Springer Optimization and Its Applications 144

Giorgio Fasano
János D. Pintér *Editors*

Modeling and Optimization in Space Engineering

State of the Art and New Challenges

Springer

Springer Optimization and Its Applications

Volume 144

Aims and Scope

Optimization has been expanding in all directions at an astonishing rate during the last few decades. New algorithmic and theoretical techniques have been developed, the diffusion into other disciplines has proceeded at a rapid pace, and our knowledge of all aspects of the field has grown even more profound. At the same time, one of the most striking trends in optimization is the constantly increasing emphasis on the interdisciplinary nature of the field. Optimization has been a basic tool in all areas of applied mathematics, engineering, medicine, economics and other sciences.

The series *Springer Optimization and Its Applications* publishes undergraduate and graduate textbooks, monographs and state-of-the-art expository works that focus on algorithms for solving optimization problems and also study applications involving such problems. Some of the topics covered include nonlinear optimization (convex and nonconvex), network flow problems, stochastic optimization, optimal control, discrete optimization, multi-objective programming, description of software packages, approximation techniques and heuristic approaches.

More information about this series at http://www.springer.com/series/7393

Giorgio Fasano • János D. Pintér
Editors

Modeling and Optimization in Space Engineering

State of the Art and New Challenges

Springer

Editors
Giorgio Fasano
Thales Alenia Space
Turin, Italy

János D. Pintér
Department of Industrial and
Systems Engineering
Lehigh University
Bethlehem, PA, USA

ISSN 1931-6828 ISSN 1931-6836 (electronic)
Springer Optimization and Its Applications
ISBN 978-3-030-10500-6 ISBN 978-3-030-10501-3 (eBook)
https://doi.org/10.1007/978-3-030-10501-3

Library of Congress Control Number: 2019935527

Mathematics Subject Classification: 05B40, 37N05, 37N40, 49-06, 65K05, 70M20, 90Bxx, 90-08, 90C11, 90C26, 90C29, 90C30, 90C90

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Two edited volumes dedicated to the challenging and wide-ranging subject of optimization in space engineering have been recently published:

Fasano and Pintér, Eds., *Modeling and Optimization in Space Engineering*, Springer, 2013

Fasano and Pintér, Eds., *Space Engineering: Modeling and Optimization with Case Studies*, Springer, 2016

The strong interest towards the topics covered by these works has motivated the third topical book project that has led to the present volume.

Space engineering tasks frequently require the analysis and solution of advanced and often very hard optimization problems. In the earliest studies, the primary concern was related to the viability of the mission to accomplish. Therefore optimization generally focused on mission analysis aspects, with specific attention to technical feasibility and mission safety. Space engineering projects typically required the analysis and optimization of trajectories and fuel consumption, with paramount consideration given to crew protection.

As time has passed, commercial interests and implied cost-efficiency aspects related to space projects have become increasingly important. A well-known example – in apparent need of complex cost-benefit and risk analysis studies – is the continuing operation of the International Space Station.

For current and forthcoming space engineering projects, today's higher environmental awareness imposes mission constraints that in the past were simply (or almost) neglected. The ambitious goals of future interplanetary explorations – specifically including manned missions – will require advanced analytical approaches to guarantee safety, to maximize the performance of the systems adopted, and to make use of mission resources as efficiently as possible.

Current optimization issues are related to a broad range of challenges including, e.g. low-thrust transfers, interplanetary trajectories, transfers to near-Earth objects, safety analysis of possible collision with space debris, re-entry vehicles, hybrid rocket engines, robust spacecraft design, on-board task scheduling, cargo loading

and packing, payload accommodation with balancing conditions, and satellite constellation image acquisition.

While the necessary depth and quality of the decisions required by space engineering projects has been increasing, we have also witnessed continuing innovation regarding theoretical advances and practical (ready-to-use) decision support tools for such applications. The results of scientific innovation, modelling, and algorithmic developments are supported and enhanced by today's advanced computational modelling and optimization environments. Starting from the earliest space engineering applications, the solution of increasingly hard optimization problems has become necessary. Until fairly recent times, numerical optimization approaches were limited to handling *linear* or *convex nonlinear continuous* models, as well as *integer linear* or *mixed integer-continuous linear* optimization problems. Recent advances in the area of optimization support also the handling of *non-convex* problem formulations: this development enables the solution of more realistic – but often much harder – optimization problems.

The present volume consists of 17 contributed chapters. Written by leading experts, the book offers in-depth discussions of the mathematical modelling and algorithmic aspects of tackling a broad range of space engineering applications.

Specific mission analysis and attitude control subjects include the following: optimal launch date analysis for interplanetary missions, evolutionary neuro-control for the global optimization of continuous-thrust trajectories, machine learning and evolutionary optimization techniques for interplanetary trajectory design, optimal finite-thrust orbital transfers, single-stage-to-orbit space-plane trajectory performance analysis, ascent trajectory optimization and neighbouring optimal guidance of multistage launch vehicles, catalogue generation of parametric time-optimal transfers for all-electric geostationary satellites, real-time optimal control dedicated tools, advanced numerical strategies for sensitivity analysis and reliability assessment of a launcher stage fallout zone, evidence-based robust optimization of pulsed laser orbital debris removal, simulation and attitude determination, and control of small satellites.

System design and configuration aspects are discussed in several chapters. Topics include the general problem of control dispatch optimization in a spacecraft and a real-world application in the framework of the European Space Agency's Next Generation Gravity Mission, packing problems with balancing conditions, and the optimal topological design of a thermal insulator for a monopropellant space thruster. The analysis of observation planning and scheduling aspects in multiple heterogeneous satellite missions also belongs to this group of studies.

This book will be of interest for researchers and practitioners working in the field of space engineering. Since it offers an in-depth exposition of the mathematical modelling and algorithmic and numerical solution aspects of the topics covered, the book will be useful also for aerospace engineering graduate and post-graduate students who wish to expand upon their knowledge, by studying real-world applications and challenges that they will meet in their professional work. The contributed chapters are focused on space engineering practice, rather than on theory. With this aspect in mind, researchers and practitioners in mathematical systems modelling,

operations research, optimization, and optimal control will also benefit from the case studies presented in this work.

The model development and optimization approaches discussed in the book can be extended also towards other application areas that are not related to space engineering. Hence, the book can be a useful reference to assist the development of new applications. Readers will obtain a broad overview of some of the most challenging space engineering operational scenarios of today and tomorrow: this aspect will benefit managers in the aerospace field, as well as in other industrial sectors.

Turin, Italy — Giorgio Fasano
Bethlehem, PA, USA — János D. Pintér

Acknowledgements

First and foremost, we wish to thank all Authors for their high-quality contribution and for their dedicated effort to make the timely completion and publication of this book possible.

We are grateful to our colleagues who assisted us with valuable peer reviews of the contributed chapters. We wish to express our special thanks to Mathieu Balesdent, Lorenzo Casalino, Simone Ceccherini, Kai Wah Chan, Bernd Dachwald, Giancarlo Genta, Marcel Jacobse, Matthias Knauer, Renke Kuhlmann, Erwin Mooij, Jérôme Morio, Mauro Pontani, Kai Schäfer, Mark Senez, and Francesco Topputo.

One of the editors (GF) thanks also Walter Cugno and Roberto Angelini of Thales Alenia Space (Turin, Italy) for their support of the research and development activities related to modelling and optimization in a range of space engineering applications.

We have been glad to work with Razia Amzad, our book project Editor at Springer, Vinodhini Srinivasan, and the entire Springer production team on this project, from its initial discussion to its completion. We look forward to continuing cooperation.

Contents

About the Editors

Giorgio Fasano is a researcher and practitioner at Thales Alenia Space, with more than three decades of experience in the field of optimization and space engineering applications. He holds an MSc degree in Mathematics and is a fellow of the Institute of Mathematics and its Applications (IMA, UK), with the designations of Chartered Mathematician (IMA, UK), as well as Chartered Scientist (Science Council, UK). Since 2015, he has collaborated as a reviewer for *Mathematical Reviews* by appointment of the American Mathematical Society (AMS). His interests include mathematical modelling, operations research, mixed-integer programming, global optimization, and optimal control. He is the author of *Solving Non-standard Packing Problems by Global Optimization and Heuristics* (Springer, 2014) and a coeditor of *Operations Research in Space and Air* (Ciriani et al., Kluwer Academic Publishers, 2003), as well as coeditor with Pintér of two other volumes (see below). He also wrote other publications related to optimization in space.

János D. Pintér is a researcher and practitioner with over four decades of experience in the broad area of systems modelling, analytics, and optimization, with an emphasis on algorithm and software development for nonlinear optimization. He holds MSc, PhD, and DSc degrees in Mathematics. So far, he has authored and edited ten books, including the award-winning monograph *Global Optimization in Action*. He is also author/co-author of (about 200) journal articles, book chapters, research reports, book, and software reviews. Dr. Pintér has been a member and officer of international professional organizations including CORS, EUROPT, HORS, and INFORMS. He has also served for extensive periods of time on the editorial board of international professional journals. Dr. Pintér is the principal developer of nonlinear optimization software products linked to modelling platforms. In 2016, he

joined the Department of Industrial and Systems Engineering at Lehigh University, where he teaches a range of operations research topics.

In addition to the current volume, Fasano and Pintér are coeditors of the following books:
Modeling and Optimization in Space Engineering, Springer, 2013
Optimized Packings with Applications, Springer, 2015
Space Engineering: Modeling and Optimization with Case Studies, Springer, 2016

Control Propellant Minimization for the Next Generation Gravity Mission

Alberto Anselmi, Stefano Cesare, Sabrina Dionisio, Giorgio Fasano, and Luca Massotti

Abstract This chapter addresses the Next Generation Gravity Mission (NGGM), a candidate Earth observation mission of the European Space Agency (ESA), currently undergoing system and technology studies. NGGM is intended to continue the series of ESA missions measuring Earth gravity from space, successfully started with the Gravity field and Ocean Circulation Explorer (GOCE) satellite which flew between 2009 and 2013. Whereas GOCE measured static gravity by a three-axis gradiometer, NGGM will monitor the temporal variations of the gravity field due to mass (primarily water) transport in the Earth system with a concept pioneered by GRACE (Gravity Recovery and Climate Experiment), with improved sensitivity, thanks to laser tracking between satellite pairs. As a monitoring mission, NGGM shall be of a long duration, 11 years according to the current scientific requirements. In addition, the laser interferometer and accelerometer payloads impose demanding requirements such as suppression of the air drag disturbances, precise pointing, and angular rate control. The long lifetime and the control requirements can only be met by using electric thrusters with high specific impulse, hence low mass consumption. Nevertheless, propellant mass minimization remains a dominant task of the mission design. This objective requires proper selection of the thruster operating ranges, as well as an optimized thruster layout and thrust dispatching algorithms. The method applied to solve the thrust dispatching problem is the subject of another chapter in this volume. The present chapter illustrates the flow-down of mission and system requirements into the proposed spacecraft implementation and operation features,

A. Anselmi (✉) · S. Cesare · S. Dionisio · G. Fasano
Thales Alenia Space, Turin, Italy
e-mail: alberto.anselmi@thalesaleniaspace.com; stefano.cesare@thalesaleniaspace.com; sabrina.dionisio@thalesaleniaspace.com; giorgio.fasano@thalesaleniaspace.com

L. Massotti
European Space Agency, ESTEC, Noordwijk, The Netherlands
e-mail: luca.massotti@esa.int

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_1

focusing on the thruster layout optimization problem. The proposed design is shown to meet the mission requirements, thus validating the methodology adopted as well as the results achieved. Further research avenues opened by the current work are outlined in the conclusions.

Abbreviations

AOCS	Attitude and Orbit Control System
CoM	Centers of mass
DFACS	Drag-Free and Attitude Control System
DoF	Degrees of freedom
E2E	End-to-end simulator
EGG	Electrostatic gravity gradiometer
ESA	European Space Agency
GNC	Guidance Navigation and Control
GO	Global optimization
GOCE	Gravity field and Ocean Circulation Explorer
GRACE	Gravity Recovery and Climate Experiment
LAGEOS	Laser Geodynamics Satellite
LL-SST	Low-low satellite-to-satellite tracking
MBW	Measurement bandwidth
MDT	Mean dynamic topography
MILP	Mixed integer linear programming
NGGM	Next Generation Gravity Mission
NLP	Nonlinear programming
POD	Precise orbit determination
RC	Repeat cycle
SD	Spectral density
SGG	Satellite gravity gradiometry
SLR	Satellite laser ranging
SSD	Satellite-to-satellite distance
SST	Satellite-to-satellite tracking

1 Measuring Gravity from Space: Previous Projects and NGGM Overview

The study of the perturbations of the orbits of artificial satellites provided the first source of detailed information about Earth's gravity field [1]. Soon, however, the non-gravitational perturbations became a fundamental limit to the accuracy and resolution of the field that could be obtained. Thus, satellites dedicated to geodesy were conceived.

The Laser Geodynamics Satellite, LAGEOS, launched in 1976 [2], was the first dedicated geodesy satellite, whose orbital path is precisely reconstructed via laser ranging. From that date, satellite laser ranging (SLR) to high-altitude geodesy satellites such as LAGEOS and its successors has been the source of the most accurate data on the very-long-wavelength range coefficients of the Earth's field and their variations.

Sampling the field with a high spatial and temporal resolution requires very low orbit altitude satellites and sophisticated metrological techniques, with matching measurement and control of the non-gravitational effects. Techniques studied since the 1970s include low-low satellite-to-satellite tracking (SST) and satellite gravity gradiometry (SGG) (see, e.g., [3–6]). Since the advent of the GPS constellation, high-low SST has been another resource for gravity field modelling.

GRACE (Gravity Recovery and Climate Experiment), operating from 2002 to 2017, was the first SST mission using microwave ranging between two co-orbiting low-altitude satellites [7]. GRACE produced consistent long-to-medium-wavelength global gravity field models and models of its temporal changes. GOCE (Gravity field and Ocean Circulation Explorer), operating from 2009 to 2013, was the first satellite gradiometry mission, and it provided high-accuracy and high-resolution static gravity field models [8].

The gravity field reveals the density distribution of the Earth's interior. Together with seismic tomography and magnetometry, gravimetry is one of the few means we have to probe the Earth's deep interior. Gravity and geoid anomalies reflect density anomalies in oceanic and continental lithosphere and the mantle, while redistribution or exchange of mass in the Earth's system results in temporal gravity and geoid changes [9]. In recent years, understanding global change has become a dominant focus of Earth research for its impact on society, as well as science. Time-variable gravity results from GRACE have provided unique insight into phenomena such as ice mass trends in Greenland and Antarctica, total water storage in major global river basins, and trends in ocean mass distribution, revealing decadal shifts in ocean circulation. No space mission except one dedicated to gravity can provide measurements of mass flux in the Earth System of such sensitivity and scope. This motivates ESA's initiative toward a Next Generation Gravity Mission (NGGM). The NGGM requires a quantum leap not only in the ranging sensor, such as that provided by laser interferometry in place of microwave sensing, but also a parallel improvement in the performance of the satellite subsystems dedicated to attitude and disturbance control, where the GOCE experience will be brought to bear.

1.1 GOCE and Its Heritage

The scientific objectives of GOCE were the determination of the Earth's steady-state gravity field anomalies with an accuracy of 1 mGal (1×10^{-5} m/s^2), and the determination of the geoid height with an accuracy between 1 and 2 cm, at length scales smaller than 100 km. GOCE achieved its objectives by measuring gravity gradients by an electrostatic gravity gradiometer (EGG), and carrying out precise orbit determination (POD) based on GPS data.

GOCE has produced, and continues to produce, fundamental results and improvements in all its application areas, i.e., oceanography, solid earth, geodesy, seismology, and aeronomy. The GOCE geoid models had a major impact for the estimation of the ocean mean dynamic topography (MDT) and the related geostrophic currents. GOCE data have been linked to seismic or seismological derived models to assess the contribution of mantle dynamics to surface topography. The techniques developed to process and interpret the gradients from GOCE have been used in applications, integrated with airborne gravity surveys, and as background models for geological exploration.

The 1060-kg GOCE spacecraft (Figure 1) was launched on March 17, 2009 from Plesetsk on a Rokot launch vehicle. Initially it flew in a Sun-synchronous orbit (96.7° inclination, ascending node at 18.00 h) at an altitude in the range of 250÷280 km. Altitude control and drag compensation of the slender spacecraft with a small (1.1 m^2) frontal cross section were realized by two ion thrusters (main and redundant) with a thrust range between 1 and 20 mN, operated in closed loop with the payload accelerometers. Three magnetic torquers provided the attitude control.

About 41 kg of xenon and 14 kg of nitrogen were allocated for drag-free control and gradiometer calibration, sufficient for the planned 20-month lifetime. The actual mission evolution was very different from the worst-case predictions (Figure 2). It turned out that Solar Cycle 24 produced the lowest maximum ever measured. This implied a level of atmospheric drag much lower than expected, to the benefit of the mission lifetime. Thanks to the low-density environment and to the conservative pre-launch satellite drag estimation, the entire mission was spent at altitudes lower than the minimum planned before flight, first around 260 km and then reaching 250,

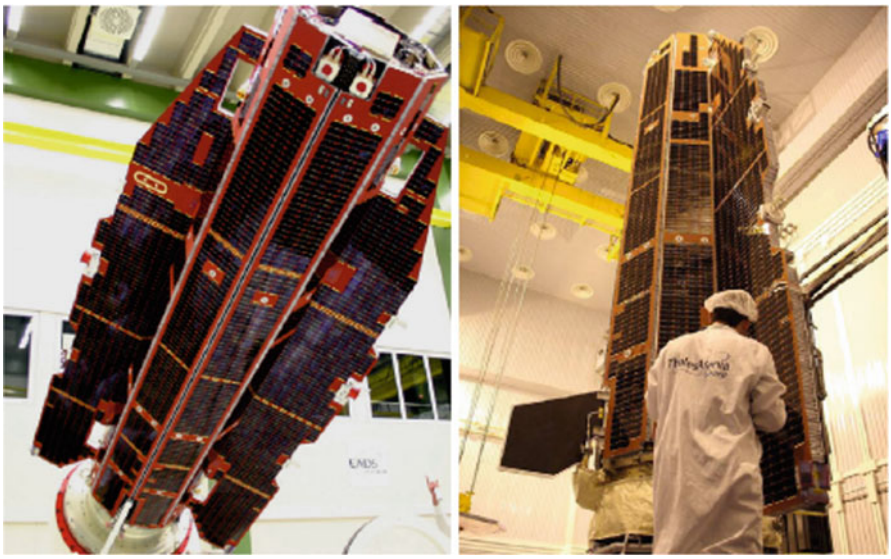


Figure 1 GOCE configuration at the flight acceptance review (March 2008)

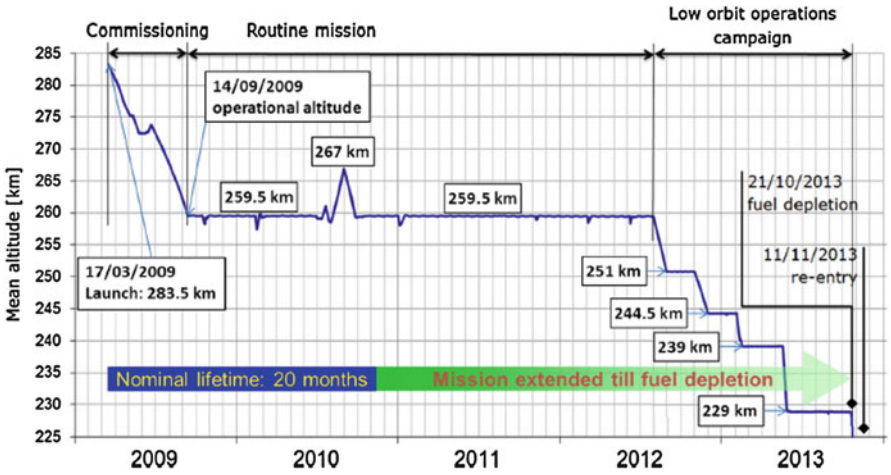


Figure 2 GOCE altitude profile (altitude = |spacecraft position vector| − 6378 km)

245, 240, and even 229 km in the final months, enhancing the scientific return of the mission.

The flawless performance of the Drag-Free and Attitude Control System (DFACS) was essential to accomplish the mission requirements. In its early design concept, the GOCE drag-free control encompassed six (attitude and orbit) degrees of freedom (DoF). This was intended to provide enhanced robustness vs. the uncertainty of both environment and gradiometer response. The corresponding design had two ion thrusters for active compensation of the main component of the drag and eight micro-thrusters for lateral drag and attitude control purposes. When it became clear that the development of the micro-thruster technology would not be compatible with the planned launch date, it was decided to move to a 4-DoF design, using one ion-thruster for in-track drag compensation and magnetic torque rods for the attitude control, supplemented by on/off cold-gas thrusters for gradiometer calibration [10]. Central to the DFACS achievement was the high-fidelity end-to-end (E2E) simulator designed and implemented at Thales Alenia Space since the early project phases to prove mission performance.

1.2 The Next Generation Gravity Mission Project

Already during the development of GOCE, ESA began preparations for the next step. Studies were promoted to establish the scientific priorities, to identify the most appropriate measurement techniques, to start the associated technology developments, and to define the optimal system scenarios of the mission provisionally called NGGM.

While GOCE aimed to provide a high-resolution static map of the Earth's gravity, the objective of NGGM is the long-term monitoring (namely, one full solar cycle, i.e., 11 years) of the time-variable gravity field. The mission aims to achieve gravity solutions with better than 1 mGal accuracy, with spatial resolution comparable to that of GOCE, at shorter time intervals than GRACE (weekly or less). With respect to GOCE, the new mission implies new measurement techniques and instrumentation, a new mission scenario, and different spacecraft design drivers. Despite the differences, however, several achievements of GOCE (e.g., demonstration of long-duration wide-band drag compensation, ultrasensitive accelerometers, and stable non-cryogenic temperature control in low-Earth orbit) stand as the basis upon which NGGM has been defined.

The time-variable gravity field takes contributions from global geophysical processes encompassing atmosphere, ocean, continental hydrology, and ice, coupled by water flow. In addition, gravity field changes are induced by solid-Earth processes like secular glacial isostatic adjustment and sudden earthquakes with co-seismic and post-seismic signals. Figure 3 shows the Earth system domains with their processes and the relevant spatial and temporal resolutions which should be sampled by the NGGM. Each geophysical process can be modelled, and the models combined and converted to gravity field spherical harmonic series, commonly used to describe the Earth's global gravity field [11].

The NGGM objectives require the production of a geoid solution with 1 mm accuracy at a 500-km spatial resolution every 3 days, and at a 150-km spatial resolution every 10 days. Comparison with Figure 3 shows that a large majority of the phenomena can be recovered from space by the mission (very high space resolution is only attainable in conjunction with local measurements).

The demanding requirements of the NGGM cannot be achieved by gradiometry. The low-low SST technique pioneered by GRACE, instead, can provide the necessary resolution in one direction, that of the line connecting the two satellites. By this approach, two satellites flying in loose formation in a low-Earth orbit act as proof masses immersed in the gravity field (Figure 4). The distance variation between the satellites is the geodetic observable (as measured by the laser interferometer) plus the background non-gravitational acceleration—measured by ultrasensitive accelerometers (like those installed on GOCE)—to be subtracted in the post-processing.

Past ESA studies have addressed the available options in terms of formation geometry, number of satellite pairs, and satellite orbit altitude and inclination, from a scientific point of view [12] and from an engineering perspective [8]. The best compromise between performance and implementation complexity and cost was found by considering the simplest mission scenario, consisting of two pairs of satellites, each pair in the so-called in-line formation (two satellites on the same circular orbit at different true anomalies with the direction of the laser beam along the line joining the centers of mass of the two satellites). Each in-line formation samples the gravity field in the along-track direction only. On a polar orbit, this formation is more sensitive to gravity field variations in the North-South than in the East-West direction. Therefore, a second pair of satellites must be launched in

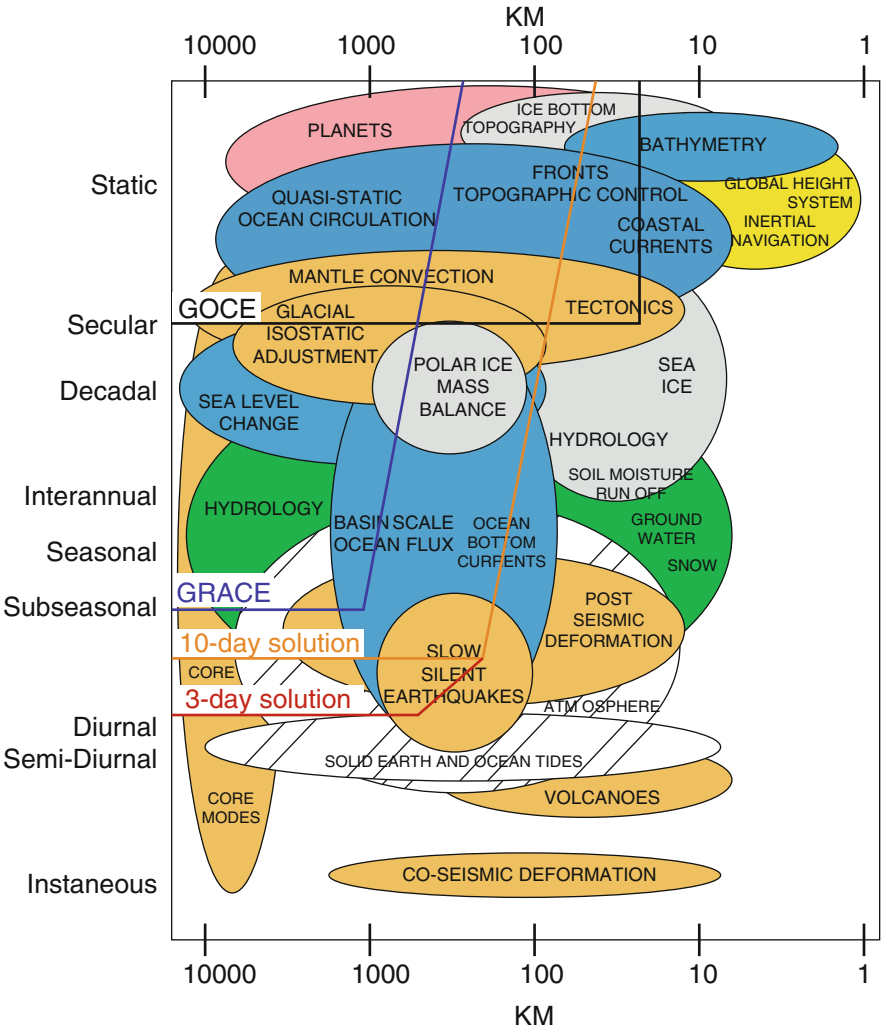


Figure 3 Earth system domains and processes and their spatial and temporal signature [11]

a medium-inclination orbit in order to tackle the aliasing problem with a proper choice (close to optimality) of the repeat cycle (RC). The formation composed by two pairs of satellites, one in near-polar orbit and one in medium-inclination orbit, has become known as the "Bender formation" ([13], Figure 5).

Circular orbits with an altitude of around 350 km and near-polar (one pair) and ~66° (second pair) inclination are suitable for the NGGM, providing all-latitude coverage, short-repeat cycles/sub-cycles, and excellent gravity signal retrieval, and are compatible with a long lifetime (Table 1). For inter-satellite distances in the

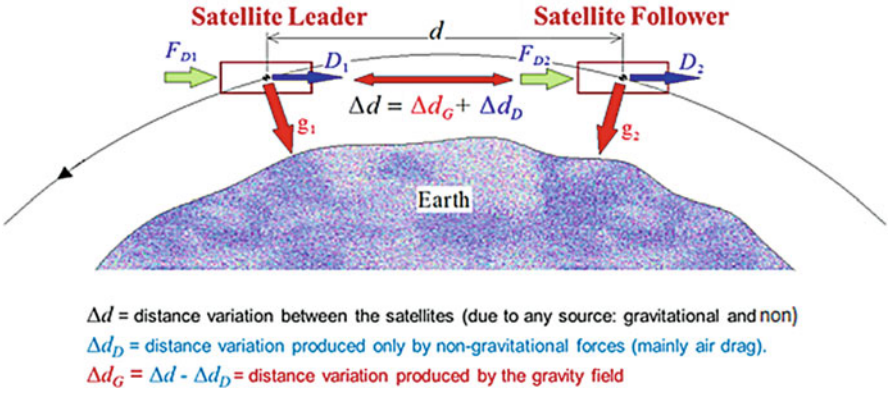


Figure 4 Principle of the LL-SST technique for measuring the Earth's gravity field

Figure 5 NGGM satellite formations

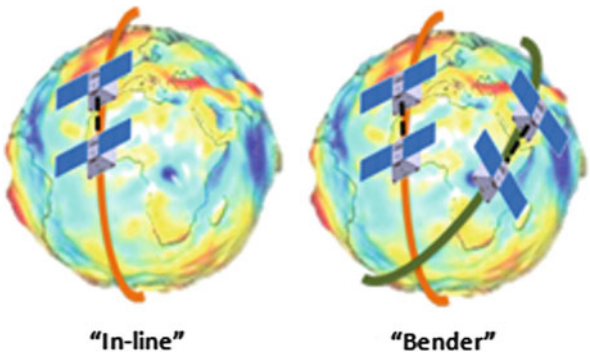


Table 1 NGGM reference orbits

	Near-polar	Inclined
Altitude	367.9 km	345.3 km
Inclination	88°	66°
Number of revolutions in repeat cycle	343	451
Repeat period [nodal day]	22	29

range of 70–100 km, the NGGM performance is relatively constant and lengths >100 km do not provide any benefit in terms of gravity field recovery. Hence, a 100 km inter-satellite distance has been selected.

Due to the non-Sun-synchronous orbits, each NGGM satellite shall tolerate large variation of the solar illumination. Disturbances to the payload shall be minimized and a complex control system shall be implemented, capable of carrying out several tasks in close coordination: orbit and altitude maintenance, formation keeping, attitude stabilization, drag compensation, and laser beam pointing at micro-radian level.

The current on-going "Consolidation of the system concept for the Next Generation Gravity Mission" study, carried out for ESA by Thales Alenia Space, focuses on the consolidation of the mission concept and of the relevant space segment. One of the main drivers is to achieve thrust authority and efficiency for compensating the air drag environment in low-Earth orbit throughout an entire solar cycle, with thrust demand varying by a large factor at orbit frequency and with epoch. This has implications on both technology (thruster performance) and DFACS control (thrust range, minimum control authority, and modulation). The thruster layout is an important factor in meeting the requirements: thus, an optimization methodology was employed aimed at finding the optimal orientation of a set of N thrust vectors (with fixed points of application, dictated by the spacecraft configuration, see Figure 6) minimizing the total propellant consumption.

