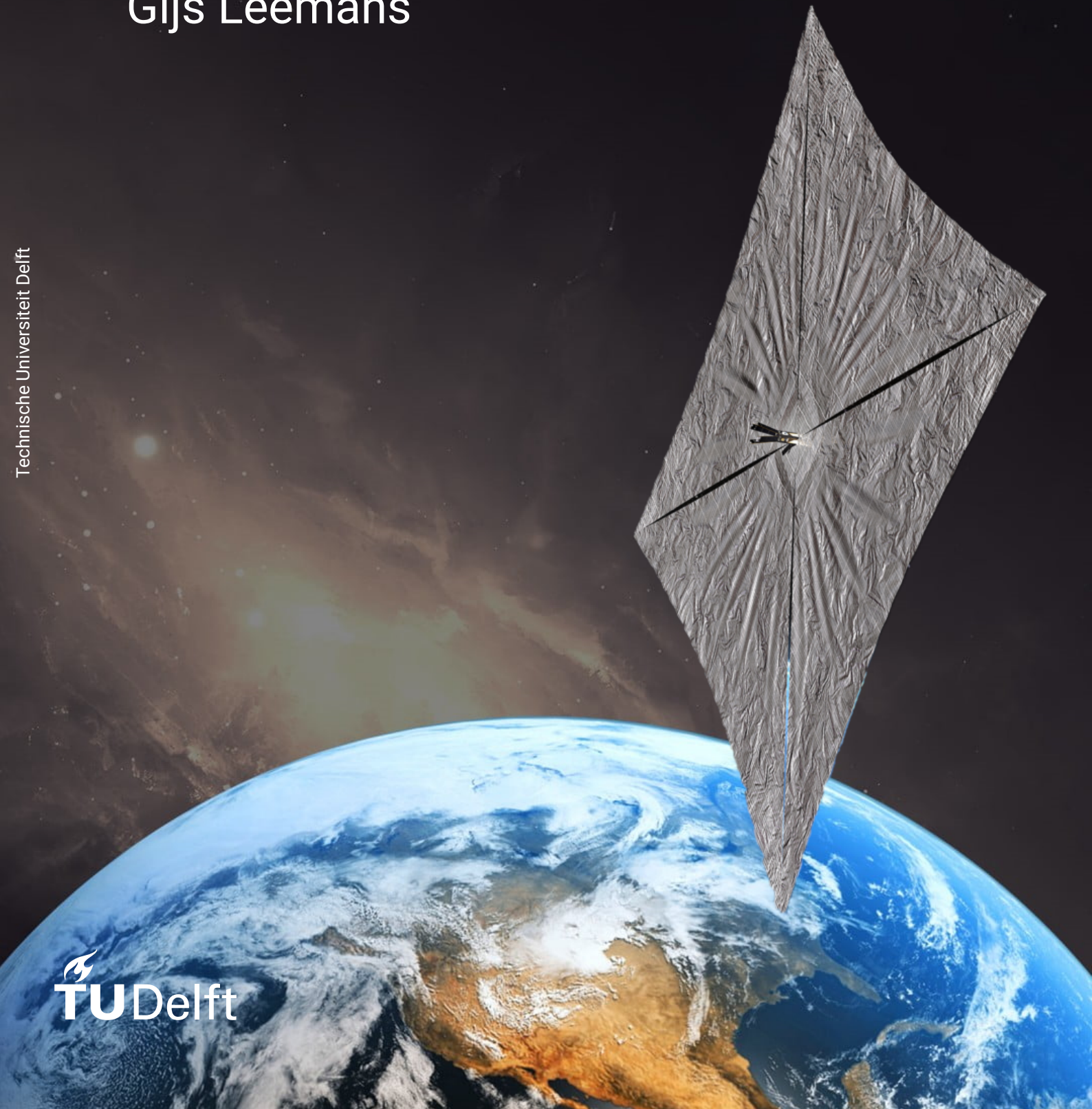


# Many-revolution Earth-centred solar-sail trajectory optimisation using differential dynamic programming

Gijs Leemans





# Many-revolution Earth-centred solar-sail trajectory optimisation using differential dynamic programming

by

Gijs Leemans

Dissertation for the degree of

**Master of Science**

at The Astrodynamics and Space Missions Section,  
Delft University of Technology,  
to be defended publicly on Tuesday, August 31, 2021

Supervisor:	Dr. ir. J. Heiligers	
Co-supervisor:	Ir. L. Carzana	
Thesis committee:	Dr. D. M. Stam	Committee chair,
	Dr. ir. J. Heiligers	Supervisor,
	Ir. L. Carzana	Co-supervisor,
	Dr. F. Oliviero	External examiner

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

*We cannot direct the winds,  
but we can adjust the sails.*

Thomas S. Monson

# Preface

Handing in this work feels somewhat strange to me. Over the last two years, my life has been a blur of assignments, exams and thesis work. Although it has been busy and frustrating at times, I did enjoy (almost) every single minute of it. After I got my bachelor's degree in Mechanical Engineering, also at Delft University of Technology, spaceflight was really the only master's programme that excited me. I knew that it was not going to be easy without having any background in the field at all, but I think it has been one of the best decisions I have ever made. From the very first time I heard about the concept of solar sailing, I was immediately intrigued by the possibilities that a form of never-ending propulsion in space could bring. I became even more fascinated by the concept when I learned that it was not pure science-fiction, but several solar sails have already successfully been deployed in space. I sincerely hope that this work can contribute to the knowledge base on optimisation for planet-centred solar-sail trajectories.

First and foremost, I want to thank Jeannette Heiligers, as the main supervisor of this work, and Livio Carzana. The weekly meetings were always a massive boost for me, it is not without a reason that Wednesday mornings were always my most productive part of the week. Even though Jeannette and Livio did not know all the ins and outs on how the optimisation algorithm worked, their questions always guided me in the right direction to solve my problems. Thanks to them, the main body of this work has already been submitted to be presented at the 2022 American Institute of Aeronautics and Astronautics SciTech forum.

Another word of gratitude goes out towards all authors that have performed research on the topic of solar sailing and low-thrust trajectory optimisation. In particular, Dr. J. Aziz, who's work on low-thrust many-revolution trajectory optimisation for solar-electric propulsion largely inspired this thesis.

Finally, I want to thank Merle and my housemates, for listening to my everyday struggles when my algorithm was not fully working yet and for all the times when they acted impressed when I showed them my latest results, which were mostly just some lines in a figure. Of course, it goes without saying that I could never be where I am without my family and friends.

*Gijs Leemans  
Delft, July 2021*



# Summary

Solar sailing is a form of low-thrust space propulsion that uses a large, thin, mirror-like structure to produce thrust from reflecting solar photons. The solar sail does not require any fuel, therefore the total work done by the sail is only limited by the lifetime of the sail itself. The propellant-less nature of solar sailing is interesting for a wide range of space applications, ranging from interplanetary trajectories to debris removal around Earth. The acceleration that the solar sail imposes on the spacecraft is very small, therefore it may take a long time for an orbital transfer to be completed. As a result, it is desirable to find the optimal attitude of the sail such that an orbital transfer can be completed as efficiently as possible. For Earth-centred trajectories, many orbital revolutions around the Earth might be needed before the desired state is reached. This makes solar-sail planet-centred trajectory optimisation more complex, as it leads to a sensitive and high-dimensional problem. Historically, optimisation of these trajectories has been performed using (semi-) analytical techniques, resulting in locally optimal solutions. An optimisation method that has never been explored for planet-centred solar-sail trajectory optimisation is differential dynamic programming (DDP). DDP is an optimisation scheme that finds the optimal control by discretising the satellite's trajectory into stages. At each stage, a local, quadratic model of the cost is minimised by approximating the cost function with a second-order Taylor series expansion around the state, controls and Lagrange multipliers. By taking the derivative of this expansion with respect to the controls and setting it to zero, the optimal control update is found. This thesis work demonstrates the usability of differential dynamic programming to obtain, for the first time, globally optimal Earth-centred solar-sail trajectories. To this end, DDP is combined with a global optimisation heuristic, monotonic basin hopping (MBH).

The solar-sail acceleration is modelled by assuming an ideal solar-sail reflectance model, which implies that all incoming radiation is specularly reflected and the acceleration acts along the vector normal to the sail surface. Eclipses are important for low-altitude planet-centred trajectories and are therefore modelled with a Heaviside-based sunlight fraction. The numerical performance of the optimisation algorithm is enhanced by integrating the sailcraft state in modified equinoctial elements and performing a Sundman transformation, which changes the independent variable for integration from time to the true anomaly.

The DDP algorithm was used on different test cases to validate and assess the performance of different aspects of the algorithm, which is elaborated upon in the coming paragraphs. First, validation of the algorithm on a simple, two-dimensional case took place. Next, the influence on varying settings of the algorithm was investigated, as well as its ability to find globally optimal solutions with MBH. Next, The ability of the DDP algorithm to enforce terminal constraints was explored for NASA's upcoming ACS3 mission, in addition to the effect of different perturbing accelerations on the optimality of the solution. At last, two many-revolution cases with constraints and perturbing accelerations were optimised.

To validate the implementation of DDP for the purpose of many-revolution Earth-centred solar-sail trajectory optimisation, the solution found by DDP is compared against the result of a locally optimal steering law which maximises the specific orbital energy of a satellite in geostationary orbit within three orbital revolutions. For this comparison, a two-dimensional dynamical model is assumed where the direction of sunlight is taken to coincide with the orbital plane and the Sun remains at a fixed position with respect to the Earth. The control is then solely determined by the sail cone angle as the clock angle is fixed at  $90^\circ$  to restrict the acceleration to the orbital plane. The resulting semi-major axis increase found by the DDP algorithm is slightly larger than the solution from the locally optimal steering law, thereby validating the optimality of the solution found by the DDP algorithm.

The performance of the DDP algorithm relies on the choice for certain settings (e.g., the 'optimality tolerance' and the 'initial trust region radius'). The effect of changing these settings is investigated to find out how robust the DDP algorithm is to different settings and what the best settings for the remaining test cases are. To test the effect of the different settings, a more complex case is investigated. The problem is expanded into three dimensions by removing the assumption on the direction of sunlight to be fixed and planar. Instead, the Sun's position as a function of time is retrieved from ephemeris data. For this more complicated, three-dimensional case, the DDP algorithm is also able to find an optimal solution. Three different values were used for each setting and the resulting semi-major axis increase and runtime of the DDP algorithm were documented for comparison. For the majority, changing the settings does not have a large influence, although there are two notable exceptions for the 'optimality tolerance' and the 'minimum trust region radius'. These two settings have a large influence on the semi-major axis increase found, as well as on the runtime of the DDP algorithm.

To find truly globally optimal solutions, a global optimisation heuristic, MBH, is used with DDP in the inner loop of the algorithm. However, MBH did not significantly improve on the trajectory found by DDP, although it did come at the cost of a considerable increase in runtime.

The DDP algorithm is used on a variety of test cases, to find the optimal solar-sail attitude to maximise the specific orbital energy (which is equivalent to increasing the orbit semi-major axis). As a next test case, NASA's upcoming ACS3 mission is investigated. The sail's attitude is again optimised for the specific orbital energy, where now also eclipse conditions are taken into account. The ACS3 sailcraft is expected to be launched in a Sun-synchronous orbit. When the

semi-major axis of the orbit changes without also changing the inclination, the Sun-synchronous attribute will be lost. Therefore, a constraint is implemented to simultaneously change the inclination of the orbit such that the orbit remains Sun-synchronous. Significant increases in the semi-major axis of the orbit are obtained while the sailcraft is kept under Sun-synchronous conditions.

Additionally, the effect of considering different perturbing accelerations acting on the sailcraft is investigated for both the geostationary case and the ACS3 mission. Previously, the only accelerations acting on the satellite were point-mass gravity from the Earth and the solar-sail acceleration itself. Now, the effect of the irregularities of the Earth's gravity field ( $J_2, J_3$  and  $J_4$ ), lunar and solar gravity are assessed. Including any of the perturbing accelerations will lead to a smaller semi-major axis increase for both the geostationary case and the ACS3 mission, while the effect on the runtime of the DDP algorithm is negligible. The magnitude of the change in semi-major axis is problem dependent, for the geostationary case, lunar gravity has the largest effect on the semi-major axis, whereas for the ACS3 mission this is the  $J_2$  effect.

Lastly, DDP has proven to be robust for many-revolution solar-sail trajectories around the Earth. With a solar-sail lightness number  $\beta = 0.0011$ , a semi-major axis increase of 3793.10 km was found after 500 orbital revolutions for a sailcraft initially in geostationary orbit. These results were obtained while considering various perturbing accelerations acting on the sailcraft ( $J_2, J_3$  and  $J_4$ , lunar and solar gravity). Also eclipses with the Earth as occulting body were taken into account. The increase in semi-major axis is very large, especially considering that a solar-sail lightness number  $\beta = 0.0011$  was used, which is an order of magnitude smaller than the current state-of-the-art. From this result it can be concluded that that a solar-sail is a viable form of propulsion for orbit raising at geostationary altitude. Another many-revolution trajectory optimisation case was investigated in the form of NASA's upcoming ACS3 mission. Here, with a solar-sail lightness number  $\beta = 0.0084$ , a semi-major axis increase of 7.85 km was found after 100 orbital revolutions. Again, various perturbing accelerations acting on the sailcraft ( $J_2, J_3$  and  $J_4$ , lunar and solar gravity) and eclipses were taken into account. In addition, a constraint was enforced to keep the orbit under Sun-synchronous conditions. The semi-major axis increase of 7.85 km is significant considering that the initial altitude of the orbit is only 700 km and that the solar sail is simultaneously used to change the orbit inclination to keep the sailcraft under Sun-synchronous conditions. Therefore, it can be concluded that that a solar-sail is a viable form of propulsion for orbit raising in low-Earth orbit with the note that the effect of atmospheric drag has not been considered here.

To conclude, DDP has proven to be a robust optimisation algorithm that can be used for many-revolution Earth-centred trajectory optimisation. The algorithm performs well for poor initial guesses and is able to produce optimal trajectories while taking different perturbing accelerations and terminal constraints into account.



# Table of contents

Nomenclature	ix
List of figures	xi
List of tables	xi
1 Introduction	1
1.1 Research objective . . . . .	2
1.2 Research questions . . . . .	2
1.3 Report outline . . . . .	2
2 Journal article	3
3 Conclusions and recommendations	37
3.1 Conclusions. . . . .	37
3.2 Recommendations. . . . .	39
A Verification and validation	41
A.1 Dynamical model . . . . .	41
A.1.1 Perturbing accelerations . . . . .	41
A.1.2 Solar-sail acceleration . . . . .	42
A.1.3 Eclipses. . . . .	42
A.2 Numerical integration . . . . .	43
A.2.1 Analytical function . . . . .	43
A.2.2 Verification with TUDAT . . . . .	44
A.2.3 Validation with TUDAT . . . . .	44
A.3 Optimisation . . . . .	45
A.3.1 Two-dimensional geostationary orbit: three orbital revolutions . . . . .	45
A.3.2 Two-dimensional polar orbit: three orbital revolutions . . . . .	46
References	46



# Nomenclature

## Abbreviations and Acronyms

AB	Adams-Bashforth
ACS3	Advanced Composite Solar Sail System
AIAA	American Institute of Aeronautics and Astronautics
AU	Astronomical Unit
BS	Bulirsch-Stoer
DDP	Differential Dynamic Programming
ECI	Earth-centred inertial
GEO	Geostationary Equatorial Orbit
GTO	Geostationary Transfer Orbit
MBH	Monotonic Basin Hopping
MEE	Modified Equinoctial Elements
PM	Point mass
RK	Runge-Kutta
SEP	Solar-Electric Propulsion
STM	State Transition Matrix
TRQP	Trust Region Quadratic Subproblem

## Greek Symbols

$\alpha$	Solar-sail cone angle
$\beta$	Solar-sail lightness number
$\delta$	Solar-sail clock angle
$\nu$	True anomaly
$\Omega$	Right ascension of the ascending node
$\omega$	Argument of periapsis

## Latin Symbols

$a$	Semi-major axis
$e$	Eccentricity
$i$	Inclination



## List of figures

A.1	Magnitude of perturbing accelerations at different altitudes. . . . .	41
A.2	Polar plots of solar-sail performance at 1 AU. . . . .	42
A.3	Two-dimensional geostationary orbit with eclipse: one orbital revolutions. . . . .	42
A.4	The sunlight fraction, modelled with a Heaviside step function, for different sharpness and transition coefficients. . . . .	43
A.5	Position difference of the satellite. . . . .	44
A.6	Two-dimensional geostationary orbit: DDP verification. . . . .	45
A.7	Two-dimensional polar orbit: DDP verification. . . . .	46

## List of tables

A.1	Initial conditions for satellite propagation testing. . . . .	44
-----	---	----



# Introduction

In the year 1610, Johannes Kepler noticed that solar radiation carries a form of pressure based on the fact that the tail of a comet always points away from the Sun. He wrote in a correspondence with Galileo: "... provide ships or sails adapted to the heavenly breezes, and there will be some who will brave that void." [1]. The idea of using solar radiation pressure as propulsion for a spacecraft was first conceived in 1924 by Fridrickh Tsander, who wrote about the usage of a solar sail to provide a spacecraft with a form of propulsion that provides a continuous acceleration, limited only by the lifetime of the sail itself [2]. The following years the idea was not given much attention, but in the fifties it was reinvented by Carl Wiley. Following his enthusiasm, more detailed studies on the subject were developed in the fifties and sixties, until eventually in the seventies the first solar sail was considered for a mission by NASA [3]. Eventually, the mission was cancelled and although the solar sail was not used, interest in the concept grew. Recently, in 2010, the first ever solar sail flew on a mission by the Japanese space agency, JAXA. This mission, IKAROS, was a mission to demonstrate the technology of a solar sail on an interplanetary trajectory [4]. Following IKAROS, more solar-sail missions have flown. For example, the Planetary Society launched a solar-sail mission, Lightsail 1, which completed an in-orbit deployment of the sail. The follow-up mission, Lightsail 2, successfully demonstrated that the sail could be used for orbit raising [5]. The upcoming NEA Scout mission by NASA, which is scheduled to be launched in November 2021, will be the first mission with a scientific objective that will use a solar sail as main form of propulsion. NEA Scout will perform reconnaissance with near-Earth asteroids [6]. In addition to NEA Scout, NASA has another solar-sail mission planned, the Advanced Composite Solar Sail System (ACS3) mission. The mission's main focus is to test NASA's novel deployable composite boom technology [7].

Since the acceleration that the solar sail imposes on the spacecraft is very small, it may take a long time for an orbital transfer to be completed. Therefore, it is desirable to find the optimal attitude of the sail such that an orbital transfer can be completed as efficiently as possible. Global time-optimal solutions have been found for *interplanetary* solar-sail trajectories with the use of optimisation algorithms such as evolutionary neurocontrol [8]. However, global optima of *planet-centred* trajectories have not yet been found. Due to the small solar-sail acceleration, many revolutions around the central body are often necessary before a desired state is reached. This makes solar-sail planet-centred trajectory optimisation more complex, as it leads to a sensitive and high-dimensional problem [9]. Historically, optimisation of these trajectories has been performed using (semi-)analytical techniques, resulting in locally optimal solutions. For example, analytical control laws have been developed that maximise the rate of change of a single orbital element, such as the semi-major axis or the inclination [3]. However, when more than one orbital element is to be changed, the problem becomes more complicated. Solutions have been proposed where the trajectory is split into different arcs, where in each arc the rate of change of a different orbital element is maximised [10]. Other solutions implement blended control laws, where individual control laws are combined with weights, and the magnitude of the weights is based on the relative importance of changing each orbital element during that particular phase of the trajectory [11]. In terms of applications, Earth-centred solar-sail trajectory optimisation has been performed for a variety of problems. For example, the use of a solar sail to mitigate space debris in the geostationary ring has been studied [12, 13]. Solar sailing is particularly interesting for these applications, since the cycle to move debris can be repeated indefinitely because the sailcraft will not run out of propellant. Other applications include, but are not limited to, Earth escape trajectories [14, 15, 16], to provide a starting point for interplanetary trajectories.

Differential dynamic programming (DDP) is an optimisation routine that has the benefit that it scales linearly with the number of control variables [17], which is a property that makes DDP particularly interesting for large optimisation problems such as mentioned in the previous paragraph. DDP splits the complete trajectory into small stages, where at each stage a local optimal control problem is solved [18]. DDP has already been used to find optimal satellite trajectories using another form of low-trust propulsion, solar-electric propulsion (SEP), to near-Earth asteroids [19] and for NASA's Dawn mission to Ceres and Vesta [20, 21]. Moreover, DDP's ability to deal with many-revolution Earth-centred trajectories

has been demonstrated with a solar-electric propulsion (SEP) transfer trajectory from geostationary transfer orbit (GTO) to geostationary equatorial orbit (GEO) in 2000 revolutions [22].

The remainder of this chapter focuses on the research objective and questions, which are elaborated upon in Section 1.1 and 1.2, respectively. The outline of this report is stated in Section 1.3.

