldsymbol{h}} = \boldsymbol{u} \tag{3.3}$$

where $\boldsymbol{h}$ is the reaction wheel angular momentum, relative to the spacecraft. However, in most studies on real-time attitude trajectory optimisation (Section 3.8), either Equation 3.1 or Equation 3.2 is used to describe the spacecraft dynamics. For this study, it was decided, together with the research partner OHB, to make the rigid-body assumption and use the dynamic equations of Equation 3.1. The first reason for this is that Euler's rotational equations are more often encountered in literature than the simplified form of Equation 3.2 or the extended form of Equation 3.3. Furthermore, as OHB typically uses these equations of motion, this choice could result in a higher usability of the research outcome, and enable easier comparisons to the optimisation techniques that are currently being used.

3.2. Attitude kinematics

The study of kinematics concerns the motion of particles and rigid bodies without consideration of the moments and forces that are acting on these particles and bodies (Diebel, 2006). In Subsection 2.2.2, the quaternion attitude parameterisation was introduced that will be used during this study. In this section, the corresponding kinematic differential equations are provided. These kinematics describe changes in the attitude of a spacecraft as a function of time and the angular velocity, which is expressed in the body-fixed frame F_B. The kinematic differential equations for the quaternion attitude parameterisation can be formulated as (Kim and Mesbahi, 2004)

Kinematic equations

$$\dot{\boldsymbol{q}} = \frac{1}{2}\,\Omega\,\boldsymbol{q} \tag{3.4}$$

where Ω is the skew-symmetric matrix

$$\Omega = \begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} \tag{3.5}$$

As mentioned in Subsection 2.2.3, no singularities are present in the kinematic equations of Equation 3.5, which is one of the main advantages of using the quaternion attitude parameterisation. The differential equations from Equation 3.4 are norm-preserving, which means that the quaternion unit norm (Equation 3.6) is preserved as long as the kinematics are properly enforced (Kim and Mesbahi, 2004).

$$\|\boldsymbol{q}\|_2 = 1, \quad \text{for} \quad t \in \left[t_0, t_f\right] \tag{3.6}$$

Another advantageous property of these quaternion kinematic differential equations is the fact that they are bilinear (linear with respect to each of two sets of mathematical variables, in this case $\boldsymbol{q}$ and $\boldsymbol{\omega}$) (Schaub and Junkins, 2009). This bilinearity is beneficial for problems where these differential equations are linearised, which is the case for the SCP-based optimisation algorithm developed in this study (Lysandrou, 2019).

3.3. Boundary conditions

The solution space of the optimal spacecraft reorientation problem can be restricted in various ways. In this study, a distinction is made between boundary conditions and constraints (Section 3.4). Boundary conditions impose a set of constraints on the optimisation problem at specified boundaries, while the constraints presented in Section 3.4 are enforced continuously during a manoeuvre. Regarding the the optimal reorientation problem, boundary conditions are used to specify the initial and final states of a reorientation manoeuvre. Two different types of boundary conditions were considered in this research project: time-independent boundary conditions and time-dependent boundary conditions. Both are discussed in, respectively, Subsections 3.3.1 and 3.3.2. It is noted that the time-dependent boundary conditions can only be used in free-final-time problems, for which the final time t_f is an optimisation parameter.

3.3.1. Time-independent boundary conditions

With *time-independent* boundary conditions, boundary conditions are referred to that are independent of time and have a constant value. These boundary conditions can be defined for all state ($\boldsymbol{q}$ and $\boldsymbol{\omega}$) and control ($\boldsymbol{u}$) variables. In this study, however, no boundary conditions were imposed on the control.

Time-independent boundary conditions

$$\boldsymbol{q}(0) = \boldsymbol{q}_0, \quad \boldsymbol{q}(t_f) = \boldsymbol{q}_f \tag{3.7}$$

$$\boldsymbol{\omega}(0) = \boldsymbol{\omega}_0, \quad \boldsymbol{\omega}(t_f) = \boldsymbol{\omega}_f \tag{3.8}$$

As these constraints take the form of linear equality constraints, they are convex and can be included in the SCP framework in their standard form (Section 4.10).

3.3.2. Time-dependent boundary conditions

In the problem specified by the research partner OHB, the optimisation problem is characterised by so-called *time-dependent* final boundary conditions. In practice, this type of boundary condition typically originates from observation targets of some satellite instrument. These targets are often moving relative to the spacecraft-centred inertial frame F_I as a function of time. As stated before, these boundary conditions can only be imposed on free-final-time problems, such as the minimum-time problem or a free-final-time, minimum-energy problem. In this study, time-dependent boundary conditions are implemented for the minimum-time reorientation problem. These boundary conditions can be formulated as

Time-dependent boundary conditions

$$\boldsymbol{q}(t_f) = \boldsymbol{p}_{q_f}(t_f) \tag{3.9}$$

$$\boldsymbol{\omega}(t_f) = \boldsymbol{p}_{\omega_f}(t_f) \tag{3.10}$$

Boundary conditions of this type, $\boldsymbol{p}_{q_f}(t_f)$ and $\boldsymbol{p}_{\omega_f}(t_f)$, are, in industry, typically either computed analytically or represented using high-order Lagrange polynomials. For the purpose of this study, it was decided to use third-order Lagrange polynomials to represent these problem boundaries. The reasoning behind this is that using third-order polynomials suffices to prove that the SCP framework could be able to incorporate time-dependent boundary conditions in the form of non-convex, nonlinear equality constraints. To illustrate, the final quaternion boundary condition is represented by a third-order polynomial as

$$\boldsymbol{p}_{q_f}(t_f) = \boldsymbol{p}_{q_f,c,1} + \boldsymbol{p}_{q_f,c,2} \cdot t_f + \boldsymbol{p}_{q_f,c,3} \cdot t_f^2 + \boldsymbol{p}_{q_f,c,4} \cdot t_f^3 \tag{3.11}$$

The four coefficients $\boldsymbol{p}_{q_f,c,1}$ to $\boldsymbol{p}_{q_f,c,4} \in \mathbb{R}^4$ thereby form the input variables that are used to specify the final attitude boundary conditions of the minimum-time optimisation problem.

3.4. Constraints

During this research project, three common constraint types were considered: constraints on control, angular rate and the attitude of the spacecraft. The latter can be further split up into two main types: attitude *keep-out* and attitude *keep-in* constraints. The resulting four types of constraints are formulated and presented in Subsections 3.4.1 to 3.4.4, respectively.

3.4.1. Control constraints

Control constraints are always present in attitude guidance problems because of the physical limitations of the spacecraft attitude control system. The most common (and relatively straightforward) formulation of the control constraints takes the shape of an axis-aligned cuboid (Reynolds et al., 2021). This constraint type is in this study referred to as the so-called *box* constraint on the control, and can be formulated as

$$|u_i| \leq u_{\text{box},i}, \quad \forall \quad i \in 1,2,3, \quad t \in \left[t_0, t_f\right] \tag{3.12}$$

Other common control constraints typically describe convex shapes that form a boundary for the control vector in 3-D space. Together with research partner OHB, it was decided to implement an ellipsoidal constraint on the control to study the capabilities and performance of the SCP optimisation algorithm in this regard. This ellipsoidal constraint on the control can be formulated as

Ellipsoidal control constraint

$$\left(\frac{u_1}{u_{\text{ellipse},1}}\right)^2+\left(\frac{u_2}{u_{\text{ellipse},2}}\right)^2+\left(\frac{u_3}{u_{\text{ellipse},3}}\right)^2\leq 1,\quad \text{for}\quad t\in\left[t_0,t_f\right] \tag{3.13}$$

A large range of alternative approaches to constrain the control torque exist. For instance, one could use either the L_1-norm or L_2-norm. The latter, for illustration purposes, is formulated as

$$\|\boldsymbol{u}\|_2\leq u_{\max},\quad \text{for}\quad t\in\left[t_0,t_f\right] \tag{3.14}$$

Both these options are convex, and can therefore be incorporated into a convex optimisation framework (Tam and Glenn Lightsey, 2016).

Depending on the attitude control system of a specific satellite, the true 3-D volume of the control authority can be even more complex than the constraints of Equations 3.12 or 3.13. However, even in these scenarios, the control constraints are often modelled as simpler shapes in order to optimise computational efficiency. For instance, in the study of Reynolds et al. (2021), a reaction-wheel array is considered that consists of four wheels, which would require the enforcement of 12 half-space constraints to obtain the theoretical maximum control volume. In that study, the simpler, more conservative box constraint (Equation 3.12) is implemented to limit the number of constraints and, thereby, enhance the computational performance of the optimisation process.

3.4.2. Angular rate constraints

Constraints that impose limits on the angular velocity of a spacecraft are also known as (slew) rate constraints. These types of constraints are typically imposed on optimal spacecraft reorientation problems to ensure that so-called *rate-limited* sensors keep functioning properly. These are sensors with maximum slew-rate constraints. Examples of rate-limited sensors are star trackers and sun sensors, which both have a bounded slew-rate envelope to ensure that they are able to provide adequate attitude estimates to the attitude determination system of a spacecraft. In addition, rate constraints could be introduced to the problem of preventing reaction-wheel saturation (Aucoin and Zee, 2019). The most general form of this constraint can be represented in a component-wise fashion as

Box rate constraint

$$|\omega_i|\leq\omega_{\text{box},i},\quad \forall\quad i\in 1,2,3,\quad t\in\left[t_0,t_f\right] \tag{3.15}$$

In consultation with the research partner OHB, the decision was made to study the box constraint on the angular rate during this research project, but as for the control constraints, other practises exist. Examples could be the enforcement of an L_2-norm or L_∞-norm (largest value in a vector). For an actual satellite, it typically depends on the requirements of the rate-limited sensors which constraint type is chosen to impose a limit on the angular rate (Reynolds et al., 2021).

3.4.3. Attitude keep-out constraints

Constraints on the permissible attitude of a satellite (or in general terms, path constraints) are common in literature on the spacecraft reorientation problem. These constraints typically take the form of exclusion cones (Eren et al., 2015). Such an exclusion constraint could, for instance, follow from the need to protect certain sensitive observation instruments, such as star trackers or cryogenically cooled telescopes (Walsh et al., 2018), from the radiation of the Sun. A specific example of such a constraint is the 85 deg attitude exclusion cone of the James Webb Space Telescope (JWST) towards the Sun (Wolfe and Osten, 2014). Such exclusion cones are commonly referred to as attitude *keep-out* constraints or attitude *forbidden zones* (Aucoin and Zee, 2019). In

Figure 3.1, a visual representation of a conical exclusion cone and two possible reorientation trajectories is provided, as seen from two different perspectives.

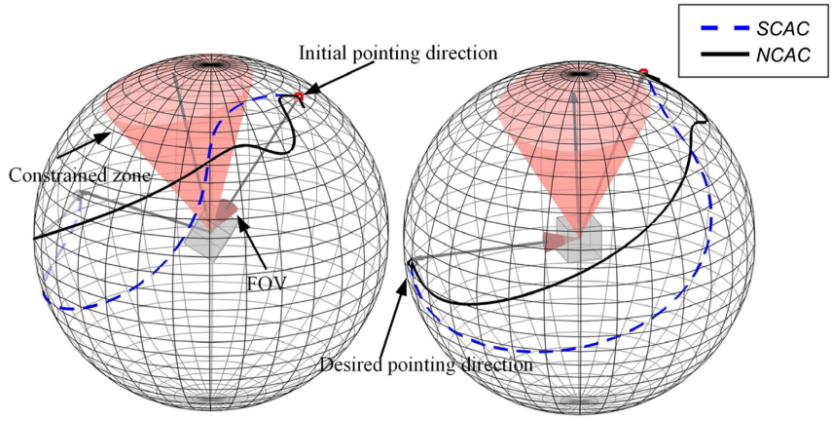


Figure 3.1: Example of two possible reorientation trajectories (referred to as NCAC and SCAC in the figure) around a conical attitude exclusion cone (red cone), as seen from two different perspectives (Hu et al., 2020). Showing the visualisation from two different perspectives allows the reader to observe the entire reorientation manoeuvre.

Figure 3.2 provides an alternative, 2-D representation of the same reorientation manoeuvres and exclusion cone that were shown in Figure 3.1.

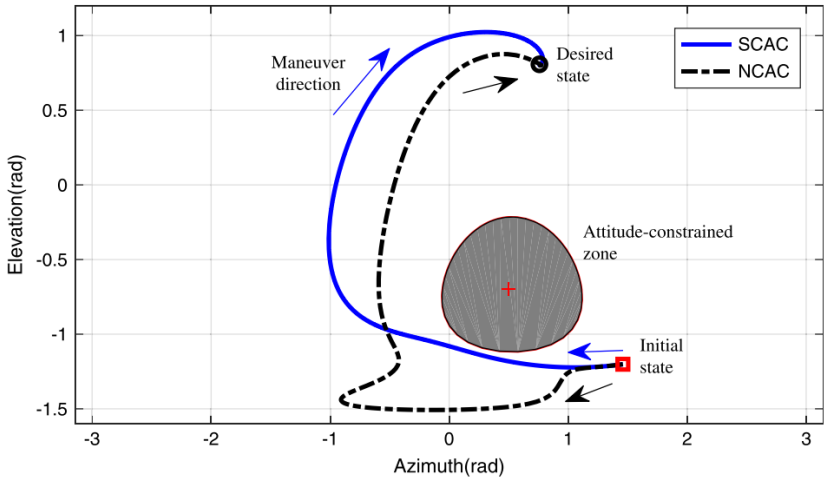


Figure 3.2: Two-dimensional, cylindrical projection of the two reorientation trajectories (SCAC and NCAC) and exclusion cone of Figure 3.1 (Hu et al., 2020).

In literature, several types of keep-out constraints can be encountered that are either static or dynamic, and either hard or soft (Kim et al., 2010). However, in this study, the most common type is considered, which is the so-called *static, hard* attitude keep-out constraint. For this constraint, the central vector that determines the orientation of the exclusion cone $\boldsymbol{y}_I$ is (assumed to be) inertially fixed, or *static*. This is generally the case when a distant object should be avoided. Another example is the protection of important equipment against incoming space debris. Furthermore, this type of constraint allows for no constraint violations, which is the reason why it is described as a *hard* constraint. The static, hard exclusion constraint can be represented as

Attitude keep-out constraint

$$\boldsymbol{x}_I(t)^T \boldsymbol{y}_I \leq \cos\left(\theta_{\text{sep,out}}\right), \quad \theta_{\text{sep,out}} \in [0, \pi] \tag{3.16}$$

where $\boldsymbol{x}_I(t)$ and $\boldsymbol{y}_I$ are unit vectors in the inertial frame F_I that represent the direction of the boresight of the instrument and the celestial object that should be avoided, respectively, and $\theta_{\text{sep,out}}$ is the required minimum

angular separation between the boresight and the object (Kim et al., 2010).

3.4.4. Attitude keep-in constraints

Similar to the attitude keep-out constraint introduced in the previous subsection, another common constraint type is the so-called attitude *keep-in* constraint. In conical form this constraint is also referred to as an attitude *inclusion cone* or *attitude mandatory zone* (Aucoin and Zee, 2019). An attitude keep-in constraint could result from, for instance, a requirement to keep a constant communication link with a specific target area on Earth during a certain reorientation manoeuvre (Frazzoli et al., 2001). Another example could be the need for a sensor to continuously keep the Sun in its field of view (FoV) for navigation purposes (Tam and Glenn Lightsey, 2016). An attitude keep-in constraint can be formulated as

Attitude keep-in constraint

$$\boldsymbol{x}_I(t)^T \boldsymbol{y}_I \geq \cos\left(\theta_{\text{sep,in}}\right), \quad \theta_{\text{sep,in}} \in [0, \pi] \tag{3.17}$$

where, again, both $\boldsymbol{x}_I(t)$ and $\boldsymbol{y}_I$ are defined in the inertial reference frame and $\theta_{\text{sep,in}}$ is the *maximum* separation angle.

3.5. Objective function

Multiple objective functions can be formulated for the optimal spacecraft reorientation problem. For this study, it was decided to focus on two of the most important and commonly used ones: the minimum-energy and the minimum-time objectives.

3.5.1. Minimum energy

This objective aims to keep the energy consumption of a specified attitude manoeuvre as low as possible. This is highly desirable due to the extremely high cost associated with transporting propellant to and generating energy in space. The following expression is commonly used for the minimum-energy problem (Ventura et al., 2015):

Minimum-energy objective function

$$J_u = w_u \int_{t_0}^{t_f} \frac{1}{2} \|\boldsymbol{u}\|_2 \mathrm{d}t \tag{3.18}$$

where w_u is a positive weight factor that is defined by the user, which becomes important when an augmented objective function takes multiple cost terms into account (which is the case for the SCP algorithm presented in Chapter 4). Furthermore, the factor $\frac{1}{2}$ is often omitted as it leads to an equivalent solution.

3.5.2. Minimum time

Minimum-time reorientation manoeuvres lead to improvements in the effective mission performance of a spacecraft. To begin with, the resulting guidance trajectories can, for instance, increase the amount of data an Earth observation satellite can collect in a given period of time (Ventura et al., 2015). Another application could be a scientific mission that aims to capture a certain astronomical event of which the time of occurrence is difficult to predict. The objective of this type of spacecraft reorientation problem is to minimise the manoeuvre time t_f (assuming $t_0 = 0$ s) that is needed to complete a manoeuvre between two spacecraft states. The corresponding term in the objective function can be expressed as (Boyarko et al., 2011)

Minimum-time objective function

$$J_t = w_t \int_0^{t_f} \mathrm{d}t = w_t \cdot t_f \tag{3.19}$$

where w_t, similar to w_u in Equation 3.18, is a user-defined weight factor that specifies the relative importance of the minimum-time term in an augmented objective function. Bilimoria and Wie (1993) showed that for a

symmetric body with constraints on all control variables, the minimum-time, rest-to-rest spacecraft reorientation problem has an optimal solution that is a so-called non-eigenaxis manoeuvre. Generating minimum-time trajectories requires a free-final-time formulation of the problem (see Section 4.3), in contrast to the fixed-final-time formulation that can be used if the exact duration of the manoeuvre is known. The former has a larger degree of freedom, typically making it more difficult to solve robustly (Lysandrou, 2019).

3.6. Non-convex problem formulation

In this section, the elements of the spacecraft reorientation problem of the previous subsections have been combined into two main problems that are studied during this research project: the minimum-energy and minimum-time spacecraft reorientation problems. The minimum-energy problem is easier to implement and was studied in literature using an SCP-based approach before the start of this project, while the minimum-time problem is more challenging to implement and was not yet studied. These problems are both non-convex and continuous. In Chapter 4, both are convexified and discretised in order to be solvable in the SCP framework. Both problems are very comparable at this stage. However, in Chapter 4, more differences are introduced. Because of that, it was decided to present both problems independently in this section for the sake of consistency.

3.6.1. Problem 1: minimum-energy reorientation

The non-convex, continuous, minimum-energy spacecraft reorientation problem studied in this research project can be formulated in its most general way as

Non-convex, continuous, minimum-energy spacecraft reorientation problem

Objective function	minimise $J = w_u \int_{t_0}^{t_f} \|\boldsymbol{u}\|_2 \mathrm{d}t$	Equation 3.18
Boundary conditions	$\boldsymbol{q}(0) = \boldsymbol{q}_0, \quad \boldsymbol{q}(t_f) = \boldsymbol{q}_f$	Equation 3.7
	$\boldsymbol{\omega}(0) = \boldsymbol{\omega}_0, \quad \boldsymbol{\omega}(t_f) = \boldsymbol{\omega}_f$	Equation 3.8
Dynamics and kinematics	$\boldsymbol{I}\,\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\,\boldsymbol{\omega} = \boldsymbol{u}$	Equation 3.1
	$\dot{\boldsymbol{q}} = \frac{1}{2}\Omega \boldsymbol{q}$	Equation 3.4
Control constraint	$\left(\frac{u_1}{u_{\text{ellipse},1}}\right)^2 + \left(\frac{u_2}{u_{\text{ellipse},2}}\right)^2 + \left(\frac{u_3}{u_{\text{ellipse},3}}\right)^2 \leq 1$	Equation 3.13
Angular rate constraint	$\lvert\omega_i\rvert \leq \omega_{\text{box},i}, \quad \forall \quad i \in 1,2,3$	Equation 3.15
Attitude constraints	$\boldsymbol{x}_I(t)^T \boldsymbol{y}_I \leq \cos\left(\theta_{\text{sep,out}}\right)$	Equation 3.16
	$\boldsymbol{x}_I(t)^T \boldsymbol{y}_I \geq \cos\left(\theta_{\text{sep,in}}\right)$	Equation 3.17

3.6.2. Problem 2: minimum-time reorientation

The non-convex, continuous, minimum-time spacecraft reorientation problem studied in this research project can be formulated in its most general way as

Non-convex, continuous, minimum-time spacecraft reorientation problem

Objective function	minimise $J = w_t \cdot t_f$	Equation 3.19
Boundary conditions	$\boldsymbol{q}(0) = \boldsymbol{q}_0, \quad \boldsymbol{q}(t_f) = \boldsymbol{p}_{q_f}(t_f)$	Equation 3.9
	$\boldsymbol{\omega}(0) = \boldsymbol{\omega}_0, \quad \boldsymbol{\omega}(t_f) = \boldsymbol{p}_{\omega_f}(t_f)$	Equation 3.10
Dynamics and kinematics	$\boldsymbol{I}\,\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\,\boldsymbol{\omega} = \boldsymbol{u}$	Equation 3.1
	$\dot{\boldsymbol{q}} = \frac{1}{2}\Omega\boldsymbol{q}$	Equation 3.4
Control constraint	$\left(\frac{u_1}{u_{\text{ellipse},1}}\right)^2 + \left(\frac{u_2}{u_{\text{ellipse},2}}\right)^2 + \left(\frac{u_3}{u_{\text{ellipse},3}}\right)^2 \leq 1$	Equation 3.13
Angular rate constraint	$\lvert\omega_i\rvert \leq \omega_{\text{box},i}, \quad \forall \quad i \in 1,2,3$	Equation 3.15
Attitude constraints	$\boldsymbol{x}_I(t)^T \boldsymbol{y}_I \leq \cos\left(\theta_{\text{sep,out}}\right)$	Equation 3.16
	$\boldsymbol{x}_I(t)^T \boldsymbol{y}_I \geq \cos\left(\theta_{\text{sep,in}}\right)$	Equation 3.17

3.7. Algorithm requirements

Before it is possible to develop and thoroughly analyse an SCP-based optimisation algorithm for solving the optimal spacecraft reorientation problem, it is important to understand which requirements such an algorithm should fulfil in the first place. In this section, these requirements are presented and discussed.

It is noted that the objective of this study is to get a general understanding of the capabilities of an SCP-based optimisation algorithm, not to design such an algorithm for a specific mission scenario. Therefore, it was decided **not** to define strict, quantitative requirements in the manner that would be expected when designing a system. Rather, this section identifies the potential requirements in a general sense in order to determine the focus areas for the analyses that are performed in this study. These requirements were fundamental for the formulation of the research (sub-)questions in Section 1.3.

REQ-1 **Computational efficiency**
A high computational efficiency (or speed) is one of the most important requirements for an onboard attitude guidance algorithm (Bonalli et al., 2019). This requirement is specifically important for onboard satellite guidance applications, as the computational resources of a spacecraft are often limited. In this study, computational efficiency is evaluated through the run times of the SCP algorithm. The customer OHB mentioned that run times in the order of tens of milliseconds are desired when using a personal computer. This means that the SCP algorithm should be multiple factors faster than the NLP optimisation methods that are currently used.

REQ-2 **Robustness**
In the context of this study, the robustness of the SCP algorithm is directly evaluated through its convergence rate. An (onboard) optimisation algorithm should be reliable and deliver predictable performance for a wide range of potential problems. OHB has mentioned that, ideally, the convergence rate of the SCP algorithm should match the convergence rate of conventional pseudospectral NLP methods. It was considered to be out of scope for this research project to provide *theoretical guarantees* for convergence, so the evaluation of the convergence rate was performed through extensive Monte Carlo analyses.

REQ-3 **Optimality**
The SCP algorithm should generate optimal guidance trajectories as this study considers the *optimal* spacecraft reorientation problem. This optimality requirement is evaluated on the basis of the manoeu-

vre duration for the minimum-time problem, and the energy cost of the manoeuvre for the minimum-energy problem. As there is no objective reference point for evaluating the optimality of the SCP algorithm, its solutions were compared to the solutions of an NLP benchmark algorithm, as pseudospectral NLP methods are widely regarded as the standard for finding locally optimal solutions to complex optimisation problems. It is desired that the SCP algorithm has comparable performance as the NLP algorithm in terms of optimality.

REQ-4 **Generality**
In order to be suitable for a variety of onboard attitude guidance applications, the SCP algorithm should be able to solve a range of different types of spacecraft reorientation problems.

REQ-5 **Accuracy**
The guidance trajectories that are obtained using the SCP algorithm should be consistent with the real-world spacecraft dynamics and kinematics. As this accuracy requirement strongly depends on the exact mission requirements, which are not specified for this study, the goal regarding this requirement was to obtain a better understanding of the level of accuracy that an SCP algorithm could achieve.

REQ-6 **Constraint satisfaction**
The SCP algorithm should satisfy the constraints that are imposed on the optimisation problem. Constraint violations could lead to a variety of problems for a space mission, ranging from loss of communication to damaged observation instruments.

This study primarily focused on evaluating the performance of the SCP algorithm on the first three items of this list of requirements, as these are considered to be the most essential criteria for potential onboard applications. This is also reflected in the research (sub-)questions presented in Section 1.3. The fourth requirement is incorporated into this study through the extension of the problem formulation according to the first research sub-question. Regarding the final two requirements, the performance results of the SCP algorithm will be provided but not evaluated against a reference value. The reason for this is that the exact levels that are required for these criteria vary strongly per mission scenario. In Chapter 6, more information is provided on how these criteria were expressed and measured.

3.8. Heritage

In this section, a brief overview is provided of previous optimisation methods that were developed and proposed to solve the optimal spacecraft reorientation problem in real time. While Subsection 3.8.1 presents a variety of methods that rely on different theoretical approaches, Subsection 3.8.2 covers all previous studies that applied the sequential convex programming framework to the spacecraft reorientation problem.

3.8.1. Real-time, optimal spacecraft reorientation

In Table 3.1, a general overview is provided of methods that were developed in literature to solve the real-time, optimal spacecraft reorientation problem. Table 3.1 is adopted from the literature study that was performed for this MSc thesis research project (Diercks, 2021). It is noted that in these studies, different types of reorientation problems were solved, differing in both their objectives and constraints. Nevertheless, the overview of Table 3.1 can be used to obtain a general understanding of the strengths and weaknesses of different optimisation methods. The scoring of the performance of the algorithms in terms of computational efficiency (speed), robustness and optimality serves an illustrative purpose. These scores do not directly relate to specified efficiency, convergence rate or overcost metrics, as these are often not provided and strongly depend on a large range of factors (computation platform, problem type, etc.), which makes a fair quantitative comparison impossible.

The main conclusion that can be drawn from Table 3.1 is that current optimisation methods that show promising performance in terms of computational efficiency either exclusively solve simplified problems (e.g., no minimum-time objective or attitude constraints) or lack performance in terms of robustness or optimality. This confirms the need for a novel optimisation method that can fulfil all, or at least the majority of the criteria identified in Section 3.7. The promising performance evaluation of SCP-based optimisation methods is based on previous applications to the optimal spacecraft reorientation problem and to other space-related optimisation problems (see Subsection 2.4.4).

Table 3.1: A (non-exhaustive) overview of the most relevant optimisation methods that have been developed for the (real-time) optimal spacecraft reorientation problem. As explained in this section, this table serves an illustrative purpose in order to create an overview of the capabilities of a wide range of methods that have been applied in literature. The promising characteristics of the SCP-based methods have partly been based on results from other space-related problems from Section 2.4. This table was adopted from (Diercks, 2021).

Method	**Reference Studies (not exhaustive)**	**Minimum-time**	**Attitude constraints**	**Control constraints**	**Speed**	**Robustness**	**Optimality**	**Comments**
Nonlinear programming	Boyarko et al. (2011), Karpenko et al. (2011)	yes	yes	yes	-	+	+	Reference method.
Geometric	Xu et al. (2018), Xu et al. (2020)	no	yes	yes	+-	+	-	Not optimal.
Potential function-based	McInnes (1994), Hu et al. (2020)	no	yes	no	+	-	-	Only applicable to specific cases, not optimal and robust.
IDVD	Boyarko et al. (2011), Ventura et al. (2015)	yes	no	yes	+-	+	+-	Trade-off between reducing the number of parameters in the NLP problem and optimality.
Iterative SDP	Kim and Mesbahi (2004), Kim et al. (2010), Kjellberg (2014), Tam and Glenn Lightsey (2016)	yes	yes	yes	+	+	-	Guaranteed convergence through solving a limited number of simple SDP problems. No global optimisation.
Iterative Rank Minimisation (IRM)	Sun and Dai (2015)	no	yes	yes	?	?	+	Very limited information available on the performance of this method.
SDP with inverse dynamics	Aucoin and Zee (2019)	no (basic method)	no (basic method)	yes, with some conserva-tiveness	+	+	+-	(Attitude) unconstrained problem can be solved fast, in a single iteration. Minimum-time and constrained problems can be solved in a much more slow, complicated and sub-optimal way.
Mixed-integer convex programming (MICP)	Eren et al. (2015)	no	yes	yes, with some conserva-tiveness	+-	+	+-	Guaranteed to find a solution. CPU time increases strongly as unit norm constraint is better approximated.
Heuristic	Spiller et al. (2015), Pontani and Melton (2016)	yes	yes	yes	-	+-	+-	A wide range of different types and methods exists.
Discretised pathfinding	Kjellberg and Lightsey (2013), Tanygin (2014)	no	yes	yes	+	+	-	Not close to optimality.
Sequential convex programming	McDonald et al. (2020), Virgili-Llop et al. (2018), Jerez et al. (2017)	potentially	yes	yes	+	+	+	

3.8.2. Sequential convex programming-based approaches

Before this research project was started, three earlier studies had developed SCP-based optimisation methods for the spacecraft reorientation problem. In Table 3.2, the main characteristics and key results of these three studies are listed.

Table 3.2: Overview of the characteristics of three SCP-based approaches to the optimal spacecraft reorientation problem.

Study	McDonald et al. (2020)	Virgili-Llop et al. (2018)	Jerez et al. (2017)
Discretisation method	Trapezoidal (TM)	Zero-order-hold (ZOH)	Euler (EM)
Dynamics	Rigid-body	Rigid-body (inverse)	Rigid-body
Attitude parameterisation	Quaternions	CRPs	Quaternions
Attitude constraints	No	Yes	Yes
Objective	Minimum-energy	Minimum-energy	Minimum-energy
Speed	? (not reported)	200 ms (median)	'milliseconds'
Optimality	Nearly identical to NLP reference solution	0.7% overcost (median)	? (not reported)
Robustness	? (not reported)	96.7% convergence rate	? (not reported)

From the studies presented in Table 3.2, it was concluded that significant research opportunities existed regarding the application of SCP-based optimisation methods to the spacecraft reorientation problem. To begin with, all existing studies only studied the relatively simple, minimum-energy reorientation problem. In addition, the promising first-order-hold discretisation method (Subsection 4.5.2) had not yet been implemented for this problem. Furthermore, these studies developed relatively 'basic' SCP algorithms that did not rely on advanced techniques, such as adaptive trust regions. Moreover, the performance analyses of the SCP algorithms that were developed can at best be described as *shallow*, as McDonald et al. (2020) analysed only a single test case and Jerez et al. (2017) did not provide any results whatsoever. Finally, the performance of the algorithm developed by Virgili-Llop et al. (2018) seemed not to meet the requirements specified in Section 3.7. It could therefore be concluded that on the basis of these three studies, the research (sub-)questions of Section 1.3 could not be answered.

During the course of this research project, three other studies were published that applied SCP-based optimisation methods to the spacecraft reorientation problem. In Table 3.3, the main characteristics and results of these studies are provided for overview purposes.

Table 3.3: Overview of the characteristics of three SCP-based approaches to the optimal spacecraft reorientation problem, which were published during this research project.

Study	Reynolds et al. (2021)	Preda et al. (2021)	Sin et al. (2021)
Discretisation method	First-order-hold (FOH)	First-order-hold (FOH)	First-order-hold (FOH)
Dynamics	Momentum wheels (Equation 3.3)	Momentum wheels (Equation 3.3)	Rigid-body
Attitude parameterisation	Quaternions	Quaternions	Quaternions
Attitude constraints	Yes	Yes	No
Objective	Minimum-energy	Maximise observation time	Minimum-time
Speed	19 ms (median, 10 nodes)	110 ms (median)	20.1 s (single case)
Optimality	? (not reported)	? (not reported)	? (not reported)
Robustness	100% convergence rate	100% convergence rate	? (not reported)

In the context of this research project, the studies of Preda et al. (2021) and Sin et al. (2021) are not extremely relevant. Sin et al. (2021) solved a relatively straightforward minimum-time problem without constraints on the angular rate and attitude of the spacecraft, and only provided performance results for a single test case. Furthermore, Preda et al. (2021) analysed a different type of optimisation problem than the ones that will be considered in this study. The objective of their study was to optimise the science time for the upcoming Comet Interceptor flyby mission of ESA. However, both studies do provide confidence regarding the choice of the quaternion attitude parameterisation in combination with the first-order-hold discretisation method, which were both implemented into the SCP algorithm that was developed in this study (Chapter 4). The study of Reynolds et al. (2021), however, is more comparable to the work that has been performed in this

study. As is discussed in Chapter 9, the performance results that were reported in this study are in line with the performance of the SCP algorithm that was developed during the MSc thesis research project.

II

Algorithm Design and Methodology

4

Sequential Convex Programming Algorithm

This chapter presents the SCP-based optimisation algorithm that was developed and implemented in this study. While this chapter addresses the main design features and choices, specific details regarding its implementation and its performance are provided in, respectively, Chapter 5 and Chapters 7 to 9. This chapter should enable the reader to obtain a thorough understanding of the SCP framework and the algorithm that was developed as part of this research project. In Section 4.1, the structure of this chapter and the contents of Sections 4.2 to 4.9 are presented. In Section 4.10, an overview is provided of the convex sub-problems that are solved by the SCP algorithm.

4.1. Overview

In this section, a top-level description is provided of the SCP algorithm and its main steps. These individual steps directly correspond to the sections in the remainder of this chapter. In Figure 4.1, the main elements and (analytical) steps of the SCP algorithm are shown. As was discussed in Subsection 2.4.4, the essence of an SCP algorithm is to replace a non-convex optimisation problem with a sequence of convex sub-problems that approximate the original non-convex problem. These convex sub-problems can be solved with highly efficient convex solving algorithms. The assumption behind the SCP framework is that the resulting approximation error decreases as more SCP iterations are performed, until a solution is obtained which is equivalent to the solution to the original, non-convex problem.

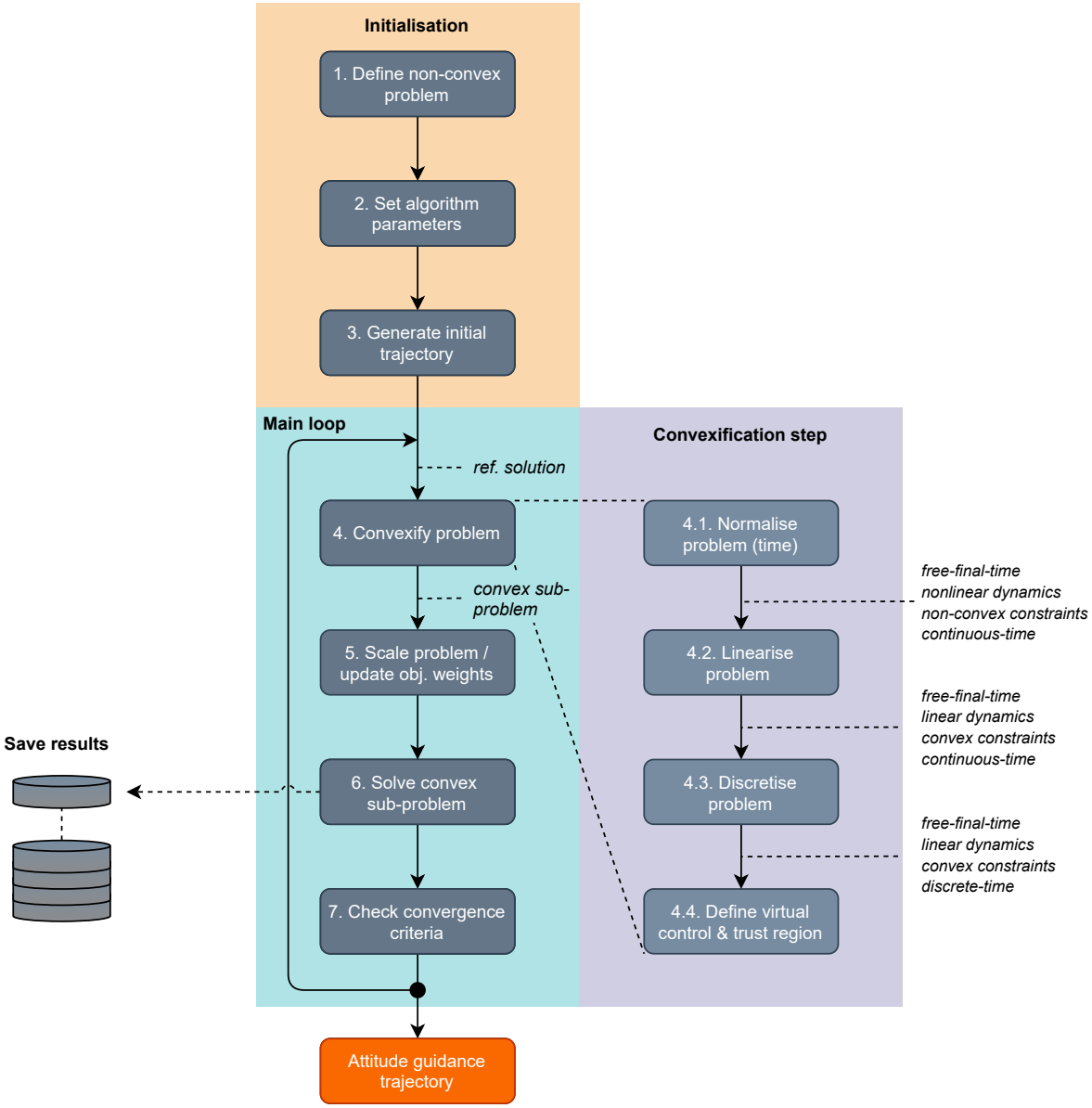


Figure 4.1: Block diagram of the main steps within the sequential convex programming algorithm.

In the remainder of this section, the steps from Figure 4.1 are briefly introduced.

1 **Define non-convex problem** (Section 3.6)
In this step, the problem elements that were introduced in Section 3.6, the spacecraft parameters, constraints, boundary conditions, and problem objective, are defined.

2 **Set algorithm parameters**
Several algorithm parameters, a set of specific settings for the SCP algorithm, have to be defined. These parameters are introduced in the remainder of this chapter.

3 **Generate initial trajectory** (Section 4.2)
In this step, an initial guess is generated for the state trajectory. The SCP algorithm requires an initial guess for its first iteration, as the convexification process relies on a reference solution.

4 **Convexify problem**
In the convexification step, the non-convex problem from Section 3.6 is convexified into a convex, numerical optimisation problem that can be solved with efficient numerical solving algorithms. This step can be further broken down into four main steps:

4.1 Normalise problem (time) (Section 4.3)
In this step, the fixed-final-time problem is reformulated into the free-final-time formulation that is required to solve the minimum-time reorientation problem.

4.2 Linearise problem (Section 4.4)
At this point, the problem is still non-convex. In the linearisation step, the elements of the problem that are non-convex (e.g., the dynamics) are linearised or convexified into convex approximations that can be dealt with by the convex solving algorithms considered in this study.

4.3 Discretise problem (Section 4.5)
Then, the convex, continuous problem is discretised in order to obtain the finite set of optimisation parameters that is required for numerical optimisation methods.

4.4 Define virtual control and trust region (Sections 4.6 and 4.7)
In this step, two distinct techniques are applied which improve the performance of the SCP algorithm.

5 Scale problem / update objective weights (Section 4.8)
It was found that scaling the optimisation parameters improved the numerical performance of the SCP algorithm. As well as this, certain algorithm parameters from step (**2**) are updated.

6 Solve convex sub-problem (Chapter 5)
Then, the scaled convex sub-problem is solved in order to obtain an updated guidance solution for the spacecraft reorientation problem.

7 Check convergence criteria (Section 4.9)
As the convexification step introduces a linearisation error, the solution to the convex sub-problem of step (**6**) is initially not equivalent to the solution of the original non-convex problem. However, as multiple iterations of this scheme (*main loop* in Figure 4.1) are performed, the local optimum is approached and the difference between subsequent solutions decreases. As a result, the linearisation error in the solution to the convex sub-problem gradually reduces as well. In this step, several convergence criteria are evaluated. If all of these convergence criteria are met, the linearisation error is deemed to be sufficiently small and the solution is *accepted*. At this point, a guidance trajectory is obtained that is, within a certain accuracy level, equivalent to a solution to the original, non-convex problem.

In the remainder of this chapter, these steps are explained in more detail.

4.2. Initialisation

The iterative process of the SCP framework always requires a reference solution around which the problem dynamics, kinematics and constraints can be convexified. As a result, an inherent challenge is formed by the need to provide an initial reference solution. This first reference solution is also referred to as the *initial guess*. In literature, it has been suggested that the SCP framework typically does not require a high-quality initial guess to find a feasible solution for most optimisation problems (McDonald et al., 2020). However, as the quality of the initial solution increases, the algorithm tends to require fewer iterations to converge, improving its computational efficiency. Considering the strict requirements on computational efficiency that govern onboard, real-time guidance applications, a high-quality initialisation is therefore an interesting research direction. During this research project, three different initialisation techniques were implemented and studied.

These three techniques only consider the initialisation of the reference state trajectory $\bar{\boldsymbol{x}}$. The reference control profile is also used in the linearisation (Section 4.4) and discretisation steps (Section 4.5) of the SCP algorithm, but has a significantly smaller effect on the approximation accuracy of the convexified dynamics than the state trajectory. Therefore, the control was simply set to zero for the corresponding initial guess, as was done in earlier studies of Reynolds et al. (2021) and McDonald et al. (2020). Thereby, developing a more accurate initial guess for the control profiles remains as a recommendation for further research. For the minimum-time problem, an initial guess for the manoeuvre duration, $t_{f,\text{guess}}$, has to be provided in addition to the initial guess for the state ($\bar{\boldsymbol{x}}$) and control ($\bar{\boldsymbol{u}}$) profiles. However, developing a strategy to accurately estimate this parameter was considered out of scope for this research project, leaving it to be a potential topic for further research as well.

4.2.1. Straight-line initialisation

Straight-line initialisation is one of the most commonly encountered initialisation techniques in literature to obtain proper reference trajectories to initialise an SCP algorithm. This technique consists of a linear interpolation of the state variables between the initial ($\boldsymbol{x}_0$) and final ($\boldsymbol{x}_f$) boundary conditions, and can be represented as (Reynolds et al., 2020a)

$$x_{k,i} = \frac{K-k}{K-1} x_{0,i} + \frac{k-1}{K-1} x_{f,i}, \quad \forall \quad i = 1,\dots,n_x, \quad k \in \mathscr{K} = \{1,\dots,K\} \tag{4.1}$$

where K is the number of grid nodes of the discretised problem (Section 4.5) and $\mathscr{K}$ the set of grid nodes. While having been proven to be useful and effective for powered descent problems in 3-D space, the 4-D (quaternion) nature of the spacecraft reorientation problem implies that the quality of this approach has its limits. Furthermore, the linear interpolation does not reflect the *speeding up* and *slowing down* phases that are characteristic of a typical, control-constrained reorientation manoeuvre.

4.2.2. Spherical linear interpolation

Spherical linear interpolation (SLERP) was first introduced by Shoemake (1985), and is one of the most popular tools to obtain an elementary interpolated great arc between two quaternions. Therefore, in contrast to the straight-line initialisation method, this approach does account for the nature of the 4-D quaternion space. It can be formulated as (Terzakis et al., 2018)

$$\boldsymbol{q}(t) = \boldsymbol{q}_0 \frac{\sin((1-t)\theta)}{\sin(\theta)} + \boldsymbol{q}_f \frac{\sin(t\theta)}{\sin(\theta)} \tag{4.2}$$

where $t \in [0,1]$ is referred to as the interpolation parameter, and $\theta = \cos^{-1}\left(\boldsymbol{q}_0 \cdot \boldsymbol{q}_f\right)$ is the angle between $\boldsymbol{q}_0$ and $\boldsymbol{q}_f$. Again, one can intuitively understand that the actual spacecraft will not perform its rotation with a constant angular rate. Therefore, although this technique represents a more realistic quaternion trajectory than the straight-line method, the challenge of accounting for the *speeding up* and *slowing down* phases has not been solved for this initialisation strategy.

4.2.3. Shape-based initialisation

The shape-based initialisation technique uses an analytical polynomial to represent the quaternion state trajectory. This technique is based on earlier work of Caubet and Biggs (2013), who used a shape-based approach to obtain attitude guidance trajectories. The main principle of this method is to represent every quaternion element by a third-order polynomial (which has four coefficients). In this manner, all four quaternion elements can be represented by the polynomials

$$q_i = a_i + b_i t + c_i t^2 + d_i t^3, \quad \forall \quad i = 1,\dots,4 \tag{4.3}$$

By solving a simple linear system for each quaternion element q_i, which only requires the use of the boundary conditions that have been defined for the optimisation problem at hand, the coefficients of these polynomials can be obtained. These linear systems can be formulated as

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \end{bmatrix} = \begin{bmatrix} q_{0,i} \\ q_{f,i} \\ \dot{q}_{0,i} \\ \dot{q}_{f,i} \end{bmatrix}, \quad \forall \quad i = 1,\dots,4 \tag{4.4}$$

Caubet and Biggs (2013) assumed a rest-to-rest reorientation problem, and could therefore simply set $\dot{q}_{0,i} = \dot{q}_{f,i} = 0$. However, an alternative approach had to be taken in this study to account for the non-zero initial and final angular rates that are imposed on the problem. To this end, the first-order quaternion derivatives required for Equation 4.4 are calculated using $\boldsymbol{\omega}_0$ and $\boldsymbol{\omega}_f$ through the following linear system (Wie, 2008):

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_4 & -q_3 & q_2 & q_1 \\ q_3 & q_4 & -q_1 & q_2 \\ -q_2 & q_1 & q_4 & q_3 \\ -q_1 & -q_2 & -q_3 & q_4 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix} \tag{4.5}$$

While Caubet and Biggs (2013) proposed to calculate the polynomial coefficients for only three quaternion elements q_i, and calculate q_4 through the quaternion unit-norm property[1], a different approach was taken in this study: it was proposed to define an analytical polynomial for *all* quaternion elements. The reason for this is that this approach allows for analytically differentiating the polynomials representing each q_i to obtain $\dot{q}_i$. Evaluations of $\dot{q}$ can then be used to obtain a rough initial guess for the angular rate parameter $\boldsymbol{\omega}$, through the relation (Wie, 2008)

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix} = 2 \begin{bmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \\ q_1 & q_2 & q_3 & q_4 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} \tag{4.6}$$

After the polynomial coefficients for all four quaternion elements have been determined, the quaternion $\boldsymbol{q}_k$ and angular rate $\boldsymbol{\omega}_k$ are determined for all K nodes of the grid (Section 4.5) to obtain the initial guess. Subsequently, the resulting K quaternions are normalised at every grid point. An alternative approach would be to first normalise $\boldsymbol{q}_k$ and then calculate $\boldsymbol{\omega}_k$. However, due to the fact that the first guess is typically a bad approximation of the optimal solution, the effects of this approach on the performance of the SCP algorithm are expected to be very limited.

4.3. Time-normalisation

In the (time-)normalisation step, the non-convex, continuous, *fixed-final-time* problem is converted into a non-convex, continuous, *free-final-time* problem. It is the first step in the convexification block of Figure 4.1. The free-final-time formulation is required for problems where the final time t_f is not a fixed, predetermined value, but instead an optimisation parameter that is, for instance, to be minimised or is restricted to lie within a certain range. Due to the fact that t_f is free for these types of problems, another variable than the real time $t \in [t_0, t_f]$, which is usually used for optimisation problems, is to be selected as the independent variable for the optimal spacecraft reorientation problem. Although there are different strategies to do so, it was decided that using a temporally normalised trajectory time $\tau \in [0,1]$ was the most logical option for the attitude problem considered in this study. The normalised trajectory time τ can be related to the real time t through the relation

$$\tau = \frac{t - t_0}{t_f - t_0} \tag{4.7}$$

Using this definition of the normalised trajectory time τ, the nonlinear dynamics and kinematics expressed in real time (Equations 3.1 and 3.4) can be reformulated as (Szmuk et al., 2020)

$$\boldsymbol{x}'(t) = \frac{d}{d\tau}\boldsymbol{x}(t) = \frac{dt}{d\tau}\frac{d}{dt}\boldsymbol{x}(t) = \left(\frac{dt}{d\tau}\right)\dot{\boldsymbol{x}}(t) = \eta\dot{\boldsymbol{x}}(t) \tag{4.8}$$

where $\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t))$ represents the original, nonlinear dynamics, and the *time dilation factor* is defined as $\eta = \frac{dt}{d\tau} \in \mathbb{R}^+$. In accordance with Equation 4.8, one can express the time-normalised dynamics as (Szmuk et al., 2020)

$$\boldsymbol{x}'(\tau) = F(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \eta) = \eta \cdot f(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)) \tag{4.9}$$

In this study, the time dilation factor η is treated as a constant for the fixed-final-time, minimum-energy problem and as an optimisation parameter for the free-final-time, minimum-time problem. Apart from the nonlinear problem dynamics and kinematics, no other problem constraints have to be time-normalised. The reader might notice that when $t_0 = 0$ (as is assumed in this study), $\eta = t_f$, turning Equation 4.9 into

$$\boldsymbol{x}'(\tau) = \eta \cdot f(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)) = t_f \cdot f(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau)) \tag{4.10}$$

4.4. Linearisation

The linearisation step is arguably one of the most important elements of the SCP algorithm. In this step the *nonlinear* dynamics and *non-convex* problem constraints are converted into *linear*, *convex* forms that can be handled by the convex solving algorithms that are applied in this study. In other words, the *non-convex*, continuous, free-final-time problem from Section 4.3 is converted into a *convex*, continuous, free-final-time approximation of the original, non-convex problem. There are four sources of non-convexity in the spacecraft reorientation problems defined in Section 3.6: the nonlinear dynamics, time-dependent boundary

[1] This approach does require the selection of the sign of q_4 according to the signs of its initial and final values (Caubet and Biggs, 2013).

conditions, objective function, and conical attitude constraints, which are discussed in Subsections 4.4.1, 4.4.2, 4.4.3, and 4.4.4, respectively.

4.4.1. Dynamics and kinematics

The main challenge of the application of convex solving methods to the spacecraft reorientation problem is the fact that this problem is characterised by highly nonlinear, non-convex dynamics and kinematics. In the so-called *linearisation step*, these nonlinear dynamics are approximated by a first-order Taylor series around a reference trajectory $\bar{\boldsymbol{z}}(\tau) = \begin{bmatrix}\bar{\boldsymbol{x}}(\tau) & \bar{\boldsymbol{u}}(\tau) & \bar{\eta}\end{bmatrix}$. Accordingly, the linear, time-varying dynamics as a function of the normalised trajectory time τ can be mathematically represented as approximating

$$\boldsymbol{x}'(\tau) = F(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \eta) \tag{4.11}$$

by (Szmuk et al., 2020)

$$\boldsymbol{x}'(\tau) \approx \boldsymbol{x}'_{\text{approx}}(\tau) = F(\bar{\boldsymbol{x}}(\tau), \bar{\boldsymbol{u}}(\tau), \bar{\eta}) + A(\tau)\left(\boldsymbol{x}(\tau) - \bar{\boldsymbol{x}}(\tau)\right) + B(\tau)\left(\boldsymbol{u}(\tau) - \bar{\boldsymbol{u}}(\tau)\right) + C(\tau)\left(\eta - \bar{\eta}\right) \tag{4.12}$$

where $\bar{\boldsymbol{x}}$, $\bar{\boldsymbol{u}}$, and η represent the optimisation parameters of the reference solution, $\boldsymbol{x}$, $\boldsymbol{u}$, and η represent the optimisation parameters for the current iteration, and A, B and C represent the first-order derivatives of the problem dynamics F as

$$A(\tau) := \left.\frac{\partial F}{\partial \boldsymbol{x}}\right|_{\bar{z}(\tau)} \tag{4.13}$$

$$B(\tau) := \left.\frac{\partial F}{\partial \boldsymbol{u}}\right|_{\bar{z}(\tau)} \tag{4.14}$$

$$C(\tau) := \left.\frac{\partial F}{\partial \eta}\right|_{\bar{z}(\tau)} \tag{4.15}$$

Using Equation 4.9 $\left(F(\bar{\boldsymbol{x}}(\tau), \bar{\boldsymbol{u}}(\tau), \bar{\eta}) = C(\tau)\bar{\eta}\right)$, Equation 4.12 can be simplified into

$$\boldsymbol{x}'_{\text{approx}}(\tau) = A(\tau)\left(\boldsymbol{x}(\tau) - \bar{\boldsymbol{x}}(\tau)\right) + B(\tau)\left(\boldsymbol{u}(\tau) - \bar{\boldsymbol{u}}(\tau)\right) + C(\tau)\eta \tag{4.16}$$

and, finally, further simplified into

$$\boldsymbol{x}'_{\text{approx}}(\tau) = A(\tau)\boldsymbol{x}(\tau) + B(\tau)\boldsymbol{u}(\tau) + C(\tau)\eta + \boldsymbol{r}(\tau) \tag{4.17}$$

where

$$\boldsymbol{r}(\tau) := -A(\tau)\bar{\boldsymbol{x}}(\tau) - B(\tau)\bar{\boldsymbol{u}}(\tau) \tag{4.18}$$

In other words, the time-normalised problem dynamics $\boldsymbol{x}'(\tau)$ can be approximated for a certain trajectory $\boldsymbol{z}(\tau) = \begin{bmatrix}\boldsymbol{x}(\tau) & \boldsymbol{u}(\tau) & \eta\end{bmatrix}$, when a reference trajectory $\bar{\boldsymbol{z}}(\tau) = \begin{bmatrix}\bar{\boldsymbol{x}}(\tau) & \bar{\boldsymbol{u}}(\tau) & \bar{\eta}\end{bmatrix}$ is provided. Typical for first-order Taylor series expansions, this approximation becomes less accurate as the difference between $\boldsymbol{z}(\tau)$ and $\bar{\boldsymbol{z}}(\tau)$ (or $\Delta\boldsymbol{z}$)) becomes larger. To put it in a different way, as $\Delta\boldsymbol{z}$ increases, the so-called *linearisation error* increases, and the convex approximation of the nonlinear dynamics becomes less accurate.

This is a key challenge within the SCP framework, as it implies that the convexified (sub-)problem with its approximated, linear dynamics is not equivalent to the original problem with nonlinear dynamics as a consequence of this linearisation error. As a result, it must be ensured that this linearisation error is within a certain limit for the final, accepted guidance trajectory.

4.4.2. Boundary conditions

While the constant boundary conditions from Subsection 3.3.1 are linear, convex equality constraints that can be directly incorporated into the convex sub-problem formulation, this is not the case for the time-dependent (final) boundary conditions from Subsection 3.3.2. The polynomials used to represent these time-dependent boundary conditions lead to nonlinear, non-convex equality constraints which cannot be incorporated into the problem formulation that is required for the application of convex solving algorithms. Therefore, these boundary constraints need to be convexified.

The technique that was applied for this purpose is referred to in literature as *successive approximation* (Liu et al., 2017). Using this technique, the nonlinear equality constraint is approximated as

$$\boldsymbol{f}(\boldsymbol{a}) \approx \boldsymbol{f}(\bar{\boldsymbol{a}}) \tag{4.19}$$

where $\boldsymbol{a}$ is some optimisation variable. In other words, the nonlinear boundary conditions are simply evaluated around the solution $\bar{\boldsymbol{a}}$ of the most recent reference trajectory. For the optimal spacecraft reorientation problem considered in this study, this method leads to two convex equality constraints for the final quaternion and angular rate, which can, respectively, be formulated as

Time-dependent boundary conditions

$$\boldsymbol{q}_f = \boldsymbol{p}_{q_f}(\bar{\eta}) \tag{4.20}$$

$$\boldsymbol{w}_f = \boldsymbol{p}_{w_f}(\bar{\eta}) \tag{4.21}$$

where $\bar{\eta}$ is the time dilation factor of the most recent SCP reference trajectory. It must be noted that the polynomials of Equation 4.20 are only evaluated for three quaternion elements, while q_4 is obtained through the quaternion unit norm to ensure that the boundary conditions satisfy the unit-norm constraint. As was noted for the shape-based initialisation strategy (Subsection 4.2.3), this approach requires the sign of q_4 to be in line with its corresponding polynomial. For future research, it is recommended to, instead, evaluate the polynomials for all four quaternion elements, and then normalise the obtained $\boldsymbol{q}_f$. This is expected to (slightly) improve the quality of the approximation and eliminate the need to select the proper sign for q_4.