The remainder of this chapter is structured as follows. Section 2 introduces the NGGM systems engineering constraints, with particular attention paid to the control requirements and their implications on both design and operation. The thruster layout optimization problem is stated together with the related objective function, the propellant consumption, bearing in mind the implicit correlations with design features and overall system operability. Section 3 begins by outlining the dedicated NGGM end-to-end simulator, utilized as the basic instrument for the analysis of the system dynamics. Then, the approach followed for solving the thruster layout problem is discussed, as an application of the methodology conceived for the control dispatch of a general dynamic system and illustrated in detail in another chapter of this book [14]. Illustrative examples from the analysis carried out during the study conclude the section. Future improvements and extensions of the analysis are outlined in Sect. 4, where some conclusions are drawn.

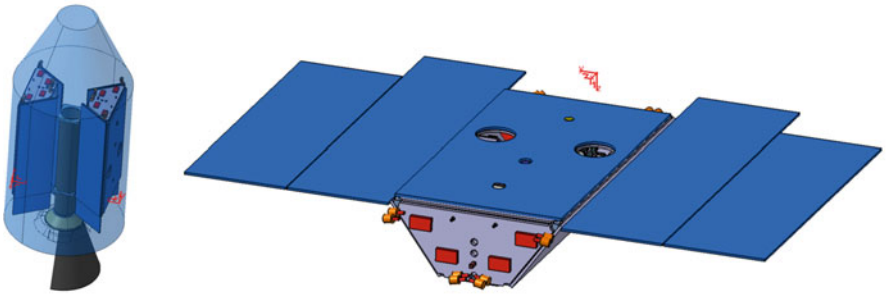


Figure 6 NGGM satellite launch and orbit configuration

2 System and Control Drivers

Numerous constraints from the spacecraft must be considered in the control design (Table 2). The NGGM spacecraft are launched in pairs by Vega-C (see Figure 6).

The diameter and length of the fairing and the central dispenser constrain the size of the spacecraft, and the allowable total mass at launch (about 2250 kg, including two spacecraft and dispenser/adapter, for a near-polar orbit at ∼400 km circular altitude) limits the mass of each satellite to about 900 kg.

The 1-kW electric power demand, driven by the electric propulsion, coupled with the unfavorable Sun illumination during parts of the year, leads to a large (16 m^2) solar array which contributes about half of the drag force to be counteracted by the in-track thrusters, the other half being due to the ∼1 m^2 front cross section. The thrusters must be capable of finely tuned modulation for the science modes and yet provide sufficiently large force for the spacecraft to execute operations, such as formation acquisition, in a reasonably short time. Miniaturized radiofrequency ion thrusters are envisaged at this stage of the study. Thrusters of this type have a limited range of maximum-to-minimum thrust level in which sufficient power and mass efficiency is guaranteed. For this reason, two versions of the thrusters are employed: one set for the thrust in the direction of motion, optimized in the range 1–10 mN; and another set for cross-track control (with a component in the in-track direction too), optimized in the range 50 μN to 1 mN.

Figure 7 shows the laser interferometer and the accelerometers which are the main components of the NGGM payload.

Figure 8 shows the NGGM measurement performance requirements consistent with the stated scientific objectives, expressed as amplitude error spectral density (SD) of the relative distance (dimensionless) and the residual acceleration. For an inter-satellite distance of 100 km, the left-hand side of Figure 8 implies a distance

Table 2 NGGM mission and system drivers

Mission drivers	System drivers
Dual launch	• *Restricted* transportable *mass* and satellite *dimensions*
11-year mission (∼1 solar cycle)	• Equipment *lifetime* • *Propellant demand* to perform orbit and altitude maintenance and drag-free science operations
Low-low SST orbit	• *AOCS* sensors and actuators and *GNC* algorithms *complexity* • *Thruster range, noise, and response time* to guarantee the mission operations (altitude maintenance, orbit control, and drag compensation)
Global coverage	• Non-Sun-synchronous orbits affect *power* and *thermal designs*

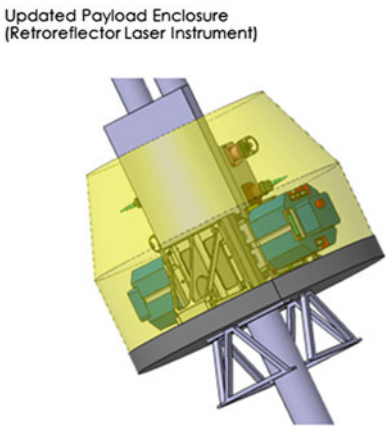


Figure 7 NGGM Payloads: laser instrument and accelerometers

measurement error $<2 \times 10^{-8}$ m/$\sqrt{\text{Hz}}$ (threshold requirement) down to a frequency of 10 mHz. This outstanding performance is achieved by using a laser interferometer fed by a source stabilized in frequency.

The inter-satellite distance variation measurement requirement must be complemented by the requirement on the relative acceleration between the satellite centers of mass (CoM) induced by the non-gravitational forces. This second observable is necessary in order to estimate and separate the non-gravitational effects produced by atmospheric drag from the first observable, thus obtaining the distance variation between the satellite CoMs produced solely by the gravity field. A measurement error SD $< 1 \times 10^{-11}$ m/s^2/$\sqrt{\text{Hz}}$ between 1 and 100 mHz is specified for relative acceleration (threshold requirement). On top of these stringent attitude pointing and drag-free requirements, the NGGM control system is in charge of the orbit and formation maintenance. Table 3 presents the set of NGGM control requirements.

As the mission spans a complete solar cycle of 11 years, the design must make provisions for all foreseeable air density conditions at the lower design altitude of 340 km, where the sensitivity to the gravity signal is still high. Depending on the phase in the cycle, the mean density may vary by a factor of 10. As an example, GOCE flew for 4 years in the altitude range between 260 and 240 km, using up to 40 kg of Xenon propellant to compensate the drag solely in the in-flight direction, all the time in low solar flux conditions (13-month smoothed solar flux index $F_{10.7} < 140$; for comparison, the 95 percentile index predicted for the next solar maximum in 2024 is 240). Even though the altitude is higher, the NGGM lifetime requirement and its 6-DoF control a priori imply much higher propellant consumption. It is therefore necessary to design the system taking into account propellant minimization from the beginning. The optimization process shall encompass both the thruster layout, which cannot be changed after launch, and the thrust dispatch, which depends on the day-to-day evolution of the environment.

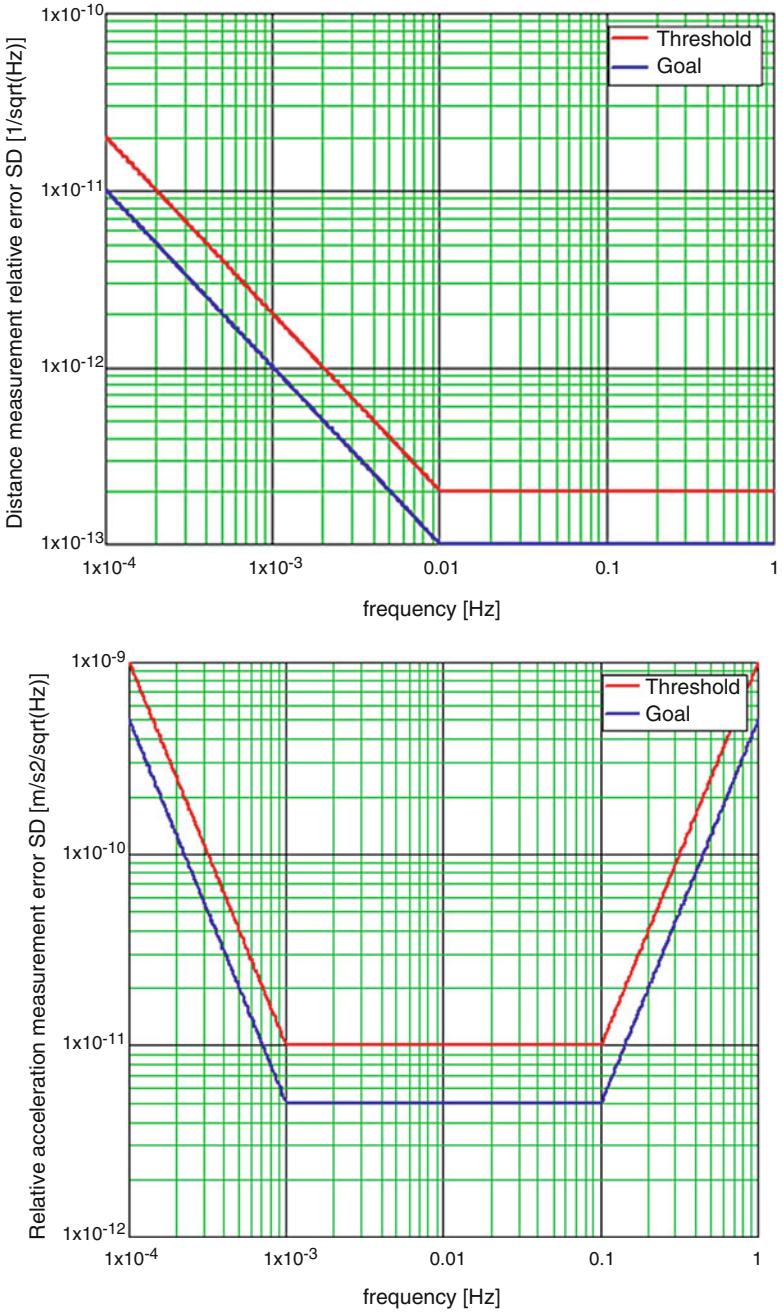


Figure 8 NGGM measurement performance requirements (amplitude spectral density, SD). Left: inter-satellite distance variation. Right: acceleration

Table 3 NGGM control performance requirements

NGGM control requirements		
Control function	Requirement	Remarks
Orbit control	• H = Href ±100 m	Href stands for reference mean orbit altitude. Current definition is in Table 1 Href and ΔH depend on the orbit repeat cycle (RC) of interest
Formation control	• SSD = 100 km + 0%/−10%	SSD stands for satellite-to-satellite distance. Its value is defined by the metrology working range and by scientific needs
Drag-free control (linear/angular)	• Lin_acc $\leq 10^{-6}$ m/s^2 • Lin_acc_SD $\leq 5{\cdot}10^{-9}$ m/s^2/$\sqrt{}$Hz in MBW • Ang_acc $\leq 10^{-6}$ rad/s^2 • Ang_acc_SD $\leq 10^{-8}$ rad/s^2/$\sqrt{}$Hz in MBW	Lin_acc/Ang_acc and Lin_acc_SD/Ang_acc_SD stand for the residual linear/angular non-gravitational accelerations in time and relevant spectral density. MBW stands for measurement bandwidth and its range is between 10^{-3} and 10^{-1} Hz
Attitude pointing control	• SSL $\leq 2{\cdot}10^{-5}$ rad • SSL_SD $\leq 10^{-5}$ rad/$\sqrt{}$Hz, for $10^{-3} \leq$ f < 10^{-2} Hz $\leq 2{\cdot}10^{-6}$ rad/$\sqrt{}$Hz, for $10^{-2} \leq$ f $\leq 10^{-1}$ Hz	SSL/SSL_SD stands for satellite-to-satellite line pointing and relevant pointing stability in terms of spectral density

3 Thruster Layout Optimization in a Real-World Scenario

Atmospheric drag is the dominant source of unpredictability for low-Earth orbiting spacecraft, due to the uncertainty about both the relevant density and the properties of the satellite surfaces. In this very challenging framework, an essential instrument is represented by the in-house developed end-to-end (E2E) simulator [8], currently in use and continuously upgraded [15]. It consists of an enhanced version of the system-and-mission-consolidation code, successfully adopted for GOCE, and represents a remarkable by-product of this project. The experience derived has indeed confirmed that the E2E models can be relied on to adequately predict the actual perturbation acting on the satellite when in flight. Figure 9 reports the GOCE drag force along the flight direction, as calculated by the E2E simulator (blue line) and measured on board (July 2011, red line).

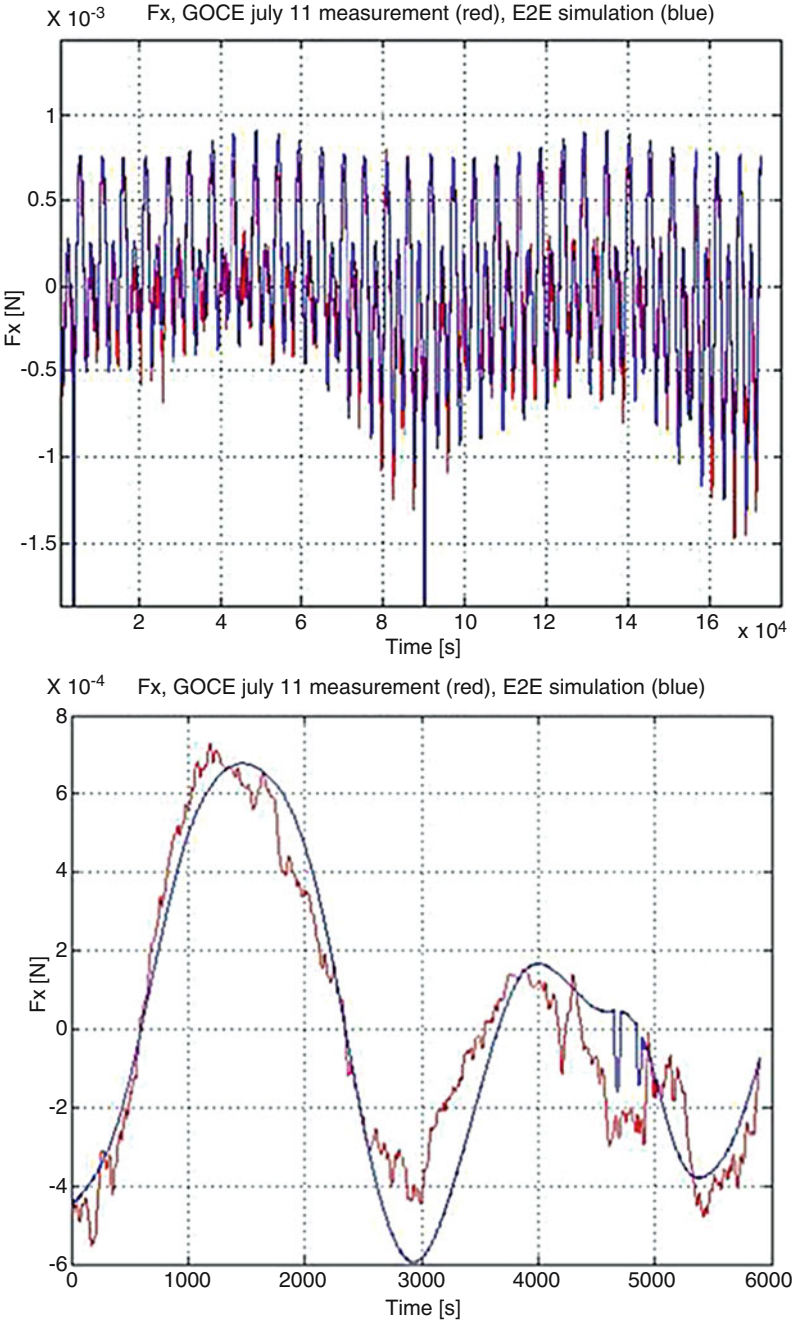


Figure 9 GOCE drag force—Simulator (blue line) and actual measurement in flight (red line)—July 2011

The last released E2E simulator is up to contemplating either a single-satellite-configuration, as in the case of GOCE, or an in-line formation of coupled satellites. It consists of two specific modules: the satellite module takes account of all sensors and actuators, the guidance, navigation, and control (GNC) modes, as well as the spacecraft physical features. The environmental characteristics, the whole system interacts with while operating its measurement process, including all error sources, are additionally considered. The post-processing module, instead, performs the outline data post-processing of the scientific output, consisting in the measurements data yielded by the satellite module.

During the last preliminary study carried out for NGGM, the E2E simulator was employed in support of the system design and as a validation framework for the obtained thruster layout solutions. The overall forces and torques required by the control, during the science mode, have been evaluated on the basis of the 11-year prediction of the solar activity modelled on Cycle 23. To the purpose, three atmospheric condition scenarios, representing the so-called minimum, mean, and maximum solar activities, have been selected, as the analysis framework. Figure 10 reports the drag perturbation force along the flight direction, as per a prediction example modelled on Cycle 23, for an orbit with a mean altitude of about 340 km.

For each scenario, a statistically representative sample has been extracted and the relevant control force/torque evaluated. This force/torque profile is the input to the thruster layout optimization exercise, as described in the remainder of this section, and a subsequent validation process is carried out by the E2E simulator.

The validation of the analytical optimal solutions to the thruster layout problem, in terms of propellant consumption and actuator life time, is obtained through

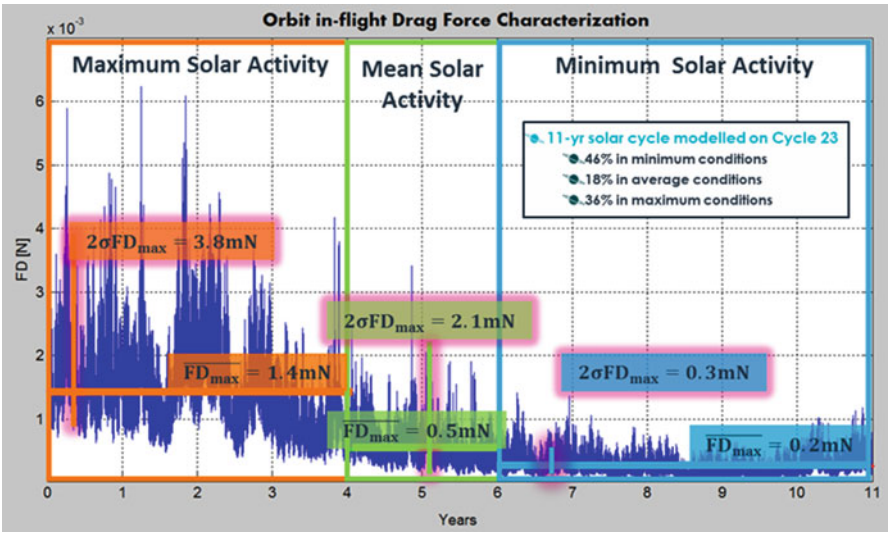


Figure 10 Example of drag perturbation force along the flight direction prediction (Cycle 23, ~340 km mean altitude)

long-run executions of the E2E simulator. For this aim, the possibility of defining any given thruster layout, parametrically, as an input of the E2E simulator, has been introduced. An ad hoc on-board dispatch algorithm, compliant with the thruster layout optimization criterion, has additionally been introduced as a further basic feature of the E2E simulator.

The obtained enhanced version has widely confirmed the potential of the overall optimization methodology outlined hereinafter.

As is gathered from what has been illustrated so far in this chapter, the thruster layout optimization issue presents even a skilled designer with a very challenging task. As anticipated, the specific exercise relevant to the NGGM study has been coped with, by adopting a novel mathematical approach, conceived, as a whole, for the control dispatch in a general dynamic system. This overall methodology is described at a detailed level in a dedicated chapter of this book [14]. Its application to the NGGM framework is outlined hereinafter, focusing on the relevant conceptual facets of the problem and carrying over the specific terminology adopted in the study. The analysis reported in the following refers to a single satellite of the NGGM formation, being conceptually identical for both.

As anticipated above for the NGGM study, three operational scenarios representative of the low/medium/high atmospheric drag conditions, respectively, have been considered, in addition to a specific control law. The given control law (for each operational scenario) is expressed in terms of overall force and torque demand, i.e., $\boldsymbol{F}$ and $\boldsymbol{T}$, as expressed with respect to a given system reference frame (X,Y,Z).

Due to the discrete nature of the control action, the whole time span considered in the analysis is discretized on the basis of a (predefined) set of time steps (instants), of duration Δ each. The given control law is expressed by the following equations:

$$
\begin{aligned}
&\forall i \in I \\
&v_{1x}u_{1i} + \cdots + v_{Nx}u_{Ni} = F_{ix}, \\
&v_{1y}u_{1i} + \cdots + v_{Ny}u_{Ni} = F_{iy}, \\
&v_{1z}u_{1i} + \cdots + v_{Nz}u_{Ni} = F_{iz}, \\
&\left(\boldsymbol{p}_1 \times \boldsymbol{v}_1\right)_x u_{1i} + \cdots + \left(\boldsymbol{p}_N \times \boldsymbol{v}_N\right)_x u_{Ni} = T_{ix}, \\
&\left(\boldsymbol{p}_1 \times \boldsymbol{v}_1\right)_y u_{1i} + \cdots + \left(\boldsymbol{p}_N \times \boldsymbol{v}_N\right)_y u_{Ni} = T_{iy}, \\
&\left(\boldsymbol{p}_1 \times \boldsymbol{v}_1\right)_z u_{1i} + \cdots + \left(\boldsymbol{p}_N \times \boldsymbol{v}_N\right)_z u_{Ni} = T_{iz}.
\end{aligned}
\tag{1}
$$

Here, the following notations have been adopted:

- I is the set of time steps considered;
- N is the number of thrusters utilized;
- $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N$ are the unit vectors determining the orientation of each thruster;
- $v_{1x}, v_{1y}, v_{1z}, \ldots, v_{Nx}, v_{Ny}, v_{Nz}$ are the direction cosines associated with $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N$;
- $u_{1i}, \ldots, u_{Ni}$ are the thrusts (scalars) associated with each thruster, respectively, at each instant i considered;

- $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$ are the position vectors of the thrusters (i.e., the force application points, corresponding to each thruster);
- $(\boldsymbol{p}_1 \times \boldsymbol{v}_1)_x, (\boldsymbol{p}_1 \times \boldsymbol{v}_1)_y, (\boldsymbol{p}_1 \times \boldsymbol{v}_1)_z, \ldots, (\boldsymbol{p}_N \times \boldsymbol{v}_N)_x, (\boldsymbol{p}_N \times \boldsymbol{v}_N)_y, (\boldsymbol{p}_N \times \boldsymbol{v}_N)_z$ are the components of the vector products $\boldsymbol{p}_1 \times \boldsymbol{v}_1, \ldots, \boldsymbol{p}_N \times \boldsymbol{v}_N$, respectively;
- F_{xi}, F_{yi}, F_{zi} and T_{xi}, T_{yi}, T_{zi} are the components of the overall force $\boldsymbol{F}_i$ and torque $\boldsymbol{T}_i$, at each instant i considered.

The following normalization conditions have to be set, for the direction cosines corresponding to each thruster r ($r = 1, \ldots, N$):

$$\begin{aligned} &\forall r \quad v_{xr}^2 + v_{yr}^2 + v_{zr}^2 = 1, \\ &\underline{V}_{\boldsymbol{r}} \leq \left(v_{\mathrm{rx}}, v_{\mathrm{ry}}, v_{\mathrm{rz}}\right)^T \leq \overline{V}_{\boldsymbol{r}}, \end{aligned} \tag{2}$$

where $\underline{V}_{\boldsymbol{r}}$ and $\overline{V}_{\boldsymbol{r}}$ are lower and upper bounds, respectively (expressed as column vectors, see *Layout* model description) on the admissible orientation for each thruster. In the present study, all external directions, with respect to the surface on which the thruster is positioned, are contemplated (although more restrictive conditions will be imposed in the future).

Since each thruster has given limitations, consisting of the minimum and maximum force that it can exert (depending on its technical characteristics), the lower/upper bounds below are primarily set (for all the instants), as basic conditions:

$$\forall i \in I \quad \underline{U}_r \leq u_{ri} \leq \overline{U}_r, \tag{3}$$

where $\underline{U}_r$ and $\overline{U}_r$ are given (positive) constants.

Further conditions such as the following can, nonetheless, be profitably added, in order to take into account of the technical limitation of the thrusters, in terms of thrust rate capability. To this purpose it is stated that the maximum allowable difference of the forces applied by the same thruster in two subsequent instants cannot exceed a given limit (this can be interpreted as a Lipschitzian condition, in a discretized form):

$$\forall i \in \{0, 1, \ldots, |I| - 2\} \quad \mid u_{r(i+1)} - u_{ri} \mid \leq L_r \tag{4}$$

where L_r is a given (positive) constant.

Similarly, the following additional constraints are introduced, with the scope of limiting the utilization of each thruster, along the whole time span and thus preventing the possible overworking of some of these (presumably, among those with a lower cost in terms of energy):

$$\forall r \quad \sum_{i \in I} u_{ri} \leq \frac{J_r}{\Delta}, \tag{5}$$

where J_r represents, for each thruster r, a (technological) upper bound on the total impulse admissible during the whole time span.

It should be observed that, according to a classical approach, equation system (1), per se, could be solved, for each instant i, by means of the Moore–Penrose pseudo-inverse matrix (see, e.g., [16]). As per a known property, this way the Euclidean norm $\|u\|_2$ is minimized. This criterion, nevertheless, differs on the whole from the overall propellant minimization. The aforementioned inversion, moreover, is not in general able to contemplate any of conditions (3), (4), or (5). The actual optimization problem in question, indeed, features the following objective function that corresponds to the overall propellant consumption minimization:

$$\min \sum_{r,i} f_r\left(u_{ri}\right), \tag{6}$$

where $f_r(u_{ri})$ is, for each instant i, the demand associated with each thruster r (the objective function thus defined is not necessarily linear). The minimization target is furthermore subject to conditions (1)–(5).

All that being stated, it has moreover to be observed that for quite predictable "physical" reasons, the resulting problem could be infeasible, i.e., the domain delimited by conditions (1)–(5) could be empty. This would mean that there is no thruster accommodation up to satisfying the control request, at any instant, for the whole time span. The existence of a single thruster layout, indeed, able to dispatch the requested control (in compliance with the given thrust bounds and additional operational conditions) for the whole set of instants considered is not granted a priori. To overcome this shortcoming, a possible relaxation of the problem could be taken into account, by adding error variables in (1) and readjusting the objective function adequately, in order to minimize (additionally) the overall error. Equations (1) may therefore be substituted with the following:

$$\begin{aligned}
&\forall i \in I \\
&v_{1x}u_{1i} + \cdots + v_{Nx}u_{Ni} = F_{ix} + \varepsilon_{Fix}, \\
&v_{1y}u_{1i} + \cdots + v_{Ny}u_{Ni} = F_{iy} + \varepsilon_{Fiy}, \\
&v_{1z}u_{1i} + \cdots + v_{Nz}u_{Ni} = F_{iz} + \varepsilon_{Fiz}, \\
&\left(\boldsymbol{p}_1 \times \boldsymbol{v}_1\right)_x u_{1i} + \cdots + \left(\boldsymbol{p}_N \times \boldsymbol{v}_N\right)_x u_{Ni} = T_{ix} + \varepsilon_{Tix}, \\
&\left(\boldsymbol{p}_1 \times \boldsymbol{v}_1\right)_y u_{1i} + \cdots + \left(\boldsymbol{p}_N \times \boldsymbol{v}_N\right)_y u_{Ni} = T_{iy} + \varepsilon_{Tiy}, \\
&\left(\boldsymbol{p}_1 \times \boldsymbol{v}_1\right)_z u_{1i} + \cdots + \left(\boldsymbol{p}_N \times v_N\right)_z u_{Ni} = T_{iz} + \varepsilon_{Tiz}, \\
&\forall i \in I \quad \varepsilon_{Fi} \in [-E_F, E_F] \\
&\forall i \in I \quad \varepsilon_{Ti} \in [-E_T, E_T],
\end{aligned} \tag{7}$$

where $\varepsilon_{Fi} = (\varepsilon_{Fxi}, \varepsilon_{Fyi}, \varepsilon_{Fzi})^T$, $\varepsilon_{Ti} = (\varepsilon_{Txi}, \varepsilon_{Tyi}, \varepsilon_{Tzi})^T$ and $E_F > 0$, $E_T > 0$ are the admitted tolerances (expressed as column vectors). This aspect, not addressed here, could be the subject of a future extension.

In the NGGM study carried out so far, the thruster positions (i.e., $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$) have been assumed to be constant (this restriction could, indeed, be dropped

in a more in-depth analysis, although stringent configuration constraints would apply anyway). Equations (1), involving both the thrust direction cosines and the corresponding thrusts as variables, for each instant of the whole period to analyze, are associated with the overall demand (to be dispatched by means of the thrusters available), in terms of forces and torques. Normalization Eq. (2) state the inter-dependence among the direction cosines associated with each thruster. The lower and upper bounds (3) delimit the force that can be exerted by each thruster (instant by instant). Inequalities (4) state, additionally, a restriction on the force trend relevant to each thruster, controlling any two subsequent instants. Constraints (5) represent, instead, global conditions relevant to the overall thruster exploitation.

In the current NGGM study, the objective function (6) has been assumed linear (although future work could drop this restriction). The following expression has namely been considered:

$$\min \sum_{r,i \in I} K_r u_{ri}, \tag{8}$$

where the constants K_r represent the propellant consumption associated with each thruster (per force unit and supposed to be time-independent).

It is understood that, considering the wide range of envisaged operational conditions, the optimization problem under discussion is of a global type. An overall global optimization (GO; see, e.g., [17]) approach is therefore unavoidable. The resulting exercise, due to the presence of conditions (1) and (2), is of a non-convex quadratically constrained type and, as such, belongs to the NP-hard class of problems, considered, in the computational complexity theory context (see, e.g., [18]), as extremely demanding. Indeed, under the conjecture (strongly supported by most of scholars) that $P \neq NP$ (where P and NP stand, respectively, for the "polynomial time" and "nondeterministic polynomial time" classes), it has been proven that NP-hard problems cannot be solved in polynomial time. Finding even satisfactory (albeit sub-optimal) solutions to NP-hard problems can be an extremely challenging task. The scale of the instance (e.g., the number of non-convex constraints or of binary variables, if any) is in general a heavily influential factor concerning the actual tractability of the problem.

The case study discussed here is very large. For instance, three different atmosphere density scenarios and approximately two whole representative orbits for each could be taken into account. Considering, for all the selected orbits, a constant time discretization Δ of 2 s, the instants associated with each scenario (i.e., two orbits) would be ~5000, amounting to a total of ~15,000. This would give rise to ~90,000 equations of type (1).

The difficulty of the overall mathematical model under consideration is essentially generated by constraints (1) and (2), since these are non-convex and an additional grade of complexity could well arise, in the case of a nonlinear objective function (in particular a non-convex one). Their overall number, in addition to the current state-of-the-art algorithms available in literature, clearly suggests that the solution of the problem tout court is not the best path.