## 1.1. Research objective

Considering the lack of globally optimal solutions for many-revolution Earth-centred solar-sail trajectories and the ability of DDP to provide optimal solutions for many-revolution low-thrust trajectories, this work aims to obtain, for the first time, globally optimal Earth-centred solar-sail trajectories through the use of DDP in combination with a global optimisation heuristic, monotonic basin hopping (MBH). MBH is needed to find globally optimal solutions since DDP itself is a local optimisation scheme. This work aims to extend the applicability of DDP to many-revolution Earth-centred solar-sail trajectories and prove its superior performance in terms of optimal solution found compared to known locally optimal solutions. For a variety of test cases the optimal solar-sail attitude will be found to maximise the specific orbital energy. Test cases range from theoretical scenarios to realistic mission applications, including NASA's upcoming ACS3 mission under Sun-synchronous conditions.

## 1.2. Research questions

Section 1.1 has shown that this work's research objective is focused on obtaining globally optimal Earth-centred solar-sail trajectories through the use of DDP. The developed algorithm will be used on different test cases. The following research questions have been developed to achieve the research objective:

1. Does solar-sail propulsion impose any constraints or limitations that prevent the use of DDP?
2. What is the effect of different settings for the optimisation algorithm on the optimality of the solution?
3. How do globally optimal Earth-centred solar-sail trajectories, when optimised using DDP, perform compared to locally optimal steering laws?
4. What is the optimal solar-sail attitude for planet-centred solar-sail trajectories under different dynamical conditions around the Earth?
  - (a) What is the optimal solar-sail attitude for orbit raising at geostationary altitude?
  - (b) What is the optimal solar-sail attitude for raising the orbit altitude of a low-Earth orbit under Sun-synchronous conditions?
  - (c) What is the effect of perturbing accelerations on the optimal trajectories?

## 1.3. Report outline

The main part of this thesis is written in the form of a conference article, which has already been accepted to be presented at the 2022 American Institute of Aeronautics and Astronautics (AIAA) SciTech Forum and Exposition. The article is titled:

"Many-revolution Earth-centred solar-sail trajectory optimisation using differential dynamic programming"

The article manuscript is presented in Chapter 2. In Chapter 3, the conclusions of this work in light of the research questions posed in Section 1.2 are given, together with recommendations for future work. The methods that have been used for verification and validation of the software that has been developed are presented in Appendix A.



# 2

Journal article

# Many-revolution Earth-centred solar-sail trajectory optimisation using differential dynamic programming

G. Leemans\*, L. Carzana† and J. Heiligers‡

*Delft University of Technology, 2600 AA, Delft, The Netherlands*

**This work demonstrates the usability of differential dynamic programming (DDP) to obtain, for the first time, globally optimal Earth-centred solar-sail trajectories. To this end, DDP is combined with a global optimisation heuristic, monotonic basin hopping. The dynamical model is implemented as a two-body problem, augmented with an ideal solar-sail reflectance model and accounts for eclipses. The numerical performance of the optimisation algorithm is enhanced by integrating the sailcraft state in modified equinoctial elements and performing a Sundman transformation to change the independent variable from time to the true anomaly. The DDP algorithm is proven to be robust for trajectories extending up to 500 revolutions and, compared to known locally optimal steering laws, allows to obtain more or equally optimal solutions. The latter is demonstrated in this paper through a set of test cases that range from theoretical scenarios to realistic mission applications, including increasing the specific orbital energy of NASA’s upcoming ACS3 mission. Additionally, the algorithm’s ability to cope with different optimisation settings, perturbing accelerations and constraints is demonstrated.**

## Nomenclature

$\alpha$	=	Sail cone angle [rad]	$e$	=	Eccentricity
$\beta$	=	Solar sail lightness number	$F$	=	Dynamics matrix
$\gamma$	=	Smoothed sunlight fraction	$f, g, h, \tilde{k}$	=	Modified equinoctial elements
$\Delta$	=	Trust region radius	$G$	=	Scaling matrix
$\delta$	=	Sail clock angle [rad]	$H$	=	Heaviside function
$\epsilon_1$	=	Reduction ratio tolerance	$\tilde{h}$	=	Logistic growth rate
$\epsilon_2$	=	Optimality tolerance	$k$	=	Stage number
$\eta$	=	Sundman transformation variable	$i$	=	Inclination [rad]
$\kappa$	=	Reduction ratio tolerance	$J$	=	Cost function
$\Lambda$	=	Sundman-transformed dynamics matrix	$l$	=	True longitude [rad]
$\lambda$	=	Lagrange multiplier	$M$	=	Scaling matrix
$\mu$	=	Gravitational parameter [m <sup>3</sup> /s <sup>2</sup> ]	$m$	=	Mass [kg]

---

\*Graduate student, Faculty of Aerospace Engineering, g.leemans@student.tudelft.nl

†PhD candidate, Faculty of Aerospace Engineering, l.carzana@tudelft.nl

‡Assistant Professor, Faculty of Aerospace Engineering, m.j.heiligers@tudelft.nl

$\nu$	=	True anomaly [rad]	$N$	=	Number of stages
$\rho$	=	Reduction ratio	$\mathbf{n}$	=	Normal vector
$\Sigma$	=	Penalty matrix	$p$	=	Semi-latus rectum [m]
$\Phi$	=	State transition matrix	$\mathbf{p}$	=	Reference vector
$\phi$	=	Objective function	$q$	=	Auxiliary modified equinoctial element
$\psi$	=	Terminal constraints	$R$	=	Radius [m]
$\Omega$	=	Right ascension of the ascending node [rad]	$\mathbf{r}$	=	Position vector [m]
$\omega$	=	Argument of periapsis [rad]	$\mathbf{r}_s$	=	Sun-sail vector [m]
$A$	=	Feed-forward control matrix	$T_{\text{es}}$	=	Sidereal day [s]
$a$	=	Semi-major axis [m]	$t$	=	Time [s]
$B$	=	Feedback control matrix	$\mathbf{u}$	=	Control vector
$C$	=	Sail-centred reference frame	$u$	=	Control parameter
$D$	=	Feedback control matrix	$\mathbf{X}$	=	Augmented state vector

## I. Introduction

Solar sailing is a form of low-thrust space propulsion that uses a large, thin, mirror-like structure to produce thrust by reflecting solar photons. The thrusting capabilities of a solar-sail are therefore only limited by the lifetime of the sail itself; there is no dependency on available propellant. Hence, solar sailing is particularly interesting for missions of long duration or missions requiring continuous thrust. Examples include the use of a solar sail to mitigate space debris in the geostationary ring [1, 2], Earth escape trajectories [3–5], and non-Keplerian orbits [6, 7]. The concept of solar sailing has been around for a long time [8], but has only recently become a reality with the successful deployment of IKAROS, a mission by JAXA [9]. Following IKAROS, several more solar-sail missions have flown and various missions are planned for the future. Next year, the launch of NASA’s Advanced Composite Solar Sail System (ACS3) mission is scheduled. The mission’s main focus is to test NASA’s novel deployable composite boom technology [10].

Since the acceleration that the solar sail imposes on the spacecraft is very small, solar-sail trajectories are often optimised for the time of flight. Global time-optimal solutions have been found for *interplanetary* solar-sail trajectories with the use of optimisation algorithms such as evolutionary neurocontrol [11]. However, global optima of *planet-centred* trajectories have not yet been found. Due to the small solar-sail acceleration, many revolutions around the central body are often necessary before the desired state is reached. This makes solar-sail planet-centred trajectory optimisation more complex, as it leads to a sensitive and high-dimensional problem [12]. Optimisation of these trajectories has been performed using (semi-)analytical techniques, resulting in locally optimal solutions. For example, control laws have been developed that maximise the rate of change of a single orbital element, such as the semi-major axis or the inclination [13]. However, when more than one orbital element is to be changed, the problem becomes more complicated. Solutions have been proposed where the trajectory is split into different arcs, where in each arc the rate of change of a different orbital element is maximised [14]. Other solutions implement a blended control law, where individual control laws are combined with weights, where the magnitude of the weights is based on the relative importance of changing

each orbital element during that particular phase of the trajectory [15].

An optimisation method that has never been explored for planet-centred solar-sail trajectory optimisation is differential dynamic programming (DDP). DDP is a second-order gradient-based optimisation routine that has the benefit that it scales linearly with the number of control variables [16], which is a property that makes DDP interesting for large optimisation problems. Although DDP is suitable for large scale optimisation, runtime is still expected to increase with the size of the problem. This problem is diminished with a method to parallelise the most computationally expensive step of the algorithm, which has a significant impact on the runtime of the algorithm [17, 18]. DDP discretises the spacecraft's trajectory into stages, where at each stage a local, quadratic model of the cost is minimised [19]. DDP has already been successfully used to find optimal Earth-centred trajectories using solar electric propulsion (SEP) [12], where optimal solutions were found for a transfer from geostationary transfer orbit (GTO) to geostationary equatorial orbit (GEO) consisting of up to 2000 revolutions.

This work aims to obtain, for the first time, globally optimal Earth-centred solar-sail trajectories through the use of DDP in combination with a global optimisation heuristic, monotonic basin hopping (MBH). MBH is needed to find globally optimal solutions since DDP itself is a local optimisation scheme. MBH is able to find the global optimum by iteratively executing the DDP algorithm with a slightly perturbed initial guess [20]. As such, this work will extend the applicability of DDP to many-revolution Earth-centred solar-sail trajectories and proof its superior performance in terms of optimal solution found compared to known locally optimal solutions. For a variety of test cases, the optimal solar-sail attitude will be found to maximise the specific orbital energy. Test cases range from theoretical scenarios to realistic mission applications, including NASA's upcoming ACS3 mission under Sun-synchronous conditions.

The remainder of the paper is structured as follows. First, the dynamical model is discussed in Section II, including the solar-sail model and a range of perturbing accelerations. Subsequently, an overview of the theoretical background of DDP and its implementation is given in Section III. Next, in Section IV, the results are presented from validation (Section IV.A), varying the algorithm's settings (Section IV.B.1), different perturbing accelerations (Section IV.B.2), the algorithm's ability to deal with terminal constraints (Section IV.C), and lastly, many-revolution trajectory optimisation (Sections IV.D and IV.E). The article closes with the conclusions in Section V.

## II. Dynamical model

The dynamics of the satellite are defined in an Earth-centred equatorial inertial reference frame, ECI( $x, y, z$ ) with the  $x$ -axis towards the vernal equinox, the  $z$ -axis perpendicular to the equatorial plane and the  $y$ -axis completing the right-handed reference frame. DDP requires the spacecraft's state to be represented as an augmented state vector [17], where the spacecraft state vector is augmented with the control vector as

$$\mathbf{X} = \begin{bmatrix} \mathbf{x} & \mathbf{u} \end{bmatrix}^T = \begin{bmatrix} \mathbf{r} & \dot{\mathbf{r}} & \mathbf{u} \end{bmatrix}^T = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & m & t & \alpha & \delta \end{bmatrix}^T \quad (1)$$

In Eq. 1  $\mathbf{r}$  is the position vector of the satellite,  $\mathbf{u}$  is the control vector consisting of the two sail control parameters,  $\alpha$  and  $\beta$ , which will be elaborated upon in Section II.B,  $\dot{\phantom{x}}$  represents the derivative with respect to time,  $m$  is the spacecraft mass and  $t$  is the current time. The derivative of the augmented state vector is composed of various terms:

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{r}} \\ \ddot{\mathbf{r}}_{\oplus} + \ddot{\mathbf{r}}_{SS} + \ddot{\mathbf{r}}_{J_2} + \ddot{\mathbf{r}}_{J_3} + \ddot{\mathbf{r}}_{J_4} + \ddot{\mathbf{r}}_{\zeta} + \ddot{\mathbf{r}}_{\odot} \\ 0 \\ 1 \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} \quad (2)$$

The terms for the acceleration in the second row of the vector in Eq. 2 are, from left to right, point-mass gravity from the Earth, the solar-sail acceleration, accelerations resulting from irregularities in the Earth's gravity field ( $J_2$ ,  $J_3$  and  $J_4$ ), lunar gravity and solar gravity. These individual contributions will be expanded on the coming sections. Unless stated otherwise, J2000 is used as the initial epoch for propagation. An important perturbing acceleration missing in Eq. 2 is the atmospheric drag, whose effect can be significant in low-altitude regimes. However, DDP optimisation requires the dynamical model to be twice differentiable, which becomes a difficulty when atmospheric drag is included. Density models most often consist of tabulated data, which are not differentiable [21]. The exponential density model is (twice) differentiable, but this model is strongly simplified and a very rough estimate [21]. The implementation of an accurate and twice differentiable drag model is therefore left for future research.

### A. Two-body dynamics

First and foremost, the largest acceleration acting on the satellite is the point-mass gravity from the Earth [21]. This acceleration can be written as

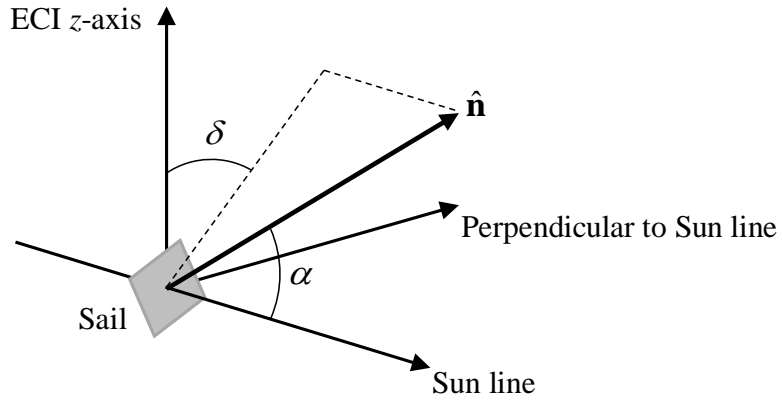
$$\ddot{\mathbf{r}}_{\oplus} = -\frac{\mu_{\oplus}}{r^3} \mathbf{r} \quad (3)$$

where  $\mu_{\oplus}$  is the Earth's gravitational parameter and  $r = \|\mathbf{r}\|$ . Table 2 provides the values of the used dynamical parameters.

### B. Solar-sail acceleration

The solar-sail acceleration is modelled by assuming an ideal solar-sail reflectance model, which implies that all incoming radiation is specularly reflected and the acceleration acts along the vector normal to the sail surface,  $\hat{\mathbf{n}}$ . Other solar-sail force models account for absorption, diffuse reflection and thermal emission of the incoming radiation (optical force model), sail billowing and degradation [13, 22, 23]. The difference between using the ideal and optical force model has been demonstrated in Reference [22], where the transfer time for both models was evaluated for an Earth-Mars transfer. The results show an increase of 12 – 13% in time of flight when using the optical model compared to the ideal model. For the case of a transfer from the GEO to halo orbits around the Sun-Earth  $L_1$  point, an increase in time of flight of approximately 8% was found [24]. The optical force model is not considered in this work, but similar orders of magnitude for the increase in time of flight may be expected when doing so in future work.

The direction of the sail normal vector,  $\hat{n}$ , is defined in frame  $C(\hat{r}_s, \hat{p}, \hat{z})$ , see Figure 1. Reference frame  $C$  has its origin at the position of the sailcraft, the  $\hat{r}_s$  axis points in the direction from the Sun towards the satellite, the  $\hat{z}$  axis in the direction of the ECI  $z$ -axis, and the third axis,  $\hat{p}$ , completes the right-handed orthogonal reference frame. The sail normal vector is defined by two angles, see again Figure 1: the cone and clock angles, where the cone angle,  $\alpha$ , is defined as the angle between the sail normal and the direction of sunlight. The clock angle,  $\delta$ , is defined as the angle between the  $z$ -axis of the ECI reference frame and the projection of the normal vector onto the plane perpendicular to the Sun-line.



**Fig. 1 Solar-sail attitude angles.**

The normal vector of the sail,  $\hat{n}$ , is defined reference frame  $C$  as

$$\mathbf{n}|_C = \begin{bmatrix} \cos \alpha \\ \sin \alpha \cos \delta \\ \sin \alpha \sin \delta \end{bmatrix} \quad (4)$$

s.t.  $\alpha \in [-\pi/2, \pi/2]$   
 $\delta \in [-2\pi, 2\pi]$

An intrinsic property of the solar sail is that the solar-sail acceleration is always directed away from the Sun. To model this property, the cone angle is restricted to a range between  $-\pi/2$  and  $\pi/2$ . The clock angle can in principle obtain any value, but is for practical reasons restricted between  $-2\pi$  and  $2\pi$ . For inclusion in Eq. 2, the solar-sail acceleration is to be found in the ECI reference frame, and therefore the normal vector needs to be transformed from reference frame  $C$  to the ECI frame according to:

$$\mathbf{n} = \begin{bmatrix} \hat{r}_s & \hat{z} & \hat{z} \times \hat{r}_s \end{bmatrix} \mathbf{n}|_C \quad (5)$$

The resulting solar-sail acceleration is determined as [13]

$$\ddot{\mathbf{r}}_{SS} = \gamma \beta \frac{\mu_{\odot}}{r_s^2} (\hat{\mathbf{r}}_s \cdot \mathbf{n})^2 \mathbf{n} \quad (6)$$

where  $\gamma$  is the smoothed sunlight fraction,  $\beta$  is the solar-sail lightness number,  $\mu_{\odot}$  the Sun's gravitational parameter and  $\mathbf{r}_s$  is the position vector from the Sun towards the sailcraft. The solar-sail lightness number is defined as the ratio of the solar radiation pressure to the solar gravitational acceleration and is used as a metric for the thrusting capabilities of the sail. Solar-sail lightness numbers from a selection of previously flown and planned solar-sail missions are shown in Table 1. During eclipse conditions, the sail does not produce an acceleration. Eclipses are modelled using a low-precision solar ephemeris model [21] and a Heaviside-based sunlight fraction approach [25]. The sunlight fraction is equal to zero when the satellite is in eclipse, and to one when the satellite is in sunlight. Only eclipses with the Earth as the occulting body are taken into account.

**Table 1 Performance parameters from previously flown and planned solar-sail missions [10, 26].**

	IKAROS	NanoSail-D2	InflateSail	LightSail 2	NEA Scout (planned)	ACS3 (planned)
Sail area [m <sup>2</sup> ]	196	10	10	32	86	80
Sailcraft mass [kg]	307	4.0	3.2	5.1	12.0	14.6
Lightness number	$9.8 \times 10^{-4}$	$3.8 \times 10^{-3}$	$4.8 \times 10^{-3}$	$9.8 \times 10^{-3}$	$1.1 \times 10^{-2}$	$8.4 \times 10^{-3}$ *

\* W.K. Wilkie, personal communication, May 2021

### C. Non-spherical Earth

In reality, the Earth is not a perfect sphere. Therefore, the gravitational acceleration from the Earth cannot solely be described by a point-mass gravity model, Zonal harmonics coefficients,  $J_i$  with  $i = 2, 3, \dots$ , can be used to model the irregularities of the gravity field. The perturbing acceleration due to the  $J_2$ ,  $J_3$  and  $J_4$  coefficients are [27]

$$\ddot{\mathbf{r}}_{J_2} = -\frac{3\mu_{\oplus}J_2R_{\oplus}^2}{2r^5} \begin{bmatrix} x \left(1 - 5\frac{z^2}{r^2}\right) & y \left(1 - 5\frac{z^2}{r^2}\right) & z \left(3 - 5\frac{z^2}{r^2}\right) \end{bmatrix}^T \quad (7a)$$

$$\ddot{\mathbf{r}}_{J_3} = -\frac{15\mu_{\oplus}J_3R_{\oplus}^3}{2r^7} \begin{bmatrix} xz \left(1 - \frac{3z^2}{r^2}\right) & yz \left(1 - \frac{3z^2}{r^2}\right) & z^2 \left(2 - \frac{r^2}{5z^2} - \frac{3z^2}{r^2}\right) \end{bmatrix}^T \quad (7b)$$

$$\ddot{\mathbf{r}}_{J_4} = -\frac{35\mu_{\oplus}J_4R_{\oplus}^4}{2r^9} \begin{bmatrix} xz^2 \left(\frac{3}{2} - \frac{9z^2}{4r^2} - \frac{3r^2}{28z^2}\right) & yz^2 \left(\frac{3}{2} - \frac{9z^2}{4r^2} - \frac{3r^2}{28z^2}\right) & z^3 \left(\frac{5}{2} - \frac{9}{4}\frac{z^2}{r^2} - \frac{15}{28}\frac{r^2}{z^2}\right) \end{bmatrix}^T \quad (7c)$$

where  $R_{\oplus}$  is the Earth's radius. Values for the zonal harmonics coefficients are given in Table 2.