Through the convergence criterion ϵ_η, which is introduced in Section 4.9, it is ensured that $|\eta - \bar{\eta}|$ is under a predefined limit for the final iteration of the SCP algorithm. Consequently, it can be ensured that the approximation error of these final boundary conditions is within a defined range.

Another approach that was considered to convexify the time-dependent boundary conditions is the *successive linearisation* convexification technique (Liu et al., 2017). This scheme represents a first-order Taylor series expansion and can be formulated as

$$\boldsymbol{q}_f = \boldsymbol{p}_{q_f}(\bar{\eta}) + \frac{d\boldsymbol{p}_{q_f}}{d\eta}(\bar{\eta})(\eta - \bar{\eta}) \tag{4.22}$$

Although this technique leads to a smaller approximation error for the boundary conditions, it became impossible to ensure that these conditions satisfied the quaternion unit-norm constraint (due to SOCP limitations). This led to cases for which the SCP algorithm could not converge. Therefore, the successive approximation technique was implemented instead. It is noted that this successive linearisation technique can be used for the time-dependent boundary condition on the angular rate (or control), as this parameter does not face such a constraint.

4.4.3. Objective function

As was discussed in Subsection 2.4.2, the convex solvers considered in this study only support an objective that is a linear function of the optimisation parameters. As a result, the Mayer-type objective function of the minimum-time problem can be directly incorporated into the convex optimisation sub-problems. However, the quadratic minimum-energy objective from Equation 3.18 is not supported by the convex solving algorithms applied in this study, and an alternative approach had to be found to implement this type of objective function into the SCP framework.

The method that was used towards this end is sometimes referred to as the *equivalent transformation* convexification method (Liu et al., 2017). This method is regularly used in an SCP context to augment an objective function with nonlinear, non-convex terms (Reynolds et al., 2020a; Sagliano et al., 2020; Wang et al., 2019b). The equivalent transformation method involves the introduction of an additional optimisation parameter, a so-called *slack variable*, which replaces the nonlinear term in the objective function. As a result, the objective function becomes a linear function of the optimisation parameters, as desired. In addition, a convex constraint is introduced to the problem which enforces the minimisation of the original, nonlinear term. In

a general way, this scheme can be represented as replacing the nonlinear objective (Liu et al., 2017)

$$J = \int_{t_0}^{t_f} \boldsymbol{a}^2 \mathrm{d}t \tag{4.23}$$

by

$$J = \int_{t_0}^{t_f} s_a \mathrm{d}t \tag{4.24}$$

where s_a is the slack variable. The additional convex constraint that is required to enforce the minimisation of the original, nonlinear term, is formulated as

$$\boldsymbol{a}^2 \leqslant s_a \tag{4.25}$$

As one can observe, the nonlinear function of the optimisation parameter, $\boldsymbol{a}^2$, is replaced by a linear function of the slack variable s_a, while this slack variable is directly related to the $\boldsymbol{a}^2$-term that is to be minimised in Equation 4.23. For the minimum-energy spacecraft reorientation problem studied in this project, using this equivalent transformation technique, Equation 3.18 is replaced by

$$J_u = w_u \int_{t_0}^{t_f} s_u dt \tag{4.26}$$

and

$$\boldsymbol{u}^2 \leqslant s_u \tag{4.27}$$

4.4.4. Attitude constraints

In this section, the attitude constraints from Subsections 3.4.3 and 3.4.4 are reformulated into a representation that can be incorporated into the convex sub-problems that are solved by the SCP algorithm. It must be noted that these constraints are not *linearised*, as the title of this section suggests. However, as these constraints are reformulated into a convex representation (convexified), this section was found to be the most suitable place to explain the process that was followed for this purpose.

Several previous studies on the optimal spacecraft reorientation problem have focused on reformulating the conical attitude constraints from Equations 3.16 and 3.17 into a variety of other representations, such as an extended form (Pontani and Melton, 2016) and a quadratic inequality form (Kim and Mesbahi, 2004). The latter is of particular interest to this study and formulates the quaternion characterisation of the exclusion cone of Equation 3.16 as

$$\boldsymbol{q}(t)^T \widetilde{A}(\boldsymbol{x}_B, \boldsymbol{y}_I, \theta_{\text{sep,out}}) \boldsymbol{q}(t) \leq 0 \tag{4.28}$$

where

$$\widetilde{A}(\boldsymbol{x}_B, \boldsymbol{y}_I, \theta_{\text{sep,out}}) = \begin{bmatrix} A & \boldsymbol{b} \\ \boldsymbol{b}^T & d \end{bmatrix} \quad \in \quad \mathbb{R}^{4\times 4} \tag{4.29}$$

where

$$A = \boldsymbol{x}_B \boldsymbol{y}_I^T + \boldsymbol{y}_I \boldsymbol{x}_B^T - \left(\boldsymbol{x}_B^T \boldsymbol{y}_I + \cos\theta_{\text{sep,out}}\right) \boldsymbol{I}_{3\times 3} \tag{4.30}$$

$$\boldsymbol{b} = \boldsymbol{y}_I \times \boldsymbol{x}_B, \quad d = \boldsymbol{x}_B^T \boldsymbol{y}_I - \cos\theta_{\text{sep,out}} \tag{4.31}$$

where $(\cdot)_B$ and $(\cdot)_I$ denote that the vector is represented in, respectively, the body-fixed reference frame F_B and the inertial frame F_I, and I_3 is the third-order identity matrix. This quadratic formulation of the attitude keep-out constraint has a real set of eigenvalues due to the symmetric nature of $\widetilde{A}$. However, it can still be non-convex, as $\widetilde{A}$ is not always positive-semidefinite, which is required for a quadratic constraint to be convex (Eren et al., 2015; Kim and Mesbahi, 2004).

Starting from the non-convex constraint from Equation 4.28, Kim and Mesbahi (2004) developed a very useful, theoretical proof that this constraint can be convexified for the quaternion attitude parameterisation through a relatively straightforward spectrum shift. They succeeded in obtaining a convex, quadratic formulation which can be represented as (Eren et al., 2015; Kim and Mesbahi, 2004; Lee and Mesbahi, 2012)

$$\boldsymbol{q}(t)^{\mathrm{T}} \hat{P}(\boldsymbol{x}_B, \boldsymbol{y}_I, \theta_{\text{sep,out}}) \boldsymbol{q}(t) \leq \mu \tag{4.32}$$

where

$$\hat{P} = \widetilde{A} + \mu I_4 \tag{4.33}$$

where μ is chosen to be strictly greater than the largest eigenvalue of $-\widetilde{A}$. Essentially, the attitude constraint is converted into a convex, quadratic constraint by shifting the spectrum of the $\widetilde{A}$ matrix. It must be noted that the spectrum shift through μ, which convexifies the constraint, is only valid when the unit norm property of the quaternions is properly enforced (Eren et al., 2015; Kim and Mesbahi, 2004). In studies of Jerez et al. (2017) and Reynolds et al. (2021), it was found that taking $\mu = 2$ suffices for the convexification of the attitude constraints in this study. The convex formulation of Equation 4.32 represents a keep-out constraint, but a similar reformulation can be applied to involve keep-in constraints into the convex problem formulation.

4.5. Discretisation

In the *discretisation step*, the infinite-dimensional, *continuous* optimisation problem is converted into a finite-dimensional numerical optimisation problem that can be solved by numerical solving algorithms. As a result, the convex, *continuous*, free-final-time problem with linearised dynamics and convex constraints from Section 4.4 is transformed into a convex, *discrete*, free-final-time problem. The section focuses on the discretisation techniques applied to the optimisation parameters, dynamics, and objective function, which are discussed in, respectively, Subsections 4.5.1, 4.5.2, and 4.5.3. For the sake of completeness, a brief elaboration is dedicated to the discretisation of the constraints in Subsection 4.5.4.

4.5.1. Optimisation parameters

The first step taken to discretise the convex, continuous problem of Section 4.4 into a finite-parameter optimisation problem is the discretisation of the continuous optimisation parameters $\boldsymbol{x}$ and $\boldsymbol{u}$. This is achieved by the introduction of K equally spaced temporal grid points (including the inital and final boundaries) that split up the normalised trajectory time $\tau \in [0,1]$. Each of these grid points is referred to with an index $k \in \mathcal{K}$, where

$$\tau_k = \frac{k-1}{K-1} \tag{4.34}$$

Accordingly, the continuous state and control optimisation parameters $\boldsymbol{x}$ and $\boldsymbol{u}$ are represented by a vector of, respectively, $(K \times n_x)$ and $(K \times n_u)$ elements as

$$\boldsymbol{x} = [\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \dots \quad \boldsymbol{x}_K] \tag{4.35}$$

$$\boldsymbol{u} = [\boldsymbol{u}_1 \quad \boldsymbol{u}_2 \quad \dots \quad \boldsymbol{u}_K] \tag{4.36}$$

4.5.2. Dynamics and kinematics

The discretisation of the problem dynamics and kinematics can be defined as finding a way to turn the continuous (time-normalised, linearised) dynamics from Equation 4.17 into a finite number of convex constraints that enforce the dynamics and kinematics between the state and control parameters (Equations 4.35 and 4.36) at the nodes of the grid. An accurate discretisation of the dynamics and kinematics that govern the spacecraft reorientation problem is crucial to ensure that the obtained attitude guidance manoeuvres are, in fact, feasible and that the obtained control profiles result in the predicted state trajectories.

Discretisation method

The desired discretisation method needs to (1) lead to a computationally efficient SCP algorithm, and (2) keep the *discretisation error* within an acceptable bound. In recent literature on sequential convex programming, a variety of different discretisation methods can be encountered, amongst others: a variety of pseudospectral methods, the trapezoidal method, and the Runge-Kutta method. Another method which has recently become popular (Szmuk et al., 2020), models the control profile with a zero-order-hold (ZOH) or a first-order-hold (FOH) assumption and applies the concept of the state transition matrix Φ_A from linear systems theory (Malyuta et al., 2019). There are few SCP-related studies that compare these different discretisation strategies in terms of performance, but two relatively recent studies by Malyuta et al. (2019) and Sagliano (2019) were found to provide enough information to select a discretisation method for this study. The main findings of the study of Malyuta et al. (2019) are presented in Figure 4.2.

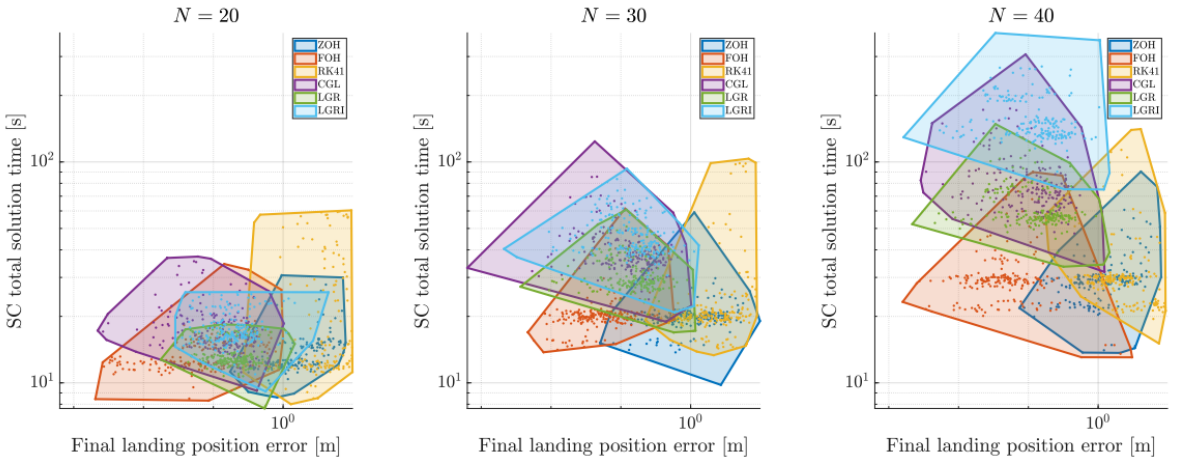


Figure 4.2: Landing position error (accuracy) versus overall solution time (computational efficiency) for a set of powered descent problems using six different discretisation techniques. N represents the number of grid nodes. The data points corresponding to each discretisation method are covered with their convex hull polytope for visualisation purposes (Malyuta et al., 2019).

The FOH transcription method was chosen for the following reasons:

- As can be observed in Figure 4.2, the FOH discretisation method shows very promising results in terms of solution times (y-axis) and a small landing position error (x-axis), where the latter is a direct indicator of the accuracy of the method.

- The FOH discretisation method relies on a first-order-hold modelling of the control (see next paragraph), which enables this method to guarantee that control constraints are not violated between the grid nodes (where these constraints are enforced). This cannot be said about the pseudospectral methods, which model the control with Lagrange polynomials that might violate control constraints between the grid nodes. Additional constraints could be added to prevent these violations from happening, but this would come at a cost of computational efficiency. As low computation times are one of the most important requirements on the SCP algorithm, it would be desirable to avoid this situation.

- The results of Figure 4.2 are in line with two studies by Sagliano (2018) and Sagliano (2019), which (to the best knowledge of the author) pioneered the application of pseudospectral discretisation methods in an SCP context. In these studies, it was found that, although pseudospectral methods show promising accuracy performance, they introduce a penalty in terms of computational performance because the resulting constraint matrices are less sparse.

- As covered in Section 3.8, during the thesis research project, other studies were published that also selected the FOH discretisation method for solving the spacecraft reorientation problem with an SCP algorithm (Preda et al., 2021; Reynolds et al., 2021; Sin et al., 2021). This could be seen as a confirmation of the promise that this method is considered to have in the wider research community.

The FOH discretisation method is both accurate and fast. It is accurate because it leverages the iterative nature of the SCP framework by extracting much more information from the previous reference solution than a simpler trapezoidal method. Furthermore, is fast because it leads to a sparser constraint matrix than, for instance, pseudospectral discretisation methods.

Control modelling

One of the main features of the FOH discretisation method consists of modelling the control parameter as a linear function (linear interpolation) between the nodes of the grid, as shown in Figure 4.3.

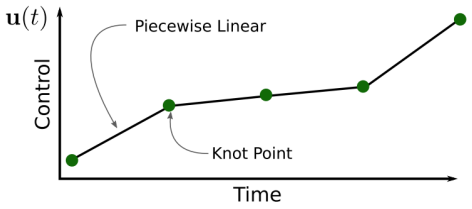


Figure 4.3: Piecewise-linear control modelling under a first-order-hold (FOH) assumption (Kelly, 2015).

This piecewise-linear modelling of the control variable can be mathematically formulated for each segment of the grid as (Szmuk et al., 2020)

$$\boldsymbol{u}(\tau) = \lambda_k^-(\tau)\boldsymbol{u}_k + \lambda_k^+(\tau)\boldsymbol{u}_{k+1}, \quad \forall \quad \tau \in [\tau_k, \tau_{k+1}] \tag{4.37}$$

where

$$\lambda_k^-(\tau) := \frac{\tau_{k+1} - \tau}{\tau_{k+1} - \tau_k}, \quad \text{and} \quad \lambda_k^+(\tau) := \frac{\tau - \tau_k}{\tau_{k+1} - \tau_k} \tag{4.38}$$

When Equation 4.37 is substituted in Equation 4.17, the following equation for the time-normalised, convex, continuous dynamics and kinematics is obtained for each *segment* between nodes k and $(k+1)$ of the reorientation manoeuvre

$$\boldsymbol{x}'_{\text{approx}}(\tau) = A(\tau)\boldsymbol{x}(\tau) + \lambda_k^-(\tau)B(\tau)\boldsymbol{u}_k + \lambda_k^+(\tau)B(\tau)\boldsymbol{u}_{k+1} + \boldsymbol{C}(\tau)\eta + \boldsymbol{r}(\tau), \quad \forall \quad \tau \in [\tau_k, \tau_{k+1}] \tag{4.39}$$

However, Equation 4.39 is still continuous and has to be discretised in order to be enforced in the numerical optimisation problem.

Discretisation procedure

The FOH discretisation method fundamentally relies on the result from linear systems theory that any $\boldsymbol{x}(\tau)$ (from Equation 4.17) on the segment $\tau \in [\tau_k, \tau_{k+1}]$ can be obtained through (Malyuta et al., 2019)

$$\boldsymbol{x}(\tau) = \Phi_A(\tau, \tau_k)\,\boldsymbol{x}(\tau_k) + \int_{\tau_k}^{\tau} \Phi_A(\tau, \xi)B(\xi)\boldsymbol{u}(\xi)\mathrm{d}\xi + \int_{\tau_k}^{\tau} \Phi_A(\tau, \xi)\boldsymbol{C}(\xi)\eta\mathrm{d}\xi + \int_{\tau_k}^{\tau} \Phi_A(\tau, \xi)\boldsymbol{r}(\xi)\mathrm{d}\xi \tag{4.40}$$

where the state transition matrix Φ_A from τ_k to ξ can be calculated using (Reynolds et al., 2020b)

$$\Phi_A(\xi, \tau_k) = I_{n_x \times n_x} + \int_{\tau_k}^{\xi} A(\zeta)\Phi_A(\zeta, \tau_k)\,\mathrm{d}\zeta \tag{4.41}$$

Using the inverse and transitive properties of the state transition matrix Φ_A (Reynolds et al., 2020b), one can arrive at the following, discrete equality constraint that enforces the linearised attitude dynamics and kinematics on the state of the spacecraft between nodes k and $(k+1)$

$$\boldsymbol{x}_{k+1} = A_k\boldsymbol{x}_k + B_k^-\boldsymbol{u}_k + B_k^+\boldsymbol{u}_{k+1} + \boldsymbol{C}_k\eta + \boldsymbol{r}_k \tag{4.42}$$

where

$$A_k := \Phi_A(\tau_{k+1}, \tau_k) \tag{4.43}$$

$$B_k^- := A_k \int_{\tau_k}^{\tau_{k+1}} \Phi_A^{-1}(\xi, \tau_k)\, B(\xi)\lambda_k^-(\xi)\mathrm{d}\xi \tag{4.44}$$

$$B_k^+ := A_k \int_{\tau_k}^{\tau_{k+1}} \Phi_A^{-1}(\xi, \tau_k)\, B(\xi)\lambda_k^+(\xi)\mathrm{d}\xi \tag{4.45}$$

$$\boldsymbol{C}_k := A_k \int_{\tau_k}^{\tau_{k+1}} \Phi_A^{-1}(\xi, \tau_k)\, \boldsymbol{C}(\xi)\mathrm{d}\xi \tag{4.46}$$

$$\boldsymbol{r}_k := A_k \int_{\tau_k}^{\tau_{k+1}} \Phi_A^{-1}(\xi, \tau_k)\, \boldsymbol{r}(\xi)\mathrm{d}\xi \tag{4.47}$$

In the implementation of this theoretical approach in the SCP algorithm, Equations 4.43 to 4.47 are integrated simultaneously with Equation 4.41, for every segment of the grid. The integrand evaluations of A, B, $\boldsymbol{C}$ and

$\boldsymbol{r}$ are based on the numerically propagated, nonlinear dynamic and kinematic equations of Equation 4.9 from the reference state (previous SCP solution) at the beginning of the segment (at node τ_k), which can be represented as

$$\overline{\boldsymbol{x}}(\tau) = \overline{\boldsymbol{x}}_k + \int_{\tau_k}^{\tau} F(\boldsymbol{x}(\zeta), \boldsymbol{u}(\zeta), \eta) \mathrm{d}\zeta \tag{4.48}$$

In Figure 4.4, this process is visualised. Using Equation 4.48, the state is propagated (red line) from the reference state value $\bar{\boldsymbol{x}}_1$ at a grid node (green point). In the meantime, Equation 4.41 and Equation 4.44 to Equation 4.47 are integrated in order to obtain the A, B, $\boldsymbol{C}$, and $\boldsymbol{r}$ matrices and vectors that are required for the convex equality constraints that enforce the dynamics and kinematics on the problem. After the integration process has reached the end of the segment, the integrands of the B, $\boldsymbol{C}$, and $\boldsymbol{r}$ terms are multiplied with the result for A_k to obtain B_k, $\boldsymbol{C}_k$, and $\boldsymbol{r}_k$.

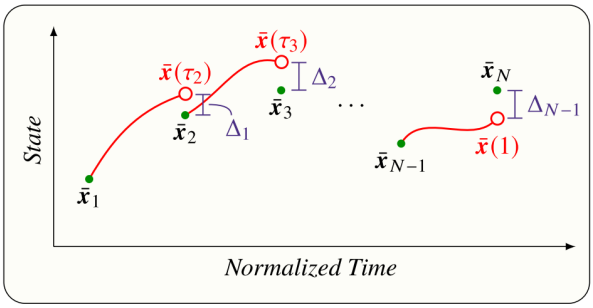


Figure 4.4: Illustration of the procedure that was used to obtain the reference trajectory used for the discretisation process (Reynolds et al., 2020b).

Effectively, using this approach, information on the problem dynamics and kinematics along the *entire* segment is introduced to the convex sub-problem. This is a major difference compared to the trapezoidal method that was used by, for instance, Liu and Lu (2014), Wang et al. (2019b), and McDonald et al. (2020), as that method only includes information on the problem dynamics and kinematics at the grid points k and $k+1$. This is the major driver behind the fact that the FOH discretisation method is characterised by a significantly higher accuracy.

In the implementation of the algorithm, this integration process is performed using a numerical Runge-Kutta 4th-order (RK4) scheme, which is, for the sake of completeness, provided in Equations 4.49 to 4.53.

$$k_1 = f\left(t_n, y_n\right) \tag{4.49}$$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + h\frac{k_1}{2}\right) \tag{4.50}$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + h\frac{k_2}{2}\right) \tag{4.51}$$

$$k_4 = f\left(t_n + h, y_n + hk_3\right) \tag{4.52}$$

$$y_{n+1} = y_n + \frac{1}{6}h\left(k_1 + 2k_2 + 2k_3 + k_4\right) \tag{4.53}$$

In earlier research, the Runge-Kutta 4th-order method was found to provide a good balance between computational performance and accuracy (Mao et al., 2018a; Reynolds et al., 2020a; Szmuk et al., 2020). As a result of the selection of this integration method, the two algorithm parameters that influence the discretisation performance of the algorithm are the number of grid nodes, K, and the number of (RK4) propagation nodes, K_{disc}, which determines the number of steps that are performed for each single grid segment to obtain the matrices to enforce the dynamics and kinematics in discretised form.

4.5.3. Objective function

The Mayer-type objective term $t_f \equiv \eta$ of the minimum-time problem from Equation 3.19 does not have to be discretised. However, the minimum-energy, Lagrange-type objective function from Equation 3.18 requires the evaluation of the integral of a continuous function and therefore has to be discretised to be included in the numerical form that is required for a finite-dimensional optimisation problem. Two methods to discretise this objective function term were studied, of which the trapezoidal method has often been used in other studies and the *successive approximation of the exact cost* method was developed during this research project.

Trapezoidal method

The trapezoidal method (TM) is often applied to discretise and approximate an integral in the objective function of an SCP optimisation sub-problem (McDonald et al., 2020; Sagliano, 2019; Wang and Grant, 2017). The general trapezoidal integration rule can be formulated as

$$\int_a^b f(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \approx \sum_{k=1}^{K-1}\left(\frac{f(\boldsymbol{x}_k)+f(\boldsymbol{x}_{k+1})}{2}\right) \tag{4.54}$$

As visualised in Figure 4.5, this integration rule is only exact when the function f that is to be integrated shows linear behaviour between the grid points. This is, unfortunately, typically not the case for the energy used during a spacecraft reorientation manoeuvre. As a result of the first-order-hold modelling of the control, the energy profile ($\boldsymbol{u}^2$) is a quadratic function. Hence, it can be concluded that this discretisation method is not exact for the reorientation problem considered in this study.

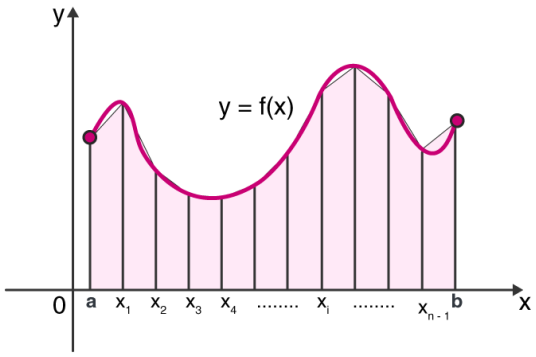


Figure 4.5: Visualisation of the trapezoidal integration technique.[2]

Using the trapezoidal scheme from Equation 4.54, the minimum-energy objective function can be formulated in discretised form as

Minimum-energy objective term

$$J_u = w_u s_u \tag{4.55}$$

where the continuous convex constraint of Equation 4.27 is replaced by

Minimum-energy convex constraint

$$\sum_{k=1}^{K} \boldsymbol{c}_{u,k}\boldsymbol{u}_k^2 \leq s_u \tag{4.56}$$

where $\boldsymbol{c}_u = [0.5 \quad 1 \quad \dots \quad 1 \quad 0.5] \in \mathbb{R}^K$, in accordance with the trapezoidal rule from Equation 4.54. In other words, all inner grid points have double the relative weight compared to the boundary points of the grid. As was stated before, this quadrature leads to a result that does not represent the *true* energy cost of a certain trajectory. Therefore, a second method was developed during this study that aims to improve the accuracy of this discretisation process.

[2] `https://medium.com/swlh/finite-element-analysis-f8eaba1ae54` (Cited: 28-06-2021).

Successive approximation of the exact cost

This discretisation technique for the minimum-energy objective function utilises both the iterative nature of the SCP algorithm and the first-order-hold assumption on the control parameter to approach the accuracy of analytically integrating the linear control over every segment. In this way, the energy-cost of the minimum-energy reorientation manoeuvre can be accurately evaluated in the convex sub-problems solved by the SCP algorithm.

First, the energy of a single segment of the grid is calculated through an analytical integration, where the control u is represented by the linear function $(ax+b)$

$$\int_0^{dt} (ax+b)^2\,dt, \quad \text{where} \quad a=\frac{u_{k+1}-u_k}{dt}, \quad \text{and} \quad b=u_k \tag{4.57}$$

$$\int_0^{dt} \left(a^2x^2+2abx+b^2\right)dt \tag{4.58}$$

$$\left[\frac{a^2x^3}{3}+\frac{2abx^2}{2}+b^2x\right]_0^{dt} \tag{4.59}$$

Substituting a and b, one obtains

$$\left[\frac{u_{k+1}^2-2u_{k+1}u_k+u_k^2}{dt^2}\frac{x^3}{3}+\frac{u_{k+1}u_k-u_k^2}{dt}x^2+u_k^2x\right]_0^{dt} \tag{4.60}$$

$$\frac{1}{3}\left(u_k^2+u_ku_{k+1}+u_{k+1}^2\right)dt \tag{4.61}$$

Unfortunately, due to the $u_{k+1}u_k$-term, this exact result cannot be directly included into the convex SOCP framework through the convexification method from Equation 4.25. Therefore, it is not possible to simply sum this analytic integral for all grid segments to obtain the exact integral of the objective function. However, one can approximate this exact integral using the iterative nature of the SCP algorithm. Using information from the control history of the previous iteration, $\bar{\boldsymbol{u}}$, it was thought that it could be possible to arrive at a more accurate estimation of the energy used. The scheme that was developed can be represented (for a single grid segment) as

$$\frac{1}{3}\left(u_k^2+\frac{1}{2}\bar{u}_ku_{k+1}+\frac{1}{2}u_k\bar{u}_{k+1}+u_{k+1}^2\right)dt \tag{4.62}$$

Omitting the constant multiplication factors of $\frac{1}{3}$ and dt, which leads to an equivalent solution, one can define the cost for all segments of the grid:

$$\begin{aligned}
&\textit{segment 1:} \quad u_0^2+\frac{1}{2}\bar{u}_0u_1+\frac{1}{2}u_0\bar{u}_1+u_1^2\\
&\textit{segment 2:} \quad u_1^2+\frac{1}{2}\bar{u}_1u_2+\frac{1}{2}u_1\bar{u}_2+u_2^2\\
&\qquad\qquad \dots\\
&\textit{segment } K-1\textit{:} \quad u_{K-1}^2+\frac{1}{2}\bar{u}_{K-1}u_K+\frac{1}{2}u_{K-1}\bar{u}_K+u_K^2
\end{aligned} \tag{4.63}$$

Summing these expressions for all $(K-1)$ segments, one obtains the following objective function

$$J_u = w_u \cdot s_u \tag{4.64}$$

where the accompanying convex inequality constraint is formulated as

$$\sum_{k=1}^{K} \boldsymbol{c}_{u,1,k}\boldsymbol{u}_k^2+\boldsymbol{c}_{u,2,k}\boldsymbol{u}_k \le s_u \tag{4.65}$$

and

$$\boldsymbol{c}_{u,1} = [0.5 \quad 1 \quad \dots \quad 1 \quad 0.5] \tag{4.66}$$

$$\boldsymbol{c}_{u,2} = [0.25\bar{u}_1 \quad (0.25\bar{u}_0+0.25\bar{u}_2) \quad (0.25\bar{u}_1+0.25\bar{u}_3) \quad \dots \quad 0.25\bar{u}_K] \tag{4.67}$$

In Section 8.2, both the trapezoidal and the successive approximation of the exact cost methods are analysed and their performance is compared.

It is noted that, in hindsight, an alternative approach is thought to exist to realise an exact integration of the minimum-energy objective in the SCP algorithm. Through introducing an additional set of k slack variables $\boldsymbol{r}_k = \boldsymbol{u}_k + \boldsymbol{u}_{k+1}$, and squaring these in the convex inequality constraint of Equation 4.65, the $u_{k+1}u_k$-term could be included directly into the formulation of the optimisation objective, eliminating the need for the successive approximations proposed in this subsection. In terms of optimality, the performance of both approaches is expected to be similar, but the additional optimisation parameters (slack variables), which increase the size of the optimisation problem, could have a negative effect on the computational efficiency of the algorithm. This alternative discretisation method serves as another example of the modelling power of the SOCP framework.

4.5.4. Constraints

The continuous constraints from Section 3.6 and Equation 4.32 are discretised in a straightforward manner: they are enforced on every single grid node. This results in the following constraints.

Control constraints

$$\left(\frac{u_{k,1}}{u_{\text{ellipse},1}}\right)^2 + \left(\frac{u_{k,2}}{u_{\text{ellipse},2}}\right)^2 + \left(\frac{u_{k,3}}{u_{\text{ellipse},3}}\right)^2 \leq 1, \quad \forall \quad k \in \mathcal{K} \tag{4.68}$$

Angular rate constraint

$$|\omega_{k,i}| \leq \omega_{\text{box},i}, \quad \forall \quad i \in 1,2,3, \quad k \in \mathcal{K} \tag{4.69}$$

Attitude constraints

$$\boldsymbol{q}_k^{\mathrm{T}} \hat{P}_{\text{out}} \boldsymbol{q}_k \leq 2, \quad \forall \quad k \in \mathcal{K} \tag{4.70}$$

$$\boldsymbol{q}_k^{\mathrm{T}} \hat{P}_{\text{in}} \boldsymbol{q}_k \leq 2, \quad \forall \quad k \in \mathcal{K} \tag{4.71}$$

As one might notice, since the control and state constraints are only enforced on the grid nodes, it automatically follows that it is not guaranteed that these constraints are satisfied along the entire trajectory. Although this problem is faced by most numerical optimisation methods, it does pose a significant challenge regarding the usefulness of the SCP optimisation algorithm specifically. This aspect is further analysed and discussed in Section 8.2.

4.6. Virtual control

The convex sub-problem that is obtained after the time-normalisation, linearisation and discretisation steps can already be solved using available (highly efficient) convex solving algorithms. However, one challenge that has arisen within the SCP framework is that problems that are feasible in their original, nonlinear, non-convex form, might become infeasible due to the conversion into a convex sub-problem. This phenomenon is called *artificial infeasibility*. Using a highly simplified reference problem, this phenomenon is visualised in Figure 4.6.

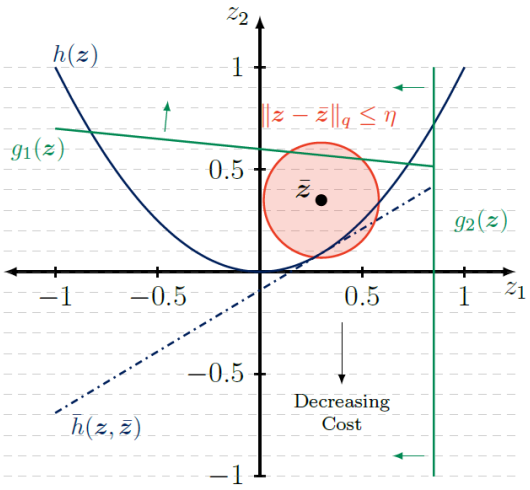


Figure 4.6: Illustration of the phenomenon of artificial infeasibility. It can be observed that the original problem (with the parabolic equality constraint h) has a feasible solution for $\boldsymbol{z}$, while the convexified problem (dashed, linear equality constraint) has none. The green lines represent inequality constraints. (Reynolds, 2020).

In Figure 4.6, the aim is to minimise an objective function for a problem with two optimisation parameters (z_1 and z_2) with a cost that decreases southwards in the graph. Three constraints are imposed on the problem: two linear inequality constraints (green lines), and one nonlinear equality constraint (blue parabola). It can be seen that, for the original NLP problem, the quadratic equality constraint and the two inequality constraints can be simultaneously satisfied. However, for the linearised, convexified problem, the linearised equality constraint (blue dashed line) and inequality constraints cannot be satisfied simultaneously, hence rendering the problem infeasible.

The strategy that is usually applied in the SCP framework to deal with this problem of artificial infeasibility is to introduce a so-called *virtual control* variable, which, regarding the example of Figure 4.6, effectively allows for a violation of the linearised equality constraint. In the context of the spacecraft reorientation problem, the virtual control variable is added to the linearised dynamic and kinematic equality constraints of Equation 4.42. This *virtual control* variable $\boldsymbol{\nu} \in \mathbb{R}^{n_x}$ is referred to as a *control* variable, as it can be interpreted as an additional control variable (next to $\boldsymbol{u}$) which can be optimised to influence the dynamics of the state $\boldsymbol{x}$. After defining this virtual control parameter, the convex, discrete, dynamic and kinematic equality constraints of the convex sub-problem can be formulated as

Dynamics and kinematics

$$\boldsymbol{x}_{k+1} = A_k \boldsymbol{x}_k + B_k^- \boldsymbol{u}_k + B_k^+ \boldsymbol{u}_{k+1} + \boldsymbol{C}_k \eta + \boldsymbol{r}_k + \boldsymbol{\nu}_k, \quad \forall \quad k = 1, \cdots, K-1 \tag{4.72}$$

It can be seen that the introduction of the virtual control variable leads to $K \times n_x$ additional optimisation variables. The introduction of this virtual control variable has been found to strongly improve the feasibility characteristics, convergence rate, and thus the robustness of the SCP algorithm (Malyuta et al., 2020; Mao et al., 2016; Szmuk et al., 2016). The use of this virtual control variable is primarily focused on the first iterations of the SCP algorithm, for which it is most important to find an approximate, rough solution to the convex sub-problem. As the use of the virtual control variable leads to dynamic infeasibility of the obtained solution, the final, converged solution should restrict its use of virtual control within a strict, low limit. Therefore, an additional term is added to the objective function of the convex optimisation sub-problem with a relatively large weight factor in order to penalise the use of virtual control. This term can be formulated as

Virtual control objective

$$J_{\text{vc}} = w_{\text{vc}} \sum_{k=1}^{K-1} \|\boldsymbol{\nu}_k\|_1 \tag{4.73}$$

By choosing w_{vc} to have a large magnitude, it can be ensured that the use of virtual control for the converged result is negligible and that the solution can be considered to be dynamically feasible. There are several alternative ways to penalise the use of virtual control, of which using the L_∞-norm is a popular choice. However, the use of the L_1-norm leads to sparse matrices in the convex sub-problem (Section 5.4) and has shown favourable performance in earlier studies of, amongst others, Reynolds et al. (2020a) and Szmuk et al. (2020).

A challenge that was encountered during the Monte Carlo analyses of this study was that for some cases, the virtual control use, although being highly penalised, continued to be larger than the imposed convergence limit. As a result, in these cases, the SCP algorithm failed to converge to a feasible, locally optimal guidance solution. An effective strategy that was implemented to avoid this phenomenon, was the introduction of a *so-called* adaptive virtual control technique. This approach consisted of increasing the weight factor w_{vc} of the virtual control objective for every SCP iteration where the use of virtual control was larger than its convergence criterion (Section 4.9), encouraging the SCP algorithm to reduce its use of virtual control. It was found that multiplying the weight factor w_{vc} with an updating factor of $\alpha_{\text{vc,adap}} = 3$ for such iterations sufficed for improving the convergence rate of the SCP algorithm.

4.7. Trust region

The implementation of a so-called *trust region*, next to the use of a virtual control variable, is a second strategy often seen in combination with an SCP algorithm. A trust region is an additional constraint on the optimisation sub-problem, which is introduced to avoid the undesirable phenomenon of *artificial unboundedness* (Benedikter et al., 2019a). This phenomenon is caused by the fact that the linearised dynamic and kinematic constraints are only a valid approximation in the neighbourhood of the reference solution that forms the basis of the linearisation process. Artificial unboundedness entails that the convex approximation of the original, non-convex problem can become unbounded, while the original problem is not. This phenomenon is illustrated in Figure 4.7.

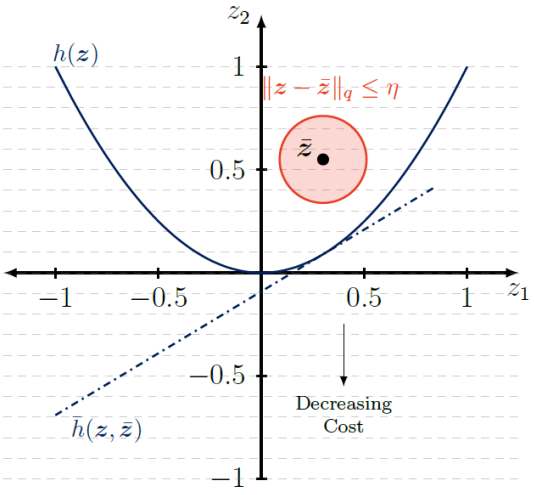


Figure 4.7: Illustration of the phenomenon of artificial unboundedness. It can be observed that the original problem (with the parabolic equality constraint) has a solution at $z_1 = 0$, while the convexified problem (dashed, linear equality constraint) allows for an infinite reduction of the cost, thereby becoming unbounded (Reynolds, 2020).

In Figure 4.7, it can be observed that for the linearised problem, the equality constraint (dash-dotted blue line) allows for an infinite reduction of the cost, rendering the problem unbounded. To deal with this challenge, a trust region imposes a limit on the differences between the optimisation parameters (the state, control, and time dilation factor) of a reference trajectory ($\bar{\boldsymbol{z}}$) and its subsequent solution ($\boldsymbol{z}$) in the SCP algorithm. In addition, the trust region ensures that the approximation error of the solution of the convex sub-problem does not become too large. For the simplified problem of Figure 4.7, the trust region is represented by the red circle. As can one can intuitively understand, selecting a trust region that is too small hinders the convergence process, requiring many iterations in order to arrive at the optimal solution, while selecting a trust region that is too large might lead to undesirable convergence properties. Furthermore, it was found that the limit on $\Delta\boldsymbol{z}$ imposed by the trust region can also increase the performance of an SCP algorithm in terms of

computational efficiency.

In recent studies on SCP-based optimisation algorithms, a wide variety of methods has been proposed to impose such a trust region on the optimisation problem. These range from simpler, *hard* and *fixed* trust regions that have a constant magnitude for all SCP iterations (Liu and Lu, 2014; McDonald et al., 2020), to more complex, *soft* trust regions. These trust regions do not have a fixed magnitude, but instead use the objective function to encourage the SCP algorithm to limit the state difference $\Delta \boldsymbol{z}$ between iterations (Sagliano et al., 2020; Szmuk et al., 2020). An even more complex class of trust regions consists of adaptive schemes, which, depending on certain algorithm parameters, change the size of the trust region after every iteration of the SCP algorithm (Benedikter et al., 2019a; Mao et al., 2018b).

Although there are many studies proposing alternative trust-region (updating) techniques, the number of studies actively comparing several options is, unfortunately, very limited. However, one of the main trends that could be recognised is that, more recently, most researchers tend to use an (adaptive) soft trust region (Benedikter et al., 2020; Bonalli et al., 2019; Reynolds et al., 2020a; Szmuk et al., 2020). Szmuk et al. (2020) noted that such a soft trust region converges faster than the adaptive, hard trust-region technique that was proposed and implemented by Mao et al. (2018b). Therefore, it was decided to implement a soft trust region for this specific study.

A soft trust region does not impose a hard constraint on the maximum difference between two subsequent iterations but rather penalises these differences by augmenting them to the objective function of the convex optimisation sub-problem. To comply with the requirement to have an objective that is a linear function of the optimisation variables, this trust region type is implemented using the equivalent transformation convexification technique that introduces a slack variable, which was presented in Subsection 4.4.3. The resulting objective function term can be formulated as

Trust region objective

$$J_{\text{tr}} = w_{\text{tr}} \cdot s_{\text{tr}} \tag{4.74}$$

where w_{tr} is a weight factor and s_{tr} the slack variable for the trust region. This objective function term indirectly minimises the state and control deviations between two subsequent iterations through the convex constraint

Trust region

$$\sum_{k=1}^{K} (\|(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k)\|_2 + \|(\boldsymbol{u}_k - \bar{\boldsymbol{u}}_k)\|_2) \leq s_{\text{tr}} \tag{4.75}$$

It should be noted that for the implementation of this trust region, scaled variables (Section 4.8) were used to ensure that the trust-region elements corresponding to the $\boldsymbol{q}$, $\boldsymbol{\omega}$ and $\boldsymbol{u}$ terms have a similar magnitude. During the experimentation phase of this research project, it was found that the trust region had a limited influence on the convergence behaviour of the SCP algorithm for the spacecraft reorientation problem studied in this project. This is further discussed in Chapters 7 and 8, where it is shown that the SCP algorithm performs best when the trust region is kept relatively large (with an order of magnitude in the objective function that is significantly smaller than the magnitude of the minimum-energy or minimum-time term). This is thought to be caused by the fact that the spacecraft reorientation problem requires a relatively small number of iterations to converge (around 3-7, as is presented in Chapter 9) compared to more complex optimisation problems, such as the planetary descent problem studied by Wang and Cui (2018). Consequently, it was decided that a detailed comparison of multiple trust-region techniques was not a priority and, therefore, out of scope for this research project.

However, it was found that an adaptive trust-region scheme could improve the robustness (convergence rate) of the SCP algorithm for the minimum-time reorientation problem. This adaptive scheme, in a similar manner as the adaptive virtual-control technique, increases the weight factor w_{tr} of the trust-region objective with a constant factor $\alpha_{\text{tr,adap}}$, after a specified number of iterations, $n_{\text{tr,adap,thres}}$, is reached and the SCP algorithm

has not (yet) converged.

Finally, a hard trust region on the time dilation factor η was found to improve the performance of the SCP algorithm for the minimum-time reorientation problem. Especially for early iterations where the virtual control use is large, it was found that this hard trust region would avoid large differences in the time dilation factor $\Delta\eta$ between iterations (which were sometimes larger than 90%). This hard region on the time dilation factor can be formulated as

Hard trust region on the time dilation factor

$$|\eta - \bar{\eta}| \leq \delta_\eta \bar{\eta} \tag{4.76}$$

where δ_η is an algorithm parameter that directly determines the size of the trust region for the time dilation factor.

4.8. Scaling

During initial tests of the SCP algorithm, it was found that the convex solving algorithm ECOS (Section 5.1) sometimes suffered from numerical problems. Instead of requiring approximately 10-20 iterations to converge, as was the case for the majority of test cases, these numerical problems resulted in ECOS requiring a much larger number of iterations. After the specified limit of 100 iterations was reached, ECOS reported having found a solution that was 'close to optimal'. This phenomenon was highly undesirable, both due to the resulting sub-optimality of the solution and due to the increase in the computation time of multiple factors. It was found that scaling the optimisation parameters solved this problem.

It was decided to scale the optimisation parameters in a manner that is commonly applied in studies on numerical optimisation, which can be represented as (Malyuta et al., 2020)

$$\eta = S_\eta \hat{\eta} + s_\eta, \quad \boldsymbol{q}_k = S_q \hat{\boldsymbol{q}}_k + s_q, \quad \boldsymbol{\omega}_k = S_\omega \hat{\boldsymbol{\omega}}_k + s_\omega, \quad \boldsymbol{u}_k = S_u \hat{\boldsymbol{u}}_k + s_u \tag{4.77}$$

where $S_{(\cdot)}$ and $s_{(\cdot)}$ represent *scaling factors* and *scaling offsets*, respectively, and the $\hat{(\cdot)}$-notation is used to represent the scaled optimisation parameters. It was found that the application of scaling factors S sufficed for solving the numerical problems of ECOS. Therefore, the scaling offsets s were **not** used for the SCP algorithm developed in this study. The scaling factors $S_{(\cdot)}$ that were applied are listed in Table 4.1.

Table 4.1: Scaling factors applied to the spacecraft reorientation problem.

Optimisation parameter	**Symbol**	**Scaling factor**
Time dilation factor	η	$t_{f,\text{guess}}$
Quaternion	$\boldsymbol{q}$	$\boldsymbol{I}_{4\times 4}$
Angular rate	$\boldsymbol{\omega}$	$\text{avg}(\boldsymbol{\omega}_{\text{box}})$
Control	$\boldsymbol{u}$	$\text{avg}(\boldsymbol{u}_{\text{ellipse}})$

where $\boldsymbol{I}$ is the identity matrix, and the average values are *constants* obtained from the constraint vectors that are specified in the formulated problem. It can be observed that, by taking this approach, all scaling factors are constant throughout the iterative solving procedure of the SCP algorithm. Updating these scaling factors after an SCP iteration has been performed is a possibility but was not pursued due to time constraints on this research project. Consequently, it is left as a potential topic for further research.

An additional advantage of the implementation of scaling factors is that several objective function terms, for instance the minimum-time and minimum-energy terms, are automatically scaled. As a result, the performance of the SCP algorithm becomes less dependent on the specific problem parameters of the optimisation problem that it is trying to solve. This also reduces the need to re-tune the SCP algorithm when it is applied to a different optimisation problem, which is characterised by, for instance, a different inertia matrix or stricter angular rate constraints.

It should be noted that this scaling was applied to all elements (constraints, trust region, objective) of the convex optimisation sub-problem that is presented in this section. However, for the sake of readability, the $\hat{(\cdot)}$ notation is dropped in the remainder of this chapter.

4.9. Convergence criteria

The last step of the SCP algorithm is to check whether the algorithm has converged. In this step, it is assessed whether the solution of the convex sub-problem of the most recent iteration of the SCP algorithm can be accepted as a solution to the original, non-convex problem. This step is performed through the definition of a set of so-called *convergence criteria* ϵ.

The purpose of the first convergence criterion is to be able to assess whether the deviation between a guidance solution of iteration $(i+1)$ lies sufficiently close to the solution of iteration i. The reason for this is that this deviation directly influences the linearisation or approximation error that differentiates the original, non-convex problem from the convex sub-problem. If this deviation is found to be sufficiently low, the solution to the convex sub-problem can be considered to adhere to the original, nonlinear dynamics and therefore to be feasible. Furthermore, in experiments it was observed that when this deviation between the solutions of subsequent iterations became small, the cost function of the optimisation problem stopped improving. However, it is noted that the approach taken does not ensure that an optimum has been reached, although the results in Chapter 9 strongly suggest that this is the case.

In literature, many different paths are taken to define such a convergence criterion. During this study, it was found that a straightforward approach succeeded in obtaining reliable and consistent results. For the minimum-energy problem, this resulted in the following convergence criterion, which is defined as

$$\sum_{k=1}^{K} \|\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k\|_1 \leq \epsilon_x \tag{4.78}$$

where $\bar{\boldsymbol{x}}$ refers to the solution of the ith iteration of the SCP algorithm. The convergence criterion from Equation 4.78 ensures that the state difference between iteration $(i+1)$ and i is smaller than a set limit. Due to the applied scaling factors, both the angular rate and quaternion parameters have a similar influence on this convergence criterion. The choice to only include the state $\boldsymbol{x}$ and not the control $\boldsymbol{u}$ was based on the assumption that the magnitude of the state and control differences between subsequent iterations are strongly correlated. It is stressed that the use of the L_1-norm for this convergence criterion is only one of the available options. Alternatives would be, for instance, to use an L_∞-norm on the state (Reynolds et al., 2020a) or to base the convergence criterion on the cost difference between subsequent iterations (Sagliano et al., 2020). As the selection of the type of convergence criterion was no focus of this research project, a comparison between the different options is left as a potential topic for future research.

A second convergence criterion was introduced to ensure that the use of virtual control of the SCP solution is within a predefined limit. This criterion was defined as

$$\sum_{k=1}^{K} \|\boldsymbol{\nu}_k\|_1 \leq \epsilon_\nu \tag{4.79}$$

Using the L_1-norm for this convergence criterion is a popular option (Szmuk et al., 2020), but similar as for Equation 4.78, other approaches could be considered in future research. For the minimum-time problem, a third criterion was introduced to ensure that the linearisation error that follows from the changes in the time dilation factor η between SCP iterations, is smaller than a predefined level. This convergence criterion can be formulated as

$$|\eta - \bar{\eta}| \leq \epsilon_\eta \tag{4.80}$$

When the SCP algorithm has not met its specified convergence criteria after a certain number of iterations, the so-called iteration limit $n_{\text{it,lim}}$, the iterative process is stopped and the problem is reported as *unconverged*.

4.10. Convex sub-problem formulation

In this section, for overview purposes, the full problem description of the minimum-energy and minimum-time convex (sub-)problems is provided, which are solved in every iteration of the SCP-based optimisation algorithm.

4.10.1. Problem 1: Minimum-energy, attitude-constrained reorientation

The convex, discrete, minimum-energy reorientation sub-problem solved sequentially by the SCP algorithm in this research project can be formulated as

Convex, discrete, minimum-energy spacecraft reorientation sub-problem

Objective function	minimise $J = J_u + J_{\text{vc}} + J_{\text{tr}}$	Equation 4.55
		Equation 4.73
	$= w_u s_u + w_{\text{vc}} \sum_{k=1}^{K-1} \lVert \boldsymbol{\nu}_k \rVert_1 + w_{\text{tr}} s_{\text{tr}}$	Equation 4.74
Minimum-energy constraint	$\sum_{k=1}^{K} \boldsymbol{c}_{u,k} \boldsymbol{u}_k^2 \leq s_u$	Equation 4.56
Trust region state and control	$\sum_{k=1}^{K} (\lVert (\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k) \rVert_2 + \lVert (\boldsymbol{u}_k - \bar{\boldsymbol{u}}_k) \rVert_2) \leq s_{\text{tr}}$	Equation 4.75
Boundary conditions	$\boldsymbol{q}(0) = \boldsymbol{q}_0, \quad \boldsymbol{q}(t_f) = \boldsymbol{q}_f$	Equation 3.7
	$\boldsymbol{\omega}(0) = \boldsymbol{\omega}_0, \quad \boldsymbol{\omega}(t_f) = \boldsymbol{\omega}_f$	Equation 3.8
Dynamics and kinematics	$\boldsymbol{x}_{k+1} = A_k \boldsymbol{x}_k + B_k^- \boldsymbol{u}_k + B_k^+ \boldsymbol{u}_{k+1} + \boldsymbol{C}_k \bar{\boldsymbol{\eta}} + \boldsymbol{r}_k + \boldsymbol{\nu}_k,$	Equation 4.72
	$\forall \quad k = 1, \ldots, K-1$	
Control constraints	$\left(\frac{u_{k,1}}{u_{\text{ellipse},1}}\right)^2 + \left(\frac{u_{k,2}}{u_{\text{ellipse},2}}\right)^2 + \left(\frac{u_{k,3}}{u_{\text{ellipse},3}}\right)^2 \leq 1, \quad \forall \quad k \in \mathcal{K}$	Equation 4.68
Angular rate constraints	$\lvert \omega_{k,i} \rvert \leq \omega_{\text{box},i}, \quad \forall \quad i \in 1,2,3, \quad k \in \mathcal{K}$	Equation 4.69
Attitude constraints	$\boldsymbol{q}_k^{\text{T}} \hat{P}_{\text{out}} \boldsymbol{q}_k \leq 2, \quad \forall \quad k \in \mathcal{K}$	Equation 4.70
	$\boldsymbol{q}_k^{\text{T}} \hat{P}_{\text{in}} \boldsymbol{q}_k \leq 2, \quad \forall \quad k \in \mathcal{K}$	Equation 4.71

As noted in Section 4.8, the optimisation parameters presented in this convex sub-problem formulation represent scaled variables $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{u}}$, where the $\hat{(\cdot)}$-notation is dropped for the sake of readability.

4.10.2. Problem 2: Minimum-time, attitude-constrained reorientation

The convex, discrete, minimum-time reorientation sub-problem solved sequentially by the SCP algorithm in this research project can be formulated as

Convex, discrete, minimum-energy spacecraft reorientation sub-problem

Objective function	minimise $J = J_\eta + J_{\text{vc}} + J_{\text{tr}}$	Equation 3.19
		Equation 4.73
	$= w_\eta \eta + w_{\text{vc}} \sum_{k=1}^{K-1} \|\boldsymbol{\nu}_k\|_1 + w_{\text{tr}} s_{\text{tr}}$	Equation 4.74
Trust region state and control	$\sum_{k=1}^{K} (\|(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k)\|_2 + \|(\boldsymbol{u}_k - \bar{\boldsymbol{u}}_k)\|_2) \leq s_{\text{tr}}$	Equation 4.75
Trust region time	$\lvert\eta - \bar{\eta}\rvert \leq \delta_\eta \bar{\eta}$	Equation 4.76
Boundary conditions	$\boldsymbol{q}(0) = \boldsymbol{q}_0, \quad \boldsymbol{q}(t_f) = \boldsymbol{q}_f(\bar{\eta})$	Equation 4.20
	$\boldsymbol{\omega}(0) = \boldsymbol{\omega}_0, \quad \boldsymbol{\omega}(t_f) = \boldsymbol{\omega}_f(\bar{\eta})$	Equation 4.21
Dynamics and kinematics	$\boldsymbol{x}_{k+1} = A_k \boldsymbol{x}_k + B_k^- \boldsymbol{u}_k + B_k^+ \boldsymbol{u}_{k+1} + \boldsymbol{C}_k \eta + \boldsymbol{r}_k + \boldsymbol{\nu}_k,$	Equation 4.72
	$\forall \quad k = 1, \ldots, K-1$	
Control constraints	$\left(\frac{u_{k,1}}{u_{\text{ellipse},1}}\right)^2 + \left(\frac{u_{k,2}}{u_{\text{ellipse},2}}\right)^2 + \left(\frac{u_{k,3}}{u_{\text{ellipse},3}}\right)^2 \leq 1, \quad \forall \quad k \in \mathcal{K}$	Equation 4.68
Angular rate constraints	$\lvert\omega_{k,i}\rvert \leq \omega_{\text{box},i}, \quad \forall \quad i \in 1,2,3, \quad k \in \mathcal{K}$	Equation 4.69
Attitude constraints	$\boldsymbol{q}_k^{\text{T}} \hat{P}_{\text{out}} \boldsymbol{q}_k \leq 2, \quad \forall \quad k \in \mathcal{K}$	Equation 4.70
	$\boldsymbol{q}_k^{\text{T}} \hat{P}_{\text{in}} \boldsymbol{q}_k \leq 2, \quad \forall \quad k \in \mathcal{K}$	Equation 4.71

As noted in Section 4.8, the optimisation parameters presented in this convex sub-problem formulation represent scaled variables $\hat{\boldsymbol{x}}$, $\hat{\boldsymbol{u}}$ and $\hat{\eta}$, where the $\hat{(\cdot)}$-notation is dropped for the sake of readability.

5

Algorithm Implementation

In this chapter, it is explained how the SCP-based optimisation algorithm from Chapter 4 was implemented. To begin with, in Section 5.1, the tools are presented that were used during this research project. Then, Section 5.2 introduces the NLP benchmark algorithm that was implemented to verify and evaluate the performance of the SCP algorithm. In Section 5.3, this NLP benchmark algorithm is verified. Finally, in Section 5.4, an overview is provided of the steps that were taken to implement the convex solving algorithm ECOS.

5.1. Tools

Several tools were used to develop and analyse the SCP-based optimisation algorithm. In Figure 5.1, an overview is provided of these tools. MATLAB functioned as the numerical computing environment that was used to formulate the optimisation problems, interface with the modelling tools, and analyse the resulting attitude guidance trajectories. In terms of the other tools, two main paths can be distinguished:

- **Sequential convex programming (SCP) algorithm**.
  This algorithm is the implementation of the optimisation algorithm that was presented in Chapter 4. MATLAB was used to formulate the convex optimisation sub-problems from Section 4.10. In early stages of this research project, a modelling tool called CVX (Grant and Boyd, 2020) was used to reformulate these convex sub-problems into a form compatible with various convex solving algorithms . The results obtained from the convex solving algorithms were then analysed and visualised using MATLAB. In later stages of the research project, a direct interface to ECOS was developed in MATLAB (as is illustrated by the red arrow in Figure 5.1), removing the need for a modelling tool. This process is further described in Section 5.4.