The model, nonetheless, shows quite a peculiar structure, susceptible to an advantageous approach. Equations (1) (once the position vectors $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$ of the thrusters are assumed to be constant, as in the present case) are bilinear (and involve a large quantity of variables, i.e., the thrusts associated with each thruster per instant considered), while (2) are (few) Euclidean-norm-quadratic equations (involving a very limited number of variables, i.e., the direction cosines).

This has led to the conception of a new ad hoc methodology (see [14]), as an alternative to the adoption of a general algorithm for (non-convex) quadratically constrained problems.

The problem is therefore advantageously partitioned into two sub-problems, entailing the implementation of two dedicated (classes of) models. The first denoted (according to the NGGM terminology) as *Layout* model, focuses (mainly) on thruster layout or, more precisely, their orientation (since in the current version the position of the thruster is assumed to be established a priori). To this purpose, quite a limited sub-set of instants, supposed to be representative of the whole mission, is taken into account. The second sub-problem is devoted to the total propellant minimization, referring to the whole operational scenario (i.e., contemplating low, medium, and high atmosphere density conditions), corresponding to the entire mission. This (according to the NGGM terminology) is referred to as the *Master* model: as a matter of fact, it mainly intends to verify the feasibility of the solution found by the *Layout* model, covering the entire set of instants relevant to the whole analysis, instead of a sub-set of it only.

Both the Layout and *Master* models include Eq. (1), bounds (3), and objective function (8). Equations (2) are considered exclusively in the *Layout* model, while inequalities (4) and (5) in the *Master* model only. In the first, all the variables of the general problem (i.e., v and u) are treated as such. In the second, instead, the variables relative to the thruster orientations (i.e., v) are fixed on the basis of the results obtained by the *Layout* model. As a consequence, while the *Layout* model keeps its characteristic of being quadratically constrained, the *Master* model becomes linear. The *Layout* model (supposedly very tough to tackle) is intended to solve small-scale instances (corresponding to appropriate sub-sets of instants). The *Master* model instead (plausibly easy to manage) is dedicated to the solution of the very large-scale instance contemplating the full set of instants.

Concerning the *Layout* model, two (GO) approaches have been looked into. The first (rigorous approach) consists in utilizing a global optimizer and solving the *Layout* model directly. The second (approximated approach) is based on the discretization of the variables (v) corresponding to the thruster orientations (this is quite advantageous, since there are only three orientation variables for each thruster). This way, the quadratic Eq. (1) become linear (for each discretized value of the variables v) and the normalization conditions (2) are dropped. The aforementioned discretization, however, requires the introduction of a number of 0–1 variables and this, inevitably, transforms the original nonlinear model into a mixed integer linear programming (MILP; see, e.g., [19]) one that also, as such, belongs to the NP-hard class. (It should be noticed here that often the choice of an approximate approach, instead of a rigorous one, can lead to more effective solutions in practice.)

By fixing, in the *Master* model, the variables (v) corresponding to the thruster orientations, all the constraints involved, i.e., (1), (3)–(5) result in being linear. As pointed out previously, the overall optimization problem (covering all the given operational scenarios) could, per se, be infeasible, giving rise to a non-eliminable amount of error. The infeasibility, nonetheless, could also be caused by the fact that the orientations of the thrusters are pre-selected (by the *Layout* model) taking into account only quite a restricted number of instants. In this case, a reduction of the overall error is plausible and a further search should be carried out. To this purpose, a post-optimization phase based on sequential linear programming can be realized (see, e.g., [20]). The *Layout* and *Master* models are reported, explicitly, hereinafter.

3.1 Layout Model

The general formulation of the *Layout* model can be expressed as follows:

$$
\begin{aligned}
&\forall i \in \underline{I} \\
&v_{1x}u_{1i} + \cdots + v_{Nx}u_{Ni} = F_{ix}, \\
&v_{1y}u_{1i} + \cdots + v_{Ny}u_{Ni} = F_{iy}, \\
&v_{1z}u_{1i} + \cdots + v_{Nz}u_{Ni} = F_{iz}, \\
&\left(\boldsymbol{p}_1 \times \boldsymbol{v}_1\right)_x u_{1i} + \cdots + \left(\boldsymbol{p}_N \times \boldsymbol{v}_N\right)_x u_{Ni} = T_{ix}, \\
&\left(\boldsymbol{p}_1 \times \boldsymbol{v}_1\right)_y u_{1i} + \cdots + \left(\boldsymbol{p}_N \times \boldsymbol{v}_N\right)_y u_{Ni} = T_{iy}, \\
&\left(\boldsymbol{p}_1 \times \boldsymbol{v}_1\right)_z u_{1i} + \cdots + \left(\boldsymbol{p}_N \times \boldsymbol{v}_N\right)_z u_{Ni} = T_{iz}, \\
&\forall r \qquad v_{xr}^2 + v_{yr}^2 + v_{zr}^2 = 1, \\
&\forall i \in \underline{I} \quad \underline{U}_r \leq u_{ri} \leq \overline{U}_r, \\
&\min \sum_{r,i \in \underline{I}} K_r u_{ri}.
\end{aligned}
\tag{9}
$$

In this model (encompassing both the continuous and discretized versions) $\underline{I} \subset I$ and inequalities (4), as well as (5) are omitted (the latter could be included in a more refined version). It contemplates, for each thruster, the set (or a sub-set, in the discretized version) of all admissible orientations that are associated with a unit semi-sphere. This is described by all unit vectors centered in the thruster position (actually the thrust application point) and directed externally, with respect to the corresponding satellite surface. A local reference frame is hence defined for each semi-sphere. Each axis (x,y,z) of this is parallel to the corresponding (X,Y,Z) of the system reference frame. Each unit vector (describing the corresponding semi-sphere) can be identified through two spherical coordinates α and β (angles in radians). The angle α ($0 \leq \alpha \leq 2\pi$) represents, for each semi-sphere, the polar coordinate, with $\alpha = 0$ corresponding to the y axis. The angle β ($0 \leq \beta \leq \frac{\pi}{2}\pi$) represents, for each semi-sphere, the azimuthal coordinate, with $\beta = \frac{\pi}{2}$ corresponding to the x axis.

The non-continuous formulation of the *Layout* model is based on the discretization of the orientations corresponding to each thruster. To this purpose, both angles α and β can be partitioned into two finite sub-sets, dividing the corresponding intervals $\alpha \in [0, 2\pi]$ and $\beta \in \left[0, \frac{\pi}{2}\right]$ by a pre-selected number. The amplitudes of the thus obtained sub-angles will be denoted in the following by Δ-angle. Different discretization approaches may be considered.

3.2 Master Model

The resulting linearly constrained model of the *Master* model can be expressed as

$$
\begin{aligned}
&\forall i \in I \\
&v_{1x}u_{1i} + \cdots + v_{Nx}u_{Ni} = F_{ix}, \\
&v_{1y}u_{1i} + \cdots + v_{Ny}u_{Ni} = F_{iy}, \\
&v_{1z}u_{1i} + \cdots + v_{Nz}u_{Ni} = F_{iz}, \\
&\left(\boldsymbol{p}_1 \times \boldsymbol{v}_1\right)_x u_{1i} + \cdots + \left(\boldsymbol{p}_N \times \boldsymbol{v}_N\right)_x u_{Ni} = T_{ix}, \\
&\left(\boldsymbol{p}_1 \times \boldsymbol{v}_1\right)_y u_{1i} + \cdots + \left(\boldsymbol{p}_N \times \boldsymbol{v}_N\right)_y u_{Ni} = T_{iy}, \\
&\left(\boldsymbol{p}_1 \times \boldsymbol{v}_1\right)_z u_{1i} + \cdots + \left(\boldsymbol{p}_N \times \boldsymbol{v}_N\right)_z u_{Ni} = T_{iz}, \\
&\forall i \in I \qquad \underline{U}_r \leq u_{ri} \leq \overline{U}_r, \\
&\forall i \in \{0, 1, \ldots |I| - 2\} \quad \mid u_{r(i+1)} - u_{ri} \mid \leq L_r, \\
&\forall r \qquad \sum_{i \in I} u_{\mathbf{ri}} \leq \tfrac{J_r}{\Delta}, \\
&\min \sum_{r,i \in I} K_r u_{ri},
\end{aligned}
\tag{10}
$$

In this model all variables *v* are fixed (on the basis of the solution obtained from the *Layout* model) and Eq. (2) are dropped.

As far as the *Layout* model, in its continuous version, is concerned, the instance below illustrates how its dimension is determined (the selection of the instant sub-set is carried out as specified later on in this section). Considering, for example, a set of 9 thrusters and 18 instants, the resulting instance contains:

- 9×18 variables representing the thrust, for each thruster, at each instant;
- 9×3 variables, corresponding to the direction cosines;
- $9 \times 18 \times 2$ lower and upper bounds on the thrust variables;
- $6 \times 18 + 9$ (bilinear and quadratic) equations (see [14]).

The following general rules account for the abovementioned instance:

- number of thrust variables = (number of thrusters) × (number of instants);
- number of orientation variables = (number of thrusters) × (3 direction cosines per thruster);
- number of lower bounds (for the thrust variables) = number of (thrust) variables;
- number of upper bounds (for the thrust variables) = number of (thrust) variables;

- number of (bilinear dispatch) equations (corresponding to (1)) = (6 matrix rows) x (number of instants);
- number of (quadratic normalization) equations (corresponding to (2)) = number of thrusters.

In the present study, an experimental analysis concerning the continuous *Layout* model has been performed by utilizing a general global NLP (nonlinear programming; see, e.g., [21]) optimizer. The tests concerned instances involving ∼100 variables and ∼70 constraints (satisfactory solutions were usually found in less than 1 h). Although even larger tests have been successfully executed, this order of magnitude represents the current standard (to obtain satisfactory solutions, in reasonable computational times). An in-depth research, aimed at tailoring the solver performances on the specific problem (including, if necessary, the adoption of a dedicated optimizer for non-convex quadratically constrained programming), is expected to extend, even significantly, this current dimensional limit.

Since the continuous *Layout* model is actually obtained by the whole original problem, simply by restricting the set of instants, the above considerations provide an indirect and empirical confirmation that to tackle it tout court would be totally unrealistic.

Regarding the discretized version of the *Layout* model, the same instance reported above, for Δ-angles corresponding to $\frac{\pi}{24}$ (rad), gives rise to:

- $(24 + 6) \times 9$ binary variables;
- 6×18 linear equations;
- $9 \times 24 \times 6 \times 18$ ∼46,000 (discretization) constraints in substitution of the $9 \times 18 \times 2$ lower and upper bounds on the thrust variables.

The abovementioned instance derives from the following statement:

- The (polar) angle α ($0 \leq \alpha \leq 2\pi$) is partitioned into a number of equal sub-angles (24 in the aforementioned example);
- The (azimuthal) angle β ($0 \leq \beta \leq \frac{\pi}{2}\pi$) is partitioned into a number of equal sub-angles (6 in the aforementioned example);
- For each thruster r, each sub-angle h of α and each sub-angle k of β, a binary variable δ_{rhk} is introduced, with the meaning:

$$\begin{aligned} &\delta_{rhk} = 1 \quad \text{if thruster } r \text{ takes the (discretized) orientation } (h, k); \\ &\delta_{rhk} = 1 \quad \text{otherwise.} \end{aligned}$$

The total number of binary variables results in being:

$$\begin{aligned} \text{binary variable number} = \big[&(\text{number of sub} - \text{angles of } \alpha) \\ &+ (\text{number of sub} - \text{angles of } \beta)\big] \times (\text{number of thrusters}) \end{aligned}$$

The number of the thus linearized (dispatch) equations (corresponding to (1)) becomes:

$$\text{number of linear equations} = (6 \text{ matrix rows}) \times (\text{number of instants}) .$$

For each thruster, at each instant, the thrust lower bound is substituted with:

$$(\text{number of sub-angles of } \alpha) \times (\text{number of sub-angles of } \beta)$$

discretization conditions and similarly concerning the corresponding upper bounds.

The total number of the discretization constraints, replacing both the lower and upper bounds (3), is therefore the following:

$$\begin{aligned}\text{number of bound constraints} = (\text{number of thruster}) \times (\text{number of sub-angles of } \alpha) \\ \times (\text{number of sub-angles of } \beta) \times (6 \text{ matrix rows corresponding to (1)}) \\ \times (\text{number of instants}) \times (2 \text{ bounds associated with each thruster}) .\end{aligned}$$

The tests considered for the discretized *Layout* model involved up to ∼40,000 constraints, ∼40,000 continuous variables, and ∼1300 binary variables (satisfactory solutions were usually found in less than 30 min).

As easily ascertained, both continuous and discretized formulations give rise to difficult instances, despite the quite restrained number of instants covered. A recursive utilization of these models turns out to be very useful in practice. For example, the discretized *Layout* model can be adopted to solve an instance relative to 9 instants and Δ-angles of $\frac{\pi}{24}$ rad, as a first start. The results obtained are utilized to initialize a subsequent optimization, restricting the search to an appropriate neighborhood of the solution found. This allows for the extension of the number of instants and the reduction of the Δ-angles involved, while keeping the model dimension below an acceptable threshold. For instance, 18 instants could be considered with Δ-angles of $\frac{\pi}{72}$ (rad). A combined use of the two *Layout* model versions may also be profitably adopted.

The discretization of the whole time span relevant to the reference operational scenarios gives rise to very large-scale models and this affects the *Master* model directly. For example, an instance with 9 thrusters and 15,000 instants contains:

- $9 \times 15{,}000$ variables representing, respectively, the thrust, for each thruster, at each instant;
- $6 \times 15{,}000$ linear equations (corresponding to (1));
- $9 \times 15{,}000 \times 2$ lower and upper bounds on the thrust variables;
- $2 \times 15{,}000$ linear inequalities (corresponding to (4)).

The tests considered for the *Master* model involved up to ∼135,000 continuous variables and ∼120,000 constraints (optimal solutions were always found in a few minutes).

As mentioned previously, the presence of a nonlinear objective function would make the problem more difficult, significantly increasing the number of variables and constraints involved (in the case of non-convex functions, the introduction of additional binary variables is moreover deemed necessary, see, e.g., [22]). This could be the focus of future investigation.

Global NLP and MILP (deterministic) solvers usually provide (as in the case of the present study) reference bounds on the objective function (e.g., lower bounds, when minimizing), to allow for an evaluation of the quality of the solution(s) obtained (their feasibility is normally verified by the optimizer, with a certain degree of approximation). When, nonetheless, NP-hard problems relevant to real-world applications are involved, especially if at large scale, the proof of optimality (i.e., the convergence to an optimal solution) is not so frequently guaranteed (even when deterministic algorithms, e.g., based on branch and bound search, see, e.g., [23], are utilized).

On the basis of the overall philosophy embraced in this work, it has to be pointed out that the scope of the whole optimization exercise is that of finding satisfactory solutions. Looking for the global optimal one(s), even though a GO approach has been adopted, is indeed renounced a priori. An empirical validation of the solution(s) can, on the other hand, be carried out by the E2E simulator.

The potential of the optimization approach outlined in this chapter is illustrated, in a realistic operative scenario by the following example, based on an overall system configuration, originally considered as the most suitable (it was, however, substituted, afterwards, with an alternative, at present kept as confidential).

The relevant design concept, involving eight engines, is conceived to perform all the requested control tasks during the scientific operational mode, contrasting the atmospheric drag of a typical NGGM orbit, with a mean altitude of about 340 km. It should be noticed that the perturbation forces, in this specific operational scenario, can vary from much less than 1 mN up to 5 mN, depending on the actual solar activity present. The thrust range considered is 50–2500 μN. Figure 11 shows the corresponding propellant consumption, as a function of the thrust exerted.

The thruster orientations, obtained by means of the discretized *Layout* model for the abovementioned atmospheric perturbation assumptions, are compared in Figure 12 with the simplified symmetrical solution of the preliminary design, for which no optimization approach had been adopted.

An example of the *Master* model results, derived in turn by the *Layout* model outcome, is provided. A worst-case instance in terms of propellant consumption, i.e., concerning the maximum solar activity scenario, is illustrated in Figure 13 that reports the commanded forces and torques. In Figure 14, the optimized solution is compared with the one obtained by applying a classical Moore–Penrose inverse matrix to the initial symmetric (non-optimal) thruster layout.

For full controllability, the ensemble of the thrusters must provide positive and negative force components along all three orthogonal axes, and positive and

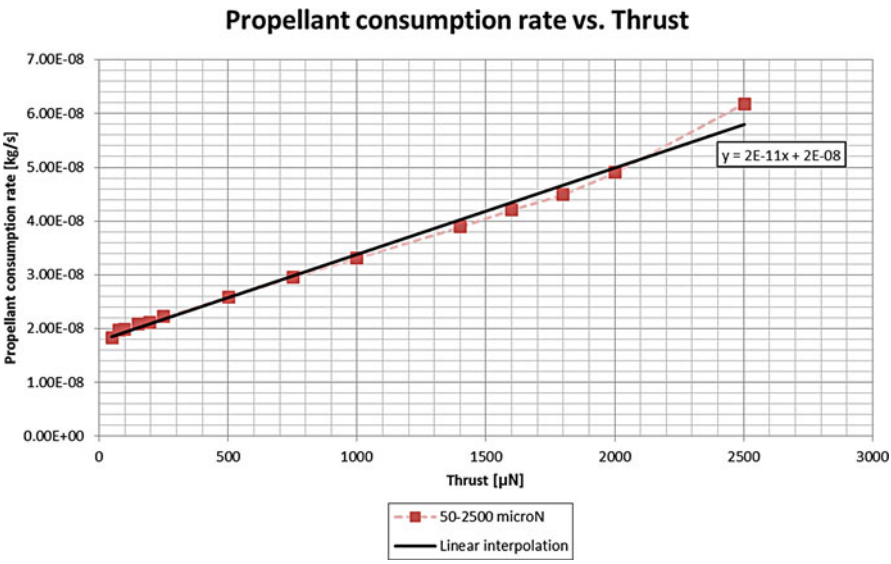


Figure 11 Propellant consumption rate vs. thrust

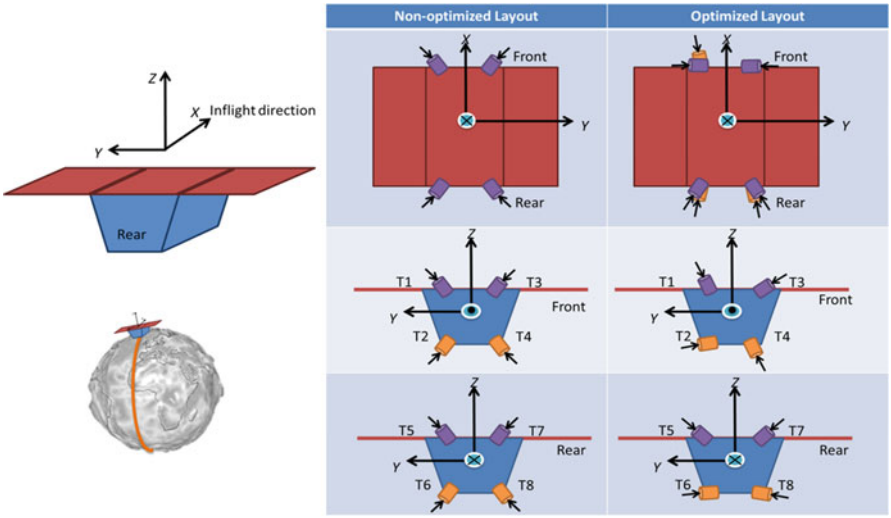


Figure 12 Optimized thruster layout vs. non-optimized solution

negative torque components about all axes. With periodic, zero-mean perturbations, this would produce a symmetric orientation of the thruster ensemble. The optimized solution, instead, takes into account the main asymmetries of the external perturbations by suitably biasing the thruster orientations. The asymmetric (non-zero-mean) perturbing terms include the main component of the drag force along

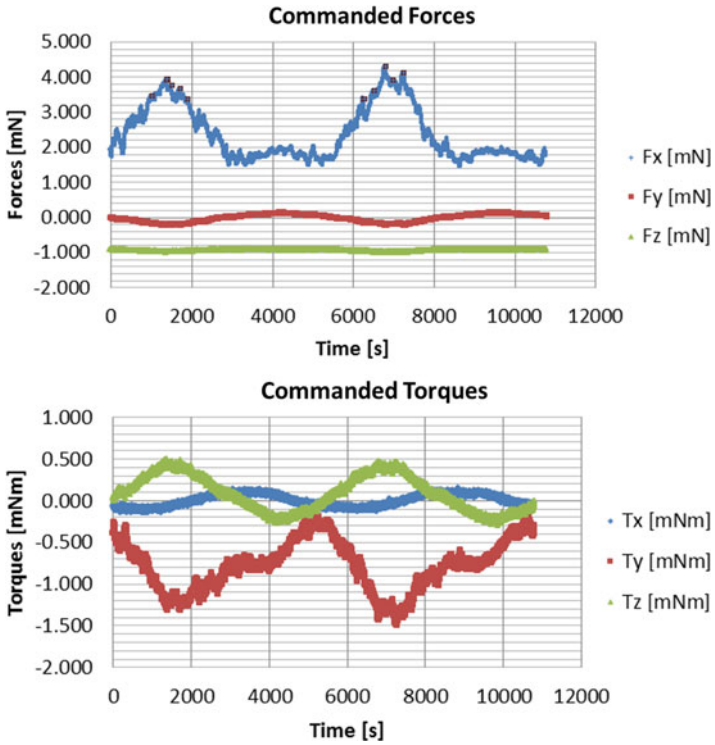


Figure 13 Example of commanded forces and torques in case of maximum solar activity (WC) at an average altitude of 340 km

−X; another large component of the drag force along +Z; and the pitch torque about −Y produced by the centers of mass not coinciding with the center of pressure. Figure 13 shows the corresponding required control forces and torques, with opposite signs. Accordingly, in the optimized solution, all four rear thrusters T5–T8 provide a dominant force component along +X (against the main drag force term), plus all positive and negative torque combinations. The forward thrusters are more individually specialized, providing, in particular, force components along −X, ±Y, and ±Z (T1 and T2), +Y (T3), and −Z (T4) as well as, again, all positive and negative torque combinations. The sensitivity of the solution achieved to the boundary conditions is yet to be studied; it is possible that many nearly equivalent solutions exist in its neighborhood. Anyway, the exercise demonstrates that a significant step toward the objective (minimum propellant consumption requirement) has been achieved.

As per the optimized solution, the total propellant needed to cover a 1-year mission is 5.8 kg, vs the 7.9 kg corresponding to the non-optimized one. The optimized configuration reduces the propellant consumption by about 27%, while providing a quasi-uniformly distributed utilization of the eight thrusters available,

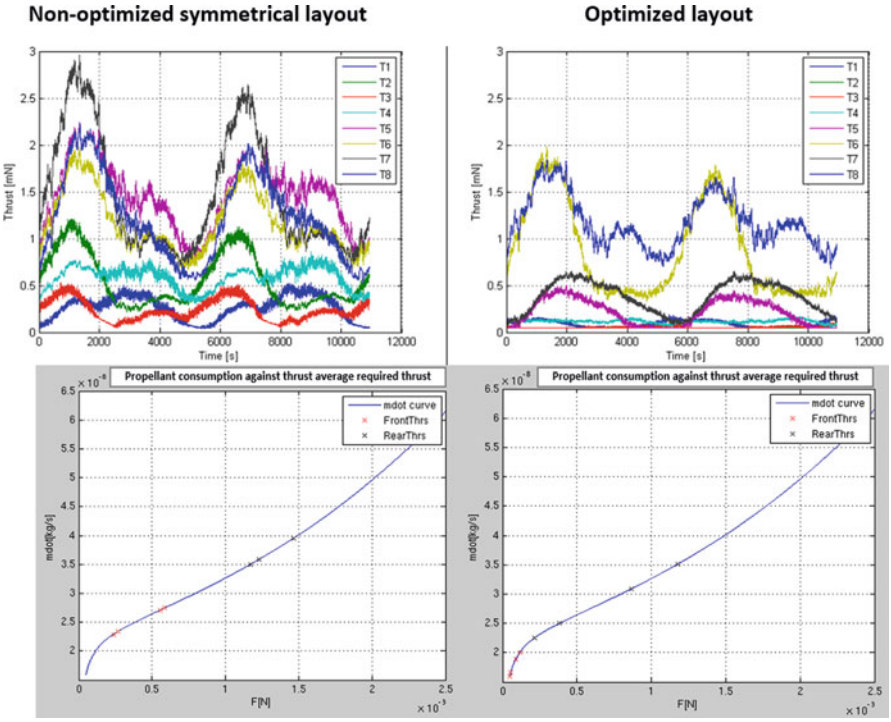


Figure 14 Optimized vs. non-optimized solution

reducing the required thrust range and the thrust saturation probability (with all the annexed technological issues in developing such electric propulsion equipment).

Concerning the computational environment, LGO [24] and IBM ILOG CPLEX 12.3 [25] have been selected as optimizers. More precisely, LGO is intended to solve the continuous version of the *Layout* model, while IBM-CPLEX the discretized one, as well as the *Master* model.

4 Future Developments and Conclusive Remarks

This chapter regards a very demanding study, funded by the European Space Agency (ESA) and currently under development, in support of the future space project denoted as the Next Generation Gravity Mission (NGGM).

Advanced investigations relevant to terrestrial gravimetry are perceived, day after day, as very important, both for the scientific knowledge and also for the social fall-out that is expected to derive from disciplines such as climatology, hydrogeology, and agriculture.

The GOCE (Gravity Field and Steady-State Ocean Circulation Explorer, ESA, 2009–2013) mission is well known for being a successful predecessor of this very ambitious scientific project. NGGM mission, indeed, has been conceived with the scope of outperforming the previous, in terms of both measurement quality and overall scientific target. As is gathered, the challenge inherent to the mission objectives is directly reflected both on the design and the operational features, by very stringent requirements. From a mission point of view, the minimization of the on-board propellant consumption certainly represents a major task, entailing the intrinsically correlated issue of thruster layout optimization. A new methodology, thought up for the control dispatch of a general dynamic system (introduced and discussed at a detailed level in a further chapter of this volume), has been beneficially applied to the NGGM case study.

This chapter focuses on the specific task of determining an optimal configuration for the thrusters, in order to minimize the overall propellant consumption and hence the relevant mass loaded on board. The topical scientific contest is outlined, pointing out its relevance, not only at a speculative level, but also for the envisaged social impact.

The feasibility itself of the NGGM mission, which is expected to cover a time span of 11 years, is strictly dependent on a permanent control of the formation system, acting, step by step, in order to regulate the satellite attitude and to counteract the atmospheric drag, particularly detrimental when navigating in low orbit. Once an appropriate control law is looked into, a further and not any easier task concerns the issue of timely satisfying the recurrent request, in terms of force and torque, by means of the thrusters available on board. Their configuration has to be established by a once-and-for-all solution and, as a consequence, the relevant design choice affects the whole propulsion performance.

In the present study, the positions of the thrusters have been selected mainly on the basis of overall considerations at a system design level. Their optimal orientation, on the other hand, has represented the object of an in-depth investigation. The aforementioned methodology for the control dispatch of a dynamic system has been applied to the specific case study, by considering a suitable selection of low, medium, and high atmospheric density scenarios, assumed to be realistically representative of the whole mission. The overall approach, taking advantage of an advanced global optimization perspective (involving both nonlinear and mixed integer linear programming), essentially consists in the (iterative/recursive) solution of two sub-tasks, referred to as *Layout* and *Master* sub-problems, respectively. Their adoption is outlined in this chapter, underlining the relevant conceptual features (and referring the reader to the dedicated work for an extensive mathematical exposition). The outcomes yielded by an ad hoc experimental analysis, carried out with the support of an end-to-end simulator, implemented to assess the system dynamic behavior, have proven to be promising. A significant part of the chapter is devoted to quite a detailed discussion on this matter.

A number of prospective research directions have been identified so far, also in light of the updated mission requests that have arisen during the consolidation of the NGGM system design. First and foremost, a significant extension of the

experimental analysis is foreseen. To this purpose, further operational scenarios, based on a finer classification of the atmospheric density, have to be examined and the relevant detail level properly upgraded. A wider selection of thruster typologies should be scrutinized, as well, in order to evaluate different technological features, both in terms of lower and upper bounds on the thrusts and the relevant consumption curves. A dedicated sensitivity analysis on the technological features of the thrusters could additionally be beneficial, in order to recommend possible development directions to manufacturers.

In order to perform the abovementioned additional experimental analysis, upgrading of the end-to-end simulator is expected. This has the scope of validating the optimization solutions obtained, by contemplating the more realistic scenarios to deal with, appropriately.

Investigations into aspects that have been neglected in the feasibility analysis achieved to date are moreover envisaged, as essential objectives of the forthcoming project developments. As in the previous stage of the study, these can be obtained as straightforward applications of the theoretical fundamentals put forward in the aforementioned chapter, devoted to the control dispatch of a dynamic system. In this respect, a first research subject could consist in a local optimization, with the scope of improving the positioning of the thrusters (within suitable neighborhoods of the identified locations). The presence of clearance zones should furthermore be taken into account, in order to prevent undesirable interaction among the various devices placed on board for service, payload, and neutralization purposes. The number (in addition to the position) of the frontal, as well as rear actuators, moreover, could be profitably reconsidered, by exploring the suitability of front-back asymmetries.

In order to refine the mathematical models adopted, the linearity assumptions on the propellant consumption curves may be profitably dropped, by considering more representative nonlinear functions as optimization targets. Different objective functions can also be evaluated, with the scope of extending the overall analysis to features that, although not as crucial as the propellant consumption minimization, deserve consideration. Distributing the workload associated with each thruster as uniformly as possible during the whole mission, in order to prevent possible overuse of a sub-set of actuators, is one example. A further extension could concern the realization of a redundant system of thrusters, up to providing, at each instant, the required control dispatch, should an actuator (or more) break down, by minimizing the number of additional devices.

The work discussed in this chapter constitutes a significant step forward toward an in-depth understanding of the very though-provoking scenarios the NGGM study gives rise to, in terms of both design and operational objectives. The way has been paved for taking on an even more demanding challenge on the horizon in the near future.