#### D. Third-body Perturbations

Point-mass gravity from the Moon and the Sun are the largest third-body accelerations acting on an Earth-orbiting spacecraft. Therefore, the third-body perturbations from the Moon and the Sun are accounted for and modelled using the method of Section 3.3.1 from Reference [21]. These accelerations cannot be described in the same way as is done for the point-mass gravity from the Earth, since the perturbations by the Sun and Moon cause an acceleration on the Earth itself as well. This acceleration has to be subtracted from the acceleration induced on the satellite. The position vector from the Moon to the satellite is written as

$$\mathbf{r}_{\mathbb{C}/SC} = \mathbf{r} - \mathbf{r}_{\mathbb{C}} \quad (8)$$

where  $\mathbf{r}_{\mathbb{C}}$  is the position vector of the Moon. The contribution of the acceleration caused by the Moon on the spacecraft can then be defined as

$$\ddot{\mathbf{r}}_{\mathbb{C}} = -\mu_{\mathbb{C}} \begin{bmatrix} \frac{x-x_{\mathbb{C}}}{\|\mathbf{r}-\mathbf{r}_{\mathbb{C}}\|^3} + \frac{x_{\mathbb{C}}}{x_{\mathbb{C}}^3} & \frac{y-y_{\mathbb{C}}}{\|\mathbf{r}-\mathbf{r}_{\mathbb{C}}\|^3} + \frac{y_{\mathbb{C}}}{y_{\mathbb{C}}^3} & \frac{z-z_{\mathbb{C}}}{\|\mathbf{r}-\mathbf{r}_{\mathbb{C}}\|^3} + \frac{z_{\mathbb{C}}}{z_{\mathbb{C}}^3} \end{bmatrix}^T \quad (9)$$

where  $\mu_{\mathbb{C}}$  is the gravitational parameter of the Moon, of which the value is given in Table 2. The same strategy is used for determining the acceleration on the satellite caused by the Sun. The position vector from the Sun to the satellite is written as

$$\mathbf{r}_{\odot/SC} = \mathbf{r} - \mathbf{r}_{\odot} \quad (10)$$

where  $\mathbf{r}_{\odot}$  is the position vector of the Sun. The contribution of the acceleration caused by the Sun on the spacecraft can then be defined as

$$\ddot{\mathbf{r}}_{\odot} = -\mu_{\odot} \begin{bmatrix} \frac{x-x_{\odot}}{\|\mathbf{r}-\mathbf{r}_{\odot}\|^3} + \frac{x_{\odot}}{x_{\odot}^3} & \frac{y-y_{\odot}}{\|\mathbf{r}-\mathbf{r}_{\odot}\|^3} + \frac{y_{\odot}}{y_{\odot}^3} & \frac{z-z_{\odot}}{\|\mathbf{r}-\mathbf{r}_{\odot}\|^3} + \frac{z_{\odot}}{z_{\odot}^3} \end{bmatrix}^T \quad (11)$$

To determine these perturbing accelerations, it is necessary to know the Moon's and the Sun's position in the ECI frame at each point in time. For obtaining these ephemeris data, analytical expressions for low-precision solar and lunar coordinates are used from Section 3.3.1 in Reference [21]. The model to obtain these coordinates assumes unperturbed motion of the Earth around the Sun, whereas for the motion of the Moon around the Earth the influence of terrestrial and solar gravity are taken into account.



**Table 2** Dynamical model parameters [27, 28].

Parameter	Value	Parameter	Value
$\mu_{\oplus}$	$3.986004 \times 10^{14} \text{ m}^3/\text{s}^2$	$J_2$	$1.082626 \times 10^{-3}$
$\mu_{\odot}$	$1.327124 \times 10^{20} \text{ m}^3/\text{s}^2$	$J_3$	$-2.533 \times 10^{-6}$
$\mu_{\zeta}$	$4.902801 \times 10^{12} \text{ m}^3/\text{s}^2$	$J_4$	$-1.6186 \times 10^{-6}$
$R_{\oplus}$	63781363 m		

### E. Modified equinoctial elements

The equations of motion (EOMs) are integrated in modified equinoctial elements (MEE). Integrating a set of orbit elements has benefits to integrating a Cartesian state, since the number of rapidly changing elements can be reduced. A formulation of the EOMs that only has one single rapidly changing element, like the Keplerian model, but does not experience the singularities that are present in the Keplerian model, are the MEE [29]. Five variables describe the shape and orientation of the orbit, while one variable describes the location of the spacecraft within the orbit. This latter variable is chosen as the true longitude. The MEE are defined as

$$\left\{ \begin{array}{l} p = a(1 - e^2) \\ f = e \cos(\omega + \Omega) \\ g = e \sin(\omega + \Omega) \\ h = \tan\left(\frac{i}{2}\right) \cos(\Omega) \\ \tilde{k} = \tan\left(\frac{i}{2}\right) \sin(\Omega) \\ l = \omega + \Omega + \nu \end{array} \right. \quad (12)$$

where  $p$  is the semi-latus rectum,  $f$  and  $g$  describe the orbit eccentricity, whereas  $h$  and  $\tilde{k}$  describe the orbit inclination and  $l$  is the true longitude. The MEE are defined using Kepler elements,  $a$ ,  $i$ ,  $e$ ,  $\omega$ ,  $\Omega$  and  $\nu$ , which are the semi-major axis, inclination, eccentricity, argument of periapsis, right ascension of the ascending node and the true anomaly, respectively. The augmented state vector in MEE form becomes

$$\mathbf{X}_{\text{MEE}} = \begin{bmatrix} p & f & g & h & \tilde{k} & l & m & t & \alpha & \delta \end{bmatrix}^T \quad (13)$$

Integrating the dynamics in MEE requires the derivative of the augmented state vector first to be evaluated in Cartesian elements (see Eq. 2), after which the derivative can be transformed to MEE by the method in Section 2.1 from Reference [30].

### F. Sundman transformation

Using MEE has proven to be even more advantageous when using a Sundman transformation to change the independent variable for integration from time to the true anomaly,  $\nu$  [12]. This Sundman transformation is given as

$$dt = \frac{1}{\sqrt{\mu_{\oplus} p}} \left( \frac{p}{q} \right)^2 d\nu \quad (14)$$

where  $q$  is an auxiliary MEE, which is defined as

$$q = 1 + f \cos(l) + g \sin(l) \quad (15)$$

The derivative of the augmented state vector is then transformed as

$$\dot{\mathbf{X}}_{\text{MEE}} = \frac{\partial \mathbf{X}_{\text{MEE}}}{\partial \nu} = \frac{\partial \mathbf{X}_{\text{MEE}}}{\partial t} \frac{\partial t}{\partial \nu} = \dot{\mathbf{X}}_{\text{MEE}} \frac{\partial t}{\partial \nu} \quad (16)$$

where  $\partial t / \partial \nu$  is obtained from Eq. 14. Note the different dot above the augmented state vector ( $\dot{\mathbf{X}}$  versus  $\dot{\mathbf{X}}$ ), where  $\dot{\mathbf{X}}$  indicates the Sundman-transformed derivative.

## III. Differential dynamic programming

This section provides the definition of the DDP method and details on its implementation. The algorithm provided here is based on the DDP algorithm in Reference [17], for the particular case of a single-phase trajectory without local stage costs.

Like any low-thrust trajectory optimisation algorithm, DDP seeks the minimum of a cost function  $J(\mathbf{x}, \mathbf{u}, t)$ , where  $\mathbf{x}(t)$  is a state trajectory and  $\mathbf{u}(t)$  is a control schedule. The general workflow of DDP is visualised in Figure 2. First, the spacecraft's trajectory is forward integrated with a given set of controls. Then, the trajectory is discretised into stages, where at each stage a local, quadratic model of the cost is minimised. The method of Dynamic Programming is based on Bellman's principle of optimality: "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision." [31]. Following this principle, the algorithm finds the control variables for the final stage that correspond to the best final solution. When these control variables are found, the algorithm repeats this task for the penultimate stage, until it has optimised all individual stages. This process is called the backward sweep, see Figure 2. With the newly found control variables, the problem is forward integrated (the forward sweep, see again Figure 2), after which the same procedure is initiated until the optimisation algorithm has converged. The remainder of this section will elaborate on the different steps taken by DDP, but first the notation is elaborated upon in the next section.

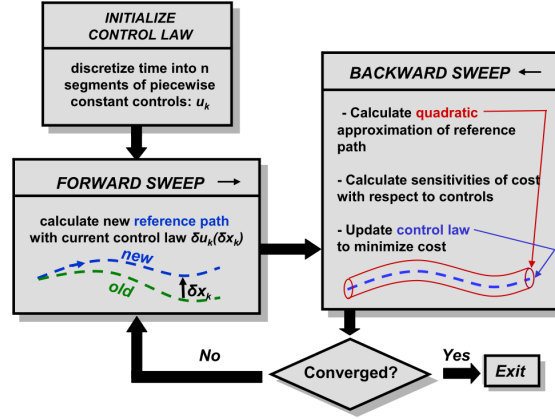


Fig. 2 DDP optimisation schematic [17].

### A. Notation

The DDP algorithm consists of a number of equations in which scalars, vectors, matrices and third-order tensors make their appearance. This section covers the notation that is used throughout the upcoming sections. Similar as in Section II, boldface is used to indicate vectors, whereas superscripts are used to indicate the entries in a vector, matrix or tensor. For example, consider the augmented state vector in MEE (note that the MEE subscript is dropped for brevity) as

$$\mathbf{X} = \begin{bmatrix} p & f & g & h & \tilde{k} & l & m & t & \alpha & \delta \end{bmatrix}^T \quad (17)$$

The  $i^{\text{th}}$  entry of  $\mathbf{X}$  is written as  $X^i$ , so  $X^3$  is equal to  $g$  when counting starts from one. The same procedure holds for matrices and third-order tensors. For example, consider the following matrix:

$$\mathbf{Y} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad (18)$$

The  $(i, a)^{\text{th}}$  entry of  $\mathbf{Y}$  is written as  $Y^{i,a}$ , so  $Y^{2,3}$  is equal to 6 when counting starts from one.

Subscripts are used for indexing and for indicating the derivative, for example:

$$J_{\mathbf{X}} = \frac{\partial J}{\partial \mathbf{X}} \quad (19)$$

where  $J_{\mathbf{X}}$  is the derivative of the cost,  $J$ , with respect to the augmented state vector,  $\mathbf{X}$ . When the augmented state vector,  $\mathbf{X}$ , has size  $n \times 1$ , then  $J_{\mathbf{X}}$  is also of size  $n \times 1$ . Note that in this case no boldface is used for the subscript. An additional subscript,  $J_{\mathbf{X},k}$ , is used to indicate the value of the gradient evaluated at stage  $k$ . Second-order derivatives are defined similarly as

$$J_{X\lambda} = \frac{\partial^2 J}{\partial X \partial \lambda} \quad (20)$$

where  $J_{X\lambda}$  is the second-order derivative of the cost,  $J$ , with respect to the augmented state vector,  $X$ , and the vector of Lagrange multipliers,  $\lambda$ . When the augmented state vector,  $X$ , has size  $n \times 1$  and the vector containing the Lagrange multipliers,  $\lambda$ , has size  $m \times 1$ , then  $J_{X\lambda}$  is of size  $n \times m$ .

Einstein summation is used to indicate summation over repeated indices. As an example, consider the simple dot product:

$$\mathbf{Y} \cdot \mathbf{Y}^T = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = 1 \times 1 + 2 \times 2 + 3 \times 3 = 14 \quad (21)$$

Instead of writing the dot product, Einstein notation can be used. The product is then written as

$$\mathbf{Y}^{\gamma_1} \mathbf{Y}^{\gamma_1} = \sum_{\gamma_1=1}^3 \mathbf{Y}^{\gamma_1} \times \mathbf{Y}^{\gamma_1} = 1 \times 1 + 2 \times 2 + 3 \times 3 = 14 \quad (22)$$

where  $\gamma_1$  is a dummy index indicating that summation should take place over repeating values of the dummy index. Matrix multiplication can also be written using Einstein notation as

$$\dot{\Phi}^{i,a} = F^{i,\gamma_1} \Phi^{\gamma_1,a} = \sum_{\gamma_1=1}^m F^{i,\gamma_1} \Phi^{\gamma_1,a} \quad (23)$$

where  $\dot{\Phi}$  is the derivative of the state transition matrix,  $F$  is the dynamics matrix (see Section III.D),  $\Phi$  the state transition matrix and  $m$  is the size of the matrix. The dummy index,  $\gamma_1$ , indicates that the  $(i, a)^{\text{th}}$  entry of  $\dot{\Phi}$  is found by summing the products of the  $\gamma_1^{\text{th}}$  entry of the  $i^{\text{th}}$  row of  $F$  with the  $\gamma_1^{\text{th}}$  entry of the  $a^{\text{th}}$  column of  $\Phi$ , from  $\gamma_1 = 1$  to  $m$ . Multiple dummy indices, for example  $\gamma_1$  and  $\gamma_2$ , may be used to indicate a double summation.

## B. Forward sweep

Optimisation starts with the forward sweep, which integrates the trajectory with a given set of controls. For the first iteration, this is the initial guess, while subsequent iterations update the control according to  $\mathbf{u} = \bar{\mathbf{u}} + \delta \mathbf{u}$ , where  $\bar{\mathbf{u}}$  is the control of the previous successful iteration (see Section III.G for how a successful iteration is defined) and  $\delta \mathbf{u}$  is the control update calculated in the backward sweep of the algorithm (see Section III.D). If the resulting trajectory meets the optimality criteria, the procedure is finished and the algorithm is said to have converged. Otherwise, a new control update is calculated in the backward sweep.

The dynamics are integrated using a fixed timestep DOPRI8 integrator, where 100 steps are taken per revolution around the Earth regardless of the shape of the trajectory. State transition matrices (STMs) have to be integrated along the state, which is best carried out using an integrator with a fixed step size [32]. When a variable step size integrator

would be used, the state transition matrices will not account for part of the changes in the trajectory that are induced by the different time steps taken.

### C. Augmented Lagrangian cost function

The cost function, which is to be minimised, is written as

$$J = \phi + \lambda^T \psi + \psi^T \Sigma \psi \quad (24)$$

where  $\phi$  is the objective that is to be minimised,  $\lambda$  the vector of Lagrange multipliers,  $\psi$  the vector with terminal constraints and  $\Sigma$  the penalty matrix to place additional weight on each constraint. The Lagrange multipliers are initialised at zero and updated with every iteration of the algorithm, see Section III.E.

### D. Backward sweep

The backward sweep is the step of the algorithm where the control update is determined. This step solves the sequence of subproblems that minimises the cost-to-go from stage  $k = N - 1, N - 2, \dots, 0$ . The cost-to-go is defined as the cost incurred from the current stage to the final destination [17]. The optimal cost-to-go at each stage is written as

$$J_k^* = \min_{\delta \mathbf{u}_k} [J_k] \quad (25)$$

where  $J_k$  is the cost-to-go from stage  $k$  until the end of the trajectory. The superscript asterisk denotes an optimal value, so  $J_k^*$  is the optimal cost-to-go from stage  $k$  until the end of the trajectory. The solution to Eq. 25 is the optimal control update  $\delta \mathbf{u}_k^*$ . The cost function is approximated with a second-order Taylor series expansion around the states, controls and Lagrange multipliers. By taking the derivative of this expansion with respect to the controls and setting it to zero, an unconstrained feedback control law for the optimal control update is found as

$$\begin{cases} \delta \mathbf{u}_k^* = A_k + B_k \delta \mathbf{x}_k + D_k \delta \lambda \\ A_k = -J_{uu,k}^{-1} J_{u,k} \\ B_k = -J_{uu,k}^{-1} J_{ux,k} \\ D_k = -J_{uu,k}^{-1} J_{u\lambda,k} \end{cases} \quad (26)$$

where  $\delta \lambda$  is the Lagrange multiplier update, see Section III.E,  $A$  is the feedforward control matrix, and  $B$  and  $D$  are the feedback control matrices. It is important to note that the subscript  $x$  is used, not  $X$ . The derivatives with respect to the state,  $\mathbf{x}$ , and the controls,  $\mathbf{u}$ , are a part of the derivative with respect to the complete augmented state vector,  $\mathbf{X}$ , as

$$J_X = \begin{bmatrix} J_x & J_u \end{bmatrix}^T = \begin{bmatrix} \frac{\partial J}{\partial p} & \frac{\partial J}{\partial f} & \frac{\partial J}{\partial g} & \frac{\partial J}{\partial h} & \frac{\partial J}{\partial k} & \frac{\partial J}{\partial l} & \frac{\partial J}{\partial m} & \frac{\partial J}{\partial t} & \mid & \frac{\partial J}{\partial \alpha} & \frac{\partial J}{\partial \delta} \end{bmatrix} \quad (27)$$

Note that  $J_{uu}$  and  $J_{ux}$  are submatrices of  $J_{XX}$ , following the same analogy as for  $J_X$  in Eq. 27. The submatrices for Eq. 26 are calculated and stored for use during the forward sweep of the algorithm. Note that  $\delta \mathbf{u}_k^*$  cannot be determined

directly, since it is dependent on  $\delta \mathbf{x}_k$ . This variable is the change in state due to the changed control, which is calculated during the forward integration of the state. A trust-region method limits the size of the hessian,  $J_{uu,k}$ , such that the resulting change in state,  $\delta \mathbf{x}_k$ , remains within the valid region of the quadratic model, see Section III.F. The remainder of the backward sweep is focused on finding the required matrices for Eq. 26. These matrices can be found using the cost-to-go derivatives with respect to the augmented state, and Lagrange multipliers as

$$J_{X,k} = \Phi^T J_{X,k+1}^* \quad (28a)$$

$$J_{\lambda,k} = J_{\lambda,k+1}^* \quad (28b)$$

$$J_{XX,k}^{i,a} = J_{XX,k+1}^{*\gamma_1,\gamma_2} \Phi^{\gamma_1,i} \Phi^{\gamma_2,a} + J_{X,k+1}^{*,\gamma_1} \Phi^{\gamma_1,ia} \quad (28c)$$

$$J_{\lambda\lambda,k} = J_{\lambda\lambda,k+1}^* \quad (28d)$$

$$J_{X\lambda,k} = \Phi^T J_{X\lambda,k+1}^* \quad (28e)$$

where  $\Phi^{i,a}$  is the state transition matrix and  $\Phi^{i,ab}$  is the second-order state transition tensor, both will be referred to as the STMs. The derivatives of the cost-to-go at stage  $k$  in Eq. 28 are dependent on the derivatives of the optimal cost-to-go at stage  $k+1$ . Details on how the stage's optimal cost-to-go derivatives are calculated are covered at the end of this section. Some additional information per subequation is provided below.

- For the derivatives with respect to the Lagrange multipliers in Eqs. 28b and 28d, the derivatives at stage  $k$  are equal to the optimal derivatives at stage  $k+1$ . In addition, the observant reader might notice that  $J_{\lambda,k}$  and  $J_{\lambda\lambda,k}$  in Eqs. 28b and 28d are not directly needed in Eq. 26. However, these two derivatives are required to calculate the update of the Lagrange multipliers,  $\delta \lambda$ , see Section III.E.
- For the derivatives where the state,  $\mathbf{x}$ , is involved (Eqs. 28a, 28c and 28e), the optimal cost-to-go derivatives from stage  $k+1$  are mapped to stage  $k$  with the use of the STMs.
- In Eq. 28c Einstein notation (as introduced in Section III.A) is used to indicate summation over two dummy indices,  $\gamma_1$  and  $\gamma_2$ , which is used to find the hessian,  $J_{XX,k}$ . For clarification, this equation can alternatively be written as

$$J_{XX,k}^{i,a} = \sum_{\gamma_1=1}^m \sum_{\gamma_2=1}^m J_{XX,k+1}^{*\gamma_1,\gamma_2} \Phi^{\gamma_1,i} \Phi^{\gamma_2,a} + J_{X,k+1}^{*,\gamma_1} \Phi^{\gamma_1,ia} \quad (29)$$

where  $m$  is the size of  $\mathbf{X}$ .

Evaluating the STMs is the most computationally expensive part of the algorithm, and is performed in parallel using OpenMP [33]. Parallel computation is possible since the state at each point in time is already known from the forward sweep. In order to determine the STMs, the dynamics matrix and tensor are defined as

$$F^{i,a} = \frac{\partial \dot{X}^i}{\partial X^a} \quad (30a)$$

$$F^{i,ab} = \frac{\partial^2 \dot{X}^i}{\partial X^a \partial X^b} \quad (30b)$$

where the  $(i, a)^{\text{th}}$  entry of the dynamics tensor,  $F^{i,a}$ , corresponds to the derivative of the  $i^{\text{th}}$  entry of  $\dot{X}$  with respect to the  $a^{\text{th}}$  entry of  $X$ . For the third-order dynamics tensor,  $F^{i,ab}$ , the  $(i, ab)^{\text{th}}$  entry of the tensor corresponds to the second-order derivative of the  $i^{\text{th}}$  entry of  $\dot{X}$  with respect to the  $a^{\text{th}}$  and  $b^{\text{th}}$  entry of  $X$ . The entries of the dynamics matrix and tensor are found analytically\*.

The previously proposed Sundman transformation requires the dynamics matrix and tensor to be transformed. As stated in Section II.F and particularly in Eq. 16, the Sundman transformation is performed by multiplying the derivative of the augmented state vector with

$$\eta = dt/d\nu \quad (31)$$

The first and second derivatives of  $\eta$  with respect to the augmented state vector are written as

$$\eta_X^i = \frac{\partial \eta}{\partial X^i} \quad (32a)$$

$$\eta_{XX}^{i,a} = \frac{\partial^2 \eta}{\partial X^i \partial X^a} \quad (32b)$$

The dynamics matrix and tensor can then be transformed as

$$\Lambda^{i,a} = F^{i,a} \eta + \dot{X}^i \eta_X^a \quad (33a)$$

$$\Lambda^{i,ab} = F^{i,ab} \eta + F^{i,a} \eta_X^b + F^{i,b} \eta_X^a + \dot{X}^i \eta_{XX}^{a,b} \quad (33b)$$

where  $\Lambda^{i,a}$  and  $\Lambda^{i,ab}$  are the Sundman-transformed dynamics matrix and tensor, respectively. Similar to the computation of the derivative of the augmented state vector in Section II.F, the dynamics matrix and tensor are both first determined in Cartesian elements, after which they are transformed to MEE. This process is covered in Appendix A of Reference [12]. Finally, the differential equations that need to be solved to find the STMs (again using Einstein notation) are:

$$\dot{\Phi}^{i,a} = \Lambda^{i,\gamma_1} \Phi^{\gamma_1,a} \quad (34a)$$

$$\dot{\Phi}^{i,ab} = \Lambda^{i,\gamma_1} \Phi^{\gamma_1,ab} + \Lambda^{i,\gamma_1\gamma_2} \Phi^{\gamma_1,a} \Phi^{\gamma_2,b} \quad (34b)$$

This set of differential equations can easily be solved by knowing the boundary conditions  $\Phi(t_k, t_k)^{i,a} = \mathbf{I}$  and  $\Phi(t_k, t_k)^{i,ab} = \mathbf{0}$ , where  $\mathbf{I}$  is the identity matrix and  $\mathbf{0}$  a matrix of zeroes. In other words, at the beginning of each stage the state transition matrix is equal to the identity matrix and the state transition tensor is equal to zero. The STMs are then evaluated at the end of each stage using a single step of the DOPRI8 integration routine.