- **Nonlinear programming (NLP) benchmark algorithm**.
  In order to perform a thorough analysis and evaluation of the performance of the SCP algorithm, a comparison with an NLP benchmark algorithm was determined to be essential. This NLP benchmark algorithm was used to verify the results of the SCP algorithm (Chapter 7), and as a reference point for the performance assessment of the SCP algorithm in terms of the three main criteria identified in Section 3.7: robustness, optimality and computational efficiency. In this study, the CasADi tool (Andersson et al., 2018) was used as modelling tool in order to formulate the nonlinear optimisation problem in a compatible format for the nonlinear solving algorithm Ipopt (Wächter and Biegler, 2005). The NLP benchmark algorithm developed in this study is an efficient pseudospectral NLP method.

In the remainder of this section, these five tools are introduced with a higher level of detail.

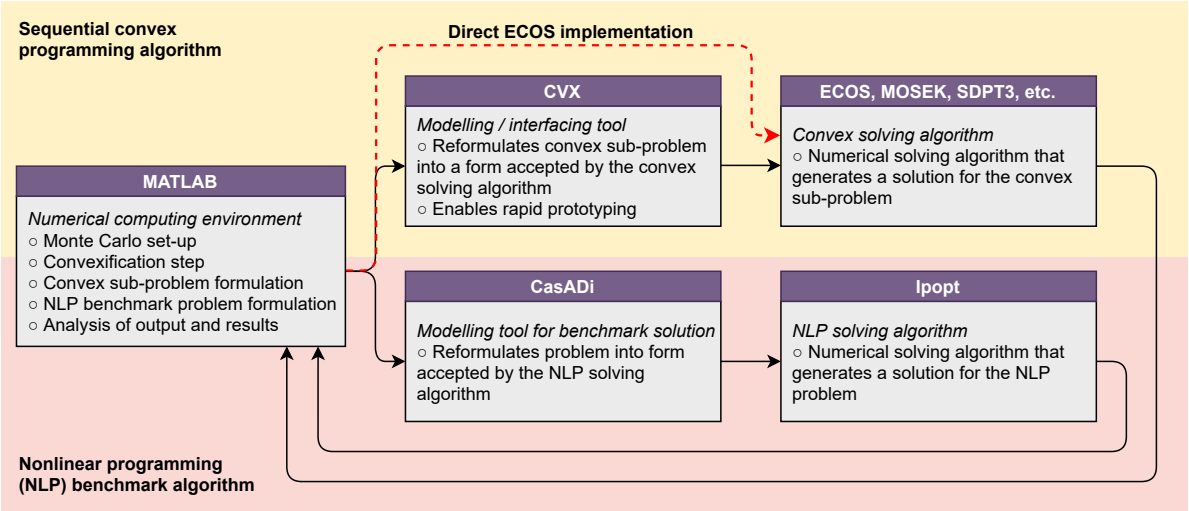


Figure 5.1: Top-level overview of the main tools that were used during this research project.

5.1.1. Numerical computing environment: MATLAB

MATLAB was selected as numerical computing environment. One reason for this is that MATLAB was identified to be compatible with all tools and solving algorithms that were required to implement both the SCP and NLP benchmark algorithms. Secondly, OHB communicated that it was desired that the thesis research would be performed using this programming platform.

For the execution of the discretisation step (Equation 4.42) of the SCP algorithm, which is known to be the second most time-consuming step after the solve step (Reynolds et al., 2020a; Szmuk et al., 2020), the MATLAB Coder package was used. Using this package, it is possible to automatically generate highly efficient C code for functions developed in MATLAB. This does require the corresponding MATLAB code to adhere to a set of rules but greatly simplifies the process of generating code that could be run on embedded (satellite) processors.

This implementation is relevant regarding the development of onboard optimisation algorithms, as these should be executable on such embedded processors using C code. In addition, MATLAB is significantly slower than efficient routines written in C. Through performing the discretisation step using C code it became possible to compare its computation time to the time required for executing the solve step (Section 8.6). To illustrate, using MATLAB, a typical SCP discretisation step is characterised by solve times in the order of 40 ms, while the efficient C code generated by MATLAB Coder only required about 1-2 ms.

5.1.2. SCP modelling tool: CVX

With the term *modelling* or *interfacing tool*, software is referred to that facilitates the modelling of optimisation problems in a relatively easy and intuitive manner. In this study, the convex (sub-)problems to be solved as part of the SCP algorithm were modelled in MATLAB using the tool CVX, a well-known tool that was built to accommodate a variety of convex solving algorithms. Using CVX, the optimisation problems could be formulated in a manner that followed common mathematics, rather than the complicated and restricted forms that are often required by the (convex) solving algorithms themselves. In Table 5.1, an overview is provided of the convex solving algorithm compatibility of CVX.

Table 5.1: Solver compatibility of the modelling tool CVX.

Modelling tool	**Convex solver compatibility**	**Free**
CVX (Grant and Boyd, 2020)	*Free*: SDPT3, SeDuMi, ECOS *Academic license*: GUROBI, MOSEK	Yes

5.1.3. Convex solving algorithm: ECOS

The main advantage of using convexification-based optimisation methods instead of the conventional class of NLP methods is that there exist state-of-the-art, highly efficient algorithms which can solve these problems in real time (Liu et al., 2017). These algorithms belong to the class of interior point methods (IPMs). After a convex (sub)-problem has been formulated, either directly by the user or through the modelling tool CVX presented in the previous subsection, this problem can be solved using one of several available convex solving algorithms.

Regarding these IPMs, a distinction is made between general-purpose solvers, which can be implemented on modern desktop computers, and so-called *embedded* solvers. In order to use an SCP-based optimisation algorithm for a real mission scenario, it has to be executable on an embedded flight processor (Dueri et al., 2016). The corresponding process, which is called embedded convex optimisation, needs to be both robust and fast, in line with the criteria identified in Section 3.7. It needs to be robust because there are no human operators that could intervene if a failure would occur during onboard operations. In contrast, general-purpose solvers used for development and testing purposes might have lower requirements on robustness. Computational efficiency is essential as the computational resources of these embedded systems are limited, while the optimisation process is required to take place in real time. Spacecraft typically use flight-hardened processors, which have architectural features that make them robust to the high-radiation environment in space. However, this comes at the cost of computational efficiency.

The five most popular convex solving algorithms considered for this study are: ECOS, MOSEK (Andersen and Andersen, 2000), SDPT3 (Tütüncü et al., 2001), and SeDuMi. However, it was quickly concluded from existing research that ECOS outperforms the other three solvers for the problem size associated with the spacecraft reorientation problem (Sagliano, 2018; Yang and Liu, 2020), which was confirmed by initial tests during our research. A second consideration is that ECOS is written in low-footprint, library-free ANSI-C, which enables it to run on embedded platforms. This is in contrast to the other SOCP solvers that are currently available, which are typically characterised by a large code size, large memory requirements, and dependencies on external libraries (Domahidi et al., 2013). Therefore, the decision was made to use ECOS for the analyses performed in this study. ECOS, known as the *Embedded Conic Solver*, is a relatively novel IPM that uses a standard primal-dual Mehrotra predictor-corrector method with Nesterov-Todd scaling and self-dual embedding techniques for SOCP optimisation problems (Domahidi et al., 2013). ECOS employs sparse linear algebra routines which make it one of the fastest methods currently available for solving convex conical problems (Wang, 2019). ECOS is publicly available on GitHub[1] and can be directly integrated into MATLAB and CVX.

5.1.4. NLP modelling tool: CasADi

In order to have a benchmark that delivers state-of-the-art performance, the decision was made to use a tool that is currently used by research partner OHB: CasADi. CasADi is an open-source tool that can solve NLP problems through interfacing to a variety of popular NLP solvers, such as Ipopt, BlockSQP, KNITRO, and SNOPT (Andersson et al., 2018). In this sense, in a similar manner as CVX, it allows for rapid and straightforward modelling of nonlinear optimisation problems.

It is noted that the commercially available MATLAB-based GPOPS-II tool was often encountered in studies on both SCP-based optimisation and on optimal spacecraft reorientation problems. GPOPS-II is typically presented as the state-of-the-art tool used by researchers to obtain optimal reference solutions for complex NLP problems. GPOPS-II applies Legendre-Gauss-Radau quadratic orthogonal collocation methods to transcribe the continuous optimisation problem into a finite-dimensional numerical NLP problem. An adaptive hp-pseudospectral mesh refinement strategy is employed to determine the number of mesh intervals and the degree of the approximating polynomial in each interval. The resulting problem is then solved using known and proven NLP solvers, such as Ipopt and SNOPT (Wang and Grant, 2016).

However, it was decided to use CasADi for this research project due to the fact that this tool is open-source and that it is currently being used at OHB. Compared to computation times reported for GPOPS-II, it is expected that CasADi delivers relatively high computational performance.

[1] https://github.com/embotech/ecos. Accessed on: 11-06-2021.

5.1.5. NLP solving algorithm: Ipopt

Ipopt, also known as 'Interior point optimiser', is a popular NLP solving algorithm for large-scale nonlinear optimisation algorithms (Wächter and Biegler, 2005). It implements a primal-dual interior point method, uses (Leyffer and Fletcher) filter methods, and is well known to be one of the fastest NLP solving algorithms (Mao et al., 2018b; Sagliano, 2019).

5.2. NLP benchmark algorithm

In this section, the main characteristics of the NLP benchmark algorithm are presented and discussed. Although this optimisation algorithm is not the focus of this study, it is important to have a good understanding of the characteristics and performance of this method to make a valid comparison between the performance of the SCP and NLP algorithms. This section is divided into two parts. First, in Subsection 5.2.1, the discretisation method is presented, followed by an elaboration regarding the enforcement of the constraints that are imposed on the reorientation problem in Subsection 5.2.2.

5.2.1. Discretisation

The NLP benchmark algorithm that was used to obtain benchmark solutions for this study uses a so-called *pseudospectral* (PS) transcription method. The class of pseudospectral NLP methods is well-known for having highly efficient performance characteristics for a wide variety of nonlinear, non-convex optimisation problems (Sagliano, 2018).

There are a large number of families of pseudospectral transcription methods based on various sets of nodes, such as the flipped Radau (fRPM), Lobatto (LPM), Chebyshev (Fahroo and Ross, 2002), or Gauss pseudospectral methods (Sagliano, 2018). As the goal of this research project is not to identify the single-best pseudospectral method, but rather to develop a proper benchmark for the evaluation of the SCP algorithm, the decision was made to use the proven Chebyshev-Gauss-Legendre (CGL) transcription method used by research partner OHB. For the sake of completeness, a general description of pseudospectral transcription methods is provided in this subsection.

The main, shared characteristic of pseudospectral transcription methods is that they model and approximate the state $\boldsymbol{x}$ and control $\boldsymbol{u}$ parameters with Lagrange polynomials. For all pseudospectral methods, a set $\mathscr{D} := \{\eta_i \in [-1,1]\}_{i=0}^{N}$ of discretisation nodes is defined that collectively form a non-uniform grid (Malyuta et al., 2019). In this set, η is referred to as the pseudo-time. These nodes η_i are usually defined on the interval $[-1,1]$. The pseudo-time η is transformed to real time t through

$$t(\eta) = \frac{t_f(\eta+1)}{2} \tag{5.1}$$

where t_f represents the final time of the manoeuvre. The state and control variables are approximated as follows

$$\boldsymbol{x}(\eta) = \sum_{i=0}^{N} \boldsymbol{x}_i \phi_i(\eta), \quad \boldsymbol{u}(\eta) = \sum_{i=0}^{N} \boldsymbol{u}_i \phi_i(\eta), \quad \text{where} \quad \phi_i(\eta) := \prod_{\substack{j=0 \\ j\neq i}}^{N} \frac{\eta-\eta_j}{\eta_i-\eta_j} \tag{5.2}$$

The functions ϕ_i are known as Lagrange interpolation basis polynomials of degree N. Furthermore, a set of M collocation nodes $\mathscr{C} \subseteq \mathscr{D}$ is defined for $M \leq N+1$. These collocation nodes are defined as the nodes where the differential equations are imposed on the state and control variables. The specific set of collocation nodes defines a certain pseudospectral method and forms the main difference with respect to the other methods (Malyuta et al., 2019). To illustrate, there can be differences in the number of nodes and their location on the interval $[-1,1]$. For every combination of $\mathscr{C}$ and $\mathscr{D}$, there is an associated pseudospectral differentiation matrix $D \in \mathbb{R}^{M\times(N+1)}$ (Garg, 2011), which imposes the differential (in this case: dynamic and kinematic) constraints on the finite-dimensional basis of the Lagrange polynomials

$$\dot{\boldsymbol{x}}'(\eta_j) = \sum_{i=0}^{N} \boldsymbol{x}_i \phi_i'(\eta_j) = \sum_{i=0}^{N} D_{ji} \boldsymbol{x}_i, \quad \forall \quad \eta_j \in \mathscr{C} \tag{5.3}$$

where j refers to the considered collocation node. As the differential $\dot{\boldsymbol{x}}'$ is defined with respect to pseudo-time η in Equation 5.3, a temporal transformation has to be applied to relate the real-time differential equations

of the system to the pseudo-time differential operator:

$$\dot{\boldsymbol{x}}' = \frac{d\boldsymbol{x}}{d\eta} = \frac{d\boldsymbol{x}}{dt}\frac{dt}{d\eta} = \frac{t_f}{2}\frac{d\boldsymbol{x}}{dt} = \frac{t_f}{2}\dot{\boldsymbol{x}} \tag{5.4}$$

At this point, the relation between the original system dynamics and the pseudospectral differential operator can be obtained in the form

$$\dot{\boldsymbol{x}}(\eta_j) = \frac{2}{t_f}\sum_{i=0}^{N} D_{ji}\boldsymbol{x}_i, \quad \forall \quad \eta_j \in \mathscr{C} \tag{5.5}$$

Put differently, this relation implies that the state derivative at a single point can be approximated by a certain combination of the values of the state $\boldsymbol{x}$ at other discretisation nodes. Using barycentric Lagrange interpolation (a fast and stable variant of Lagrange polynomial interpolation), the differentiation matrix D can be computed within machine-rounding error (Berrut and Trefethen, 2004; Malyuta et al., 2019; Sagliano, 2018). It must be noted, that to account for the non-uniformity of the pseudospectral grid, a so-called pseudospectral weight vector for Clenshaw–Curtis quadrature $\boldsymbol{w}_{\text{ps}}$ should be introduced to the discretised objective function to accurately represent integrands (Sagliano et al., 2020):

$$J = w_\mu \cdot \boldsymbol{w}_{\text{ps}}^T \boldsymbol{\mu} \tag{5.6}$$

where J is the objective funciton, w_μ is a standard weight factor, and $\boldsymbol{\mu} \in \mathbb{R}^{N+1}$ is some cost variable or function.

5.2.2. Constraint enforcement

The most straightforward way to enforce the problem constraints is to impose them on the grid nodes. However, this does not ensure that the constraints are satisfied on the inter-nodal segments of the grid. This forms especially a challenge for the (CGL) pseudospectral NLP method implemented in this study, as its non-uniform grid and corresponding large inter-nodal segments could lead to significant constraint violations. To illustrate, in Figure 5.2, the grid of the CGL collocation method is shown for 15 grid points, a number that was often used during this research project.

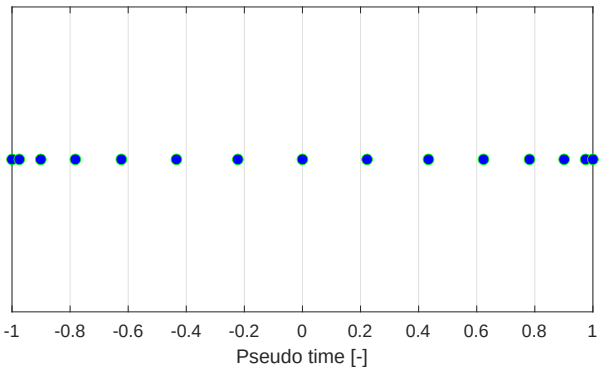
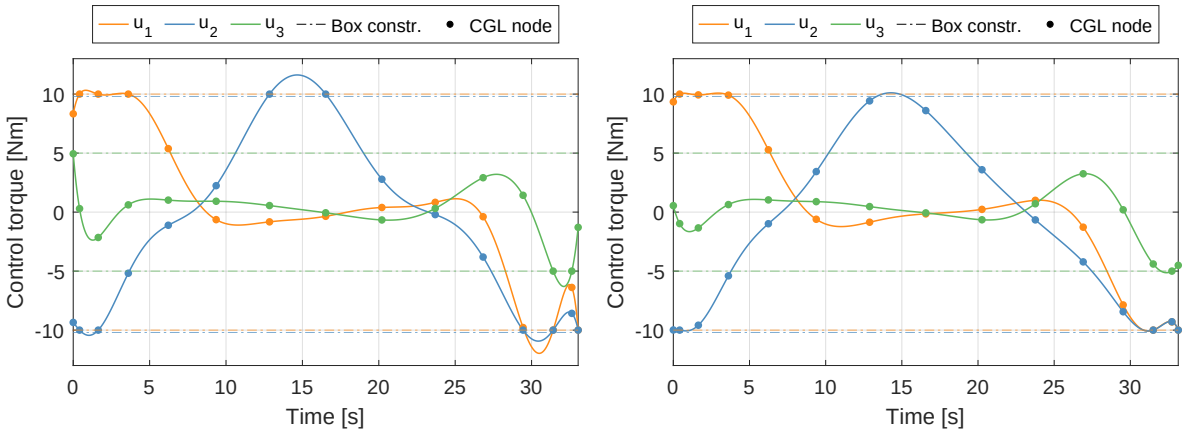


Figure 5.2: Discretisation nodes of the CGL grid.

To deal with this challenge, additional constraints were imposed on the problem in CasADi. Using a Lagrange interpolation of several optimisation parameters, the box constraint on the angular rate and the ellipsoidal constraint on the control were imposed on 30 linearly-spaced nodes. This was decided to enable a fair comparison with the SCP algorithm developed in this study, because its first-order-hold (FOH) control modelling (Subsection 4.5.2) prevents any control violations whatsoever. In addition, the box constraint on the angular rate was eventually enforced at 29 grid points for the SCP algorithm, as explained in Section 8.3. In order to ensure that both the NLP and SCP algorithms impose angular rate constraints in a similar manner, for the NLP algorithm it was decided to only enforce the rate constraints at the 30 linearly-spaced nodes (in hindsight, 29 would have led to a slightly fairer comparison), and not on the CGL discretisation nodes. The effectiveness of this approach is illustrated in Figure 5.3, where the NLP control solution for a minimum-time, attitude-constrained reorientation manoeuvre is shown, with and without the enforcement of the 30 linearly-spaced

constraints. As visualised, this reference problem formulation includes a $[10 \quad 10 \quad 5]^T$ box constraint on the control.



(a) NLP benchmark control solutions obtained when the control constraint is only enforced at the (CGL) discretisation nodes.

(b) NLP benchmark control solutions obtained when the control constraint is also enforced at 30 linearly-spaced discretisation nodes.

Figure 5.3: Control solutions obtained using the NLP benchmark algorithm for a reference minimum-time reorientation problem.

As can be observed in Figure 5.3, without enforcing the additional constraints (Figure 5.3a) one ends up with NLP control profiles that violate the box constraint. These profiles, in reality, would not be executable through the physical limitations of an attitude control system. Furthermore, the performance comparison performed in this study would be less valid, as the control violations could lead to shorter manoeuvre durations than would be feasible considering all constraints. As the SCP algorithm models the inter-nodal control in a linear fashion, it does not face this same problem for this specific variable. However, as discussed in Chapter 7, inter-nodal constraint violations are a challenge for the angular rate, attitude keep-out and attitude keep-in constraints.

5.3. NLP benchmark algorithm verification and validation

As the NLP benchmark algorithm has a vital role in this study as reference point for the performance analysis of the SCP algorithm, it is critical to verify that the NLP method that was implemented during this research project delivers both correct and optimal results. To this end, in this section, three relevant reference cases from literature are analysed. Then, the results obtained from the NLP algorithm are compared with the results that were reported in the original studies.

5.3.1. Minimum-energy reorientation

Ventura et al. (2015) used the GPOPS-II solver in order to obtain a benchmark for the minimum-energy spacecraft reorientation problem. This test case was selected because of the use of the popular and proven GPOPS-II solver and because Ventura et al. (2015) verified their results with the findings of Boyarko et al. (2011), who used the same test case to benchmark a novel algorithm for optimal spacecraft reorientation. Furthermore, both studies discussed their problem set-up and results to the extent that they could be replicated in this study. The parameters (and results) of this test case are provided in Table 5.2.

Table 5.2: Parameters of the minimum-energy reference test case (Ventura et al., 2015).

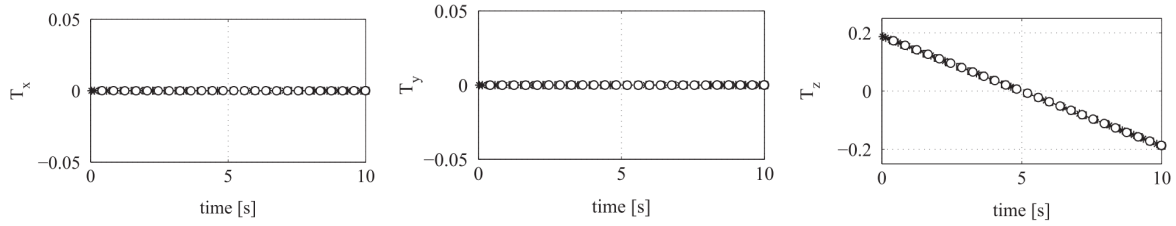
Parameter	**Value**	**Unit**
Inertia matrix $\boldsymbol{I}$	I_3	$kg\,m^2$
Maximum angular velocity $\boldsymbol{\omega}_{max}$	Unconstrained	-
Maximum control torque $\boldsymbol{u}_{max}$	$[1,1,1]^T$	Nm
Objective	Minimum-energy	-
Number of grid nodes	100	-
Initial attitude $\boldsymbol{q}_0$	$[0,0,0,1]^T$	-
Final attitude $\boldsymbol{q}_f$	$\left[0,0,\sin\frac{1}{2}\phi,\cos\frac{1}{2}\phi\right]^T$, with $\phi = 180°$	-
Initial angular rate $\boldsymbol{\omega}_0$	$[0,0,0]^T$	rad/s
Final angular ate $\boldsymbol{\omega}_f$	$[0,0,0]^T$	rad/s
Manoeuvre duration t_f	10	s
Cost j	$5.922{\cdot}10^{-2}$	-

In Table 5.3, the results are provided that were obtained through solving the optimisation problem defined by the parameters from Table 5.2 with the NLP algorithm that was implemented in this study.

Table 5.3: Benchmark verification results of the minimum-energy test case.

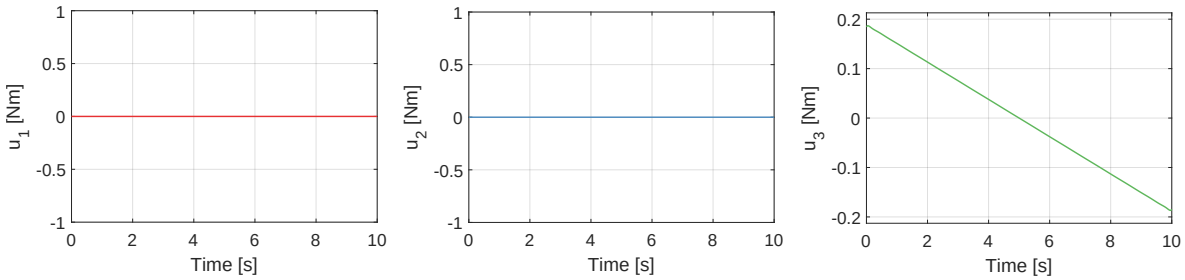
Method	**Cost J**	**Difference [%]**
GPOPS (Ventura et al., 2015)	$5.922 \cdot 10^{-2}$	-
NLP benchmark algorithm (CGL PS method)	$5.920 \cdot 10^{-2}$	$-3.38 \cdot 10^{-2}$

As can be observed in Table 5.3, the solution of the NLP benchmark algorithm is almost equal to the reference solution. The small difference could be explained by several factors. For instance, it could be explained by the fact that GPOPS-II uses a different (hp-adaptive) grid than the NLP benchmark algorithm. Another cause could be the method that was used to integrate the obtained control profiles to obtain the energy (cost). However, for the purpose of serving as the benchmark for this study, this difference is more than acceptable, as the expected differences between the SCP and NLP benchmark algorithms in terms of optimality are significantly higher. For the purpose of visualising the verification process, Figures 5.4 and 5.5 show the control profiles that were obtained using both methods.



(a) u_1 control profile (referred to with T_x). (b) u_2 control profile (referred to with T_y). (c) u_3 control profile (referred to with T_z).

Figure 5.4: Reference control profiles for the minimum-energy reorientation test case from Table 5.2 (Ventura et al., 2015).



(a) u_1 control profile. (b) u_2 control profile. (c) u_3 control profile.

Figure 5.5: Control profiles for the minimum-energy reorientation test case from Table 5.2 which were obtained using the NLP benchmark algorithm.

The control profiles of both methods can be seen to match closely. In addition, it can be observed that only the u_3 control parameter has a non-zero magnitude, which could have been expected since a symmetric inertia matrix $\boldsymbol{I}$ was used, and the specified manoeuvre represents a 180° rotation around the body-fixed z-axis.

5.3.2. Minimum-time reorientation

Both Ventura et al. (2015) and Boyarko et al. (2011) also extensively studied the minimum-time attitude guidance problem. Table 5.4 presents an overview of the parameters and the obtained results of one of the test cases that were analysed in these studies. Again, this problem was solved using the reliable and proven GPOPS-II tool.

Table 5.4: Parameters of the minimum-time reference test case (Boyarko et al., 2011).

Parameter	**Value**	**Unit**
Inertia matrix $\boldsymbol{I}$	I_3	kg m^2
Maximum angular velocity $\boldsymbol{\omega}_{\max}$	Unconstrained	-
Maximum control torque $\boldsymbol{u}_{\max}$	$[1,1,1]^T$	Nm
Objective	Minimum-time	-
Number of grid nodes	100	-
Initial attitude $\boldsymbol{q}_0$	$[0,0,0,1]^T$	-
Final attitude $\boldsymbol{q}_f$	$\left[0,0,\sin\frac{1}{2}\phi,\cos\frac{1}{2}\phi\right]^T$, with $\phi = 180°$	-
Initial angular velocity $\boldsymbol{\omega}_0$	$[0,0,0]^T$	rad/s
Final angular velocity $\boldsymbol{\omega}_f$	$[0,0,0]^T$	rad/s
Cost t_f	3.243	s

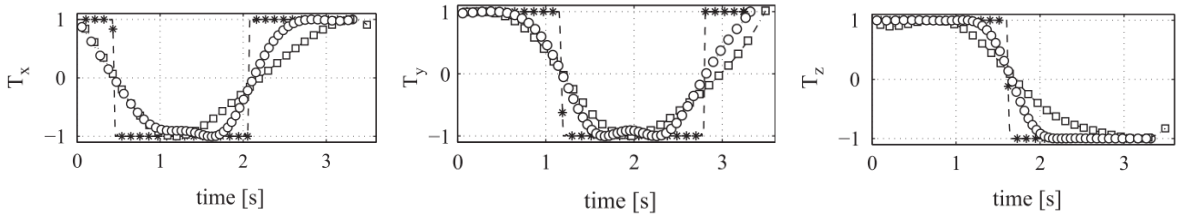
It can be noted that the problem that is presented in Table 5.4 only differs from the minimum-energy problem from Subsection 5.3.1 with respect to the objective of the optimisation problem (minimum-time instead of minimum-energy). Table 5.5 shows the results that were obtained using the different optimisation algorithms.

Table 5.5: Benchmark verification results of the minimum-time test case (Ventura et al., 2015).

Method	**Cost $J = t_f$ [s]**	**Difference [%]**
GPOPS (Boyarko et al., 2011)	3.243	-
GPOPS (Ventura et al., 2015)	3.243	-
NLP benchmark algorithm (CGL PS method)	3.2599	0.5

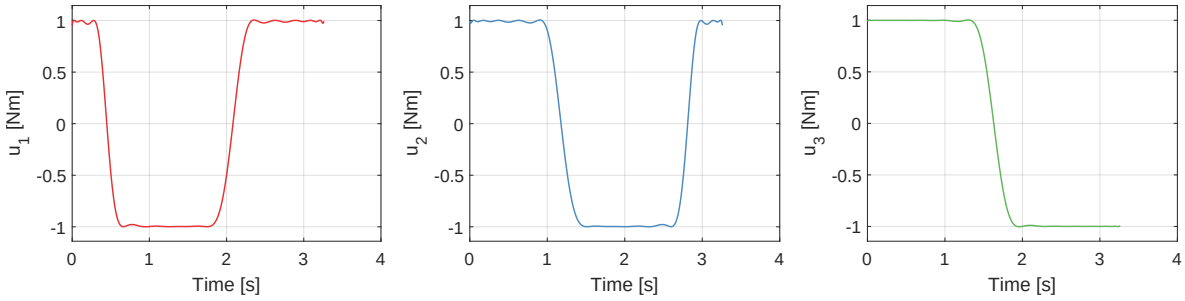
It can be seen that the difference between the cost of the reference algorithm and the NLP benchmark algorithm is higher than for the minimum-energy problem. However, this difference can be explained by the fact that the NLP benchmark algorithm used only 30 instead of 100 grid points. As can be seen through comparing the control profiles shown in Figures 5.6 and 5.7, this causes the solution of the NLP algorithm to be characterised by more gradual changes in the control profiles at the switch points between the maximum and minimum control magnitudes. As a result, the discrete steps in the optimal control profiles cannot be represented as accurately. The reason for only using 30 grid points was that the NLP algorithm was found to become unstable for large numbers of grid points (more than 40). However, as the NLP algorithm in this study only uses 10 - 20 discretisation nodes, this was found not to pose a problem for its application as benchmark algorithm.

Because the SCP algorithm found the optimal solution to the reorientation test case that corresponds to its number of grid points K_{NLP}, the NLP benchmark algorithm was considered to be verified.



(a) u_1 control profile (referred to with T_x). (b) u_2 control profile (referred to with T_y). (c) u_3 control profile (referred to with T_z).

Figure 5.6: Reference control profiles for the minimum-time reorientation test case from Table 5.4 (Ventura et al., 2015). The different profiles and symbols correspond to the three different optimisation methods that were studied. The control solution with the ⋆-symbols represents the reference solution of the GPOPS-II solver that is of interest to this verification analysis.



(a) u_1 control profile. (b) u_2 control profile. (c) u_3 control profile.

Figure 5.7: Control profiles for the minimum-time reorientation test case from Table 5.4, which were obtained using the NLP benchmark algorithm. The oscillatory behaviour of these control profiles is known as the Gibbs phenomenon.

5.3.3. Minimum-energy, attitude keep-out constrained reorientation

In addition to the attitude-unconstrained minimum-energy and minimum-time verification test cases studied in the previous subsections, a final test case was performed in order to verify (1) that the keep-out constraints in this study are correctly formulated and (2) that the NLP benchmark algorithm satisfies this type of constraints. To this end, a suitable minimum-energy, attitude keep-out constrained spacecraft reorientation problem was identified, which was solved using a variety of methods in studies of Spiller et al. (2018), Virgili-Llop et al. (2018), and Virgili-Llop et al. (2018). The fact that this case had been solved by multiple authors was determined to provide enough confidence regarding its correctness. The problem parameters of this test case are provided in Table 5.6.

Table 5.6: Parameters of the minimum-energy, attitude keep-out constrained verification test case (Virgili-Llop et al., 2018).

Parameter	**Value**	**Unit**
Inertia matrix $\boldsymbol{I}$	diag([3000 4500 6000])	kg m^2
Maximum angular velocity $\boldsymbol{\omega}_{\max}$	Unconstrained	-
Maximum control torque $u_{\max}$	0.25	Nm
Objective	Minimum-energy	-
Number of grid nodes	25	-
Instrument boresight $\boldsymbol{x}_B$	$[0,-\sin\xi,-\cos\xi]^T$ with $\xi = 38°$	-
Exclusion cone 1, $\boldsymbol{y}_{\text{out},1}$	$[\cos\alpha\cos\beta,\cos\alpha\sin\beta,\sin\alpha]^T, \alpha = -54°, \beta = -171.9°, \theta_1 = 40°$	-
Exclusion cone 2, $\boldsymbol{y}_{\text{out},2}$	$[\cos\alpha\cos\beta,\cos\alpha\sin\beta,\sin\alpha]^T, \alpha = -64.8°, \beta = -18°, \theta_2 = 19°$	-
Initial attitude $\boldsymbol{q}_0$	$[0,0,0,1]^T$	-
Final attitude $\boldsymbol{q}_f$	$[0.866, 0.500, 0, 0]^T$	-
Initial angular velocity $\boldsymbol{\omega}_0$	$[0,0,0]^T$	rad/s
Final angular velocity $\boldsymbol{\omega}_f$	$[0,0,0]^T$	rad/s
Manoeuvre duration t_f	$10\,\tau_t$	s
Cost J	$0.01631\,\tau_u^2\tau_t$	-

In Table 5.6, it can be seen that several problem parameters are formulated in a different manner than in

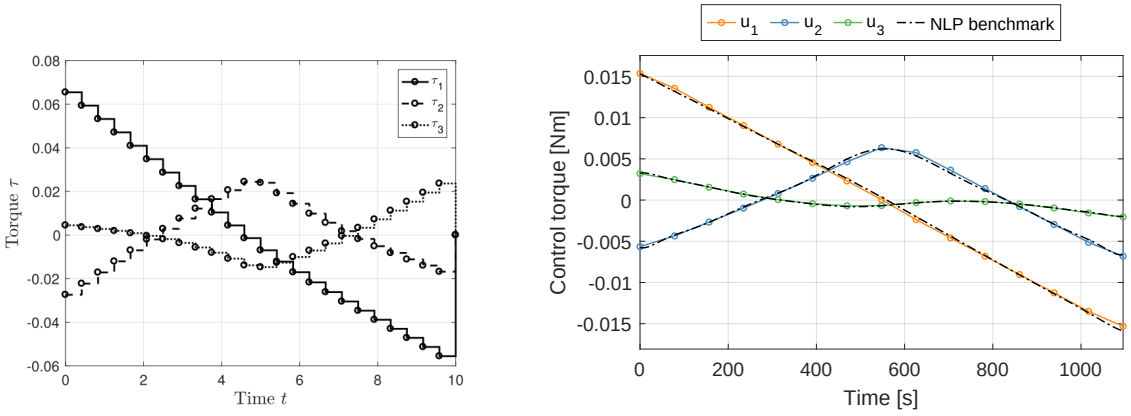
other parts of this work. For the purpose of reproducibility, it was decided to adhere to the original formulation of Virgili-Llop et al. (2018) as much as possible. In line with the original formulation, $\tau_u = u_{\max}$ and $\tau_t = \sqrt{I_x / u_{\max}}$ represent normalisation factors for the control and time parameters. Finally, it is noted that these reference studies used the Classical Rodrigues attitude parameterisation. Therefore, $\boldsymbol{q}_0$ and $\boldsymbol{q}_f$ had to be obtained using a parameter conversion process. Table 5.7 presents the (performance) results that were obtained using the different optimisation methods. For the sake of completeness, the solution of the SCP algorithm developed in this study was also included. However, the performance of this algorithm is not discussed in this subsection, as it is extensively studied in Chapters 7 to 9.

Table 5.7: Benchmark verification results of the minimum-energy, attitude keep-out constrained test case.

Method	**Cost J [-]**	**Difference [%]**
GPOPS (Virgili-Llop et al., 2018)	0.01631 $\tau_u^2 \tau_t$	0
PSO (average) (Spiller et al., 2018)	0.01628 $\tau_u^2 \tau_t$	-0.18
SCP reference (Virgili-Llop et al., 2018)	0.01643 $\tau_u^2 \tau_t$	0.74
NLP Benchmark CGL-PS method (NLP benchmark)	0.01635 $\tau_u^2 \tau_t$	0.26
SCP (this study)	0.01635 $\tau_u^2 \tau_t$	0.26

In Table 5.7, it can be observed that all results are very similar. The main takeaway is that the NLP benchmark algorithm delivers practically the same cost as the GPOPS-II NLP solver that was used in both reference studies. The fact that there is a slight difference between the GPOPS and NLP benchmark solutions is not a major issue, as the focus of this study is rather to assess whether the SCP algorithm can provide solutions that are optimal (in the order of a few percent), than assessing whether that solution is optimal within 0.1%. There could be a variety of causes for the difference in optimality that can be observed. It could result, for instance, from the fact that the reference studies use an hp-adaptive pseudospectral discretisation method, Classical Rodrigues parameters instead of quaternions, an alternative approach to calculating the final performance, constraint violations, or, perhaps, a different approach to formulating and enforcing the conical keep-out constraints.

To visualise these results, in Figure 5.8, the optimal control profiles are shown for the SCP approach from Virgili-Llop et al. (2018) (Figure 5.8a), the NLP benchmark algorithm (Figure 5.8b), and the SCP algorithm developed in this study (Figure 5.8b, coloured lines). The control profiles of the reference SCP algorithm of Virgili-Llop et al. (2018) are shown (instead of the more optimal profiles of the GPOPS-II algorithm), as these were the only profiles provided in the reference studies.



(a) Control solution of the reference SCP algorithm developed in the study of Virgili-Llop et al. (2018). The torque and time are dimensionless.

(b) Control solution of the SCP and NLP benchmark algorithms developed in this study.

Figure 5.8: Control profiles for the minimum-energy, keep-out constrained reorientation test case from Table 5.7 which were obtained using the SCP reference, the NLP benchmark and the SCP algorithms. It should be noted that the axes of Figure 5.8a are dimensionless while the axes of Figure 5.8b are not.

The similarities between the control profiles in Figure 5.8 confirm that the the NLP benchmark algorithm found a locally optimal solution. It can be seen in Figure 5.8a that Virgili-Llop et al. (2018) imposed a zero-order-hold on the control as part of the SCP algorithm that was developed in their study, in contrast to the

first-order-hold modelling implemented in this study. It is thought to be likely that this feature is one of the main causes of the performance difference between the optimality of their SCP algorithm and the optimality of the corresponding GPOPS-II solution in Table 5.7. Interestingly, while the obtained u_1 and u_2 control profiles can be observed to match closely between the solutions of the NLP benchmark and the reference SCP methods, the u_3 control can be observed to differ significantly.

5.4. Direct ECOS implementation

As was shown in Figure 5.1, initially, the convex optimisation sub-problems of the SCP algorithm were reformulated into an ECOS-compatible format using the modelling tool CVX. Using such a modelling (or interfacing) tool allows for rapid development, testing and many design iterations. However, the process of reformulating an optimisation problem into an ECOS-compatible format, as performed by CVX, is extremely slow, with run times that are typically three orders of magnitude larger (seconds) than the computation times of ECOS (milliseconds). Furthermore, in contrast to ECOS, CVX cannot be run on embedded hardware, which is a prerequisite for onboard, autonomous applications of the SCP algorithm. Therefore, it was considered an interesting research direction to get a thorough understanding of how the CVX step could be eliminated and how ECOS could be implemented directly into the SCP algorithm. Thereby, the SCP algorithm would be brought one step closer to actual implementation.

In this section, it will be discussed how the convex spacecraft reorientation sub-problems were reformulated into a form that could be solved directly with the ECOS convex solving algorithm. This direct ECOS implementation has only been performed in a limited number of studies in the context of SCP and has practically never been documented in a detailed manner. Therefore, it was decided that for future work, it would be helpful to describe the process of implementing ECOS directly with a high level of detail.

5.4.1. ECOS requirements

In order to understand how the optimal spacecraft reorientation problems from Subsections 4.10.1 and 4.10.2 can be solved without resorting to a modelling tool such as CVX, the first step is to understand the type of input which is required by ECOS. ECOS solves problems of the type

$$\begin{aligned} &\text{minimise} \quad \boldsymbol{c}_0^{\mathrm{T}} \boldsymbol{x}_e \\ &\text{subject to} \quad A\boldsymbol{x}_e = \boldsymbol{b}, \quad G\boldsymbol{x}_e \preceq_K \boldsymbol{h} \end{aligned} \tag{5.7}$$

where the cone K can represent the product of three different types of cones (Domahidi and Jerez, 2015; Reynolds et al., 2020a):

- $K \triangleq \mathbb{R}^n_+$, where $\mathbb{R}^n_+ \triangleq \{x \in \mathbb{R}^n \mid x \geq 0\}$ is a linear cone of dimension n.
- $Q_1^{n_1} \times Q_1^{n_2} \times \cdots \times Q_M^{n_M}$, where $Q^n \triangleq \{(t,x) \in \mathbb{R}^{d_{Q,n}} \mid \|x\|_2 \leq t\}$ is a second-order cone of dimension $d_{Q,n}$.
- $\mathscr{E}_1 \times \mathscr{E}_2 \times \ldots \mathscr{E}_N$, where every $\mathscr{E} \triangleq \mathrm{cl}\{(x,y,z) \in \mathbb{R}^3 \mid e^{x/z} \leq y/z, z > 0\}$ is an exponential cone.

Put differently, the $G\boldsymbol{x}_e \preceq_K \boldsymbol{h}$ inequality constraint can incorporate a range of different conical inequality constraints, including linear, (reformulated) quadratic and SOCP constraints. For the formulation of the spacecraft reorientation problem considered in this study, only the linear and second-order cone constraint types are used. The problem formulation presented in Equation 5.7 is exactly how the convex sub-problems, which need to be solved in the SCP algorithm, have to be formulated. In other words, an approach had to be found to include all elements of the optimal spacecraft reorientation problem, such as the boundary conditions, constraints, dynamics and objective function, in the vectors $\boldsymbol{c}_0^{\mathrm{T}}$, $\boldsymbol{b}$ and $\boldsymbol{h}$, and the matrices A and G. In this section, this process is outlined for the minimum-time, attitude-constrained reorientation problem from Subsection 4.10.2. This reformulation process is very similar for the minimum-energy reorientation problem, so it was decided to only highlight the major differences in this section for the sake of conciseness.

5.4.2. Optimisation parameters

For the direct ECOS implementation, all optimisation parameters, including slack variables, need to be combined into a single vector $\boldsymbol{x}_e$, in line with the problem format from Equation 5.7. For the minimum-time

reorientation problem of Subsection 4.10.2, this results in the following vector of optimisation parameters:

$$\boldsymbol{x}_e = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \\ \eta \\ \boldsymbol{\nu}^+ \\ \boldsymbol{\nu}^- \\ \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_u \\ \boldsymbol{\lambda}_x \\ \boldsymbol{\lambda}_u \end{bmatrix} \tag{5.8}$$

where

$$\boldsymbol{x} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,7} & x_{2,1} & \dots & x_{K,n_x} \end{bmatrix}^T \tag{5.9}$$

$$\boldsymbol{u} = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & u_{2,1} & u_{2,2} & \dots & u_{K,n_u} \end{bmatrix}^T \tag{5.10}$$

$$\eta = \eta \tag{5.11}$$

$$\boldsymbol{\nu}^+ = \begin{bmatrix} \nu_{1,1} & \nu_{1,2} & \dots & \nu_{1,7} & \nu_{2,1} & \dots & \nu_{K,n_x} \end{bmatrix}^T \tag{5.12}$$

$$\boldsymbol{\nu}^- = \begin{bmatrix} \nu_{1,1} & \nu_{1,2} & \dots & \nu_{1,7} & \nu_{2,1} & \dots & \nu_{K,n_x} \end{bmatrix}^T \tag{5.13}$$

$$\boldsymbol{\mu}_x = \begin{bmatrix} \mu_{x,1,1} & \mu_{x,1,2} & \dots & \mu_{x,1,7} & \mu_{x,2,1} & \dots & \mu_{x,K,n_x} \end{bmatrix}^T \tag{5.14}$$

$$\boldsymbol{\mu}_u = \begin{bmatrix} \mu_{u,1,1} & \mu_{u,1,2} & \mu_{u,1,3} & \mu_{u,2,1} & \mu_{u,2,2} & \dots & \mu_{u,K,n_u} \end{bmatrix}^T \tag{5.15}$$

$$\boldsymbol{\lambda}_x = \begin{bmatrix} \lambda_{x,1} & \lambda_{x,2} & \dots & \lambda_{x,K} \end{bmatrix}^T \tag{5.16}$$

$$\boldsymbol{\lambda}_u = \begin{bmatrix} \lambda_{u,1} & \lambda_{u,2} & \dots & \lambda_{u,K} \end{bmatrix}^T \tag{5.17}$$

where $\boldsymbol{x}$, $\boldsymbol{u}$ and η represent the state, control and time dilation factor, $\boldsymbol{\nu}^+$ and $\boldsymbol{\nu}^-$ are required to implement the virtual control L_1-norm in the objective function, and $\boldsymbol{\mu}_x$, $\boldsymbol{\mu}_u$, $\boldsymbol{\lambda}_x$, and $\boldsymbol{\lambda}_u$ are four slack variables required for the implementation of the trust region of Section 4.7. To enable the reader to obtain a better understanding of the structure of this vector, the sizes of its respective elements can be provided as

$$\boldsymbol{x}_e = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \\ \eta \\ \boldsymbol{\nu}^+ \\ \boldsymbol{\nu}^- \\ \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_u \\ \boldsymbol{\lambda}_x \\ \boldsymbol{\lambda}_u \end{bmatrix} \in \begin{bmatrix} \mathbb{R}^{Kn_x \times 1} \\ \mathbb{R}^{Kn_u \times 1} \\ \mathbb{R}^{1} \\ \mathbb{R}^{Kn_x \times 1} \\ \mathbb{R}^{Kn_x \times 1} \\ \mathbb{R}^{Kn_x \times 1} \\ \mathbb{R}^{Kn_u \times 1} \\ \mathbb{R}^{K \times 1} \\ \mathbb{R}^{K \times 1} \end{bmatrix} \tag{5.18}$$

It is interesting to observe that the virtual-control and trust-region techniques add a significant number of optimisation parameters to the spacecraft reorientation problem.

5.4.3. Boundary conditions

The boundary conditions of the spacecraft reorientation problem can be formulated in a relatively straightforward manner, where the major challenge is to select the optimisation variables at the boundary points of the grid through the A matrix of Equation 5.7. The subsets $A_{\text{bc},0}$ and $A_{\text{bc},f}$ of A, and $\boldsymbol{b}_{\text{bc},0}$ and $\boldsymbol{b}_{\text{bc},f}$ of $\boldsymbol{b}$ were formulated as

$$A_{\text{bc},0} = \Big[\overbrace{\begin{bmatrix} I_{n_x} & 0_{n_x \times (K-1)n_x} \end{bmatrix}}^{\boldsymbol{x}} \quad \overbrace{0_{n_x \times Kn_u}}^{\boldsymbol{u}} \quad \overbrace{0}^{\eta} \quad \overbrace{0_{n_x \times Kn_x}}^{\boldsymbol{\nu}^+} \quad \overbrace{0_{n_x \times Kn_x}}^{\boldsymbol{\nu}^-} \quad \overbrace{0_{n_x \times K(n_x+n_u+2)}}^{\boldsymbol{\mu}_x, \boldsymbol{\mu}_u, \boldsymbol{\lambda}_x, \boldsymbol{\lambda}_u} \Big] \tag{5.19}$$

$$\boldsymbol{b}_{\text{bc},0} = \begin{bmatrix} \boldsymbol{q}_0 \\ \boldsymbol{\omega}_0 \end{bmatrix} \tag{5.20}$$

$$A_{\text{bc},f} = \begin{bmatrix} \begin{bmatrix} 0_{n_x \times (K-1)n_x} & I_{n_x} \end{bmatrix} & 0_{n_x \times Kn_u} & 0 & 0_{n_x \times Kn_x} & 0_{n_x \times Kn_x} & 0_{n_x \times K(n_x+n_u+2)} \end{bmatrix} \tag{5.21}$$

$$\boldsymbol{b}_{\text{bc},f} = \begin{bmatrix} \boldsymbol{q}_f(\bar{\eta}) \\ \boldsymbol{\omega}_f(\bar{\eta}) \end{bmatrix} \tag{5.22}$$

In Equation 5.19, the relation between the elements in matrix $A_{\text{bc},0}$ and the vector of optimisation parameters $\boldsymbol{x}_e$ is shown for the sake of clarity. In the remainder of this section, this relation is not shown for all matrices that are presented for the sake of conciseness. Furthermore, where possible, matrices and vectors will be presented in forms that are somewhat more concise.

5.4.4. Dynamics and kinematics

The dynamic and kinematic constraints from Equation 4.72 are incorporated into the ECOS equality constraint matrix A and vector $\boldsymbol{b}$ as

$$A_{\text{dyn}} = \begin{bmatrix} A_{\text{ecos}} & B_{\text{ecos}} & C_{\text{ecos}} & 1_{Kn_x} & 0_{Kn_x} & 0_{Kn_x \times K(n_x+n_u+2)} \end{bmatrix} \tag{5.23}$$

$$\boldsymbol{b}_{\text{dyn}} = \begin{bmatrix} -\boldsymbol{r}_1 \\ -\boldsymbol{r}_2 \\ \vdots \\ -\boldsymbol{r}_K \end{bmatrix} \tag{5.24}$$

where

$$A_{\text{ecos}} = \begin{bmatrix} A_1 & -I_{n_x} & 0 & \cdots & 0 & 0 \\ 0 & A_2 & -I_{n_x} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & A_{K-1} & -I_{n_x} \end{bmatrix} \tag{5.25}$$

$$B_{\text{ecos}} = \begin{bmatrix} B_1^- & B_1^+ & 0 & \cdots & 0 & 0 \\ 0 & B_2^- & B_2^+ & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & B_{K-1}^- & B_{K-1}^+ \end{bmatrix} \tag{5.26}$$

$$C_{\text{ecos}} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_{K-1} \end{bmatrix} \tag{5.27}$$

As can be observed and will be further elaborated upon in Subsection 5.4.7, only the $\boldsymbol{v}^+$ virtual control parameter is included in A_{dyn}. Consequently, $\boldsymbol{v}^+$ represents the virtual control variable as defined in Equation 4.72, and $\boldsymbol{v}^-$ is solely introduced in order to include the L_1-norm in the ECOS-compatible objective function (Subsection 5.4.7).