References

1. Kozai, Y.: The earth gravitational potential derived from satellite motion. Space Sci. Rev. **5**(6), 818–879 (1966)
2. Tapley, B.D., et al.: Station coordinates, baselines, and earth rotation from LAGEOS laser ranging: 1976–1984. J. Geophys. Res. **90**, 9235–9248 (1985)
3. Bender, P.L., et al.: Integrated laser doppler method for measuring planetary gravity fields. In: Colombo, O.L. (ed.) From Mars to Greenland: Charting Gravity with Space and Airborne Instruments. Springer, New York (1992)
4. NRC (National Research Council): Applications of a Dedicated Gravitational Satellite Mission. Technical report, Academy Press, Washington D.C. (1979)
5. Wells, W.C. (ed.): Spaceborne gravity gradiometers. NASA Conference Publication 2305, Greenbelt, Maryland (1984)
6. Williamstown Report: The terrestrial environment. In: Kaula, W. (ed.) Solid-Earth and Ocean Physics: Application of Space and Astronomic Techniques, Report of a Study at Williamstown, Mass., to the NASA (1969)
7. Tapley, B.D., et al.: GRACE measurements of mass variability in the earth system. Science. **305**, 503–506 (2004)
8. Cesare, S., et al.: The European way to gravimetry: from GOCE to NGGM. Adv. Space Res. **57**, 1047–1064 (2016)
9. Rummel, R.: Geoid and gravity in earth sciences – an overview. Earth Moon Planet. **3**(2004), 94 (2004)
10. Sechi, G., et al.: In-flight results from the drag-free and attitude control of the GOCE satellite. In: Proceedings of the 18th IFAC World Congress, Milano, Italy (2011)
11. Gruber, T., et al.: Simulation of the time-variable gravity field by means of coupled geophysical models. Earth Syst. Sci. Data. **3**, 19–35 (2011)
12. Iran Pour S., et al.: Assessment of Satellite Constellations for Monitoring the Variations in Earth's Gravity Field. University of Stuttgart Final Report to ESA (2015)
13. Bender, P.L., et al.: A possible dual-GRACE mission with 90 and 63 inclination orbits. In: Proceedings of 3rd International Symposium on Formation Flying, Missions and Technologies, Noordwijk, The Netherlands (2009)
14. Fasano, G.: Dynamic system control dispatch: a global optimization approach. In: Fasano, G., Pintér, J.D. (eds.) Modeling and Optimization in Space Engineering – State of the Art and New Challenges. Springer, New York (2019)
15. Bacchetta, A., et al.: The results of the next generation gravity mission AOCS study. In: Proceedings of 9th International ESA Conference on GNC Systems (2014)
16. Ben-Tal, A., Nemirovski, A.: On polyhedral approximations of the second-order cone. Math. Oper. Res. **26**(2), 193–205 (2001)
17. Horst, R., Pardalos, P.M., Thoai, N.V.: Introduction to Global Optimization, 2nd edn. Kluwer, Dordrecht (2000)
18. Goldreich, O.: Computational Complexity: a Conceptual Perspective. Cambridge University Press, Cambridge (2008)
19. Jünger, M., et al. (eds.): 50 Years of Integer Programming 1958–2008. From the Early Years to the State-of-the-Art. Springer, Berlin/Heidelberg (2010)
20. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, Berlin/New York (2006)
21. Ruszczyński, A.P.: Nonlinear Optimization, p. xii+454. Princeton University Press, Princeton (2006)
22. Williams, H.P.: Model Building in Mathematical Programming, 5th edn. Wiley, Hoboken (2013)

23. Pintér, J.D.: Global Optimization in Action. Kluwer, Dordrecht (1996)
24. Pintér, J.D.: LGO - A Model Development and Solver System for Global-Local Nonlinear Optimization. User's Guide. (Current edition) Published and distributed by Pintér Consulting Services, Inc., Canada (2017)
25. IBM: IBM ILOG CPLEX Optimization Studio 2017. http://www-03.ibm.com/software/products/fr/ibmilogcpleoptistud. Accessed 1 Dec 2017

Global Optimization of Continuous-Thrust Trajectories Using Evolutionary Neurocontrol

Bernd Dachwald and Andreas Ohndorf

Abstract Searching optimal continuous-thrust trajectories is usually a difficult and time-consuming task. The solution quality of traditional optimal-control methods depends strongly on an adequate initial guess because the solution is typically close to the initial guess, which may be far from the (unknown) global optimum. Evolutionary neurocontrol attacks continuous-thrust optimization problems from the perspective of artificial intelligence and machine learning, combining artificial neural networks and evolutionary algorithms. This chapter describes the method and shows some example results for single- and multi-phase continuous-thrust trajectory optimization problems to assess its performance. Evolutionary neurocontrol can explore the trajectory search space more exhaustively than a human expert can do with traditional optimal-control methods. Especially for difficult problems, it usually finds solutions that are closer to the global optimum. Another fundamental advantage is that continuous-thrust trajectories can be optimized without an initial guess and without expert supervision.

1 Introduction

Traditionally, continuous-thrust trajectories are optimized by the application of numerical optimal-control methods that are based on the calculus of variations. These can be divided into direct ones, such as nonlinear programming (NLP) methods, and indirect ones, such as neighboring extremal and gradient methods. All these are generally classified as local trajectory optimization methods, where

B. Dachwald (✉)
Faculty of Aerospace Engineering, FH Aachen University of Applied Sciences, Aachen, Germany
e-mail: dachwald@fh-aachen.de

A. Ohndorf
German Aerospace Center (DLR), Space Operations and Astronaut Training, Oberpfaffenhofen, Wessling, Germany
e-mail: andreas.ohndorf@dlr.de

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_2

the term optimization does not mean to find the best solution, but rather to find *a* solution. Prior to optimization, the NLP and the gradient methods require an initial guess for the control history, whereas the neighboring extremal methods require an initial guess for the starting adjoint vector of Lagrange multipliers or costate vector [30]. Unfortunately, the convergence behavior of local trajectory optimization methods (especially of the indirect ones) is very sensitive to the initial guess, and an adequate initial guess is often hard to find, even for an expert in astrodynamics and optimal-control theory. Similar initial guesses often produce very dissimilar optimization results, so that the initial guess cannot be improved iteratively and trajectory optimization becomes more of an art than science [3]. Even if the optimizer finally converges to an optimal trajectory, this trajectory is typically close to the initial guess, and that is rarely close to the (unknown) global optimum. Because the optimization process requires nearly permanent expert supervision, the search for a good trajectory can become very time-consuming and thus expensive. Another drawback of local trajectory optimization methods is the fact that the initial conditions (launch date, initial propellant mass, initial velocity vector, etc.)—although they are crucial for mission performance—are generally chosen according to the expert's judgment and are therefore not an explicit part of the optimization problem.

To transcend the drawbacks of local trajectory optimization methods, a smart global trajectory optimization method was developed by Dachwald [5] and Ohndorf [19]. This method—termed evolutionary neurocontrol (ENC)—fuses artificial neural networks (ANNs) and evolutionary algorithms (EAs) into so-called evolutionary neurocontrollers (ENCs). The implementation of ENC for continuous-thrust trajectory optimization was termed InTrance, which stands for *In*telligent *Tra*jectory optimization using *n*euro*c*ontroller *e*volution. To find a near-globally optimal trajectory, InTrance requires only the mission objective and intervals for the initial conditions as input. It does not require an initial guess or the near-permanent supervision of a trajectory optimization expert. During the optimization process, InTrance searches not only the optimal spacecraft control but also the optimal initial conditions within the specified intervals.

In the beginning, to keep it simple, InTrance was designed by Dachwald [5] only for heliocentric single-phase trajectory optimization problems (e.g., to transfer from Earth with a given max. C3 to rendezvous another planet, to escape the solar system with a max. velocity, or to impact an asteroid with a max. velocity). This will be described in Sects. 2 and 4, and selected single-phase results will be shown in Sect. 5. Later, InTrance was extended by Ohndorf [19] for multi-phase problems with more than one central body (e.g., for a transfer from low-Earth-orbit to a low-Moon or a low-Mars-orbit or for multiple-asteroid rendezvous missions). This will be described in Sect. 6 and selected multi-phase results will be shown in Sect. 7.

2 Evolutionary Neurocontrol

2.1 Continuous-Thrust Trajectory Optimization as a Delayed Reinforcement Learning Problem

Reinforcement learning (RL) problems form an important class of difficult learning problems, for which the optimal behavior of the learning system (called agent) has to be learned solely through interaction with the environment, which gives an immediate or delayed evaluation, also called reward or reinforcement [12, 31]. This evaluation is analogous to an objective function and will therefore also be denoted by the symbol J. The agent's behavior is defined by an associative mapping from situations to actions $\mathsf{S} : \mathscr{X} \mapsto \mathscr{A}$ ($\mathscr{X}$ is called state space and $\mathscr{A}$ is called action space). Within this chapter, this associative mapping that is typically called policy in the RL-related literature is termed strategy. The optimal strategy $\mathsf{S}^\star$ of the agent is defined as the one that maximizes the sum of positive reinforcements and minimizes the sum of negative reinforcements over time. If, given a situation $X \in \mathscr{X}$, the agent tries an action $A \in \mathscr{A}$ and the environment immediately returns an evaluation $J(X, A)$ of the (X, A) pair, one has an immediate reinforcement learning problem. Delayed reinforcement learning problems form a class of even more difficult learning problems, for which the environment gives only a single evaluation $J(X, A)[t]$, collectively for the sequence of (X, A) pairs occurring in time during the agent's operation.

From the perspective of machine learning, a spacecraft steering strategy may be defined as an associative mapping S that gives—at any time along the trajectory—the current spacecraft control $\mathbf{u}$ from some input $\mathbf{X}$ that comprises the variables that are relevant for the optimal steering of the spacecraft (the current state of the relevant environment). Because the trajectory is the result of the spacecraft steering strategy, the trajectory optimization problem is actually a problem of finding the optimal spacecraft steering strategy $\mathsf{S}^\star$. This is a delayed reinforcement learning problem because a spacecraft steering strategy cannot be evaluated before its trajectory is known and a reward can be given according to the fulfillment of the optimization objective(s) and constraint(s). One obvious way to implement spacecraft steering strategies is to use ANNs because they have already been applied successfully to learn associative mappings for a wide range of problems. For other recent applications of machine learning and evolutionary techniques in interplanetary trajectory design see also the chapter by Izzo et al. [11] in this book.

2.2 Evolutionary Neurocontrol

InTrance uses a feedforward ANN with a sigmoid neural transfer function. Mathematically, an ANN is a continuous parameterized function (called network function)

$$\mathsf{N}_{\boldsymbol{\pi}} : \mathscr{X} \subseteq \mathbb{R}^{n_i} \mapsto \mathscr{Y} \subseteq (0, 1)^{n_o} \tag{1}$$

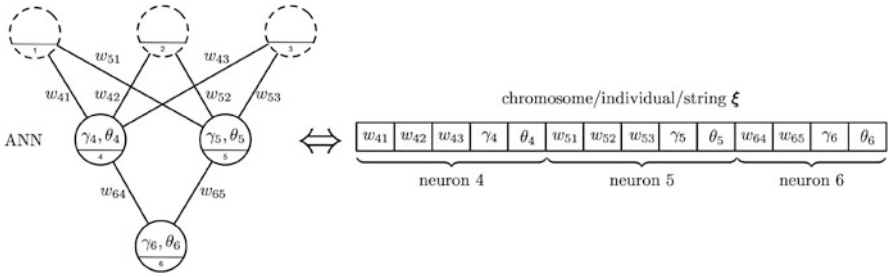


Figure 1 Mapping of an ANN onto a chromosome (note that the input neurons do not have parameters)

that maps from an n_i-dimensional input space $\mathscr{X}$ onto an n_o-dimensional output space $\mathscr{Y}$. The parameter set $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_{n_\pi}\}$ of the network function comprises the n_π internal ANN parameters (i.e., the weights of the neuron connections and the biases of the neurons). ANNs have been successfully applied as neurocontrollers (NCs) for reinforcement learning problems [9]. The most simple way to apply an ANN for controlling a dynamical system is by letting the ANN, at every discrete time step $\bar{t}$, provide the control $\mathbf{u}(\mathbf{Y}(\bar{t}))$ (with $\mathbf{Y}(\bar{t}) \in \mathscr{Y}$) from some input $\mathbf{X}(\bar{t}) \in \mathscr{X}$ that contains the relevant information for the control task. The NC's behavior is completely characterized by its network function $\mathsf{N}_{\boldsymbol{\pi}}$ that is—for a given network topology—again completely characterized by its parameter set $\boldsymbol{\pi}$. If the correct output is known for a set of given inputs (the training set), the difference between the resulting output and the known correct output can be utilized to learn the optimal network function $\mathsf{N}^\star \triangleq \mathsf{N}_{\boldsymbol{\pi}^\star}$ by adapting $\boldsymbol{\pi}$ in a way that minimizes this difference for all input/output pairs in the training set. A variety of learning algorithms has been developed for this kind of learning, the backpropagation algorithm being the most widely known. Unfortunately, learning algorithms that rely on a training set fail when the correct output for a given input is not known, as it is the case for delayed reinforcement learning problems. In this case, EAs may be used as robust learning methods for determining $\mathsf{N}^\star$ [33, 35, 41] because the ANN parameter set $\boldsymbol{\pi}$ can be mapped onto a real-valued string $\boldsymbol{\xi}$ (also called chromosome or individual) that provides an equivalent description of the network function $\mathsf{N}_{\boldsymbol{\pi}}$, as it is shown in Figure 1.

2.3 *Additionally Encoded Problem Parameters*

If an EA is already employed for the NC parameter optimization, it is manifest to use it also for the co-optimization of additional problem parameters. Therefore, InTrance encodes the relevant initial conditions $\mathbf{p}$ additionally on the chromosome (e.g., launch date, hyperbolic excess velocity vector, initial propellant mass), all within user-defined intervals. Thus $\boldsymbol{\xi} = (\boldsymbol{\pi}, \mathbf{p})$. This way, the initial conditions

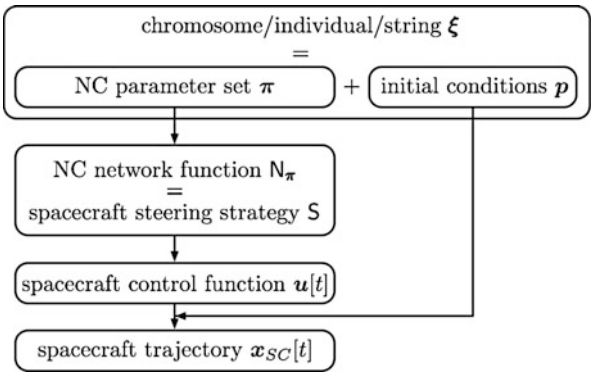


Figure 2 Transformation of a chromosome into a trajectory

are made an explicit part of the optimization problem. Figure 2 sketches the transformation of a chromosome $\boldsymbol{\xi}$ into a trajectory $\mathbf{x}_{\mathrm{SC}}[t]$. By searching for the fittest individual $\boldsymbol{\xi}^{\star}$, the EA searches for the optimal spacecraft trajectory $\mathbf{x}^{\star}_{\mathrm{SC}}[t]$.

2.4 Neurocontroller Input and Output

Two fundamental questions arise concerning the application of an NC for spacecraft steering: (1) "what *input* should the NC get?" (or "what should the NC *know* to steer the spacecraft?") and (2) "what *output* should the NC give?" (or "what should the NC *do* to steer the spacecraft?"

To determine the currently optimal spacecraft control $\mathbf{u}(\bar{t}_i)$, the spacecraft steering strategy should have to know—at *any* time step $\bar{t}_i$—the current spacecraft state $\mathbf{x}_{\mathrm{SC}}(\bar{t}_i)$ and the current target state $\mathbf{x}_{\mathrm{T}}(\bar{t}_i)$. For spacecraft with an electric propulsion system, the spacecraft state includes also the current propellant mass $m_{\mathrm{P}}(\bar{t}_i)$. Thus $\mathsf{S} : \{(\mathbf{x}_{\mathrm{SC}}, \mathbf{x}_{\mathrm{T}}, m_{\mathrm{P}})\} \mapsto \{\mathbf{u}\}$. The number of potential input sets, however, is still large because $\mathbf{x}_{\mathrm{SC}}$ and $\mathbf{x}_{\mathrm{T}}$ may be given in coordinates of any reference frame and in combinations of them. An ANN with cartesian coordinates and the propellant mass as a potential input set is depicted in Figure 3.

Now, what output should the NC give? At any time step $\bar{t}_i$, each output neuron j gives a value $Y_j(\bar{t}_i) \in (0, 1)$. The number of potential output interpretations is also large because there are many alternatives to define $\mathbf{u}$, and to calculate $\mathbf{u}$ from $\mathbf{Y}$. The following approach was chosen in InTrance because it gave good results for the majority of the problems that have been investigated: the NC provides a three-dimensional output vector $\mathbf{d}'' \in (0, 1)^3$ from which a unit vector $\mathbf{d}$ is calculated via

$$\mathbf{d}' \triangleq 2\mathbf{d}'' - \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \in (-1, 1)^3 \quad \text{and} \quad \mathbf{d} \triangleq \frac{\mathbf{d}'}{|\mathbf{d}'|} \tag{2}$$

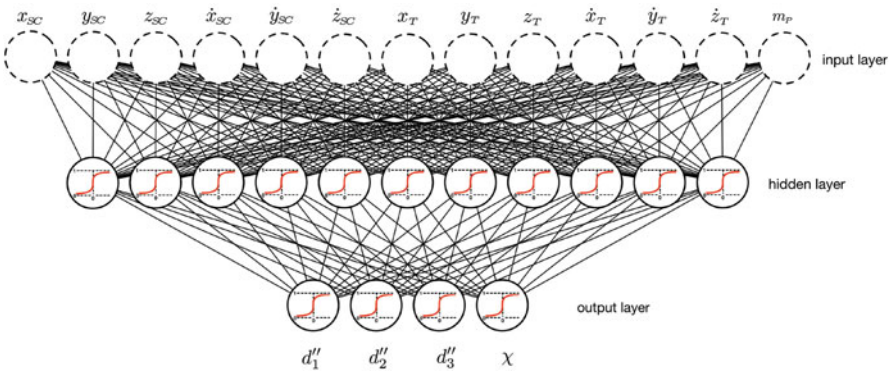


Figure 3 Example for a neurocontroller that implements a spacecraft steering strategy

This unit vector is interpreted as the desired thrust direction and is therefore called direction unit vector. The output must also include the engine throttle $0 \leq \chi \leq 1$, so that $\mathbf{u} \triangleq (\mathbf{d}, \chi)$, hence $\mathsf{S} : \{(\mathbf{x}_{\mathrm{SC}}, \mathbf{x}_{\mathrm{T}}, m_{\mathrm{P}})\} \mapsto \{\mathbf{d}, \chi\}$. This potential output set is also depicted in Figure 3. For bang-bang control, the output of the neuron that is associated with the throttle u_χ may be interpreted as $\chi = 0$ if $u_\chi < 0.5$ and $\chi = 1$ if $u_\chi \geq 0.5$.

2.5 *Neurocontroller Fitness Assignment*

In EAs, the objective function J that rates the optimality of a chromosome is called fitness function. In this section, we show how InTrance implements the fitness function for the rendezvous and the flyby problem.

The optimality of a trajectory may be defined with respect to various primary objectives (e.g., transfer time or propellant consumption). When an ENC is used for trajectory optimization, the accuracy of the trajectory with respect to the final constraints must also be considered as secondary optimization objectives because they are not enforced otherwise. If, for example, the transfer time for a rendezvous is to be minimized, the fitness function must include the transfer time $T \triangleq \bar{t}_f - \bar{t}_0$, the final distance to the target $\Delta r_f \triangleq |\mathbf{r}_{\mathrm{T}}(\bar{t}_f) - \mathbf{r}_{\mathrm{SC}}(\bar{t}_f)|$, and the final relative velocity to the target $\Delta v_f \triangleq |\mathbf{v}_{\mathrm{T}}(\bar{t}_f) - \mathbf{v}_{\mathrm{SC}}(\bar{t}_f)|$, hence $J = J(T, \Delta r_f, \Delta v_f)$. If, for example, the propellant mass for a flyby problem is to be minimized, T and Δv_f are not relevant, but the consumed propellant Δm_{P} must be included in the fitness function. In this case, $J = J(\Delta m_{\mathrm{P}}, \Delta r_f)$. Because the ENC unlikely generates a trajectory that satisfies the final constraints exactly ($\Delta r_f = 0\,\mathrm{m}$, $\Delta v_f = 0\,\mathrm{m/s}$), a max. allowed distance $\Delta r_{f,\max}$ and a max. allowed relative velocity $\Delta v_{f,\max}$ have to be defined. Using $\Delta r_{f,\max}$ and $\Delta v_{f,\max}$, the distance and relative velocity at the target can be normalized to $\Delta R_f \triangleq \Delta r_f / \Delta r_{f,\max}$ and $\Delta V_f \triangleq \Delta v_f / \Delta v_{f,\max}$.

Because at the beginning of the search process most individuals do not meet the final constraints with the required accuracy ($\Delta R_f < 1$, $\Delta V_f < 1$), a max. transfer time $T_{\max}$ must be defined for the numerical integration of the trajectory.

Sub-fitness functions may be defined with respect to all primary and secondary optimization objectives. It was found that the performance of ENC strongly depends on an adequate choice of the sub-fitness functions and on their composition to the (overall) fitness function. This is reasonable because the fitness function has not only to decide autonomously which trajectories are good and which are not, but also which trajectories might be promising in the future optimization process. The primary sub-fitness function

$$J_{m_\mathrm{P}} \triangleq \frac{m_\mathrm{P}(\bar{t}_0)}{2m_\mathrm{P}(\bar{t}_0) - m_\mathrm{P}(\bar{t}_f)} - \frac{1}{3}$$

and the secondary sub-fitness functions

$$J_r \triangleq \log\left(\frac{1}{\Delta R_f}\right) \qquad \text{and} \qquad J_v \triangleq \log\left(\frac{1}{\Delta V_f}\right)$$

were empirically found to produce good results for rendezvous and flyby trajectories, if the propellant consumption is to be minimized. J_r and J_v are positive when the respective accuracy requirement is fulfilled and negative when it is not. Another empirical finding was that the search process should first concentrate on the accuracy of the trajectory and then on the primary optimization objective. Therefore, the sub-fitness function for the primary optimization objective is modified to

$$J'_{m_\mathrm{P}} \triangleq \begin{cases} 0 & \text{if } J_r < 0 \vee J_v < 0 \\ J_{m_\mathrm{P}} & \text{if } J_r \geq 0 \wedge J_v \geq 0 \end{cases}$$

for the rendezvous problem and

$$J'_{m_\mathrm{P}} \triangleq \begin{cases} 0 & \text{if } J_r < 0 \\ J_{m_\mathrm{P}} & \text{if } J_r \geq 0 \end{cases}$$

for the flyby problem. To minimize the propellant mass for a rendezvous, for example, the following fitness function may be conceived:

$$J(\Delta m_\mathrm{P}, \Delta r_f, \Delta v_f) \triangleq J'_{m_\mathrm{P}} + \frac{1}{\sqrt{\Delta R_f^2 + \Delta V_f^2}}$$

To minimize the propellant mass for a flyby, only the positions must match:

$$J(\Delta m_\mathrm{P}, \Delta r_f) \triangleq J'_{m_\mathrm{P}} + \frac{1}{\Delta R_f}$$

2.6 Implementation of Evolutionary Neurocontrol in InTrance

Figure 4 shows how InTrance implements evolutionary neurocontrol for continuous-thrust trajectory optimization. To find the optimal spacecraft trajectory, InTrance runs in two loops. The EA in the outer NC optimization loop holds a population $P = \{\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_q\}$ of chromosomes. Each chromosome $\boldsymbol{\xi}$ of the population comprises an NC parameter set $\boldsymbol{\pi}$ and initial problem conditions $\mathbf{p}$, i.e., $\boldsymbol{\xi} = (\boldsymbol{\pi}, \mathbf{p})$. Every time a new chromosome $\boldsymbol{\xi}_j$ is generated via reproduction, it is examined within the (inner) trajectory integration loop for its suitability to generate an optimal trajectory. Within the inner trajectory integration loop, the NC steers the spacecraft according to the initial conditions $\mathbf{p}_j$ on its chromosome and according to its NC network function, which is completely defined by its parameter set $\boldsymbol{\pi}_j$. Within the trajectory optimization loop, the NC takes the current spacecraft state $\mathbf{x}_{\mathrm{SC}}(\bar{t}_i)$ and the current target state $\mathbf{x}_{\mathrm{T}}(\bar{t}_i)$ as input, and maps them via the network function and the transformations described in Sect. 2.4 onto the spacecraft control $\mathbf{u}(\bar{t}_i)$. Then, $\mathbf{x}_{\mathrm{SC}}(\bar{t}_i)$ and $\mathbf{u}(\bar{t}_i)$ are inserted into the equations of motion and numerically integrated over one time step to yield $\mathbf{x}_{\mathrm{SC}}(\bar{t}_{i+1})$. The new state is then fed back into the NC. The trajectory integration loop stops when the final constraints are met with sufficient accuracy or when a given time limit is reached. Then, back in the NC optimization loop, the trajectory is rated by the EA's fitness function $J(\boldsymbol{\xi}_j)$. The fitness of $\boldsymbol{\xi}_j$ is crucial for its probability to reproduce and create offspring. Under this selection pressure, the EA "breeds" more and more suitable steering strategies that generate better and better trajectories. Finally, the EA that is used within this work converges against a single steering strategy, which, in the best case, gives a near-globally optimal trajectory $\mathbf{x}^{\star}_{\mathrm{SC}}[t]$.

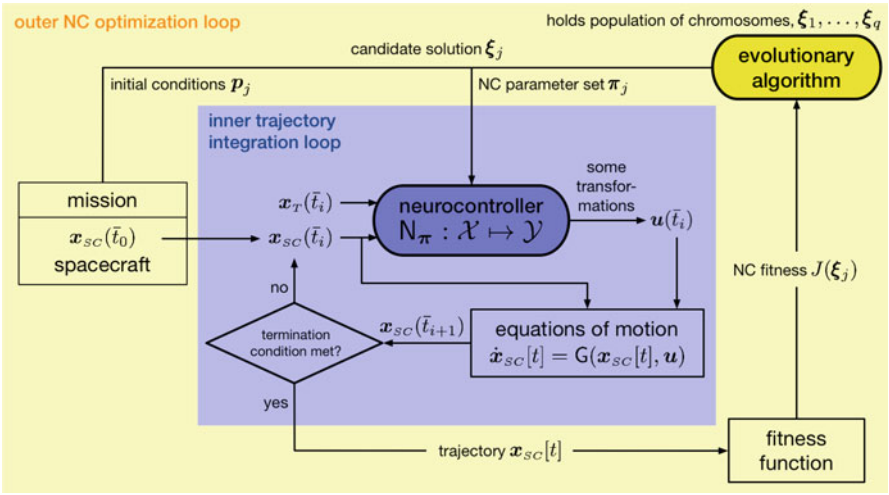


Figure 4 Implementation of evolutionary neurocontrol for continuous-thrust trajectory optimization in InTrance

3 Evolutionary Algorithm Implementation Issues

3.1 Real-Valued Parameter Encoding

InTrance implements real-valued parameter encoding because experiments like those performed by Michalewicz [17] indicate that the real-valued parameter representation is faster, more consistent from run to run, and provides higher precision, especially in large search spaces. It is also closer to the problem space, which facilitates the development of problem-specific evolutionary operators.

3.2 Multi-Objective Tournament Selection and Steady-State Reproduction

InTrance implements tournament selection because of its computational efficiency. A parent individual is selected by choosing randomly two individuals from the population and allowing only the better one to reproduce. Thus the reproduction probability of each individual is independent of its absolute fitness, and the selective pressure remains constant throughout the search process [1]. Another advantage of tournament selection is that each tournament can be performed (randomly) with respect to a different objective. Such a selection mechanism prefers individuals that perform reasonably well with respect to all objectives, supporting multi-objective optimization. Another approach that has been selected due to its computational efficiency is steady-state reproduction, also called one-at-a-time reproduction because, at each time step, only one reproduction takes place. This is computationally less expensive than generational reproduction. Steady-state reproduction in combination with tournament selection, as shown in Figure 5, is conceptually very simple. Two

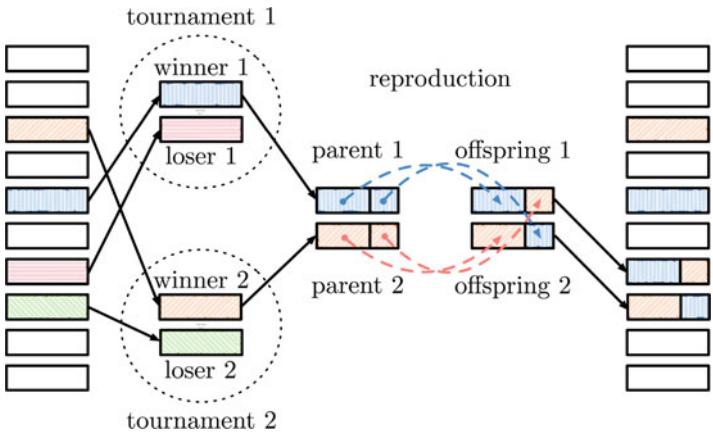


Figure 5 Steady-state reproduction with tournament selection

tournaments are performed to determine the two parent chromosomes, which are the winners of the tournaments. They stay in the population, while the two offspring chromosomes replace the two tournament losers.

3.3 Real Delta Coding

A method termed delta coding (DC) was proposed by Whitley et al. [34] to enhance the precision and convergence behavior of genetic search (for binary strings). It is based on the idea that a string can also express a distance to some previous solution, which is called partial solution. DC runs in search epochs. In each epoch, only a dynamically selected subspace of the total search space, which is constructed around the most recent partial solution, is explored. By periodically re-initializing the population, the partial solution is improved, and premature convergence is avoided. Dachwald [4] and Tsinas and Dachwald [33] extended the idea of DC to real-valued strings and termed this method floating point delta coding (FPDC). The algorithm that is implemented in InTrance is a revised version of FPDC and is termed real delta coding (RDC). It features a search space reduction mechanism that guarantees convergence and provides a kind of gradient search behavior during optimization. Because the details of RDC are quite complex, the reader is referred to [5] for a more detailed description.

3.4 Evolutionary Operators

InTrance implements four crossover operators: one-point crossover [18], uniform crossover [32], arithmetical crossover [17], and finally crossover-nodes [18], which is tailored to ANN chromosomes. InTrance implements also a computationally efficient mutation operator, which is termed fast uniform mutation. For details, the reader is referred to [5] and [19].

4 InTrance Evaluation

To assess the viability and performance of ENC for continuous-thrust trajectory optimization, the convergence behavior of InTrance was evaluated, and the quality of the obtained solutions was assessed by comparing them with reference trajectories for problems found in the literature. All reference problems employed a one-body simulation model, where the gravitational influence of the planets and other disturbing forces were neglected.