Now that it is clear how to obtain the STMs in Eq. 28, the derivatives of the cost-to-go at stage  $k$  in Eq. 28 can be calculated, assuming that the optimal cost-to-go derivatives from stage  $k + 1$  are known. Afterwards, the optimal

---

\* Maple worksheets available at: [https://github.com/gijsleemans1/Analytical\\_derivatives](https://github.com/gijsleemans1/Analytical_derivatives). Accessed: 9/6/2021

cost-to-go derivatives from the current stage,  $k$ , are calculated as

$$ER_k = ER_{k+1} + J_{u,k}^T A_k + \frac{1}{2} A_k^T J_{uu,k} A_k \quad (35a)$$

$$J_{x,k}^{*T} = J_{x,k}^T + J_{u,k}^T B_k + A_k^T J_{uu,k} B_k + A_k^T J_{ux,k} \quad (35b)$$

$$J_{\lambda,k}^{*T} = J_{\lambda,k}^T + J_{u,k}^T D_k + A_k^T J_{uu,k} D_k + A_k^T J_{u\lambda,k} \quad (35c)$$

$$J_{xx,k}^{*T} = J_{xx,k}^T + B_k J_{uu,k}^T B_k + B_k^T J_{ux,k} + J_{ux,k} B_k \quad (35d)$$

$$J_{\lambda\lambda,k}^{*T} = J_{\lambda\lambda,k}^T + D_k J_{uu,k}^T D_k + D_k^T J_{u\lambda,k} + J_{u\lambda,k} D_k \quad (35e)$$

$$J_{x\lambda,k}^{*T} = J_{x\lambda,k}^T + B_k J_{uu,k}^T D_k + B_k^T J_{u\lambda,k} + J_{ux,k} D_k \quad (35f)$$

where  $ER_k$  is the expected reduction in cost at stage  $k$ . The expected reduction is not needed to calculate the control update in Eq. 26, but is used to determine whether an iteration of the DDP algorithm has been successful or not, see Section III.G. At the final stage, the optimal and nominal costs are equivalent, i.e.  $J_N^* = J$ . This means that the derivatives of the optimal cost-to-go are equal to the derivatives of the augmented Lagrangian, which can be found analytically\*. Note that there is no expected reduction at the final stage.

To summarise, the backward sweep starts by calculating the derivatives of the optimal cost-to-go at the final stage analytically and the evaluation of the STMs at all stages. Then, an iterative approach is started to calculate the current stage's cost-to-go derivatives (see Eq. 28), the feedback and feedforward matrices (see Eq. 26), and optimal cost-to-go derivatives (see Eq. 35). After all the aforementioned terms have been calculated for stage  $k$ , the algorithm does the same for stage  $k - 1$ . This process is repeated until the algorithm has evaluated the feedback and feedforward matrices for the first stage.

### E. Lagrange multiplier update

The Lagrange multipliers are updated once per iteration of the algorithm, after the backward sweep has been completed, as

$$\delta\lambda = -J_{\lambda\lambda}^{-1} J_{\lambda} \quad (36)$$

where  $J_{\lambda\lambda} = J_{\lambda\lambda,0}^*$  and  $J_{\lambda} = J_{\lambda,0}^*$ , indicating the derivatives are evaluated at stage  $k = 0$ . Also, the expected reduction has to be updated as

$$ER_0 = ER_0 + J_{\lambda}^T \delta\lambda + \frac{1}{2} \delta\lambda^T J_{\lambda\lambda} \delta\lambda \quad (37)$$

where  $ER_0$  is the expected reduction at stage  $k = 0$ .



## F. Trust region quadratic subproblem

As was stated in Section III.D, the trust region quadratic subproblem (TRQP) restricts the size of the hessian,  $J_{uu,k}$ , so that the resulting change in state,  $\delta \mathbf{x}_k$ , remains within the valid region of the quadratic model. This sub-algorithm is employed at each stage as soon as  $J_{uu,k}$  is calculated in Eq. 28c. The expected reduction (see Eq. 35a),  $A_k$ ,  $B_k$  and  $D_k$  (see Eq. 26) are all updated if  $J_{uu,k}$  is changed by the TRQP. The trust region subproblem is formulated as

$$\min_{\delta \mathbf{u}_k} J_{u,k} \delta \mathbf{u}_k + \frac{1}{2} \delta \mathbf{u}_k^T J_{uu,k} \delta \mathbf{u}_k \quad \text{such that} \quad \|G \delta \mathbf{u}_k\| \leq \Delta \quad (38)$$

where  $G$  is a scaling matrix, which was set to the identity matrix throughout this work,  $\|\square\|$  is the Euclidean norm and  $\Delta$  is the trust region radius. Many different methods for solving the TRQP exist in literature. The methods of Reference [34] have been proven to be robust for solving the TRQP in DDP. In particular, algorithm 7.3.4. (with the enhancement of introducing a scaling matrix,  $M$ ), has been used to solve the TRQP in DDP [12, 17]. This method is from hereon referred to as the "classical TRQP". According to Reference [17], a less rigorous and efficient approach is arbitrary hessian shifting, from hereon referred to as the "simple TRQP". In the simple TRQP, the same user-defined value is added to the diagonal of the hessian at every stage. If the resulting control update is outside of the trust region, the value on the diagonal is changed such that the control update lies exactly on the boundary of the trust region. The TRQP not only restricts  $J_{uu}$ , but also  $J_{\lambda\lambda}$ . However, there is one notable difference between the TRQP for the control update and the multiplier update:  $J_{uu}$  is required to be positive definite, while  $J_{\lambda\lambda}$  is required to be negative definite. So, the control parameter subproblem is defined as  $\text{TRQP}(J_u, J_{uu}, \Delta)$  and the multiplier subproblem as  $\text{TRQP}(-J_\lambda, -J_{\lambda\lambda}, \Delta)$ .

## G. Iterations

After the backward sweep has been completed, it is determined whether the iteration should be accepted or rejected. Therefore, a reduction ratio is calculated as

$$\rho = \frac{\delta J}{ER_0} \quad (39)$$

where  $\delta J$  is the difference in final cost between successive iterations and  $ER_0$  is the expected reduction at the first stage. When  $\rho \approx 1$ , the model is valid because the reduction in cost is as predicted. The iteration is then accepted and for the next iteration the trust-region is enlarged. If  $\rho$  is not close to one, the iteration should be rejected and the trust-region should be reduced for the next iteration. The STMs will not have to be calculated again when the iteration is rejected, since the state will not have changed because the controls remain unchanged. The algorithm can immediately proceed from the backward sweep onwards. The size of the trust region for the subsequent iteration is calculated as

$$\Delta_{\#p+1} = \begin{cases} \min((1 + \kappa)\Delta_{\#p}, \Delta_{\max}), & \text{if } \rho \in (1 - \epsilon_1, 1 + \epsilon_1) \\ \max((1 - \kappa)\Delta_{\#p}, \Delta_{\min}), & \text{otherwise} \end{cases} \quad (40)$$

where  $\#p$  is the iteration counter,  $\kappa$  determines by how much the trust region is enlarged or shrunk during each iteration,  $\Delta_{\max}$  and  $\Delta_{\min}$  are the maximum and minimum trust region radii, respectively, and  $\epsilon_1$  gives the tolerance on  $\rho$  which is

deemed acceptable. The complete algorithm is said to be converged when the expected reduction of the latest iteration is smaller than the optimality tolerance,  $\epsilon_2$ .

## H. Monotonic basin hopping

Classical dynamic programming algorithms minimise the cost function by discretising the stages and controls, which has the consequence that the algorithm satisfies the conditions for global optimality [17]. The downsides of these classical algorithms is that they have large storage requirements due to the "curse of dimensionality". Differential dynamic programming overcomes this issue by introducing the quadratic trust region, forcing the solution to be inside a given corridor around a reference solution, thereby reducing the dimension of the search space. This has the consequence that DDP can only be considered to find locally optimal solutions. However, since DDP is robust to poor initial guesses, it can be paired with a global search method, such as monotonic basin hopping (MBH), to find globally optimal solutions [12].

MBH works as follows [20]: first, the inner optimisation algorithm, DDP, runs with an initial guess. The result of the optimisation is a set of control variables that results in a locally optimal trajectory. This set of control variables is randomly perturbed within the control bounds, and fed to the optimisation algorithm as a new initial guess. If the optimisation yields a trajectory with lower cost, the iteration is accepted, the perturbed control becomes the new control and the procedure is repeated. If the new trajectory has a higher cost, the iteration is rejected and the original controls are again randomly perturbed. This is repeated until a stopping criterion is met.

## I. Control Bounds

Control parameters can be limited by control bounds  $u^L \leq u \leq u^U$ , where  $u^L$  is a lower bound and  $u^U$  is an upper bound of the control parameter  $u$ . The original DDP algorithm treats control bounds with a null-space method [18]. This work proposes a different method of incorporating control bounds, namely a Heaviside-based approach. The benefit hereof is that the information regarding the bounds is saved in the STMs. The Heaviside function is defined as

$$H\{j\} = \begin{cases} 1, & j > 0 \\ 0, & j < 0 \end{cases} \quad (41)$$

Since the DDP algorithm requires all equations of motion to be twice differentiable, the Heaviside function is approximated with the smooth logistic function as

$$H\{j\} \approx \frac{1}{2} + \frac{1}{2} \tanh(\tilde{h}j) \quad (42)$$

where  $\tilde{h}$  is a tuning parameter that determines the steepness of the ascent and descent of the Heaviside function. Increasing  $\tilde{h}$  too much will lead to the derivatives becoming very large at the bounds and the algorithm becoming unstable. Throughout this work,  $\tilde{h}$  was set to 100. The new control parameter  $u_{new}$ , calculated from the original  $u$  with the combination of Heaviside functions is:

$$u_{new} = u + (u_L - u) \times H\{-\tilde{h}(u - u_L)\} + (u_U - u) \times H\{\tilde{h}(u - u_U)\} \quad (43)$$

When the control is within the control bounds, both Heaviside functions will equate to zero and  $u_{new} = u$ . If the control is below the lower bound,  $u < u^L$ , the Heaviside term  $H\{-\tilde{h}(u - u_L)\}$  becomes equal to one, which will add the difference between the lower bound and the current control value,  $u_L - u$ , to the current control, resulting in the control being exactly on the lower bound. If the control is above the upper bound,  $u > u^U$ , the Heaviside term  $H\{\tilde{h}(u - u_U)\}$  becomes equal to one, which will add the difference between the upper bound and the current control value,  $u_U - u$ , which is a negative value, to the current control. This results in the control being exactly on the upper bound.

## IV. Results

This section presents the results through a set of five test cases to demonstrate the applicability of DDP to optimise Earth-centred solar-sail trajectories as well as demonstrating the impact on the performance of DDP depending on the complexity of the dynamics considered, the settings of the algorithm and the inclusion of constraints. The first test case in Section IV.A is included for validation purposes and to show the superior performance of the DDP optimisation approach proposed in this paper over the locally optimal results known from the literature. Subsequently, in Section IV.B the case is extended to more complex dynamics and a thorough analysis of various settings used for the DDP algorithm is given. Section IV.C demonstrates the algorithm's performance on a real-life example in the form of NASA's upcoming ACS3 mission. Constraints are implemented that force the ACS3 sailcraft to keep its Sun-synchronous attribute. While all test cases in Sections IV.A-IV.C are optimised over the course of only three orbital revolutions, Sections IV.D and IV.E extend these optimisation to many revolutions around Earth. Note that for all test cases the specific orbital energy is optimised.

### A. Two-dimensional geostationary orbit: three orbital revolutions

This first test case concerns the optimisation of a spacecraft in geostationary orbit (42,241 km altitude, zero inclination and eccentricity) over the course of three orbital revolutions. Since no constraints are used, the augmented Lagrangian cost function can be written as

$$J = \phi = - \left( \frac{\dot{r}^2}{2} - \frac{\mu_{\oplus}}{r} \right) \quad (44)$$

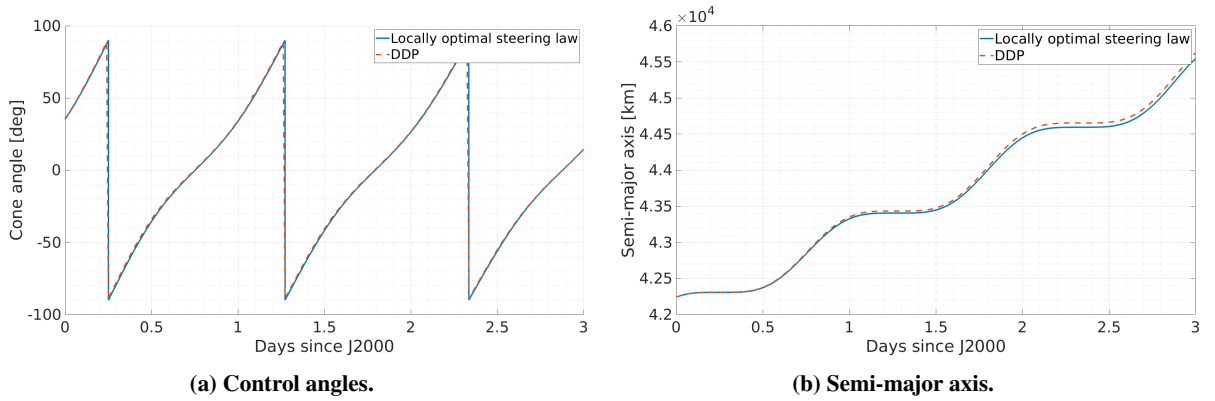
Note the minus sign in Eq. 44, since the specific orbital energy is to be maximised, whereas the cost function is minimised by the DDP algorithm. A two-dimensional dynamical model is assumed where the direction of sunlight is assumed to coincide with the orbital plane and the Sun is located at a distance of one astronomical unit (AU), fixed along the negative  $x$ -axis of the ECI frame. The position vector of the Sun is written as

$$\mathbf{r}_{\odot} = \begin{bmatrix} -1AU & 0 & 0 \end{bmatrix} \quad (45)$$

The control is then solely determined by the sail cone angle,  $\alpha$ , as the clock angle,  $\delta$ , is fixed at  $90^\circ$ . For

the cone angle, an initial guess of 0 is used across all stages. For now, only point-mass gravity from the Earth and the solar-sail acceleration itself are acting on the sailcraft, i.e., the following terms in Eq. 2 equate to zero:  $\ddot{\mathbf{r}}_{J_2} = \ddot{\mathbf{r}}_{J_3} = \ddot{\mathbf{r}}_{J_4} = \ddot{\mathbf{r}}_{\odot} = \ddot{\mathbf{r}}_{\oplus} = 0$ . These assumptions and simplifications are made to allow validation of the DDP algorithm. The output of the DDP algorithm is compared to the outcome following from a locally optimal steering law from Reference [13], where a solar-sail lightness number of 0.17 is used, which is adopted here as well.

The resulting control profile and evolution of the semi-major axis are shown in Figure 3. The control profiles found by DDP and the locally optimal steering law are very similar, although a small difference is visible. The final semi-major axis found by DDP is indeed approximately 80 m larger than the one found from the locally optimal steering law. Both solutions are local optima, but the solution found by the DDP algorithm is a better local optimum. With a stricter optimality tolerance, the DDP algorithm might be able to find an even larger semi-major axis increase. Since the DDP algorithm finds a similar control profile and a solution that is even more optimal than the locally optimal steering law, it can be concluded that the solution is a valid optimum. This better optimum comes at the cost of a computational time of 853 seconds, compared to a runtime of approximately 0.4 seconds for the steering law solution<sup>†</sup>. Note that the steering law solution is obtained with a varying stepsize integrator, whereas the DDP solution is obtained with a fixed stepsize integrator. The varying stepsize integrator might take larger steps, thereby decreasing the computational effort required. When more complex and longer-duration solar-sail scenarios are considered, such as the ones in the next four sections, the increase in computational time is expected to be justified. The improvement to the optimal solution is expected to be proportionally larger when more complex problems are evaluated. Furthermore, contrary to the locally optimal steering law approach, DDP allows to include terminal and path-inequality constraints on the optimisation problem. Significant reductions in runtime can be achieved when using the locally optimal steering law as initial guess for the DDP algorithm, which reduces the runtime to only 64 seconds.



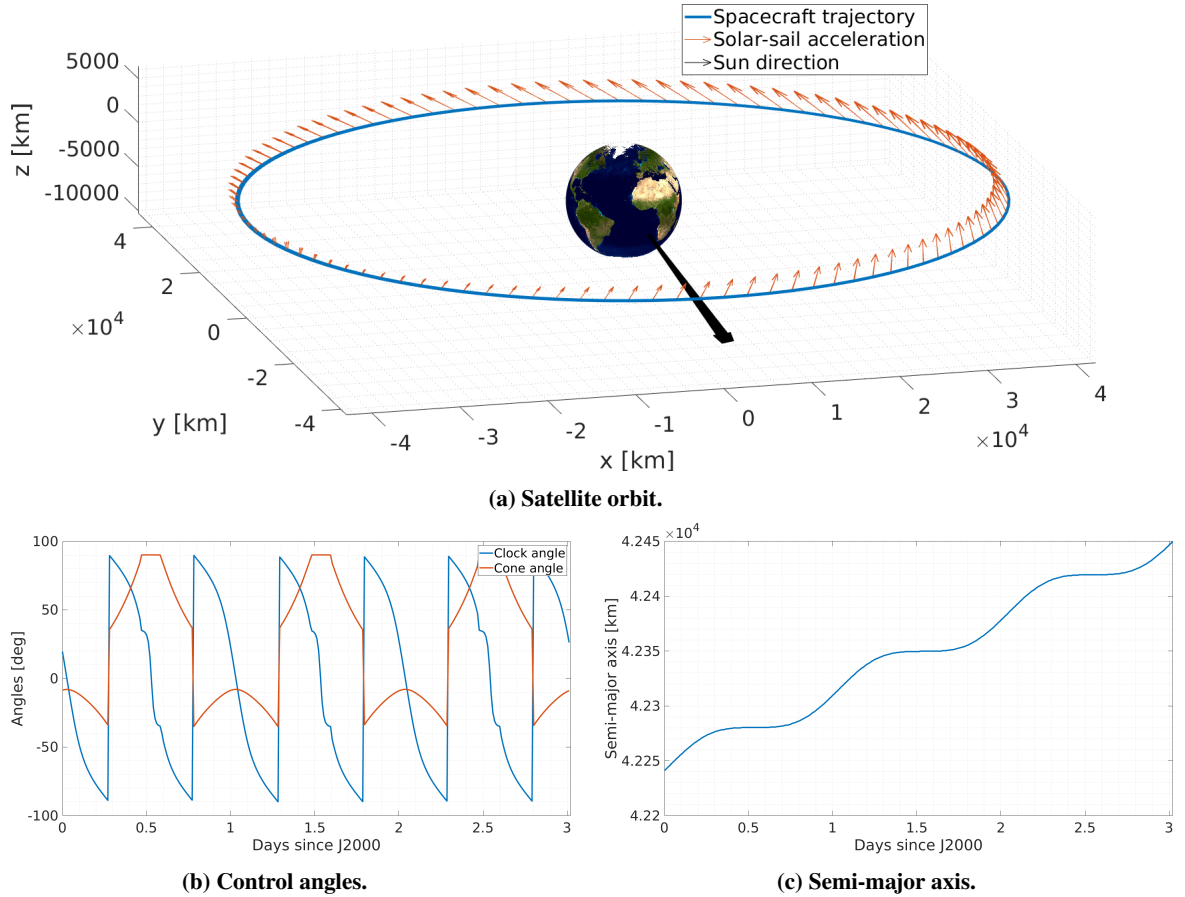
**Fig. 3 Two-dimensional geostationary orbit: comparison with locally optimal steering law [13].**

<sup>†</sup>Matlab ODE45 integration scheme with relative and absolute tolerance  $10^{-12}$ . Computation on personal laptop CPU: Quad-core Intel(R) Core(TM) i7-10610U @1.80 GHz

### B. Three-dimensional geostationary orbit: three orbital revolutions

As a first extension, the problem of the previous section is expanded into three dimensions by removing the assumption on the direction of sunlight to be fixed and planar. Instead, the Sun's position is retrieved from ephemeris data, as explained in Section II.D. Also, a more realistic lightness number of 0.011 is assumed. The clock angle is no longer fixed to  $90^\circ$ , but is optimised by the DDP algorithm from an initial guess of zero at every stage. All other case details remain the same as those described in Section IV.A.