5.4.5. Linear inequality constraints

Linear inequality constraints can be modelled in an ECOS-compatible manner ($G\boldsymbol{x}_e \preceq_K \boldsymbol{h}$) in a relatively straightforward manner. For illustration purposes, this process is described in this subsection for the angular rate box constraint, which can be formulated as

$$G_{\text{lin},\omega,\text{box}}\boldsymbol{x}_e \preceq_K \boldsymbol{h}_{\text{lin},\omega,\text{box}} \tag{5.28}$$

This equation can be expanded as

$$\begin{bmatrix} G_{\text{lin},\omega,\text{box},1} & 0_{2(3K)\times Kn_u} & 0 & 0_{2(3K)\times Kn_x} & 0_{2(3K)\times Kn_x} & 0_{2(3K)\times K(n_x+n_u+2)} \end{bmatrix} \boldsymbol{x}e \preceq_K \boldsymbol{h}_{\text{lin},\omega,\text{box}} \tag{5.29}$$

where

$$G_{\text{lin},\omega,\text{box},1} = \left[\begin{array}{cccc} [0_{3\times4} & I_3] & 0 & \cdots & 0 \\ 0 & & [0_{3\times4} & I_3] & \cdots & 0 \\ \vdots & & \vdots & & \ddots & \vdots \\ 0 & & 0 & & \cdots & [0_{3\times4} \quad I_3] \\ \hline [0_{3\times4} & -I_3] & 0 & & \cdots & 0 \\ 0 & & [0_{3\times4} & -I_3] & \cdots & 0 \\ \vdots & & \vdots & & \ddots & \vdots \\ 0 & & 0 & & \cdots & [0_{3\times4} \quad -I_3] \end{array}\right], \quad \boldsymbol{h}_{\text{lin},\omega,\text{box}} = \left[\begin{array}{c} \boldsymbol{\omega}_{\text{box}} \\ \boldsymbol{\omega}_{\text{box}} \\ \vdots \\ \boldsymbol{\omega}_{\text{box}} \\ \hline \boldsymbol{\omega}_{\text{box}} \\ \boldsymbol{\omega}_{\text{box}} \\ \vdots \\ \boldsymbol{\omega}_{\text{box}} \end{array}\right] \tag{5.30}$$

$G_{\text{lin},\omega,\text{box},1}$ consists of two parts in order to account for both the upper and lower bound on the angular rate of the spacecraft. In a similar fashion, $G_{\text{lin},u,\text{box}}$ and $\boldsymbol{h}_{\text{lin},u,\text{box}}$ can be obtained for the box constraint on the control. Furthermore, it is possible to impose the hard trust-region constraint δ_η on the time dilation factor η, which was presented in Section 4.7, as

$$G_{\text{lin},\delta_\eta}\boldsymbol{x}_e \preceq_K \boldsymbol{h}_{\text{lin},\delta_\eta} \tag{5.31}$$

$$\left[\begin{array}{ccccc} 0_{1\times Kn_x} & 0_{1\times Kn_u} & 1 & 0_{1\times 2Kn_x} & 0_{1\times K(n_x+n_u+2)} \\ 0_{1\times Kn_x} & 0_{1\times Kn_u} & -1 & 0_{1\times 2Kn_x} & 0_{1\times K(n_x+n_u+2)} \end{array}\right] \boldsymbol{x}_e \preceq_K \left[\begin{array}{c} (1+\delta_\eta)\bar{\eta} \\ -(1-\delta_\eta)\bar{\eta} \end{array}\right] \tag{5.32}$$

5.4.6. SOCP constraints

Several SOCP constraints had to be formulated in order to incorporate various elements of the attitude reorientation problems considered in this study: the ellipsoidal constraint on the control, the conical attitude constraints, the minimum-energy objective, and constraints related to the trust region. This subsection outlines the procedure that was followed to formulate these SOCP constraints. An important step in this process is to reformulate the general second-order cone constraint of Equation 2.12, $\|A\boldsymbol{x}+\boldsymbol{b}\|_2 \leq \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}+d$ into the form $G\boldsymbol{x}_e \preceq_K \boldsymbol{h}$. For this purpose, a reformulation presented in the work of Wenzel and Seelbinder (2017) was used, which can be represented as

$$\left[\begin{array}{c} -A \\ -\boldsymbol{c}^{\mathrm{T}} \end{array}\right] \boldsymbol{x}_e \preceq \left[\begin{array}{c} \boldsymbol{b} \\ d \end{array}\right] \tag{5.33}$$

where A, $\boldsymbol{b}$, $\boldsymbol{c}$, and d represent the same vectors and matrices as in Equation 2.12. The desired formulation, $G\boldsymbol{x}_e \preceq_K \boldsymbol{h}$, could be obtained after defining

$$G = \left[\frac{-A}{-\boldsymbol{c}^{\mathrm{T}}}\right], \quad \text{and} \quad \boldsymbol{h} = \left[\frac{\boldsymbol{b}}{d}\right] \tag{5.34}$$

As an example, the ellipsoidal constraint on the control for a single grid point k can be formulated as

$$G_{\text{2-cone},u,\text{ellipse},k}\boldsymbol{x}_e \preceq_K \boldsymbol{h}_{\text{2-cone},u,\text{ellipse},k} \tag{5.35}$$

which can be further expanded into

$$\left[\frac{0_{1\times Kn_x} \quad G_{\text{2-cone},u,\text{ellipse},k,1} \quad 0_{1\times 1} \quad 0_{1\times Kn_x} \quad 0_{1\times Kn_x} \quad 0_{1\times K(n_x+n_u+2)}}{0_{1\times n_z}}\right] \boldsymbol{x}_e \preceq_K \boldsymbol{h}_{\text{2-cone},u,\text{ellipse},k} \tag{5.36}$$

where

$$G_{\text{2-cone},u,\text{ellipse},k,1} = \left[\begin{array}{ccccccc} 0_{1\times Kn_x} & 0_{1\times n_u(k-1)} & \boldsymbol{u}_{\text{ellipse}}^{-1} I_3 & 0 & \cdots & 0 \end{array}\right] \tag{5.37}$$

$$\boldsymbol{h}_{\text{2-cone},\omega,\text{ellipse},k} = \left[\frac{0}{1}\right] \tag{5.38}$$

In other words, for every grid point k, $G_{\text{2-cone},u,\text{ellipse},k}$ and $\boldsymbol{h}_{\text{2-cone},u,\text{ellipse},k}$ are generated and appended to the matrix G and vector $\boldsymbol{h}$. In order to communicate the size of these constraints to ECOS, their number of rows is saved as an input variable for the solver as

$$\text{dims.q}(k) = \text{size}(G_{\text{2-cone},u,\text{ellipse},k}) \tag{5.39}$$

where 'dims' is an input variable for the ECOS solver. In this way $G_{\text{2-cone},u,\text{ellipse}}$ can be constructed as

$$G_{\text{2-cone},u,\text{ellipse}} = \begin{bmatrix} G_{\text{2-cone},u,\text{ellipse},1} \\ \vdots \\ G_{\text{2-cone},u,\text{ellipse},K} \end{bmatrix} \tag{5.40}$$

and $\boldsymbol{h}_{\text{2-cone},u,\text{ellipse}}$ as

$$\boldsymbol{h}_{\text{2-cone},u,\text{ellipse}} = \begin{bmatrix} \boldsymbol{h}_{\text{2-cone},u,\text{ellipse},1} \\ \vdots \\ \boldsymbol{h}_{\text{2-cone},u,\text{ellipse},K} \end{bmatrix} \tag{5.41}$$

In a similar manner, the conical attitude keep-out and keep-in constraints can be represented in an ECOS-compatible format. All of these constraints are then collected in the $G_{\text{2-cone}}$ matrix and $\boldsymbol{h}_{\text{2-cone}}$ vector as

$$G_{\text{2-cone}} = \begin{bmatrix} G_{\text{2-cone,u,ellipse}} \\ G_{\text{2-cone,keepout}} \\ G_{\text{2-cone,keepin}} \end{bmatrix} \tag{5.42}$$

and

$$\boldsymbol{h}_{\text{2-cone}} = \begin{bmatrix} \boldsymbol{h}_{\text{2-cone,u,ellipse}} \\ \boldsymbol{h}_{\text{2-cone,keepout}} \\ \boldsymbol{h}_{\text{2-cone,keepin}} \end{bmatrix} \tag{5.43}$$

5.4.7. Virtual control

As was discussed in Subsection 5.4.2, the virtual control term is represented using two different variables, $\boldsymbol{\nu}^+$ and $\boldsymbol{\nu}^-$ in the ECOS problem formulation. This is required to impose the L_1-norm in the problem objective in a manner that meets the ECOS format of Equation 5.7. Several steps were taken to implement this L_1-norm. The first step was to only include the $\boldsymbol{\nu}^+$ variable in the dynamic and kinematic equality constraints from Equation 5.23. Consequently, this variable $\boldsymbol{\nu}^+$ represents the *actual* sign of the virtual control from Equation 4.72, which can either be positive or negative. However, to impose the L_1-norm in the linear objective function, the negative values have to be reformulated into their positive counterparts. For this purpose, the $\boldsymbol{\nu}^-$ parameter was introduced, following a strategy proposed by Reynolds et al. (2020a). Then, through the introduction of $2Kn_x$ inequality constraints, the $\boldsymbol{\nu}^-$ parameter was defined as a positive outer bound on the virtual control. This scheme can be represented as

$$-\boldsymbol{\nu}^- \leq \boldsymbol{\nu}^+ \leq \boldsymbol{\nu}^- \tag{5.44}$$

In other words, while $\boldsymbol{\nu}^+$ can have either a positive or negative sign, $\boldsymbol{\nu}^-$ is always strictly positive. In practice, the corresponding (linear) ECOS constraints can be formulated as

$$G_{\text{lin},\nu}\boldsymbol{x}_e \preceq_K \boldsymbol{h}_{\text{lin},\nu} \tag{5.45}$$

where

$$G_{\text{lin},\nu} = \begin{bmatrix} 0_{Kn_x\times Kn_x} & 0_{Kn_x\times Kn_u} & 0 & I_{Kn_x} & -I_{Kn_x} & 0_{Kn_x\times K(n_x+n_u+2)} \\ 0_{Kn_x\times Kn_x} & 0_{Kn_x\times Kn_u} & 0 & -I_{Kn_x} & -I_{Kn_x} & 0_{Kn_x\times K(n_x+n_u+2)} \end{bmatrix} \tag{5.46}$$

$$\boldsymbol{h}_{\text{lin},\nu} = \begin{bmatrix} 0_{Kn_x\times 1} \\ 0_{Kn_x\times 1} \end{bmatrix} \tag{5.47}$$

5.4.8. Trust region

To implement the L_2-norm type trust region as presented in Section 4.7, additional slack variables, equality constraints and inequality constraints were introduced to the convex optimisation problem. First, in order to obtain the state and control difference between the optimisation parameters $\boldsymbol{x}$ and $\boldsymbol{u}$ and their respective reference trajectories $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{u}}$, the parameters $\boldsymbol{\mu}_x \in \mathbb{R}^{Kn_x}$ and $\boldsymbol{\mu}_u \in \mathbb{R}^{Kn_u}$ were introduced and defined using a set of equality constraints. These parameters are defined as

$$\boldsymbol{\mu}_x = \boldsymbol{x} - \bar{\boldsymbol{x}}, \quad \boldsymbol{\mu}_u = \boldsymbol{u} - \bar{\boldsymbol{u}} \tag{5.48}$$

Or, in an equality-constraint formulation that is ECOS-compatible, as

$$A_{\text{tr},x} = \begin{bmatrix} \overbrace{-I_{Kn_x}}^{\boldsymbol{x}} & \overbrace{0_{Kn_x\times(Kn_u+1+Kn_x+Kn_x)}}^{\boldsymbol{u},\eta,\boldsymbol{v}^+,\boldsymbol{v}^-} & \overbrace{I_{Kn_x}}^{\boldsymbol{\mu}_x} & \overbrace{0_{Kn_x\times(Kn_u+K+K)}}^{\boldsymbol{\mu}_u,\boldsymbol{\lambda}_x,\boldsymbol{\lambda}_u} \end{bmatrix} \tag{5.49}$$

$$\boldsymbol{b}_{\text{tr},x} = \begin{bmatrix} -\bar{\boldsymbol{x}} \end{bmatrix} \tag{5.50}$$

In a similar fashion, $A_{\text{tr},u}$ and $\boldsymbol{b}_{\text{tr},u}$ could be obtained. Afterwards, the second set of trust region-related slack variables, $\boldsymbol{\lambda}_x \in \mathbb{R}^K$ and $\boldsymbol{\lambda}_u \in \mathbb{R}^K$, were used to obtain the L_2-norm at every single grid point for both the state and control deviations through the implementation of a number of SOCP constraints. To illustrate, the corresponding inequality constraint for the state deviation at a single grid node k can be formulated as

$$\|\boldsymbol{\mu}_{x,k}\|_2 \preceq \lambda_{x,k} \tag{5.51}$$

Or, in the conical form that is ECOS-compatible, as

$$G_{2\text{-cone,tr},x,k} = \left[\begin{array}{ccccc|cc|cc} \cdots & \overbrace{0_{1\times Kn_x}}^{\boldsymbol{\mu}_x} & & & & \overbrace{0_{1\times Kn_u}}^{\boldsymbol{\mu}_u} & \overbrace{[0_{1\times(k-1)} \quad 1 \quad 0_{1\times(K-k)}]}^{\boldsymbol{\lambda}_x} & \overbrace{0_{1\times K}}^{\boldsymbol{\lambda}_u} \\ \cdots & [0_{n_x\times n_x(k-1)} & I_{n_x} & 0_{n_x\times n_x(K-k)}] & & 0_{n_x\times Kn_u} & 0_{n_x\times K} & 0_{n_x\times K} \end{array}\right] \tag{5.52}$$

$$\boldsymbol{h}_{2\text{-cone,tr},x,k} = \left[\begin{array}{c} 0 \\ \hline 0_{n_x\times 1} \end{array}\right] \tag{5.53}$$

Such a conical constraint was formulated for every element of λ_x and λ_u, and these were appended to the G matrix and $\boldsymbol{h}$ vector of Equation 5.7. As a final step, $\boldsymbol{\lambda}_x$ and $\boldsymbol{\lambda}_u$ were directly penalised in the objective function to enforce the trust region, as shown in Subsection 5.4.9.

In hindsight, it is thought that an alternative approach could have been taken to implement this type of trust region, which does not require the introduction of the slack variables $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_u$. Instead, the reference variables $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{u}}$ could be directly included in the SOCP constraints (Equation 5.52 and Equation 5.53) through the $\boldsymbol{b}$-term from Equation 5.33. Potentially, this alternative approach could improve the computational efficiency of the SCP algorithm due to the reduced size of the corresponding optimisation problem. Comparing the performance of both approaches is left as a recommendation for further research.

5.4.9. Objective function

The minimum-time objective function from Subsection 4.10.2 can now be formulated in its required linear form as

$$\boldsymbol{c}_0^T = \begin{bmatrix} \overbrace{0_{1\times Kn_x}}^{\boldsymbol{x}} & \overbrace{0_{1\times Kn_u}}^{\boldsymbol{u}} & \overbrace{w_\eta}^{\eta} & \overbrace{0_{1\times Kn_x}}^{\boldsymbol{v}^+} & \overbrace{w_{\text{vc}}\cdot 1_{1\times Kn_x}}^{\boldsymbol{v}^-} & \overbrace{0_{1\times K(n_x+n_u)}}^{\boldsymbol{\mu}_x,\boldsymbol{\mu}_u} & \overbrace{w_{\text{tr}}\cdot 1_{1\times 2K}}^{\boldsymbol{\lambda}_x,\boldsymbol{\lambda}_u} \end{bmatrix} \tag{5.54}$$

It becomes clear that through using this approach the virtual control is penalised through an L_1-norm.

The main difference between the ECOS implementations of the minimum-time and minimum-energy reorientation problems can be traced back to this objective function. For the minimum-energy problem, the time-dilation factor η is replaced in the vector of optimisation parameters with the slack variable s_u. This slack variable s_u represents the energy that is required for the reorientation manoeuvre through the following SOCP constraint

$$G_{2\text{-cone},u}\boldsymbol{x}_e \preceq_K \boldsymbol{h}_{2\text{-cone},u} \tag{5.55}$$

which was formulated in the manner presented in Subsection 5.4.6.

5.4.10. Full formulation

As noted in Subsection 5.4.1, the goal of this subsection was to obtain the vectors $\boldsymbol{c}_0^{\text{T}}$, $\boldsymbol{b}$ and $\boldsymbol{h}$, and the matrices A and G that are required to solve the spacecraft reorientation problem studied in this research project using the convex solving algorithm ECOS. Using the expressions from the previous subsections, these parameters can now be obtained (for the minimum-time problem) as

$$A = \begin{bmatrix} A_{\text{bc},0} \\ A_{\text{dyn}} \\ A_{\text{bc},f} \\ A_{\text{tr},x} \\ A_{\text{tr},u} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_{\text{bc},0} \\ \boldsymbol{b}_{\text{dyn}} \\ \boldsymbol{b}_{\text{bc},f} \\ \boldsymbol{b}_{\text{tr},x} \\ \boldsymbol{b}_{\text{tr},u} \end{bmatrix} \tag{5.56}$$

and

$$G = \begin{bmatrix} G_{\text{lin},\omega,\text{box}} \\ G_{\text{lin},u,\text{box}} \\ G_{\text{lin},\delta_\eta} \\ G_{\text{lin},\nu} \\ G_{\text{2-cone}} \\ G_{\text{2-cone,tr},x} \\ G_{\text{2-cone,tr},u} \end{bmatrix}, \quad \boldsymbol{h} = \begin{bmatrix} \boldsymbol{h}_{\text{lin},\omega,\text{box}} \\ \boldsymbol{h}_{\text{lin},u,\text{box}} \\ \boldsymbol{h}_{\text{lin},\delta_\eta} \\ \boldsymbol{h}_{\text{lin},\nu} \\ \boldsymbol{h}_{\text{2-cone}} \\ \boldsymbol{h}_{\text{2-cone,tr},x} \\ \boldsymbol{h}_{\text{2-cone,tr},u} \end{bmatrix} \tag{5.57}$$

The resulting A and G matrices are shown in, respectively, Figures 5.9 and 5.10. It can be observed that both matrices are highly sparse, which is the main reason that the SCP algorithm combined with the FOH discretisation method excels in terms of computational efficiency.

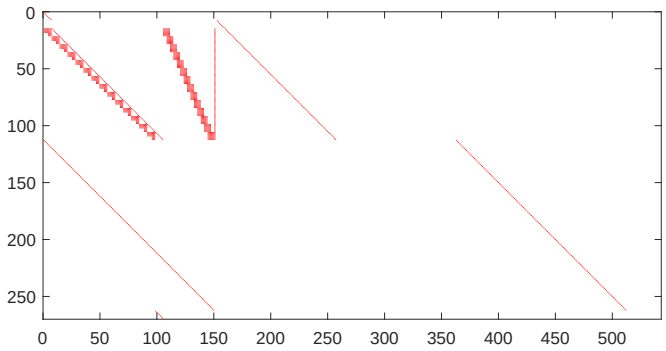


Figure 5.9: Resulting A matrix for the ECOS solving algorithm for the minimum-time, attitude-constrained reorientation problem (1.19% non-zero elements). The problem has 542 optimisation parameters.

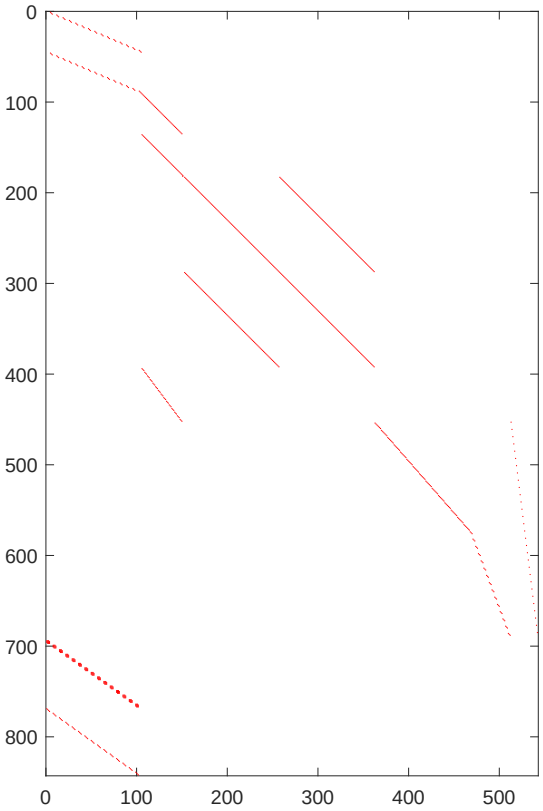


Figure 5.10: Resulting G matrix for the ECOS solving algorithm for the minimum-time, attitude-constrained reorientation problem (0.24% non-zero elements). The problem has 542 optimisation parameters.

This page was intentionally left blank.

6

Monte Carlo Test Campaign Set-up

As stated before, the goal of this study is to develop an SCP-based optimisation algorithm that can reliably solve the optimal spacecraft reorientation problem in real time. To this end, it was determined that a well-structured Monte Carlo analysis would be essential to:

1. Analyse the influence of different design choices and algorithm parameters on the performance of the SCP algorithm (Chapter 8).
2. Develop an in-depth understanding of the performance of the SCP algorithm in terms of the requirements that were identified in Section 3.7 (Chapter 9).

In this chapter, the approach towards the generation of the Monte Carlo test cases is briefly outlined in Section 6.1, and, in Section 6.2, it is discussed how the Monte Carlo results were used to evaluate the performance of the SCP algorithm. Two examples of these Monte Carlo tests cases are provided, verified, and studied in Chapter 7.

6.1. Problem generation

For each test case within the Monte Carlo analyses, it was desired to formulate a unique, complex spacecraft reorientation problem. In this section, it is briefly discussed how these cases were generated. The Monte Carlo test cases correspond to the minimum-energy and minimum-time optimisation problems that were introduced in Section 3.6. Having an understanding of the structure of these Monte Carlo cases can assist in the interpretation of the results provided in later chapters. The elements included in the Monte Carlo test cases are:

- **Spacecraft parameters**
  For the satellite reorientation problem studied in this project, the only relevant satellite parameter is its inertia matrix. Together with research partner OHB, a (constant) reference value was selected for this parameter.
- **Boundary conditions**
  For every Monte Carlo test case, random initial and final states were generated. In this process, it was ensured that the angular rate conditions satisfied the corresponding box constraint. The resulting manoeuvres represent attitude rotations of up to 180 degrees. For the time-dependent boundary conditions, four different final boundaries were generated that represent the boundary conditions at $t = 0$ s, $t = 40$ s, $t = 80$ s, and $t = 120$ s. The boundary conditions for other values of the manoeuvre duration t_f could be obtained through interpolating these four points with a third-order polynomial.
- **Manoeuvre duration**
  For the minimum-energy problem, a final time t_f was specified.
- **Constraints**
  For both the box and angular rate constraints, constant values were used. Furthermore, an attitude keep-out constraint was formulated for the (body-fixed) x-axis of the considered spacecraft. To make

the Monte Carlo test cases as complex as possible, this attitude keep-out constraint was always located at the midpoint of the great-circle segment that connected the x-axis of the spacecraft at the start and end of its manoeuvre on the unit sphere. Finally, an attitude keep-in constraint was formulated for the (body-fixed) z-axis of the spacecraft. The magnitude of the separation angles corresponding to these attitude constraints varied among the Monte Carlo cases.

As stated earlier, an example of a minimum-energy and a minimum-time Monte Carlo test case are provided in Chapter 7. It is noted that the problems formulated in Section 3.6 include both attitude keep-out and keep-in constraints. In this study, this problem formulation is referred to as the *attitude-constrained* reorientation problem. The two other problem types studied in the remainder of this report are listed in Table 6.1.

Table 6.1: The three different problem types studied in this work.

Problem type	**Attitude constraints**
Attitude-unconstrained reorientation	No attitude constraints
Attitude keep-out constrained reorientation	Keep-out attitude constraint
Attitude-constrained reorientation	Both keep-out and keep-in attitude constraints

Apart from the constraints on the attitude, the Monte Carlo test cases corresponding to these different problem types are identical.

6.2. Performance evaluation

In this section, it is briefly discussed how the algorithm requirements and criteria introduced in Section 3.7 were measured and visualised for the Monte Carlo analyses.

6.2.1. Performance metrics

The five performance requirements for an onboard optimisation algorithm for the spacecraft reorientation problem that were identified in Section 3.7 are evaluated in the Monte Carlo analyses through the following parameters:

- **Robustness**
  The robustness of the SCP algorithm is evaluated directly through the *convergence rate* that was obtained from the Monte Carlo analyses. The convergence rate is defined as the rate of cases for which the SCP algorithm was able to find a solution that meets all convergence criteria from Section 4.9.

- **Optimality**
  There exists, unfortunately, no objective standard of the 'true' optimal guidance solution for a certain Monte Carlo test case. Therefore, in order to be able to draw conclusions about the performance of the SCP algorithm regarding optimality, a reference solution was required. To provide this reference solution, all Monte Carlo test cases were also solved with the NLP benchmark algorithm from Section 5.2. The reputation of NLP optimisation methods as the standard for finding optimal solutions (Boyarko et al., 2011; Ventura et al., 2015) was considered to be sufficient for answering the research questions of this study. In this manner, the optimality of the SCP algorithm is evaluated as

$$\text{optimality} = \frac{J_{\text{SCP}} - J_{\text{NLP}}}{J_{\text{NLP}}} \cdot 100\% \tag{6.1}$$

- **Computational efficiency**
  The computational efficiency of the SCP algorithm is evaluated through the *computation times* that were required to solve the Monte Carlo test cases. However, this is not as straightforward as it seems, as simply measuring the time that is required in MATLAB to solve the entire SCP routine is no accurate representation of the efficiency of the SCP algorithm regarding onboard applications. The reason for this is that MATLAB is known to be significantly slower than the compiled language (e.g., C) that would be used for an onboard implementation. Therefore, it was decided that a combination of the two most time-consuming steps in the algorithm would result in a more accurate evaluation. These two steps consist of (1) the runtimes that are reported by the ECOS convex solving algorithm for every convex sub-problem that is solved and (2) the time required for performing the discretisation of the problem

dynamics and kinematics. Both steps are performed using compiled code and, therefore, do not suffer from the low computational efficiency of MATLAB. This approach is in accordance with most studies on sequential convex programming (Reynolds et al., 2020a; Sagliano, 2019; Szmuk et al., 2020).

- **Accuracy**
  The accuracy of the SCP guidance solutions is evaluated by propagating the control solutions of the SCP algorithm with the ODE45 function in MATLAB. The settings that were used are: 'RelTol' = 1×10^{-10} and 'AbsTol' = 1×10^{-10}. The final quaternion that results from this propagation is then compared to the quaternion specified by the boundary condition. The resulting difference is expressed as the rotation angle between both quaternions. In this manner, the accuracy of the SCP algorithm is assessed through the error in the solution of an initial value problem. It should be noted that, in reality, a controller tracks the guidance results ($\boldsymbol{q}$ and $\boldsymbol{\omega}$). Consequently, the fact that these propagation errors exist does not mean that, in practice, the spacecraft would end up off-pointing with respect to a specified target. However, these propagation errors are a good indication of the accuracy of the obtained guidance results.

- **Constraint violations**
  Using the results that are obtained from the propagation of the spacecraft dynamics, the inter-nodal constraint violations can be calculated for the angular rate and attitude constraints. The rate violations are expressed in percentages, while the attitude violations are expressed in degrees, for the sake of intuitiveness.

6.2.2. Presentation of results

Several options were considered to present the results of the Monte Carlo analyses, amongst others, tables and histograms. However, after some experimentation, it was found that box plots provided the most intuitive and fast way to distinguish trends and make observations. In addition, box plots provide insight into both the median or typical performance of the algorithm and the characteristics of certain outliers. These outliers are very important for onboard applications, as unpredictable performance can lead to problems regarding the mission that a spacecraft is part of. In Figure 6.1, the legend is provided containing all box plot elements that are presented in the remainder of this study.

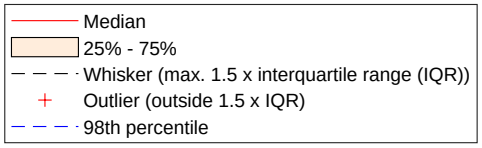


Figure 6.1: Legend for the box plots representing the performance of the SCP algorithm in the Monte Carlo analyses.

In these box plots, the *interquartile range* (IQR) represents the difference between the 25th and 75th percentile (boxes in the figures). All data points outside of a range of 1.5 × IQR from the 25th and 75th percentile are considered to be outliers. For some analyses, not all elements of Figure 6.1 are shown. For instance, the outliers are sometimes omitted to focus on the mean performance.

This page was intentionally left blank.

III

Algorithm Improvement and Results

7

Single Case Analyses

In this chapter, the performance of the SCP algorithm is analysed on a single-case level in order to obtain an initial understanding of the characteristics and performance of the SCP guidance algorithm that was introduced in Chapters 4 and 5. Furthermore, the resulting guidance trajectories are verified using the results of the NLP benchmark algorithm and through an analysis of the problem constraints. The attitude-constrained minimum-energy and minimum-time problems are analysed in, respectively, Sections 7.1 and 7.2.

7.1. Case 1: Minimum-energy spacecraft reorientation

The first test case that was analysed is the so-called minimum-energy, attitude-constrained spacecraft reorientation problem that was presented in Subsection 4.10.1. As noted in Subsection 3.5.2, the minimum-energy problem has a smaller degree of freedom than the minimum-time problem, which makes it relatively easier to solve using numerical methods. Therefore, this problem was identified as a natural starting point for the single-case analyses of this chapter.

7.1.1. Input parameters

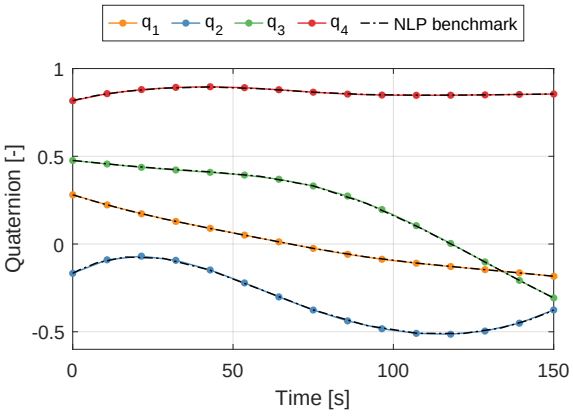
In Table 7.1, all problem and algorithm parameters are provided that were used for this specific test case. The problem parameters (boundary conditions, constraints, etc.) are obtained from one of the Monte Carlo test cases presented in Chapter 6. It is stressed that the algorithm settings used for this test case were already extensively tuned in order to provide good performance. In Chapter 8, an in-depth analysis will be provided of this tuning process and the sensitivities of a number of the most important algorithm parameters and other design elements. As a final note, for this test case, the trapezoidal rule was used for the discretisation of the minimum-energy objective, and the shape-based first guess was used to initialise the SCP algorithm.

Table 7.1: Problem and algorithm parameters for the minimum-energy, attitude-constrained reorientation problem studied in Section 7.1.

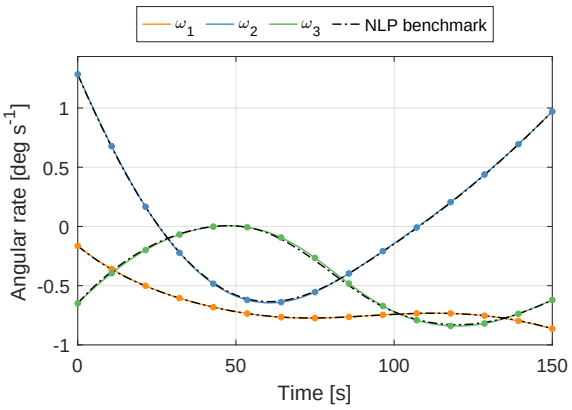
Parameter	Value	Unit	Parameter	Value	Unit
Problem parameters					
t_f	150	s	$\boldsymbol{I}$	$\begin{bmatrix} 1500 & -50 & 75 \\ -50 & 1000 & 25 \\ 75 & 25 & 600 \end{bmatrix}$	$\mathrm{kg\,m^2}$
$\boldsymbol{u}_{\text{ellipse}}$	$[10\ 10\ 5]^T$	Nm			
$\boldsymbol{\omega}_{\text{box}}$	$[2\ 3\ 1.5]^T$	$\mathrm{deg\,s^{-1}}$			
$\boldsymbol{q}_0$	$\begin{bmatrix} 0.2805 \\ -0.1669 \\ 0.4762 \\ 0.8165 \end{bmatrix}$	-	$\boldsymbol{q}_f$	$\begin{bmatrix} -0.1837 \\ -0.3758 \\ -0.3072 \\ 0.8548 \end{bmatrix}$	-
$\boldsymbol{\omega}_0$	$\begin{bmatrix} -0.1638 \\ 1.2844 \\ -0.6488 \end{bmatrix}$	$\mathrm{deg\,s^{-1}}$	$\boldsymbol{\omega}_f$	$\begin{bmatrix} -0.8633 \\ 0.9708 \\ -0.6213 \end{bmatrix}$	$\mathrm{deg\,s^{-1}}$
$\boldsymbol{x}_{B,\text{out}}$	$[1\ 0\ 0]^T$	-	$\boldsymbol{x}_{B,\text{in}}$	$[0\ 0\ 1]^T$	-
$\boldsymbol{y}_{I,\text{out}}$	$\begin{bmatrix} 0.8714 \\ -0.1778 \\ -0.4573 \end{bmatrix}$	-	$\boldsymbol{y}_{I,\text{in}}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	-
$\theta_{\text{sep,out}}$	33.11	deg	$\theta_{\text{sep,in}}$	66.55	deg
Algorithm parameters					
K	15	-	K_{disc}	5	-
w_u	1×10^{2}	-	$n_{\text{it,lim}}$	25	-
w_{tr}	1×10^{-3}	-	w_{vc}	1×10^{2}	-
$\alpha_{\text{tr,adap}}$	5	-	$\alpha_{\text{vc,adap}}$	3	-
$n_{\text{tr,adap,thres}}$	8	-	ϵ_{vc}	5×10^{-5}	-
ϵ_{x}	0.6	-			

7.1.2. Guidance results

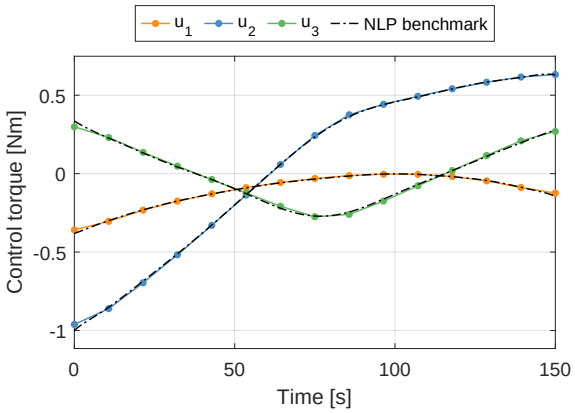
In Figure 7.1, the quaternion, angular rate and control profiles generated by the SCP and NLP algorithms are shown. The coloured points represent the values of the optimisation parameters at the 15 nodes of the SCP grid, which are directly obtained from the converged SCP solution. The coloured lines represent the interpolated time histories of the state and control parameters. The interpolated quaternion and control profiles were obtained through linear interpolation, whereas the angular rate solution of the SCP algorithm was interpolated with a third-order polynomial. This interpolation was performed purely for visualisation purposes. As a result, the continuous quaternion and angular rate time histories shown in Figure 7.1 do not represent the *actual* manoeuvre trajectory. This actual trajectory is only obtained after propagating the obtained control profile, which was performed at a later stage in the analysis and is presented in Subsection 7.1.5. Finally, the dash-dotted black lines represent the guidance solution of the NLP benchmark algorithm.



(a) SCP and NLP quaternion profiles.



(b) SCP and NLP angular rate profiles.



(c) SCP and NLP control profiles.

Figure 7.1: Quaternion, angular rate and control profiles obtained using the SCP and NLP benchmark algorithms.

Several interesting observations can be based on Figure 7.1:

- It can be concluded that the SCP results closely match the results of the NLP benchmark for this specific test case. Therefore, the conclusion can be drawn that the SCP algorithm successfully found a locally optimal solution to the minimum-energy, attitude-constrained spacecraft reorientation problem. As stated before, it cannot be guaranteed that the globally optimal solution to the convex sub-problem is equivalent to the globally optimal solution to the original, non-convex problem.
- The one main difference that can be observed between the SCP and NLP solutions is a slightly different control profile, which becomes visually distinguishable near the initial boundary of the reorientation manoeuvre. However, the effects of this difference appear to be small, as it does not lead to visual differences in the quaternion and angular rate profiles. Nevertheless, this is an interesting observation. It was found that this difference is related to the discretisation rule that is used as part of the SCP algorithm to discretise the minimum-energy integral in the objective function. An in-depth discussion of this phenomenon and the choice of discretisation technique is presented in Section 8.2.
- Inspecting the structure of the optimal control profiles of the SCP and NLP solutions, it can be seen that the first-order-hold assumption (linear interpolation) of the control accurately represents the continuous shape that can be observed in the Lagrange polynomial-modelled control profiles of the NLP benchmark. This was the case for all minimum-energy reorientation cases that were individually analysed during this study and shows that the FOH assumption on the control does not represent a disadvantage for the SCP method regarding the minimum-energy reorientation problem.

7.1.3. Algorithm performance

In Table 7.2, the most important performance criteria corresponding to this test case are presented in terms of the metrics presented in Chapter 6.

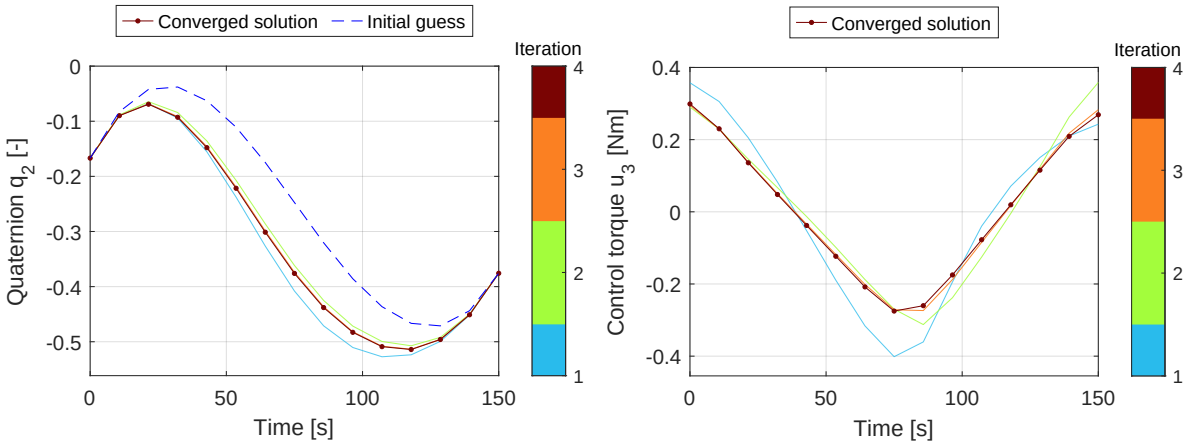
Table 7.2: Performance parameters of the SCP and NLP algorithms for the minimum-energy test case presented in Table 7.1.

Parameter	Value	Unit
SCP computation time	0.0368	sec
NLP computation time	0.189	sec
Relative optimality $((J_{SCP} - J_{NLP})/J_{NLP})$	0.108	%
Propagation error (accuracy)	$1.09 \cdot 10^{-3}$	deg
SCP iterations required for convergence	4	[-]

As expected, based on the fact that the SCP and NLP guidance profiles that were shown in Figure 7.1 are highly similar, the SCP results are highly optimal, under the assumption that the NLP benchmark represents a globally optimal solution. Furthermore, it can be seen that the SCP algorithm is significantly faster than the NLP benchmark. However, in this chapter, no in-depth analysis will be performed of these performance criteria, as it is desired to base performance-related observations and conclusions rather on a large number of Monte Carlo cases than on a single test case. Therefore, detailed performance analyses can be found in Chapters 8 and 9.

7.1.4. Convergence process

In order to better understand the behaviour and characteristics of the SCP algorithm, it was found worthwhile to analyse the iterative convergence process that characterises the SCP framework. In Figure 7.2, this iterative process of finding a solution is visualised for, respectively, the state variable q_2 and the control variable u_3.



(a) Convergence process of the q_2 quaternion parameter.

(b) Convergence process of the u_3 control parameter.

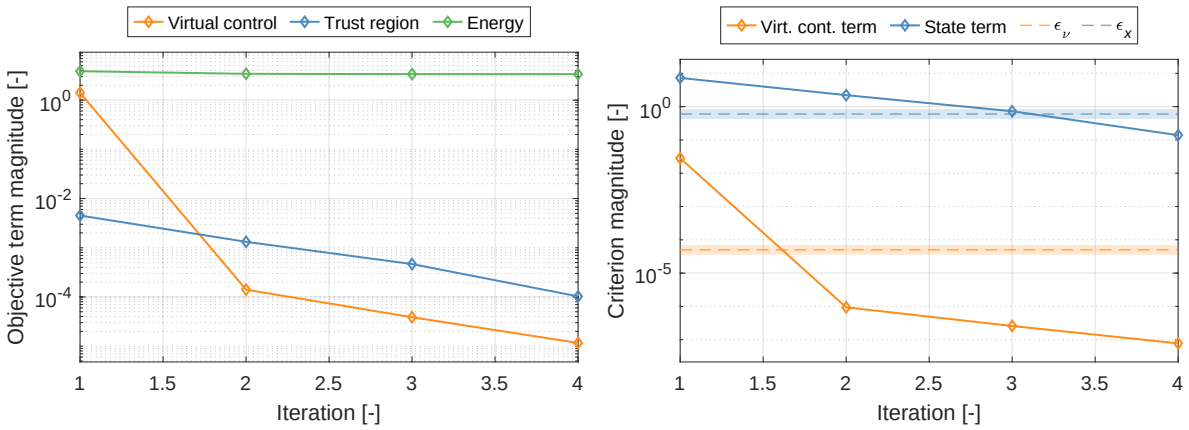
Figure 7.2: Convergence process of the SCP algorithm for the minimum-energy, attitude-constrained reorientation problem that was presented in Table 7.1.

Based on Figure 7.2, several interesting observations can be made:

- The differences between the solutions of subsequent convex sub-problems gradually become smaller as the number of iterations increases and convergence is approached. As stipulated, this process continues until the difference between two subsequent state trajectories is smaller than a predefined limit, which is assessed through the convergence criteria ϵ (Section 4.9). When these criteria are met, in this case after four iterations, the algorithm stops, and the most recent solution is referred to as *accepted*. Within a bound of both linearisation and discretisation error, the solution of this most recent convex sub-problem is expected to represent a locally optimal solution with respect to the original, non-convex problem.
- The control profiles that can be observed in Figure 7.2b are in line with expectations. In initial iterations, when the algorithm 'tries' to find feasible solutions, these are usually sub-optimal due to the large linearisation errors that are present. As a result, the amount of control that is used (and thus the amount of energy that is required) for the manoeuvre can be seen to be relatively high, while, as convergence and thus the local optimum is approached, the required control (at least for the u_3 control parameter) decreases. This behaviour was commonly observed for minimum-energy reorientation test cases.

- The initial guess of the state parameter q_2 is surprisingly accurate, and the algorithm already approaches the local optimum in its second iteration. For the minimum-energy problem, this behaviour was often observed, although, in many occasions, more iterations were required (see Chapter 9). The performance of the shape-based first guess is further analysed in Section 8.1.

In Figure 7.3, the convergence process of the objective function and the convergence criteria is presented.



(a) Convergence history of the objective function.

(b) Convergence history of the convergence criteria.

Figure 7.3: Convergence process of the objective function and convergence criteria of the SCP algorithm for the minimum-energy, attitude-constrained reorientation problem presented in Table 7.1.

The behaviour of these terms is as expected:

- Although it can barely be distinguished on the logarithmic scale of this figure, after further inspection, it was found that the energy-term in the objective function starts with a relatively high magnitude, due to the sub-optimal initial guess and first iteration, and then shows a decreasing trend for subsequent iterations.

- Furthermore, it can be seen that the virtual control variable is used to a significant extent for the initial iteration of the SCP algorithm, as its magnitude is several orders of magnitude higher than its corresponding convergence criterion. Due to the large linearisation and discretisation errors that are present for the first iteration, it appears that the SCP algorithm requires the use of virtual control to find a feasible solution. After the first iteration, it can be seen that the use of virtual control decreases to a level that is acceptable for convergence.

- Finally, the state deviation gradually decreases and meets its convergence criterion after four iterations. This is in line with the results from Figure 7.2, where it could be seen that the difference between the state parameters of the final two iterations, $\Delta \boldsymbol{x}$, becomes very small.

After paying attention to the guidance results, the performance of the SCP algorithm and the convergence process, in Figure 7.4, a 3-D visualisation is provided of the reorientation manoeuvres that were obtained using the SCP and NLP algorithms. Both the keep-out and keep-in constraints are shown, in addition to the body-fixed x- and z-axis of the spacecraft. To be clear, the z-axis (blue line) should be kept *inside* the keep-in constraint (blue cone), and the x-axis (red line) should be kept *outside* of the keep-out constraint (red cone). Furthermore, the convergence history is shown in the form of the body-fixed x-axis trajectories of all SCP iterations. These are barely distinguishable for this specific case but will be more insightful in the remainder of this study.

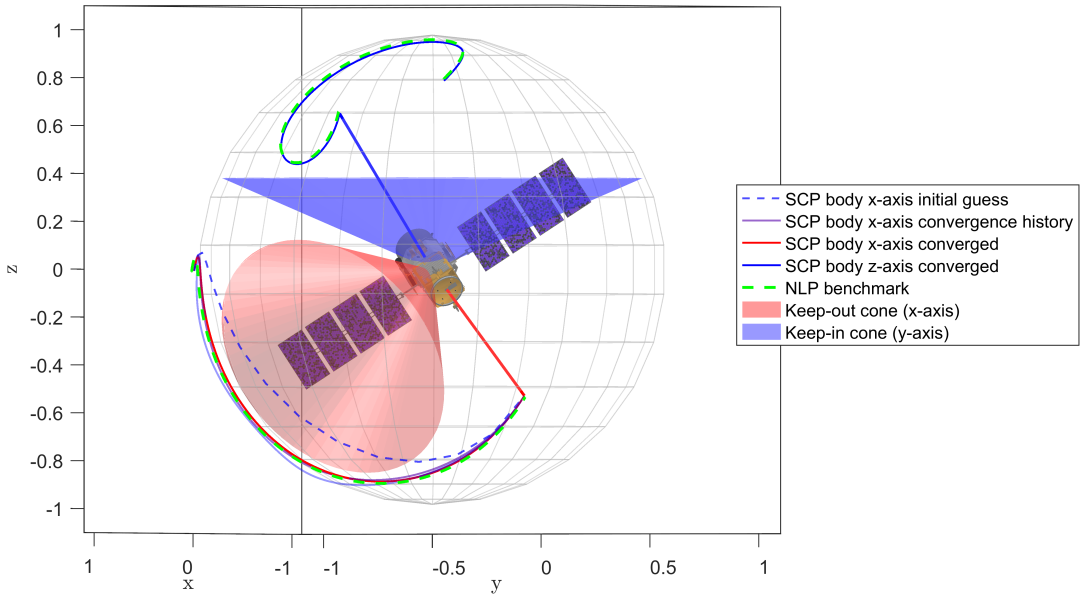


Figure 7.4: 3-D visualisation of the SCP and NLP guidance trajectories for the minimum-energy, attitude-constrained reorientation problem formulated in Table 7.1.

Observing the body-fixed x-axis of the spacecraft, it can be seen that the algorithm converged on a guidance trajectory around the attitude keep-out cone. Furthermore, it can be observed that the body-fixed z-axis of the spacecraft is continuously pointing inside the attitude keep-in cone during the manoeuvre, as required. Therefore, it can be concluded that the SCP algorithm successfully found an attitude constraint-satisfying, optimal trajectory using a relatively poor initial guess that violated the keep-out constraint.

7.1.5. Further verification and analysis

The strategy that was used in this research project to verify the solutions of the SCP algorithm consists of multiple analyses and observations, of which several can be made on the basis of the results provided in the previous subsections:

1. In Figure 7.1, it can be seen that the SCP algorithm converges on the exact same guidance trajectory as the verified (see Section 5.3) NLP benchmark solution. This provides strong evidence for the verification of the results of the SCP algorithm. As discussed before, pseudospectral NLP methods are widely recognised in literature and industry to be the benchmark for finding optimal solutions to nonlinear problems.

2. As becomes evident from Figure 7.1c, the constraints on the control are satisfied by both the SCP and NLP solutions.

3. Another indicator of the correctness of the results is provided by the 3-D visualisation in Figure 7.4, where it can be observed that both the keep-out and keep-in conical constraints are respected, as required.

In addition, in this section, the obtained results are analysed from two other perspectives to solidify the verification of the SCP algorithm. To begin with, in Figure 7.5, a 3-D visualisation is provided of the ellipsoidal constraint on the control and the solution that was obtained from the SCP algorithm.

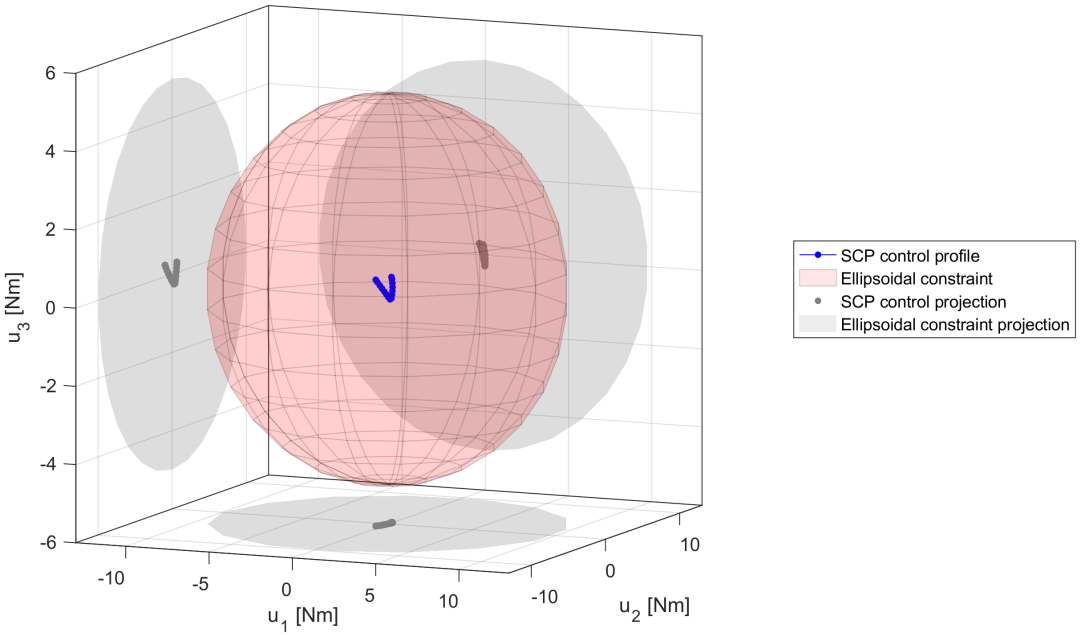
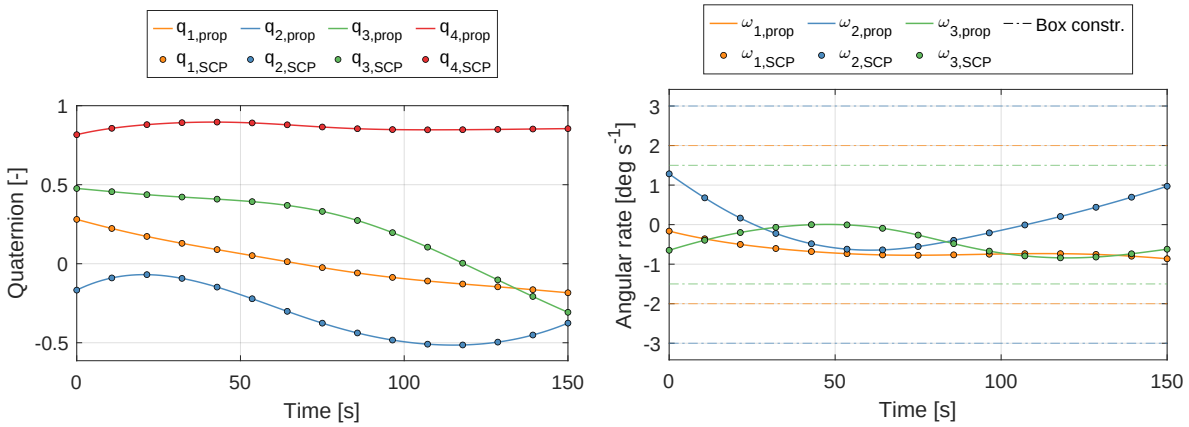


Figure 7.5: 3-D visualisation of the control profile obtained from the SCP algorithm and the ellipsoidal constraint on the control, for the minimum-energy, attitude constrained reorientation problem formulated in Table 7.1.

Clearly, the SCP solution does not violate the specified control constraints. In this case, this is a consequence of the minimum-energy nature of the reorientation, for which control magnitudes near the boundary are avoided as those result in a high cost in terms of energy. It is expected that this situation is different for the minimum-time problem studied in Section 7.2.

The second analysis outlined in this subsection concerns the propagation of the original, nonlinear problem dynamics and kinematics with the control profiles obtained from the SCP algorithm. The goal of this step is threefold: (1) to verify whether the obtained trajectories are feasible and lead to the expected reorientation manoeuvres, (2) to identify and quantify the constraint violations between the grid nodes, and (3) to analyse the performance of the SCP algorithm in terms of the accuracy of its guidance solutions. The first two objectives will be covered in this section, while an in-depth accuracy analysis (latter objective) is provided through the Monte Carlo analyses of Chapter 9. In Figure 7.6, the time histories of the propagated state of the spacecraft $\boldsymbol{x}$ are visually presented along with the SCP solution at the grid nodes. For verification purposes, the box constraints on the angular rate are also indicated in the figure.



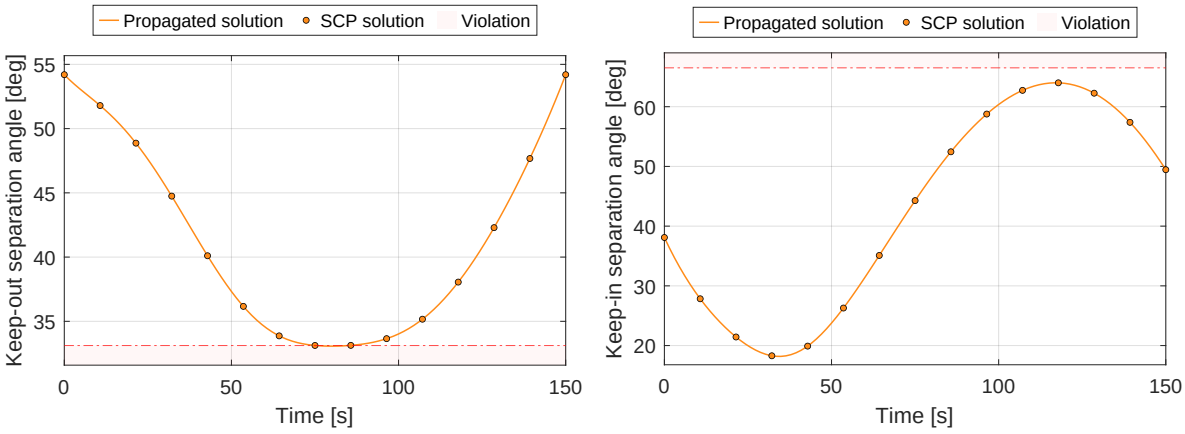
(a) SCP solution and propagated time histories of the quaternion attitude.

(b) SCP solution and propagated time histories of the angular rate.

Figure 7.6: SCP solution of the state parameter $\boldsymbol{x}$ and the corresponding time histories obtained through propagation of the SCP control profiles from Figure 7.1c.

From Figure 7.6, it follows that the SCP algorithm successfully managed to enforce the original, non-convex problem dynamics and kinematics. At least to a visually observable level, the propagated state trajectories perfectly correspond with the SCP solution at the grid nodes. The exact propagation accuracy is further analysed through the Monte Carlo analyses provided in Chapter 9. Furthermore, it can be concluded that the box constraint on the angular rate is respected, which is no surprise for the relatively long (in terms of time) minimum-energy manoeuvre studied in this section.

The time histories of the propagated state parameters of the spacecraft can also be used to analyse the inter-nodal satisfaction of the keep-out and keep-in attitude constraints. In Figure 7.7, the angular separation angles between the body-fixed x-axis and body-fixed z-axes and, respectively, the central axes of the keep-out and keep-in constraints are shown.



(a) Constraint satisfaction of the attitude keep-out constraint.

(b) Constraint satisfaction of the attitude keep-in constraint.

Figure 7.7: Constraint satisfaction of the attitude keep-out and keep-in constraints of the SCP solution for the minimum-energy, attitude-constrained reorientation problem formulated in Table 7.1.

It can be seen that for this specific test case, the guidance trajectory of the SCP algorithm satisfies both the keep-out and keep-in constraints, apart from a minor keep-out constraint violation (0.048 deg) between the grid nodes. These attitude constraint violations will be further analysed through the Monte Carlo analyses presented in Chapter 9.

7.2. Case 2: Minimum-time spacecraft reorientation

The second problem that was analysed on a single-case level is the minimum-time, attitude-constrained spacecraft reorientation problem presented in Subsection 4.10.2. As noted in Subsection 3.5.2, the minimum-time problem is characterised by a larger degree of freedom than the minimum-energy problem. Furthermore, solutions to the minimum-time problem typically approach the constraints on the angular rate and control. As a result, it is known to be more difficult to find an optimal solution for this type of reorientation problem than for the minimum-energy problem discussed in the previous section.

7.2.1. Input parameters

In Table 7.3, the problem and algorithm parameters for this test case are provided.

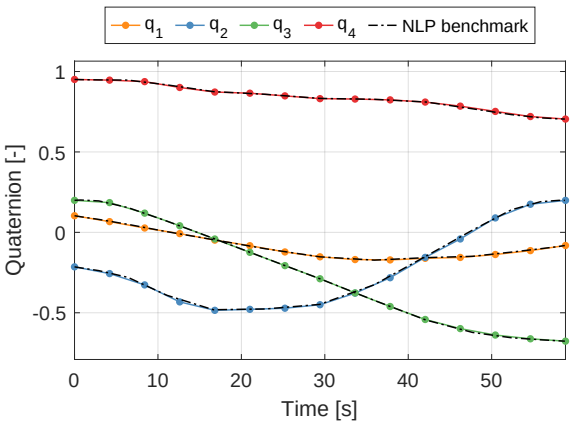
Table 7.3: Problem and algorithm parameters for the minimum-energy, attitude-constrained reorientation problem that is studied in Section 7.2.

Parameter	Value	Unit	Parameter	Value	Unit
Problem parameters					
$t_{f,\text{guess}}$	50	s	$\boldsymbol{I}$	$\begin{bmatrix} 1500 & -50 & 75 \\ -50 & 1000 & 25 \\ 75 & 25 & 600 \end{bmatrix}$	kg m^2
$\boldsymbol{u}_{\text{ellipse}}$	$[10\ 10\ 5]^T$	Nm			
$\boldsymbol{\omega}_{\text{box}}$	$[2\ 3\ 1.5]^T$	deg s^{-1}			
$\boldsymbol{q}_0$	$\begin{bmatrix} 0.1033 \\ -0.2145 \\ 0.1990 \\ 0.9506 \end{bmatrix}$	-	$\boldsymbol{q}_f$	$\begin{bmatrix} -0.080 & -0.082 & -0.082 & -0.082 \\ 0.203 & 0.199 & 0.199 & 0.1970 \\ -0.681 & -0.677 & -0.677 & -0.679 \\ 0.699 & 0.704 & 0.704 & 0.703 \end{bmatrix}$	-
$\boldsymbol{\omega}_0$	$\begin{bmatrix} -0.0175 \\ -0.0125 \\ 0.0144 \end{bmatrix}$	deg s^{-1}	$\boldsymbol{\omega}_f$	$\begin{bmatrix} 0.933 & 0.806 & 0.843 & 0.882 \\ 0.528 & 0.615 & 0.576 & 0.512 \\ -0.073 & -0.167 & -0.084 & -0.076 \end{bmatrix}$	deg s^{-1}
$\boldsymbol{x}_{B,\text{out}}$	$[1\ 0\ 0]^T$	-	$\boldsymbol{x}_{B,\text{in}}$	$[0\ 0\ 1]^T$	-
$\boldsymbol{y}_{I,\text{out}}$	$\begin{bmatrix} 0.8714 \\ -0.1778 \\ -0.4573 \end{bmatrix}$	-	$\boldsymbol{y}_{I,\text{in}}$	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$	-
$\theta_{\text{sep,out}}$	37.73	deg	$\theta_{\text{sep,in}}$	58.31	deg
Algorithm parameters					
K	15	-	K_{disc}	5	-
w_η	1	-	δ_η	30	%
w_{tr}	1×10^{-3}	-	w_{vc}	5×10^{1}	-
$\alpha_{\text{tr,adap}}$	5	-	$\alpha_{\text{vc,adap}}$	3	-
$n_{\text{tr,adap,thres}}$	8	-	ϵ_η	5×10^{-2}	-
ϵ_{x}	0.9	-	ϵ_{vc}	5×10^{-5}	-
$n_{\text{it,lim}}$	25	-			

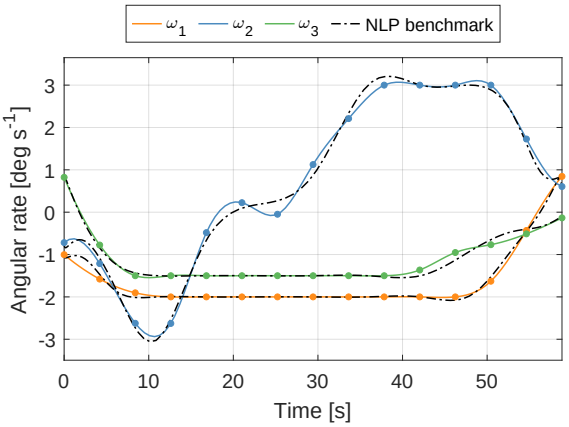
As can be noticed, in constrast to the $\boldsymbol{q}_f$ and $\boldsymbol{\omega}_f$ from Table 7.1, for the minimum-time test case four separate quaternions and angular rates are provided in order to define the time-dependent boundary conditions. In these matrices, the columns represent the final boundary conditions at $t = 0$ s, $t = 40$ s, $t = 80$ s and $t = 120$ s, respectively.

7.2.2. Guidance results

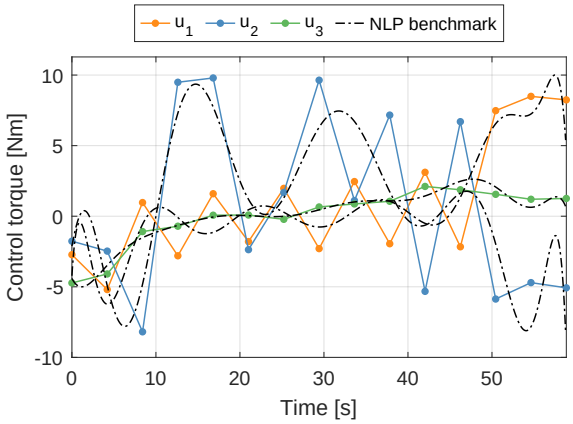
Figure 7.8 shows the quaternion attitude, angular rate and control profiles that are obtained using both the SCP and NLP benchmark algorithms.