Here, we focus on a single reference problem,[1] a solar-sail transfer from Earth to Mercury for an ideal solar sail with a characteristic acceleration[2] of $a_c = 0.55\,\mathrm{mm/s^2}$. A reference trajectory was calculated by Leipold et al. [13, 16] using a local trajectory optimization method. This reference trajectory launches at Earth on 15 Jan 2003 (MJD 52654.5) with zero hyperbolic excess energy and takes 665 days to rendezvous Mercury.

To evaluate the convergence behavior and stability of InTrance, it was run using five different initial NC populations with 50 randomly initialized individuals. The launch date was set to the one used by Leipold et al. [13, 16]. A 12-30-3 neurocontroller with 12 input neurons, one hidden layer with 30 hidden neurons, and 3 output neurons was used, where the input neurons receive the actual spacecraft state $\mathbf{x}_{\mathrm{SC}}$ and the actual target body state $\mathbf{x}_{\mathrm{T}}$ in cartesian coordinates, and the output neurons define—according to Equation (2)—directly the thrust direction unit vector $\mathbf{d}$. The max. transfer time was set to 600 days, and the sailcraft was "allowed" to change its attitude once every day (piecewise constant control). The final accuracy limit (convergence criterion) was set to $\Delta r_{f,\max} = 0.1 \times 10^6\,\mathrm{km}$ and $\Delta v_{f,\max} = 0.1\,\mathrm{km/s}$.[3] Figure 6 shows the trajectories of the five InTrance-runs. The left side of Figure 7 shows the two solar-sail steering angles. Although the trajectories and the steering angles differ considerably for the five runs, the respective transfer times differ by less than 3%. This gives some confidence that the best trajectory found is not far from to the global optimum. Also, it is 91 days (16%) faster than the reference trajectory, revealing that the latter is far from the global

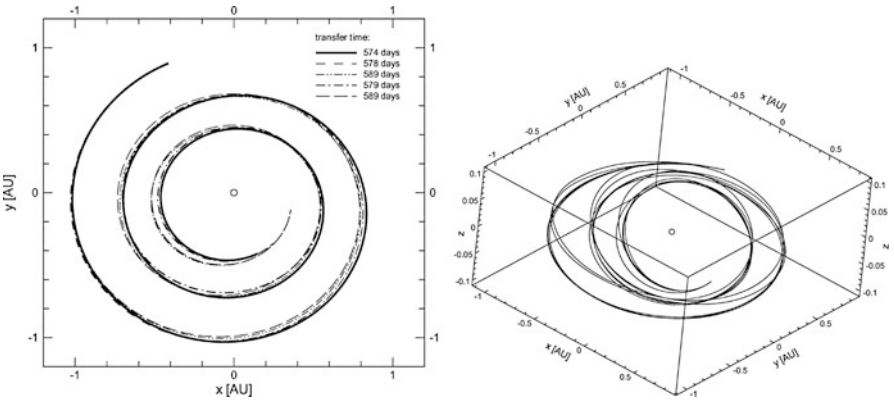


Figure 6 Mercury rendezvous trajectories for the five different initial NC populations, 2D (left) and 3D (right)

[1] A detailed description of all evaluation calculations can be found in [5].

[2] The characteristic acceleration of a solar sail describes its lightness and is its main performance parameter. It is defined as the max. acceleration at Earth distance from the Sun.

[3] $\Delta v_{f,\max} = 0.1\,\mathrm{km/s}$ was also used by Leipold [13], whereas $\Delta r_{f,\max}$ was not given there.

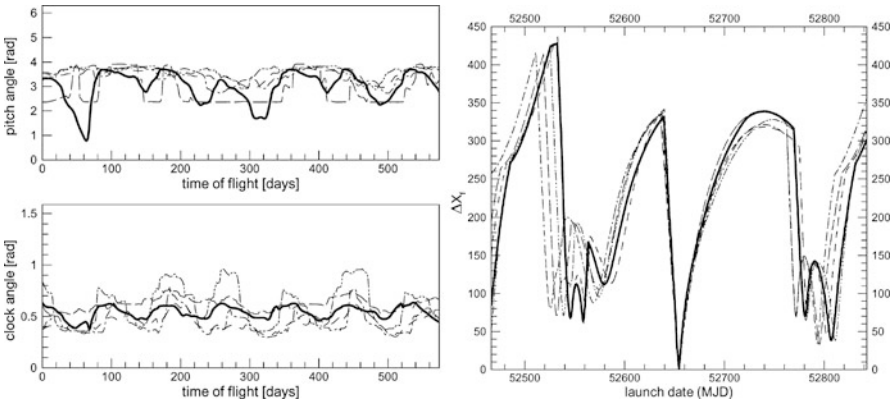


Figure 7 Steering angles (left) and launch date fingerprint (right) for the five different initial NC populations (the bold line is for the best trajectory found, $\Delta X_f = \sqrt{1/2(\Delta R_f^2 + \Delta V_f^2)}$)

optimum. The final distance to Mercury is $\Delta r_f \approx 0.057 \times 10^6$ km and the final relative velocity is $\Delta v_f \approx 0.057$ km/s, both being much better than the required accuracy limits.

To assess the similarity of the five InTrance-steering strategies, so-called launch date fingerprints (LDFs) have been calculated, as they are shown on the right side of Figure 7. These LDFs show the accuracy of the steering strategy for different launch dates within a 1-year interval around the nominal launch date (MJD 52654.5). From the LDFs, one can see that all five steering strategies achieve a good accuracy only for the launch date for which they have been "bred" by the EA, so that they are not *universally* valid. A universal steering strategy would achieve a good accuracy for *all* launch dates. For each specific launch date, the final trajectory deviation is approximately the same for all five ENCs. This similarity of the LDFs indicates that—generating similar outputs from similar[4] inputs—the underlying steering strategies can also be expected to be similar.

Because a 12-30-3 neurocontroller is only one of many possible NCs that may be used for this trajectory optimization problem, different input and output sets and different numbers of hidden neurons/layers have also been tested [5, 6]. It was found that the quality of the solutions is quite robust with respect to those variations. Especially the number of hidden neurons had little effect on the results. Most NCs achieved the required accuracy in all five InTrance-runs, and even NCs with only 10 hidden neurons provided acceptable results. It was noted, however, that only NCs with cartesian inputs achieved the required accuracy in all five runs, although the motion of spacecraft in interplanetary space is better described in polar

[4]Note that the inputs diverge gradually due to the differences in the steering strategies.

coordinates or by orbital elements. This is probably due to the fact that the variation of the input in cartesian coordinates is larger than in polar coordinates and in orbital elements.

In the calculations above, the launch date was fixed. InTrance was later used to further optimize the Earth–Mercury transfer trajectory by varying the launch date within a given interval. The optimal launch date around the reference launch date was found to be 31 Mar 03, 75 days later than the reference launch date. The best solution found takes 502 days to rendezvous Mercury, being 163 days faster than the reference trajectory. The final distance to Mercury is $\Delta r_f \approx 0.02 \times 10^6$ km and the final relative velocity is $\Delta v_f \approx 0.02$ km/s, both being much better than the required values.

5 Selected Single-Phase Trajectory Optimization Results

ENC and InTrance are not constrained by the type of continuous-thrust propulsion system and even trajectories for hybrid or staged continuous-thrust propulsion systems can be optimized. Many of those propulsion systems have already been implemented into InTrance (e.g., solar electric propulsion, nuclear electric propulsion, radioisotope electric propulsion, different thrusters, solar sails, laser-enhanced solar sails). Nevertheless, in this section, two quite difficult solar-sail problems are described, for which no other solutions are known so far. Problems that use other propulsion systems will be covered in Sect. 7.

5.1 Solar System Escape with Solar Sail

Although the solar radiation pressure decreases with the square of the solar distance, solar sails enable fast flyby missions to the outer planets and solar system escape missions, without gravity assist maneuvers. Sauer [27] observed that the solar sail may gain a large amount of orbit energy by making a close approach to the Sun, turning the trajectory into a hyperbolic one, a maneuver for which Leipold coined the term “solar photonic assist” or SPA [14, 15]. Sauer [29] made parametric studies for near-interstellar missions up to 1000 AU using ideal very-high-performance solar sails that reach a high solar system escape velocity with just a single SPA. Leipold observed that solar sails with a more conservative performance require multiple SPAs to achieve solar system escape. He calculated some trajectories for ideal solar sails but made no parametric studies. Both did not make trajectory calculations for low-performance solar sails, probably because the trajectories get more complex (more SPAs) when the sail lightness, i.e., the characteristic acceleration a_c, decreases. Using InTrance, Dachwald [7] was able to make parametric studies also for low-performance solar sails with many SPAs. Some solutions are shown in Figure 8.

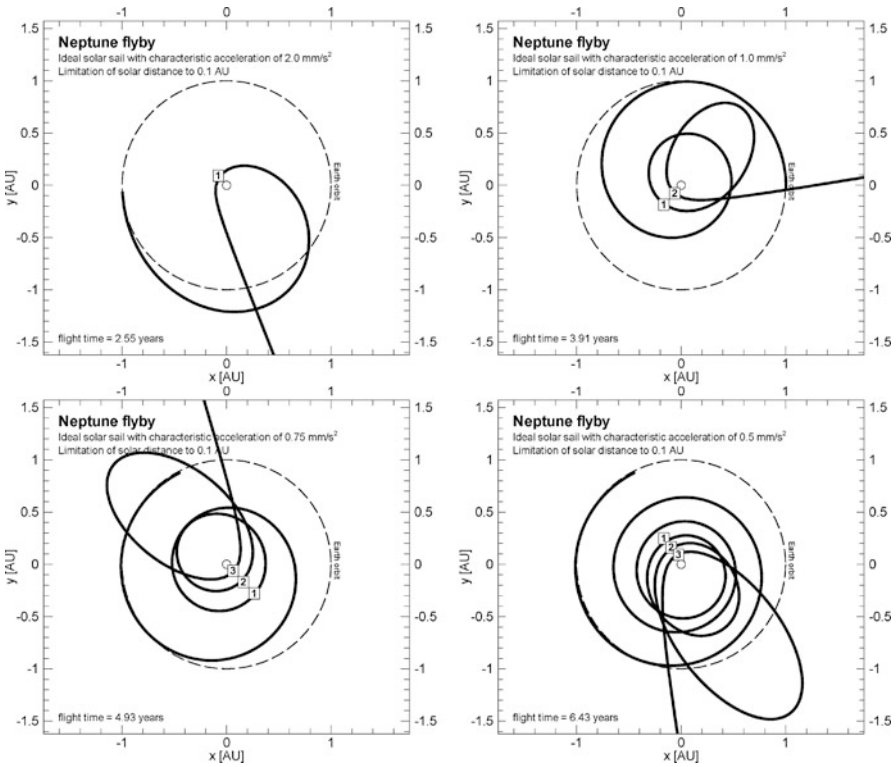


Figure 8 In Trance solutions for Neptune flyby trajectories for ideal solar sails with different characteristic accelerations

The minimum flight time to an outer solar system target depends not only on the lightness of the solar sail but also on the allowed minimum solar distance: the smaller the minimum solar distance, the larger the amount of orbit energy that can be gained by a solar approach. For realistic (non-ideal) sails, the minimum solar distance, however, is limited by the temperature limit T_{lim} of the sail film. In the previous work by Sauer and Leipold, a minimum solar distance r_{min} was used as a path constraint for trajectory optimization, with the argument that such a constraint enforces that some T_{lim} will not be exceeded during the closest solar approach. The sail temperature, however, depends not only on the solar distance but also on the light incidence angle, as defined by the sail attitude. Using InTrance, it was shown in [7] that faster trajectories can be obtained for a given sail temperature limit, if not the allowed minimum solar distance, but the allowed max. sail temperature is directly used as a path constraint for optimization, as shown in Figure 9.

InTrance was used to calculate optimal SPA trajectories to all major outer solar system bodies and to 200 AU for various characteristic accelerations and sail temperature limits. Figure 10 shows that, despite the complex SPA topology, the

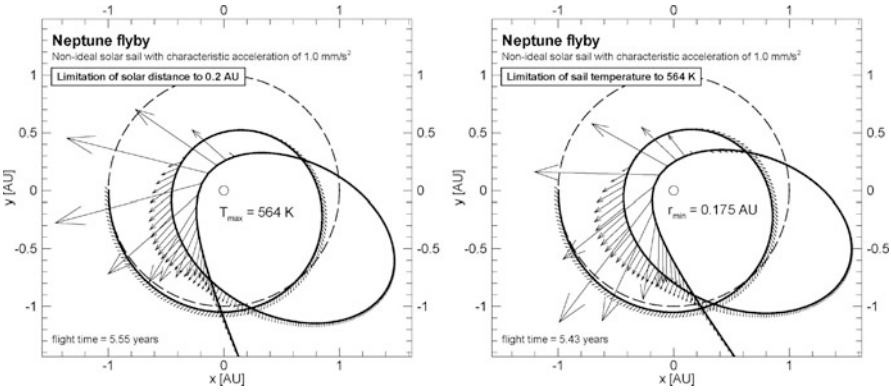


Figure 9 InTrance solutions for Neptune flyby trajectories for a real solar sail with different optimization path constraints, solar distance limit (left) and sail film temperature limit (right)

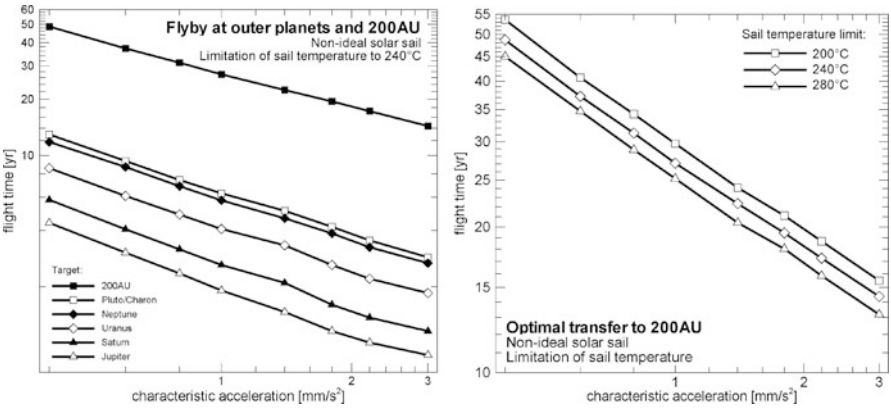


Figure 10 Flight times for outer planet and 200 AU flybys depending on the solar-sail characteristic acceleration (left) and minimum flight times to 200 AU for different sail temperature limits (right)

variation of flight time over characteristic acceleration follows simple power laws. Again, this gives some confidence that the InTrance solutions are close to the global optimum.

5.2 Trajectory Optimization for NASA's Solar Polar Imager Mission

NASA's Solar Polar Imager (SPI) mission was one of the several Sun–Earth connection solar-sail roadmap missions envisioned by NASA. The SPI target orbit was a heliocentric circular orbit at 0.48 AU (in 3:1 resonance with Earth) with an inclination of 75°. The reference mission design was based on a solar sail with a

characteristic acceleration of 0.35 mm/s^2. It was first found by Wright [39, 40] and further examined by Sauer [28] that the best way to perform a large inclination change with a solar sail is to first spiral inwards to a close solar distance, and then to use the large available solar radiation pressure to crank the orbit. NASA's reference trajectory, calculated by Sauer [37] and Wie [36], is shown on the left side of Figure 11. Not approaching the Sun closer than 0.48 AU, the sail film temperature cannot exceed $T_{\mathrm{lim}} = 100\,^{\circ}\mathrm{C}$, which is a quite conservative value. Therefore, this was termed a "cold" mission scenario by Dachwald et al. [8].

Using the same sail film temperature limit as a path constraint, the transfer trajectory found by InTrance, shown on the right side of Figure 11, approaches the Sun closer than the reference trajectory (to about 0.4 AU solar distance) and therefore, exploiting the larger solar radiation pressure, takes—even with a lower hyperbolic excess energy for interplanetary insertion—only 6.4 instead of 6.7 years.

For a higher sail temperature limit of 240 °C (a "hot" mission scenario), the best transfer trajectory found by InTrance, shown on the left side of Figure 12, approaches the Sun much closer (to about 0.22 AU solar distance) before it spirals outwards again to the target orbit, resulting in an even shorter transfer duration of only 4.7 years. Such a trajectory was not seen before with any other optimization method. Looking at the steering angles on the right side of Figure 12, however,

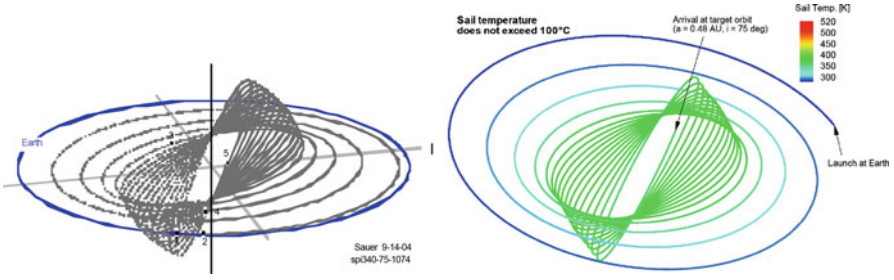


Figure 11 NASA's SPI reference trajectory by Sauer (left, courtesy of NASA/JPL) and InTrance trajectory (right)

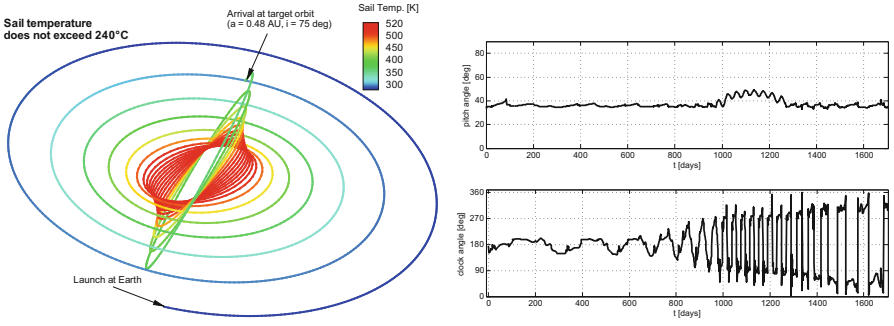


Figure 12 Trajectory (left) and steering angles (right) for the "hot" mission scenario

their "noisy" behavior shows that—due to the poor local search behavior of ENC—the final trajectory is not locally optimal. Therefore, colleagues that work with an indirect local method were asked to use the InTrance trajectory as an initial guess and refine the solution locally. It turned out, however, that this was not possible. But they found out that the found InTrance trajectory is indeed not locally optimal, because they found a solution that is about 1.35% (12 days) faster, after about 900 days of flight time, shortly after the cranking of the orbit has begun (G. Mengali and A. Quarta, 2006, personal communication).

6 Extension of Evolutionary Neurocontrol for Multi-Phase Trajectory Optimization

The optimization of multi-phase continuous-thrust missions provides additional challenges because (1) the optimization objective for the entire mission may be more complex, (2) the distinct phases affect each other during optimization, and (3) additional constraints are introduced, which may be quite complex, as in the case of a min. and max. time between two phases, e.g., for staying at a body. A quite simple two-phase SEP-powered Earth–Mars–Earth transfer may be used to exemplify the potential problems and constraints that may hamper the optimization of such multi-phase mission designs as well as the strategies used in InTrance to tackle those problems.

A multi-phase scenario offers additional optimization objectives compared to those of single-phase missions. Each phase k of the n mission phases has an initial time $t_0^{(k)}$, final time $t_f^{(k)}$, initial propellant mass $m_{\mathrm{P},0}^{(k)}$, final propellant mass $m_{\mathrm{P},f}^{(k)}$, and so on. Besides minimizing the mission duration

$$T_{\mathrm{mission}} = t_f^{(n)} - t_0^{(1)} \quad (= t_f^{(2)} - t_0^{(1)} \text{ for } n = 2) \tag{3}$$

or the total propellant consumption

$$\Delta m_{\mathrm{P}} = m_{\mathrm{P},0}^{(1)} - m_{\mathrm{P},f}^{(n)} \quad (= m_{\mathrm{P},0}^{(1)} - m_{\mathrm{P},f}^{(2)} \text{ for } n = 2) \tag{4}$$

it is also possible to maximize the stay-time between the n phases for a given max. mission duration (as in the case of a multiple-asteroid rendezvous)

$$T_{\mathrm{stay}} = \sum_{k=1}^{n-1} \left(t_0^{(k+1)} - t_f^{(k)} \right) \quad (= t_0^{(2)} - t_f^{(1)} \text{ for } n = 2) \tag{5}$$

or to minimize the total time in flight (e.g., for radiation protection)

$$T_{\mathrm{flight}} = \sum_{k=1}^{n} \left(t_f^{(k)} - t_0^{(k)} \right) \quad (= (t_f^{(1)} - t_0^{(1)}) + (t_f^{(2)} - t_0^{(2)}) \text{ for } n = 2) \tag{6}$$

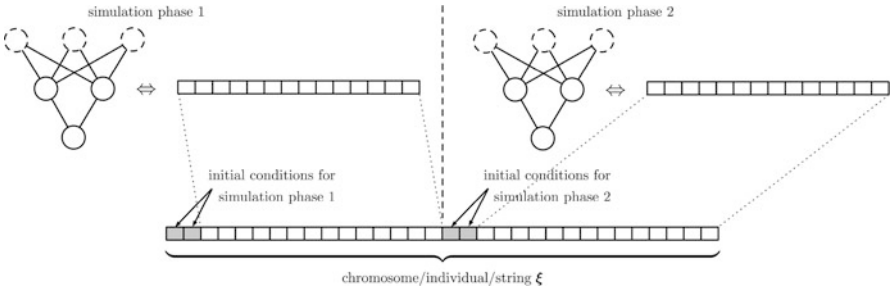


Figure 13 Mapping of the NC parameters onto a chromosome for multi-phase trajectory optimization

Depending on the chosen optimization objective, the phase steering strategies $\mathsf{S}^{(k)}$, the departure dates $t_0^{(k)}$, and the initial propellant masses $m_{\mathrm{P},0}^{(k)}$ will be different, and these are the variables to be optimized in the overall mission context. Even if all single-phase transfers are optimal, the entire mission might be far from being optimal. This is even more complicated by additional constraints like a min. and max. stay-time at Mars, as it would be required by a human mission.

For the multi-phase continuous-thrust optimization approach chosen in InTrance, the chromosome holds all NC and mission- and phase-related parameters, as shown in Figure 13. In this example, the steering strategy of the two phases is incorporated by a separate ANN for each phase. The ANN parameter sets and the initial conditions of each phase become substrings of the chromosome, as it will be justified below. By encoding all phases onto a single chromosome, the solutions for the different phases can co-evolve to yield an optimal solution for the entire trajectory.

The most obvious option for multi-phase trajectory optimization is to evaluate the phases one by one, with phase $k + 1$ always starting exactly at the final state of phase k. This way, only the initial conditions of the first phase would have to be encoded on the chromosome, which reduces the number of optimization parameters. It always guarantees physically valid trajectories. As it constrains the solution space of phase $k + 1$ to a subspace that starts directly at the end of phase k, however, good solutions for later phases cannot be found, if already the first phase drives the optimization into a sub-optimal direction. A better option also evaluates the phases one by one but encodes phase-specific initial conditions additionally on the chromosome. At the expense of additional optimization parameters, the solution space of subsequent phases is therefore not constrained by the solution for the first phase. As a disadvantage, however, this option generates discontinuous trajectories because the initial conditions of phase $k + 1$ now generally differ from the final conditions of phase k. Therefore, the assertion of a continuous and smooth final trajectory is now an additional task for the optimization procedure.

For multi-phase trajectory optimization, InTrance requires a sophisticated mission simulation framework. This was developed by Ohndorf [19], using a linked

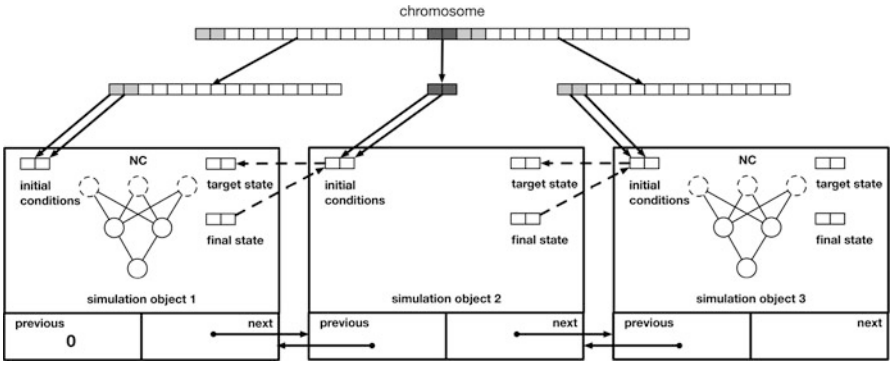


Figure 14 Concatenation of simulation objects

chain of simulation objects for the mission, where each object in the list represents a single mission phase. Starting with the first object, each object calls the following one in sequential order, as soon as a phase transition condition is met (more about phase transitions below). Figure 14 shows this concept for a mission that comprises three phases, with the first and the last phase employing an NC for spacecraft steering and the middle one being a ballistic arc. The simulation object chain is implemented as a doubly linked list. This way, each object can access its predecessor and its successor, and iteratively also all other objects in the list. This assures that each object can access the public information of all other objects, which is important for the optimization of the transition or coupling conditions.

For evaluation, the EA passes each newly generated chromosome to the first object of the (problem-specific) chain of simulation objects. Based on the user-provided problem description, this simulation object reads the initial conditions encoded on the chromosome, initializes an NC with the corresponding values encoded on the chromosome, and integrates the trajectory. Then, based on the achieved final state and the phase-specific target state (rendezvous, flyby, etc.), it computes the chromosome's phase-related sub-fitness. In the case of a flyby problem, the target state may be, for example, a flyby within a required distance of less than 100,000 km. If the achieved state (i.e., the flyby distance), for example, is only 150,000 km the underachievement of 50,000 km is reflected in the phase-related sub-fitness. After the computation of the phase-related sub-fitness, the first simulation object calls the second simulation object and passes the entire chromosome on to its successor in the simulation object chain. The second object then repeats the setup of the phase-related initial conditions setup, as read from the chromosome and from the problem description provided by the user. If the second phase is a free-flight phase, it does not require an NC, as it is shown in the example of Figure 14. In this case, the final state of the phase is determined solely by its initial state and the laws of motion, which include the external forces. Again, the execution of this simulation object ends with the computation of the phase-related sub-fitness. The subsequent call of the third and final object of this

simulation object chain completes the evaluation of the chromosome. Based on all phase-related sub-fitness values, the final object then calculates the overall fitness of the entire chromosome. The algorithm then goes back through the calling chain and into the outer EA loop. An important feature of the implemented simulation chain in Figure 14 is that the target state of a simulation object may be subject to optimization itself. In this example, the first simulation object may take the initial conditions of the second simulation object as its target state. Therefore, the EA not only creates chromosomes that achieve higher phase-related sub-fitness values, i.e., that better solve a space mission's individual flight phases, but implicitly also optimizes the entire mission setup.

7 Selected Multi-Phase Trajectory Optimization Results

This section presents InTrance solutions for two multi-phase continuous-thrust transfer problems, an Earth–Moon transfer and a Dawn-like multiple-asteroid rendezvous with Vesta and Ceres.

7.1 Earth–Moon Transfer

An Earth–Moon transfer provides a good two-phase continuous-thrust example problem. The two phases around the two different central bodies are separated by the crossing of the Moon's gravitational sphere of influence (SOI). The example problem is a time-of-flight optimal transfer from an Earth-bound orbit into a low-Moon orbit with only partly specified final conditions, as shown in Table 1. The resulting trajectory should qualitatively resemble the trajectory of SMART-1, one of the few actually flown continuous-thrust missions so far and the only one to the Moon [2, 10].

The first leg of the Earth–Moon transfer was computed in cartesian Earth-centered inertial (ECI) frame coordinates and the second one in cartesian Moon-centered inertial (MCI) frame coordinates. The handover was at the Moon's SOI, where the state vector at the end of the first phase was taken as the initial state for the second phase. Therefore, the resulting trajectory did not have any discontinuities. The DE405 ephemerides catalog of JPL is used for position and velocity of the Earth and the Moon. The initial spacecraft mass was set to 671 kg, the max. thrust was

Table 1 Initial and target orbit for Earth–Moon transfer

Orbit type	Central body	a	e	i	Ω	ω	M
Initial Earth orbit	Earth	24,460 km	0.73°	6°	279.2°	177.9°	3.5°
Final Moon orbit	Moon	2000 km	0°	84°	Free	Free	Free

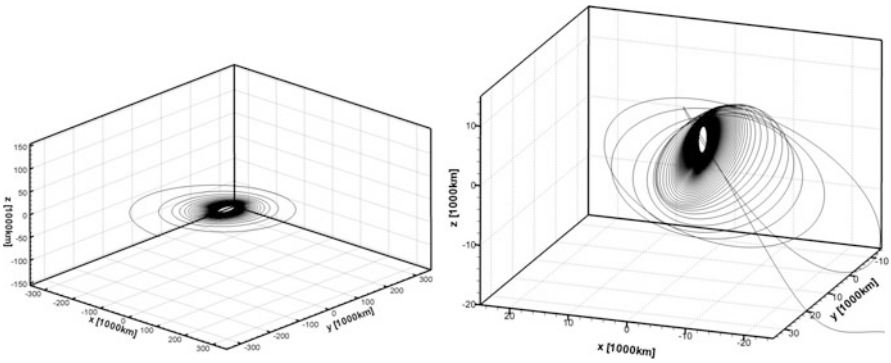


Figure 15 InTrance solution for 2-phase Earth–Moon transfer (left: Earth orbit to Moon SOI, right: Moon SOI to Moon orbit

271 mN, and the specific impulse was 3714 s. Each phase used a dedicated NC for spacecraft steering, the first phase a 26-15-6 NC topology, and the second phase a 23-35-6 NC topology. The resulting trajectory is shown in Figure 15. It has a flight time of 183 days and uses 105 kg of propellant. The final distance to the target orbit is $\Delta r_f = 694$ km and the final relative velocity is $\Delta v_f = 91$ m/s.

This example highlights a substantial advantage of ENC for many-revolution trajectories. Such trajectories have many nodes.[5] Using ENC, the number of optimization parameters is greatly reduced. Originally, one has three optimization parameters for every node (thrust direction and magnitude). While complex trajectories may have in the order of 100,000 node variables, an ANN has only in the order of 1000 variables. In this case, the solution space would be reduced by about 99,000 dimensions. Therefore, ANNs allow the EA an efficient search for the optimal trajectory. The original solution space would be too large to be searched by an EA.