The resulting trajectory is shown in Figure 4a, where also the direction towards the Sun and the solar-sail acceleration are visualised. The optimal control profile, see Figure 4b, shows that both the cone and clock angles vary periodically with every revolution of the satellite around the Earth. The resulting semi-major axis increase from Figure 4c behaves similar to the increase in the two-dimensional case, although there is a shift in phase, since the phasing of the Sun and spacecraft is different when using the true ephemeris of the Sun. Furthermore, the magnitude of the increase is different as a different solar-sail lightness number is used.



**Fig. 4 Three-dimensional geostationary orbit: three orbital revolutions.**

### 1. DDP algorithm settings

Various settings of the DDP algorithm can be tuned, as explained in Section III. The influence of changing these settings for the current problem is investigated to find out how robust the DDP algorithm is to different settings and what the best settings are for the remaining test cases. In this section, the effect of changing the reduction ratio tolerance,  $\epsilon_1$ , the optimality tolerance,  $\epsilon_2$ , the initial trust region radius,  $\Delta_0$ , the maximum trust region radius,  $\Delta_{\max}$ , the minimum trust region radius,  $\Delta_{\min}$ , and the trust region scaling factor,  $\kappa$ , is assessed. For each setting three different values are used. The combination of settings which are used and their resulting semi-major axis increase, as well as the algorithm's runtime, are presented in Table 3. Computation was executed on sixteen threads of an Intel(R) Xeon(R) CPU E5-2683 v3 @2.00GHz.

Two settings seem to have the largest influence on the semi-major axis increase, the optimality tolerance:  $\epsilon_2$ , and the minimum trust region radius,  $\Delta_{\min}$ , which were both defined in Section III.G. The smaller the optimality tolerance, the larger the final semi-major axis. This is as expected, since the optimality tolerance is the convergence criterion for the DDP algorithm. When a smaller value is used, the algorithm will iterate longer to find a more optimal trajectory. Any minimum trust region radius larger than zero leads to suboptimal trajectories. The trust region should be allowed to shrink when unsuccessful iterations take place, but it cannot become smaller than the minimum trust region radius. The trust region scaling factor,  $\kappa$ , also has a small effect on the optimality of the trajectory, for which an optimal value of 0.01 was found. The differences in semi-major axis resulting from varying the other settings are minor.

In general, the differences in runtime for different settings is small, the only large differences are observed for different optimality tolerances, where a smaller tolerance will lead to a longer runtime, and for varying the initial trust region radius, where the algorithm converges quickly to a suboptimal solution.

The settings that result in the largest semi-major axis are:  $\epsilon_1 = 0.1$ ,  $\epsilon_2 = 1 \times 10^{-7}$ ,  $\Delta_0 = \pi/2$ ,  $\Delta_{\max} = \pi/2$ ,  $\Delta_{\min} = 0$ , and  $\kappa = 0.01$ . However, it is likely that by making the optimality tolerance,  $\epsilon_2$ , even smaller, the resulting semi-major axis will become even larger. However, this increase in optimality will come at the cost of an increase in computational time. The aforementioned settings will be used in the remainder of this work, with the exception of the optimality tolerance, for which  $\epsilon_2 = 1 \times 10^{-6}$  will be used. The difference in final semi-major axis between using  $\epsilon_2 = 1 \times 10^{-6}$  and  $\epsilon_2 = 1 \times 10^{-7}$  is very small, but comes at the cost of an increase in runtime of approximately 65%.

**Table 3 Three-dimensional geostationary orbit: effect of varying DDP settings.**

$\epsilon_1$	$\epsilon_2$	$\Delta_0$	$\Delta_{\max}$	$\Delta_{\min}$	$\kappa$	$\Delta a$ [km]	Runtime [s]
0.05	$1 \times 10^{-6}$	$\pi$	$\pi$	0	0.05	208.848116	271.9
0.2	$1 \times 10^{-6}$	$\pi$	$\pi$	0	0.05	208.843899	240.3
0.5	$1 \times 10^{-6}$	$\pi$	$\pi$	0	0.05	208.845164	241.9
0.1	$1 \times 10^{-5}$	$\pi$	$\pi$	0	0.05	208.767372	131.0
0.1	$1 \times 10^{-6}$	$\pi$	$\pi$	0	0.05	208.850688	213.6
0.1	$1 \times 10^{-7}$	$\pi$	$\pi$	0	0.05	208.864644	351.5
0.1	$1 \times 10^{-6}$	$\pi/2$	$\pi$	0	0.05	208.851152	214.1
0.1	$1 \times 10^{-6}$	$3\pi/2$	$\pi$	0	0.05	208.850688	213.7
0.1	$1 \times 10^{-6}$	$2\pi$	$\pi$	0	0.05	208.850688	215.0
0.1	$1 \times 10^{-6}$	$\pi$	$\pi/2$	0	0.05	208.851109	213.7
0.1	$1 \times 10^{-6}$	$\pi$	$3\pi/2$	0	0.05	208.851109	214.1
0.1	$1 \times 10^{-6}$	$\pi$	$2\pi$	0	0.05	208.850013	213.9
0.1	$1 \times 10^{-6}$	$\pi$	$\pi$	0.05	0.05	208.815523	160.0
0.1	$1 \times 10^{-6}$	$\pi$	$\pi$	0.1	0.05	113.530786	6.8
0.1	$1 \times 10^{-6}$	$\pi$	$\pi$	0.2	0.05	113.530786	6.3
0.1	$1 \times 10^{-6}$	$\pi$	$\pi$	0	0.01	208.851657	219.8
0.1	$1 \times 10^{-6}$	$\pi$	$\pi$	0	0.1	208.850182	217.9
0.1	$1 \times 10^{-6}$	$\pi$	$\pi$	0	0.2	208.846977	222.2

There are more settings that can be changed than just the settings of the DDP algorithm itself. In Section III.F, two different trust region algorithms were discussed, the "classical TRQP" and the "simple TRQP". The classical TRQP is tuned by changing the value of the scaling matrix,  $M$ , whereas the simple TRQP is tuned by changing the magnitude of the hessian shift. The results of using the different algorithms with different settings are presented in Table 4. There are significant differences in runtime between the simple and classical algorithms: the simple algorithm converges multiple times faster, and it produces more optimal trajectories. It is interesting to note that the setting that converges fastest, is also the setting resulting in the most optimal trajectory. Therefore, in the remainder of this work, the simple TRQP with a hessian shift of 0.01 is used.

Finally, the effect of using MBH has been investigated. Different combinations of settings for the chance and magnitude of the perturbation on the control variables were used. However, MBH was hardly able to improve on the solution found by DDP. In the cases where it did improve, the improvement was only very minor and comes at the cost of

**Table 4 Three-dimensional geostationary orbit: effect of varying TRQP settings.**

TRQP	Scaling/Shift	$\Delta a$ [km]	Runtime [s]
Simple	$1 \times 10^{-1}$	208.736761	1347.2
Simple	$1 \times 10^{-2}$	208.851109	223.1
Simple	$1 \times 10^{-3}$	208.676761	692.1
Simple	$1 \times 10^{-4}$	208.670943	750.7
Classical	$1 \times 10^{-3}$	208.016431	2254.4
Classical	$1 \times 10^{-4}$	208.466405	2455.0
Classical	$1 \times 10^{-5}$	206.779676	4957.1
Classical	$1 \times 10^{-6}$	206.632903	6358.9

a significant increase in computational time. It is likely that the same improvement could have been found without using MBH, but with a stricter optimality tolerance on the DDP algorithm itself.

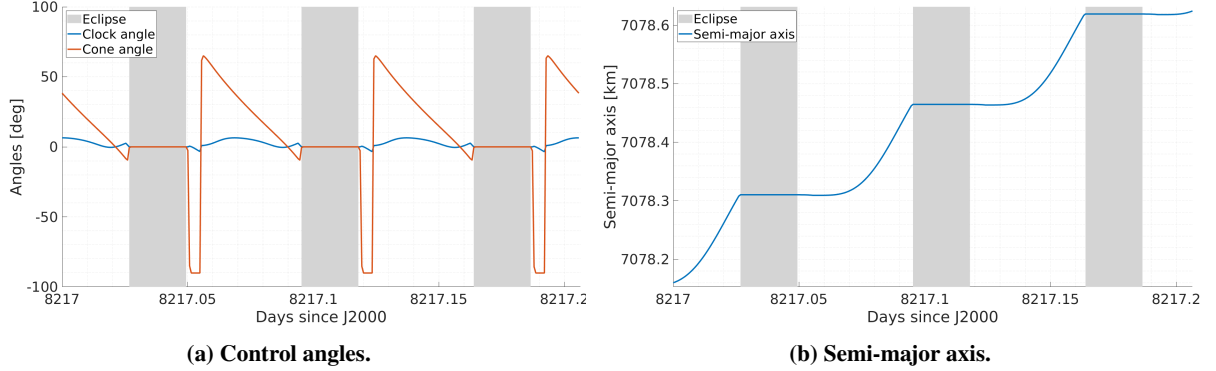
## 2. Effect of perturbing accelerations

So far, the only accelerations acting on the sailcraft have been point-mass gravity from the Earth and the solar-sail acceleration. The effect of including other perturbing accelerations in the dynamical model is shown in Table 5. Including the third-body perturbation from the Moon produced the largest effect for which a decrease in the final semi-major axis of approximately 0.8 km is observed. Note that the magnitude of the semi-major axis decrease due to lunar gravity will most likely change when a different initial epoch is considered, since the acceleration acting on the satellite is dependent on the Moon's position which changes over time. The effect of considering  $J_2$ ,  $J_3$ ,  $J_4$  and solar gravity is very small, where  $J_2$  has some effect, but considerably less than the gravitational acceleration from the Moon. There is no significant impact on the runtime. With all perturbing accelerations taken into account, a final semi-major axis increase of 208.05 km is obtained, which is significant considering that only three orbital revolutions are simulated. Therefore, it appears that a current state-of-the-art solar sail is a viable method of propulsion for orbit raising at geostationary altitude.

**Table 5 Three-dimensional geostationary orbit: effect of varying dynamical model settings.**

Perturbations	$\Delta a$ [km]	Runtime [s]
None	208.851109	223.1
Moon	208.062980	232.8
$J_2$ and Moon	208.051849	227.3
$J_2$ , $J_3$ , $J_4$ , Moon and Sun	208.049825	242.5





**Fig. 5 ACS3 mission: three orbital revolutions.**

### C. ACS3 mission: three orbital revolutions

For the analysis in this section, the sail characteristics from NASA’s upcoming ACS3 mission are considered. It is assumed that the sailcraft will be launched into a 700 km altitude noon-midnight Sun-synchronous orbit (inclination  $i = 98.19^\circ$ ), with a solar-sail characteristic acceleration of  $0.05 \text{ mm/s}^2$  (corresponding to a solar-sail lightness number of  $0.0084$ )<sup>‡</sup>. The scheduled launch date is assumed to be June 1<sup>st</sup> 2022, which means that the sail will be operational from approximately July 1<sup>st</sup> 2022 onwards<sup>‡</sup>. Using the aforementioned initial conditions, the specific orbital energy of the satellite is optimised for a total of three revolutions. Since no constraints are considered yet, the cost function is identical to Eq. 44. The resulting control profile and semi-major axis increase are visualised in Figure 5. Considering the noon-midnight orientation of the Sun-synchronous orbit, the sailcraft spends significant amounts of time in eclipse, which is clearly visible in Figure 5: during eclipse, the semi-major axis does not increase and the cone and clock angles are set to zero.

For the results presented in Figure 5, the only accelerations acting on the sailcraft have been point-mass gravity from the Earth and the solar-sail acceleration (including eclipses). The effect of including other perturbing accelerations in the dynamical model is shown in Table 6. Including the  $J_2$  effect has the greatest impact on the final semi-major axis. The other contributions of the non-spherical Earth,  $J_3$  and  $J_4$ , have some effect, whereas the inclusion of lunar and solar gravity makes no significant difference. Interestingly, the algorithm converges quicker when  $J_2$  is included, whereas it takes longer with the other perturbing accelerations.

<sup>‡</sup>W.K. Wilkie (NASA Langley Research Center), personal communication, May 2021

**Table 6 ACS3 mission: effect of varying dynamical model settings.**

Perturbations	$\Delta a$ [km]	Runtime [s]
None	0.464314	260.9
$J_2$	0.388996	237.0
$J_2, J_3$ and $J_4$	0.385446	303.4
$J_2, J_3, J_4$ , Moon and Sun	0.385513	304.6

### 1. Constrained DDP optimisation

As stated previously, the sailcraft is placed in a Sun-synchronous orbit. However, as soon as the semi-major axis of the orbit changes, the Sun-synchronous attribute is lost. Only by simultaneously changing the inclination of the orbit can the orbit remain Sun-synchronous. The relation between the inclination and semi-major axis for a Sun-synchronous orbit is as follows [35]:

$$\dot{\Omega} = -3\pi \frac{J_2 R_\oplus^2}{p^2} \cos(i) \frac{1}{T} = \frac{2\pi}{T_{\text{es}}} \quad (46)$$

where  $T$  is the orbital period of the satellite and  $T_{\text{es}}$  is the length of a sidereal day. When both sides of Eq. 46 are equal, the orbit is Sun-synchronous. This test case is used to demonstrate that DDP can deal with terminal constraints by implementing a terminal constraint to the cost function, which is minimised alongside the original objective. The constraint is formulated by requiring that the difference between the left- and right-hand sides of Eq. 46 is as close as possible to zero. Then, by assuming a near-circular orbit, the constraint is written as

$$\psi = \cos(i) + \frac{4\pi/T_{\text{ES}}}{3J_2 R_\oplus^2 \sqrt{\mu_\oplus}} a^{7/2} \quad (47)$$

The complete cost function can then be written as

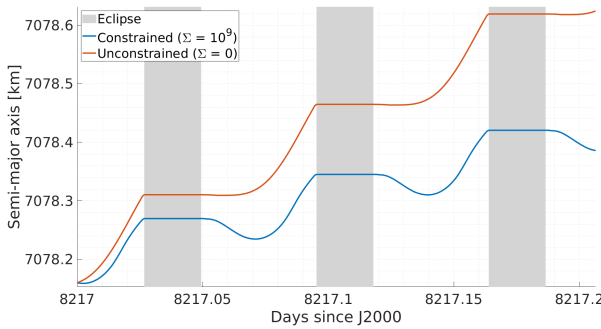
$$J = -\left(\frac{\dot{r}^2}{2} - \frac{\mu_\oplus}{r}\right) + \lambda^T \left( \cos(i) + \frac{4\pi/T_{\text{ES}}}{3J_2 R_\oplus^2 \sqrt{\mu_\oplus}} a^{7/2} \right) + \left( \cos(i) + \frac{4\pi/T_{\text{ES}}}{3J_2 R_\oplus^2 \sqrt{\mu_\oplus}} a^{7/2} \right)^2 \Sigma \quad (48)$$

When the orbit is Sun-synchronous, the value of the constraint will be equal to zero. When the orbit is not Sun-synchronous, the constraint will have a nonzero value and the DDP algorithm will adjust the control variables to minimise the constraint violation. Weights can be added to individual constraints in the augmented cost function using the penalty matrix,  $\Sigma$ . Increasing the weight on the inclination constraint will force the final inclination to be closer to the inclination that is required for Sun-synchronous conditions, but will lead to a smaller final semi-major axis, see Table 7. Here,  $\Delta i$  is the "real" inclination change resulting from the optimal trajectory found by DDP, whereas  $\Delta i_t$  is the "theoretical" inclination change that would be required for the final orbit to be Sun-synchronous. If the DDP algorithm is able to keep the trajectory under Sun-synchronous conditions, these "real" and "theoretical" inclinations are

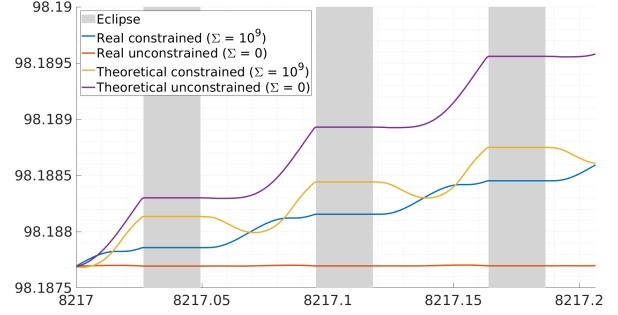
the same. Contrary, when no constraint is implemented, the "real" inclination will not change, since the DDP algorithm is solely focused on maximising the semi-major axis. The closer the "real" and "theoretical" inclination are to each other, the larger  $\Delta i/\Delta i_t$ , and the closer the final orbit is to Sun-synchronous conditions, but the smaller the final semi-major axis, since part of the sail acceleration is devoted to changing the inclination. In addition, the runtime of the algorithm until convergence increases with increasing weights. The evolution of the semi-major axis and inclination for the unconstrained and constrained ( $\Sigma = 1 \times 10^9$ ) cases are shown in Figure 6a and 6b, respectively. It is clear that the final semi-major axis for the constrained case is smaller. Furthermore, the inclination for the constrained case stays close to the theoretical value, while for the unconstrained case the inclination hardly changes.

**Table 7 ACS3 mission: effect of varying weights on inclination constraint.**

$\Sigma$	$\Delta a$ [km]	$\Delta i$ [deg]	$\Delta i_t$ [deg]	$\Delta i/\Delta i_t$ [%]	Runtime [s]
$0 \times 10^6$	0.464208	$3.49504 \times 10^{-6}$	0.00189234	0.18	264.0
$5 \times 10^6$	0.444366	0.000312147	0.00181144	17.2	490.1
$10 \times 10^6$	0.392154	0.000562415	0.00159857	35.2	675.0
$75 \times 10^6$	0.257702	0.000862530	0.00105042	82.1	1044.6
$150 \times 10^6$	0.240958	0.000882698	0.000982161	89.9	1138.8
$1000 \times 10^6$	0.225821	0.0009001161	0.000920452	97.8	1342.0



(a) Semi-major axis comparison.



(b) Inclination comparison.

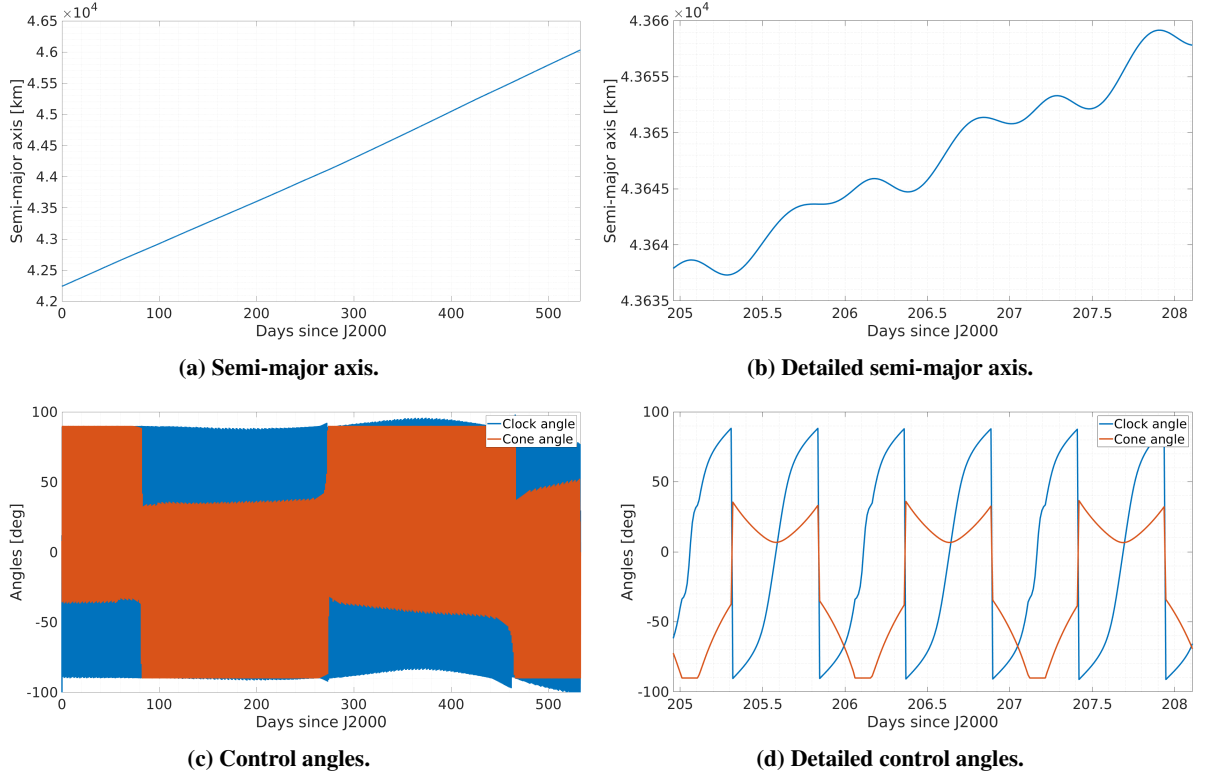
**Fig. 6 ACS3 mission: constrained Sun-synchronous inclination, three orbital revolutions.**

#### D. Three-dimensional geostationary orbit: 500 orbital revolutions.

Previously, in Section IV.B, a three-dimensional geostationary case was investigated for three orbital revolutions. Here, the case is extended to 500 orbital revolutions, with some more important differences. First, the number of orbital revolutions is increased from three to 500 and, second, a smaller solar-sail lightness number is used,  $\beta = 0.0011$ . With a smaller lightness number the sailcraft will remain closer to the geostationary altitude, which is desirable since here

the optimal control profile can be compared with the profile found in Section IV.B. With this comparison, it can be assessed whether the DDP algorithm performs similarly for different cases (different number of orbital revolutions and solar-sail lightness number) under the same dynamical conditions. Again, for validation purposes, the optimisation is first executed when only the point-mass gravity of the Earth is considered and the results are compared to the results from the previously introduced locally optimal steering law [13]. The final semi-major axis increase found by the locally optimal steering law is marginally larger than the increase found by DDP (less than one kilometre over a total semi-major axis increase of 3822 km). It can therefore once again be concluded that the solution found by DDP is a valid optimum. As state before, it could be that the DDP algorithm finds a better solution when a stricter optimality tolerance is used.