(a) SCP and NLP quaternion profiles.



(b) SCP and NLP angular rate profiles.



(c) SCP and NLP control profiles.

Figure 7.8: Quaternion, angular rate and control profiles obtained using the SCP and NLP algorithms for the minimum-time, attitude-constrained reorientation problem defined in Table 7.3.

Based on the SCP and NLP results presented in Figure 7.8, three interesting observations were made:

- The most striking differences between the SCP and NLP solutions can be observed for the control profiles. Although a significant resemblance can be distinguished in terms of structure, obvious differences exist between both profiles. This is a direct consequence of the different methods that both optimisation algorithms use to model the control: while the NLP benchmark uses Lagrange polynomials to model the control, the SCP algorithm uses the first-order-hold modelling presented in Subsection 4.5.2. This modelling difference resulted in few differences between the SCP and NLP solutions for the minimum-energy problem, due to the relatively gradual nature of the optimal control profile. However, the optimal profile for the minimum-time profile seems to contain more abrupt changes, and the different control modelling techniques cause both algorithms to respond differently to this situation. However, interestingly, the differences between the SCP and NLP control profiles seem to have a rather limited influence on the angular rate solutions and an even smaller influence on the quaternion solutions. Therefore, the preliminary conclusion can be drawn that the state profile is relatively insensitive to changes in the control.
- Something that is worth keeping in mind is that the oscillations in the SCP control profiles can be problematic for some mission scenarios, as an attitude control system could struggle to actuate upon such an abruptly changing guidance profile. This phenomenon is further analysed in Section 8.3.
- Another interesting observation can be made in Figure 7.8b, where it can be observed that the magnitudes of angular rate parameters are at (and cross) the boundaries of the corresponding box constraint for the majority of the manoeuvre duration. This could be expected for a minimum-time manoeuvre and will be further analysed in Subsection 7.2.5 through the propagation of the obtained control profiles.

7.2.3. Algorithm performance

In Table 7.4, the most important performance criteria corresponding to this test case are presented.

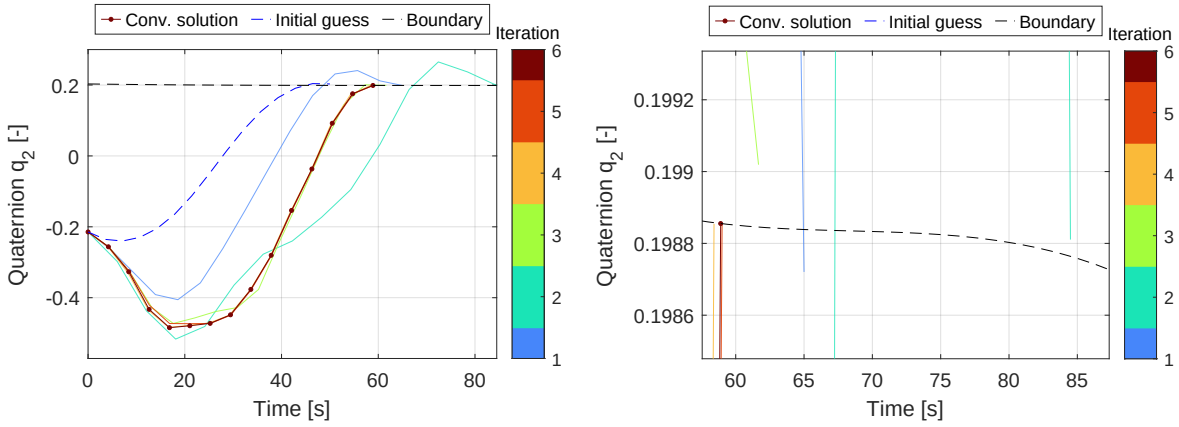
Table 7.4: Performance parameters of the SCP and NLP algorithms for the minimum-time test case from Table 7.3.

Parameter	Value	Unit
SCP computation time	0.081	sec
NLP computation time	0.437	sec
Relative optimality $((J_{SCP} - J_{NLP})/J_{NLP})$	-0.0349	%
Propagation error (accuracy)	$6.39 \cdot 10^{-3}$	deg
SCP iterations required for convergence	6	[-]

The most interesting finding from Table 7.4 is that it supports the observation that the large differences between the SCP and NLP control solutions have a small influence on the performance of the algorithm in terms of optimality. In fact, it can be seen that the guidance solution of the SCP algorithm represents a slightly faster manoeuvre and is, therefore, more optimal. Again, a detailed analysis of the performance of the SCP algorithm is provided in Chapter 9.

7.2.4. Convergence process

In Figure 7.9, the iterative process of finding an optimal solution is visualised for the state parameter q_2. This time, only the convergence history of a state parameter is shown, as the convergence histories of the control parameters did not provide any additional insight in addition to Figure 7.2.



(a) Convergence history of the q_2 attitude parameter.

(b) Convergence process of the q_2 attitude parameter. Detail view of Figure 7.9a.

Figure 7.9: Convergence process of the SCP algorithm for the q_2 attitude parameter.

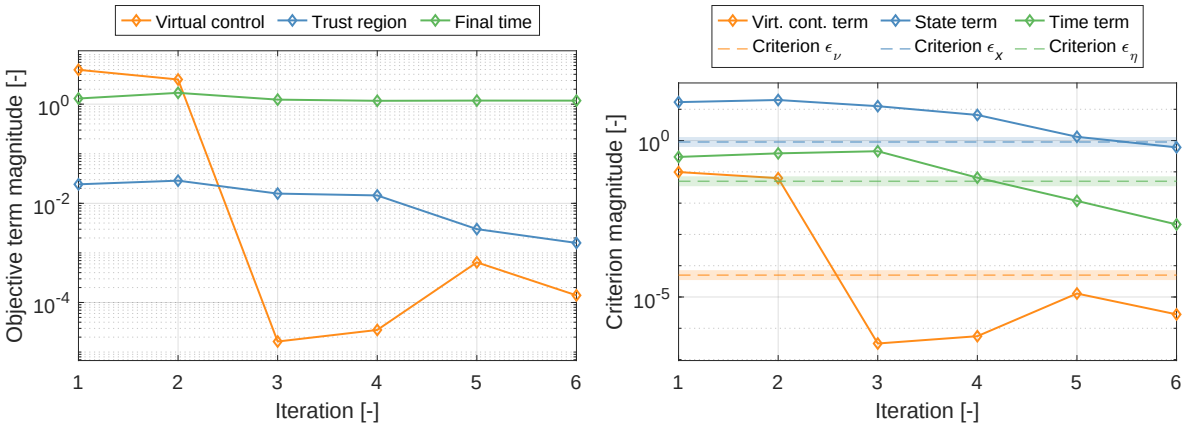
The two main insights that can be extracted from Figure 7.9 are:

- In Figure 7.9a, it can be seen that the manoeuvre duration t_f, which is to be minimised, strongly increases for the first two iterations of the SCP algorithm. This behaviour was observed for the majority of the cases where the virtual-control term of the objective function was dominant during the first iteration(s). A successful strategy that was found to mitigate this challenge was the inclusion of the *hard* trust-region constraint on the time dilation factor δ_η, as was discussed in Section 4.7. In this specific case, this hard constraint, which limits Δt_f (or, equivalently, $\Delta\eta$) to 30% of its reference value, actively constrains the increase in t_f for the first two iterations. Without this strategy, undesired changes in η were observed of multiple factors compared to its reference value.

- In Figure 7.9b, the behaviour of the *successive approximation* convexification technique that is applied to the final boundary condition is illustrated. As can be observed, the solutions of the convex subproblems of the initial iterations (1,2 and 3) do not satisfy the third-order polynomial that represents the time-dependent boundary condition for the q_2 parameter, due to the fact that the difference in the time dilation factor ($\Delta\eta$), and therefore also the approximation error, are relatively large. As $\Delta\eta$

becomes smaller for subsequent iterations, the approximation error reduces for this boundary condition. Eventually, the converged solution can be seen to (visually) satisfy the time-dependent boundary condition.

In Figure 7.10, the convergence process of the objective function and the convergence criteria is presented.



(a) Convergence history of the objective function.

(b) Convergence history of the convergence criteria.

Figure 7.10: Convergence process of the objective function and convergence criteria of the SCP algorithm for the minimum-time, attitude-constrained reorientation problem presented in Table 7.3.

Several interesting observations can be made based on these figures:

- For this case, the virtual control variable is used to an even larger extent to find a feasible solution for the first two convex sub-problems than for the minimum-energy problem of Section 7.1. The weighted virtual-control objective term has a magnitude that is factors larger than the second-largest (η) objective term. Due to the relatively high magnitude of this term, the SCP algorithm is encouraged to reduce the use of virtual control significantly. From iteration 3, it can be seen that the virtual control use is at a level that is acceptable for convergence.

- In Figure 7.10b, it can be seen that the convergence criterion for the final time ϵ_η was met much earlier than the convergence criterion for the state ϵ_x. Logically, this depends on the specification of the magnitudes of the performance criteria ϵ, but even when ϵ_η was made significantly more strict (lower magnitude), it was observed that the state term was in almost all of the cases the term that was eventually slower to converge. This means that the ϵ_η parameter has a smaller influence on the algorithm behaviour than the ϵ_x parameter.

In Figure 7.11, a 3-D visualisation is provided of the SCP and NLP solutions for the body-fixed x- and z-axis of the spacecraft, the corresponding keep-out and keep-in attitude constraints, and the convergence history of the SCP solution for the body-fixed x-axis.

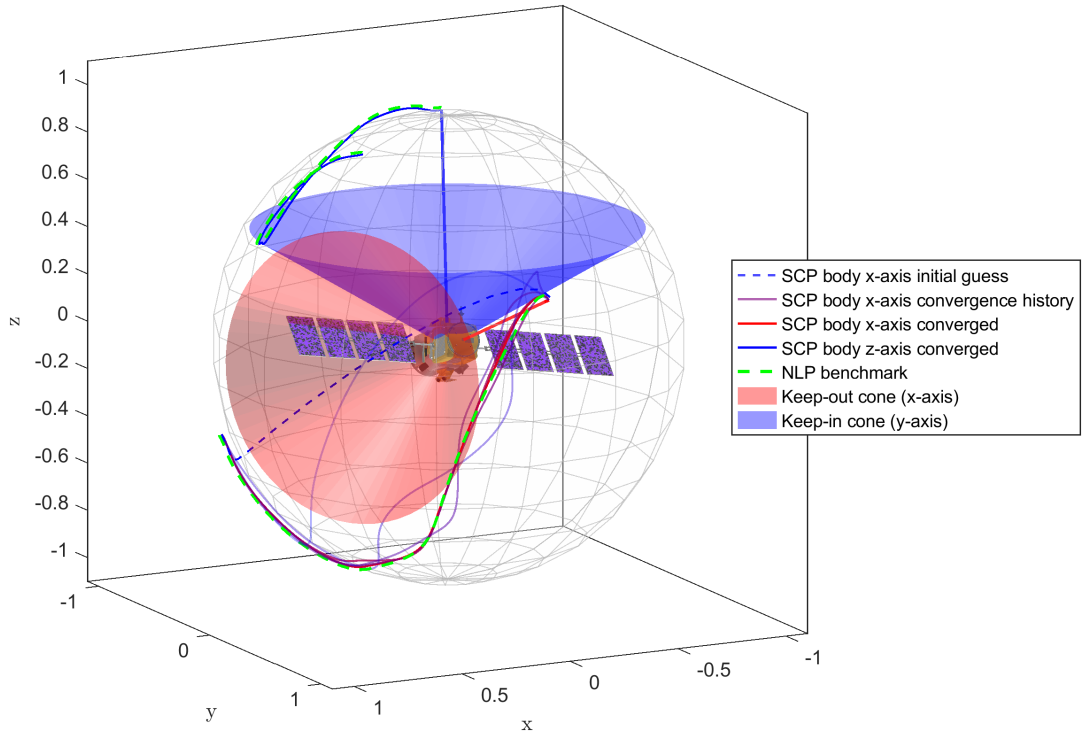


Figure 7.11: 3-D visualisation of the guidance solutions of the SCP and NLP algorithms for the minimum-time, attitude-constrained reorientation problem from Table 7.3.

The most interesting observation that can be made regarding Figure 7.11 is that the body-fixed x-axis of the spacecraft seems to violate the conical keep-out constraint for the initial iterations of the SCP algorithm. Intuitively, this should not be possible, as the convex solving algorithm ECOS should not allow for constraint violations. However, it was discovered that these violations are a consequence of a complex interaction between the virtual control and the (convex) quadratic attitude constraints. It was found that the significant use of virtual control for the first two iterations of the SCP algorithm, which was identified in Figure 7.10a, allowed for a violation of the quaternion norm-preserving dynamic and kinematic equations. As a result, the quaternion norm is not equal to 1 for several nodes of these trajectories. Because the preservation of the quaternion unit norm is the main requirement for the validity of the convex attitude constraint formulation, as outlined in Subsection 4.4.4, this resulted in the situation where the attitude constraints could be violated. This phenomenon was observed frequently but posed no major issue as the use of virtual control and the constraint violations were reduced to an acceptable level before a converged solution was obtained.

7.2.5. Further verification and analysis

Similar as for the minimum-energy problem from Section 7.1, the verification of the SCP algorithm for the minimum-time problem mostly relied on the fact that it shows similar performance as the NLP benchmark algorithm in Table 7.4, and similar results in Figure 7.8 and Figure 7.11. In this section, the propagated results and the constraint satisfaction are analysed. In Figure 7.12, the SCP control profile is shown with the corresponding ellipsoidal constraint.

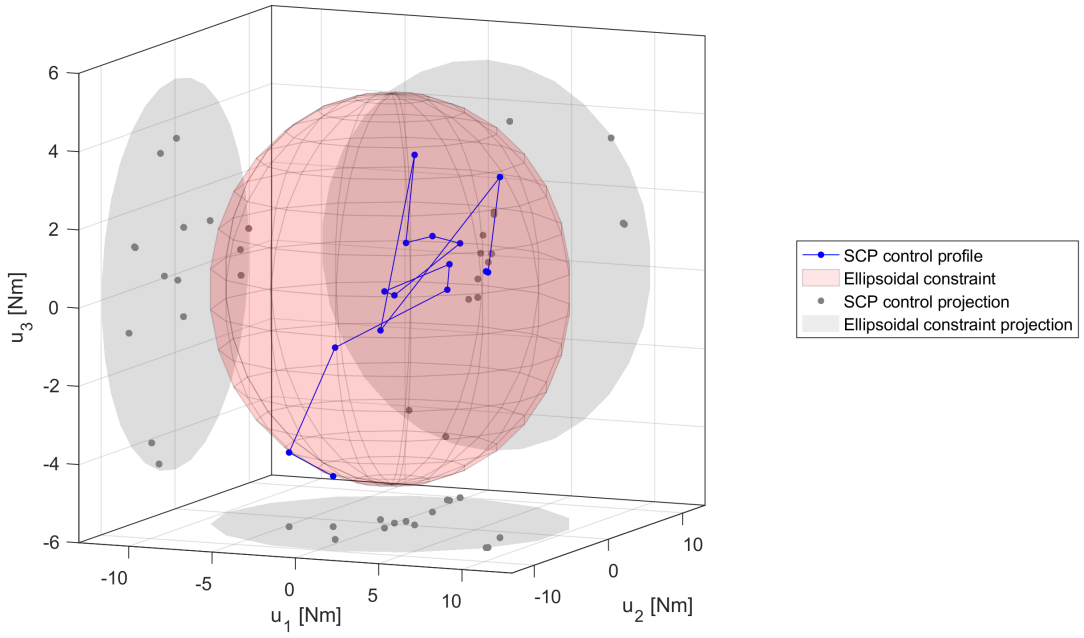
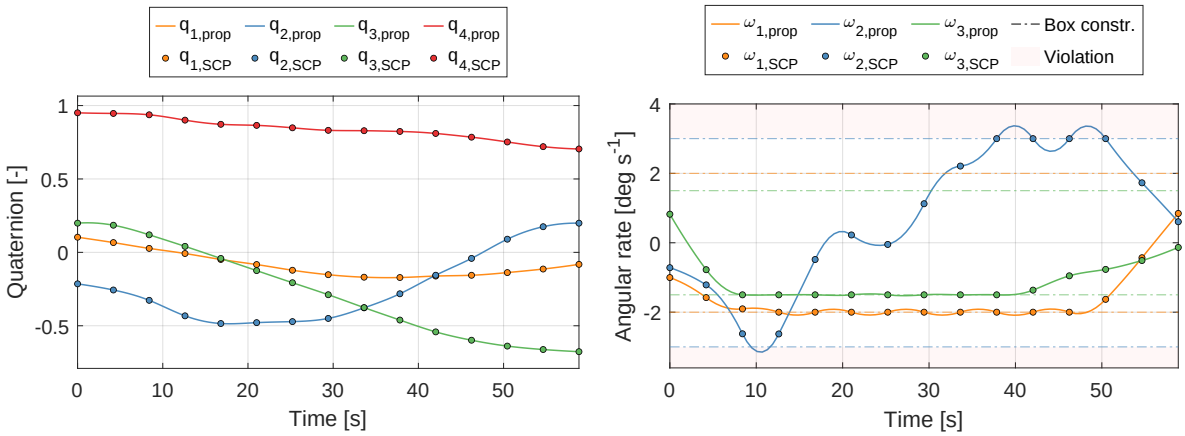


Figure 7.12: 3-D visualisation of the control profile obtained from the SCP algorithm and the ellipsoidal constraint on the control, for the minimum-time, attitude-constrained reorientation problem from Table 7.3.

The control behaviour is relatively erratic (in line with Figure 7.8c) but can be seen to satisfy the ellipsoidal constraint at all grid nodes. Due to the first-order-hold modelling of the control, it automatically follows that it can be guaranteed that the ellipsoidal constraint is satisfied for the entire reorientation manoeuvre, including the inter-nodal segments. This is an interesting advantage of the SCP algorithm, because this is not the case for pseudospectral NLP methods, which are known to often find solutions that violate control constraints and cannot be executed by the spacecraft attitude control system.

In Figure 7.13, the time histories that correspond to the propagated state parameters are shown, along with the SCP solution at the grid nodes. For verification purposes, the box constraint on the angular rate is indicated in the figure as well.



(a) SCP solution and propagated time histories of the quaternion parameters.

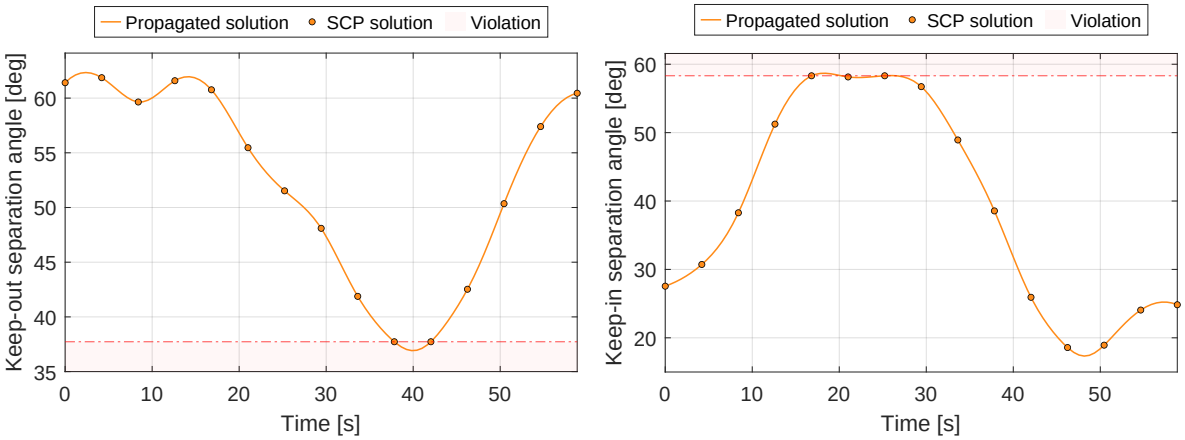
(b) SCP solution and propagated time histories of the angular rate parameters.

Figure 7.13: SCP solution and propagated time histories of the state parameters for the reorientation test case of Table 7.3.

As becomes evident from Figure 7.13b, significant constraint violations are present in the angular rate time histories that are obtained from propagating the SCP control profiles. When comparing the results of Figure 7.13b with the control profiles from Figure 7.8c, it becomes apparent that the strong 'spikes' or oscillations in the control profiles are in phase with these violations of the box constraint on the angular rate. Apparently,

in its search to find a locally optimal solution, the SCP algorithm 'cheats' by introducing constraint-violating angular rate oscillations in order to obtain a solution that is more optimal. This behaviour was detected in most minimum-time problems and, with violations of up to 35%, posed a challenge to the usability of the SCP-based optimisation algorithm developed in this study. Therefore, finding a method to solve this problem became one of the research priorities of this study. This method was successfully developed and is presented in Section 8.3.

Using the propagated results from Figure 7.13, the inter-nodal violations of the attitude constraints can be analysed. In Figure 7.7, the angular separation between the body-fixed x-axis and body-fixed z-axis and, respectively, the central axes of the keep-out and keep-in constraints are shown.



(a) Constraint satisfaction of the attitude keep-out constraint.

(b) Constraint satisfaction of the attitude keep-in constraint.

Figure 7.14: Constraint satisfaction of the attitude keep-out and keep-in constraints of the SCP solution for the minimum-time, attitude-constrained reorientation problem formulated in Table 7.3.

Slight constraint violations can be observed in Figure 7.14. A more extensive analysis of these violations is provided for the Monte Carlo analyses discussed in Chapter 9.

This page was intentionally left blank.

8

Performance Analysis and Optimisation

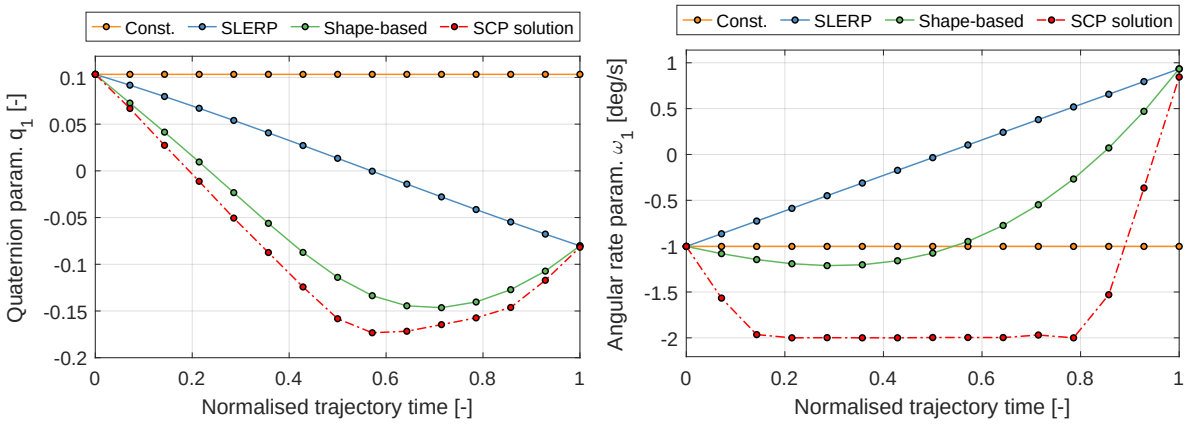
In Chapter 7, the performance of the SCP-based optimisation algorithm developed in this study was analysed on a single-case level using only two spacecraft reorientation test cases. This chapter aims to analyse the SCP algorithm on a more detailed level to obtain a better understanding of the influence of the most important design elements and parameters of the algorithm on its performance. In addition, where possible, the performance of the SCP algorithm is improved. It is noted that several of these performance improvements were already taken into account for the two test cases that were analysed in Chapter 7. However, it seemed logical to first verify and study the performance of the SCP algorithm on a higher level before presenting this more in-depth analysis of its specific features. To assess the performance of the SCP algorithm, Monte Carlo analyses were performed in line with the set-up introduced in Chapter 6. For some of these analyses, no outliers are shown in order to focus on the median performance of the SCP algorithm. Furthermore, it is stressed that this chapter analyses the *relative* performance of the SCP algorithm as a function of several design features and parameters. The *absolute* performance of the algorithm is studied in Chapter 9.

In terms of structure, Section 8.1 presents an analysis of the different initialisation strategies that were introduced in Section 4.2. Then, in Section 8.2, the performance of the novel method to discretise the minimum-energy objective function is studied. Afterwards, Section 8.3 presents a novel way to enforce constraints on inter-nodal segments. In Section 8.4, the effects of the adaptive trust region technique on the robustness of the SCP algorithm is analysed. Next, in Section 8.5, the objective function weight factors are tuned. In addition, in Section 8.6, the computation times of the SCP algorithm are analysed. Finally, in Section 8.7, the sensitivity of the performance of the algorithm to two important algorithm parameters is studied.

8.1. Initialisation strategies

As discussed in Section 4.2, three different initialisation strategies were compared and studied during this research project. In this section, the influence of these different initialisation strategies on the performance of the SCP algorithm is analysed. Regarding this analysis, it was decided to replace the straight-line initialisation method presented in Section 4.2 with a constant initialisation method, as only insignificant differences were observed between the performance of the straight-line and SLERP initialisation methods. Furthermore, from a robustness perspective, it was decided to analyse the influence of a poor initialisation method on the algorithm performance. To illustrate, in Figure 8.1, three initial guesses are shown for the q_1 and ω_1 parameters of a reference test case. These initial guesses correspond to the three different initialisation techniques that were analysed: the constant, SLERP and shape-based initialisation methods. In addition, the SCP solutions for these two state parameters are provided for comparison purposes.

In Figure 8.1, it can be seen that the shape-based initialisation method approximates the optimal (SCP) solution to the reorientation problem better than both the constant and SLERP initialisation techniques. However, it must be noted that this is not the case for all situations and reorientation problems. Especially for attitude-constrained test cases, it was found that the initial guesses commonly violate the attitude keep-out constraints and, therefore, often are not an accurate approximation of the optimal solution to the problem.



(a) Initial guesses and SCP result for the quaternion parameter q_1.

(b) Initial guesses and SCP result for the angular rate parameter ω_1.

Figure 8.1: Initial guesses for the state parameter q_1 and the angular rate parameter ω_1 for three different initialisation techniques. The test case considered is a minimum-time problem with both attitude keep-out and keep-in constraints, in accordance with the formulation presented in Subsection 4.10.2. In the legend, 'const.' refers to the *constant* initialisation technique.

To get a more thorough understanding of the influence of these different initialisation techniques on the performance of the SCP algorithm, two series of Monte Carlo analyses were performed: one for the relatively easy minimum-energy, attitude-unconstrained reorientation problem, and one for the more difficult minimum-time, attitude-constrained (both keep-out and keep-in constraints) problem. The set-up of the Monte Carlo analysis and the evaluation of its results were discussed in Chapter 6. In Figure 8.2, the influence of the three initialisation methods on the performance of the SCP algorithm is shown for the minimum-energy problem. As noted earlier, no outliers are shown in this figure. The reason for this is that the focus of this chapter is to analyse the *typical* or *median* performance of the SCP algorithm. A detailed analysis of the outliers is presented in Chapter 9. In Figure 8.2, the performance is evaluated based on the metrics that were introduced in Chapter 6. Furthermore, each Monte Carlo analysis consisted of 500 test cases.

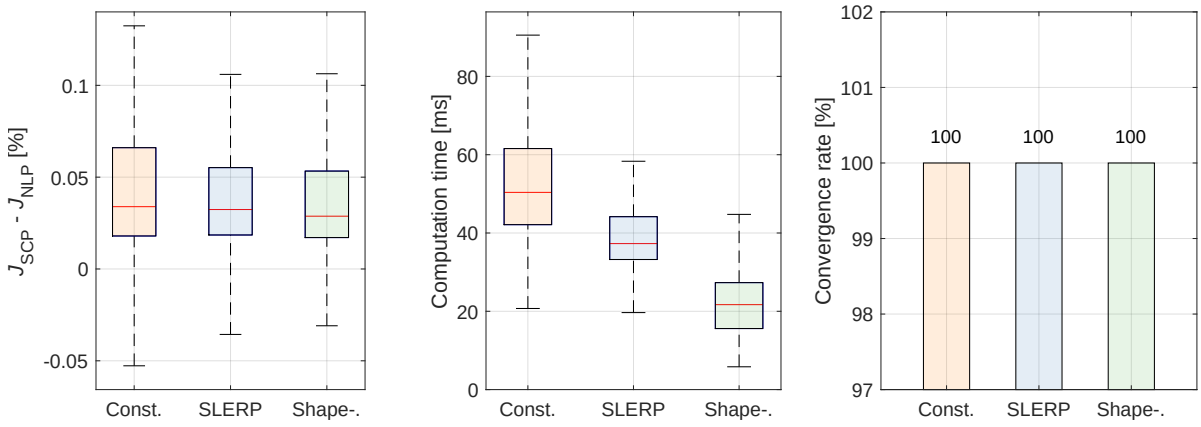


Figure 8.2: Performance of the SCP algorithm in terms of optimality, computational efficiency, and robustness as a function of different initialisation strategies. The Monte Carlo analyses considered a minimum-energy, attitude-unconstrained reorientation problem and consisted of 500 test cases. 'Const.', and 'Shape-.' refer to the constant and shape-based initialisation techniques, respectively.

From Figure 8.2, it can be concluded that the shape-based initialisation strategy strongly outperforms the constant and SLERP techniques that were used in other studies, for instance, in the works of McDonald et al. (2020) and Reynolds et al. (2021). The effects on the computation time can be considered to be significant, as a factor 2 reduction in median computation time can be observed. Furthermore, in terms of optimality, it can be seen that the median performance is comparable, from which it can be concluded that the initialisation method does not influence the optimality of the obtained guidance solutions. Finally, the 100% convergence rates support the notion of some researchers that the convergence of SCP-based convexification methods is initialisation-independent for this problem (McDonald et al., 2020).

In Figure 8.3, the influence of the three different initialisation strategies on the performance of the SCP algorithm in terms of optimality, computational efficiency, and robustness is provided for the more complex minimum-time, attitude-constrained reorientation problem. Again, the Monte Carlo analyses consisted of 500 test cases. It should be noted that the obtained convergence rates are relatively low because some Monte Carlo cases were found to be infeasible. However, this chapter focuses on the *relative* performance, while the *absolute* performance is studied in Chapter 9.

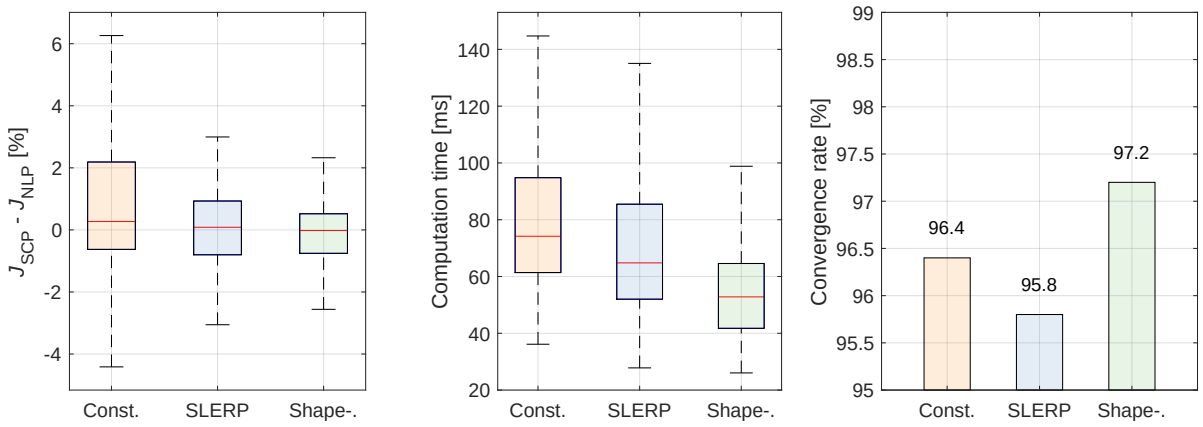


Figure 8.3: Performance of the SCP algorithm in terms of optimality, computational efficiency, and robustness as a function of different initialisation strategies. The Monte Carlo analyses considered a minimum-time, attitude-unconstrained reorientation problem and consisted of 500 test cases. 'Const.', and 'Shape-.' refer to the constant and shape-based initialisation techniques, respectively.

It can be observed that for the minimum-time problem shown in Figure 8.3, the gain in computational efficiency is there, but smaller than for the minimum-energy problem of Figure 8.2. This is in line with expectations, as for an attitude-constrained problem, the initial guesses often violate attitude constraints and, therefore, approximate the (locally) optimal solutions not as well as for the unconstrained problem. Moreover, minimum-time solutions often closely approach the limits of the box constraint on the angular rate, something not well represented by the more gradual profiles of the shape-based initialisation method. Perhaps even more interesting is the fact that the convergence rate varies for the three different initialisation methods. This contradicts other studies, such as the work of McDonald and Wang (2019), that proclaim that the convergence of SCP-based optimisation methods is independent of poor initialisation. It seems that this is not the case for more complex optimisation problems, such as the attitude-constrained reorientation problem. In addition, it is interesting to note that the constant initialisation technique achieves a better convergence rate than the SLERP technique.

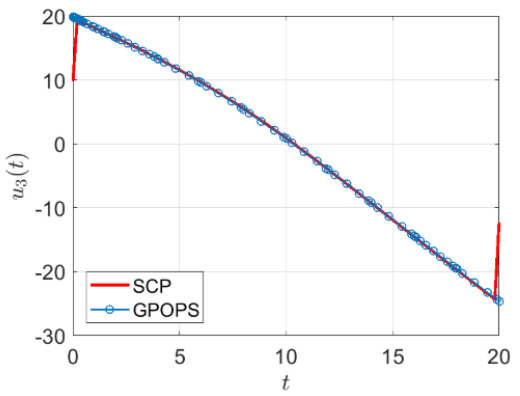
8.2. Objective function discretisation

As discussed in Subsection 4.5.3, two distinct ways to discretise the Lagrange (integral) term of the minimum-energy objective function were studied during this research project. These two approaches, the *trapezoidal* method and the method of *successive approximation of the exact cost* are analysed in, respectively, Subsections 8.2.1 and 8.2.2.

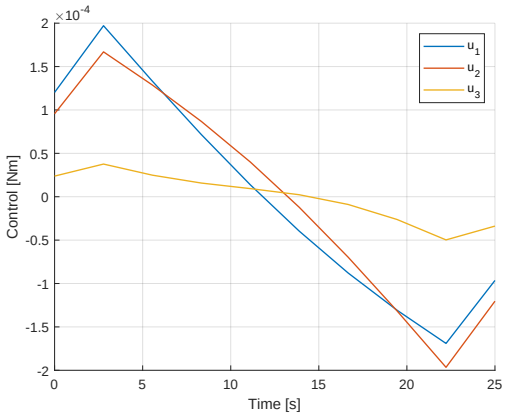
8.2.1. Trapezoidal method

In early stages of this research project, it was discovered that in several studies encountered in literature, also ones in which the SCP framework was applied to the spacecraft reorientation problem, the trapezoidal discretisation method seems to have been implemented incorrectly. Examples are the studies of McDonald et al. (2020) and Reynolds et al. (2021)[1]. It appears that these studies did not multiply the cost (energy) evaluations at the first and last grid points with half the weight relative to the cost evaluations at the inner grid points, in line with the formulation presented in Subsection 4.5.3. As a result, a strong, unexpected kink in the obtained control profiles can be observed near the boundaries of the manoeuvre, which negatively impacts the optimality of the obtained solution. Figure 8.4 shows two examples of this alternative implementation of the trapezoidal discretisation method, obtained from other studies.

[1] Through personal correspondence with the author.



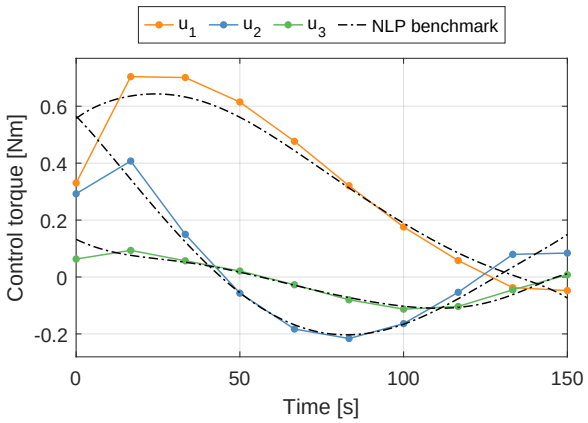
(a) u_3 control profile for a minimum-energy reorientation problem studied in the work of McDonald et al. (2020).



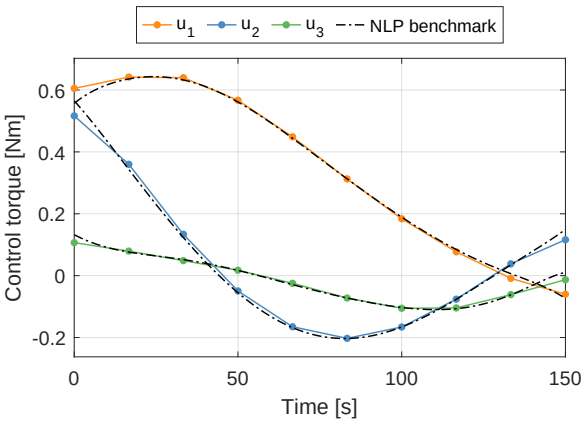
(b) Control profiles for a minimum-energy reorientation problem corresponding to the work of Reynolds et al. (2021).[2]

Figure 8.4: Illustrations from literature of control profiles that resulted from an alternative implementation of the trapezoidal discretisation of a minimum-energy objective.

It can be observed that the effects of this alternative implementation are larger (on the time scale) as the number of grid nodes, K, is lower, as McDonald et al. (2020) defined $K = 100$ and Reynolds et al. (2021) defined $K = 10$ for the control solutions presented in Figure 8.4. Implementing the trapezoidal rule using the scheme presented in Subsection 4.5.3, improves the optimality of the obtained solutions, as is illustrated in Figure 8.5.



(a) Minimum-energy reorientation problem with an alternative implementation of the trapezoidal quadrature, which applies an equal weight to all grid points. The corresponding optimality of the solution ($J_{SCP} - J_{NLP}$) is 2.6106%.



(b) Minimum-energy reorientation problem with the implementation of the trapezoidal rule from Subsection 4.5.3. The corresponding optimality of the solution ($J_{SCP} - J_{NLP}$) is 0.1226%.

Figure 8.5: Control profiles of the SCP and NLP benchmark algorithms for a single minimum-energy test case for two different implementations of the trapezoidal discretisation rule.

From Figure 8.5, it can be concluded that the implementation of the trapezoidal rule in this study leads to more optimal solutions. However, although this implementation of the trapezoidal quadrature rule represents a significant step towards finding an optimal solution, it is still possible to observe (visual) differences between the control profiles of the SCP and NLP benchmark algorithms. These are predominantly visible at the boundaries of the reorientation manoeuvre. It was thought that this source of non-optimality is caused by the inferior integration accuracy of the trapezoidal method, which is used in the SCP algorithm, compared to the pseudospectral quadrature of the NLP benchmark algorithm. It is widely known that pseudospectral quadrature rules are the most accurate way to perform an integration for a discretised problem (Sagliano, 2018).

[2]Through personal correspondence with the author.

This observation led to the development of the alternative discretisation technique presented in Subsection 4.5.3. This method, which was developed and pioneered in this study, is analysed in the following subsection to evaluate its performance against the trapezoidal approach.

8.2.2. Successive approximation of the exact cost

Using the novel method proposed in Subsection 4.5.3, which leverages the iterative nature of the SCP algorithm, it was found that results could be obtained that were more optimal than the ones found using the trapezoidal discretisation method. In Figure 8.6, the control profiles obtained using this approach are presented for the same test case that was considered in Figure 8.5.

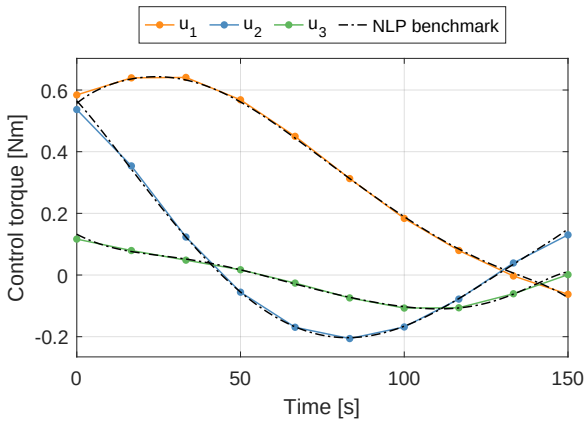


Figure 8.6: Control profiles of the SCP and NLP benchmark algorithms for the single minimum-energy test case of Figure 8.5. This time, the SCP algorithm uses the technique of successive approximation of the exact cost to integrate the minimum-energy term in the objective function. The corresponding optimality $J_{SCP} - J_{NLP}$ is equal to 0.0359%.

In Figure 8.6, it can be seen that the discretisation method developed in this study improves the optimality of the SCP guidance trajectory. Although its improvement in optimality in *relative* terms is high, as the optimality 'overcost' relative to the NLP benchmark solution is decreased by around 60%, in absolute terms, the gain in optimality is small, less than 0.1%. The remaining differences between the SCP and NLP control profiles are thought to result from the FOH control modelling of the SCP algorithm. In order to be able to draw general conclusions regarding the optimality improvements that follow from this discretisation method, a Monte Carlo analysis was performed that consisted of 500 minimum-energy, attitude-unconstrained test cases. In Figure 8.7, the results of this Monte Carlo analysis in terms of optimality and computational efficiency are shown. 100% percent convergence rates were obtained for all four Monte Carlo analyses. Again, no outliers are shown to focus on the *mean* performance of the SCP algorithm.

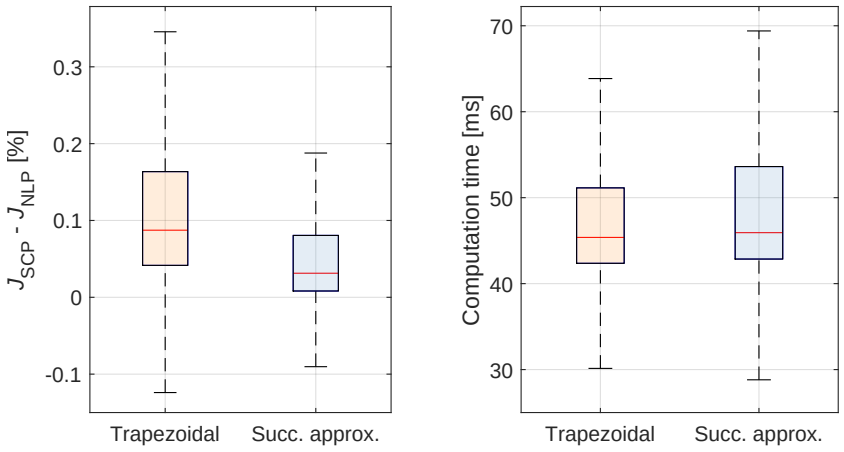


Figure 8.7: Optimality and computational efficiency of the SCP guidance trajectories obtained using two different discretisation techniques for the minimum-energy objective function. These results are based on a Monte Carlo analysis of 500 test cases. 'Succ. approx.' refers to the successive approximation of the exact cost discretisation technique.

In line with the results of the single test case that was studied in Figure 8.6, it can be observed that the *successive approximation of the exact cost* technique developed in this study consistently improves the optimality of the guidance solutions of the SCP algorithm. However, as mentioned earlier, the performance gain is tiny in absolute terms. In fact, it was determined to be of little significance for the spacecraft reorientation problem studied in this research project, as in this study, the optimality is mostly analysed in the order of percentages, not in tens of percentages. Therefore, the trapezoidal discretisation technique was used for the analyses presented in the remainder of this report. However, for potential applications of the SCP framework to other optimisation problems for which optimality has a higher priority, this method to discretise a Lagrange objective could be an essential improvement to match the performance of NLP optimisation methods.

8.3. Constraint satisfaction

As was discussed for the minimum-time test case of Section 7.2, it was frequently observed during inspections of Monte Carlo test cases that the guidance solutions of the SCP algorithm led to large constraint violations on the inter-nodal grid segments. In particular for the minimum-time, rate-constrained problem, severe constraint violations of up to 35% were observed. Moreover, these violations were often accompanied by undesired oscillations in the control and rate profiles of the SCP guidance solutions. These oscillations are problematic because they might be challenging to actuate upon by a real, physical attitude control system. Therefore, it could be concluded that these constraint violations represented a major challenge for the practical usability of an SCP algorithm.

In Subsection 5.2.2, it was discussed that the NLP benchmark algorithm uses a strategy that performs a Lagrange interpolation of the optimisation parameters to impose additional, inter-nodal constraints on the state variables. However, this strategy was unfortunately not an option to resolve this challenge for the algorithm developed in this study. The reason for this is that the SCP algorithm does not model the state parameters with Lagrange polynomials. To the knowledge of the author, this problem of inter-nodal constraint violations has not been studied or mentioned in literature on SCP, let alone that an effective solution has been proposed. Therefore, an SCP-specific method was developed during this study to solve this problem. This method uses information from the discretisation process of the dynamics and kinematics (Subsection 4.5.2) and is presented in the remainder of this section. In this case, the method is used to enforce the box constraint on the angular rate at the $(K-1)$ temporal midpoints of the grid segments.

The first step of this method is to perform the Runge-Kutta integration process described in Subsection 4.5.2 from the start of a segment to its midpoint (instead of to the end of the segment). The resulting matrices and vectors that are obtained at this point can be mathematically represented as

$$A_{k+\frac{1}{2}} := I_{n_x \times n_x} + \int_{\tau_k}^{\tau_{k+\frac{1}{2}}} A(\xi)\Phi_A(\xi,\tau_k)\,\mathrm{d}\xi \tag{8.1}$$

$$B^-_{k+\frac{1}{2}} := A_{k+\frac{1}{2}} \int_{\tau_k}^{\tau_{k+\frac{1}{2}}} \Phi_A^{-1}(\xi,\tau_k)\,B(\xi)\lambda_k^-(\xi)\mathrm{d}\xi \tag{8.2}$$

$$B^+_{k+\frac{1}{2}} := A_{k+\frac{1}{2}} \int_{\tau_k}^{\tau_{k+\frac{1}{2}}} \Phi_A^{-1}(\xi,\tau_k)\,B(\xi)\lambda_k^+(\xi)\mathrm{d}\xi \tag{8.3}$$

$$\boldsymbol{C}_{k+\frac{1}{2}} := A_{k+\frac{1}{2}} \int_{\tau_k}^{\tau_{k+\frac{1}{2}}} \Phi_A^{-1}(\xi,\tau_k)\,\boldsymbol{C}(\xi)\mathrm{d}\xi \tag{8.4}$$

$$\boldsymbol{r}_{k+\frac{1}{2}} := A_{k+\frac{1}{2}} \int_{\tau_k}^{\tau_{k+\frac{1}{2}}} \Phi_A^{-1}(\xi,\tau_k)\,\boldsymbol{r}(\xi)\mathrm{d}\xi \tag{8.5}$$

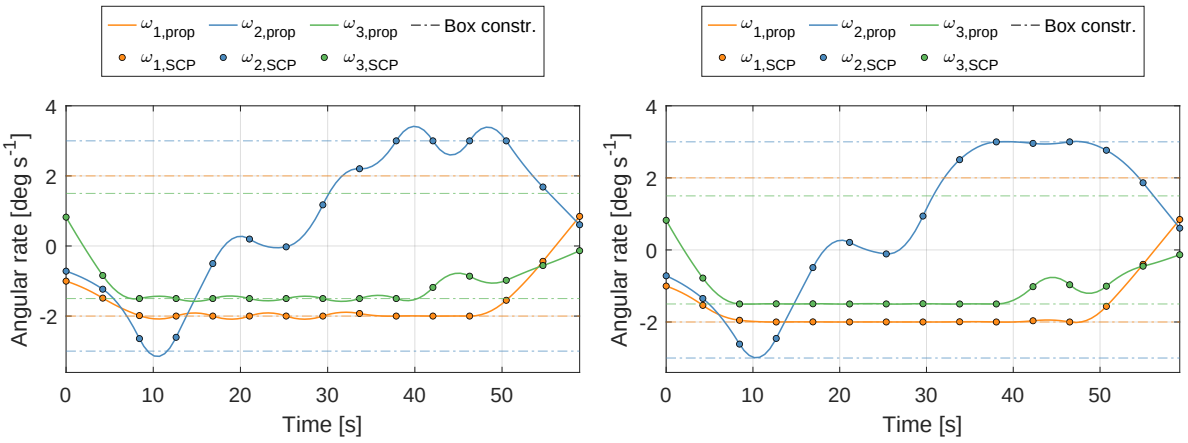
where the obtained matrices and vectors are referred to with the $(\cdot)_{k+\frac{1}{2}}$ subscript. As can be observed, the only difference between the matrices and vectors obtained for the dynamic and kinematic constraints (Equations 4.43 to 4.47) and the matrices and vectors obtained for the additional enforcement of the angular rate constraints (Equations 8.1 to 8.5), is the length of the integration domain. This is a useful observation, as it means that no additional integration process has to be performed. Instead, the intermediate results of the discretisation step from Section 4.5 can be stored and re-used for this purpose. Using the obtained matrices

and vectors, the additional angular rate constraints can be formulated as

$$\begin{aligned} -\boldsymbol{\omega}_{\text{box}} \quad &\leq \quad A_{k+\frac{1}{2}}(5:7,:)\boldsymbol{x}_k + B^{-}_{k+\frac{1}{2}}(5:7,:)\boldsymbol{u}_k \quad \dots \\ &+ \quad B^{+}_{k+\frac{1}{2}}(5:7,:)\boldsymbol{u}_{k+1} + \boldsymbol{S}_{k+\frac{1}{2}}(5:7)\eta + \boldsymbol{r}_{k+\frac{1}{2}}(5:7) \quad \leq \quad \boldsymbol{\omega}_{\text{box}}, \quad \forall \quad k \in 1,\dots,K-1 \end{aligned} \tag{8.6}$$

As can be seen in Equation 8.6, the optimisation parameters that are involved in the additional rate constraints, $\boldsymbol{x}_k$, $\boldsymbol{u}_k$, $\boldsymbol{u}_{k+1}$ and η are optimisation parameters corresponding to the grid of the discretised, convex sub-problem. As a result, this strategy does not require the introduction of additional optimisation parameters while enabling the enforcement of problem constraints on inter-nodal segments.

This approach leads to $2(K-1)$ additional (linear) inequality constraints in the ECOS problem formulation of Section 5.4, which enforce the box constraint on the angular rate at the temporal midpoint of every segment of the grid. In order to analyse the effectiveness of this strategy, in Figure 8.8 the propagated angular rate time histories are provided for the minimum-time test case that was studied and discussed in Section 7.2, both with and without the inclusion of the additional constraints of Equation 8.6.

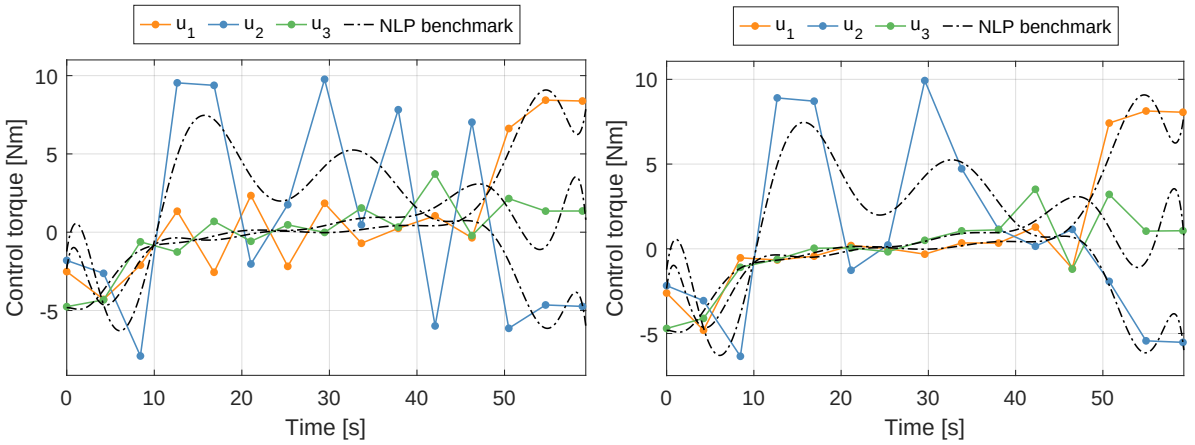


(a) Propagated angular rate time histories and SCP solution. Obtained without the additional constraints of Equation 8.6.

(b) Propagated angular rate time histories and SCP solution. Obtained after imposing the additional constraints of Equation 8.6.

Figure 8.8: Angular rate time histories obtained through propagation of the SCP control solution. The angular rate solution of the SCP algorithm is provided at the grid points. The parameters of the minimum-time test case are provided in Table 7.3.

It can be seen in Figure 8.8 that the technique that was developed to enforce the box constraint on the angular rate ($\boldsymbol{\omega}_{\text{box}}$) at additional nodes significantly reduces the severity of the constraint violations. Therefore, the preliminary conclusion can be drawn that this technique effectively reduces the magnitude of inter-nodal constraint violations. In Figure 8.9, the control profiles that correspond to the results of Figure 8.8 are shown.



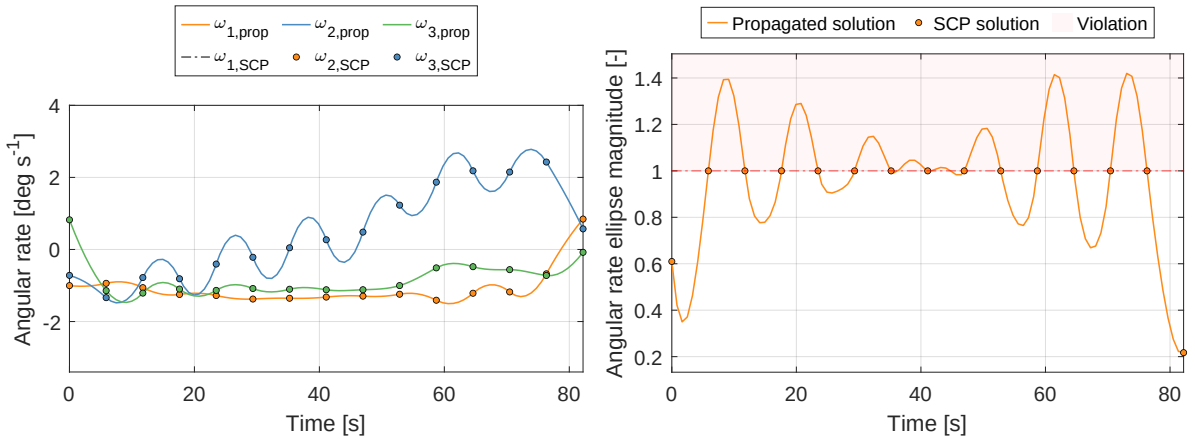
(a) Control solutions of the SCP and NLP benchmark algorithms. Obtained without the additional constraints of Equation 8.6.

(b) Control solutions of the SCP and NLP benchmark algorithms. Obtained after imposing the additional constraints of Equation 8.6.

Figure 8.9: Control solutions of the SCP and NLP benchmark algorithms for the minimum-time reorientation problem from Table 7.3.

As expected, the implementation of additional rate constraints through Equation 8.6 results in the disappearance of most of the oscillations in the control solution of the SCP algorithm. This is a desirable result, as a gradually changing control profile is typically easier to follow for the attitude control systems of most spacecraft.

As a further illustration, in Figure 8.10, both the propagated angular rate time history and constraint violations are shown for the same, minimum-time test case of Table 7.3. However, this time, an ellipsoidal instead of a box constraint is imposed on the angular rate (in the same format of the ellipsoidal control constraint of Equation 3.13). The reason for including this example is that the oscillations in both the control and angular rate parameters were found to be even more severe for this type of constrained reorientation problem.

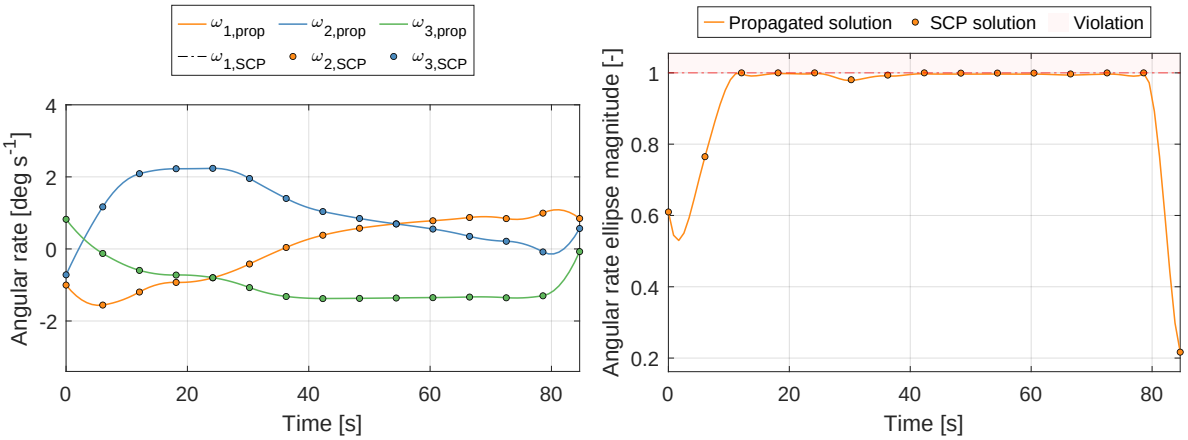


(a) Propagated angular rate time histories and SCP solution.