7.2 Dawn-Like Multiple-Asteroid Rendezvous

NASA's Dawn mission [21, 23, 24, 26], set to rendezvous the two largest objects in the main asteroid belt, Vesta and Ceres, is another example for a multi-phase continuous-thrust mission. It uses SEP for the entire mission. Based on the original mission specification, InTrance was used to recalculate the respective phases and to optimize the entire mission for minimum duration [20].

For the recalculation of the real Dawn mission trajectory with InTrance, the following assumptions were made. The launch should not be later than 30 Sep 2007

[5] A node is a trajectory point at which the thrust vector is determined and then held constant for the integration of the equations of motion until the next node.

(MJD 54373). Because InTrance currently still lacks a robust implementation of gravity assists, the mission should not perform a gravity assist at Mars, contrary to the real Dawn mission. A maximum hyperbolic excess energy of $C_3 = 11.3\,\mathrm{km}^2/\mathrm{s}^2$, which corresponds to a hyperbolic excess velocity of 3.362 km/s, was allowed [23], with the direction being subject to optimization by InTrance. Duty cycle limits and the initial operational commissioning phases were not included.

The stay-time at Vesta should be at least 250 days, and the arrival at Ceres should not be later than at the end of March 2015 (MJD 57112). The final accuracy limit was set to $\Delta r_f = 0.5 \times 10^6$ km and $\Delta v_f = 0.5$ km/s. The dry mass was set to 790 kg, and the propellant mass was optimized by InTrance for both phases. Mass reductions due to the hydrazine depletion of Dawn's attitude control system were also not modeled. To account for the load of other spacecraft components, the available electrical power at 1 AU was set to 9.8 kW instead of the 10.3 kW supplied at that distance by the solar generator of the real Dawn spacecraft. Degradation was not simulated, and the power law exponent was set to 1.7, according to [25]. Thrust and propellant consumption curves of the NSTAR ion thruster were taken from [38]. The maximum thruster input power was 2.6 kW.

Figure 16 shows the InTrance solution for the Dawn trajectory. The launch mass is 1139 kg, with 346 kg of propellant. The launch is on 21 Aug 2007 (MJD 54333) with the predefined hyperbolic excess velocity of 3.362 km/s. The interplanetary-insertion direction, however, was determined by InTrance. The spacecraft arrives after 1472 days on 01 Sep 2011 (MJD 55805) at Vesta. The arrival conditions are $\Delta r_f = 0.162 \times 10^6$ km and $\Delta v_f = 0.195$ km/s. 264 kg of propellant was consumed. The stay-time at Vesta is 251 days, and on 08 May 2012 (MJD 56055) Dawn leaves Vesta towards Ceres. After 1047 days, the spacecraft arrives at Ceres on 21 Mar 2015 (MJD 57054) with $\Delta r_f = 0.480 \times 10^6$ km and $\Delta v_f = 0.239$ km/s. The total mission duration until arrival at Ceres is 2770 days.

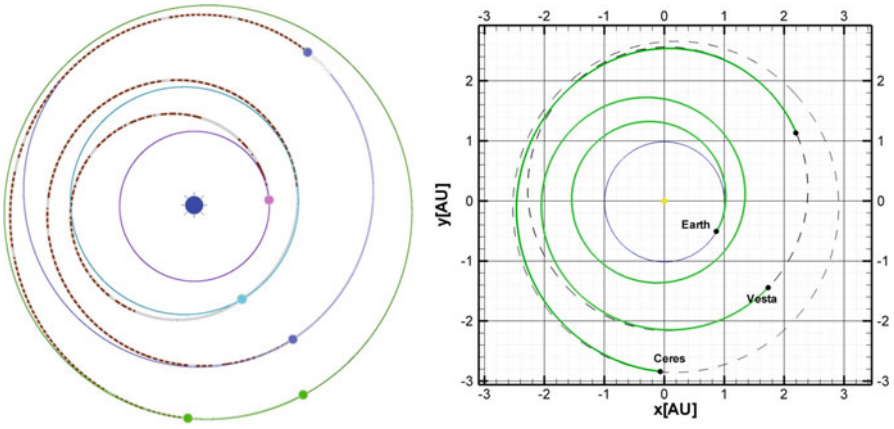


Figure 16 Original Dawn reference trajectory (left, taken from [22], courtesy of NASA/JPL) and InTrance solution (without Mars gravity assist, right)

Except for the excluded Mars gravity assist, the InTrance trajectory on the right side of Figure 16 looks very similar to the reference trajectory on the left side. Its propellant consumption, however, is 20% less than the reference value of 425 kg. With the original propellant mass, however, the transfers from Earth to Vesta and from Vesta to Ceres are not achievable without a violation of set constraints, i.e., mission duration or stay-time at Vesta, probably due to the missing Mars gravity assist. Dawn, however, requires this additional 87 kg of propellant for orbit maneuvers at the asteroids and as a safety margin in the propellant budget. The implementation of ENC for heliocentric multi-rendezvous missions in InTrance is nevertheless considered valid, as the described recalculation led to qualitatively similar results. Due to the Mars gravity assist of the reference trajectory, a deeper comparison is difficult.

8 Chapter Summary and Conclusions

Evolutionary neurocontrol (ENC) is a novel approach to calculate a wide variety of continuous-thrust trajectories for various propulsion systems and trajectory optimization problems that may be close to the (unknown) global optimum. Its trajectory optimization software implementation—InTrance—also allows the co-optimization of the initial conditions for a mission. The solutions obtained with InTrance are at least similar and for difficult problems usually even better than expert-generated solutions (using local trajectory optimization methods). To the best of our knowledge, ENC has produced the only near-globally optimal trajectories for some of the very difficult problems shown in this chapter. The application of InTrance requires no initial guess and only basic knowledge in astrodynamics. Knowledge of optimal-control theory is not needed. In principle, ENC can also be applied for various other difficult and/or poorly structured optimal-control problems in space mission design and engineering, like attitude control, launcher ascent trajectories, atmospheric entry, descent, and landing trajectories, autonomous planetary surface and subsurface vehicles, autonomous flight around small bodies. The primary focus of future enhancements is currently the robust implementation of gravity assists and the coupling with a local trajectory optimization for the fine-tuning of the final results.

References

1. Bäck, T.: Evolutionary algorithms. ACM SIGBIO Newsl. **12**, 26–31 (1992)
2. Betts, T.J., Erb, S.O.: Optimal low thrust trajectories to the moon. SIAM J. Appl. Dyn. Syst. **2**(2), 144–170 (2003)
3. Coverstone-Carroll, V., Hartmann, J., Mason, J.: Optimal multi-objective low-thrust spacecraft trajectories. Comput. Methods Appl. Mech. Eng. **186**, 387–402 (2000)

4. Dachwald, B.: Optimierung des Lernverhaltens neuronaler Netze mit Hilfe genetischer Algorithmen. Diploma thesis, Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik, Institut für Meßtechnik, July 1993 (in German)
5. Dachwald, B.: Low-thrust trajectory optimization and interplanetary mission analysis using evolutionary neurocontrol. Doctoral thesis, Universität der Bundeswehr München, Fakultät für Luft- und Raumfahrttechnik (2004)
6. Dachwald, B.: Optimization of interplanetary solar sailcraft trajectories using evolutionary neurocontrol. J. Guid. Control. Dyn. **27**(1), 66–72 (2004)
7. Dachwald, B.: Optimal solar sail trajectories for missions to the outer solar system. J. Guid. Control. Dyn. **28**(6), 1187–1193 (2005)
8. Dachwald, B., Ohndorf, A., Wie, B.: Solar sail trajectory optimization for the Solar Polar Imager (SPI) mission. In: AAS/AIAA Astrodynamics Specialist, Conference, Keystone, August 2006. AAS Paper 2006-6177
9. Dracopoulos, D.: Evolutionary Learning Algorithms for Neural Adaptive Control. Perspectives in Neural Computing. Springer, Berlin (1997)
10. Herman, A.L., and Conway, B.A.: Optimal, low-thrust, earth-moon orbit transfer. J. Guid. Control. Dyn. **21**(1), 131–147 (1998)
11. Izzo, D., Sprague, C., Tailor, D.: Machine learning and evolutionary techniques in interplanetary trajectory design. In: Fasano, G., Pintér, J. (eds.) Modeling and Optimization in Space Engineering. Springer International Publishing, Cham (2019)
12. Keerthi, S., Ravindran, B.: A tutorial survey of reinforcement learning. Technical report, Department of Computer Science and Automation, Indian Institute of Science, Bangalore (1995)
13. Leipold, M.: Solar sail mission design. Doctoral thesis, Lehrstuhl für Flugmechanik und Flugregelung; Technische Universität München (1999). DLR-FB-2000-22
14. Leipold, M.: To the sun and Pluto with solar sails and micro-sciencecraft. Acta Astronaut. **45**(4–9), 549–555 (1999)
15. Leipold, M., Wagner, O.: 'Solar photonic assist' trajectory design for solar sail missions to the outer solar system and beyond. In: Stengle, T. (ed.) Spaceflight Dynamics 1998, vol. 100, Part 2 of Advances in the Astronautical Sciences, pp. 1035–1045. Univelt, Inc., Escondido (1998)
16. Leipold, M., Seboldt, W., Lingner, S., Borg, E., Herrmann, A., Pabsch, A., Wagner, O., Brückner, J.: Mercury sun-synchronous polar orbiter with a solar sail. Acta Astronaut. **39**(1–4), 143–151 (1996)
17. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, 3rd, revised and extended edition. Springer, Berlin (1999)
18. Montana, D., Davies, L.: Training feedforward neural networks using genetic algorithms. In: Joint Conference on Artificial Intelligence, Los Altos (1989)
19. Ohndorf, A.: Multiphase low-thrust trajectory optimization using evolutionary neurocontrol. Doctoral thesis, TU Delft, Space Systems Engineering (2016)
20. Ohndorf, A., Dachwald, B.: InTrance - a tool for multi-objective multi-phase low-thrust trajectory optimization. In: 4th International Conference on Astrodynamic Tools and Techniques (2010)
21. Rayman, M., Mase, R.: The second year of Dawn mission operations: Mars gravity assist and onward to Vesta. Acta Astronaut. **67**(3–4), 483–488 (2010)
22. Rayman, M., Mase, R.: Dawn's exploration of Vesta, Naples (2012)
23. Rayman, M., Patel, K.: The Dawn project's transition to mission operations: on its way to rendezvous with (4) Vesta and (1) Ceres. Acta Astronaut. **66**(1–2), 230–238 (2010)
24. Rayman, M., Fraschetti, T., Raymond, C., Russel, C.: Dawn: a mission in development for exploration of main belt asteroids Vesta and Ceres. Acta Astronaut. **58**(11), 605–616 (2006)
25. Rayman, M., Fraschetti, T., Raymond, C., Russel, C.: Coupling of system resource margins through the use of electric propulsion: implications in preparing for the Dawn mission to Ceres and Vesta. Acta Astronaut. **60**, 930–938 (2007)

26. Russel, C., Barucci, M., Binzel, R., Capria, M., Christensen, U., Coradini, A., De Sanctis, M., Feldman, W., Jaumann, R., Keller, H., Konopliv, A., McCord, T., McFadden, L., McKeegan, K., McSween, H., Raymond, C., Sierks, H., Smith, D., Spohn, T., Sykes, M., Vilas, F., Zuber, M.: Exploring the asteroid belt with ion propulsion: dawn mission history, status and plans. Adv. Space Res. **40**, 193–201 (2007)
27. Sauer, C.: Optimum solar-sail interplanetary trajectories. In: AIAA/AAS Astrodynamics Conference, San Diego, August 1976. AIAA Paper 76-792
28. Sauer, C.: A comparison of solar sail and ion drive trajectories for a Halley's comet rendezvous mission. In: AAS/AIAA Astrodynamics Conference, Jackson, September 1977. AAS Paper 77-104
29. Sauer, C.: Solar sail trajectories for solar polar and interstellar probe missions. In: Howell, K., Hoots, F., Kaufman, B. (eds.) Astrodynamics 1999. Advances in the Astronautical Sciences, vol. 103, pp. 547–562. Univelt, Inc., Escondido (2000)
30. Stengel, R.: Optimal Control and Estimation. Dover Books on Mathematics. Dover, New York (1994)
31. Sutton, R., Barto, A.: Reinforcement Learning. MIT Press, Cambridge (1998)
32. Syswerda, G.: Uniform crossover in genetic algorithms. In: Third International Conference on Genetic Algorithms, San Mateo (1989)
33. Tsinas, L., Dachwald, B.: A combined neural and genetic learning algorithm. In: Proceedings of the 1st IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, 27–29 June 1994, Orlando, vol. 2, pp. 770–774. IEEE, Piscataway (1994)
34. Whitley, D., Mathias, K., Fitzhorn, P.: Delta coding: an iterative search strategy for genetic algorithms. In: Fourth International Conference on Genetic Algorithms, San Mateo (1991)
35. Whitley, D., Dominic, S., Das, R., Anderson, C.: Genetic reinforcement learning for neurocontrol problems. Mach. Learn. **13**, 259–284 (1993)
36. Wie, B.: Thrust vector control of solar sail spacecraft. In: AIAA Guidance, Navigation, and Control Conference, San Francisco, August 2005. AIAA Paper 2005-6086
37. Wie, B., Thomas, S., Paluszek, M., Murphy, D.: Propellantless AOCS design for a 160-m, 450-kg sailcraft of the Solar Polar Imager mission. In: 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Tucson, July 2005. AIAA Paper 2005-3928
38. Williams, S.N., Coverstone-Carroll, V.: Mars missions using solar electric propulsion. J. Spacecr. Rocket. **37**(1), 71–77 (2000)
39. Wright, J.: Solar sailing – evaluation of concept and potential. Technical report, Battelle Columbus Laboratories, Columbus, OH, March 1976. BMI-NLVP-TM-74-3
40. Wright, J., Warmke, J.: Solar sail mission applications. In: AIAA/AAS Astrodynamics Conference, San Diego, 18–20 August 1976. AIAA Paper 76-808
41. Yao, X.: Evolutionary artificial neural networks. In: Kent, A., et al. (eds.) Encyclopedia of Computer Science and Technology, vol. 33, pp. 137–170. Marcel Dekker, New York (1995)

Nonparametric Importance Sampling Techniques for Sensitivity Analysis and Reliability Assessment of a Launcher Stage Fallout

Pierre Derennes, Vincent Chabridon, Jérôme Morio, Mathieu Balesdent, Florian Simatos, Jean-Marc Bourinet, and Nicolas Gayton

Abstract Space launcher complexity arises, on the one hand, from the coupling between several subsystems such as stages or boosters and other embedded systems, and on the other hand, from the physical phenomena endured during the flight. Optimal trajectory assessment is a key discipline since it is one of the cornerstones of the mission success. However, during the real flight, uncertainties can affect the different flight phases at different levels and be combined to lead to a failure state of the space vehicle trajectory. After their propelled phase, the different stages reach successively their separation altitudes and may fall back into the ocean. Such

P. Derennes
ONERA/DTIS, Université de Toulouse, Toulouse, France

UPS IMT, Université de Toulouse, Toulouse, France
e-mail: pierre.derennes@onera.fr

V. Chabridon
ONERA/DTIS, Université Paris Saclay, Palaiseau Cedex, France

Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, Clermont-Ferrand, France
e-mail: vincent.chabridon@onera.fr

J. Morio (✉)
ONERA/DTIS, Université de Toulouse, Toulouse, France
e-mail: jerome.morio@onera.fr

M. Balesdent
ONERA/DTIS, Université Paris Saclay, Palaiseau Cedex, France
e-mail: mathieu.balesdent@onera.fr

F. Simatos
Institut Supérieur de l'Aéronautique et de l'Espace (ISAE-SUPAERO), Université de Toulouse, Toulouse, France

J.-M. Bourinet · N. Gayton
Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, Clermont-Ferrand, France

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_3

a dynamic phase is of major importance in terms of launcher safety since the consequence of a mistake in the prediction of the fallout zone can be dramatic in terms of human security and environmental impact. For that reason, the handling of uncertainties plays a crucial role in the comprehension and prediction of the global system behavior. Consequently, it is of major concern to take them into account during the reliability analysis. In this book chapter, two new sensitivity analysis techniques are considered to characterize the system uncertainties and optimize its reliability.

1 Introduction

During the launch of a satellite or other space systems, the most important event is, of course, the ascent phase. Nevertheless, a successful launch is not the end of the launcher task. Once their mission is completed, the launch vehicle stages are jettisoned and fall back into the ocean. The estimation of launch vehicle fallout safety zone is a crucial problem in aerospace since it potentially involves dramatic repercussions on the population and the environment [35].

The goal of this chapter is to illustrate the use of advanced sensitivity analysis methods on an aerospace test-case. For that purpose, a simplified fallout trajectory simulation is used to be representative of the phenomena encountered but with a reduced simulation cost (e.g., use of mass point model). The performance obtained for the proposed sensitivity analysis methods is independent of the fidelity of the model. For more realistic problems such as [22], other parameters can be considered in the dynamics of the vehicle (e.g., perturbation of atmospheric density, winds) but are not taken into account in this study for the sake of simplicity and interpretation of the given results.

The launcher stage fallout simulation may be modeled as an input-output black-box function. The inputs are notably some characteristics of the launch vehicle and some conditions (initial or arising during the flight) of the fallout. They are affected by epistemic and aleatory uncertainties and considered as a random vector with a given probability density function (PDF). It is assumed that this PDF is described by a parametric model of density. The output corresponds to the position of fallout and is also a random variable because of the input randomness. A quantity of interest is notably the probability that a launch vehicle stage falls at a distance greater than a given safety limit. Indeed this estimation is strategic for the qualification of such vehicles.

Determining the most important inputs on the launcher stage impact point position and on its failure probability is thus a key question regarding safety. It is exactly the purpose of sensitivity analysis. Indeed, the aim of sensitivity analysis of model output (SAMO) [24] is to study how the output of the simulation model varies regarding the inputs. It enables, for instance, to identify model inputs that cause significant uncertainty in the output and should therefore be the focus of attention or to fix model inputs that have no effect on the output. Reliability-based sensitivity

analysis (RbSA) [8, 28] aims at quantifying the impact of the variability affecting any input quantity on an estimated output measure of safety. In this second type of sensitivity analysis, the quantity of interest is no longer the model output, but a reliability measure such as, in the present chapter, the failure probability of the launch vehicle stage fallout. SAMO and RbSA are quite complementary techniques because a given input may have a negligible influence on the whole output variation, but could not be neglected for the estimation of a failure probability and conversely.

SAMO and RbSA require the estimation of multidimensional integrals. For that purpose, Monte Carlo sampling [41] is a well-known approach that takes advantage of the law of large numbers. To decrease the variance of Monte Carlo sampling estimate, different techniques have been proposed. In this chapter, both SAMO and RbSA are performed using a nonparametric importance sampling (NIS) technique [9] whose aim is to estimate the optimal auxiliary sampling distribution of an integral with kernel density estimators [45].

This book chapter is organized as follows. First, a brief presentation of the launcher stage fallout test-case is proposed in Sect. 2 and a description of its different inputs and output is provided. Then, a new estimation scheme of moment independent SAMO measure is given in Sect. 3 followed by an application to the launcher stage fallout test-case to determine the most influential inputs on the output distribution. Sect. 4 of this chapter is devoted to RbSA with local approach to study how the uncertainty on some input variables plays a role on the variability of the failure probability. To sum up, Sect. 5 aims at providing a brief synthesis of the different results of sensitivity analysis. Finally, a conclusion gathering the most important outcomes of this chapter is given in Sect. 6.

2 Launcher Stage Fallout and the Uncertainty Quantification Methodology

2.1 Description of the Test-Case

Space launcher complexity arises from the coupling between several subsystems, such as stages or boosters and other embedded systems. Optimal trajectory assessment is a key discipline since it is one of the cornerstones of the mission success (for ascent as well as for re-entry trajectories). However, during the real flight, aleatory uncertainties can affect the different flight phases at different levels (e.g., due to weather perturbations or stage combustion) and be combined to lead to a failure state of the space vehicle trajectory. After their propelled phase, the different stages reach successively their separation altitudes and may fall back into the ocean (see Figure 1). Such a dynamic phase is of utmost importance in terms of launcher safety since the consequence of a mistake in the prediction of the fallout zone can be dramatic in terms of human security and environmental impact. As a

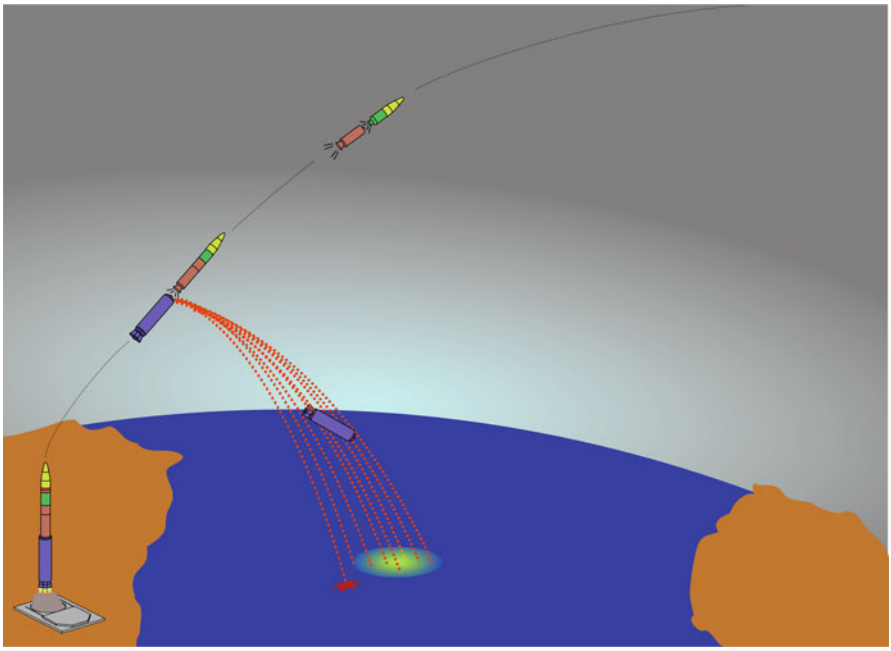


Figure 1 Illustration scheme of a launch vehicle first stage fallout phase into the Atlantic Ocean. Multiple fallout trajectories are drawn (red dotted lines), leading to the safe zone (yellow circular surface). Due to uncertainties, one fallout trajectory may lead to a failure impact point (red star)

consequence, the handling of uncertainties plays a crucial role in the comprehension and prediction of the global system behavior. That is the reason why it is of prime importance to take it into account during the reliability analysis.

The simulation model used in this chapter can be considered as a black-box model denoted by $\mathcal{M} : \mathbb{R}^{d=6} \to \mathbb{R}$. Here, it is a simplified trajectory simulation code of the dynamic fallout phase of a generic launcher first stage [31]. The advantage of a black-box model is to enlarge the applicability of the proposed statistical approaches illustrated in this chapter to any test-cases in this range of models. As a matter of fact, the following methods proposed in this chapter are said to be "non-intrusive" with respect to the model under study. The d-dimensional (here $d = 6$) input vector of the simulation code, denoted $\mathbf{X}$, contains the following basic variables (i.e., physical variables) representing some initial conditions, environmental variables, and launch vehicle characteristics:

X_1: stage altitude perturbation at separation (Δa (m));
X_2: velocity perturbation at separation (Δv (m s^{-1}));
X_3: flight path angle perturbation at separation ($\Delta\gamma$ (rad));
X_4: azimuth angle perturbation at separation ($\Delta\psi$ (rad));
X_5: propellant mass residual perturbation at separation (Δm (kg));
X_6: drag force error perturbation (ΔC_d dimensionless).

These variables are assumed to be independent for the sake of simplicity. As an output, the code will give back the scalar distance $Y = \mathcal{M}(\mathbf{X})$ which represents the distance D_{code} between the theoretical fallout position into the ocean and the estimated one due to the uncertainty propagation.

2.2 *Uncertainty Quantification Methodology Applied to the Launcher Stage Fallout Code*

Uncertainty quantification (UQ) methodology is devoted to the study of the impact of input uncertainties on the behavior of a complex system. Figure 2 provides a summary of the main UQ steps. Starting from the physical black-box model $\mathcal{M}(\cdot)$ (cf. block A) and assuming that this model is verified, calibrated, and validated [39], one may identify and represent the input uncertainties by choosing a dedicated mathematical formalism (probabilistic or extra-probabilistic [1]) to encode them (cf. block B). The choice of the formalism depends on the input available information. Here, it is supposed that enough data is available to postulate existence of the input PDFs. Then, one can distinguish between two phases:

– the *forward* UQ which corresponds to the propagation of input uncertainties to the output (cf. block C) but also contains the analysis of the output variability and reliability assessment through the definition of a failure criterion;

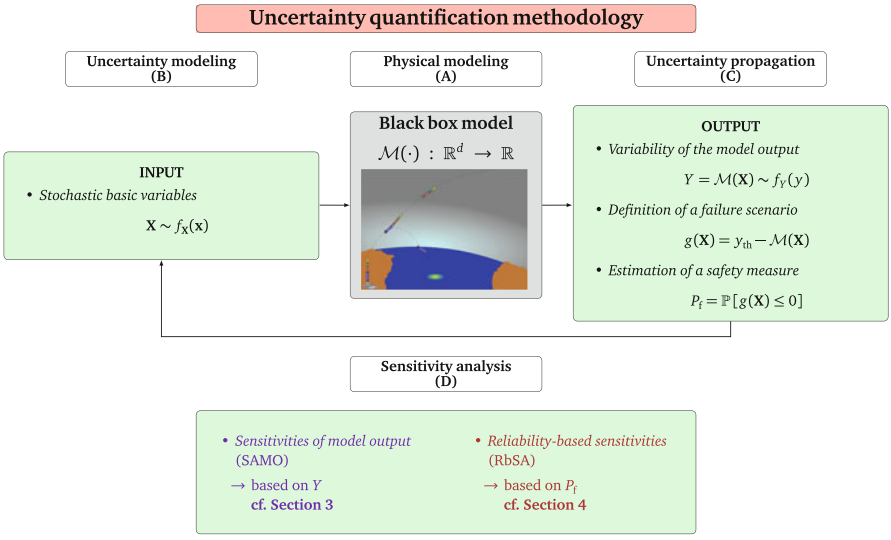


Figure 2 Uncertainty quantification methodology applied to the launcher stage fallout case (adapted from [43])

Table 1 Input probabilistic model

Variable X_i [a]	Distribution	Mean μ_{X_i}	Std σ_{X_i}
$X_1 = \Delta a$ (m)	Normal	0	1650
$X_2 = \Delta v$ (m s^{-1})	Normal	0	3.7
$X_3 = \Delta\gamma$ (rad)	Normal	0	0.001
$X_4 = \Delta\psi$ (rad)	Normal	0	0.0018
$X_5 = \Delta m$ (kg)	Normal	0	70
$X_6 = \Delta C_d$ (1)	Normal	0	0.1

[a]The input variables are independent

– the *backward* UQ which mainly corresponds to SAMO or RbSA (cf. block D), but also to inverse problems such as model calibration.

This UQ methodology can be seen as an iterative loop procedure which plays a role, either in the preliminary design process or in the certification procedure (e.g., regarding safety requirements) of complex systems such as aerospace ones.

In the context of the launch vehicle fallout case, the input variables are known to be affected by uncertainties (for instance, by natural variability or due to lack-of-knowledge). Thus, applying UQ methodology leads to consider a probabilistic model for the input vector $\mathbf{X}$, i.e., by assuming the existence of a joint PDF $f_{\mathbf{X}} : \mathcal{D}_{\mathbf{X}} \subseteq \mathbb{R}^d \to \mathbb{R}_+$. Since the input variables are assumed to be independent, this joint PDF corresponds to the product of the marginal PDFs f_{X_i} of the input variables $X_i,\ i \in \{1, \ldots, d\}$. The input probabilistic model for the launch vehicle fallout case is given in Table 1. For the sake of illustration, the numerical values used in this example are hypothetic.

Propagating the uncertainties from the input to the output of the black-box computer code (going from block B to block C in Figure 2) can be achieved by various methods (e.g., Monte Carlo simulations or any other advanced techniques such as those reviewed in [31, 42]). Thus, the model output is no more a single scalar value, but becomes a random variable characterized by its own PDF f_Y. An estimation of this PDF f_Y is given in Figure 3.

In the next section, it is proposed to apply a recent numerical scheme to estimate density-based SAMO measure that determines the components of $\mathbf{X}$ that have the most significant influence on the distribution of Y.

3 Global Sensitivity Analysis of the Model Output with Moment Independent Sensitivity Measures

SAMO presents two main objectives: on the one hand, to identify the most influential inputs that one may then seek to know with the greatest possible accuracy to reduce the output variability, and on the other hand, to determine non-influential inputs, which then makes it possible to decrease the model complexity. There are essentially two families of SAMO techniques: local sensitivity analysis and global sensitivity analysis. The local approaches correspond to the assessment

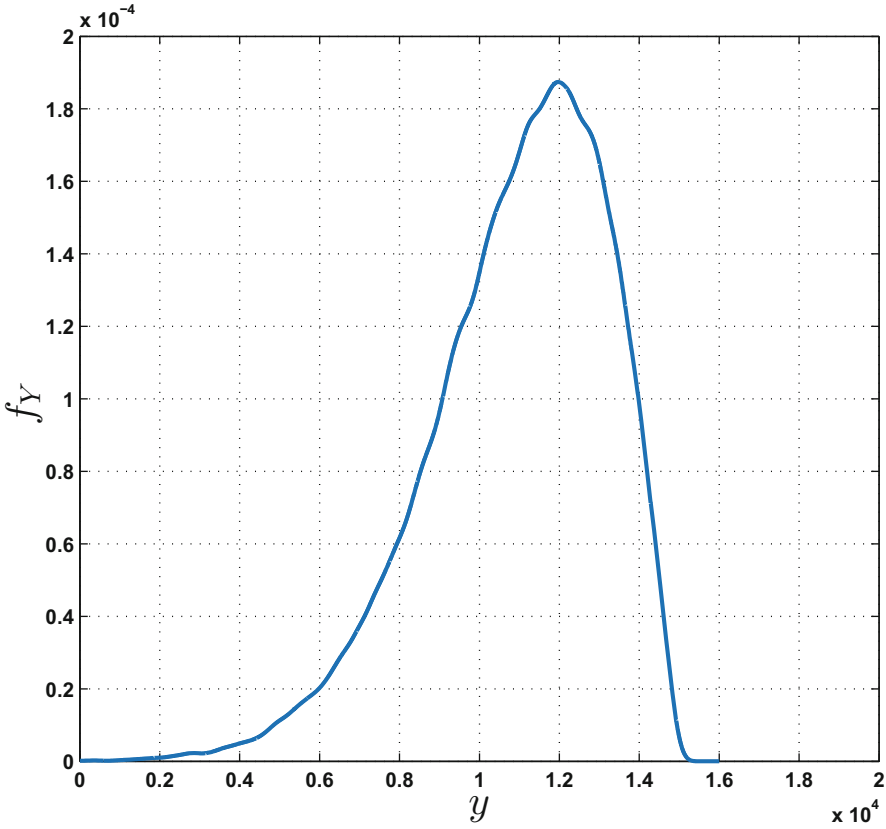


Figure 3 Probability density function estimation of the model output Y

of the local impact of an input on the model output by concentrating on the sensitivity in the vicinity of a set of nominal values. It may be defined as the partial derivatives of the model output. In contrast, global sensitivity analysis methods consider the whole variation range of the inputs: there are various techniques such as screening methods, graphical and smoothing tools, variance-based and moment independent methods. Variance-based importance measures [34, 40] are one of the most widely used importance measures. They are based on Sobol's indices, which express the share of variance of the output that is due to a given input or input combination. However, these methods focus on the second-order moment of the output distribution, which is not always sufficient to represent the entire variability of the distribution, as illustrated in [3]. To overcome this drawback, several alternatives are available, see [25] and the associated references for a review of these methods. In particular, Borgonovo [4] proposed distribution-based sensitivity indices that are currently gaining an increasing attention [5].