From here on, all perturbing accelerations,  $J_2$ ,  $J_3$ ,  $J_4$ , solar gravity, lunar gravity and eclipses are taken into account. The resulting control angles and semi-major axis increase are shown in Figure 7. Though this figure does not provide details for individual revolutions, it does clearly show that the control profile adapts itself to the moving position of the Sun. Furthermore, the shape of the control profile for the first three revolutions was found to be exactly the same as that found in Section IV.B. The figures in the right column of Figure 7 show a detail of the figures in the left column after approximately 200 days have passed. The controls here are similar to the controls seen in Figure 4b, but mirrored due to the difference in the Sun's position. However, the semi-major axis increase shows a different pattern. In the three-revolution case without perturbations, the semi-major axis would always increase or remain unchanged, whereas for the 500 revolution case also a decrease in semi-major axis is observed. This behaviour can be attributed to the perturbations acting on the sailcraft, of which the lunar gravity is the predominant factor. A final semi-major axis increase of 3793.1 km is found. This increase is very large, especially considering that a solar-sail lightness number  $\beta = 0.0011$  was used, which is an order of magnitude smaller than the current state-of-the art. In Section IV.B the conclusion was drawn that a solar-sail is a viable form of propulsion for orbit raising at geostationary altitude, which is in agreement with the result from this section.



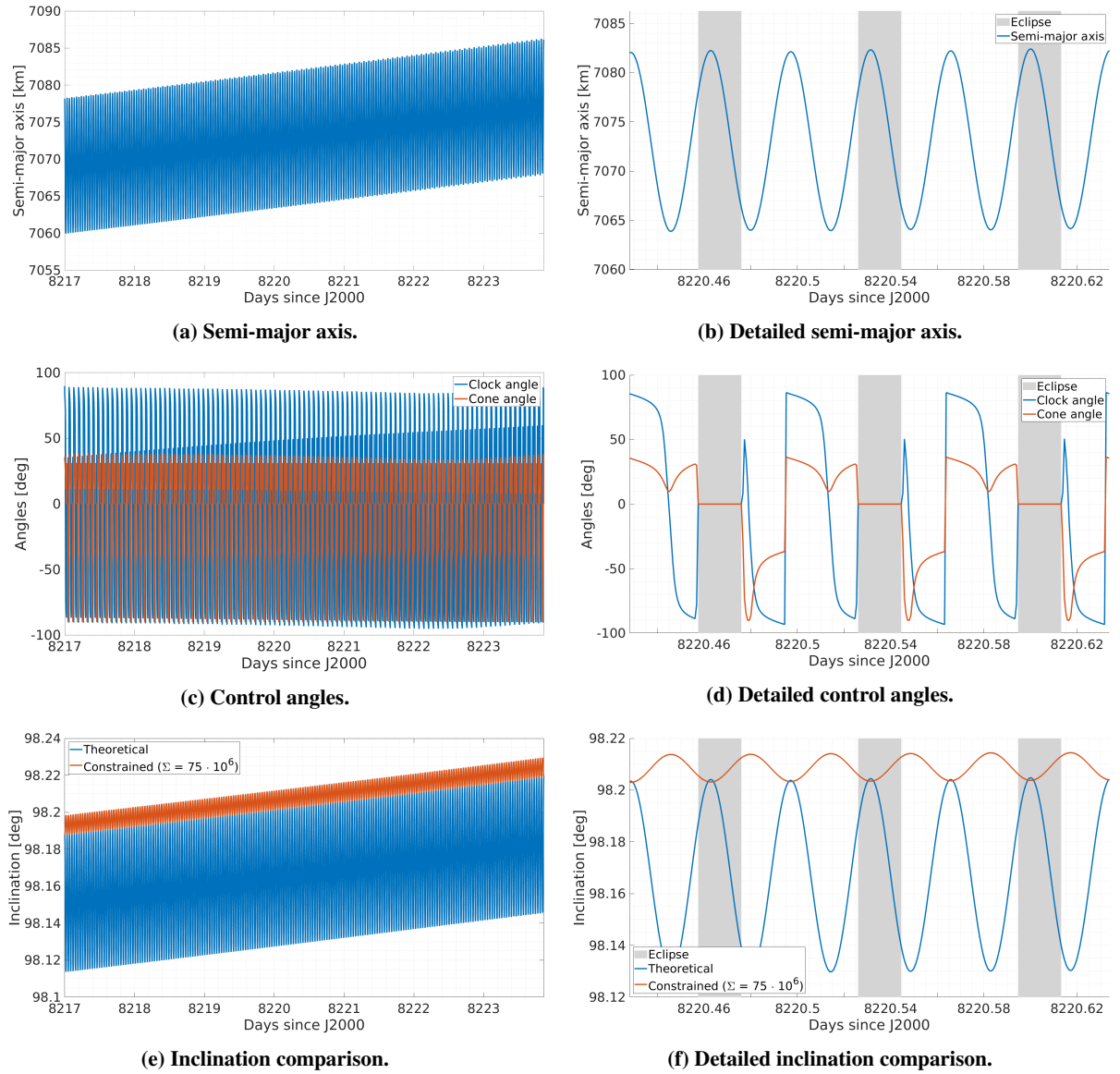
**Fig. 7 Three-dimensional geostationary orbit: 500 orbital revolutions.**

#### E. ACS3 mission: 100 orbital revolutions

Previously, in Section IV.C, the ACS3 mission was investigated for three orbital revolutions. Here, the case is extended to 100 orbital revolutions, with some more important differences. First, the number of orbital revolutions is increased to 100, and, second, all perturbing accelerations ( $J_2$ ,  $J_3$ ,  $J_4$ , solar gravity, lunar gravity) and the Sun-synchronous inclination constraint ( $\Sigma = 75 \times 10^6$ ) are taken into account. The resulting control angles, semi-major axis increase and inclination profile are shown in Figure 8. The figures in the right column again show a detail of the figures in the left column. A final semi-major axis increase of 7.85 km is found and the inclination constraint is met at  $\Delta i / \Delta i_t = 99.4\%$ . Considering that different weights have been used in the cost function for the three and 100 revolution cases, this result suggests that the optimal value for the weight is problem-dependent. Table 7 indeed shows that for the three-revolution case the constraint is met at 82.1% with a weight of  $\Sigma = 75 \times 10^6$ . The semi-major axis increase which is achieved by the sail is significantly smaller than the increase found at geostationary altitude in Section IV.D, but the time in which the increase is achieved is also very different. Here, the almost eight kilometre increase is achieved in approximately seven days, compared to over 500 days for the geostationary case. Additionally, the perturbing accelerations have a different magnitude and the Sun-synchronous constraint negatively affects the semi-major axis increase. When taking these more challenging conditions into account, the solar sail is still able to achieve a significant increase in semi-major axis while the sailcraft is kept under Sun-synchronous conditions.

Several attempts have been made to increase the number of revolutions to one month of elapsed time, which would

equate to approximately 452 orbital revolutions. However, no successful iterations were found by DDP with the settings mentioned in Section IV.B. Increasing the reduction ratio tolerance,  $\rho$ , leads to the algorithm finding successful iterations and eventually to convergence, but the results show discontinuities in the controls and the algorithm is not able to meet the constraints. Attempts to use the result of an unconstrained optimisation as initial guess to the constrained optimisation also proved unsuccessful. Careful tuning of both the weight on the constraint and the optimisation settings might be needed to find a combination of settings that works for the many-revolution constrained optimisation case, which is deemed beyond the scope of this article but is recommended for future investigations into the applicability of DDP for many-revolution constraint optimisation.



**Fig. 8 ACS3 mission: 100 orbital revolutions.**

## V. Conclusions

In this work, the use of differential dynamic programming (DDP) to find optimal Earth-centred solar-sail trajectories has been investigated. First, the DDP algorithm has been used on a two-dimensional case to maximise the specific orbital energy of a satellite in geostationary equatorial orbit (GEO) for three orbital revolutions when the ecliptic and equatorial planes are assumed to coincide. For this case and starting from a zero-initial guess for the states and controls, the DDP algorithm has been proven to find more optimal solutions than state-of-the-art optimisation schemes in the form of locally optimal steering laws. This better optimum comes at the cost of an increased computational time, but DDP also has the benefit of being able to solve more complex and constrained Earth-centred solar-sail optimisation problems. Significant reductions in runtime can be achieved when using the locally optimal steering law as the initial guess for the DDP algorithm.

The algorithm performs similarly optimal for a three-dimensional case (i.e., considering the actual obliquity of the ecliptic), where again the specific orbital energy of a satellite in GEO has been maximised. Significant increases in the semi-major axis of the orbit were obtained (208 km after three revolutions) with a state-of-the-art solar-sail lightness number, indicating that a solar sail is a viable form of propulsion for orbit raising at geostationary altitude. The three-dimensional geostationary case has also been successfully compared against the locally optimal steering laws for further validation purposes.

The effect of using different settings for the DDP algorithm has been investigated. The optimality tolerance and minimum trust region radius influence the final result the most, with a smaller optimality tolerance resulting in more optimal solutions, but at the cost of an increase in runtime. The minimum trust region radius should be kept at zero. Different algorithms to solve the trust region quadratic subproblem were also tested. A hessian shifting technique obtained the most optimal results while also requiring the least computational time. Monotonic basin hopping (MBH) has been used to find globally optimal solutions with DDP in the inner loop of the algorithm. However, MBH did not significantly improve on the trajectory found by DDP.

By taking eclipses into account, optimal orbit-raising solutions were also found for NASA's upcoming ACS3 mission, which is assumed to be launched into a noon-midnight Sun-synchronous orbit. Additionally, constraints were successfully implemented to force the sailcraft to keep its Sun-synchronous attribute during the orbit raising manoeuvre. Alongside the sail's ability to keep the sailcraft under Sun-synchronous conditions, a notable increase in the semi-major axis of 0.23 km was obtained after three orbital revolutions.

As known from the literature, the effect of perturbing accelerations, such as  $J_2$ ,  $J_3$ ,  $J_4$ , lunar and solar gravity is shown to be different for the GEO case compared to the ACS3 mission due to the different altitudes regimes in which both cases operate. For the ACS3 mission,  $J_2$  was demonstrated to have the largest effect on the final semi-major axis, while for the GEO case lunar gravity is the most dominant perturbation. However, irregardless of which perturbations are included in the dynamical model, DDP is able to find optimal trajectories.

Finally, DDP has been proven to be robust for many-revolution solar-sail trajectories around the Earth. The GEO case has been optimised for 500 orbital revolutions and the ACS3 mission (including constraints) for 100 orbital revolutions. Increasing the number of orbital revolutions for the ACS3 mission beyond 100 requires careful tuning of the algorithm

to find successful iterations. To conclude, DDP is a viable tool for constrained and unconstrained many-revolution Earth-centred solar-sail trajectory optimisation which allows to obtain more or equally optimal solutions compared to state-of-the-art locally optimal steering laws.

## References

- [1] Kelly, P., and Bevilacqua, R., “Geostationary debris mitigation using minimum time solar sail trajectories with eclipse constraints,” *Optimal Control Applications and Methods*, Vol. 42, No. 1, 2021, pp. 279–304. <https://doi.org/10.1002/oca.2676>.
- [2] Kelly, P., and Bevilacqua, R., “An optimized analytical solution for geostationary debris removal using solar sails,” *Acta Astronautica*, Vol. 162, 2019, pp. 72–86. <https://doi.org/10.1016/j.actaastro.2019.05.055>.
- [3] Macdonald, M., and McInnes, C. R., “Realistic Earth Escape Strategies for Solar Sailing,” *Journal of Guidance, Control and Dynamics*, Vol. 28, No. 2, 2005. <https://doi.org/10.2514/1.5165>.
- [4] Green, A. J., “Optimal escape trajectory from a high earth orbit by use of solar radiation pressure.” Ph.D. thesis, Massachusetts Institute of Technology, 1977.
- [5] Guerrant, D., and Heaton, A., “Earth escape capabilities of the heliogyro solar sail,” *Advances in the Astronautical Sciences*, Vol. 150, 2014, pp. 639–658.
- [6] Heiligers, J., Ceriotti, M., McInnes, C. R., and Biggs, J. D., “Displaced Geostationary Orbit Design Using Hybrid Sail Propulsion,” *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 6, 2011. <https://doi.org/10.2514/1.53807>.
- [7] McKay, R. J., Macdonald, M., Biggs, J., and McInnes, C., “Survey of Highly Non-Keplerian Orbits with Low-Thrust Propulsion,” *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 3, 2011, pp. 645–666. <https://doi.org/10.2514/1.52133>.
- [8] Fu, B., Sperber, E., and Eke, F., “Solar sail technology—A state of the art review,” *Progress in Aerospace Sciences*, Vol. 86, 2016. <https://doi.org/10.1016/j.paerosci.2016.07.001>.
- [9] Tsuda, Y., Mori, O., Funase, R., Sawada, H., Yamamoto, T., Saiki, T., Endo, T., Yonekura, K., Hoshino, H., and Kawaguchi, J., “Achievement of IKAROS — Japanese deep space solar sail demonstration mission,” *Acta Astronautica*, Vol. 82, No. 2, 2013. <https://doi.org/10.1016/j.actaastro.2012.03.032>.
- [10] Wilkie, W. K., “Overview of the NASA Advanced Composite Solar Sail System (ACS3) Technology Demonstration Project,” *AIAA Scitech 2021 Forum*, 2021. <https://doi.org/10.2514/6.2021-1260>.
- [11] Dachwald, B., “Optimization of very-low-thrust trajectories using evolutionary neurocontrol,” *Acta Astronautica*, Vol. 57, 2005. <https://doi.org/10.1016/j.actaastro.2005.03.004>.
- [12] Aziz, J. D., “Low-Thrust Many-Revolution Trajectory Optimization,” , 2018. URL [https://scholar.colorado.edu/concern/graduate\\_thesis\\_or\\_dissertations/jq085k18x](https://scholar.colorado.edu/concern/graduate_thesis_or_dissertations/jq085k18x).
- [13] McInnes, C. R., *Solar sailing: technology, dynamics and mission applications*, Springer Science & Business Media, 2004.
- [14] Gao, Y., “Near-Optimal Very Low-Thrust Earth-Orbit Transfers and Guidance Schemes,” *Journal of Guidance, Control and Dynamics*, Vol. 30, No. 2, 2007. <https://doi.org/10.2514/1.24836>.



- [15] Macdonald, M., and McInnes, C. R., "Analytical Control Laws for Planet-Centered Solar Sailing," *Journal of Guidance, Control and Dynamics*, Vol. 28, No. 5, 2005. <https://doi.org/DOI:10.2514/1.11400>.
- [16] Jacobson, D. H., and Mayne, D. Q., "Differential dynamic programming," 1970.
- [17] Lantoine, G., and Russell, R. P., "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 1: Theory," *Journal of Optimization Theory and Applications*, Vol. 154, 2012. <https://doi.org/10.1007/s10957-012-0039-0>.
- [18] Lantoine, G., and Russell, R. P., "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 2: Application," *Journal of Optimization Theory and Applications*, Vol. 154, 2012. <https://doi.org/10.1007/s10957-012-0038-1>.
- [19] Aziz, J. D., Parker, J. S., Scheeres, D. J., and Englander, J. A., "Low-thrust many-revolution trajectory optimization via differential dynamic programming and a sundman transformation," *The Journal of the Astronautical Sciences*, Vol. 65, No. 2, 2018, pp. 205–228.
- [20] Leary, R. H., "Global optimization on funneling landscapes," *Journal of Global Optimization*, Vol. 18, No. 4, 2000, pp. 367–383.
- [21] Montenbruck, O., and Gill, E., *Satellite Orbits, Models, Methods, Applications*, Springer, 2001.
- [22] Mengali, G., and Quarta, A. A., "Optimal Three-Dimensional Interplanetary Rendezvous Using Nonideal Solar Sail," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 1, 2005. <https://doi.org/10.2514/1.8325>.
- [23] Dachwald, B., MacDonald, M., McInnes, C. R., Mengali, G., and Quarta, A. A., "Impact of Optical Degradation on Solar Sail Mission Performance," *Journal of Spacecraft and Rockets*, Vol. 44, No. 4, 2007. <https://doi.org/10.2514/1.21432>.
- [24] Heiligers, J., Diedrich, B., Derbes, W., and McInnes, C., "Sunjammer: preliminary end-to-end mission design," *AIAA/AAS Astrodynamics Specialist Conference*, 2014. <https://doi.org/10.2514/6.2014-4127>.
- [25] Aziz, J., Scheeres, D., Parker, J., and Englander, J., "A smoothed eclipse model for solar electric propulsion trajectory optimization," *Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan*, 2019, pp. 17–181. <https://doi.org/10.2322/tastj.17.181>.
- [26] Spencer, D. A., Johnson, L., and Long, A. C., "Solar sailing technology challenges," *Aerospace Science and Technology*, Vol. 93, 2019. <https://doi.org/10.1016/j.ast.2019.07.009>.
- [27] Kéchichian, J. A., "Inclusion of Higher Order Harmonics in the Modeling of Optimal Low-Thrust Orbit Transfer," *The Journal of the Astronautical Sciences*, Vol. 56, No. 1, 2008. <https://doi.org/0.1007/BF03256541>.
- [28] Wakker, K. F., *Fundamentals of Astrodynamics*, Institutional Repository Library Delft University of Technology, 2015.
- [29] Walker, M. J. H., Ireland, B., and Owens, J., "A set of modified equinoctial elements," *Celestial Mechanics*, Vol. 36, 1985.
- [30] Betts, J. T., "Optimal low-thrust orbit transfers with eclipsing," *Optimal Control Applications and Methods*, Vol. 36, No. 2, 2015, pp. 218–240. <https://doi.org/https://doi.org/10.1002/oca.2111>.

- [31] Bellman, R., and Kalaba, R. E., *Dynamic programming and modern control theory*, Vol. 81, Citeseer, 1965.
- [32] Pellegrini, E., and Russel, R. P., “On the Computation and Accuracy of Trajectory State Transition Matrices,” *Journal of Guidance, Control and Dynamics*, Vol. 39, 2016. <https://doi.org/10.2514/1.G001920>.
- [33] Dagum, L., and Menon, R., “OpenMP: an industry standard API for shared-memory programming,” *IEEE computational science and engineering*, Vol. 5, No. 1, 1998, pp. 46–55.
- [34] Conn, A. R., Gould, N. I., and Toint, P. L., *Trust region methods*, SIAM, 2000.
- [35] Wertz, J. R., *Spacecraft Orbit and Attitude Systems, Orbit & Constellation Design & Management*, Microcosm Press and Springer, 2009.

## Conclusions and recommendations

The aim of this work was to obtain, for the first time, globally optimal Earth-centred solar-sail trajectories through the use of differential dynamic programming (DDP). This objective is twofold, since DDP has never been used for solar-sail trajectory optimisation before. First, DDP's ability to deal with solar-sail trajectories is assessed, after which the algorithm can be used for a variety of different solar-sail mission scenarios under different dynamical conditions. The conclusions of this work are presented in Section 3.1 by answering the research questions as defined in Section 1.2, followed by recommendations for future research in Section 3.2.

### 3.1. Conclusions

The research questions of Section 1.2 were structured as four main questions, with one of the questions having several subquestions. The first two questions focus on the theoretical aspects of DDP, whereas the other two questions focus on the performance and usability of DDP for different solar-sail mission scenarios. All questions are answered separately in this section.

**1. Does solar-sail propulsion impose any constraints or limitations that prevent the use of DDP?**

The short answer to this research question is 'no', as long as the derivatives of the equations of motion are defined correctly in the dynamics matrix and tensor for optimisation, DDP is able to optimise solar-sail trajectories. There is one limitation of DDP that is not related directly to the solar-sail acceleration but towards general trajectory optimisation. DDP requires the equations of motion to be twice differentiable. This means DDP cannot handle high-precision ephemerides or atmospheric density models that make use of tabulated data.

**2. What is the effect of different settings for the optimisation algorithm on the optimality of the solution?**

Different optimisation settings can have a significant impact on the optimality of the solution as well as on the runtime of the algorithm. The effect for varying settings has been investigated for the case of orbit raising at geostationary altitude within three orbital revolutions. Two settings appeared to have the largest influence on the solution: the minimum trust region radius, which should be kept to zero, and the optimality tolerance. The smaller the optimality tolerance, the more optimal the solution is. However, this comes at the cost of increased runtime. Different algorithms to solve the trust region quadratic subproblem were also tested. A hessian shifting technique obtained the most optimal results and also required the least computational time. Monotonic basin hopping (MBH) has been used to find globally optimal solutions with DDP in the inner loop of the algorithm. However, MBH did not significantly improve on the trajectory found by DDP while significantly increasing the runtime of the algorithm.