(b) Ellipsoidal rate constraint violation.

Figure 8.10: Angular rate time histories and angular rate constraint violations obtained through propagation of the SCP control solution. These results were obtained without the additional constraints of Equation 8.6. The parameters of the minimum-time test case are provided in Table 7.3.

It can be seen that the ellipsoidal angular rate constraint is violated by up to 40%, which is unacceptable. Figure 8.11 clearly shows the positive effects of imposing additional (ellipsoidal) rate constraints using Equation 8.6.



(a) Propagated angular rate time histories and SCP solution.

(b) Ellipsoidal rate constraint violation.

Figure 8.11: Angular rate time histories and angular rate constraint violations obtained through propagation of the SCP control solution. These results were obtained after imposing the additional constraints of Equation 8.6. The parameters of the minimum-time test case are provided in Table 7.3.

To obtain a thorough understanding of the effectiveness of this technique, a Monte Carlo analysis was performed to study the effects of the implementation of Equation 8.6 on the performance of the SCP algorithm in terms of optimality, computational efficiency and angular rate constraint satisfaction. This analysis was performed for the minimum-time, attitude-constrained reorientation problem defined in Subsection 4.10.2. The results of this Monte Carlo analysis are presented in Figure 8.12. The box plots for the results regarding optimality and computation times contain no outliers in order to focus on the median performance of the algorithm. For the violations, however, outliers are of high interest and, therefore, shown.

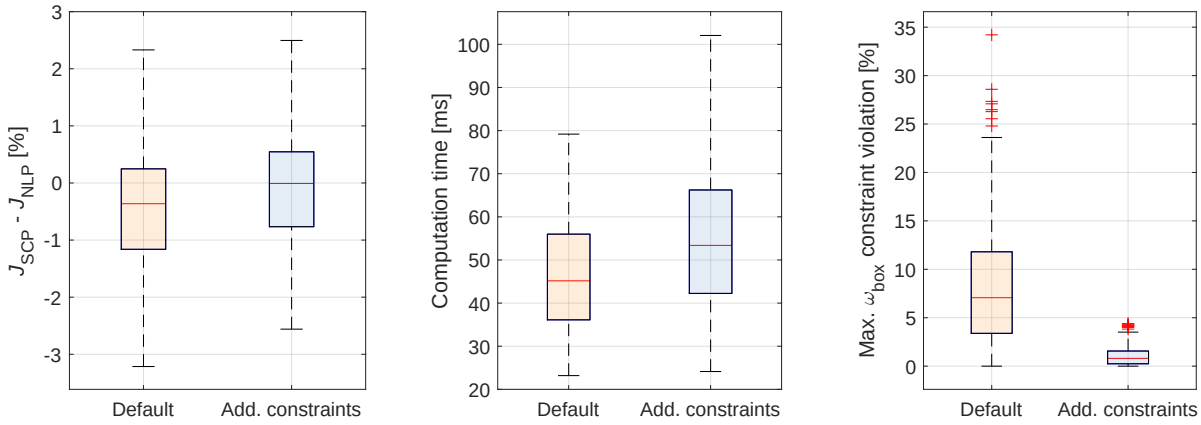
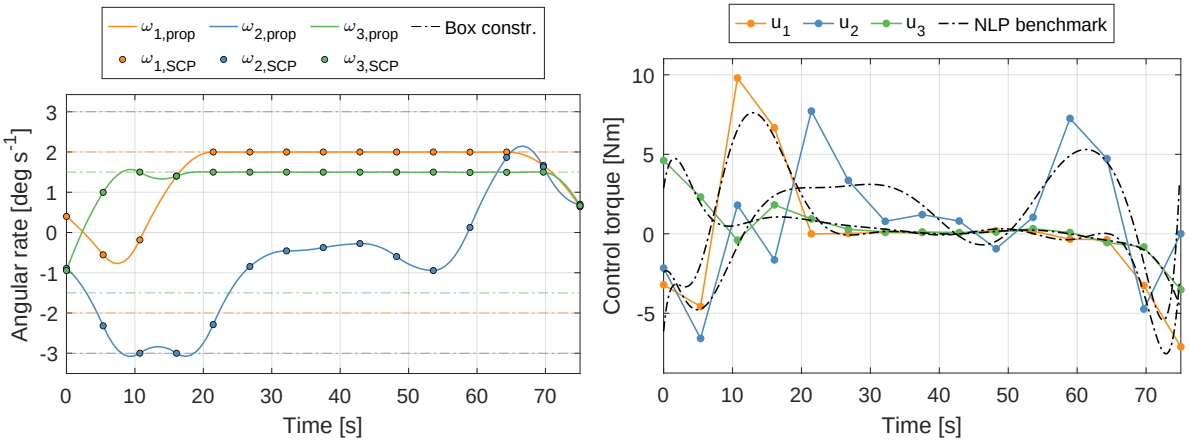


Figure 8.12: Performance of the SCP algorithm in terms of optimality, computational efficiency and angular rate constraint satisfaction, both with ('Default') and without ('Add. constraints') the additional rate constraints of Equation 8.6 for the minimum-time, attitude-constrained reorientation problem. Each Monte Carlo analysis consisted of 500 test cases.

Based on Figure 8.12, it can be concluded that the constraint satisfaction was significantly improved using the additional rate constraints, signalled by a decrease in the maximum box constraint violation from 35% to 5%. Furthermore, this approach can be seen to come at the cost of both optimality and computation time, which is no surprise. Restricting angular rate constraint violations could be expected to decrease the optimality while enforcing additional constraints could be expected to increase computation times. One might wonder why there are still constraint violations after the implementation of Equation 8.6. It was found that these violations occur at the midpoints of the *half*-segments of the grid where the box constraint is not enforced, even after the enforcement of Equation 8.6. An example of such a test case is provided in Figure 8.13. In this figure, the propagated angular rate time histories and control profiles are shown for the case that represents the worst violation (around 5%) from Figure 8.12.



(a) Propagated angular rate time histories and SCP solution

(b) Control solutions of the SCP and NLP benchmark algorithms.

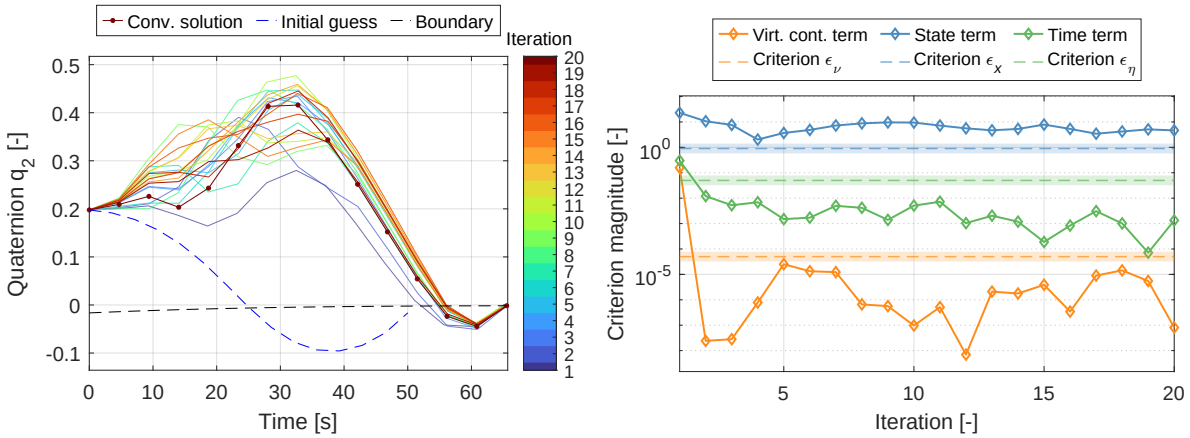
Figure 8.13: SCP solution corresponding to the largest rate-violating outlier of the box plot of Figure 8.12.

It can be observed that the violation occurs for the ω_2 parameter near the beginning of the manoeuvre. As expected, the solution seems to satisfy the rate constraints at the midpoint of all grid segments due to the additional constraints. However, there is still some room left for the SCP algorithm to use constraint violations to reduce the manoeuvre time and improve optimality.

For the purpose of this study, the obtained reduction sufficed as proof of the effectiveness of the developed method to enforce problem constraints on inter-nodal segments. However, if constraint violations for some future applications need to be reduced to a further extent, the same technique can be used to impose additional constraints on different points. Furthermore, this method could also be used to enforce other constraints, such as the attitude keep-out constraint, on inter-nodal segments. Moreover, it could be interesting to impose the additional constraints at different points than the midpoints of the grid segments. When carefully observing Figure 8.13, it can be seen that the extrema of the angular rate profiles occur at nearly the same time as the zero-crossings of the corresponding control profiles (not exactly, due to control cross-coupling). Using information from previous SCP iterations, it could be possible to impose the additional constraints near these points, potentially further reducing the constraint violations. Due to the scope of this research project, these topics are left as recommendations for further research.

8.4. Adaptive trust region

As outlined in Section 4.7, another design feature that was implemented and studied during this research project is the so-called *adaptive* trust region. For this type of trust region, the weight factor of the (soft) trust-region term that is augmented to the objective function of the convex sub-problems is not constant. It was found that this adaptive trust region improved the performance of the SCP algorithm by increasing its convergence rate for a specific set of minimum-time reorientation problems where the SCP algorithm struggled to find a local optimum. An example of such a test case is provided in Figure 8.14. This figure shows the convergence histories of the state parameter q_2 and the convergence criteria, which were obtained *without* the implementation of the adaptive trust region.



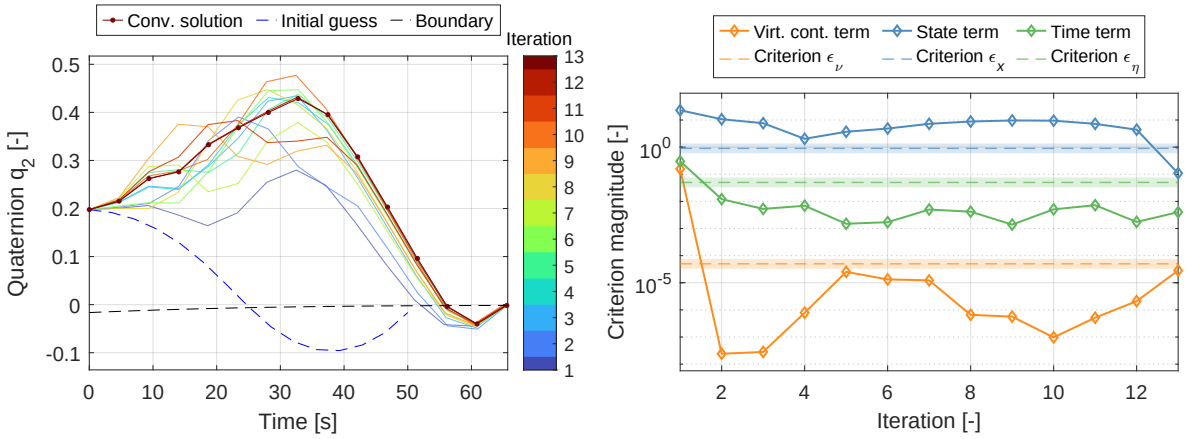
(a) Convergence history of the q_2 state parameter.

(b) Convergence history of the convergence criteria.

Figure 8.14: Example of a minimum-time reorientation problem for which the SCP algorithm failed to converge. The results were obtained without the implementation of the adaptive trust-region technique. The iteration limit of the SCP algorithm was set to 20.

In Figure 8.14, it can be observed that the SCP algorithm failed to converge within the limit of 20 SCP iterations. The algorithm seems to search in the right direction, as all solutions to the convex sub-problems have a very similar final time t_f. This final time of approximately 65 seconds was found to represent a local minimum through comparison with the solution of the NLP benchmark algorithm. However, it seems that there are a number of different manoeuvres for this problem that all have a similar manoeuvre duration and cost (t_f). As the SCP algorithm keeps finding different solutions to the subsequent convex sub-problems, the criterion ϵ_x is not met. It is noted that this behaviour was only observed for a very small number of minimum-time reorientation problems. This phenomenon is thought to be caused by the inherent larger degree of freedom that is characteristic of the minimum-time spacecraft reorientation problem.

In Figure 8.15, however, it is shown that the implementation of the adaptive trust-region strategy enables the SCP algorithm to converge for the test case of Figure 8.14.



(a) Convergence history of the q_2 state parameter.

(b) Convergence history of the convergence criteria.

Figure 8.15: Same test case as presented in Figure 8.14, but this time the SCP algorithm used the adaptive trust-region technique that was presented in Section 4.7.

From Figure 8.15, it can be concluded that the adaptive trust-region scheme from Section 4.7 enabled the SCP algorithm to converge to a solution. Through increasing the weight factor of the trust-region term (thereby decreasing its effective size) after the 10th iteration of the iterative SCP algorithm, the convex sub-problems 'encourage' a decrease in the difference between the state solution $\boldsymbol{x}$ and its most recent reference solution $\bar{\boldsymbol{x}}$. Consequently, the algorithm has converged after the 13th iteration.

8.5. Objective function weight factor tuning

As became apparent in Chapter 4, there is a large number of algorithm parameters involved in the design and implementation of the SCP-based optimisation algorithm. Two important ones are the (initial) weight factors of the virtual-control (w_{vc}) and trust-region (w_{tr}) terms of the objective function of the convexified sub-problems from Subsections 4.10.1 and 4.10.2. It was found that these two parameters have a significant influence on the performance of the SCP algorithm. This sensitivity has, to the knowledge of the author, not yet been researched in detail in available literature on SCP. To simplify the parameter analysis and tuning process, the weight factor of the optimisation term (either the minimum-energy or minimum-time term, depending on the problem formulation) was set to have an order of magnitude of approximately one. This was achieved through scaling of the optimisation parameters and the introduction of an adequate weight factor. Multiple Monte Carlo analyses were performed for both the minimum-energy and minimum-time problems to select the values for these two parameters that optimise the performance of the SCP algorithm. For the sake of conciseness, this section only presents the tuning process for the minimum-time problem.

The strategy that was followed to understand the influence of the virtual-control and trust-region weight factors on the performance of the SCP algorithm was to perform multiple Monte Carlo analyses while only varying a single parameter. Consequently, this analysis was performed independently for the virtual-control and trust-region weight factors.

Virtual-control weight factor

In Figure 8.16, the Monte Carlo results are provided in terms of optimality, computational efficiency, and the convergence rate. The performance in terms of accuracy was similar for all weight factors, which was to be expected as the convergence criteria were identical for all Monte Carlo analyses.

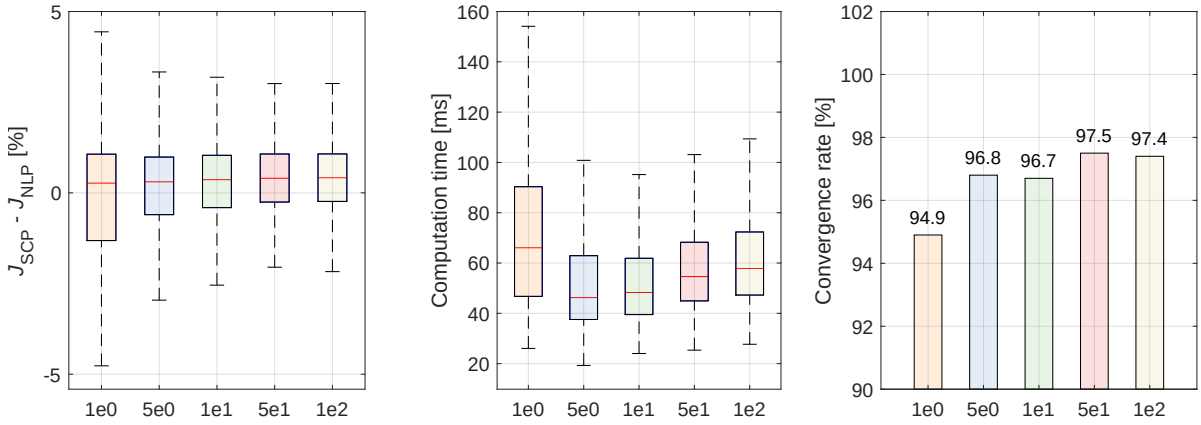


Figure 8.16: Performance of the SCP algorithm as a function of the virtual-control weight factor [-] for the minimum-time, attitude-constrained reorientation problem. Each Monte Carlo analysis consisted of 500 test cases.

In Figure 8.16, it can be seen that the weight factor w_{vc} does not influence the optimality of the solutions to a very large extent, apart from the fact that for $w_{vc} = 1$ the spread in the results becomes somewhat larger. However, in terms of computational efficiency, a trend can be observed where the performance increases (lower computation times) as the magnitude of the weight factor decreases. This trend continues until a certain point is reached where the computation times start to increase again. This increase can be observed for $w_{vc} = 1$. For this weight factor, it can be seen that the SCP algorithm struggles to find (optimal) solutions, signalled by both the increase in computation times and the decrease in the convergence rate. A possible explanation for the fact that a lower virtual-control weight factor negatively influences the convergence rate is that allowing the SCP algorithm to excessively use the virtual control variable hinders it from finding a feasible solution. A weight factor of 5×10^1 was selected for the SCP algorithm as it combines acceptable computational efficiency with a seemingly higher convergence rate than for lower weight factors.

Trust-region weight factor

In Figure 8.17, the results of the weight factor tuning process are provided for the trust-region weight factor of the minimum-time reorientation problem.

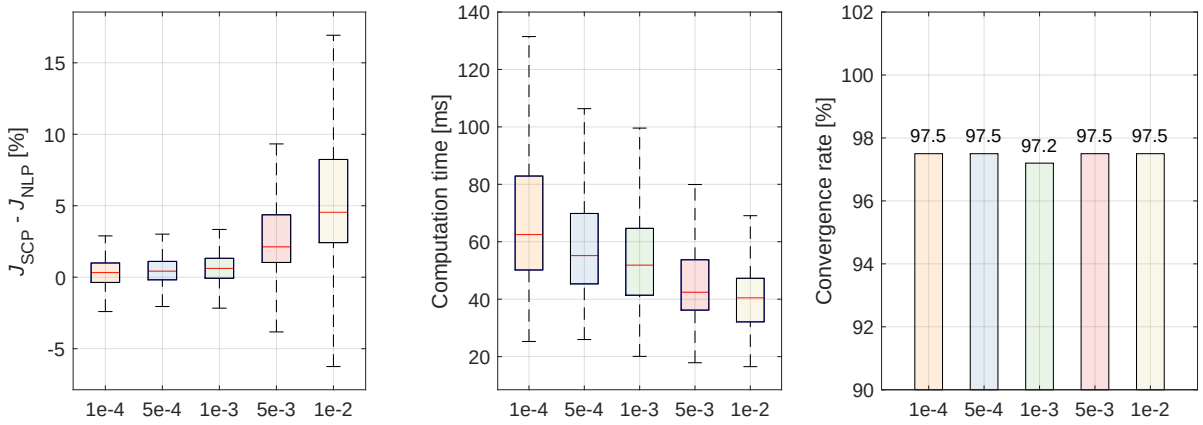


Figure 8.17: Performance of the SCP algorithm as a function of the trust-region weight factor [-] for the minimum-time, attitude-constrained reorientation problem. Each Monte Carlo analysis consisted of 500 test cases.

To begin with, it can be observed that, starting from a weight factor of 1×10^{-3}, the SCP solutions start to become sub-optimal compared to the solutions of the NLP benchmark algorithm. This behaviour is in line with expectations, as a higher trust-region weight factor (smaller trust region) results in a decrease of the state difference $(\boldsymbol{x} - \bar{\boldsymbol{x}})$ between two subsequent iterations. At some point, this can cause the algorithm to meet the convergence criterion ϵ_x for a particular iteration, while it would not have met this criterion if the trust region had been larger. In the latter scenario, the algorithm would have performed another iteration, bringing it closer to the locally optimal solution. This hypothesis is confirmed by the fact that, from a weight factor of 1×10^{-3}, it was found that the computation times decrease due to a lower number of SCP iterations that were performed. Furthermore, in terms of computational efficiency, there is a clear trend that computation time decreases as the trust-region weight increases. Finally, in terms of robustness, the lower convergence rate for a weight of 1×10^{-3} was thought to be accidental rather than representing a general trend, considering the equal convergence rates for all other weight factors. For the SCP algorithm, a trust-region weight factor of 1×10^{-3} was selected, as it provides optimal results combined with good computational efficiency.

8.6. Computation times

In literature on sequential convex programming, there has been some debate on the assessment of the computational efficiency of SCP-based optimisation algorithms. As was mentioned in Chapter 6, the main issue is that MATLAB takes significantly more time to execute the optimisation algorithm than a compiled language, which would be used on an embedded system in the context of autonomous guidance. Currently, most researchers assume that the sum of all solve times of the ECOS runs corresponding to the SCP iterations, serves as a proper indication of the computation time of the entire SCP algorithm (Reynolds et al., 2021; Szmuk et al., 2020). Reynolds et al. (2020a) reported that, in their study, the time spent outside the convex solver only accounted for 2% of the total solution time of the SCP algorithm, supporting this line of thought. However, as the spacecraft reorientation problem solves relatively quickly compared to more complex problems such as the 6-DoF, constrained, powered-descent problem, it cannot be known with certainty that this result holds for the SCP algorithm that was developed in this study. This uncertainty is, for instance, confirmed by results of a recent study of Preda et al. (2021), who studied a different type of spacecraft reorientation problem and found that for their implementation, the discretisation step of Subsection 4.5.2 represented around 50% of the total solution time (although it must be noted that the Python routine that was implemented in that study does not seem to be the fastest option).

Therefore, as the implementation of the SCP algorithm in this study uses highly-efficient C-code generated by MATLAB Coder for the discretisation step, it was thought to be very interesting to analyse how the discretisation step compares to the ECOS solving step in terms of runtime. In Figure 8.18, the results of this analysis are shown. In this figure, the median computation times of two Monte Carlo analyses of Chapter 9 are shown. These analyses consider the minimum-energy and minimum-time, attitude-constrained reorientation problems.

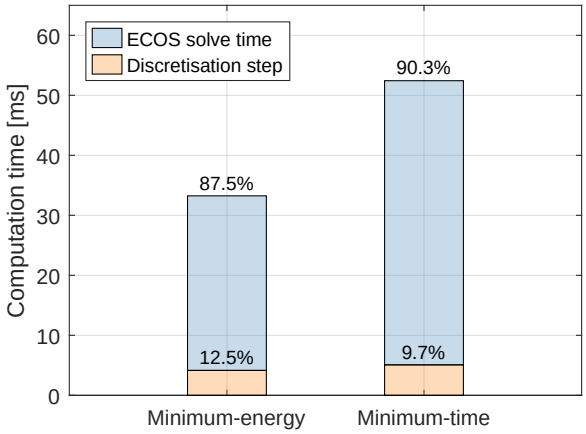


Figure 8.18: Median computation times of the Monte carlo analyses of the minimum-energy and minimum-time, attitude-constrained reorientation problems, split up into the computation times of the discretisation and ECOS solve steps.

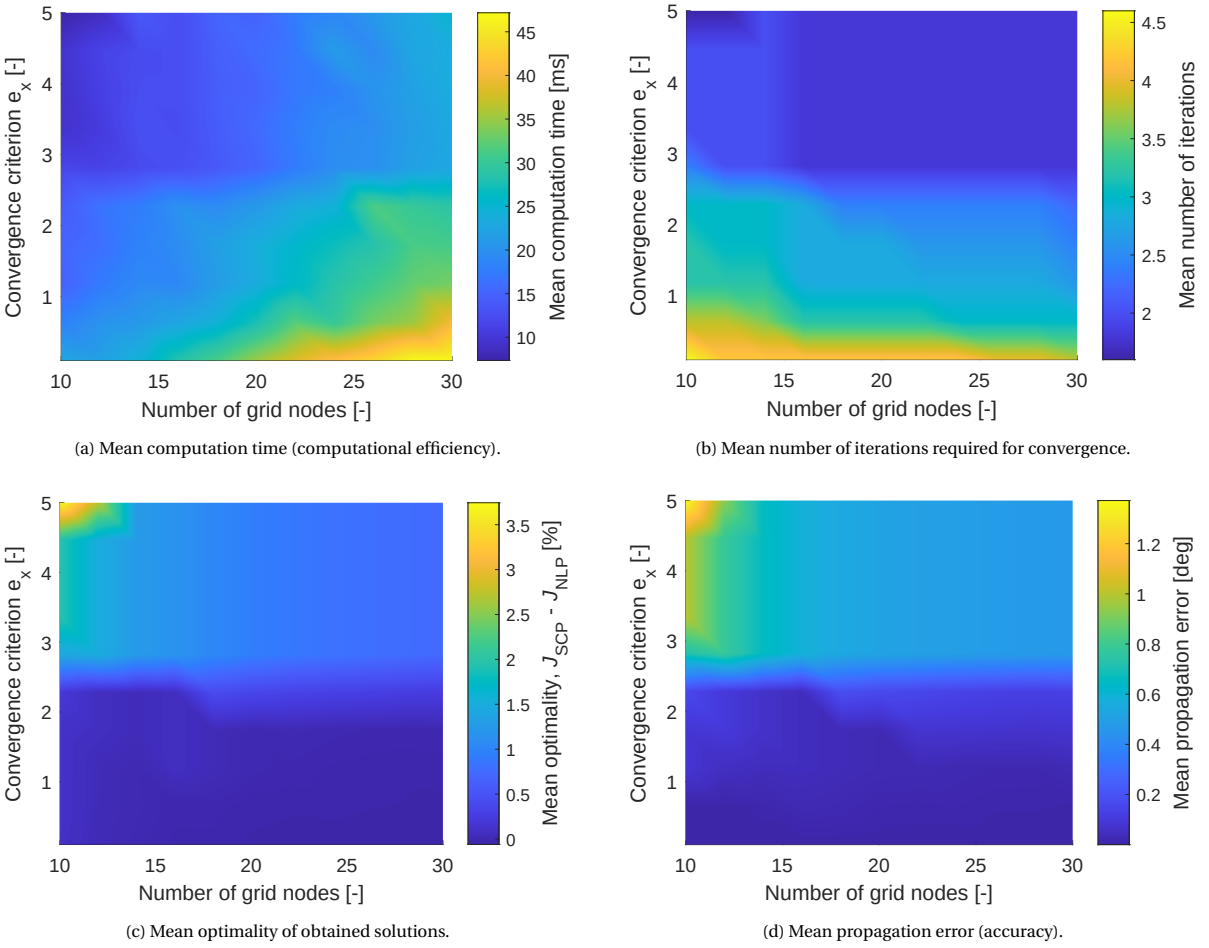
In Figure 8.18, it can be seen that the discretisation step represents approximately 10% of the total solution time. This is higher than reported by Reynolds et al. (2020a) but can, as mentioned before, be explained by the fact that the time required to find a solution for a spacecraft reorientation problem (at least in our set-up) seems significantly lower than the times reported in that study. However, it can be concluded that the assumption that the ECOS solve step can serve as a proper indication of the total solution time, still holds. Finally, the fact that it requires relatively less time to execute the discretisation step for the minimum-time problem than for the minimum-energy problem, could have been expected. Due to the larger degree of freedom of the minimum-time problem, ECOS typically requires more iterations to find a solution for a convex sub-problem. Meanwhile, in the algorithm implemented in this study, the discretisation step is identical for both the minimum-time problem and for the minimum-energy problem. It is thought that a performance improvement could be realised by eliminating the time-normalisation step (Section 4.3) for the minimum-energy problem.

8.7. Sensitivity analysis

The SCP algorithm that was developed in this study requires the definition of a large number of algorithm variables, which all influence the performance of the algorithm. For instance, in Section 8.5, it was analysed how the weight factors of the objective function could be tuned in order to realise performance improvements. Regarding the remaining parameters, most were selected through an iterative process of trial, error, and the analysis of the results of corresponding test cases. During this process, it was discovered that two algorithm parameters have a relatively large influence on the performance characteristics of the SCP algorithm, especially in terms of computational efficiency and accuracy. These two parameters are the number of discretisation nodes K and the convergence criterion for the state deviation ϵ_x. It was decided to analyse the sensitivity of the SCP algorithm regarding these two parameters. The purpose of this analysis is not to select the 'best' combination of parameters, as this would require detailed algorithm requirements for a specified mission scenario, but rather to identify general trends that could be of use when the performance of the SCP algorithm has to be optimised for a specific application. In this section, this sensitivity analysis is performed for the minimum-energy, attitude-unconstrained reorientation problem, but a similar analysis could be performed for different types of reorientation problems.

Figure 8.19 provides the results of this sensitivity analysis. For different combinations of the parameters K and ϵ_x, the mean performance of the SCP algorithm for a set of five Monte Carlo cases is shown. It must be noted that the convergence criterion ϵ_x was scaled for the number of grid nodes K, where the values provided in Figure 8.19 represent the parameters for $K = 15$. Furthermore, the NLP benchmark solution that forms the reference point of the optimality evaluation had a constant number of grid nodes $K_{\text{NLP}} = 15$ to realise a fair comparison in terms of optimality.

Several interesting observations can be made from Figure 8.19. To begin with, Figure 8.19a shows that the computational efficiency of the algorithm strongly depends on both parameters that were studied. The trend is as expected: the computation time increases as the number of discretisation nodes K (and therefore the



(a) Mean computation time (computational efficiency).

(b) Mean number of iterations required for convergence.

(c) Mean optimality of obtained solutions.

(d) Mean propagation error (accuracy).

Figure 8.19: Sensitivity of the performance of the SCP algorithm to the number of discretisation nodes K and the convergence criterion ϵ_x. For this analysis, the x- and y-axis were split up into 11 and 10 nodes, respectively. Every data point represents the mean value of five Monte Carlo test cases for the minimum-energy, attitude-unconstrained reorientation problem.

problem size) increases, and the convergence criterion ϵ_x is tighter (and it takes longer to converge). However, while the change in performance as a function of K is gradual, the change as a function of ϵ_x is stepwise: for values of ϵ_x smaller than 3, a sudden increase in the computation times can be observed. This stepwise behaviour is caused by the fact that from this point, the SCP algorithm typically required an additional iteration to convergence, as can be seen in Figure 8.19b. This behaviour is in line with expectations, as the convergence criterion ϵ_x influences the number of iterations that are performed before the algorithm converges. As the typical number of iterations required to converge is relatively low for the minimum-energy, attitude-unconstrained reorientation problem, a single additional iteration of the SCP algorithm is clearly visible in Figure 8.19a.

The same stepwise behaviour can also be identified for the results regarding the optimality (Figure 8.19c) and accuracy (Figure 8.19d) of the SCP algorithm. The reduction in accuracy for a stricter convergence criterion can be explained by the insight that such a stricter convergence criterion reduces the maximum difference between the converged solution and the last reference solution ($\Delta \boldsymbol{z}$) of the SCP algorithm. Consequently, the corresponding approximation of the original, non-convex problem dynamics becomes more accurate, and the accuracy of the SCP algorithm increases. It is noted that the accuracy (or propagation error) is expected to show quadratic behaviour as a function of the convergence criterion ϵ_x, as the approximation error of a first-order Taylor series expansion is a function of $(\Delta \boldsymbol{x})^2$. As a convex sub-problem that better approximates the original, non-convex problem also has an optimal solution that better approximates the solution of the original problem, it is no surprise that the general trends visible for sensitivities of Figure 8.19c and Figure 8.19d are highly similar. For the minimum-time problem, however, a larger sensitivity of the optimality with respect to the number of grid nodes K is expected, due to the less gradual nature of the optimal control

profiles. More grid nodes could enable the FOH-modelled control of the SCP algorithm to better approximate the control profiles of the *true* optimum.

Regarding the selection of parameters, an interesting finding is that the number of discretisation nodes K has a relatively small influence on both the accuracy and optimality of the SCP algorithm for the minimum-energy problem. The FOH discretisation method can deliver accurate results for low numbers of grid nodes K in combination with a strict convergence criterion ϵ_x. Considering the high computational efficiency of the SCP algorithm for a low number of grid nodes K, it could be interesting for onboard applications to consider such a combination of parameters. However, something that must be kept in mind is the enforcement of constraints. If control, attitude, and rate constraints would be exclusively enforced at the K grid nodes, reducing this parameter would lead to more extensive inter-nodal constraint violations.

When developing an SCP-based optimisation algorithm, it is suggested to perform a similar sensitivity analysis to get an initial 'feeling' of the sensitivity of the performance of the algorithm to these important algorithm parameters. The next step would be to perform an iterative cycle of Monte Carlo analyses while tweaking the algorithm parameters until satisfactory performance is obtained.

9

Monte Carlo Analyses

In this chapter, the results of the Monte Carlo analyses that were performed during this research project are presented. The aim of this chapter is to obtain a thorough understanding of the performance of the SCP-based optimisation algorithm in terms of the three main criteria identified in Section 3.7: robustness, optimality, and computational efficiency. In addition, characteristics regarding accuracy, the convergence process and constraint violations are provided and discussed.

A total of six Monte Carlo analyses were performed for different formulations of the spacecraft reorientation problem. The goal of this was to analyse the extent to which the type of reorientation problem influences the performance of the SCP algorithm. For both the minimum-time and minimum-energy spacecraft reorientation problems, three Monte Carlo analyses were executed for three different types of reorientation problems:

1. **Attitude-unconstrained** reorientation (Subsections 9.1.1 and 9.2.1)
   This problem formulation consists of all constraints and parameters presented in Section 4.10, except for the (conical) keep-out and keep-in attitude constraints. Even if these attitude constraints are imposed on a spacecraft, most of the reorientation manoeuvres are usually not hindered or obstructed by these constraints, as these restrict only a part of the (quaternion) attitude space. Therefore, the attitude-unconstrained reorientation problem serves as a useful indicator for the typical performance behaviour of the SCP algorithm, even when attitude constraints are enforced.

2. **Attitude keep-out constrained** reorientation (Subsections 9.1.2 and 9.2.2)
   For the second set of Monte Carlo analyses, attitude keep-out constraints were added to the problem formulation. These constraints are often included in both industry and literature (Section 3.8), and add a layer of complexity to the optimisation problem.

3. **Attitude-constrained** reorientation (Subsections 9.1.3 and 9.2.3)
   The final set of Monte Carlo analyses contains both attitude keep-out and keep-in constraints. This combination of these constraints further increases the complexity of the problem. For these analyses, the full problem formulations from Section 4.10 were solved.

In other words, the six different reorientation problems differ both in their objective (minimum-energy versus minimum-time) and in the extent to which attitude constraints are imposed on the problem. It should be noted that, in addition to the convex problem formulations from Section 4.10, the angular rate constraints were enforced on the midpoints of the 14 segments of the grid, in accordance with the method that was proposed in Section 8.3. Furthermore, each Monte Carlo analysis consisted of 2,500 test cases. Finally, in the following sections, the minimum-energy results are discussed with a higher level of detail than the minimum-time results, as similar trends could be identified for both problems, and it was tried not to repeat too much information for the sake of clarity.

In terms of structure, Sections 9.1 and 9.2 present the results and an in-depth discussion of the Monte Carlo analyses of, respectively, the minimum-energy and minimum-time reorientation problems. Then, in Section 9.3, the main findings of these analyses in terms of the three main performance criteria are collected,

compared to the results of the NLP benchmark algorithm, and used to answer the research questions that form the foundation of this study.

9.1. Minimum-energy reorientation problem

In this section, the performance of the SCP algorithm concerning the minimum-energy spacecraft reorientation problem is presented and analysed for three different Monte Carlo analyses. In Table 9.1, the SCP algorithm parameters are presented that were used for the Monte Carlo analyses presented in this section.

Table 9.1: SCP algorithm parameters for the minimum-energy reorientation problem studied in Section 9.1.

Parameter	Value	Unit	Parameter	Value	Unit
K	15	-	K_{disc}	5	-
w_u	1×10^{2}	-	$n_{\text{it,lim}}$	25	-
w_{tr}	1×10^{-3}	-	w_{vc}	1×10^{2}	-
$\alpha_{\text{tr,adap}}$	5	-	$\alpha_{\text{vc,adap}}$	3	-
$n_{\text{tr,adap,thres}}$	8	-	ϵ_{vc}	5×10^{-5}	-
ϵ_{x}	0.6	-			

9.1.1. Minimum-energy, attitude-unconstrained reorientation

In this subsection, the Monte Carlo results are provided for the minimum-energy, attitude-unconstrained spacecraft reorientation problem. In Figure 9.1, the convergence rate and behaviour of the SCP algorithm are visualised.

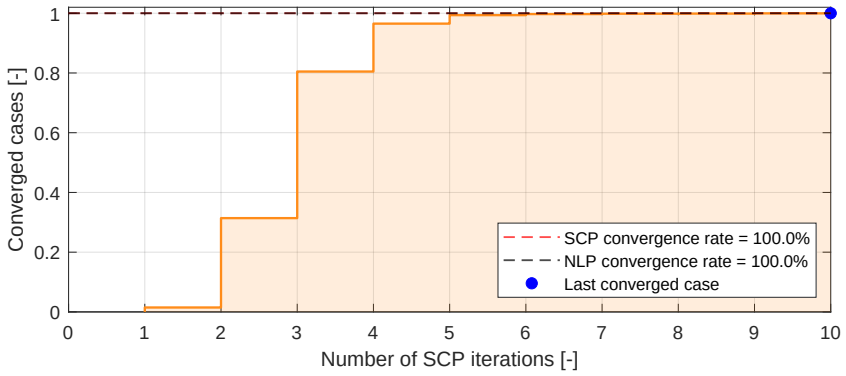


Figure 9.1: Empirical cumulative distribution function (CDF) of the number of SCP iterations required to obtain a converged solution for the minimum-energy, attitude-unconstrained reorientation problem. The Monte Carlo analysis consisted of 2,500 test cases.

Three interesting observations can be extracted from Figure 9.1:

- Both the SCP and NLP benchmark optimisation algorithms achieve a 100% convergence rate. This means that for all 2,500 Monte Carlo test cases, a feasible solution was found. This is a significant result from the perspective of a potential onboard implementation, as robustness is one of the key considerations for such applications (Section 3.7).
- The SCP algorithm converges rapidly, typically requiring only a few iterations. This was found to be related to the fact that the optimal, minimum-energy solution was usually approximated very well by the shape-based initial guess, as was discussed in Section 8.1. This explanation is supported by the fact that the SCP algorithm converged within a single iteration for some test cases.
- The SCP algorithm converged within a finite number of iterations (10) for all test cases. It is interesting to note that this number was significantly lower than the iteration limit $n_{\text{it,lim}}$ of 25 iterations. This finding suggests that there exists a predictable upper bound for the computational efficiency of the SCP algorithm, as the variance of this performance parameter is mainly driven by a varying number of iterations required for convergence.

In the remainder of this subsection (and chapter), the other performance criteria from Section 6.2 are analysed. The decision was made to use box plots for this purpose, as these, as outlined before in Chapter 6,

were found to provide the best overview of both the mean performance and the characteristics of outliers. In Figure 9.2, the legend is provided that applies to all box plots presented in the rest of this chapter.

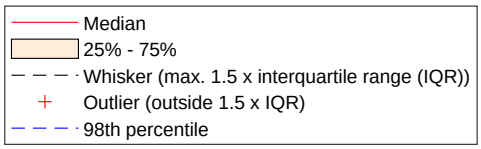


Figure 9.2: Legend for the box plots representing the performance of the SCP algorithm in the Monte Carlo analyses.

Regarding these box plots, the *interquartile range* (IQR) represents the difference between the 25th and 75th percentile (orange box in the figures). All data points outside of a range of 1.5 × IQR from the 25th and 75th percentile are considered to be outliers. Using this representation, in Figure 9.3, the performance of the SCP algorithm in terms of optimality, computational efficiency and accuracy is shown.

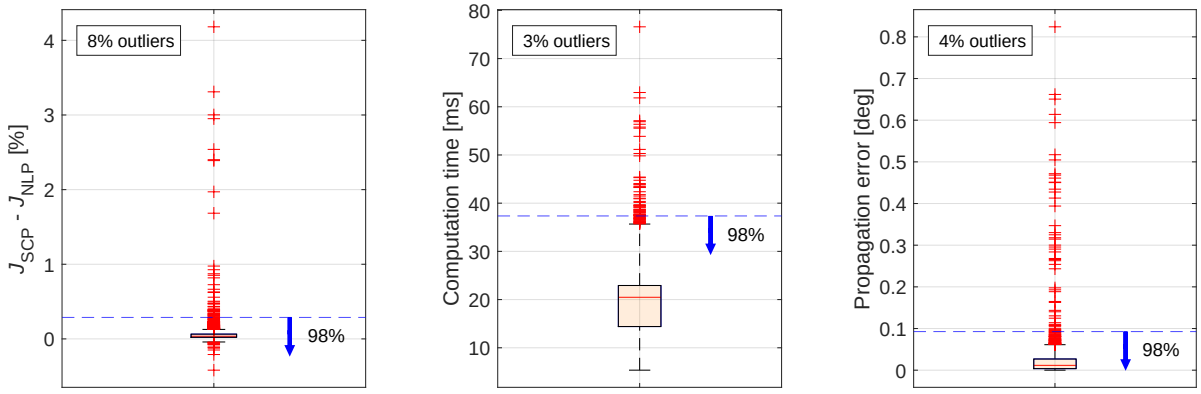


Figure 9.3: Performance of the SCP algorithm in terms of optimality, computational efficiency and accuracy for the minimum-energy, attitude-unconstrained spacecraft reorientation problem. The Monte Carlo analysis consisted of 2,500 test cases.

Then, in Figure 9.4, the results of Figure 9.3 are presented without the consideration of outliers to enable the reader to get a grasp of the median performance of the SCP algorithm for the majority of the Monte Carlo test cases, without being distracted by outliers.

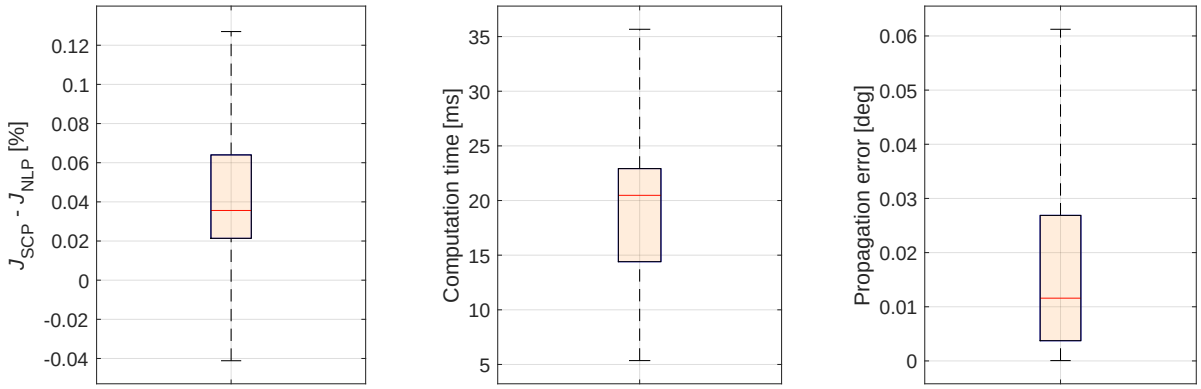


Figure 9.4: Detailed view of Figure 9.3, which shows the Monte Carlo results without consideration of the outliers.

Both Figures 9.3 and 9.4 are very insightful:

- In terms of optimality, the SCP algorithm performs excellent regarding our goal of finding minimum-energy guidance trajectories. For practically all cases, the difference in optimality between the SCP and NLP solutions is within 1%.

- In terms of computational efficiency, it can be observed that the SCP algorithm is fast. In literature, the NLP methods are often associated with computation times in the order of hundreds of milliseconds or even seconds. These results confirm the hypothesis that the SCP framework can deliver promising computation times. The computational efficiency of the SCP algorithm is further analysed and compared to the efficiency of the NLP benchmark algorithm in Subsection 9.3.3.

- In terms of accuracy, the mean propagation error was slightly higher than 0.01 degrees, which seems to be accurate considering other studies on reorientation guidance (f.i. the study of Ventura et al. (2015)). However, as was discussed in Section 3.7, a final conclusion can only be drawn when explicit accuracy requirements are formulated for a certain mission scenario. As covered in Section 8.7, when defining the parameters of the SCP algorithm, specifically the state convergence criterion ϵ_x, one basically performs a trade-off between computational efficiency and accuracy. This process could be complicated by the behaviour that can be observed in Figure 9.3. The reason for this is that the maximum propagation error is almost three orders of magnitude larger than the median propagation error. In order to improve the accuracy of these outliers, a computational burden would be placed on many problems for which the results could already be accurate enough.

9.1.2. Minimum-energy, attitude keep-out constrained reorientation

In Figure 9.5, the convergence characteristics of the SCP algorithm with respect to the keep-out constrained problem are shown. As noted earlier, this problem is known to be more complex than the attitude-unconstrained reorientation problem.

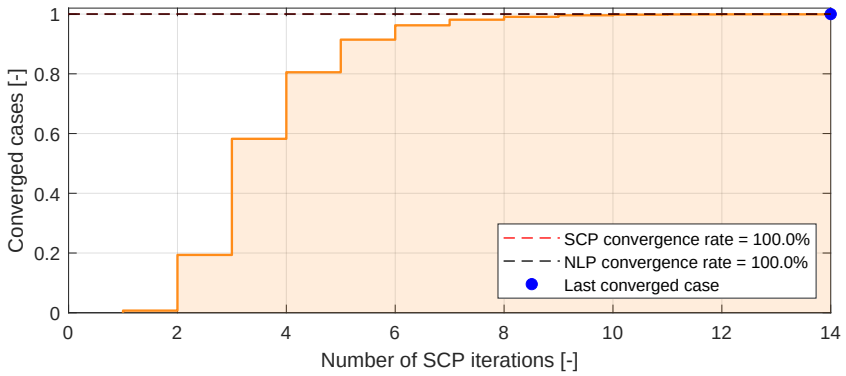


Figure 9.5: Empirical cumulative distribution function of the number of SCP iterations required to obtain a converged solution for the minimum-energy, keep-out constrained spacecraft reorientation problem. The Monte Carlo analysis consisted of 2,500 test cases.

Two main insights were obtained from Figure 9.5:

- Similar as for the attitude-unconstrained problem presented in Figure 9.1, both the SCP and NLP benchmark algorithms show a 100% convergence rate. From this observation, it can be concluded that the conical keep-out constraint in itself has no negative consequences on the robustness of the SCP algorithm for the minimum-energy problem.

- In contrast to the attitude-unconstrained problem from Figure 9.1, the SCP algorithm needs a maximum of not 10, but 14 iterations to find a solution that meets all convergence criteria. This is in line with the general notion that the keep-out constrained problem is more challenging to solve than the attitude-unconstrained problem. Nevertheless, the majority of test cases can be seen to have converged after around 10 iterations.

In order to analyse the performance of the SCP algorithm regarding the other criteria of Section 3.7, in Figure 9.6, the optimality, computational efficiency, and accuracy results of the Monte Carlo analysis are provided.

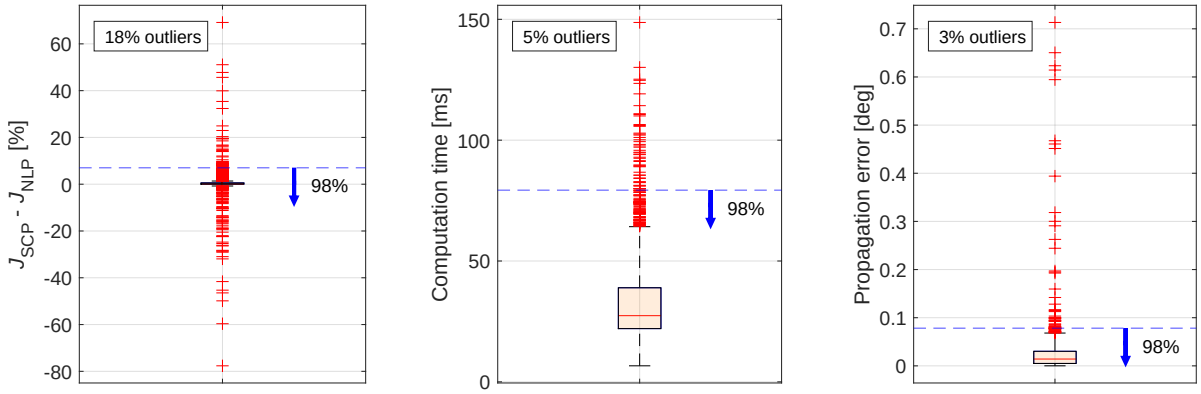


Figure 9.6: Performance of the SCP algorithm in terms of optimality, computational efficiency and accuracy for the minimum-energy, keep-out constrained spacecraft reorientation problem. The Monte Carlo analysis consisted of 2,500 test cases.

Again, Figure 9.7 shows a detailed view of the results from Figure 9.6 which does not consider outliers.

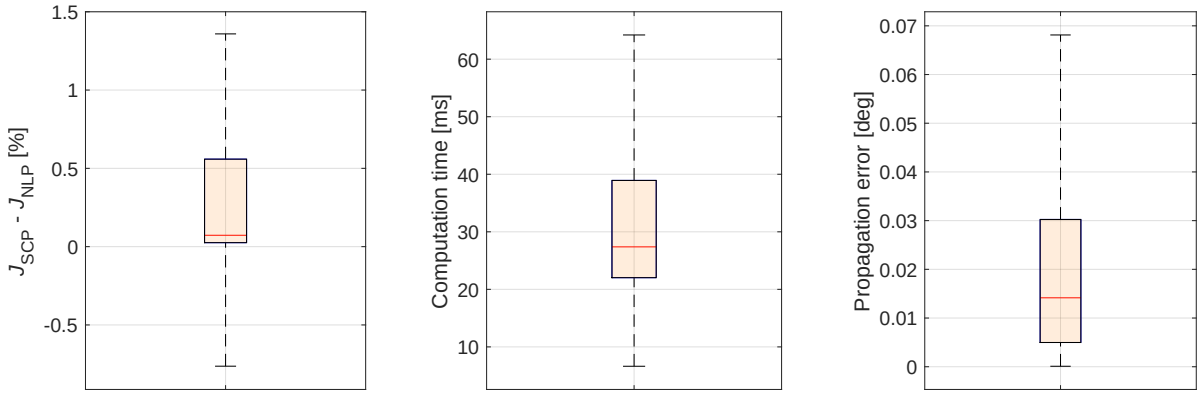


Figure 9.7: Detailed view of Figure 9.6, which shows the Monte Carlo results without consideration of the outliers.

Based on Figure 9.6 and Figure 9.7, it can be concluded that:

- In terms of optimality, although the majority of SCP solutions show a performance difference to the NLP benchmark within a few percent, large outliers can be observed. After inspecting the results that correspond to these outliers, it was concluded that these differences in optimality mainly result from the fact that both the SCP and NLP algorithms are local optimisation methods. It was observed that, because the problem formulation specifies that the keep-out constraints are located directly between the initial and final attitude of the manoeuvre, the SCP and NLP solutions often took a path around the 'opposite' sides of the conical keep-out constraint. An example of this phenomenon is provided in Figure 9.8. In this case, these different trajectories lead to a difference in performance of 22.9%. However, neither the NLP nor the SCP algorithm was found to consistently find solutions that were more optimal.

- The computational efficiency decreases due to the introduction of the attitude keep-out constraint compared to the unconstrained case. It was found that this was caused by both the increased number of iterations that the SCP algorithm required to converge, and by an increase in the ECOS solve times due to the increased problem size.

- As expected, as both the discretisation procedure and convergence tolerances for the keep-out constrained and attitude-unconstrained problems were identical, the propagation errors of the SCP guidance solutions have approximately the same magnitude as for the attitude-unconstrained problem. Therefore, it can be concluded that tuning the accuracy of the SCP algorithm is relatively independent of the type of minimum-energy reorientation problem.

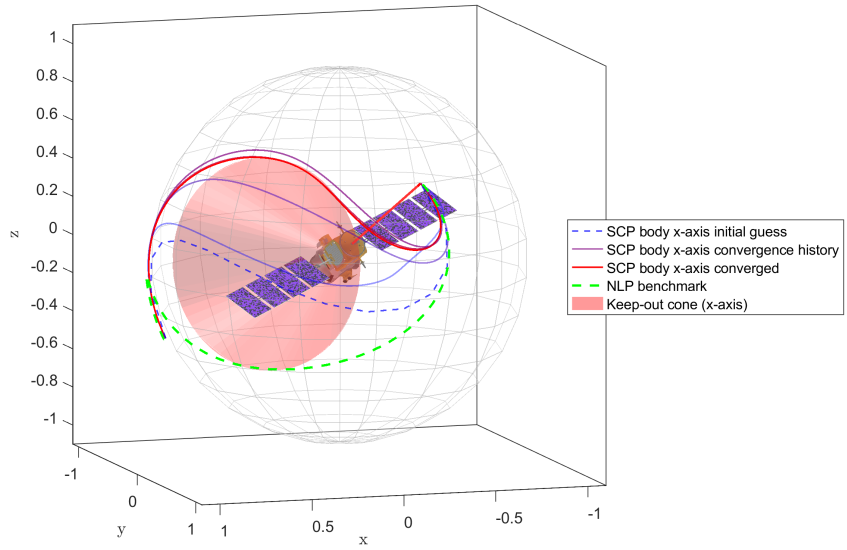


Figure 9.8: 3-D visualisation of an example, minimum-energy, keep-out constrained test case where the optimality difference $\Delta J = 22.9\%$ (the SCP solution is more expensive than the NLP solution). The NLP and SCP solutions pass along a different 'side' of the keep-out constraint.

Because the attitude keep-out constraint is only enforced at the grid nodes, it was determined to be interesting to analyse the maximum constraint violations that were encountered during the test cases of this Monte Carlo analysis. In Figure 9.9, these violations are shown.

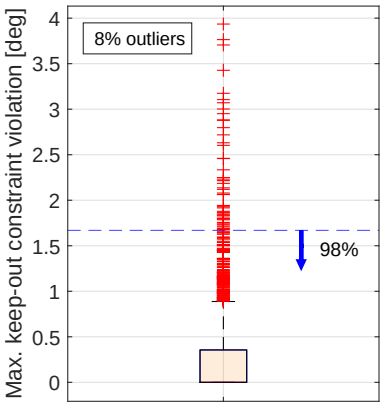


Figure 9.9: Attitude keep-out constraint violations for the minimum-energy, keep-out constrained spacecraft reorientation problem. The Monte Carlo analysis consisted of 2,500 test cases.

For most cases, it can be observed that the maximum constraint violation is within 2 degrees and relatively small. However, outliers were found of up to 4 degrees. In order to provide additional insight, in Figure 9.11 a 3-D visualisation is provided for the SCP guidance solution for the test case that results in the highest keep-out constraint violation in Figure 9.9, which is about 4 degrees.

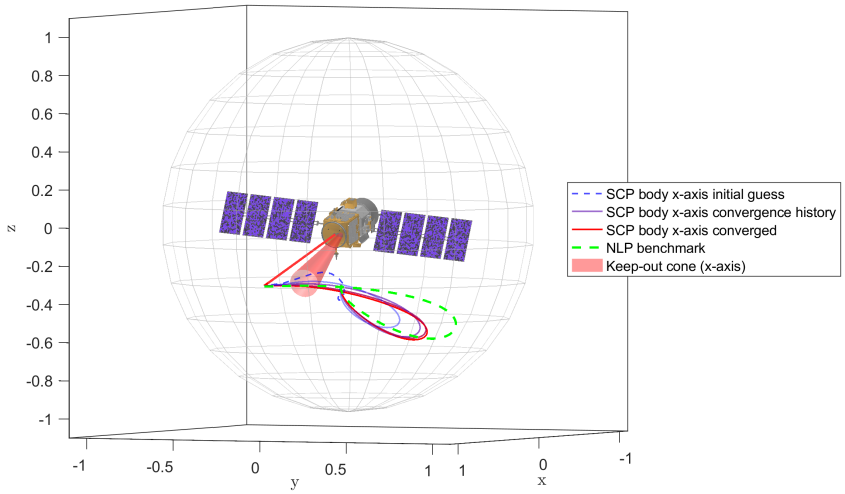


Figure 9.10: 3-D visualisation of the minimum-energy, keep-out constrained test case that represents the maximum keep-out constraint violation (4 degrees) of Figure 9.9.

In Figure 9.11, the corresponding angular separation angle between the x-axis of the body-fixed reference frame and the central axis of the keep-out constraint cone is shown.