3.1 The δ-Sensitivity Measures

In order to define the moment independent sensitivity measures (also known as δ-sensitivity measures) initially proposed by Borgonovo [4], it is assumed throughout this section that for every $i \in \{1, \ldots, d\}$, the pair (X_i, Y) admits a probability density function (PDF) $f_{X_i,Y}$. This implies in particular that the random variables X_i, Y, and Y conditioned on $X_i = x_i$ for any $x_i \in \mathbb{R}$ also admit a PDF denoted respectively by f_{X_i}, f_Y, and $f_Y^{X_i=x_i}$ in the following.

The moment independent sensitivity analysis method is a global, quantitative, and model free SAMO method, which focuses on finding the inputs that, when held fixed, lead to the most significant modification of the output distribution. This difference between the conditional and unconditional model output densities $f_Y^{X_i=x_i}$ and f_Y is quantified by the *shift* $s(x_i)$ defined as their L_1 distance, which measures the area enclosed between their representative curve (see Figure 4):

$$s(x_i) \overset{\text{def}}{=} \left\| f_Y - f_Y^{X_i=x_i} \right\|_{L^1(\mathbb{R})} = \int \left| f_Y(y) - f_Y^{X_i=x_i}(y) \right| \mathrm{d}y \,. \tag{1}$$

So as to consider the whole range of values the random variable X_i can take into account, the sensitivity of the output Y with respect to the input X_i is defined by the renormalized expectation of the shift over X_i, i.e., the δ-sensitivity measure is given by:

$$\delta_i = \frac{1}{2}\mathbb{E}\left[s(X_i)\right] \,. \tag{2}$$

Owing to its appealing advantages, this importance measure has attracted more and more attention of practitioners recently. Firstly, it is monotonic transformation invariant, i.e., δ_i equals to the δ-sensitivity measure of the model $\tilde{Y} := \varphi \circ \mathcal{M}(\mathbf{X})$ for any C^1 diffeomorphism φ, which can be beneficial in practice. Moreover, this SAMO technique makes no assumption on the model, in particular the function $\mathcal{M}$ may be nonlinear and the input variables may be correlated. Eventually, this approach does not focus on a particular moment as the variance-based SAMO methods, which consider only the second-order moment, which is not always sufficient to represent the entire variability of the output distribution.

Finally, one can mention that Equation (2) can be generalized to a strict group of inputs $I \subset \{1, \ldots, d\}$ by:

$$\delta_I \overset{\text{def}}{=} \frac{1}{2}\mathbb{E}\left[s(\mathbf{X}_I)\right] \text{ with } \mathrm{s}(\mathbf{x}_\mathrm{I}) \overset{\text{def}}{=} \left\| \mathrm{f_Y} - \mathrm{f_Y^{\mathbf{X}_I=\mathbf{x}_I}} \right\|_{\mathrm{L}^1(\mathbb{R})} = \int \left| \mathrm{f_Y(y)} - \mathrm{f_Y^{\mathbf{X}_I=\mathbf{x}_I}(y)} \right| \mathrm{dy} \,, \tag{3}$$

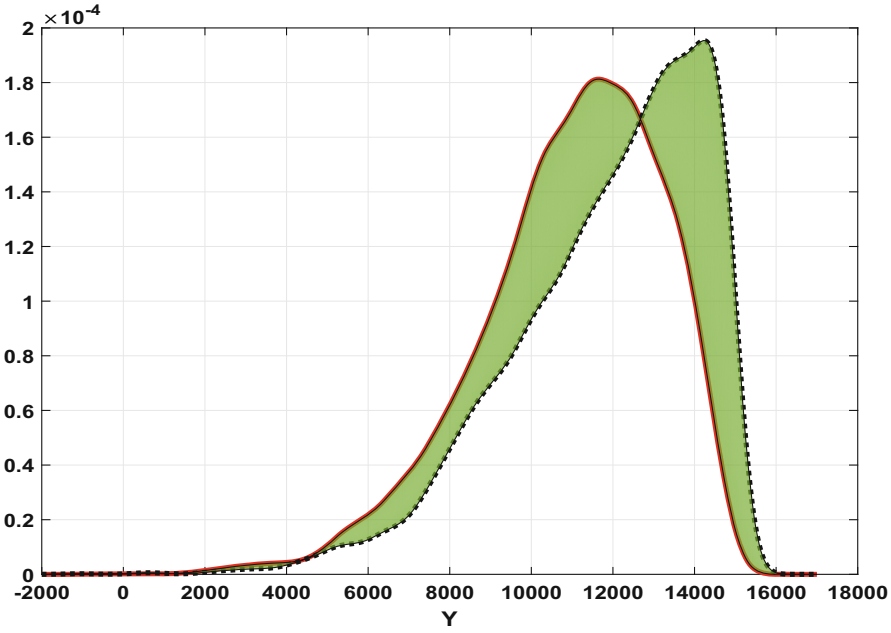


Figure 4 Representation of a kernel estimate of the output density f_Y (full line curve), a kernel estimate of the output density $f_Y^{X_4=-0.027}$ conditioned on X_4 fixed at its mean value -0.027 (dotted curve) and the associated shift (colored area)

where $\mathbf{X}_I \stackrel{\text{def}}{=} (X_i, i \in I)$. Throughout this section, the study is restricted to the case of the first-order indices δ_i, but all the results can be generalized to the higher order indices.

3.2 Estimation Scheme of δ_i

By its properties, δ_i index is attracting increasing attention and research has mostly focused on the delicate question of its estimation. Indeed, estimating the measure δ_i while minimizing the number of calls to the model response is a challenging task because of the unknown conditional and unconditional model output densities $f_Y^{X_i=x_i}$ and f_Y that intervene in a convoluted way (i.e., through an L^1-norm) in their definition (see Equations (1) and (2)).

The measure δ_i can be re-interpreted as an L_1-difference between the joint distribution $f_{X_i,Y}$ and the density of the random variables X_i and Y if they were independent. Indeed, from Equations (1) and (2) it follows immediately that:

$$\begin{aligned}\delta_i &= \frac{1}{2}\int f_{X_i}(x)\left(\int \left|f_Y(y) - f_Y^{X_i=x}(y)\right| dy\right) \mathrm{d}x \\ &= \frac{1}{2}\int f_{X_i}(x)\left|f_Y(y) - \frac{f_{X_i,Y}(x,y)}{f_{X_i}(x)}\right| \mathrm{d}x\mathrm{d}y \\ &= \frac{1}{2}\left\| f_{X_i} f_Y - f_{X_i,Y}\right\|_{L_1(\mathbb{R}^2)} .\end{aligned}$$

This interpretation opens the way for various estimation procedures of δ_i. Here, a new estimation scheme based on importance sampling procedure is exposed [16].

Step IS1. Generate $(\mathbf{X}^1, \ldots, \mathbf{X}^N)$ independent and identically distributed (i.i.d.) copies of $\mathbf{X}$, with common distribution $\mathbf{X}$, and then obtain N observations of the model by $Y^k = \mathcal{M}(\mathbf{X}^k)$ for $k = 1, \ldots, N$.

Step IS2. Use the sample $(\mathbf{X}^k, Y^k)$ to estimate the PDFs f_Y and $f_{X_i,Y}$ by kernel density estimation (KDE):

$$\hat{f}_Y(y) \overset{\text{def}}{=} \frac{1}{Nh}\sum_{k=1}^{N} K\left(\frac{y - Y^k}{h}\right), \ y \in \mathbb{R}, \tag{4}$$

and

$$\hat{f}_{X_i,Y}(x,y) \overset{\text{def}}{=} \frac{1}{Nh_1h_2}\sum_{k=1}^{N} K\left(\frac{x - X_i^k}{h_1}\right) K\left(\frac{y - Y^k}{h_2}\right), \ (x,y) \in \mathbb{R}^2 , \tag{5}$$

where K is the Gaussian kernel $K(t) = \frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}t^2)$ and where the bandwidths h, h_1, and h_2 are estimated with the diffusion-based method proposed in [6]. This method chooses the bandwidth parameters optimally without ever using or assuming a parametric model for the data or any "rule of thumb."

Step IS3. Let η be any sampling distribution on $\mathbb{R}^2$ which is allowed to depend on the sample $(\mathbf{X}^k, Y^k)$. Let $(\mathbf{V}^1, \ldots, \mathbf{V}^{N'})$ be N' i.i.d. random variables drawn according to η with $\mathbf{V}^k = (V_1^k, V_2^k) \in \mathbb{R}^2$. Get the estimator $\hat{\delta}_i^{IS,\eta}$ of δ_i defined by:

$$\hat{\delta}_i^{IS,\eta} \overset{\text{def}}{=} \frac{1}{2N'}\sum_{k=1}^{N'} \frac{\left|\hat{f}_Y(V_2^k) f_{X_i}(V_1^k) - \hat{f}_{X_i,Y}(\mathbf{V}^k)\right|}{\eta(\mathbf{V}^k)} , \tag{6}$$

i.e., the *importance sampling estimator* of $\frac{1}{2}\left\| f_{X_i}\hat{f}_Y - \hat{f}_{X_i,Y}\right\|_{L_1(\mathbb{R}^2)}$.

This method combines a single Monte Carlo loop and a KDE procedure. Furthermore, it requires only N calls to the black-box function $\mathcal{M}$ for the estimation of all the δ-sensitivity measures.

3.3 Choice of the Sampling Distribution

Since the variables $\mathbf{V}^k$ are i.i.d given the variables $(\mathbf{X}^k, \mathbf{Y}^k)$, the law of total variance gives the following variance decomposition:

$$\mathbb{V}\mathrm{ar}\left(\hat{\delta}_i^{IS,\eta}\right) = \frac{1}{4}\mathbb{V}\mathrm{ar}\left(\left\| f_{X_i}\hat{f}_Y - \hat{f}_{X_i,Y}\right\|_{L_1(\mathbb{R}^2)}\right) + \frac{1}{4N'}\mathbb{E}\left[\mathbb{V}\mathrm{ar}\left(\hat{h}(\mathbf{V}) \mid (\mathbf{X}^k, Y^k)\right)\right], \tag{7}$$

where

$$\hat{h}(x,y) = \mathbb{1}_{\eta(x,y)>0}\frac{\left|\hat{f}_Y(y)f_{X_i}(x) - \hat{f}_{X_i,Y}(x,y)\right|}{\eta(x,y)}.$$

This decomposition clearly highlights the two errors made by the estimator $\hat{\delta}_i^{IS,\eta}$: the term $\mathbb{V}\mathrm{ar}(\hat{\Delta})$ corresponds to the error induced by the KDE procedure of Step **IS2** and the second term to the error induced by the importance sampling Step **IS3**.

According to [17, Theorem 1] and Lebesgue's dominated convergence theorem, the first term tends to 0 when N tends to $+\infty$ upon standard assumptions on the bandwidths h, h_1, and h_2. As far as the second one is concerned, the variance term:

$$\mathbb{V}\mathrm{ar}\left(\hat{h}(\mathbf{V}) \mid (\mathbf{X}^k, Y^k)\right) = \int \frac{\left|\hat{f}_Y f_{X_i} - \hat{f}_{X_i,Y}\right|^2}{\eta} - \left(\int \left|\hat{f}_Y f_{X_i} - \hat{f}_{X_i,Y}\right|\right)^2, \tag{8}$$

may be infinite if the distribution η is not well chosen. Nevertheless in practice, assuming that η is nearly proportional to the numerator $|\hat{f}_Y f_{X_i} - \hat{f}_{X_i,Y}|$, this term can be made as small as desired because of the factor $\frac{1}{N'}$ without further calls to the possibly expensive black-box function $\mathcal{M}$. In addition, this term equals to zero when the sampling distribution η is given by the function:

$$\eta_{opt} = \frac{|f_{X_i}\hat{f}_Y - \hat{f}_{X_i,Y}|}{\|f_{X_i}\hat{f}_Y - \hat{f}_{X_i,Y}\|_{L_1(\mathbb{R}^2)}}, \tag{9}$$

called *optimal sampling distribution*. Unfortunately η_{opt} cannot be used directly in practice because of the unknown normalization constant, but it can be approximated by some $\hat{\eta}_{opt}$ using the nonparametric importance sampling procedure described in [45]. Assuming that **Steps IS1** and **IS2** have been performed and that we have the KDE $\hat{f}_Y$ and $\hat{f}_{X_i,Y}$ at our disposal, $\hat{\eta}_{opt}$ is derived by the following implementation steps:

- Generate a sample $\left\{(\tilde{X}_i^1, \tilde{Y}^1), \ldots, (\tilde{X}_i^{N''}, \tilde{Y}^{N''})\right\}$ according to an initial distribution η_0. It has to be noticed that no additional calls to the model output are needed.
- Compute the weights:

$$w(\tilde{X}_i^k, \tilde{Y}^k) = \frac{\left| f_{X_i}(\tilde{X}_i^k)\hat{f}_Y(\tilde{Y}^k) - \hat{f}_{X_i,Y}(\tilde{X}_i^k, \tilde{Y}^k)\right|}{\eta_0(\tilde{X}_i^k, \tilde{Y}^k)},\ k = 1, \ldots, N''.$$

- Estimate η_{opt} by the weighted kernel estimator:

$$\hat{\eta}_{opt}(x, y) \stackrel{\text{def}}{=} \frac{1}{N''\tilde{h}_1\tilde{h}_2\tilde{w}} \sum_{k=1}^{N''} w(\tilde{X}_i^k, \tilde{Y}^k) K\left(\frac{x - \tilde{X}_i^k}{\tilde{h}_1}\right) K\left(\frac{y - \tilde{Y}^k}{\tilde{h}_2}\right), \quad (10)$$

where $\tilde{w} = \frac{1}{N''} \sum_{k=1}^{N''} w(\tilde{X}_i^k, \tilde{Y}^k)$. Some results dealing with the convergence of the estimator $\hat{\eta}_{opt}$ are established in [45].

In the next section, the importance sampling estimator $\hat{\delta}_i^{IS,\eta_{opt}}$, denoted by $\hat{\delta}_i^{Opt}$, is used to analyze the sensitivity of the launch vehicle stage fallout model.

3.4 Application to the Launch Vehicle Stage Fallout Model

In this section, the method described in Sects. 3.2 and 3.3 is employed for the launch vehicle stage fallout model. An indicator of the efficiency of the estimator $\hat{\delta}_i^{Opt}$ of the importance measure δ_i is the coefficient of variation:

$$\text{cv}(\hat{\delta}_i^{Opt}) = \frac{\sqrt{\mathbb{V}\text{ar}(\hat{\delta}_i^{Opt})}}{\mathbb{E}\left(\hat{\delta}_i^{Opt}\right)}.$$

The mean and the standard deviation of $\hat{\delta}_i^{Opt}$ are approximated using a Monte Carlo procedure. Considering m estimates $(\hat{\delta}_i^1, \ldots, \hat{\delta}_i^m)$, the respective unbiased estimators of the mean and the standard deviation are computed such that:

$$\bar{\delta}_i \stackrel{\text{def}}{=} \frac{1}{m} \sum_{k=1}^{m} \hat{\delta}_i^k \text{ and } \bar{\sigma}_i \stackrel{\text{def}}{=} \sqrt{\frac{1}{m-1} \sum_{k=1}^{m} (\hat{\delta}_i^k - \bar{\delta}_i)^2},$$

Table 2 Estimates of the δ-sensitivity measures of the launch vehicle stage fallout test-case

	Input					
	X_1	X_2	X_3	X_4	X_5	X_6
Mean $\bar{\delta}_i$	0.0362	0.1556	0.0746	0.2976	0.0355	0.0490
$\mathrm{cv}(\hat{\delta}_i^{Opt})$	0.1042	0.0381	0.0699	0.0310	0.1012	0.0863

which provides the following estimator of the coefficient of variation:

$$\mathrm{cv}(\hat{\delta}_i^{Opt}) \approx \frac{\bar{\delta}_i}{\bar{\sigma}_i} .$$

All the six δ-sensitivity indices $(\delta_i)_{1\leq i\leq 6}$ displayed in Table 2 are estimated by applying $m = 100$ runs of the proposed estimation scheme. In order to ensure a good compromise between the minimization of the model calls and the efficiency of the kernel estimation step, the parameter N is fixed to 5×10^3. The results lead to the importance ranking $X_5 < X_1 < X_6 < X_3 < X_2 < X_4$ which highlights that the most influential inputs are X_2, velocity perturbation at separation, and X_4, azimuth angle perturbation at separation, which present δ-sensitivity indices greater than 15%.

These results may be compared with other popular sensitivity measures of the contribution of the input X_i: the Sobol's indices first introduced by Sobol [40] which aim to appreciate the contribution of the variable X_i to the variance of the output Y. The *first-order sensitivity index* [40] is stated as follows:

$$S_i \overset{\text{def}}{=} \frac{\mathbb{V}\mathrm{ar}\left(\mathbb{E}\left[Y|X_i\right]\right)}{\mathbb{V}\mathrm{ar}\left(Y\right)} ,$$

and the *total effect index* [21] is defined by:

$$S_{Ti} \overset{\text{def}}{=} \frac{\mathbb{E}\left[\mathbb{V}\mathrm{ar}\left(Y|\mathbf{X}_{\sim i}\right)\right]}{\mathbb{V}\mathrm{ar}\left(Y\right)} ,$$

where $\mathbb{E}\left[\mathbb{V}\mathrm{ar}\left(Y|\mathbf{X}_{\sim i}\right)\right]$ is the expected variance that is left when all inputs but X_i are known.

In order to get a fair comparison, $m = 100$ estimations of both Sobol's indices are used by adapting the code provided in [2] allowing to compute their respective mean and coefficient of variation, as for the δ-sensitivity measures. The results are reported in Table 3. The first-order indices lead to the ranking $X_5 < X_1 < X_6 < X_3 < X_4 < X_2$ which is quite similar to the previous one, except that the contribution to the output variance of the main effect of X_2 is greater than X_4. On that test-case, considering only the variance leads to underestimate the influence of X_4 relatively to X_2 as the δ-sensitivity indices describe the influence of a given input on the whole distribution of output and not only the variance. The total effect indices provide additional information showing the variables X_3 and X_6 play an important role when they are combined with other variables. It may be interesting to compare

Table 3 Estimates of the first order and total effect Sobol's indices of the launch vehicle stage fallout test-case

	Input					
	X_1	X_2	X_3	X_4	X_5	X_6
Mean S_i	0.0047	0.2642	0.0712	0.1915	0.0040	0.0223
cv(S_i)	3.4040	0.0590	0.2414	0.0783	4.1093	0.8090
Mean S_{Ti}	0.0151	0.6577	0.3826	0.2104	0.0169	0.2191
cv(S_{Ti})	0.0266	0.0211	0.0238	0.0270	0.0248	0.0231

these indices with the higher order Borgonovo indices δ_I defined in Equation (3), which implies to estimate the joint PDF $f_{X_I,Y}$. However, KDE is not robust in the case of high dimensional densities. Practitioners essentially consider Sobol's indices for SAMO for the sake of simplicity. Nevertheless, even if it is more complex to estimate δ-sensitivity indices than Sobol's indices, one can show that it is of interest to focus on the δ-sensitivity indices as it captures the complete distribution of an input and, in the same time, the computational cost required to estimate δ-sensitivity indices does not depend on the input dimension of the problem.

In this test-case, all the three sensitivity measures indicate that the inputs X_1 and X_5 have little influence in view of their low indices. Then, the uncertainty of the model output may be reduced by controlling the error of the velocity perturbation and azimuth angle perturbation at separation.

4 Reliability and Sensitivity Analyses Under Distribution Parameter Uncertainty

In the previous section, a new SAMO strategy based on the δ-sensitivity measures coupled with a nonparametric importance sampling procedure has been applied to the model output. However, the marginal PDFs f_{X_i}, for $i \in \{1, \ldots, d\}$, were supposed to be perfectly known, i.e., that the distribution parameters (e.g., means and standard deviations given in Table 1) are true deterministic values, which are able to catch the underlying physics. However, among the available information to construct a parametrized probabilistic model for the input basic variables, one often has only access to limited data, possibly unadapted literature-based recommendations and finally subjective expert opinions [19]. Thus, an imperfect state of knowledge [13, 18] may conduct to a misestimation of the failure probability and lead to dramatic consequences in terms of risk mitigation. Statistical uncertainty arises in the estimation procedure of the probability distribution parameters when one can only deal with insufficient measures or data. In some cases, it may also happen that neither data nor expert judgment is available, which imposes to the engineer yet to make a choice for the values of parameters. This problem is often encountered in the field of complex systems for which data acquisition is difficult.

4.1 Basics of Reliability Assessment

The system failure of the case presented in Sect. 2 can be considered if the fallout distance Y exceeds a given threshold distance $\mathrm{d}_{\mathrm{safe}}$. Such a failure criterion can be characterized by the limit-state function (LSF) $g : \mathbb{R}^{d=6} \to \mathbb{R}$ defined such that:

$$g(\mathbf{X}) = \mathrm{d}_{\mathrm{safe}} - \mathcal{M}(\mathbf{X}) = \mathrm{d}_{\mathrm{safe}} - Y. \tag{11}$$

The failure probability associated with this failure scenario is thus given by:

$$P_{\mathrm{f}} = \mathbb{P}\left[g(\mathbf{X}) \leq 0\right] = \mathbb{P}\left[Y > \mathrm{d}_{\mathrm{safe}}\right] = \int_{\mathcal{D}_{\mathbf{X}}} \mathbb{1}_{\mathcal{F}_{\mathbf{x}}}(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) \mathrm{d}\mathbf{x} = \mathbb{E}_{f_{\mathbf{X}}}\left[\mathbb{1}_{\mathcal{F}_{\mathbf{x}}}(\mathbf{X})\right] \tag{12}$$

where $\mathcal{F}_{\mathbf{x}} = \{\mathbf{x} \in \mathcal{D}_{\mathbf{X}} : g(\mathbf{x}) \leq 0\}$ is the so-called failure domain, $\mathrm{d}\mathbf{x} = \mathrm{d}x_1 \ldots \mathrm{d}x_d$, and $\mathbb{1}_{\mathcal{F}_{\mathbf{x}}}(\cdot)$, the indicator function of the failure domain defined by: $\mathbb{1}_{\mathcal{F}_{\mathbf{x}}}(\mathbf{x}) = 1$ if $\mathbf{x} \in \mathcal{F}_{\mathbf{x}}$ and $\mathbb{1}_{\mathcal{F}_{\mathbf{x}}}(\mathbf{x}) = 0$ otherwise.

Finally, estimating a failure probability implies to evaluate the integral defined in Equation (12). However, depending on various constraints (rareness of the failure event, high-dimensionality of the input space, nonlinearity of the model, expensive simulation cost of a single code run), this integral may be difficult and/or costly to evaluate. To do so, various techniques are available in the literature. Among them, one can distinguish between approximation-based techniques [27] and sampling-based techniques [37]. Approximation-based techniques rely on an approximation of the LSF, either by a Taylor series expansion (which leads to the well-known first/second-order reliability methods (FORM/SORM) [27]) or by a surrogate model whose aim is to replace the true but unknown LSF by a cheaper function (e.g., Gaussian processes, polynomial chaos expansions, support vector machines). Simulation-based techniques rely on the use of Monte Carlo samples to estimate the integral in Equation (12). Starting from the crude Monte Carlo (CMC) sampling, one can derive other more advanced sampling techniques in order to reduce the variance of estimation, and thus the number of calls to the computer code. Among these techniques, one can cite importance sampling (IS) and subset sampling (SS). For a review of these techniques, the interested reader may refer to [9, 31, 37].

Following mathematical and historical reasons [27], some of these methods (especially FORM/SORM) have been developed in the so-called *standard normal space* (denoted as "**U**-space") in which all random components of $\mathbf{X}$ become independent standard Gaussian variates gathered in the vector $\mathbf{U}$. Among the simulation methods, the use of such a standard normal space is not always required (e.g., CMC relies on simulations in the original *physical space*, denoted as "**X**-space"). The general idea is to construct a regular transformation $T : \mathcal{D}_{\mathbf{X}} \to \mathbb{R}^d$ allowing (in terms of probability distributions) to get:

$$\mathbf{U} = T(\mathbf{X}) \Leftrightarrow \mathbf{X} = T^{-1}(\mathbf{U}) \tag{13}$$

where $\mathbf{U} = (U_1, U_2, \ldots, U_d)^\top$ is a d-dimensional standard Gaussian vector of independent normal variates U_i with zero means and unit standard deviations. Then, one can define a new mapping for the LSF in the standard normal space considering $G : \mathbb{R}^d \to \mathbb{R}$ defined such that:

$$\mathbf{U} \mapsto G(\mathbf{U}) = \left(g \circ T^{-1}\right)(\mathbf{U}) \tag{14}$$

which allows to rewrite the failure probability:

$$P_\mathrm{f} = \mathbb{P}\left[G(\mathbf{U}) \leq 0\right] = \int_{\mathcal{F}_\mathbf{u}} \varphi_d(\mathbf{u})\mathrm{d}\mathbf{u} = \int_{\mathbb{R}^d} \mathbb{1}_{\mathcal{F}_\mathbf{u}}(\mathbf{u})\varphi_d(\mathbf{u})\mathrm{d}\mathbf{u} = \mathbb{E}_{\varphi_d}\left[\mathbb{1}_{\mathcal{F}_\mathbf{u}}(\mathbf{U})\right] \tag{15}$$

where $\mathcal{F}_\mathbf{u} = \{\mathbf{u} \in \mathbb{R}^d : G(\mathbf{u}) \leq 0\}$ stands for the failure domain in the standard normal space, $\mathrm{d}\mathbf{u} = \mathrm{d}u_1\mathrm{d}u_2 \ldots \mathrm{d}u_d$ and $\varphi_d : \mathbb{R}^d \to \mathbb{R}_+$ is the d-dimensional standard Gaussian PDF of $\mathbf{U}$. Usually, the transformation between the two spaces can be either the Nataf one [26] or the Rosenblatt one [36].

Combining approximation methods and the use of the standard normal space leads to the concept of *most probable failure point* (MPFP) which is the closest point of the failure domain to the origin of the standard normal space [27]. The MPFP is a cornerstone of the FORM/SORM methods. Even if this notion is a pure geometrical concept which loses its probabilistic meaning as the input dimensionality increases [23], one can still use it to visualize the shape of the failure domain. To do so, one can use FORM to get the MPFP and then study how the LSF behaves, in the **U**-space, by considering two-dimensional cross-cuts (i.e., one fixes all the inputs but two) [7, 20].

Figure 5 provides the cross-cuts in the (u_i, u_j)-plane for the launcher stage fallout test-case. The black cross is the origin and the black square represents the MPFP. The black line is the limit-state surface (denoted by LSS, formally defined by $\mathcal{F}_0 = \{\mathbf{u} : G(\mathbf{u}) = 0\}$), highlighting the separation between the safe domain (in green) and the failure domain (in orange). The analysis of these cross-cuts leads to point out two remarks:

- firstly, one can notice that, for some combinations, the LSS is highly nonlinear (e.g., for the pairs (u_1, u_4), (u_2, u_4), (u_3, u_4), (u_4, u_5)). This indicates that the methods relying on a linear assumption of the LSS (such as FORM) should be avoided for reliability assessment in this specific case;
- secondly, one can notice that the two cross-cuts $((u_2, u_3), (u_2, u_6))$ present possible multiple MPFPs. In particular, (u_2, u_3) would suggest the presence of a second MPFP of opposite coordinates.