**3. How do globally optimal Earth-centred solar-sail trajectories, when optimised using DDP, perform compared to locally optimal steering laws?**

DDP has been compared against the solutions of locally optimal steering laws for the case of orbit raising in geostationary equatorial orbit (GEO, 42,241 km altitude, zero inclination and eccentricity). First, a two-dimensional case was considered (ecliptic and equatorial planes coincide), where a solar-sail lightness number of  $\beta = 0.17$  was used and the trajectory was optimised for three orbital revolutions. The final semi-major axis found by the DDP algorithm was approximately 80 m larger than the one found from the locally optimal steering law. Next, a three-dimensional case was considered (accounting for the obliquity of the ecliptic), where a solar-sail lightness number of  $\beta = 0.0011$  was used and the trajectory was optimised for 500 orbital revolutions. Here, the final semi-major axis found by the DDP algorithm was approximately one kilometer smaller than the one found from the locally optimal steering law. Note that the DDP algorithm may find a better solution when a stricter optimality tolerance is used. The solution from DDP comes at the cost of an increased computational time compared to the solutions of the steering laws. However, DDP has the ability to deal with more complex problems where terminal and path-inequality constraints are present. For such problems the increase in computation effort is justified.

#### 4. What is the optimal solar-sail attitude for planet-centred solar-sail trajectories under different dynamical conditions around the Earth?

The answer to this research question is based on the outcome of the subquestions. After the subquestions have been answered, this overarching research question will be answered as well.

##### (a) What is the optimal solar-sail attitude for orbit raising at geostationary altitude?

To answer this question, the solar-sail attitude for a sailcraft in GEO has been optimised to maximise the specific orbital energy. First, a solar-sail lightness number of  $\beta = 0.011$  was used and a trajectory spanning over three orbital revolutions was considered. The resulting control profile varies between  $+90^\circ$  and  $-90^\circ$  with the position of the sailcraft in its orbit and approximately repeats itself every orbital revolution. The control profile for both angles is nonlinear. The clock angle experiences a jump from approximately  $-90^\circ$  to  $+90^\circ$  twice every orbital revolution, whereas the cone angle experiences a jump from approximately  $-50^\circ$  to  $+50^\circ$  twice every orbital revolution. The resulting semi-major axis increases when the satellite is in the region of the orbit where the satellite moves away from the Sun, and is stationary in the region where the satellite moves towards the Sun. Next, a solar-sail lightness number of  $\beta = 0.0011$  was used and a trajectory over 500 orbital revolutions was optimised, considering various perturbing accelerations ( $J_2$ ,  $J_3$ ,  $J_4$ , lunar and solar gravity). The optimal control for the first three revolutions is similar to the control found for the three-revolution case. For subsequent orbital revolutions, the control profile adapts itself to the changing position of the Sun. After 500 orbital revolutions, a semi-major axis increase of 3793.1 km was found. This increase is very large, especially considering that a solar-sail lightness number  $\beta = 0.0011$  is an order of magnitude smaller than the current state-of-the-art. Therefore, a solar sail can be considered a viable method of propulsion for orbit raising at geostationary altitude.

##### (b) What is the optimal solar-sail attitude for raising the orbit altitude of a low-Earth orbit under Sun-synchronous conditions?

To answer this question, the solar-sail attitude for a sailcraft in a 700 km altitude noon-midnight Sun-synchronous orbit, with a solar-sail lightness number of  $\beta = 0.0084$ , has been optimised to maximise the specific orbital energy. This mission scenario is representative of NASA's upcoming ACS3 mission. A constraint was successfully implemented to force the final orbit, after an increase in semi-major axis was achieved, to be Sun-synchronous as well. During eclipses, the control angles are set to zero and the semi-major axis does not change. When the satellite is not in eclipse, the control angles contribute to a change in inclination and semi-major axis. The clock angle varies every revolution between  $+90^\circ$  and  $-90^\circ$ , whereas the cone angle varies between approximately  $+40^\circ$  and  $-90^\circ$ . The rate of change of both the cone and clock angles are large during each orbital revolution, meaning that there are several sharp transitions. Before and after every sharp transition the angles commonly experience a smaller rate of change. The control profile is approximately repeated with every orbital revolution. Different weights can be placed on the inclination constraint, where a larger weight will cause the final inclination to be closer to the theoretical Sun-synchronous inclination, but comes at the cost of a smaller semi-major axis increase. Next, the number of orbital revolutions has been increased to 100 and various perturbing accelerations were imposed on the sailcraft ( $J_2$ ,  $J_3$ ,  $J_4$ , lunar and solar gravity). The control profile is similar to the control profile of the three-revolution case. The semi-major axis increases with every orbital revolution. After 100 orbital revolutions, a semi-major axis increase of 7.85 km was found, while the sailcraft retained its Sun-synchronous conditions. This is a significant increase in the semi-major axis, especially considering the challenging conditions of low-Earth orbit regarding perturbing accelerations and eclipses.

##### (c) What is the effect of perturbing accelerations on the optimal trajectories?

The effect for various perturbing accelerations has been investigated for both the three-revolution geostationary case and for the ACS3 mission. The effect on the semi-major axis increase caused by considering various perturbing accelerations due to irregularities in the Earth's gravity field ( $J_2$ ,  $J_3$  and  $J_4$ ), lunar and solar gravity, is different for the geostationary case compared to the ACS3 mission. For the geostationary case, including lunar gravity has the largest effect on the final semi-major axis, whereas for the ACS3 mission the  $J_2$  effect has the largest influence. This difference can be attributed to the different altitudes at which the satellites orbit in the different scenarios. The difference in runtime for the DDP algorithm between considering different perturbing accelerations is negligible. Although different perturbing accelerations are dominant for the different cases, DDP is able to find the optimal trajectory regardless of which perturbations are included in the dynamical model. Note that the effect of atmospheric drag has not been investigated.

With the answers to the subquestions in mind, the main research question can be answered. The optimal solar-sail attitude varies greatly for different satellite trajectories, different dynamical models and different constraints. In general, the control profile repeats itself after an orbital revolution has been completed, while taking into account that the Sun is moving with respect to the Earth. Despite the greatly varying dynamical conditions, the DDP algorithm is able to find optima for all optimisation cases considered in this work.

## 3.2. Recommendations

Several recommendations for further research are formulated based on the findings of this work. The recommendations are formulated such that the DDP algorithm can be used for a wider range of solar-sail optimisation problems in the future. Therefore, proposals are made regarding the optimisation algorithm itself, the dynamical model as well as possible applications.

### DDP algorithm

DDP has the ability to deal with path-inequality constraints in the form of locally incurring costs at each stage, which have not been implemented in this work. For realistic solar-sail trajectories the rate by which the cone and clock angles can change is limited [23]. Local stage costs can be used as a penalty method for the algorithm to enforce a maximum rate of change for the control angles, which is something that can be important for realistic solar-sail trajectory optimisation.

The runtime of the optimisation algorithm can be a considerable factor when optimising many-revolution solar-sail trajectories. A straightforward way to decrease the runtime is to increase the number of logical threads on which the algorithm is run. Since the most computationally expensive part of the algorithm is parallelised, twice the computational power results in an approximate decrease of runtime by a factor two. Because increasing the number of threads is not always an option, significant reductions in runtime can also be achieved when using locally optimal steering laws as initial guess to the DDP algorithm.

### Dynamical model

The trajectory design that has been implemented in this work can be improved by increasing the fidelity of the dynamical model. Currently, an ideal solar-sail acceleration model is used, which assumes that all incoming radiation is specularly reflected. More realistic solar-sail acceleration models account for absorption, diffuse reflection and thermal emission of the incoming radiation (optical force model), sail billowing and degradation [3, 24, 25]. Furthermore, the eclipse model that is implemented only accounts for the Earth as occulting body. For Earth-centred orbits, the effect of the Moon as an occulting body can be investigated as well.

Perhaps the most significant improvement to the dynamical model can be made in the low-altitude regime, where atmospheric drag is the dominant perturbing acceleration. Atmospheric drag has not been included in this work, since DDP requires the equations of motion to be twice differentiable. Most models for the atmospheric density as a function of altitude make use of tabulated data, with the notable exception for the exponential density model, which is strongly simplified and only provides a very rough estimate [26]. A workaround has to be found which allows a high-fidelity density model to be twice differentiable.

### Possible applications

The optimisation problems that were considered in this work all had the specific orbital energy as main objective to maximise. The algorithm's ability to deal with other objectives, where different (combinations of) orbital elements are to be maximised or minimised, should be investigated. Formulating the problem in a way that propagation takes place until a desired state is reached is particularly interesting, since this would result in the number of stages for DDP becoming dynamic. Other extensions could include the use of multiple terminal constraints, where the sailcraft's final state is restricted in both position and velocity to match a target orbit (e.g., a transfer from geostationary transfer orbit (GTO) to GEO), or the implementation of hybrid thrust, where a combination of a solar sail and solar-electric propulsion is used to accelerate the spacecraft. Hybrid thrust is a very attractive concept, since the drawbacks of the individual forms of propulsion cancel each other [27]. The solar-electric propulsion allows for a component of the acceleration vector to be in the direction from the satellite towards the Sun, which is impossible with a solar sail. Application for hybrid thrust include, but are not limited to, Earth-observation [27], displaced geostationary orbits [28] and interplanetary trajectories [29].



# A

## Verification and validation

This appendix describes the methods that have been used to verify and validate the software that has been developed. First, the dynamical model is verified in Section A.1. The numerical integrator is verified and validated in Section A.2. Lastly, the optimisation algorithm itself is verified in Section A.3.

### A.1. Dynamical model

The dynamical model that has been used in this work consists of several elements. First, the accelerations acting on the satellite, with the exception of the solar-sail acceleration, are verified. Next, the solar-sail acceleration is treated separately. Finally, the eclipse model is verified.

#### A.1.1. Perturbing accelerations

The magnitude of the various accelerations acting on the satellite changes with the altitude of the satellite. The accelerations have been verified by comparing Figure A.1 with Figure 3.1 from Reference [26]. The magnitude of the acceleration resulting from the point-mass gravity (PM) from Earth, the Moon and the Sun, as well as the  $J_2$ ,  $J_3$  and  $J_4$  effect are shown as a function of altitude. The order of magnitude for all accelerations was found to be the same in Figure A.1 and Figure 3.1 from Reference [26]. Furthermore, the behaviour of the accelerations, whether they increase or decrease with altitude, was found to be similar for all terms. Therefore, the accelerations shown in Figure A.1 are assumed to be correctly implemented.

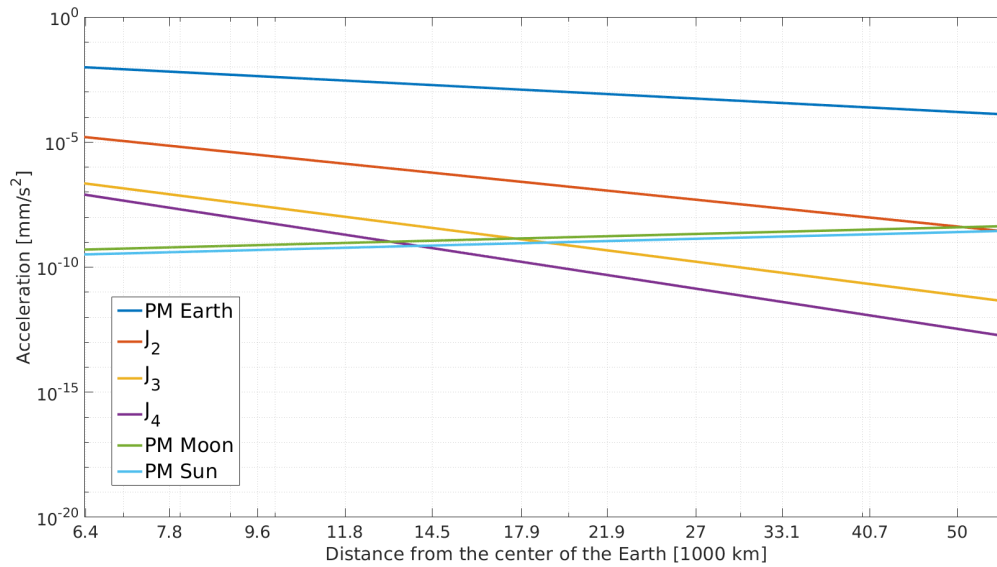


Figure A.1: Magnitude of perturbing accelerations at different altitudes.

### A.1.2. Solar-sail acceleration

This work uses an ideal solar-sail acceleration model, which implies that all incoming radiation is specularly reflected and the acceleration acts along the vector normal to the sail surface. In Figure A.2a, from Reference [30], the magnitude of the forces along,  $F_s$ , and normal,  $F_l$ , to the Sun-sail line for a varying cone angle are given for an ideal solar sail at a distance of one AU from the Sun, with a solar-sail lightness number of  $\beta = 0.0363$  and a total sail area of  $1068 \text{ m}^2$ . The figure was reproduced in Figure A.2b, where exactly the same magnitude and direction of the solar-sail forces is observed. A similar procedure was performed for varying the solar-sail clock angle, which resulted in a figure similar to Figure A.2b, but inclined with respect to the  $(F_s, F_l)$  plane.

Verification not only took place by comparing images, also with the known fact that the cone angle that should maximise the force in the transverse direction is  $\alpha = 35.26^\circ$  [3]. The cone angle that maximises the force in the transverse direction using the developed software was found to be  $\alpha = 35.3^\circ$ . The difference is explained by the fact that tests were only conducted by taking steps in the angle of  $0.1^\circ$ , making  $35.3^\circ$  the closest possible to  $35.26^\circ$  that can be achieved.

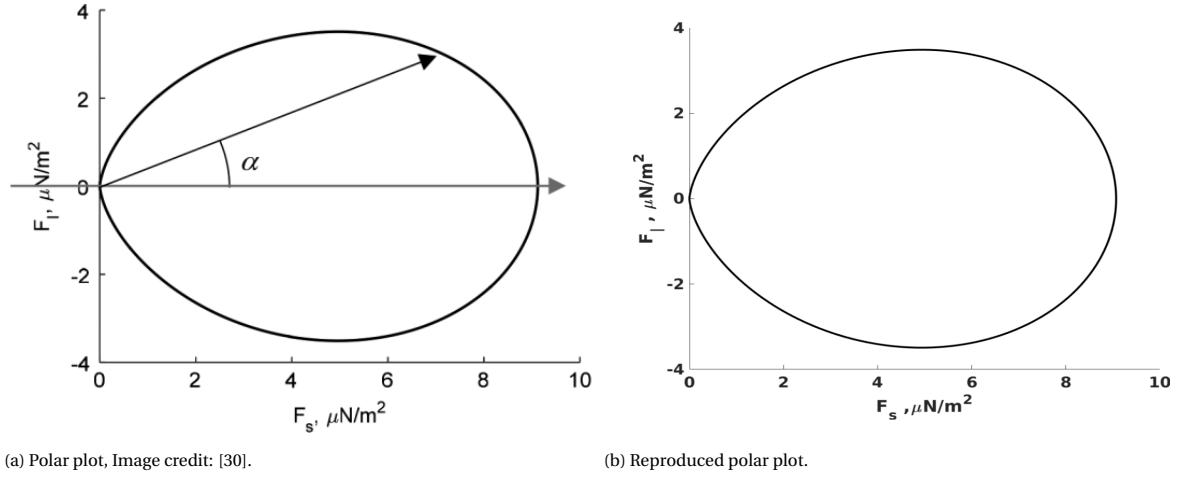


Figure A.2: Polar plots of solar-sail performance at 1 AU.

### A.1.3. Eclipses

Eclipses were modelled by using a Heaviside step function to approximate the sunlight fraction [22]. This model is twice differentiable, which is a requirement for the optimisation algorithm. In Figure A.3, the trajectory is visualised for a satellite at geostationary altitude. A two-dimensional dynamical model is assumed where the direction of sunlight is taken to coincide with the orbital plane and the Sun is located at a distance of one astronomical unit (AU) along the negative  $x$ -axis of the ECI frame. This configuration ensures that eclipses occur once per revolution, which is clearly visible from the absence of the solar-sail acceleration behind the Earth in Figure A.3.

There are two parameters of the eclipse model which can be tuned: the sharpness coefficient,  $c_s$ , and the transition coefficient,  $c_t$ . Figure 5.2 from Reference [22] shows the effect of varying these settings on the sunlight fraction. This figure has been reproduced in Figure A.4, which is exactly similar.

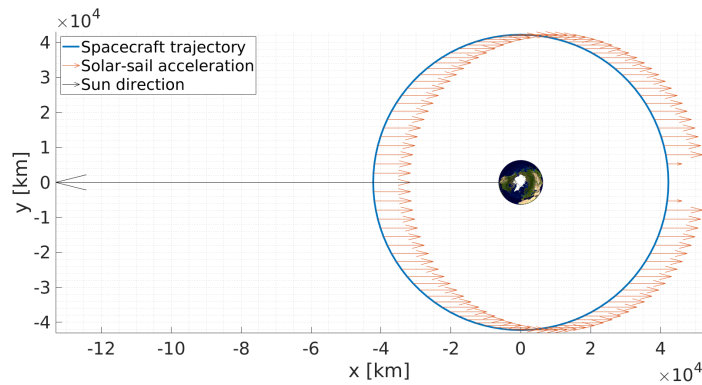
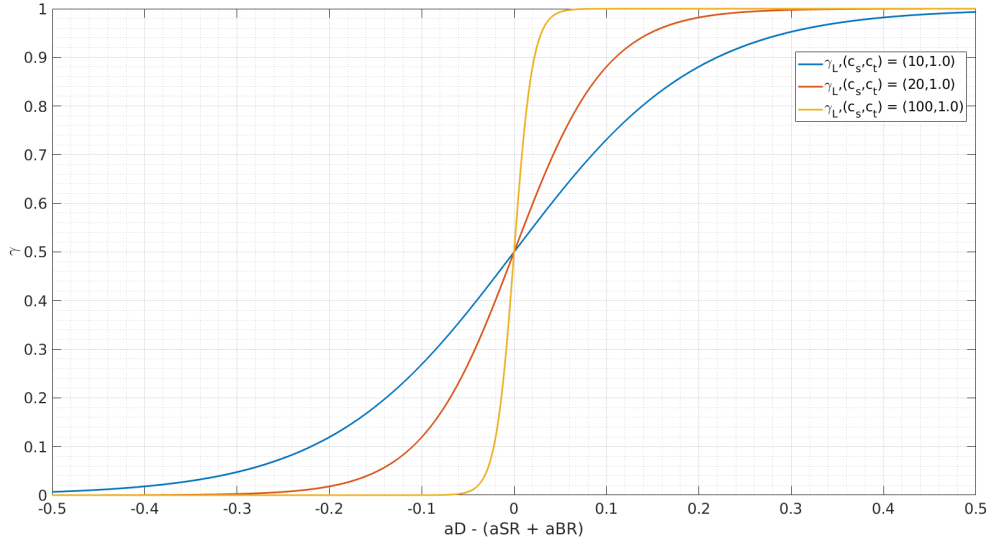
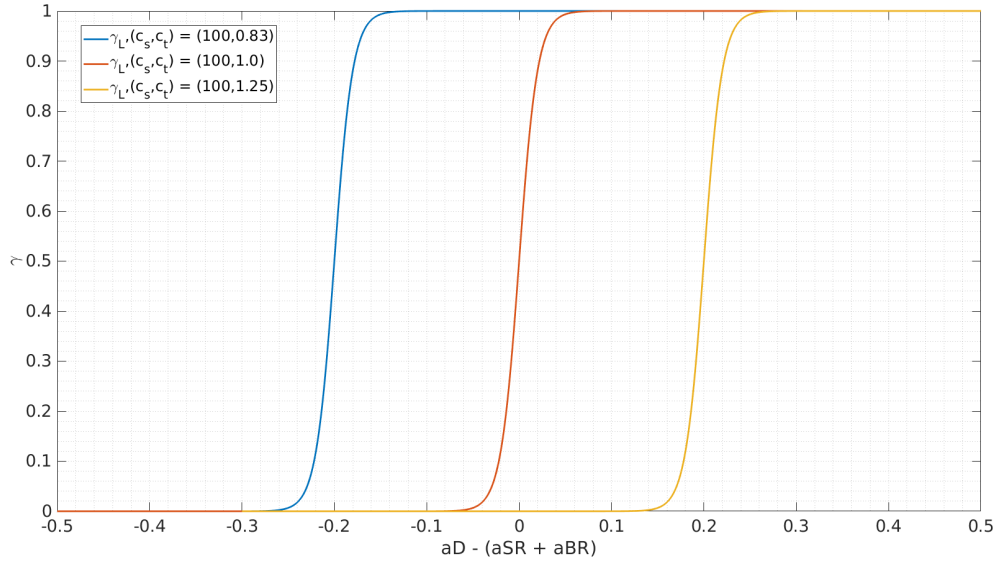


Figure A.3: Two-dimensional geostationary orbit with eclipse: one orbital revolutions.





(a) The sunlight fraction for different sharpness coefficients.



(b) The sunlight fraction for different transition coefficients.

Figure A.4: The sunlight fraction, modelled with a Heaviside step function, for different sharpness and transition coefficients.

## A.2. Numerical integration

In this work, a DOPRI8 fixed step size numerical integrator has been implemented. The software is first verified by integrating an analytical function for which the solution is known. Next, the integrator is compared with results from TU Delft's open-source TUDAT software package for the integration of a satellite trajectory<sup>1</sup>. Finally, the choice for the DOPRI8 scheme is validated by comparison with other numerical integrators.