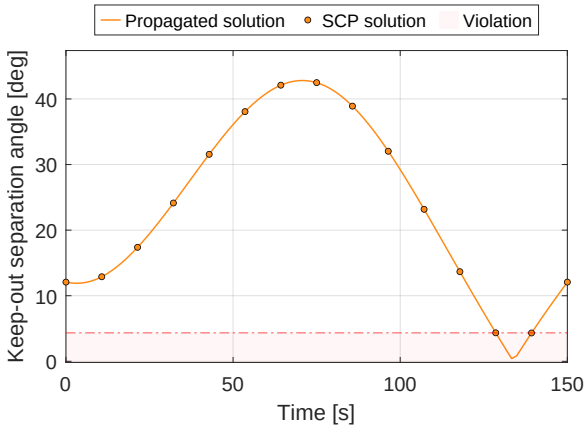


Figure 9.11: Angular separation between the body-fixed x-axis of the spacecraft and the vector representing the central axis of the keep-out constraint for the minimum-energy, keep-out constrained test case of Figure 9.11.

In this case, the relatively large constraint violation is a consequence of the fact that the attitude keep-out cone is very small. This allows both the SCP and NLP algorithms to find a solution that 'skips' the constraint altogether, using inter-nodal constraint violations. This case was in line with the broader observation that the keep-out constraint violations increased as the minimum separation angle of the corresponding keep-out cones were smaller.

In practice, it would be very unlikely that such a small keep-out constraint would be enforced on an attitude reorientation problem. If this would be an actual mission scenario, there could be two main ways to deal with these constraint violations. The first is to increase the angular separation angle of the keep-out constraint with a 5-degree safety margin. This would suffice to guarantee that, for the majority of cases, the keep-out attitude constraint is satisfied at all times. An alternative approach would be to use the novel method introduced in Section 8.3 to enforce the conical keep-out constraint at a set of additional nodes that do not belong to the grid of the discretised optimisation variables. This would, however, as discussed in Section 8.3, lead to an increase in the computation time required for finding a guidance solution.

9.1.3. Minimum-energy, attitude-constrained reorientation

In this subsection, the results are provided of the Monte Carlo analysis of the minimum-energy, attitude-constrained reorientation problem. Figure 9.12 shows the convergence results of this Monte Carlo analysis.

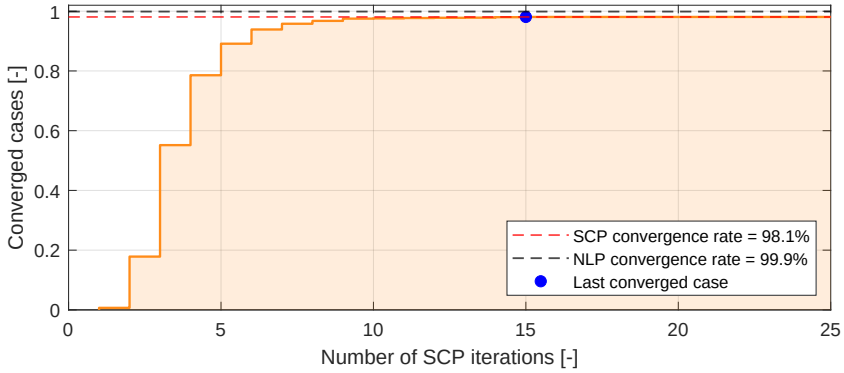


Figure 9.12: Empirical cumulative distribution function of the number of SCP iterations required to obtain a converged solution for the minimum-energy, attitude-constrained spacecraft reorientation problem. The Monte Carlo analysis consisted of 2,500 test cases.

Two important findings can be extracted from Figure 9.12:

- To begin with, it can be seen that at first sight, the SCP algorithm seems to have had difficulties to converge successfully in comparison to the previous two Monte Carlo analyses, as is reflected by the convergence rate being only 98.1%. Meanwhile, the convergence rate of the NLP benchmark algorithm is 99.9%, which supposedly indicates that there exist feasible solutions to the problems where the SCP algorithm failed to converge. However, it was found that this conclusion cannot be drawn that easily, as will be extensively discussed after the next observation.

- After a certain number of iterations, in this case 15, it was found that the SCP algorithm either had converged or would not converge at all. During the analysis, increasing the iteration limit of the SCP algorithm did not change this behaviour. Therefore, it can be concluded that there is no point in increasing the iteration limit significantly above 15-20, at least for this Monte Carlo test set-up. Furthermore, it was found that inconvergence typically could be detected early, as ECOS would provide the status 'Unbounded' already from iterations 5-7 for the majority of cases where it failed to converge. This property could enable an SCP algorithm to abort early when it cannot find a solution for an optimisation problem, something which is not always possible for NLP optimisation methods. These are known to run for extended amounts of time before reporting that no solution can be found.

The following paragraphs aim to provide an in-depth analysis of the fact that Figure 9.12 shows a lower convergence rate for the SCP algorithm than for the NLP benchmark algorithm.

After a thorough analysis of a large number of individual test cases for which the SCP algorithm failed to converge, the hypothesis was formed that there were test cases present in the set of Monte Carlo cases that would be infeasible if all attitude constraints would be enforced continuously, in contrast to only at the nodes along the attitude manoeuvre. In other words, it was thought that there are cases that are theoretically infeasible but for which the SCP and NLP algorithms could still find (discrete) solutions through inter-nodal constraint violations. In order to prove this hypothesis, the same Monte Carlo analysis, with identical test cases, was solved with a third algorithm. This algorithm is identical to the NLP benchmark algorithm that was used up to this point, except for the fact that for this algorithm, the attitude constraints were enforced at 30 (linearly-spaced) nodes, instead of the 15 CGL grid points. The results of this Monte Carlo analysis regarding the convergence rate of the three algorithms can be seen in Figure 9.13. In this figure, the third algorithm is referred to with $\text{NLP}_{\text{feasible}}$, as this algorithm is considered to be a better indicator for the test cases that are feasible.

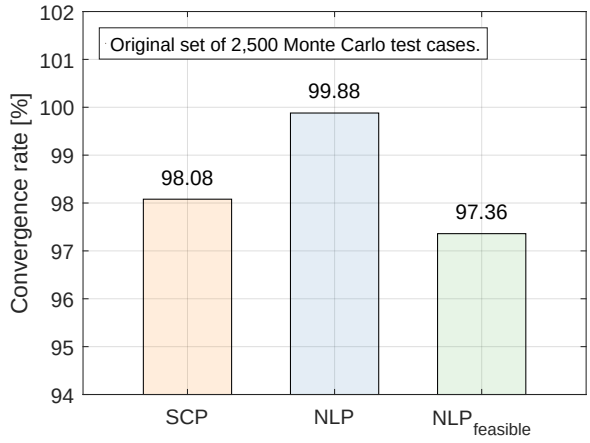


Figure 9.13: Convergence results of the Monte Carlo analysis for the minimum-energy, attitude-constrained reorientation problem, including the results of a second NLP algorithm. This algorithm enforces the attitude constraints at a larger number of nodes.

As can be seen in Figure 9.13, the $\text{NLP}_{\text{feasible}}$ algorithm only converges for 97.36% of all 2,500 test cases. Given the 99.88% convergence rate for the original NLP algorithm (which enforces the attitude constraints at a smaller number of nodes), it becomes very likely that the NLP and SCP methods indeed achieve a higher convergence rate than 97.36% due to inter-nodal keep-out and keep-in constraint violations. This theory was confirmed through analysing a large number of individual test cases, of which one is provided in Figure 9.14.

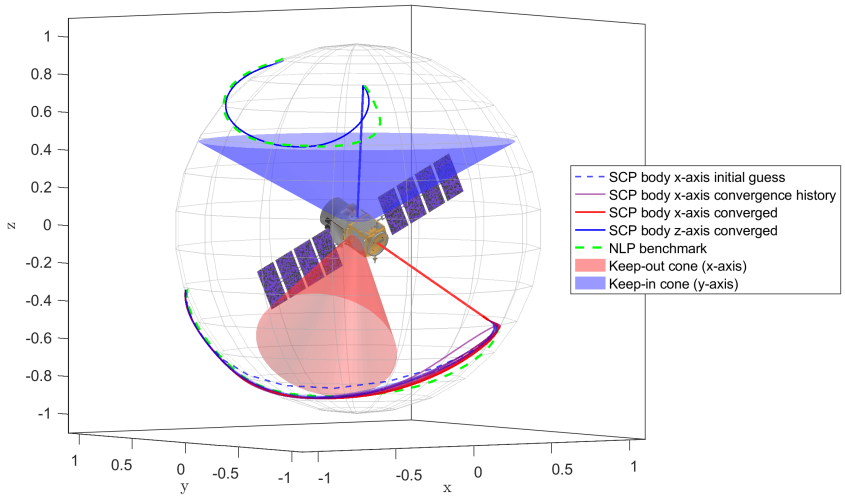
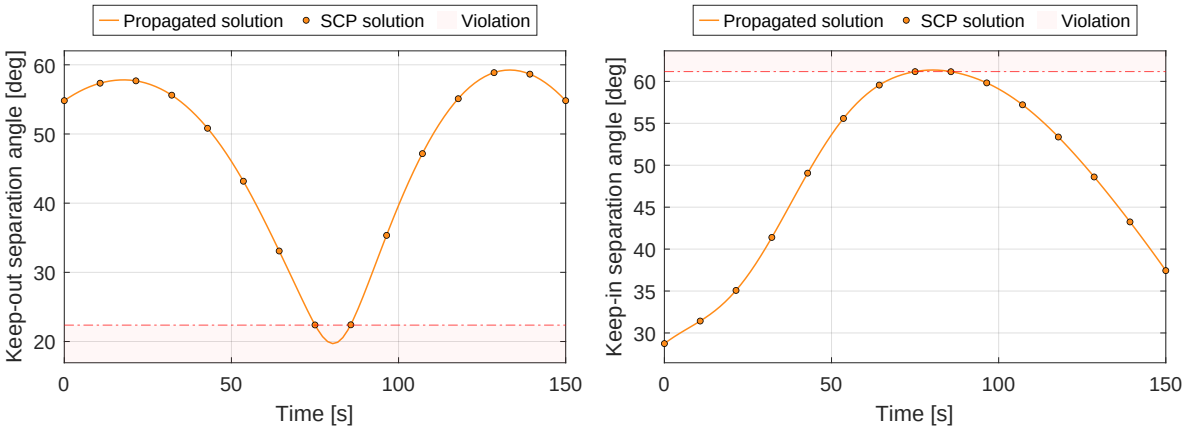


Figure 9.14: 3-D visualisation of a minimum-energy, attitude-constrained test case that does converge for the SCP and NLP algorithms, but does not converge for the $\text{NLP}_{\text{feasible}}$ algorithm.

For the test case shown in Figure 9.14, both the SCP and NLP algorithms were able to find a solution, while the $\text{NLP}_{\text{feasible}}$ algorithm failed to converge. It can be seen that the guidance trajectories of both the SCP and NLP methods touch the boundaries of both attitude constraints, indicating that this test case could be infeasible if the attitude constraints were enforced continuously. To provide further evidence for this hypothesis, Figure 9.15 shows the propagated angular separation angles between the central axes of the keep-out and keep-in constraints and, respectively, the body-fixed x- and z-axes of the spacecraft for the test case presented in Figure 9.14.



(a) Angular separation between the body-fixed x-axis of the spacecraft and the central axis of the keep-out constraint.

(b) Angular separation between the body-fixed z-axis of the spacecraft and the central axis of the keep-in constraint.

Figure 9.15: Attitude constraint satisfaction of the SCP solution and propagated time histories corresponding to the minimum-energy, attitude-constrained test case of Figure 9.14.

It Figure 9.15, it becomes clear that at exactly the same point in the manoeuvre, the keep-out constraint and, to a lesser extent, the keep-in constraint are violated, providing more evidence for the hypothesis that this case would be infeasible if the attitude constraints would be enforced continuously, instead of only at 15 nodes. Combining the results of Figure 9.13 with the results of the single-case analysis from Figure 9.15, the hypothesis that the Monte Carlo analysis contained infeasible test cases was considered proven.

At this point, the question remains how the performance of the SCP algorithm in terms of robustness can be evaluated if the results from Figure 9.12 contain infeasible test cases. The decision was made that it would be more fair to evaluate the convergence rate of the SCP algorithm within the 97.36% of the original 2,500 Monte Carlo cases for which the $\text{NLP}_{\text{feasible}}$ algorithm converged, as these test cases are thought to be more or less feasible, although small constraint violations could still be present in the obtained solutions. Using this approach, the robustness of the SCP algorithm would be defined as its capability to converge for feasible reorientation problems, and not as its capability to use constraint violations to find solutions for problems that are infeasible. The results of this analysis for the reduced set of Monte Carlo test cases are provided in Figure 9.16.

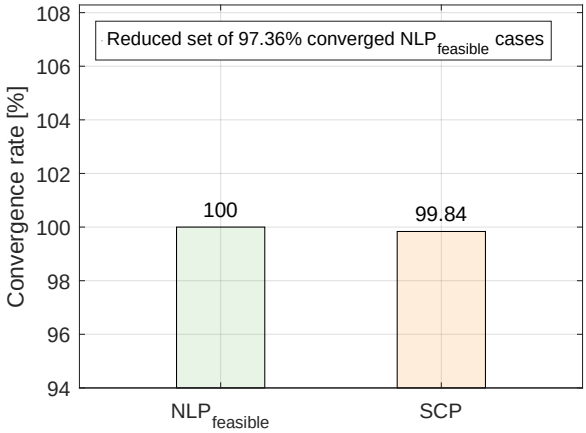


Figure 9.16: Convergence results of the Monte Carlo analysis for the minimum-energy, attitude-constrained reorientation problem. This time, only the 97.36% of the cases were considered for which the $\text{NLP}_{\text{feasible}}$ algorithm converged to a solution.

From Figure 9.16, it can be observed that the convergence rate of the SCP algorithm for this set of test cases was 99.84%, which is significantly higher than the 98.08% from Figure 9.12. The important conclusion that can be drawn from this analysis is that the robustness of the SCP algorithm for the minimum-energy, attitude-constrained reorientation problem appears to be highly comparable to the NLP benchmark algorithm. However, an exact comparison cannot be drawn as the taken approach does not consider test cases that *are* fea-

sible, but where the $\text{NLP}_{\text{feasible}}$ algorithm failed to find a solution. Nevertheless, for the purpose of this study, it was determined that the obtained convergence rate of 99.84% sufficed to draw conclusions about the robustness of the SCP algorithm at a level that is required for answering the research question.

After inspection, the three cases (0.16%) for which the SCP algorithm did not and the $\text{NLP}_{\text{feasible}}$ algorithm did converge were found to be feasible. One of these three test cases is visualised in Figure 9.17.

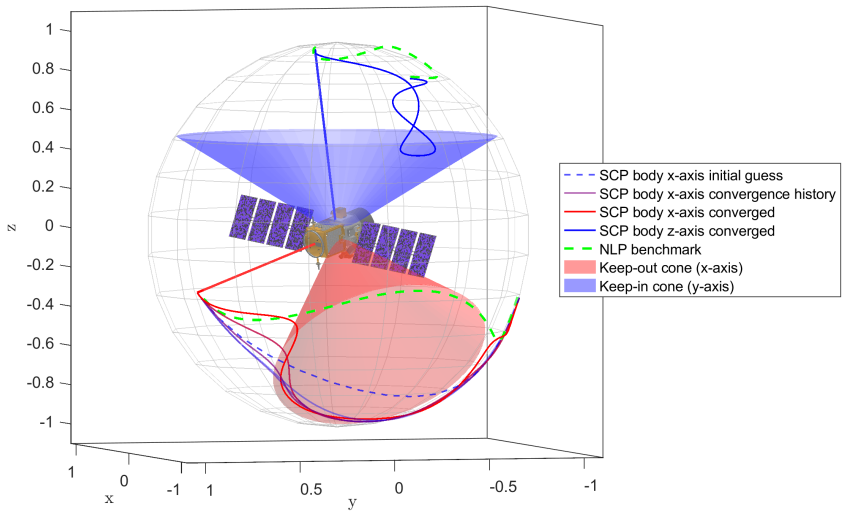


Figure 9.17: 3-D visualisation of one of the three minimum-energy, attitude-constrained test cases for which the SCP algorithm failed to converge while the $\text{NLP}_{\text{feasible}}$ algorithm did converge.

In Figure 9.17, it can be seen that the SCP algorithm tried to find a solution at the side of the keep-out cone where it (visually) seems impossible to find a solution that satisfies both attitude constraints. Meanwhile, the NLP solution, which takes a path along the other side of the constraint, is obviously able to satisfy all attitude constraints. This behaviour was detected for all three cases for which the SCP algorithm failed to converge. In these test cases, as described in Subsection 7.2.4, the SCP algorithm uses virtual control to violate the conical constraints for early iterations. However, because of the conflicting attitude constraints, the SCP algorithm is unable to reduce this use of virtual control in later iterations. This is illustrated in Figure 9.18, which presents the convergence criteria satisfaction and virtual control use of the SCP algorithm.

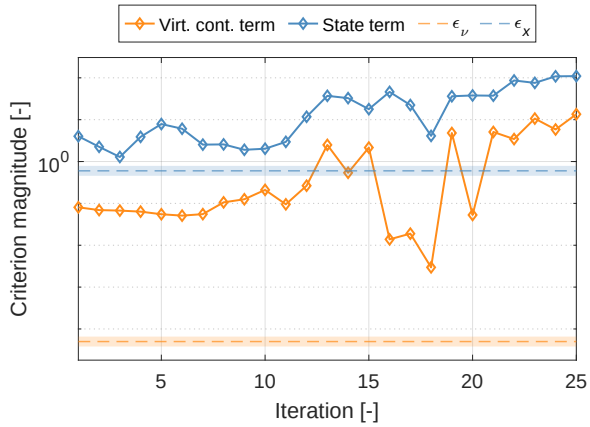


Figure 9.18: History of the convergence criteria corresponding to the minimum-energy, attitude-constrained tests case from Figure 9.17. The SCP algorithm was halted after 4 iterations.

Figure 9.18 shows that the SCP algorithm resorts to using virtual control at a level much higher than allowed for convergence, and apparently does not have the capacity to 'refocus' and try to find a solution on the other side of the keep-out constraint. Considering the difference between the typical convergence behaviour of the SCP and NLP algorithms, this might have been expected. The SCP algorithm either converges within 10-15

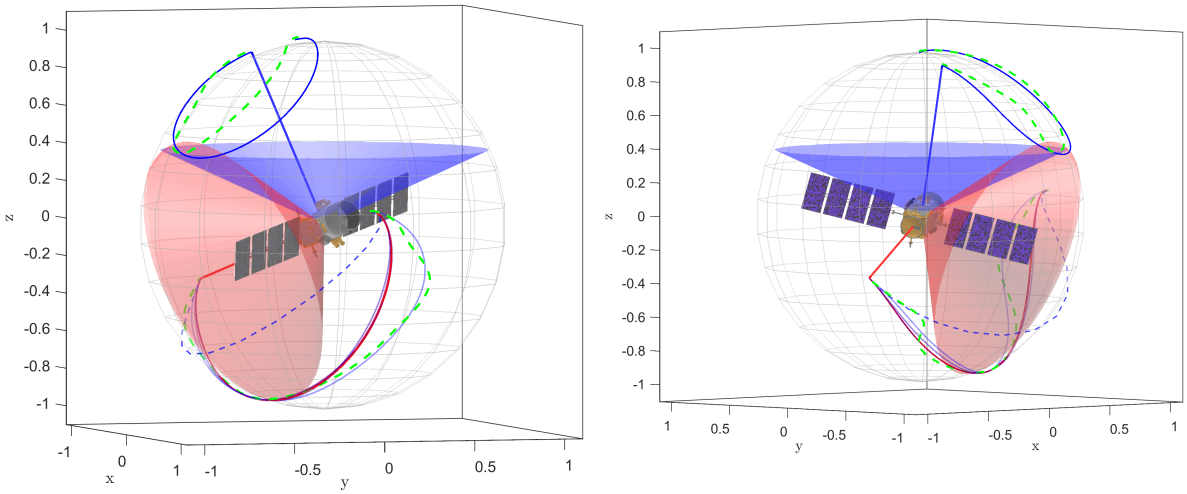
iterations, or does not converge at all. In contrast, the NLP algorithm sometimes requires up to 1000 iterations to find a solution. Furthermore, the NLP algorithm uses advanced logic, including multiple 'restoration phases', in which the algorithm 'zooms out' in order to consider a search direction that is significantly different from the one it was focused on during its recent iterations. This logic is not included in the SCP algorithm. As a result, it is highly likely that the SCP algorithm cannot recover from a first guess or early iteration that leads it to focus on a side of the keep-out constraint that eventually turns out to be infeasible. However, it was found that the SCP algorithm could converge for these three test cases when a different initial guess would be provided. Therefore, it is thought that the obtained convergence rate of 99.84% could be increased through the development of initialisation-dependent logic for the SCP algorithm. For instance, if the SCP algorithm would still use virtual control after 5 iterations, it could be considered to re-initialise with an alternative initial guess. However, due to the scope of this research project, this topic is left as a recommendation for future research.

The final thing that needs to be clarified is that the NLP algorithm was able to converge for more cases (99.88%) than the SCP algorithm (98.08%) in the original set of 2,500 Monte Carlo test cases, as shown in Figure 9.13. Three causes for this phenomenon were identified:

- The SCP algorithm uses an equally spaced grid, while the NLP algorithm uses a non-equally spaced grid (see Subsection 5.2.2). As the keep-out and keep-in constraints are exclusively enforced at the grid nodes, this implies that for the NLP algorithm, several inter-nodal grid distances are slightly larger, allowing for larger constraint violations. This was found to enable the NLP algorithm to find a solution for some of the test cases were the SCP algorithm could not.

- Through analysing the test cases for which the NLP algorithm did and the SCP algorithm did not converge, it was found that the NLP algorithm was often more 'creative' and persistent (using up to 1000 iterations and computation times of multiple seconds) in terms of finding complex, constraint-violating solutions, while the SCP algorithm simply resorted to using virtual control and did not converge. An example of such a case is provided in Figure 9.19. In this test case, the SCP algorithm continued to use virtual control and did not converge, while the NLP solution (for the x-axis) can be seen to take a non-straightforward route. The NLP solution was found to accelerate substantially near the attitude constraint, effectively enlarging the distance that could be travelled between two subsequent nodes of the grid. Using this strategy, larger constraint violations could be achieved and used to find a solution to the infeasible case. The occurrence of this phenomenon is supported by the keep-out constraint violations of the NLP algorithm for all cases where the NLP algorithm *did* and the SCP algorithm *did not* converge, which are shown in Figure 9.20. The inter-nodal keep-out constraint violations for this set of test cases can be seen to be significantly larger than for the cases where the SCP algorithm did converge (Figure 9.23).

- The SCP and NLP algorithms apply a different strategy to model the control. While the SCP algorithm uses a first-order-hold assumption, the NLP algorithm uses Lagrange polynomials. It is thought that the use of Lagrange polynomials allows for more flexibility to achieve the large angular accelerations that the NLP algorithm commonly applies to find a solution for an infeasible test case. In addition, the NLP algorithm could allow for (small) inter-nodal violations of the control constraint.



(a) Perspective focused on the start of the manoeuvre.

(b) Perspective focused on the end of the manoeuvre.

Figure 9.19: 3-D visualisation of a test case for the minimum-energy, attitude constrained reorientation problem for which the NLP algorithm did and the SCP algorithm did not converge, as seen from two perspectives.

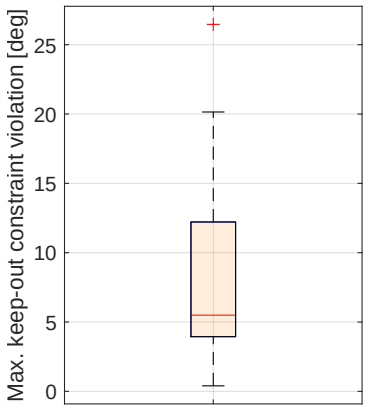


Figure 9.20: Violations of the attitude-keep out constraint of the NLP solutions for the minimum-energy, attitude-constrained reorientation problem. The results only represent cases for which the NLP algorithm did converge and the SCP algorithm did not.

In Figure 9.21, the Monte Carlo results regarding optimality, computational efficiency and accuracy are provided.

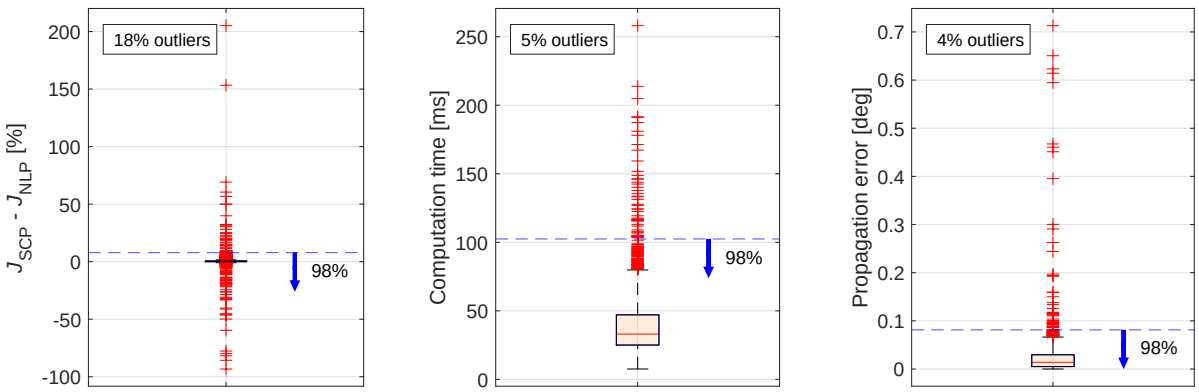


Figure 9.21: Performance of the SCP algorithm in terms of optimality, computational efficiency and accuracy for the minimum-energy, attitude-constrained spacecraft reorientation problem. These results represent all cases of the original set of 2,500 Monte Carlo cases for which both the SCP and NLP algorithms converged.

Figure 9.22 shows the results from Figure 9.21 without the consideration of outliers.

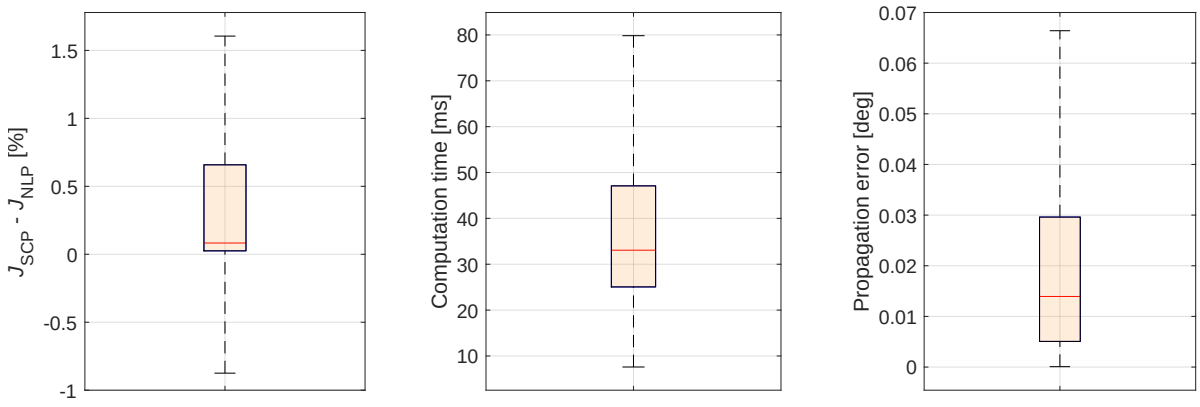


Figure 9.22: Detailed view of Figure 9.21, which shows the Monte Carlo results without consideration of the outliers.

On the basis of Figures 9.21 and 9.22, it was observed that:

- Regarding the optimality of the obtained guidance solutions, the performance is relatively similar to the performance for the keep-out constrained problem of Subsection 9.1.2, except for two outliers around +150% and +200%,
- In terms of computational efficiency and accuracy, the differences with respect to the earlier Monte Carlo analyses are in line with expectations, as the computation times increase due to the additional (keep-in) constraint and the accuracy does not notably change.

Finally, Figure 9.23 shows the constraint violations that were encountered during the Monte Carlo analysis.

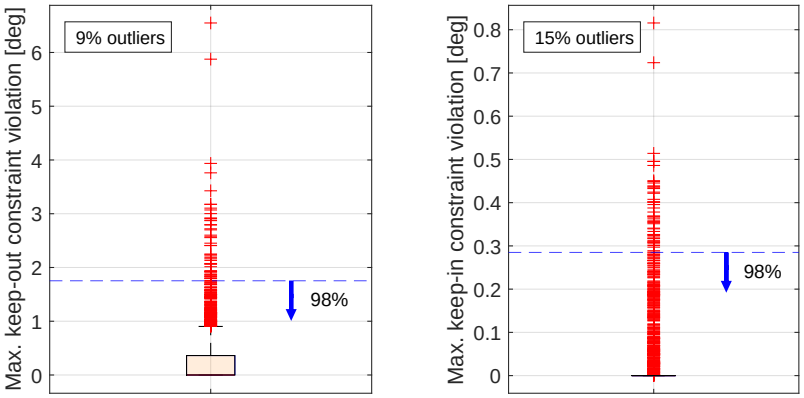


Figure 9.23: Attitude constraint violations for the minimum-energy, attitude-constrained spacecraft reorientation problem. These results represent all cases of the original set of 2,500 Monte Carlo cases for which both the SCP and NLP algorithms converged.

Interestingly, it can be observed in Figure 9.23 that constraint violations appear to pose a smaller problem for the keep-in attitude constraint than for the keep-out constraint. Therefore, if all violations would be unacceptable, it is suggested to simply impose a safety margin on this constraint instead of imposing additional constraints using the strategy from Section 8.3. The resulting penalty in optimality that results would likely be preferable over the penalty in computational efficiency that would result from additional constraints. It should be noted, though, that this margin would also increase the number of infeasible cases.

9.2. Minimum-time reorientation problem

In this section, the results of the three Monte Carlo analyses are presented that were performed to analyse the performance of the SCP algorithm regarding the same three problem types as were studied in the previous

section, but this time with a minimum-time objective instead of a minimum-energy objective. For the purpose of conciseness, the discussion of the results is somewhat more brief in this section than in Section 9.1. The reason for this is that most of the interesting observations have already been identified and discussed extensively in Section 9.1, and it was desired to avoid repetition. In Table 9.2, the SCP algorithm parameters are provided that were used for the Monte Carlo analyses presented in this section. As stated before, the 14 additional rate constraints from Section 8.3 were included in the problem formulation as well.

Table 9.2: SCP algorithm parameters for the minimum-time reorientation problem studied in Section 9.2.

Parameter	Value	Unit	Parameter	Value	Unit
K	15	-	K_{disc}	5	-
w_η	1	-	δ_η	30	%
w_{tr}	1×10^{-3}	-	w_{vc}	50	-
$\alpha_{\text{tr,adap}}$	5	-	$\alpha_{\text{vc,adap}}$	3	-
$n_{\text{tr,adap,thres}}$	8	-	ϵ_η	5×10^{-2}	-
ϵ_{x}	0.9	-	ϵ_{vc}	5×10^{-5}	-
$n_{\text{it,lim}}$	25	-			

9.2.1. Minimum-time, attitude-unconstrained reorientation

In this subsection, the results of the Monte Carlo analysis of the minimum-time, attitude-unconstrained spacecraft reorientation problem are provided. To begin with, in Figure 9.24, the convergence rate is presented as a function of the number of performed SCP iterations.

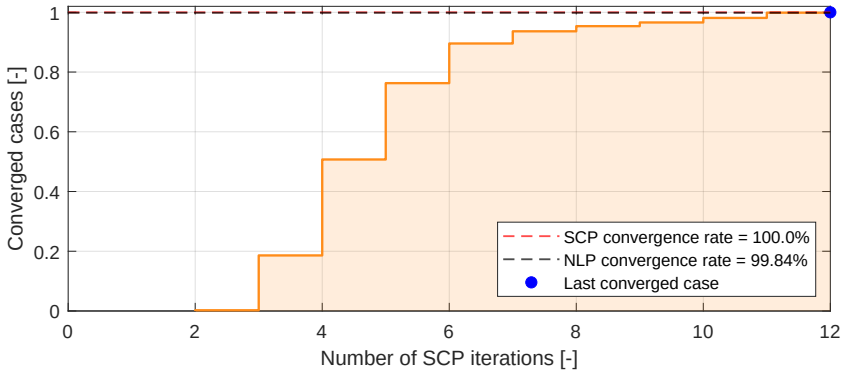


Figure 9.24: Empirical cumulative distribution function of the number of SCP iterations required to obtain a converged solution for the minimum-time, attitude-unconstrained spacecraft reorientation problem. The Monte Carlo analysis consisted of 2,500 test cases.

Interesting findings that can be extracted from these results are:

- In contrast to the minimum-energy, attitude-unconstrained problem, it can be seen that the NLP benchmark algorithm failed to find a solution for 4 of the 2,500 Monte Carlo test cases. This finding is line in with the experience of the research partner OHB, which has also encountered seemingly random NLP failures for a very small number of small-angle rotations. The 4 NLP failures from Figure 9.24 also considered small-angle rotations. Because of the fact that the focus of this study was not to optimise the NLP benchmark algorithm but rather to analyse the potential of an SCP-based optimisation method, it was decided that a detailed study of the causes of the inconvergence of the NLP algorithm was out of scope and left as a possible topic for further research.

- As for the results of the minimum-energy analyses, it was found that after a relatively low number of 12 SCP iterations, the SCP algorithm had converged for all test cases. This strengthens the idea that the SCP algorithm converges in a predictable range of computation times.

In Figure 9.25, the performance of the SCP algorithm in terms of optimality, computational efficiency and accuracy is visualised.

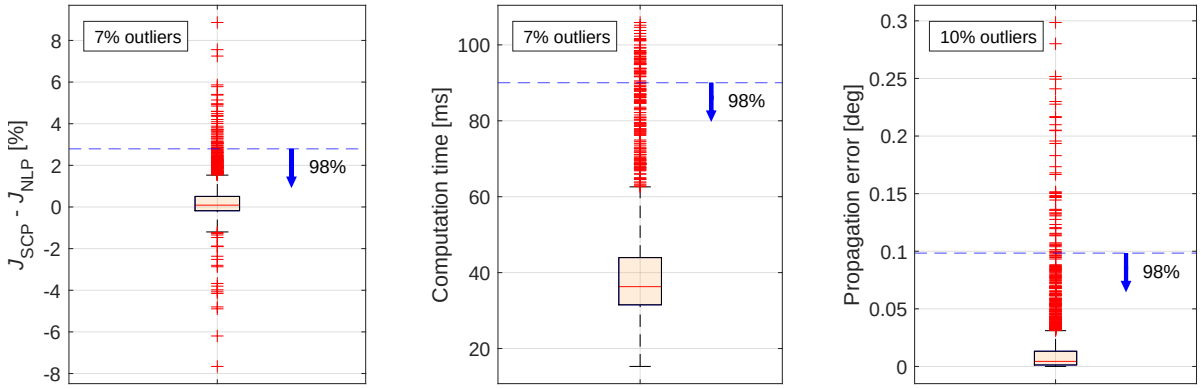


Figure 9.25: Performance of the SCP algorithm in terms of optimality, computational efficiency and accuracy, for the minimum-time, attitude-unconstrained spacecraft reorientation problem. These results represent all cases of the original set of 2,500 Monte Carlo cases for which both the SCP and NLP algorithms converged.

In Figure 9.26, the results of Figure 9.25 are shown without consideration of the outliers in order to better visualise the median performance of the SCP algorithm.

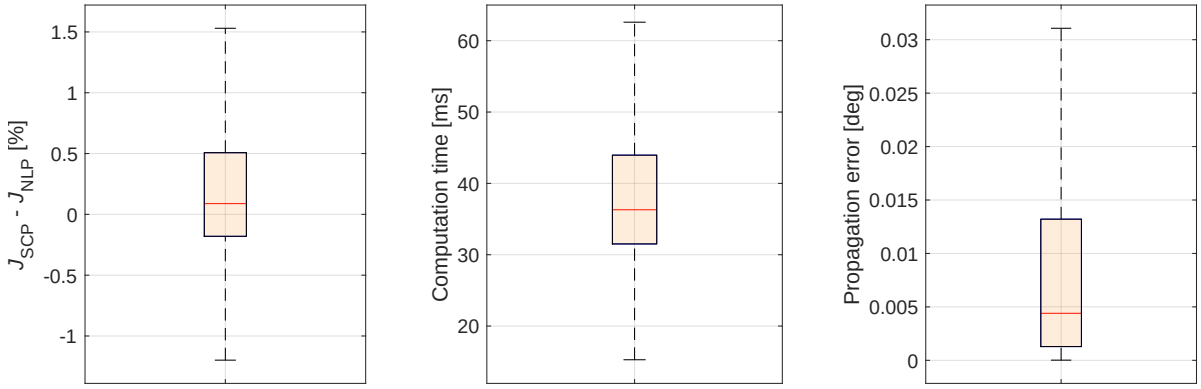


Figure 9.26: Detailed view of Figure 9.25, which shows the Monte Carlo results without consideration of the outliers. These results represent all cases of the original set of 2,500 Monte Carlo cases for which both the SCP and NLP algorithms converged.

The two observations that were made on the basis of these figures are:

- In terms of optimality, the performance of the SCP and NLP benchmark algorithms is comparable for the majority of the test cases. However, it can also be observed that the majority of the outliers represent cases where the NLP algorithm found a more optimal solution than the SCP algorithm. After a close inspection of these individual test cases, a potential cause was identified. It is thought that the Lagrange modelling of the control of the NLP algorithm allows for more flexibility regarding the search for an optimal control profile, compared to the FOH-modelled control profiles of the SCP algorithm. For instance, for some cases it was observed that the NLP solution showed a faster acceleration than the SCP solution at the beginning of the manoeuvre. This could be seen to lead to a shorter (in terms of time), more optimal reorientation manoeuvre. Regarding future research, it would be interesting to investigate how a non-equally spaced grid with more grid nodes at the start and end of the manoeuvre could potentially improve the optimality of the SCP algorithm.

- Regarding computational efficiency, the results indicate that it typically takes more time to solve a minimum-time problem than a minimum-energy problem, as is expected due to the larger degree of freedom of the minimum-time problem.

Then, in Figure 9.27, the violations of the box constraint on the angular rate are shown. It was decided to show these violations for the minimum-time Monte Carlo analyses, as the angular rate constraint has a major

influence on the extent to which the final time t_f can be minimised and, therefore, is often closely approached by the optimal solution.

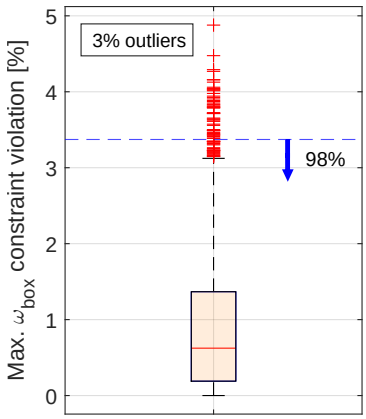


Figure 9.27: Violations of the box constraint on the angular rate for the minimum-time, attitude-unconstrained spacecraft reorientation problem. These results represent all cases of the original set of 2,500 Monte Carlo cases for which both the SCP and NLP algorithms converged.

In Figure 9.27, it can be seen that the violations of the box constraint on the angular rate are in line with the remaining violations that were reported and discussed in Section 8.3.

9.2.2. Minimum-time, attitude keep-out constrained reorientation

In this subsection, the results of the Monte Carlo analysis for the minimum-time, keep-out constrained spacecraft reorientation problem are provided. To begin with, in Figure 9.28, the convergence rate is presented as a function of the number of performed SCP iterations.

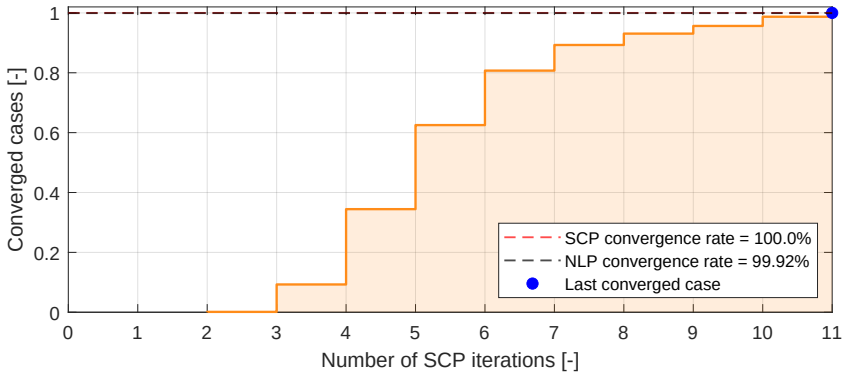


Figure 9.28: Empirical cumulative distribution function of the number of SCP iterations required to obtain a converged solution for the minimum-time, keep-out constrained spacecraft reorientation problem. These results represent all cases of the original set of 2,500 Monte Carlo cases for which both the SCP and NLP algorithms converged.

From this figure, the same conclusion can be drawn as for the minimum-energy, keep-out constrained problem: adding conical keep-out constraints in itself has no negative influences on the convergence rate and thus the robustness of the SCP algorithm.

In Figure 9.29, the performance of the SCP algorithm in terms of optimality, computational efficiency and accuracy is visualised.

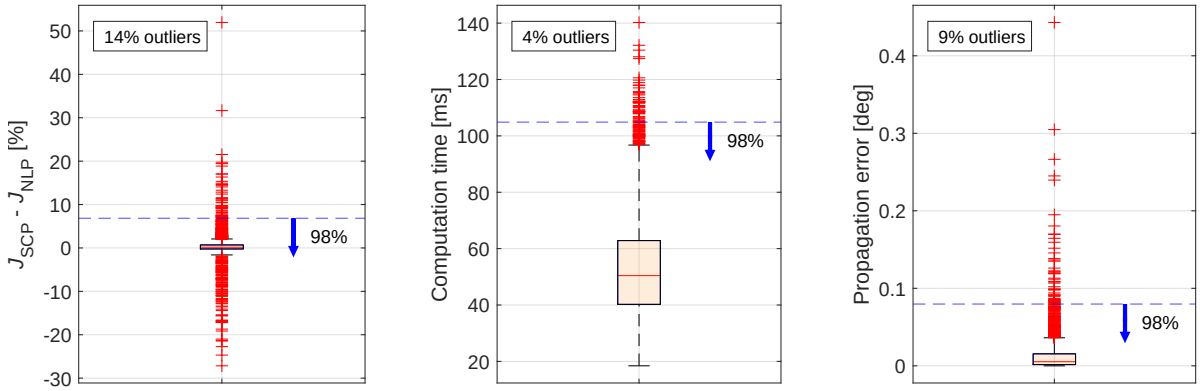


Figure 9.29: Performance of the SCP algorithm in terms of optimality, computational efficiency and accuracy for the minimum-time, keep-out constrained spacecraft reorientation problem. These results represent all cases of the original set of 2,500 Monte Carlo cases for which both the SCP and NLP algorithms converged.

In Figure 9.30, the results of Figure 9.29 are shown without consideration of the outliers to get a better understanding of the mean performance of the SCP algorithm for this type of reorientation problem.

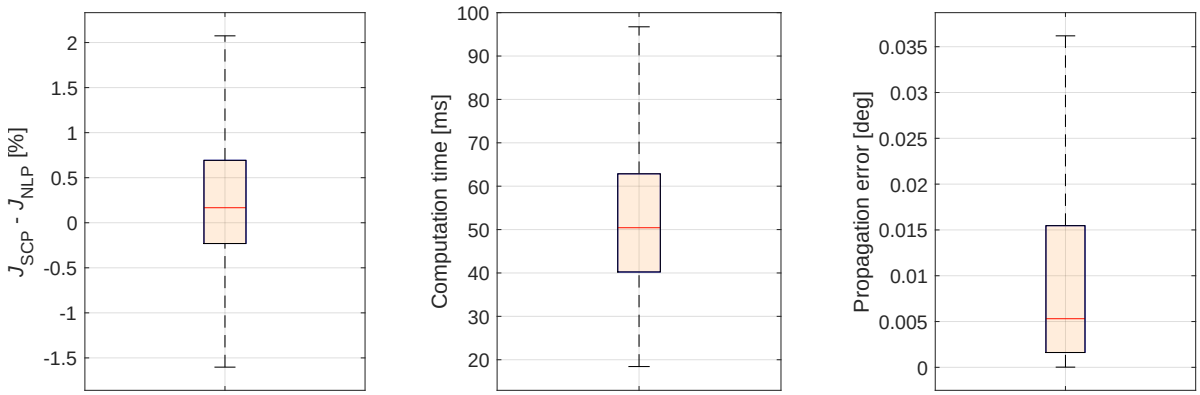


Figure 9.30: Detailed view of Figure 9.29, which shows the Monte Carlo results without considering the outliers.

As for the convergence rate, no new insights can be obtained from Figures 9.29 and 9.30 as they show the same trends that were already identified for the minimum-energy problem. To begin with, the computation times increase relative to the attitude-unconstrained problem due to the enforcement of the additional keep-out constraint. In addition, the number and spread of outliers increases because the NLP and SCP algorithms, for some of the Monte Carlo test cases, found solutions that moved the x-axis of the spacecraft on opposite sides of the conical keep-out constraint.

Then, in Figure 9.31, the violations of the conical attitude keep-out constraint and box constraint on the angular rate are provided.

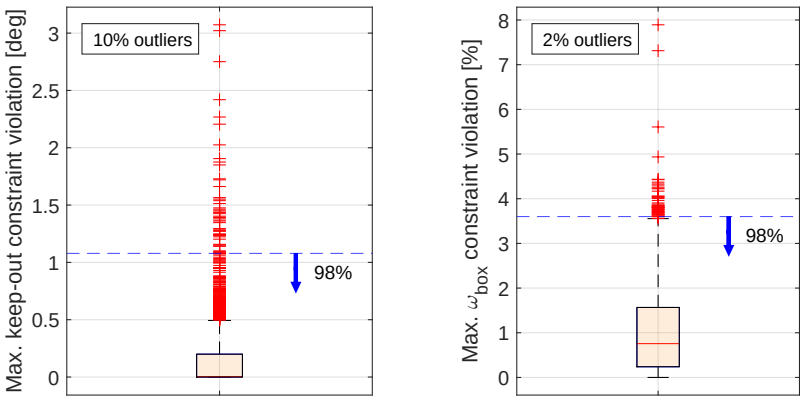


Figure 9.31: Attitude keep-out and angular rate box constraint violations of the minimum-time, keep-out constrained spacecraft reorientation problem. These results represent all cases of the original set of 2,500 Monte Carlo cases for which both the SCP and NLP algorithms converged.

As can be seen, the keep-out constraint violations are comparable in terms of magnitude to the constraint violations that occurred for the minimum-energy analyses. However, regarding the violations of the box constraint on the angular rate, two outliers can be identified of around 8%, which supports the relevance of the research recommendation from Section 8.3 to study the effects of imposing additional inter-nodal rate constraints on the spacecraft reorientation problem.

9.2.3. Minimum-time, attitude-constrained reorientation

This subsection presents the results of the Monte Carlo analysis of the minimum-time, attitude-constrained spacecraft reorientation problem. To begin with, in Figure 9.32, the convergence rate is presented as a function of the number of SCP iterations that have been performed.

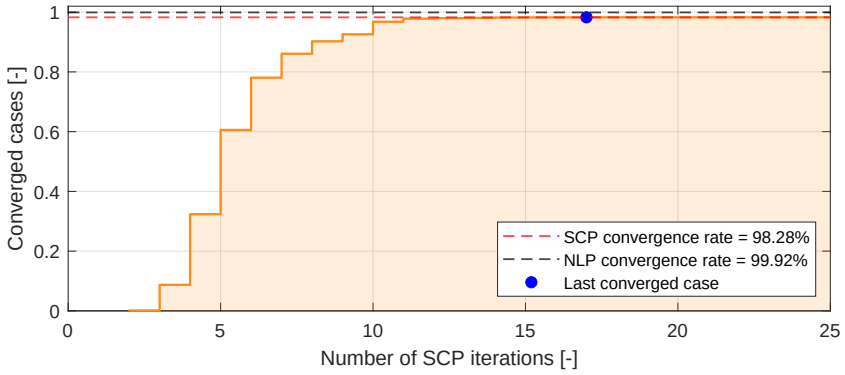
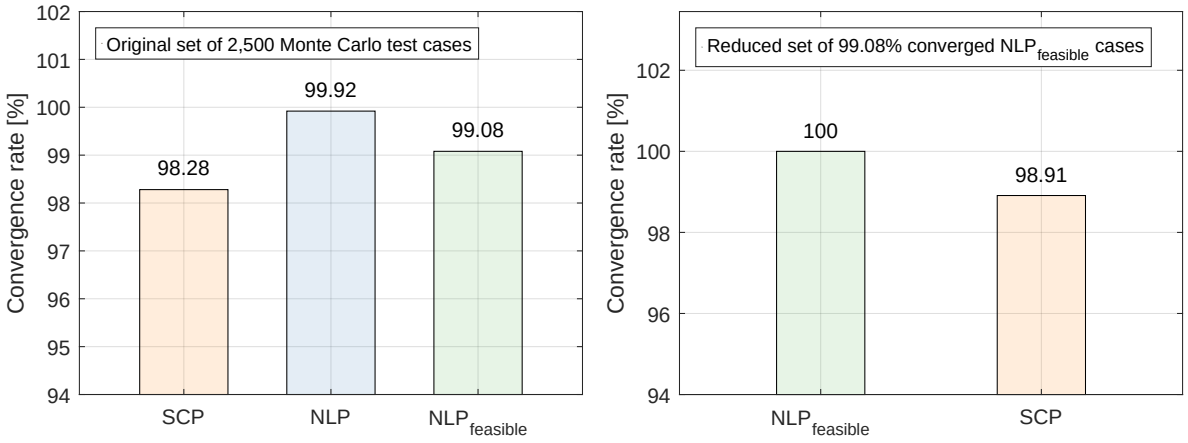


Figure 9.32: Empirical cumulative distribution function of the number of SCP iterations required to obtain a converged solution for the minimum-time, attitude-constrained spacecraft reorientation problem. The Monte Carlo analysis consisted of 2,500 test cases.

As for the minimum-energy, attitude-constrained problem, it can be observed that the SCP algorithm fails to converge for a significant number of Monte Carlo test cases. However, as was extensively discussed in Subsection 9.1.3, the Monte Carlo analysis was found to contain test cases that are infeasible if all constraints would be enforced continuously. Therefore, all test cases were solved a third time with the $\text{NLP}_{\text{feasible}}$ algorithm that enforced that attitude constraints at 30 instead of 15 nodes, following the same strategy as for the minimum-energy problem. The convergence results of this analysis are shown in Figure 9.33.



(a) Convergence results for the full Monte Carlo set of 2,500 test cases.

(b) Convergence results for the reduced set of 99.08% test cases for which the $NLP_{feasible}$ algorithm converged.

Figure 9.33: Convergence results of the Monte Carlo analysis for the minimum-time, attitude-constrained reorientation problem, including the results of a second NLP algorithm which enforces the attitude constraints at a larger number of nodes.

Interestingly, in contrast to the results of the minimum-energy case of Subsection 9.1.3, the $NLP_{feasible}$ algorithm shows a higher convergence rate for the full set of Monte Carlo test cases than the SCP algorithm, as can be observed in Figure 9.33b. This is also reflected in Figure 9.33, where it can be observed that the 98.91% convergence rate of the SCP algorithm in the reduced set of MC test cases is nowhere close to the 99.84% that was obtained for the minimum-energy problem.

Based on an in-depth analysis of the 1.19% of the test cases for which the $NLP_{feasible}$ algorithm did and the SCP algorithm did not converge (Figure 9.33b), it was found that all SCP failures could be attributed to the SCP algorithm trying the find a solution at the 'wrong' side of the conical keep-out constraint, a phenomenon that was also analysed in Subsection 9.1.3. It is not fully clear why this phenomenon occurs more often for the minimum-time problem than for the minimum-energy problem, but it could be caused by the fact that minimum-time solutions are typically located closer to the keep-out constraint than minimum-energy solutions. However, as for the minimum-energy problem, it was found that an alternative initialisation could result in the SCP algorithm successfully converging and finding the optimal solution. Therefore, the same recommendation is applicable to the attitude-constrained minimum-time problem as for the minimum-energy problem: introducing initialisation-based logic to the SCP algorithm is expected to be a promising option to realise robustness improvements for the attitude-constrained reorientation problem.

For this minimum-time problem, it was seen that the NLP algorithm used an even more 'creative' strategy to find solutions to infeasible test cases than for the minimum-energy problems. An example of this strategy is provided in Figure 9.34.

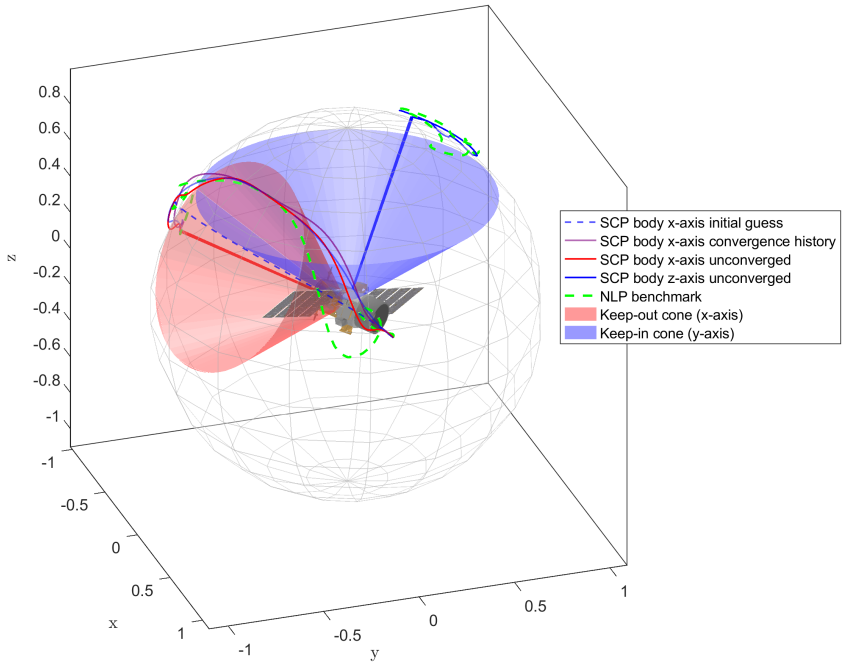


Figure 9.34: 3-D visualisation of a minimum-time, attitude-constrained test case for which the NLP algorithm did and the SCP algorithm did not converge. Only 3 iterations are shown of the SCP algorithm.

It was discovered that the spacecraft slows down at the beginning of the manoeuvre of Figure 9.34, increasing the total manoeuvre time. Then, it rapidly speeds up and 'skips' the keep-out constraint. Through the increased manoeuvre duration and constant number of grid points where the attitude constraints are enforced, the NLP algorithm could realise large constraint violations and find a solution. In contrast, the SCP algorithm would simply resort to using virtual control and fail to converge.

In Figure 9.35, the performance of the SCP algorithm in terms of optimality, computational efficiency and accuracy is visualised.

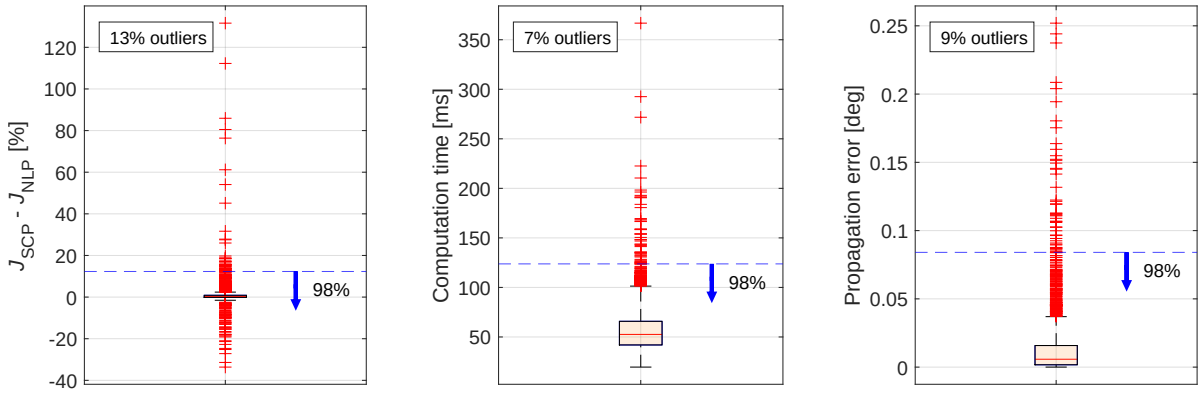


Figure 9.35: Performance of the SCP algorithm in terms of optimality, computational efficiency and accuracy, for the minimum-time, attitude-constrained spacecraft reorientation problem. These results represent all cases of the original set of 2,500 Monte Carlo cases for which both the SCP and NLP algorithms converged.

In Figure 9.36, the results of Figure 9.35 are shown without consideration of the outliers to better visualise the performance of the SCP algorithm for the majority of the test cases.