### A.2.1. Analytical function

First, the implemented integration routine was tested by integrating a function of which the analytical solution is known,  $\int_0^{200} \exp(2 \cdot t) dt$ . The integral evaluates to  $\frac{1}{2} (\exp(200) - 1)$ . Numerical integration with timestep  $\Delta t = 0.0005$  lead to a value of  $2.61031 \cdot 10^{173}$ , a difference of 0.01% with the analytical value. It is expected that there is some difference between the analytical and numerical result, since rounding and truncation errors play a role for the numerical scheme. A second analytical function on which the integrator was tested is  $\int_0^{200} 5 \cdot t^2$ , again with timestep  $\Delta t = 0.0005$ . The result found

<sup>1</sup> Documentation at <http://tudat.tudelft.nl>; Code at <http://github.com/tudat/tudatBundle>; Accessed: 16/02/2021.

with the numerical integrator is 13333300, while the analytical solution is 13333333.33; a difference of 0.00025%. Lastly, a linear function was integrated, where the numerical integrator found the analytical solution.

### A.2.2. Verification with TUDAT

A satellite, with the initial condition as specified in Table A.1, was propagated for a period of approximately 24 days using the developed numerical integrator and with the TUDAT software package. The magnitude of the position difference at every point in time is visualised in Figure A.5. The difference in position increases with time, and some periodicity is observed. This difference can be explained by a difference in precision of the gravitational parameter of Earth that is used. The exact gravitational parameter that is used in TUDAT is not known. In the author's code,  $\mu_{\oplus} = 3.986004118 \cdot 10^{14}$  was used. Changing this parameter to a lower precision, i.e.  $\mu_{\oplus} = 3.986 \cdot 10^{14}$  exaggerates the behaviour seen in Figure A.5. From this, it is concluded that the error occurs due to a difference in precision of the gravitation parameter of the Earth that is used, and that there is no error in the software itself.

Orbital element	Value
$a$	6787746.891 m
$e$	0.000731104
$i$	51.68714486°
$\Omega$	127.5486706°
$\omega$	74.21987137°
$\nu$	24.10027677°

Table A.1: Initial conditions for satellite propagation testing.

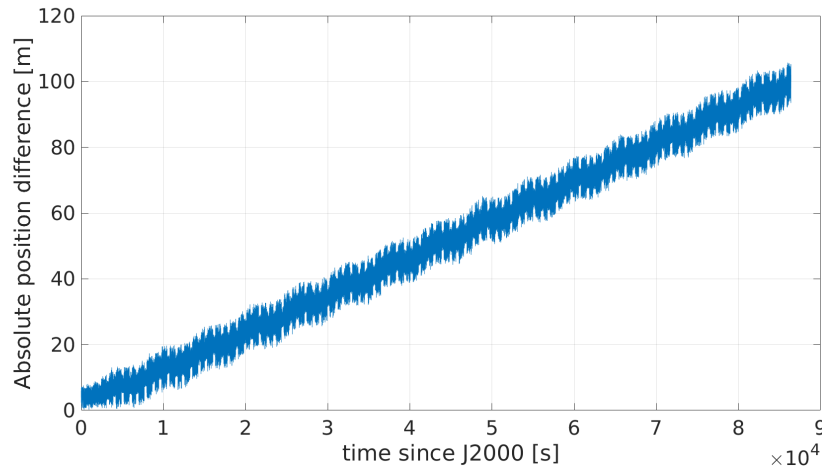


Figure A.5: Position difference of the satellite.

### A.2.3. Validation with TUDAT

The choice to use a DOPRI8 integration scheme has been validated by comparison of the scheme's performance with other numerical integrators. This comparison was carried out in TUDAT, since TUDAT has many different integration routines already implemented. First, a benchmark solution is needed, to which the performance of the various integrators can be compared. The benchmark is obtained by integrating the satellite trajectory with a very small timestep, but it is important to not take the timestep too small such that rounding errors start dominating the solution. Therefore, a timestep of  $\Delta t = 0.1$  s was used. The DOPRI8 routine was compared against the Runge-Kutta 4(5), Runge-Kutta 5(6), Adam-Bashforth (6-11), Adam-Bashforth (8-8), Bulirsch-Stoer (5) and Bulirsch-Stoer (10) integration routines, all with fixed timesteps.

Propagating the satellite, with orbital elements as in Table A.1, with a timestep of  $\Delta t = 1$  s, the magnitude of the position difference at the final epoch between the benchmark solution and the other integrators are comparable, all within a range of 0.1 m. However, there is a significant difference in the number of function evaluations needed to obtain the solution. The Runge-Kutta (RK) methods require the least function evaluations, whereas the Bulirsch-Stoer (BS) methods

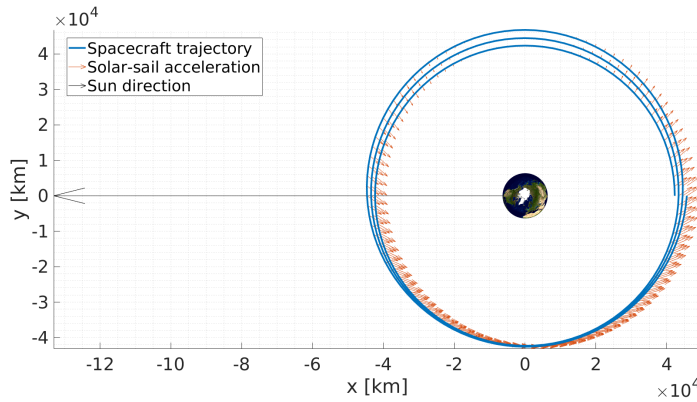
require one order of magnitude more evaluations. The number of function evaluations for the Adam-Bashforth (AB) and DOPRI8 methods are in between those needed for the RK and BS methods. The difference in performance between the various integration routines becomes larger when a larger timestep is used, i.e.  $\Delta t = 500$  s. Here, the difference in final position magnitude for the AB and RK methods are much larger compared to the DOPRI8 and BS methods. Although the DOPRI8 and BS methods find a comparable final solution, the BS methods need approximately an order of magnitude more function evaluations to get to this solution. Therefore, the choice to use DOPRI8 is valid.

## A.3. Optimisation

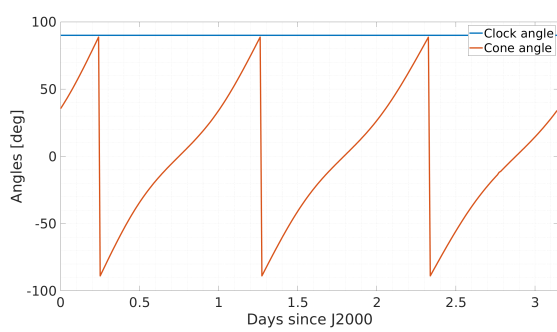
The optimisation algorithm that has been developed in this work has been verified by comparison with locally optimal steering laws for Earth-centred solar-sail trajectories from Reference [3]. Here, the specific orbital energy of a satellite placed at geostationary altitude (42,241 km altitude and zero eccentricity) was maximised over the course of three orbital revolutions. A two-dimensional dynamical model is assumed where the Sun is located at a distance of one astronomical unit (AU) along the negative  $x$ -axis of the ECI frame. Only point-mass gravity from the Earth and the solar-sail acceleration, assuming a solar-sail lightness number of 0.17, are acting on the sailcraft. Verification of the optimisation algorithm is performed separately for the cone and clock angles. First, the case is considered where the sailcraft is in an orbit with zero inclination, the clock angle,  $\delta$  is fixed at  $90^\circ$ , such that the control is solely determined by the cone angle,  $\alpha$ . Next, the case is considered where the sailcraft is in a polar orbit with an inclination of  $90^\circ$ , the cone angle,  $\alpha$ , is fixed at  $\arctan(1/\sqrt{2})$ , which is the optimal angle to maximise the acceleration in the transverse direction [3]. The control is then solely determined by the clock angle,  $\delta$ .

### A.3.1. Two-dimensional geostationary orbit: three orbital revolutions

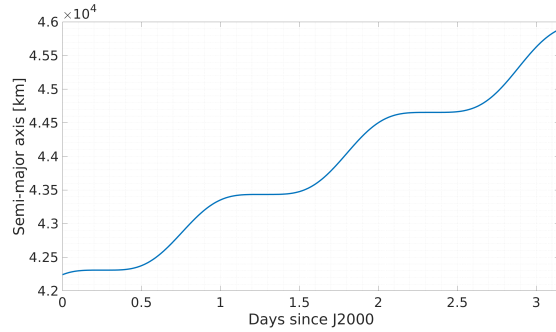
This verification case concerns the optimisation of the specific orbital energy of a spacecraft in geostationary orbit (42,241 km altitude, zero inclination and eccentricity) over the course of three orbital revolutions. As mentioned above, the control is solely determined by the sail cone angle,  $\alpha$ , as the clock angle,  $\delta$ , is fixed at  $90^\circ$ . For the cone angle, an initial guess of 0 is used at all stages. The optimal trajectory found by DDP is shown in Figure A.6. The control profile of the cone angle is very similar to the locally optimal steering law from Reference [3], the same holds for the resulting semi-major axis.



(a) Satellite trajectory.



(b) Solar-sail attitude angles.



(c) Semi-major axis.

Figure A.6: Two-dimensional geostationary orbit: DDP verification.

### A.3.2. Two-dimensional polar orbit: three orbital revolutions

This verification case concerns the optimisation of the specific orbital energy of a spacecraft in a polar orbit (42,241 km altitude,  $90^\circ$  inclination and zero eccentricity) over the course of three orbital revolutions. Again, as mentioned in the introduction of this section, the control is solely determined by the sail clock angle,  $\delta$ , as the cone angle,  $\alpha$ , is fixed at  $\arctan(1/\sqrt{2})$ . For the clock angle, an initial guess of 0 is used at all stages. The optimal trajectory found by DDP is shown in Figure A.7. From the direction of the solar-sail acceleration vector in Figure A.7a it is observed the the acceleration is now always in the direction of the velocity vector of the satellite. The control profile of the clock angle is different from what is obtained from the locally optimal steering law in Reference [3]. This is explained by the fact that the angles are defined differently, a transformation can be made as  $\delta = -(\delta_{\text{McInnes}} - 90^\circ)$ , where  $\delta_{\text{McInnes}}$  is the optimal clock angle from Reference [3]. With this transformation, the control profile is very similar to the locally optimal steering law from Reference [3]. The same hold for the resulting semi-major axis.

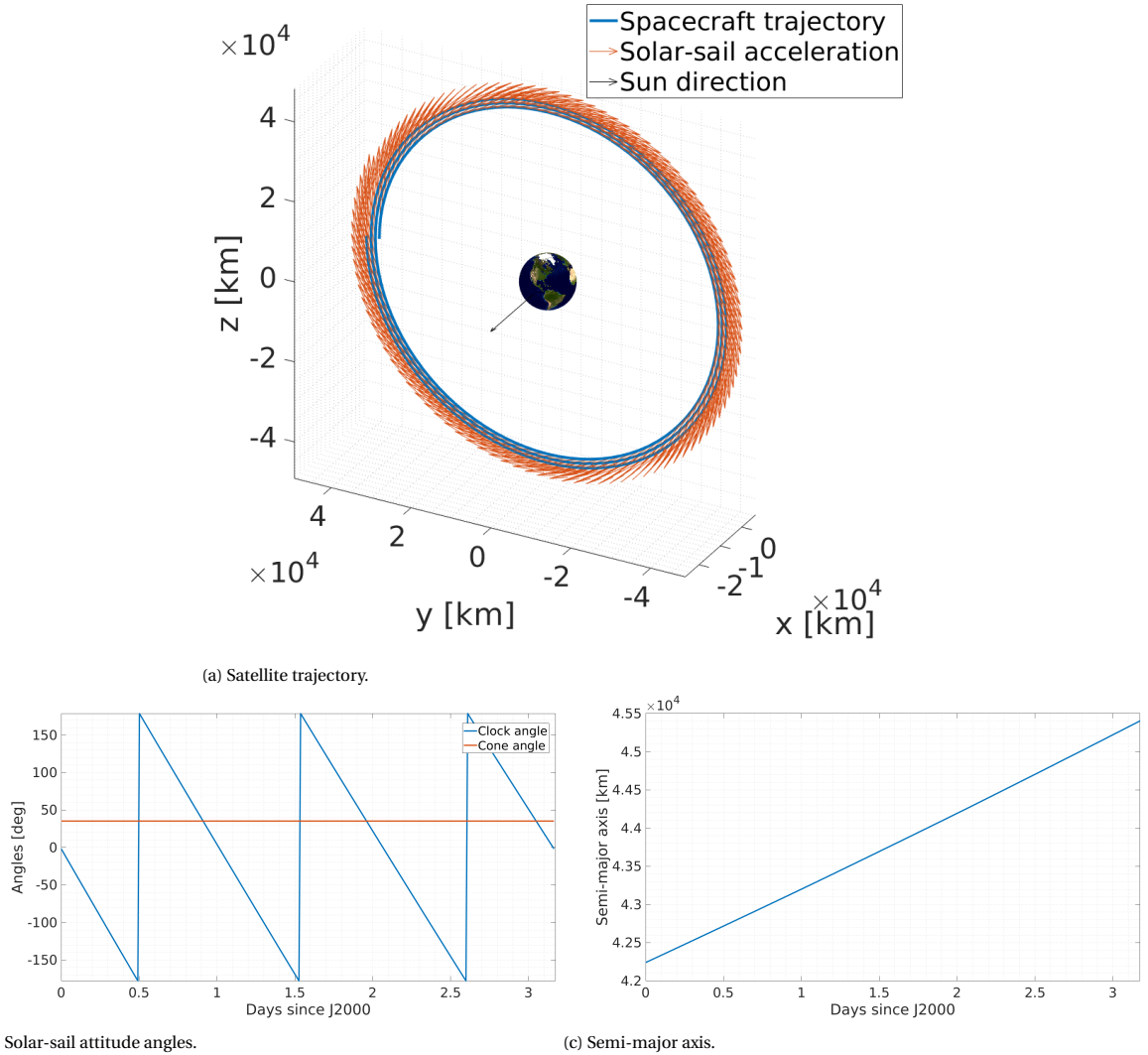


Figure A.7: Two-dimensional polar orbit: DDP verification.

# Bibliography

- [1] J. Kepler, F. Hammer, and W. Lehmann. "Joannis Kepleri mathematici Caesarei dissertatio cum nuncio sidereo nuper ad mortales misso à Galilaeo Galilaeo mathematico Patavina [lat. u. dt.] Pragae 1610: Sedesanus". 1610.
- [2] B. Fu, E. Sperber, and F. Eke. "Solar sail technology—A state of the art review". In: *Progress in Aerospace Sciences* 86, (2016). DOI: 10.1016/j.paerosci.2016.07.001.
- [3] C. R. McInnes. "Solar sailing: technology, dynamics and mission applications". Springer Science & Business Media, (2004).
- [4] Y. Tsuda, O. Mori, R. Funase, H. Sawada, T. Yamamoto, T. Saiki, T. Endo, K. Yonekura, H. Hoshino, and J. Kawaguchi. "Achievement of IKAROS — Japanese deep space solar sail demonstration mission". In: *Acta Astronautica* 82.2, (2013). DOI: 10.1016/j.actaastro.2012.03.032.
- [5] J. Mansell, D. A. Spencer, B. Plante, A. Diaz, M. Fernandez, J. Bellardo, B. Betts, and B. Nye. "Orbit and Attitude Performance of the LightSail 2 Solar Sail Spacecraft". In: *AIAA Scitech 2020 Forum*. (2020). DOI: 10.2514/6.2020-2177.
- [6] L. McNutt, L. Johnson, P. Kahn, J. Castillo-Rogez, and A. Frick. "Near-earth asteroid (NEA) scout". In: *AIAA Space 2014 Conference and Exposition*. (2014). DOI: 10.2514/6.2014-4435.
- [7] W. K. Wilkie. "Overview of the NASA Advanced Composite Solar Sail System (ACS3) Technology Demonstration Project". In: *AIAA Scitech 2021 Forum*. (2021). DOI: 10.2514/6.2021-1260.
- [8] B. Dachwald. "Optimization of very-low-thrust trajectories using evolutionary neurocontrol". In: *Acta Astronautica* 57, (2005). DOI: 10.1016/j.actaastro.2005.03.004.
- [9] J. D. Aziz, J. S. Parker, D. J. Scheeres, and J. A. Englander. "Low-thrust many-revolution trajectory optimization via differential dynamic programming and a sundman transformation". In: *The Journal of the Astronautical Sciences* 65.2, (2018), 205–228.
- [10] Y. Gao. "Near-Optimal Very Low-Thrust Earth-Orbit Transfers and Guidance Schemes". In: *Journal of Guidance, Control and Dynamics* 30.2, (2007). DOI: 10.2514/1.24836.
- [11] M. Macdonald and C. R. McInnes. "Analytical Control Laws for Planet-Centered Solar Sailing". In: *Journal of Guidance, Control and Dynamics* 28.5, (2005). DOI: DOI: 10.2514/1.11400.
- [12] P. Kelly and R. Bevilacqua. "Geostationary debris mitigation using minimum time solar sail trajectories with eclipse constraints". In: *Optimal Control Applications and Methods* 42.1, (2021), 279–304. DOI: 10.1002/oca.2676.
- [13] P. Kelly and R. Bevilacqua. "An optimized analytical solution for geostationary debris removal using solar sails". In: *Acta Astronautica* 162, (2019), 72–86. DOI: 10.1016/j.actaastro.2019.05.055.
- [14] M. Macdonald and C. R. McInnes. "Realistic Earth Escape Strategies for Solar Sailing". In: *Journal of Guidance, Control and Dynamics* 28.2, (2005). DOI: 10.2514/1.5165.
- [15] A. J. Green. "Optimal escape trajectory from a high earth orbit by use of solar radiation pressure." PhD thesis. Massachusetts Institute of Technology, (1977).
- [16] D. Guerrant and A. Heaton. "Earth escape capabilities of the heliogyro solar sail". In: *Advances in the Astronautical Sciences* 150, (2014), 639–658.
- [17] D. H. Jacobson and D. Q. Mayne. "Differential dynamic programming". In: (1970).
- [18] G. Lantoine and R. P. Russell. "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 1: Theory". In: *Journal of Optimization Theory and Applications* 154, (2012). DOI: 10.1007/s10957-012-0039-0.
- [19] C. Colombo, M. Vasile, and G. Radice. "Optimal low-thrust trajectories to asteroids through an algorithm based on differential dynamic programming". In: *Celestial Mechanics and Dynamical Astronomy* 105, (75 2009). DOI: 10.1007/s10569-009-9224-3.

- [20] M. D. Rayman, T. C. Fraschetti, C. A. Raymond, and C. T. Russell. "Dawn: A mission in development for exploration of main belt asteroids Vesta and Ceres". In: *Acta Astronautica* 58.11, (2006), 605–616. ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2006.01.014.
- [21] G. Whiffen. "Mystic: Implementation of the Static Dynamic Optimal Control Algorithm for High-Fidelity, Low-Thrust Trajectory Design". In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. (2012). DOI: 10.2514/6.2006-6741.
- [22] J. D. Aziz. "Low-Thrust Many-Revolution Trajectory Optimization". PhD thesis. 2018. URL: [https://scholar.colorado.edu/concern/graduate\\_thesis\\_or\\_dissertations/jq085k18x](https://scholar.colorado.edu/concern/graduate_thesis_or_dissertations/jq085k18x).
- [23] J. Orphee, B. Diedrich, B. Stiltner, C. Becker, and A. Heaton. "Solar sail attitude control system for the NASA Near Earth Asteroid Scout Mission". In: *The Fourth International Symposium on Solar Sailing*. Vol. 17091. (2017).
- [24] G. Mengali and A. A. Quarta. "Optimal Three-Dimensional Interplanetary Rendezvous Using Nonideal Solar Sail". In: *Journal of Guidance, Control, and Dynamics* 28.1, (2005). DOI: 10.2514/1.8325.
- [25] B. Dachwald, M. MacDonald, C. R. McInnes, G. Mengali, and A. A. Quarta. "Impact of Optical Degradation on Solar Sail Mission Performance". In: *Journal of Spacecraft and Rockets* 44.4, (2007). DOI: 10.2514/1.21432.
- [26] O. Montenbruck and E. Gill. "Satellite Orbits, Models, Methods, Applications". Springer, (2001). ISBN: 3-540-67280-X.
- [27] M. Ceriotti and C. R. McInnes. "Hybrid solar sail and solar electric propulsion for novel Earth observation missions". In: *Acta Astronautica* 69.9, (2011), 809–821. ISSN: 0094-5765. DOI: <https://doi.org/10.1016/j.actaastro.2011.06.007>.
- [28] Y. Liu, J. Heiligers, and M. Ceriotti. "Loosely-displaced geostationary orbits with hybrid sail propulsion". In: *Aerospace Science and Technology* 79, (2018), 105–117. ISSN: 1270-9638. DOI: <https://doi.org/10.1016/j.ast.2018.05.034>.
- [29] G. Mengali and A. A. Quarta. "Trajectory design with hybrid low-thrust propulsion system". In: *Journal of Guidance, Control, and Dynamics* 30.2, (2007), 419–426. DOI: 10.2514/1.22433.
- [30] J. Heiligers, D. Guerrant, and D. Lawrence. "Exploring the Heliogyro's Orbital Control Capabilities for Solar Sail Halo Orbits". In: *Journal of Guidance, Control, And Dynamics* 40.10, (2017). DOI: 10.2514/1.G002184.