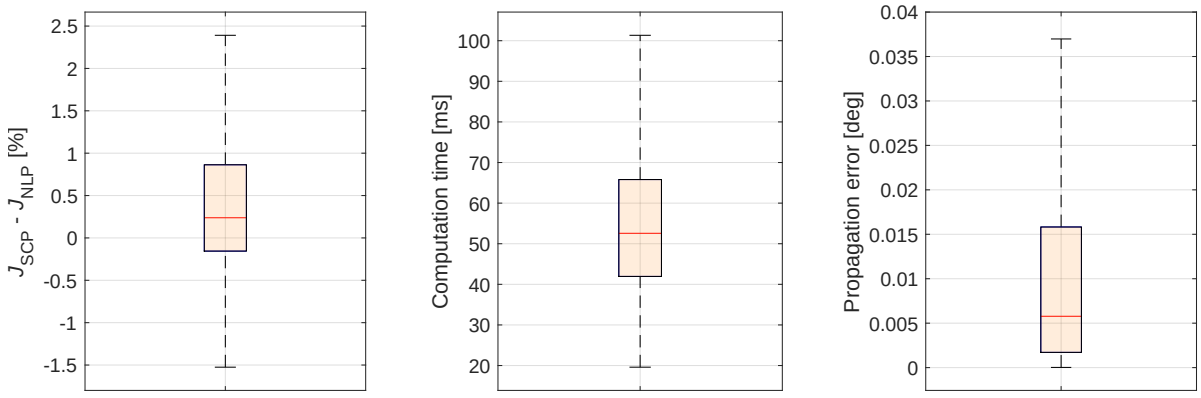


Figure 9.36: Detailed view of Figure 9.29, which shows the results of the Monte Carlo analysis without consideration of the outliers.

All results shown in Figures 9.35 and 9.36 are in line with the observations that were made regarding the five other Monte Carlo analyses presented in this chapter.

Finally, in Figure 9.37, the violations of the conical attitude keep-in constraint, keep-out constraint, and the box constraint on the angular rate are provided.

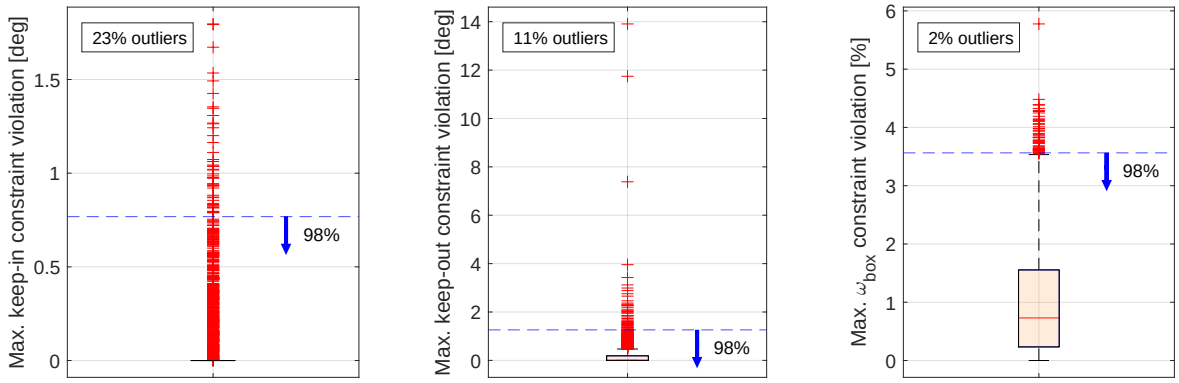


Figure 9.37: Attitude keep-in, attitude keep-out and angular rate box constraint violations of the minimum-time, attitude-constrained spacecraft reorientation problem. These results represent all cases of the original set of 2,500 Monte Carlo cases for which both the SCP and NLP algorithms converged.

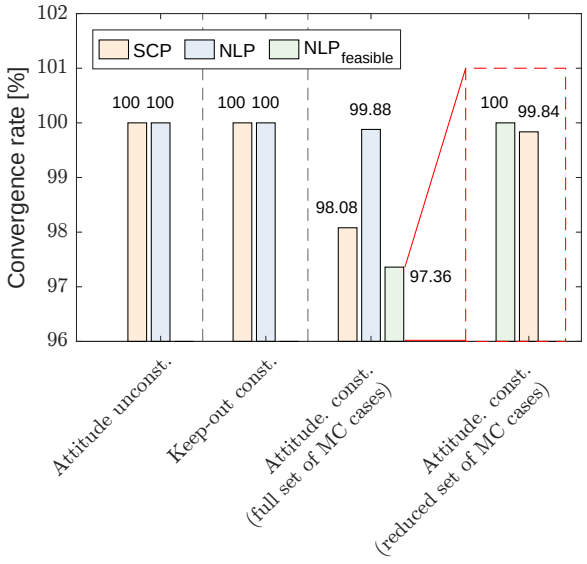
The three surprisingly large outliers for the keep-out constraint violation were found to be related to cases for which the $\text{NLP}_{\text{feasible}}$ algorithm did not converge. Therefore, these cases represent infeasible reorientation problems.

9.3. Overall performance assessment

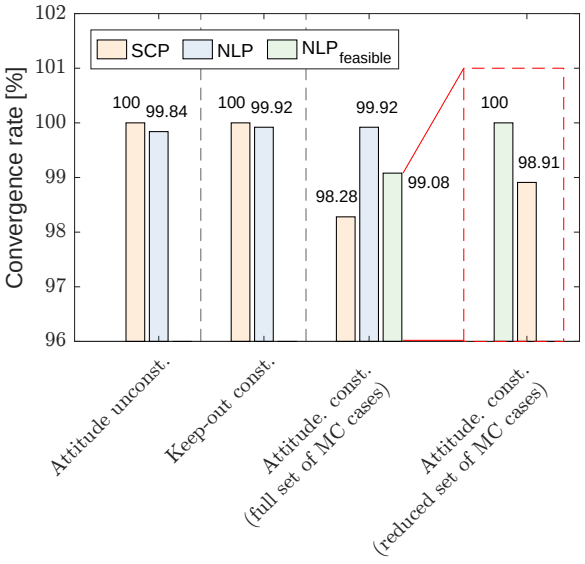
In this section, the main results and findings from all six Monte Carlo analyses that were presented in the previous two sections of this chapter are collected to draw top-level conclusions regarding the performance of the SCP algorithm. These conclusions are drawn with respect to the three main algorithm requirements that were identified in Section 3.7: robustness, optimality and computational efficiency, which are discussed in, respectively, Subsections 9.3.1, 9.3.2 and 9.3.3.

9.3.1. Robustness

Figure 9.38 presents the convergence rates of the SCP and NLP algorithms that were obtained for the six Monte Carlo analyses.



(a) Convergence rates of the three Monte Carlo analyses of the minimum-energy reorientation problem.



(b) Convergence rates of the three Monte Carlo analyses of the minimum-time reorientation problem.

Figure 9.38: Convergence rates of the SCP and NLP algorithms for the six Monte Carlo analyses. For the attitude-constrained Monte Carlo analyses, the SCP convergence rates for the reduced sets of Monte Carlo test cases is shown as well, in line with the discussions of Subsections 9.1.3 and 9.2.3.

Regarding the attitude-unconstrained and attitude keep-out constrained reorientation problems, the conclusion is evident: the SCP algorithm shows excellent performance in terms of robustness, as it converged for 100% of all 2,500 Monte Carlo test cases.

For the minimum-energy and minimum-time, attitude-constrained reorientation problems, the conclusion is less straightforward. As was extensively discussed in Subsection 9.1.3, it was found that the original set of 2,500 Monte Carlo cases contained infeasible test cases, for some of which the NLP and, to a lesser extent, the SCP algorithms were able to find solutions using inter-nodal constraint violations. Therefore, it was decided to base conclusions regarding robustness on a smaller selection of the original 2,500 test cases, for which a third algorithm ($NLP_{feasible}$) was able to converge to a solution. This third algorithm enforced the attitude constraints on 30 instead of 15 nodes, and was therefore considered to be an indication of a set of test cases that are actually feasible. The results of the SCP algorithm in terms of robustness for this smaller set of test cases are highlighted in Figure 9.38 with red boxes.

For this reduced set of test cases, the SCP algorithm performed better for the minimum-energy problem (99.84% convergence rate) than for the minimum-time problem (98.91% convergence rate). All the failures could be attributed to the SCP algorithm trying to find a solution around a 'side' of the keep-out constraint where no feasible solution could be obtained. This appears to be a downside of the SCP algorithm, as a convergence rate lower than 99% could be insufficient for onboard applications. However, it was found that alternative initialisation trajectories could result in the SCP algorithm finding an optimal solution for these test cases. Therefore, it is thought that adding initialisation-dependent logic to the SCP algorithm, which is to some extent inherently included in most NLP solving algorithms (through a so-called 'restoration phase'), could improve the convergence rate of the SCP algorithm. The development of this initialisation-dependent logic is recommended for further research. Furthermore, it should be noted that the convergence rates obtained for this Monte Carlo analysis are conservative. The Monte Carlo test cases were designed to be very complex, signalled by the number of cases that turned out to be infeasible. Consequently, it could well be that for a problem with smaller attitude keep-out constraints, these convergence problems would not occur.

9.3.2. Optimality

In Figure 9.39, the results of all six Monte Carlo analyses are shown in terms of the optimality of the obtained guidance trajectories.

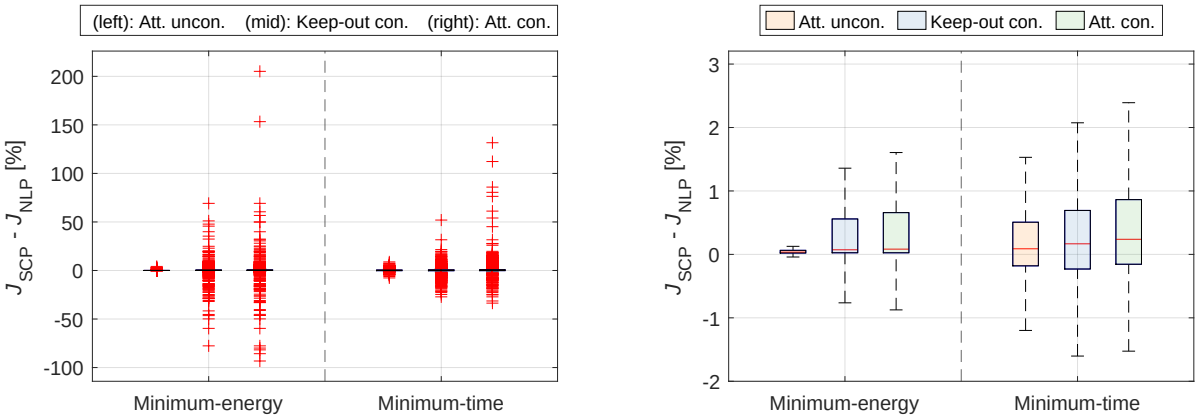


Figure 9.39: Performance in terms of optimality of the SCP algorithm for the six Monte Carlo analyses. The figure on the left shows all outliers, while the figure on the right does not consider these in order to assess the median performance.

From Figure 9.39, it can be concluded that:

- The SCP algorithm shows similar median performance as the NLP benchmark algorithm. This can be considered to be an excellent result, as pseudospectral NLP methods are considered as the standard for finding optimal results in industrial applications and in most studies on optimal spacecraft reorientation (Boyarko et al., 2011; Ventura et al., 2015). The slight degradation in optimality for the minimum-time keep-out constrained and attitude-constrained problems is thought to be caused by the fact that, for this specific problem set-up, the Lagrange-modelled control of the NLP benchmark algorithm better represents the true optimal control profile than the FOH-modelled control of the SCP algorithm. It should be noted that, for different problem formulations (e.g., larger number of nodes, no angular rate constraints), it is thought that the FOH-modelled control could better approximate the discrete control profiles that typically correspond to optimal minimum-time solutions. Another cause could be that the non-equidistant, pseudospectral grid of the NLP benchmark algorithm allows for larger constraint violations than the equidistant grid of the SCP algorithm.

- A significant number of outliers were observed for both the minimum-energy and minimum-time problems, in particular for the reorientation problems that include attitude constraints. However, no significant trend was identified that showed that either the SCP or NLP algorithm consistently outperformed the other algorithm. These outliers were found to be caused by the fact that both algorithms are local optimisation methods that sometimes arrived at different locally optimal solutions. Consequently, these outliers present no argument against the conclusion that the SCP algorithm shows good performance in terms of optimality.

9.3.3. Computational efficiency

In Figure 9.40, the obtained computation times of both the SCP and NLP algorithms are provided for the three different minimum-energy reorientation problems that were studied in Section 9.1.

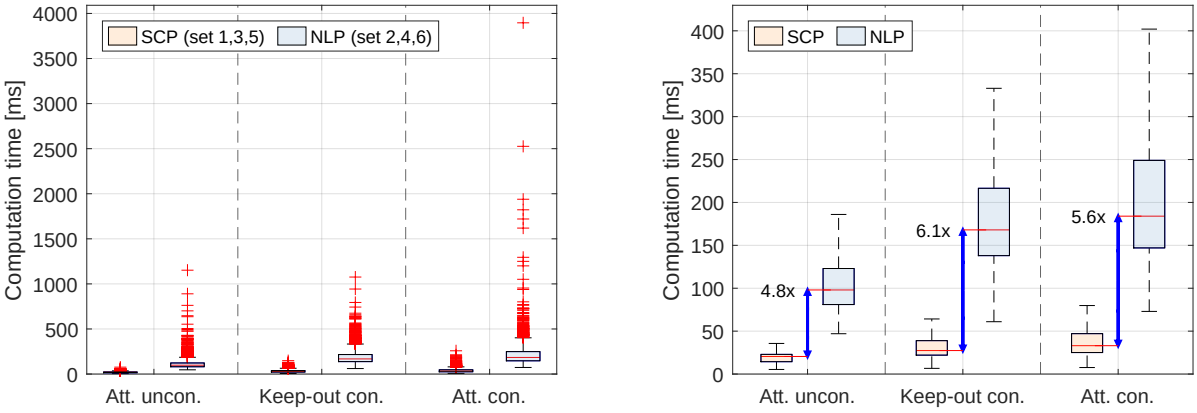


Figure 9.40: Performance in terms of computational efficiency of the SCP and NLP algorithms for the three minimum-energy Monte Carlo analyses. The figure on the left shows all outliers, while the figure on the right does not consider these to assess the median performance.

In Figure 9.40, these results are shown for the minimum-time Monte Carlo analyses of Section 9.2.

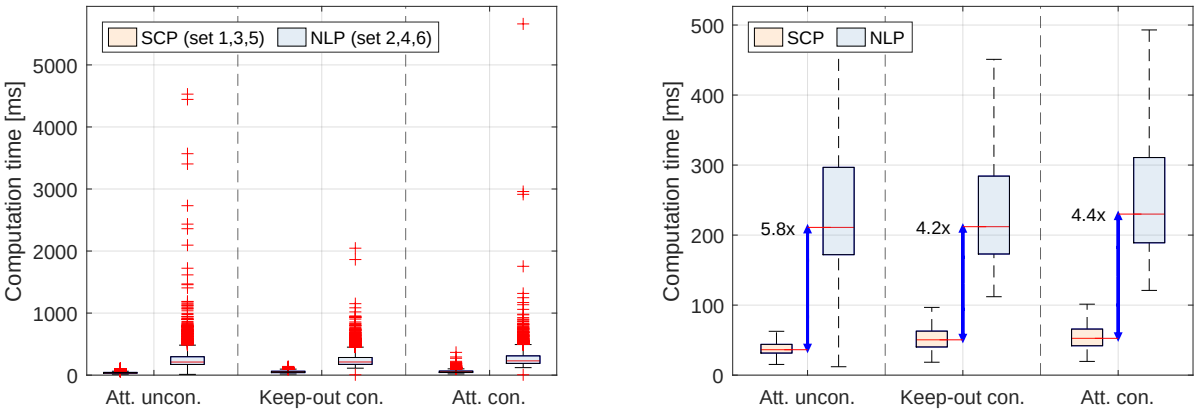


Figure 9.41: Performance in terms of computational efficiency of the SCP and NLP algorithms for the three minimum-time Monte Carlo analyses. The figure on the left shows all outliers, while the figure on the right does not consider these to assess the median performance.

The comparison between the SCP and NLP benchmark algorithms with regard to computational efficiency is very relevant, as the main promise of the SCP framework is that it is thought to be significantly faster than the pseudospectral NLP methods that are currently used for optimal spacecraft reorientation.

However, it is noted that no final conclusions can be drawn from this comparison. The computation times of these algorithms do strongly depend on the specific set-up of this study and the specific problem that is being solved. A range of factors, amongst others, the number of grid nodes, type of NLP solver, convergence tolerances, computation platform, constraint enforcement, and many others, all influence the performance of both the SCP and NLP algorithms. Furthermore, although the NLP method used in this study is already highly efficient compared to the NLP solvers used in attitude guidance literature (through code generation and algorithmic differentiation using CasADi), optimising the performance of the NLP algorithm was not the priority of this study. It could be possible that more optimised, faster NLP optimisation algorithms exist or could be developed. In the end, the only way to be able to confidently assess whether an SCP algorithm would be fast enough for onboard implementation would be to define specific requirements for a space mission and evaluate the performance of the SCP algorithm on actual satellite hardware.

That said, two conclusions can be drawn from the results presented in Figures 9.40 and 9.41:

- The median computational efficiency of the SCP algorithm is a factor of 4.2 to 6.1 higher than the efficiency of the NLP benchmark algorithm, depending on the exact problem type that is being solved. This

strongly supports the conclusion that the SCP algorithm can deliver very high computational speeds, possibly at the level required for onboard applications.

- Potentially as important as the increase in median computational efficiency, it was found that the computation times of the SCP algorithm are bounded within a smaller range than for the NLP benchmark algorithm. In other words, outliers regarding computation times show more predictable behaviour for the SCP algorithm than for the NLP benchmark algorithm. This finding is important regarding onboard applications, as these require predictable performance from the perspectives of robustness and reliability. The outliers of the NLP algorithm that can be observed in Figures 9.40 and 9.41 represent an important weakness of the corresponding class of optimisation methods.

10

Conclusions and Recommendations

In this chapter, the main conclusions of this study are presented, and recommendations are provided for further research. First, the conclusions and answers to the research (sub-)questions are listed in Section 10.1. Next, based on these conclusions, in Section 10.2, recommendations are provided for further development of SCP-based optimisation methods for robustly solving the optimal spacecraft reorientation problem in real time.

10.1. Conclusions

The research question that was formulated in Section 1.3 is:

Could a sequential convex programming-based optimisation method be a candidate for solving the optimal spacecraft reorientation problem in onboard applications?

Everything considered, the answer to this question is 'yes'. In the remainder of this section, the conclusions are presented that form a foundation for this answer. To provide structure, the main conclusions are grouped with respect to the research sub-questions defined in Section 1.3 to which they provide an answer.

Research sub-question 1: *Could SCP-based optimisation methods be used to solve the complex minimum-time reorientation problem with time-dependent boundary conditions and conical attitude constraints?*

- **SCP-based optimisation algorithms can be used to solve complex optimal spacecraft reorientation problems**.
  In this study, an SCP-based optimisation method was developed that can incorporate a variety of complex problem elements: a minimum-time objective function, combinations of conical keep-out and keep-in attitude constraints, time-dependent boundary conditions and ellipsoidal constraints. This is an important advantage of SCP-based optimisation algorithms compared to other real-time optimisation methods for the spacecraft reorientation problem that are currently available, as these typically rely on a variety of problem simplifications (see Section 3.8).

Research sub-question 2: *How could the performance of current SCP-based optimisation algorithms be improved?*

- **The first-order-hold discretisation method (which uses the state transition matrix) implemented in this study shows performance that is comparable to or better than the state-of-the-art reported for different discretisation methods**.
  Compared to earlier studies that applied the sequential convex programming framework to the spacecraft reorientation problem (e.g., from McDonald et al. (2020) and Virgili-Llop et al. (2018)), the performance is either matched or improved. These findings are in line with the performance of this discretisation method that has been reported in studies concerning other space-related optimisation problems (e.g., from Szmuk et al. (2020) and Malyuta et al. (2019)).

- **Large inter-nodal constraint violations were found to pose a challenge to the usability of SCP-based optimisation algorithms. A novel method was developed in this study and shows potential to overcome this challenge**.
  These inter-nodal constraint violations were particularly large, up to an unacceptable 35%, for problems that combined a minimum-time objective with angular rate constraints. This phenomenon has, to the knowledge of the author, not been discussed in literature at the time of writing, let alone solved. Therefore, a novel technique was developed that leverages the iterative nature of the SCP framework and re-uses information from the discretisation process. Using this method, worst-case angular-rate violations were reduced from 35% to 5%, while leading to a significantly smaller decrease in computational efficiency than simply adding more discretisation nodes. Moreover, strategies were identified that could result in further reductions to a defined level corresponding to some mission requirement.

- **A range of other techniques and methods were identified to have a positive impact on the performance of the SCP algorithm**.
  To begin with, both adaptive trust-region and virtual-control techniques were found to increase the convergence rate (robustness) of the SCP algorithm. Furthermore, the implementation of a shape-based initialisation method led to a significant increase in median computational efficiency of up to 40%, compared to initialisation methods used in other studies (e.g., from Reynolds et al. (2021) and McDonald et al. (2020).

Research sub-question 3: *How could the first steps towards an onboard implementation of an SCP algorithm be executed?*

This research question was answered in two ways. To begin with, the convex solver ECOS was implemented directly, eliminating the use of interfacing tools that are not compatible with satellite hardware. In addition, using MATLAB Coder, highly efficient C-code was generated to perform the discretisation step of the SCP algorithm, which is known as its second-most time-consuming step. As ECOS is also available as a set of C files, the remaining step required for realising an onboard implementation of the SCP algorithm is converting the MATLAB routine to a C routine. These implementations resulted in the following conclusion:

- **The discretisation step of the SCP algorithm accounts for about 10% of the total solve time**.
  Executing both the discretisation and ECOS solve steps with C-code that can be used on embedded systems allowed for a proper comparison of computation times. This finding validates the common assumption in literature that the ECOS solve step accounts for the majority of the total solution time.

Research sub-question 4: *What are the performance characteristics of an SCP-based optimisation algorithm in terms of computational efficiency, optimality and robustness?*

In this study, six extensive Monte Carlo analyses were performed, which allowed for a thorough evaluation of the performance of the SCP algorithm.

- **The SCP algorithm was found to deliver strong performance in terms of computational efficiency**.
  The main promise of the SCP framework is that it is supposed to be significantly faster than the NLP optimisation methods that are currently used. In this study, the median computational efficiency of the SCP algorithm was found to be a factor of 4.2 to 6.1 higher than the computational efficiency of the NLP benchmark algorithm, depending on the problem type. Furthermore, even larger efficiency differences were observed between the outliers of the SCP and NLP algorithms. However, tests need to be performed on actual satellite hardware to draw final conclusions on whether SCP-based optimisation algorithms are computationally fast enough for onboard applications.

- **The guidance solutions of the SCP algorithm show satisfactory performance in terms of optimality**.
  For all six Monte Carlo analyses, the optimality of the SCP solutions for the majority of the test cases was within 1% of the optimality of the NLP benchmark algorithm. As pseudospectral NLP algorithms are widely regarded as the standard for finding optimal solutions for nonlinear problems, this is a very promising result. For some cases, due to the fact that both the SCP and NLP algorithms are local optimisation methods, differences were encountered in the optimality of the solutions. However, neither the SCP nor the NLP algorithm was found to consistently outperform the other method.

- **The SCP algorithm shows excellent performance in terms of robustness, except for reorientation problems with both attitude keep-out and keep-in constraints**.
  For both the minimum-energy and minimum-time problems, 100% convergence rates were obtained for the Monte Carlo analyses except for the two analyses that included both attitude keep-out and keep-in constraints. For these problems, convergence rates of 99.84% and 98.91% were obtained for, respectively, the minimum-energy and minimum-time reorientation problems. However, it is believed that these convergence rates could be increased in future research through the introduction of initialisation-based logic to the SCP algorithm.

10.2. Recommendations

In this section, several recommendations are provided for further research. These recommendations are split up into two main directions: recommendations to bridge the gap from general research to actual implementations of SCP-based optimisation algorithms, and recommendations to further improve the performance of SCP-based optimisation algorithms.

The recommendations to increase the maturity level of SCP-based optimisation algorithms for the spacecraft reorientation problem are:

- For further research, it is recommended to define specific system requirements for a realistic space mission scenario that uses onboard attitude guidance. Due to the novelty of the SCP framework, this study took a more general approach to assessing whether SCP-based optimisation algorithms could be a suitable candidate for onboard applications. Formulating realistic requirements on efficiency, optimality, robustness, accuracy, and constraint violations would be required for a concluding assessment of the potential of SCP-based optimisation methods.

- In addition, it would be necessary to investigate the performance of the SCP-based optimisation algorithm using satellite hardware. The computation times reported in this study were obtained using a personal computer, which is not representative for real-world satellite hardware. Therefore, the next step would be to analyse the performance of the SCP-based optimisation algorithm on an attitude control system test bench.

The recommendations to further develop and increase the performance of SCP-based optimisation algorithms are:

- The discretisation method is one of the most influential design elements of an SCP algorithm. While the FOH method implemented in this study is considered to be one of the most promising methods, it would be interesting to thoroughly compare its performance to pseudospectral discretisation methods, for instance, the methods proposed by Sagliano (2019) and Sagliano et al. (2020). The reason for this is that it is thought (Subsection 4.5.2) that pseudospectral methods perform relatively well when the number of grid nodes of the optimisation problem is small, which is expected to be the case for most applications of the spacecraft reorientation problem.

- The method developed in this study to enforce constraints on inter-nodal segments was shown to be very effective. It would be interesting to further develop this method. Specifically, as mentioned in Section 8.3, it would be recommended to develop techniques that include certain logic to select the most effective points on the inter-nodal segments to apply these additional constraints to further reduce the extent of the (angular rate) constraint violations.

- A third recommendation concerns the development of initialisation-dependent logic for the SCP algorithm. The SCP algorithm was found to sometimes fail to converge for reorientation problems that included both attitude keep-out and keep-in constraints. As these failures were found to be initialisation-dependent, it would be interesting to develop techniques that could increase the convergence rate of the SCP algorithm.

- Although the initialisation method developed and implemented in this study shows significant performance improvements with respect to the methods proposed in available literature, there is still room for improvement. Specifically, this research project did not focus on optimising the initial guess for the

duration of the reorientation manoeuvre and its control profile, leaving both as promising areas to improve the performance of the SCP algorithm (in terms of computational efficiency). Furthermore, the initial guess of the state trajectory could be improved by taking into account information regarding the 3-D geometry of the reorientation manoeuvre and the imposed conical attitude constraints.

- At the moment of writing, the ECOS convex solving algorithm appears to be the preferred option for the majority of studies on the SCP framework. As the convex solving algorithm has a significant influence on the performance of an SCP-based optimisation algorithm, it would be interesting to either develop alternative convex solvers or implement them if they become available. For the more developed lossless convexification framework, for instance, customised solvers were developed that showed significant improvements in computational efficiency (Dueri et al., 2016).
- Finally, to the knowledge of the author, current studies employing SCP-based optimisation methods work with a constant number of grid nodes. It would be very interesting to investigate whether an adaptive number of nodes (between iterations) could lead to performance improvements. For instance, by using a lower number of nodes for the initial iterations of the sequential algorithm, the computational efficiency could potentially be increased. In addition, defining a higher number of nodes near certain constraints could reduce the occurrence and severity of inter-nodal constraint violations.

Bibliography

Behçet Açikmeşe and Lars Blackmore. Lossless Convexification of a Class of Optimal Control Problems with Non-Convex Control Constraints. *Automatica*, 47(2):341–347, 2011. ISSN 00051098. doi: 10.1016/j.automatica.2010.10.037.

Behçet Açikmeşe and Scott R. Ploen. Convex Programming Approach to Powered Descent Guidance for Mars Landing. *Journal of Guidance, Control, and Dynamics*, 30(5):1353–1366, 2007. ISSN 15333884. doi: 10.2514/1.27553.

Behçet Açikmeşe, Mi Mi Aung, Jordi Casoliva, Swati Mohan, Andrew Johnson, Daniel Scharf, David Masten, Joel Scotkin, Aron Wolf, and Martin W. Regehr. Flight Testing of Trajectories Computed by G-FOLD: Fuel Optimal Large Divert Guidance Algorithm for Planetary Landing. *Advances in the Astronautical Sciences*, 148(January):1867–1880, 2013. ISSN 00653438.

Erling D. Andersen and Knud D. Andersen. The Mosek Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm. In *High Performance Optimization*, volume 33, chapter 8, pages 197–232. Springer, Boston, 2000. ISBN 9781441948199.

Joel A. E. Andersson, Joris Gillis, Greg Horn, James B. Rawlings, and Moritz Diehl. CasADi - A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 2018.

Michael S. Andrle and John L. Crassidis. Geometric Integration of Quaternions. *Journal of Guidance, Control, and Dynamics*, 36(6):1762–1767, 2013. ISSN 15333884. doi: 10.2514/1.58558.

Robin Aucoin and Robert E. Zee. Real-Time Optimal Slew Maneuver Planning for Small Satellites. In *Annual AIAA/USU Conference on Small Satellites*, Logan, United States, 2019.

Boris Benedikter, Alessandro Zavoli, and Guido Colasurdo. A Convex Approach to Rocket Ascent Trajectory Optimization. In *8th European Conference for Aeronautics and Aerospace Sciencies (EUCASS)*, Madrid, Spain, 2019a. doi: 10.13009/EUCASS2019-430.

Boris Benedikter, Alessandro Zavoli, and Guido Colasurdo. A Convex Optimization Approach for Finite-Thrust Time-Constrained Cooperative Rendezvous. *American Astronoutical Society 19-763*, pages 1–16, 2019b.

Boris Benedikter, Alessandro Zavoli, Guido Colasurdo, Simone Pizzurro, and Enrico Cavallini. Convex Optimization of Launch Vehicle Ascent Trajectory with Heat-Flux and Splash-Down Constraints. *American Astronoutical Society 20-647*, (August):1–20, 2020. URL `http://arxiv.org/abs/2008.13239`.

Camille Bergin, Gillian McGlothin, Spencer McDonald, and Zhenbo Wang. Minimum-Fuel Low-Thrust Transfers for Spacecraft: An Alternative Convex Approach. (January):1–18, 2020. doi: 10.2514/6.2020-1692.

Jean Paul Berrut and Lloyd N. Trefethen. Barycentric Lagrange Interpolation. *SIAM Review*, 46(3):501–517, 2004. ISSN 00361445. doi: 10.1137/S0036144502417715.

John T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998. ISSN 15333884. doi: 10.2514/2.4231.

M. R. Bhagat. Convex Guidance for Envisat Rendezvous, 2016. URL `http://repository.tudelft.nl/`.

Karl D. Bilimoria and Bong Wie. Time-Optimal Three-Axis Reorientation of a Rigid Spacecraft. *Journal of Guidance, Control, and Dynamics*, 16(3):446–452, 1993. ISSN 07315090. doi: 10.2514/3.21030.

Paul T. Boggs and Jon W. Tolle. Sequential Quadratic Programming. *Acta Numerica*, 4(January):1–51, 1995. ISSN 14740508. doi: 10.1017/S0962492900002518.

Riccardo Bonalli, Abhishek Cauligi, Andrew Bylard, and Marco Pavone. GuSTO: Guaranteed Sequential Trajectory Optimization via Sequential Convex Programming. In *IEEE International Conference on Robotics and Automation*, pages 6741–6747, Montreal, Canada, 2019. ISBN 9781538660263. doi: 10.1109/ICRA.2019.8794205.

George A. Boyarko, Marcello Romano, and Oleg A. Yakimenko. Time-optimal reorientation of a spacecraft using an inverse dynamics optimization method. *Journal of Guidance, Control, and Dynamics*, 34(4):1197–1208, 2011. ISSN 15333884. doi: 10.2514/1.49449.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004. ISBN 978-0-521-83378-3.

Albert Caubet and James D. Biggs. Optimal attitude motion planner for large slew maneuvers using a shape-based method. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, pages 1–9, Boston, United States, 2013. ISBN 9781624102240. doi: 10.2514/6.2013-4719.

R. Chai, Al Savvaris, Antonios Tsourdos, Senchun Chai, and Yuanqing Xia. A Review of Optimization Techniques in Spacecraft Flight Trajectory Design. *Progress in Aerospace Sciences*, 109, 2019. ISSN 03760421. doi: 10.1016/j.paerosci.2019.05.003. URL `https://doi.org/10.1016/j.paerosci.2019.05.003`.

James Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 58:1–35, 2006. ISSN 14602431. URL `ftp://sbai2009.ene.unb.br/Projects/GPS-IMU/George/arquivos/Bibliografia/79.pdf`.

Matthijs Diercks. *Real-time, optimal spacecraft reorientation using convex programming*. Delft University of Technology, Delft, 2021.

Alexander Domahidi and Juan Jerez. Exercise 8: ECOS: Embedded Conic Solver, 2015. URL `http://syscop.de/teaching/numerical-optimal-control/`.

Alexander Domahidi, Eric Chu, and Stephen Boyd. ECOS: An SOCP solver for embedded systems. *European Control Conference (ECC)*, pages 3071–3076, 2013. doi: 10.23919/ecc.2013.6669541.

Daniel Dueri, Behçet Açíkmeşe, Daniel P. Scharf, and Matthew W. Harris. Customized Real-Time Interior-Point Methods for Onboard Powered-Descent Guidance. *Journal of Guidance, Control, and Dynamics*, 40(2):197–212, 2016. ISSN 15333884. doi: 10.2514/1.G001480.

Utku Eren, Behçet Açikmeşe, and Daniel P. Scharf. A Mixed Integer Convex Programming Approach to Constrained Attitude Guidance. *European Control Conference (EEC)*, pages 1120–1126, 2015. doi: 10.1109/ECC.2015.7330690.

Fariba Fahroo and I. Micheal Ross. Direct Trajectory Optimization by a Chebyshev Pseudospectral Method. *Journal of Guidance, Control, and Dynamics*, 25(1):160–166, 2002. ISSN 15333884. doi: 10.2514/2.4862.

Rebecca Foust, Soon Jo Chung, and Fred Y. Hadaegh. Optimal guidance and control with nonlinear dynamics using sequential convex programming. *Journal of Guidance, Control, and Dynamics*, 43(4):633–644, 2020. ISSN 15333884. doi: 10.2514/1.G004590.

E. Frazzoli, M. A. Dahleh, and E. Feron. A Randomized Attitude Slew Planning Algorithm for Autonomous Spacecraft. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Montreal, 2001.

Divya Garg. *Advances in Global Pseudospectral Methods for Optimal Control*. University of Florida, Florida, 2011.

Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, 2020. URL `http://cvxr.com/cvx`.

Haitham Hindi. A Tutorial on Convex Optimization. In *American Control Conference*, pages 1–14, Palo Alto, 2004. URL `papers2://publication/uuid/10E3D710-413C-4675-891B-B1197F0E255B`.

Qinglei Hu, Yueyang Liu, Hongyang Dong, and Youmin Zhang. Saturated attitude control for rigid spacecraft under attitude constraints. *Journal of Guidance, Control, and Dynamics*, 43(4):790–805, 2020. ISSN 15333884. doi: 10.2514/1.G004613.

Juan Jerez, Sandro Merkli, Samir Bennani, and Hans Strauch. FORCES-RTTO: A tool for on-board real-time autonomous trajectory planning. In *10th international ESA Conference on Guidance, Navigation & Control*, number 1, pages 1–27, Salzburg, 2017.

M. Karpenko, S. Bhatt, N. Bedrossian, A. Fleming, and I. M. Ross. Flight implementation of pseudospectral optimal control for the TRACE space telescope. In *AIAA Guidance, Navigation, and Control Conference 2011*, number August, Portland, 2011. ISBN 9781600869525. doi: 10.2514/6.2011-6506.

Matthew P. Kelly. Transcription Methods for Trajectory Optimization: a beginners tutorial, 2015. URL `http://arxiv.org/abs/1707.00284`.

Yoonsoo Kim and Mehran Mesbahi. Quadratically Constrained Attitude Control via Semidefinite Programming. *IEEE Transactions on Automatic Control*, 49(5):731–735, 2004. ISSN 00189286. doi: 10.1109/TAC.2004.825959.

Yoonsoo Kim, Mehran Mesbahi, Gurkirpal Singh, and Fred Y. Hadaegh. On the Convex Parametrization of Spacecraft Orientation in Presence of Constraints and its Applications. *IEEE transactions on Aerospace and Electronic Systems*, 46(May 2014):1097 – 1109, 2010. doi: 10.1109/TAES.2010.5545176. URL `http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5545176&url=http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5545176`.

Henri C. Kjellberg and E. Glenn Lightsey. Discretized Constrained Attitude Pathfinding and Control for Satellites. *Journal of Guidance, Control, and Dynamics*, 36(5):1301–1309, 2013. ISSN 15333884. doi: 10.2514/1.60189.

Henri Christian Kjellberg. *Constrained attitude guidance and control for satellites*. Number December. Austin, 2014. doi: 10.13140/2.1.4877.6483.

J. B. Kuipers. *Quaternions & Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton University Press, Princeton, New Jersey, 1999.

Unsik Lee and Mehran Mesbahi. Spacecraft synchronization in the presence of attitude constrained zones. In *American Control Conference*, pages 6071–6076, Montreal, 2012. ISBN 9781457710964.

M. V. Levskii. The problem of the time-optimal control of spacecraft reorientation. *Journal of Applied Mathematics and Mechanics*, 73(1):16–25, 2009. ISSN 00218928. doi: 10.1016/j.jappmathmech.2009.03.012.

Daniel Liberzon. *Calculus of Variations and Optimal Control Theory*. Princeton University Press, Princeton, 2007.

Xinfu Liu and Ping Lu. Solving nonconvex optimal control problems by convex optimization. *Journal of Guidance, Control, and Dynamics*, 37(3):750–765, 2014. ISSN 15333884. doi: 10.2514/1.62110.

Xinfu Liu, Zuojun Shen, and Ping Lu. Entry Trajectory Optimization by Second-Order Cone Programming. *Journal of Guidance, Control, and Dynamics*, 39(2):227–241, 2015. ISSN 15333884. doi: 10.2514/1.G001210.

Xinfu Liu, Ping Lu, and Binfeng Pan. Survey of convex optimization for aerospace applications. *Astrodynamics*, 1(1):23–40, 2017. ISSN 2522-008X. doi: 10.1007/s42064-017-0003-8.

Miguel Sousa Lobo, Lieven Vandenberghe, Stephen Boyd, and Hervé Lebret. Applications of second-order cone programming. *Linear Algebra and Its Applications*, 284(1-3):193–228, 1998. ISSN 00243795. doi: 10.1016/S0024-3795(98)10032-0.

Ping Lu and Xinfu Liu. Autonomous Trajectory Planning for Rendezvous and Proximity Operations by Conic Optimization. *Journal of Guidance, Control, and Dynamics*, 36(2):375–389, 2013. ISSN 15333884. doi: 10.2514/1.58436.

Pádraig S Lysandrou. *A Successive Convexification Optimal Guidance Implementation for the Pinpoint Landing of Space Vehicles*. PhD thesis, University of Colorado, Colorado, 2019.

Danylo Malyuta, Taylor P. Reynolds, Michael Szmuk, Mehran Mesbahi, Behçet Açıkmeşe, and John M. Carson. Discretization Performance and Accuracy Analysis for the Powered Descent Guidance Problem. *AIAA Scitech Forum*, (January), 2019. doi: 10.2514/6.2019-0925.

Danylo Malyuta, Taylor Reynolds, Michael Szmuk, Behcet Acikmese, and Mehran Mesbahi. Fast Trajectory Optimization via Successive Convexification for Spacecraft Rendezvous with Integer Constraints. (January):1–24, 2020. doi: 10.2514/6.2020-0616.

Yuanqi Mao, Michael Szmuk, and Behçet Açikmeşe. Successive convexification of non-convex optimal control problems and its convergence properties. *IEEE 55th Conference on Decision and Control*, pages 3636–3641, 2016. ISSN 0018-9219. doi: 10.1109/CDC.2016.7798816.

Yuanqi Mao, Daniel Dueri, Michael Szmuk, and Behçet Açıkmeşe. Successive Convexification of Non-Convex Optimal Control Problems with State Constraints. *IFAC-PapersOnLine*, 50(1):4063–4069, 2017. ISSN 24058963. doi: 10.1016/j.ifacol.2017.08.789.

Yuanqi Mao, Michael Szmuk, and Behcet Açikmeşe. A Tutorial on Real-time Convex Optimization Based Guidance and Control for Aerospace Applications. *Proceedings of the American Control Conference*, pages 2410–2416, 2018a. ISSN 07431619. doi: 10.23919/ACC.2018.8430984.

Yuanqi Mao, Michael Szmuk, Xiangru Xu, and Behçet Açikmeşe. Successive Convexification: A Superlinearly Convergent Algorithm for Non-convex Optimal Control Problems. *Automatica, preprint*, (April), 2018b. URL `http://arxiv.org/abs/1804.06539`.

Jacob Mattingley and Stephen Boyd. CVXGEN: A code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012. ISSN 13894420. doi: 10.1007/s11081-011-9176-9.

Spencer McDonald and Zhenbo Wang. Real-Time Optimal Trajectory Generation for UAV to Rendezvous with an Aerial Orbit. In *AIAA Aviation Forum*, number June, Dallas, United States, 2019. doi: 10.2514/6.2019-3620.

Spencer McDonald, Timothy Grizzel, and Zhenbo Wang. A Real-Time Approach to Minimum-Energy Reorientation of an Asymmetric Rigid Body Spacecraft. Number January, pages 1–17, Orlando, 2020. doi: 10.2514/6.2020-1203.

Colin R. McInnes. Large Angle Slew Maneuvers with Autonomous Sun Vector Avoidance. *Journal of Guidance, Control, and Dynamics*, 17(4):875–877, 1994. ISSN 07315090. doi: 10.2514/3.21283.

Daniel Morgan, Giri P. Subramanian, Soon Jo Chung, and Fred Y. Hadaegh. Swarm Assignment and Trajectory Optimization using Variable-Swarm, Distributed Auction Assignment and Sequential Convex Programming. *International Journal of Robotics Research*, 35(10):1261–1285, 2016. ISSN 17413176. doi: 10.1177/0278364916632065.

Mauro Pontani and Robert G. Melton. Heuristic optimization of satellite reorientation maneuvers. *AIAA/AAS Astrodynamics Specialist Conference, 2016*, (September):1–29, 2016. doi: 10.2514/6.2016-5581.

L. S. Pontryagin. *Mathematical Theory of Optimal Processes*. Taylor & Francis, Milton Park, 1987.

Valentin Preda, Andrew Hyslop, and Samir Bennani. Optimal Science-time Reorientation Policy for the Comet Interceptor Flyby via Sequential Convex Programming. *CEAS Space Journal*, pages 1–18, 2021. ISSN 18682510. doi: 10.1007/s12567-021-00368-2.

Taylor P. Reynolds. *Computational Guidance and Control for Aerospace Systems*. PhD thesis, 2020.

Taylor P. Reynolds, Danylo Malyuta, Mehran Mesbahi, Behçet Açıkmeşe, and John M. Carson. A Real-Time Algorithm for Non-Convex Powered Descent Guidance. In *AIAA Scitech 2020 Forum*, pages 1–24, Orlando, 2020a. ISBN 9781624105951. doi: 10.2514/6.2020-0844.

Taylor P. Reynolds, Michael Szmuk, Danylo Malyuta, Mehran Mesbahi, Behçet Açıkmeşe, and John M. Carson. Dual quaternion-based powered descent guidance with state-triggered constraints. *Journal of Guidance, Control, and Dynamics*, 43(9):1584–1599, 2020b. ISSN 15333884. doi: 10.2514/1.G004536.

Taylor P. Reynolds, Charles L. Kelly, Cole Morgan, Arnela Grebovic, Jerrold Erickson, Henry Brown, William C. Pope, Jonathan Casamayor, Kyle Kearsley, Gorkem Caylak, Kyle E. Fisher, Cameron Wutzke, Kylle Ashton, James Rosenthal, Devan Tormey, Ellory Freneau, Garrett Giddings, Hasan Emin Horata, Anika Dighde, Saharsh Parakh, Jiaping Zhen, John C. Purpura, Daniel B. Pratt, and Anders Hunt. SOC-i: A CubeSat Demonstration of Optimization-Based Real-Time Constrained Attitude Control. *IEEE Aerospace Conference 2021 (preprint)*, 2021.

Krister de Ridder. *Convex Guidance, Navigation, and Control for Pin-Point Lunar Landing*. PhD thesis, Delft University of Technology, Delft, 2016.

Marco Sagliano. Pseudospectral convex optimization for powered descent and landing. *Journal of Guidance, Control, and Dynamics*, 41(2):320–334, 2018. ISSN 15333884. doi: 10.2514/1.G002818.

Marco Sagliano. Generalized hp Pseudospectral-Convex Programming for Powered Descent and Landing. *Journal of Guidance, Control, and Dynamics*, 42(7):1562–1570, 2019. ISSN 15333884. doi: 10.2514/1.G003731.

Marco Sagliano, Ansgar Heidecker, José Macés Hernández, Stefano Farì, Markus Schlotterer, Svenja Woicke, David Seelbinder, and Etienne Dumont. Onboard Guidance for Reusable Rockets: Aerodynamic Descent and Powered Landing. *Preprint*, (October), 2020.

Daniel P. Scharf, Martin W. Regehr, Geoffery M. Vaughan, Joel Benito, Homayoon Ansari, Mimi Aung, Andrew Johnson, Jordi Casoliva, Swati Mohan, Daniel Dueri, Behçet Açikmeşe, David Masten, and Scott Nietfeld. ADAPT Demonstrations of Onboard Large-Divert Guidance with a VTVL Rocket. *IEEE Aerospace Conference Proceedings*, 2014. ISSN 1095323X. doi: 10.1109/AERO.2014.6836462.

Daniel P. Scharf, Behçet Açíkmeşe, Daniel Dueri, Joel Benito, and Jordi Casoliva. Implementation and Experimental Demonstration of Onboard Powered-Descent Guidance. *Journal of Guidance, Control, and Dynamics*, 40(2):213–229, 2017. ISSN 15333884. doi: 10.2514/1.G000399.

H. Schaub and J. L. Junkins. *Analytical Mechanics for Spacecraft Systems*. American Institute of Aerosnautics and Astronautics, Reston, VA, 2 edition, 2009.

Ken Shoemake. Animating Rotation with Quaternion Curves. *Conference on Computer Graphics and Interactive Techniques*, page 245–254, 1985.

Malcom D. Shuster. A Survey of Attitude Representations. *Journal of the Astronautical Sciences*, 41(4):439–517, 1993. doi: 10.2514/6.2012-4422.

Emmanuel Sin, Sreeja Nag, Vinay Ravindra, Alan Li, and Murat Arcak. Attitude Trajectory Optimization for Agile Satellites in Autonomous Remote Sensing Constellations. In *AIAA Scitech 2021 Forum*, Nashville, 2021. URL `http://arxiv.org/abs/2102.07940`.

Zheng-yu Song, Cong Wang, Stephan Theil, David Seelbinder, Marco Sagliano, Xin-fu Liu, and Zhi-jiang Shao. Survey of Autonomous Guidance Methods for Powered Planetary Landing. *Frontiers of Information Technology & Electronic Engineering*, (5):1–23, 2020.

Dario Spiller, Fabio Curti, and Luigi Ansalone. Inverse dynamics particle swarm optimization for spacecraft minimum-time maneuvers with constraints. In *Conference of the Italian Association of Aeronautics and Astronautics*, Turin, 2015.

Dario Spiller, Robert G. Melton, and Fabio Curti. Inverse dynamics particle swarm optimization applied to bolza problems. *Advances in the Astronautical Sciences*, 162(January):1235–1254, 2018. ISSN 00653438.

Chuangchuang Sun and Ran Dai. Spacecraft Attitude Control Under Constrained Zones via Quadratically Constrained Quadratic Programming. In *AIAA Guidance, Navigation, and Control Conference*, Kissimmee, 2015. ISBN 9781624103391. doi: 10.2514/6.2015-2010.

Michael Szmuk and Behçet Açıkmeşe. Successive convexification for 6-DoF mars rocket powered landing with free-final-time. In *AIAA Guidance, Navigation, and Control Conference, 2018*, number 210039, pages 1–15, Kissimmee, 2018. ISBN 9781624105265. doi: 10.2514/6.2018-0617.

Michael Szmuk, Behçet Açıkmese, Andrew W. Berning, and Geoffrey Huntington. Successive convexification for fuel-optimal powered landing with aerodynamic drag and non-convex constraints. In *2016 AIAA Guidance, Navigation, and Control Conference*, number January, San Diego, 2016. ISBN 9781624103896.

Michael Szmuk, Carlo Alberto Pascucci, Daniel Dueri, and Behçet Açikmeşe. Convexification and real-time on-board optimization for agile quad-rotor maneuvering and obstacle avoidance. In *IEEE International Conference on Intelligent Robots and Systems*, pages 4862–4868, Vancouver, 2017. ISBN 9781538626825. doi: 10.1109/IROS.2017.8206363.

Michael Szmuk, Danylo Malyuta, Taylor P. Reynolds, Margaret Skye McEowen, and Behçet Açikmeşe. Real-Time Quad-Rotor Path Planning Using Convex Optimization and Compound State-Triggered Constraints. In *IEEE International Conference on Intelligent Robots and Systems*, pages 7666–7673, Macao, China, 2019. ISBN 9781728140049. doi: 10.1109/IROS40897.2019.8967706.

Michael Szmuk, Taylor P. Reynolds, and Behçet Açıkmeşe. Successive Convexification for Real-Time Six-Degree-of-Freedom Powered Descent Guidance with State-Triggered Constraints. *Journal of Guidance, Control, and Dynamics*, 43(8):1399–1413, 2020. ISSN 15333884. doi: 10.2514/1.G004549.

Margaret Tam and E. Glenn Lightsey. Constrained spacecraft reorientation using mixed integer convex programming. *Acta Astronautica*, 127:31–40, 2016. ISSN 00945765. doi: 10.1016/j.actaastro.2016.04.003. URL `http://dx.doi.org/10.1016/j.actaastro.2016.04.003`.

Sergei Tanygin. Three-axis constrained attitude pathfinding and visualization: Charting a course on higher-dimensional map. In *AIAA/AAS Astrodynamics Specialist Conference 2014*, number August, pages 1–30, San Diego, USA, 2014. ISBN 9781624103087. doi: 10.2514/6.2014-4101.

George Terzakis, Manolis Lourakis, and Djamel Ait-Boudaoud. Modified Rodrigues Parameters: An Efficient Representation of Orientation in 3D Vision and Graphics. *Journal of Mathematical Imaging and Vision*, 60 (3):422–442, 2018. ISSN 15737683. doi: 10.1007/s10851-017-0765-x.

W. T. Thomson. *Introduction to Space Dynamics*, volume 15. Wiley, New York, 1962. doi: 10.1063/1.3058399.

R. H. Tütüncü, K. C. Toh, and M. J. Todd. SDPT3 - a Matlab software package for version 3.0. Technical report, 2001.

Jacopo Ventura, Marcello Romano, and Ulrich Walter. Performance evaluation of the inverse dynamics method for optimal spacecraft reorientation. *Acta Astronautica*, 110:266–278, 2015. ISSN 00945765. doi: 10.1016/j.actaastro.2014.11.041. URL `http://dx.doi.org/10.1016/j.actaastro.2014.11.041`.

Josep Virgili-Llop, Alanna Sharp, and Marcello Romano. Reorientation of rigid spacecraft using onboard convex optimization. *Advances in the Astronautical Sciences*, 167:2783–2796, 2018. ISSN 00653438.

Josep Virgili-Llop, Costantinos Zagaris, Richard Zappulla, Andrew Bradstreet, and Marcello Romano. A convex-programming-based guidance algorithm to capture a tumbling object on orbit using a spacecraft equipped with a robotic manipulator. *International Journal of Robotics Research*, 38(1):40–72, 2019. ISSN 17413176. doi: 10.1177/0278364918804660.

Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2005. ISSN 00255610. doi: 10.1007/s10107-004-0559-y.

Alex Walsh, Ann Arbor, and James Richard Forbes. Constrained Attitude Control on SO (3) via Semidefinite Programming. *Journal of Guidance, Control, and Dynamics*, 41(11):2480–2485, 2018.

Jinbo Wang and Naigang Cui. A Pseudospectral-Convex Optimization Algorithm for Rocket Landing Guidance. *AIAA Guidance, Navigation, and Control Conference*, (210039):1–18, 2018. doi: 10.2514/6.2018-1871.

Jinbo Wang, Naigang Cui, and Changzhu Wei. Rapid trajectory optimization for hypersonic entry using a pseudospectral-convex algorithm. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(14):5227–5238, 2019a. ISSN 20413025. doi: 10.1177/0954410019840839.

Xiao Wang, Yulin Wang, Shengjing Tang, and Jie Guo. Entry Trajectory Optimization via hp Pseudospectral Convex Programming. In *16th International Conference on Informatics in Control, Automation and Robotics*, volume 1, pages 61–69, Prague, Czech Republic, 2019b. ISBN 9789897583803. doi: 10.5220/0007909800610069.

Zhenbo Wang. Optimal Trajectories and Normal Load Analysis of Hypersonic Glide Vehicles via Convex Optimization. *Aerospace Science and Technology*, 87:357–368, 2019. ISSN 12709638. doi: 10.1016/j.ast.2019.03.002. URL `https://doi.org/10.1016/j.ast.2019.03.002`.

Zhenbo Wang and Michael J. Grant. Constrained trajectory optimization for planetary entry via sequential convex programming. In *AIAA Atmospheric Flight Mechanics Conference*, number June, pages 1–23, Washington, USA, 2016. doi: 10.2514/6.2016-3241.

Zhenbo Wang and Michael J. Grant. Constrained trajectory optimization for planetary entry via sequential convex programming. *Journal of Guidance, Control, and Dynamics*, 40(10):2603–2615, 2017. ISSN 15333884. doi: 10.2514/1.G002150.

Zhenbo Wang and Michael J. Grant. Autonomous Entry Guidance for Hypersonic Vehicles by Convex Optimization. *Journal of Spacecraft and Rockets*, 55(4):993–1006, 2018a. ISSN 15336794. doi: 10.2514/1.A34102.

Zhenbo Wang and Michael J. Grant. Minimum-Fuel Low-Thrust Transfers for Spacecraft: A Convex Approach. *IEEE Transactions on Aerospace and Electronic Systems*, 54(5):2274–2290, 2018b. ISSN 15579603. doi: 10.1109/TAES.2018.2812558.

Zhenbo Wang and Ye Lu. Improved Sequential Convex Programming Algorithms for Entry Trajectory Optimization. *Journal of Spacecraft and Rockets*, pages 1–14, 2020. ISSN 0022-4650. doi: 10.2514/1.a34640.

Zhu Wang, Guangtong Xu, Li Liu, and Teng Long. Obstacle-avoidance trajectory planning for attitude-constrained quadrotors using second-order cone programming. In *2018 Aviation Technology, Integration, and Operations Conference*, Atlanta, USA, 2018. ISBN 9781624105562. doi: 10.2514/6.2018-3035.

Andreas Wenzel and David Seelbinder. *On-Board Convex Optimization for Powered Descent Landing of EAGLE*. PhD thesis, Lulea University of Technology, Lulea, 2017.

James R. Wertz. *Spacecraft Attitude Determination and Control. Dordrecht*. Reidel Publishing Company, Dordrecht, Holland, 1978. ISBN 9789027712042.

Bong Wie. *Spacecraft Dynamics and Control*. American Institute of Aeronautics and Astronautics, 2nd edition, 2008.

M. A. Wolfe and R. Osten. JWST Primer v 3.0. Technical report, Baltimore, United States, 2014.

Rui Xu, Hui Wang, Wenming Xu, Pingyuan Cui, and Shengying Zhu. Rotational-path decomposition based recursive planning for spacecraft attitude reorientation. *Acta Astronautica*, 143(October 2017):212–220, 2018. ISSN 00945765. doi: 10.1016/j.actaastro.2017.11.035. URL `https://doi.org/10.1016/j.actaastro.2017.11.035`.

Rui Xu, Hui Wang, Shengying Zhu, Huiping Jiang, and Zhaoyu Li. Multiobjective planning for spacecraft reorientation under complex pointing constraints. *Aerospace Science and Technology*, 104:106002, 2020. ISSN 12709638. doi: 10.1016/j.ast.2020.106002. URL `https://doi.org/10.1016/j.ast.2020.106002`.

Runqiu Yang and Xinfu Liu. Fuel Optimal Powered Descent Guidance with Free Final-Time and Path Constraints. *Acta Astronautica*, 172(March):70–81, 2020. ISSN 00945765. doi: 10.1016/j.actaastro.2020.03.025. URL `https://doi.org/10.1016/j.actaastro.2020.03.025`.

Xiang Zhou, Hong Bo Zhang, Lei Xie, Guo Jian Tang, and Wei Min Bao. An Improved Solution Method via the Pole-Transformation Process for the Maximum-Crossrange Problem. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 234(9):1491–1506, 2020. ISSN 20413025. doi: 10.1177/0954410020914809.