Tracking multiple MPFPs can be achieved using a modification of the FORM algorithm proposed in [15]. Briefly, this method consists in repeating a FORM analysis with a modified LSF which triggers the search outside the area where a MPFP has been found. Here, by applying this method, one can find that a

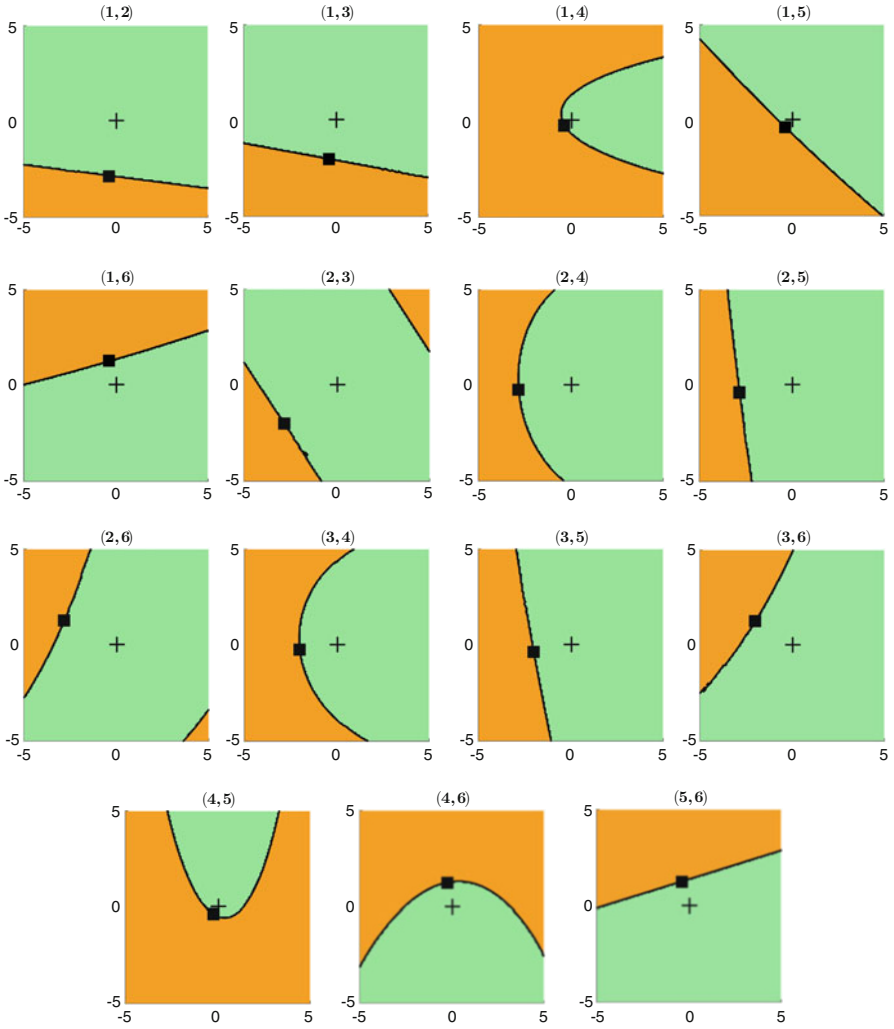


Figure 5 Illustration of cross-cuts in **U**-space (black cross ≡ origin of **U**-space, black square ≡ MPFP, black line ≡ limit-state surface, light gray (green) area ≡ safe domain, dark gray (orange) area ≡ failure domain)

second MFPF exists with the coordinates in the standard normal space $(u_2, u_3) = (3.084, 2.058)$ and $(u_2, u_6) = (3.084, -1.798)$. This result is just a qualitative view and does not help to correctly infer about the failure probability since the LSF is highly nonlinear. Thus, in the following, one will consider a nonparametric importance sampling scheme [45] for failure probability estimation in the launcher stage fallout test-case as it is able to cope with nonlinear LSS and the presence of multiple MPFPs.

4.2 Reliability Assessment Under Distribution Parameter Uncertainty

Dealing with distribution parameter uncertainty requires, first, to choose a way to represent this type of uncertainty, and second, to find a numerical strategy to propagate it and to take it into account in the failure probability estimation.

In this chapter, the Bayesian view is adopted [32, 33] in the sense that a parametric prior distribution is assumed for the uncertain parameters. Thus, the input probabilistic model, with the associated PDFs, is as follows:

$$\mathbf{X} \sim f_{\mathbf{X}|\boldsymbol{\Theta}}(\mathbf{x}|\boldsymbol{\theta}) : \mathcal{D}_{\mathbf{X}} \subseteq \mathbb{R}^d \to \mathbb{R}_+ \quad \text{(stochastic physical variables)} \tag{16a}$$

$$\boldsymbol{\Theta} \sim f_{\boldsymbol{\Theta}|\boldsymbol{\xi}}(\boldsymbol{\theta}|\boldsymbol{\xi}) : \mathcal{D}_{\boldsymbol{\Theta}} \subseteq \mathbb{R}^k \to \mathbb{R}_+ \quad \text{(stochastic distribution parameters)} \tag{16b}$$

$$\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_q)^\top \in \mathcal{D}_{\boldsymbol{\xi}} \subseteq \mathbb{R}^q \quad \text{(deterministic hyper-parameters).} \tag{16c}$$

In this hierarchical model, the vector of deterministic hyper-parameters $\boldsymbol{\xi}$ represents the available prior information (e.g., arising from sparse data or from expert judgment).

From the UQ methodology point of view, one can reconsider the initial methodology by adding some components in the different UQ steps such as illustrated in Figure 6. It results in the consideration of a *bi-level* input uncertainty which has to be propagated through the code. Modifying the input probabilistic model leads to reconsider the way one propagates and takes into account the bi-level uncertainty.

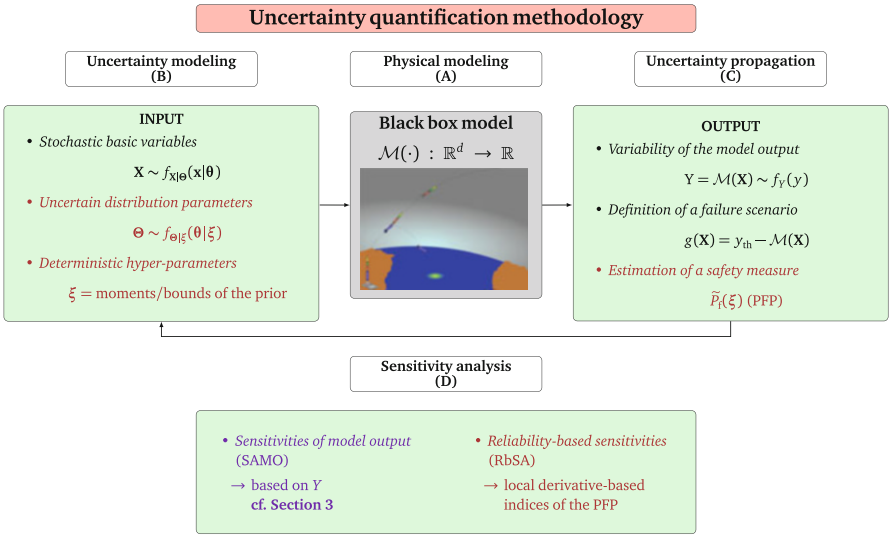


Figure 6 Uncertainty quantification methodology under bi-level uncertainty

Consequently, one can reconsider the failure probability in Equation (12) as a conditional failure probability since it depends on the realization $\boldsymbol{\theta}$ of the vector of distribution parameters. Then, it comes

$$P_{\mathrm{f}}(\boldsymbol{\theta}) = \mathbb{P}\left[g(\mathbf{X}) \leq 0 \mid \boldsymbol{\Theta} = \boldsymbol{\theta}\right] = \int_{\mathcal{D}_{\mathbf{X}}} \mathbb{1}_{\mathcal{F}_{\mathbf{x}}}(\mathbf{x}) f_{\mathbf{X}|\boldsymbol{\Theta}}(\mathbf{x}|\boldsymbol{\theta}) \mathrm{d}\mathbf{x}$$
$$= \mathbb{E}_{f_{\mathbf{X}|\boldsymbol{\Theta}}}\left[\mathbb{1}_{\mathcal{F}_{\mathbf{x}}}(\mathbf{X}) \mid \boldsymbol{\Theta} = \boldsymbol{\theta}\right]. \tag{17}$$

Then, one can propagate the second uncertainty level by looking at the mean estimator of all the failure probabilities regarding the variability of the distribution parameters [14], namely the "*predictive failure probability*" (PFP), defined such that:

$$\widetilde{P}_{\mathrm{f}}(\boldsymbol{\xi}) \overset{\text{def}}{=} \mathbb{E}_{f_{\boldsymbol{\Theta}|\boldsymbol{\xi}}}\left[P_{\mathrm{f}}(\boldsymbol{\Theta})\right] = \mathbb{E}_{f_{\boldsymbol{\Theta}|\boldsymbol{\xi}}}\left[\mathbb{E}_{f_{\mathbf{X}|\boldsymbol{\Theta}}}\left[\mathbb{1}_{\mathcal{F}_{\mathbf{x}}}(\mathbf{X}) \mid \boldsymbol{\Theta}\right]\right] = \int_{\mathcal{D}_{\boldsymbol{\Theta}}} P_{\mathrm{f}}(\boldsymbol{\theta}) f_{\boldsymbol{\Theta}|\boldsymbol{\xi}}(\boldsymbol{\theta}|\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\theta}. \tag{18}$$

This quantity can be estimated by two different approaches: a double-loop approach over both integration domains (called *nested reliability approach*, NRA) [29] or a single-loop one (called *augmented reliability approach*, ARA).

In this second strategy, one considers an "augmented" random vector $\mathbf{Z} \overset{\text{def}}{=} (\mathbf{X}, \boldsymbol{\Theta})^{\top}$ defined on $\mathcal{D}_{\mathbf{Z}} = \mathcal{D}_{\mathbf{X}} \times \mathcal{D}_{\boldsymbol{\Theta}}$ (where $\times$ is the Cartesian product) with joint pdf $f_{\mathbf{Z}|\boldsymbol{\xi}}(\mathbf{z}|\boldsymbol{\xi}) \overset{\text{def}}{=} f_{\mathbf{X}|\boldsymbol{\Theta}}(\mathbf{x}|\boldsymbol{\theta}) f_{\boldsymbol{\Theta}|\boldsymbol{\xi}}(\boldsymbol{\theta}|\boldsymbol{\xi})$ such that the expression in Equation (18) can be rewritten as follows:

$$\widetilde{P}_{\mathrm{f}}(\boldsymbol{\xi}) = \int_{\mathcal{D}_{\boldsymbol{\Theta}}} \int_{\mathcal{D}_{\mathbf{X}}} \mathbb{1}_{\mathcal{F}_{\mathbf{x}}}(\mathbf{x}) f_{\mathbf{X}|\boldsymbol{\Theta}}(\mathbf{x}|\boldsymbol{\theta}) f_{\boldsymbol{\Theta}|\boldsymbol{\xi}}(\boldsymbol{\theta}|\boldsymbol{\xi}) \mathrm{d}\mathbf{x}\mathrm{d}\boldsymbol{\theta}$$
$$= \int_{\mathcal{D}_{\mathbf{Z}}} \mathbb{1}_{\mathcal{F}_{\mathbf{z}}}(\mathbf{z}) f_{\mathbf{Z}|\boldsymbol{\xi}}(\mathbf{z}|\boldsymbol{\xi}) \mathrm{d}\mathbf{z} = \mathbb{E}_{f_{\mathbf{Z}|\boldsymbol{\xi}}}\left[\mathbb{1}_{\mathcal{F}_{\mathbf{z}}}(\mathbf{Z})|\boldsymbol{\xi}\right] \tag{19}$$

where $\mathcal{F}_{\mathbf{z}} = \{\mathbf{z} \in \mathcal{D}_{\mathbf{Z}} : g(\mathbf{z}) \leq 0\}$. As discussed in [10], ARA offers several possibilities compared to NRA in terms of simulation cost reduction, estimation accuracy, and robustness with respect to (w.r.t.) several numerical challenges concerning real aerospace test-cases.

Estimating a rare event probability with CMC can be cumbersome and can even become untraceable for costly-to-evaluate computer codes. As previously mentioned, IS is now a well-known variance-reduction technique [38] which enables to reduce the simulation cost. The idea is to use a so-called auxiliary density $\eta(\cdot)$ to generate samples such that more samples lead to the failure event $\{g(\mathbf{z}) \leq 0\}$. To introduce it, one can start from the observation that the following equality holds:

$$\widetilde{P}_{\mathrm{f}}(\boldsymbol{\xi}) = \int_{\mathcal{D}_{\mathbf{Z}}} \mathbb{1}_{\mathcal{F}_{\mathbf{z}}}(\mathbf{z}) f_{\mathbf{Z}|\boldsymbol{\xi}}(\mathbf{z}|\boldsymbol{\xi}) \mathrm{d}\mathbf{z} = \int_{\mathcal{D}_{\mathbf{Z}}} \mathbb{1}_{\mathcal{F}_{\mathbf{z}}}(\mathbf{z}) \frac{f_{\mathbf{Z}|\boldsymbol{\xi}}(\mathbf{z}|\boldsymbol{\xi})}{\eta(\mathbf{z})} \eta(\mathbf{z}) \mathrm{d}\mathbf{z}$$
$$= \int_{\mathcal{D}_{\mathbf{Z}}} \mathbb{1}_{\mathcal{F}_{\mathbf{z}}}(\mathbf{z}) w(\mathbf{z}) \eta(\mathbf{z}) \mathrm{d}\mathbf{z} \tag{20}$$

where $w(\mathbf{z}) \overset{\text{def}}{=} \frac{f_{\mathbf{Z}|\boldsymbol{\xi}}(\mathbf{z}|\boldsymbol{\xi})}{\eta(\mathbf{z})}$ is called the *likelihood ratio* [38]. This weight is introduced in the probability estimator to take into account the change in the PDF to generate samples. Thus, considering a sample $\{\mathbf{Z}^{(i)}\}_{i=1}^{N}$ of N i.i.d. copies drawn according to $\eta(\mathbf{z})$, the IS estimator for the PFP in the ARA framework is

$$\widehat{\widetilde{P}}_{\mathrm{f}} \underset{\mathrm{IS}}{=} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\mathcal{F}_{\mathbf{z}}}(\mathbf{Z}^{(i)})\, w(\mathbf{Z}^{(i)}). \tag{21}$$

The estimator $\widehat{\widetilde{P}}_{\mathrm{f}}$ of $\widetilde{P}_{\mathrm{f}}$ is unbiased and its variance $\mathbb{V}\mathrm{ar}\left(\widehat{\widetilde{P}}_{\mathrm{f}}\right)$ reduces to zero as the density $\eta(\cdot)$ equals the optimal auxiliary density $\eta^*(\cdot)$ which is given by:

$$\eta^*(\mathbf{z}) = \frac{\mathbb{1}_{\mathcal{F}_{\mathbf{z}}}(\mathbf{z})\, f_{\mathbf{Z}|\boldsymbol{\xi}}(\mathbf{z}|\boldsymbol{\xi})}{\widetilde{P}_{\mathrm{f}}}. \tag{22}$$

Since this quantity depends on the PFP one would like to estimate, this intricate problem can be solved by using *adaptive importance sampling* (AIS) techniques [44]. These techniques aim at using different adaptive strategies to sequentially approximate the optimal auxiliary density. In this chapter, a *nonparametric adaptive importance sampling* (NAIS) scheme (see [31] for any further detail about the NAIS method), adapted to the ARA framework, is used for the estimation of the PFP and the sensitivities of the PFP w.r.t. the a priori hyper-parameters, as explained in the following section.

4.3 Reliability-Based Sensitivity Analysis

Getting an estimate of the PFP is crucial for reliability assessment under the bi-level uncertainty. However, such an estimate depends on the a priori choice set in the prior distribution $f_{\boldsymbol{\Theta}|\boldsymbol{\xi}}(\boldsymbol{\theta}|\boldsymbol{\xi})$, i.e., in the choice of the type of prior distribution and the choice of the hyper-parameters $\boldsymbol{\xi}$. In this chapter, an estimator of the sensitivity of the PFP w.r.t. this a priori choice of hyper-parameters is proposed.

The gradient of the predictive failure probability $\widetilde{P}_{\mathrm{f}}$ w.r.t. the vector of the hyper-parameters $\boldsymbol{\xi}$ is defined as follows:

$$\boldsymbol{\nabla} \widetilde{P}_{\mathrm{f}}(\boldsymbol{\xi}) = \left(\frac{\partial \widetilde{P}_{\mathrm{f}}(\boldsymbol{\xi})}{\partial \xi_j},\ j = 1, \ldots, q \right)^{\top}. \tag{23}$$

In this chapter, the case where ξ_j is an hyper-parameter of a prior distribution with an unbounded support is treated. However, one could consider the case where ξ_j is an hyper-parameter of a prior distribution with a bounded or truncated support. This case is detailed in [12].

The partial derivative of the PFP w.r.t. the j-th component of $\boldsymbol{\xi}$ is given by:

$$S_j^{loc} \overset{\text{def}}{=} \frac{\partial \widetilde{P}_{\text{f}}(\boldsymbol{\xi})}{\partial \xi_j} = \frac{\partial}{\partial \xi_j}\left[\int_{\mathcal{D}_{\boldsymbol{\Theta}}} P_{\text{f}}(\boldsymbol{\theta})\, f_{\boldsymbol{\Theta}|\boldsymbol{\xi}}(\boldsymbol{\theta}|\boldsymbol{\xi})\mathrm{d}\boldsymbol{\theta}\right] = \int_{\mathcal{D}_{\boldsymbol{\Theta}}} P_{\text{f}}(\boldsymbol{\theta}) \frac{\partial f_{\boldsymbol{\Theta}|\boldsymbol{\xi}}(\boldsymbol{\theta}|\boldsymbol{\xi})}{\partial \xi_j}\mathrm{d}\boldsymbol{\theta}. \tag{24}$$

Using the so-called *importance sampling trick* [38] so as to get an expectation w.r.t. the same probability measure as the one used for the failure probability estimation, it comes

$$\frac{\partial \widetilde{P}_{\text{f}}(\boldsymbol{\xi})}{\partial \xi_j} = \int_{\mathcal{D}_{\boldsymbol{\Theta}}} P_{\text{f}}(\boldsymbol{\theta}) \frac{\partial \ln f_{\boldsymbol{\Theta}|\boldsymbol{\xi}}(\boldsymbol{\theta}|\boldsymbol{\xi})}{\partial \xi_j} f_{\boldsymbol{\Theta}|\boldsymbol{\xi}}(\boldsymbol{\theta}|\boldsymbol{\xi})\mathrm{d}\boldsymbol{\theta} \tag{25a}$$

$$= \int_{\mathcal{D}_{\boldsymbol{\Theta}}} \left(\int_{\mathcal{D}_{\mathbf{X}}} \mathbb{1}_{\mathcal{F}_{\mathbf{x}}}(\mathbf{x})\, \kappa_j(\boldsymbol{\theta}, \boldsymbol{\xi})\; f_{\mathbf{X}|\boldsymbol{\Theta}}(\mathbf{x}|\boldsymbol{\theta})\mathrm{d}\mathbf{x}\right) f_{\boldsymbol{\Theta}|\boldsymbol{\xi}}(\boldsymbol{\theta}|\boldsymbol{\xi})\mathrm{d}\boldsymbol{\theta} \tag{25b}$$

$$= \mathbb{E}_{f_{\mathbf{Z}|\boldsymbol{\xi}}}\left[\mathbb{1}_{\mathcal{F}_{\mathbf{z}}}(\mathbf{Z})\, \kappa_j(\boldsymbol{\Theta}, \boldsymbol{\xi})\right] \tag{25c}$$

where $\kappa_j(\boldsymbol{\theta}, \boldsymbol{\xi}) \overset{\text{def}}{=} \frac{\partial \ln f_{\boldsymbol{\Theta}|\boldsymbol{\xi}}(\boldsymbol{\theta}|\boldsymbol{\xi})}{\partial \xi_j}$ is called the "*score function*" [11]. One should notice that, to avoid any confusion, in the above equations and in the rest of the chapter, the vector $\boldsymbol{\Theta}$ is explicitly written instead of $\mathbf{Z} = (\mathbf{X}, \boldsymbol{\Theta})^\top$ since the dependence w.r.t. $\boldsymbol{\xi}$ is through $\boldsymbol{\Theta}$. Examples of score functions for a variety of distributions can be found in [30]. Then, considering N i.i.d. samples $\{\mathbf{Z}^{(i)}\}_{i=1}^N$, one can derive the following IS estimator:

$$\widehat{S_j^{loc}} \underset{\text{IS}}{=} \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\mathcal{F}_{\mathbf{z}}}(\mathbf{Z}^{(i)})\; w(\mathbf{Z}^{(i)})\; \kappa_j(\boldsymbol{\Theta}^{(i)}, \boldsymbol{\xi}). \tag{26}$$

As a remark, the gradient given in Equation (26) can be estimated as a simple post-treatment of the previous samples used in Equation (21) for the PFP estimation, with no additional computational effort.

4.4 Application to the Launch Vehicle Fallback Zone Estimation Code

In addition to the first level of uncertainty, one assumes that epistemic uncertainty is affecting two mean values, respectively, μ_{X_2} and μ_{X_3}, i.e., the mean values of the perturbations affecting the velocity at separation and the perturbations affecting the flight path angle at separation as shown in Table 4. These physical quantities can be difficult to measure and to control in real conditions. For the sake of illustration, the numerical values used in this example are hypothetic. In the numerical experiment, the threshold safety distance d_{safe} is chosen to be equal to 15 km.

Numerical results gathered in Table 5 are averaged over a hundred replications of the algorithm. Left column corresponds to the reference results obtained by

Table 4 Input probabilistic model under bi-level uncertainty

Variable[a]	Distribution	Parameter #1	Parameter #2
$X_1 = \Delta a$ (m)	Normal	$\mu_{X_1} = 0$	$\sigma_{X_1} = 1650$
$X_2 = \Delta v$ (m s^{-1})	Normal	μ_{X_2} *uncertain*[b]	$\sigma_{X_2} = 3.7$
$X_3 = \Delta\gamma$ (rad)	Normal	μ_{X_3} *uncertain*	$\sigma_{X_3} = 0.001$
$X_4 = \Delta\psi$ (rad)	Normal	$\mu_{X_4} = 0$	$\sigma_{X_4} = 0.0018$
$X_5 = \Delta m$ (kg)	Normal	$\mu_{X_5} = 0$	$\sigma_{X_5} = 70$
$X_6 = \Delta C_d$ (1)	Normal	$\mu_{X_6} = 0$	$\sigma_{X_6} = 0.1$
$\Theta_2 = \mu_{X_2}$ (m s^{-1})	Normal	$\xi_1 = \mu_{\mu_{X_2}} = 0$	$\xi_2 = \sigma_{\mu_{X_2}} = 3.7$
$\Theta_3 = \mu_{X_3}$ (rad)	Normal	$\xi_3 = \mu_{\mu_{X_3}} = 0$	$\xi_4 = \sigma_{\mu_{X_3}} = 0.001$

[a]The basic variables are independent. The distribution parameters are independent too
[b]For fixed values $\mu_{X_2} = 0$, $\mu_{X_3} = 0$ and a threshold distance $d_{safe} = 15$ km, $P_f = 1.36 \times 10^{-4}$

Table 5 Numerical results

	Ref.: ARA/CMC ($d_{safe} = 15$ km) ($N_{x,\theta} = 10^6$ samples)		ARA/NAIS ($d_{safe} = 15$ km) ($N_{x,\theta} = 10^4$ samples/step)		ARA/NAIS ($d_{safe} = 20$ km) ($N_{x,\theta} = 10^4$ samples/step)	
	Estimate	cv	Estimate	cv	Estimate	cv
$\widehat{\widetilde{P}_f}$	4.40×10^{-3}	(1.38%)	4.40×10^{-3}	(2.08%)	1.19×10^{-4}	(2.85%)
$\widehat{S_1^{loc}}$	-9.13×10^{-4}	(3.44%)	-9.12×10^{-4}	(5.90%)	-3.66×10^{-5}	(7.00%)
$\widehat{S_2^{loc}}$	2.95×10^{-3}	(2.32%)	2.95×10^{-3}	(3.22%)	1.41×10^{-4}	(3.62%)
$\widehat{S_3^{loc}}$	-2.31	(3.88%)	-2.30	(5.82%)	-9.18×10^{-2}	(7.67%)
$\widehat{S_4^{loc}}$	6.43	(2.18%)	6.41	(3.77%)	3.10×10^{-1}	(4.24%)
ν	–	–	13	–	207	–

ARA/CMC (see [11]). The values of the estimates, for both the PFP and the sensitivities, are provided with their values of coefficient of variation (cv). The simulation budgets $N_{x,\theta}$ in the augmented space are provided. However, for ARA/NAIS, only the results for $N_{x,\theta} = 10^4$ samples per step are given. The number of samples per step is a tuning parameter in the NAIS algorithm [31]. For the sake of comparison, the numerical efficiency ν is provided in the last row of the table. This efficiency indicates by how much one can divide the ARA/CMC simulation budget to allow an estimation of the PFP of the same accuracy (i.e., same coefficients of variation between ARA/CMC and ARA/NAIS).

From these results, one can see first that ARA/NAIS manages to estimate accurately the PFP. As a first remark, this PFP is slightly greater than the failure probability under single-level uncertainty (see Equation (12)) recalled below Table 4. This shows again that, in this case, the distribution parameter uncertainty makes the system less safe. In terms of sensitivities, the PFP seems to be more

sensitive to the hyper-parameters ξ_4 and ξ_3 which are respectively the standard deviation and the mean of $\Theta_3 = \mu_{X_3}$. Here, the lack of knowledge affecting the mean value of the flight path angle perturbation really plays a key role on the final predictive failure probability. This is a relevant information for refining the a priori probabilistic model for Θ_3 (especially in terms of variance reduction) and set up an investigation policy about the possible reduction of epistemic (statistical) uncertainty affecting Θ_3. Concerning the efficiency, for $d_{safe} = 15$ km, i.e., for a moderate rareness of the failure event, ARA/NAIS allows to divide the simulation budget by 13 for the same level of accuracy compared to ARA/CMC.

Finally, as the failure event becomes rare (i.e., for the case $d_{safe} = 20$ km, see the right column in Table 5), one can see that ARA/NAIS outperforms ARA/CMC by allowing to reduce the simulation budget by 207. Similar comparisons can be drawn to the previous case regarding the relative influence of the hyper-parameters. However, one can still notice that increasing the rareness of the failure event decreased, in proportion, the relative influence of ξ_4.

Figure 7 provides an illustration of the simulation of 10^3 fallback trajectories (impact points) of a first launcher stage into the ocean. The "safe zone" (i.e., the dark blue disk) is set to a given optimal center point (latitude and longitude coordinates estimated by the trajectory simulation code) and a radius of $d_{safe} = 15$ km. On three different plots, the behaviors of numerical strategies are represented: namely, NRA/CMC (nested crude Monte Carlo), ARA/CMC, and ARA/NAIS. For each case, the total number of impact points into the ocean is 10^3. However, on the left, one can notice the sequential trend of NRA/CMC, while in the middle, one can see the better covering of the ARA/CMC. However, in both cases, a lot of samples are useless regarding the failure domain. On the right, ARA/NAIS manages to efficiently draw samples in the regions of interest, i.e., impact points corresponding to the two most probable failure points.

As for Figure 8, one can see the possibilities ARA/NAIS offers in terms of simulation budget reduction compared to NRA/CMC (which is the most expensive

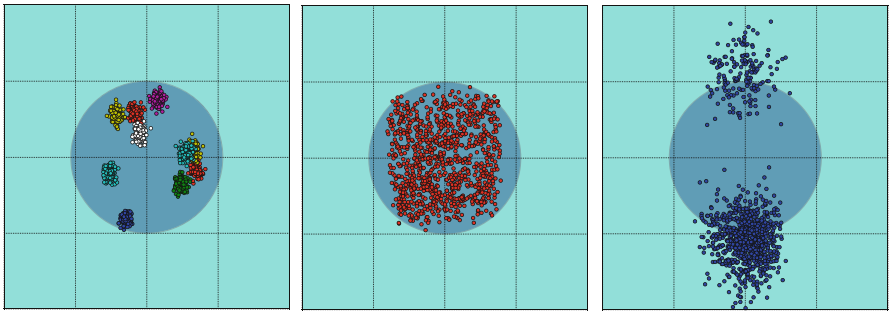


Figure 7 Illustration of the simulation of first launcher stage impact points into the ocean (10^3 samples for each method) and the safe zone (darker disk): NRA/CMC (left)—ARA/CMC (middle)—ARA/NAIS (right)

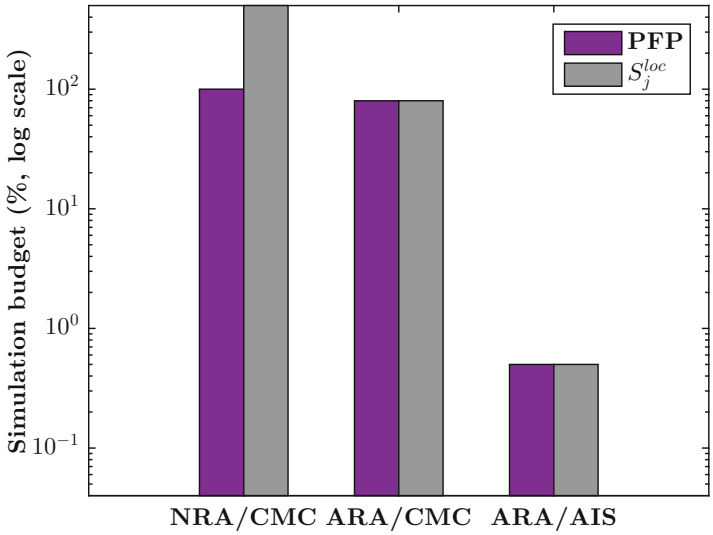


Figure 8 Evolution of the estimation budget of PFP and its sensitivities S_j^{loc} for each method (NRA/CMC is taken as the reference for the percentage, coupled with a finite difference scheme to estimate the sensitivities). The results are represented in log scale

approach since it relies on a double-Monte-Carlo-loop sampling strategy) and ARA/CMC.

5 Synthesis About Numerical Results for the Launch Vehicle Case

The numerical results described in Sect. 3 show that the inputs X_2 and X_4 are very influent in regard to their respective δ-sensitivity measures δ_i and first-order Sobol's indices S_i. Nevertheless, Borgonovo approach does not provide the same importance ranking than Sobol one since $\delta_2 < \delta_4$ but $S_4 < S_2$. Then, considering only the variance may lead to underestimate the influence of an input on the whole distribution of the output. In addition, one can note that the influence of X_3 in the sense of both Borgonovo and Sobol is slightly higher than X_1, X_5, and X_6. Then, investigate the combined contribution of the inputs may provide additional information. This is confirmed by the total effect indices which highlight the important role of X_3. Unfortunately, the higher order δ-sensitivity measures δ_I are very difficult to estimate with precision because of the estimation of an r-dimensional density with $r > 3$. To conclude, the δ-sensitivity measure indicates that the velocity perturbation at separation and mostly the azimuth angle perturbation at separation have an important impact on the output distribution. Furthermore, an additional investigation with total effect indices shows that the

flight path angle perturbation at separation has also a significant influence on the distance Y.

From a reliability assessment perspective, as studied in Sect. 4, one can see that studying the behavior of the LSS in the standard normal space corroborates the SAMO results, i.e., that X_2 and X_4 should influence the reliability. Moreover, when considering a bi-level input uncertainty, the RbSA results show that the PFP is very sensitive to the distribution hyper-parameters defining the prior of the mean of X_3 that corroborates the result obtained for the total Sobol's indices for this variable as mentioned previously. As a result, one should investigate more about the probabilistic model associated with X_3 since this variable plays a role both on the model output and on the PFP.

As a conclusion, SAMO and RbSA provide different levels of information about the sensitivity of different quantities of interest based on the model output. By combining them, one may find some common trends (or opposite trends) which can help the user, either to get a deeper understanding of the black-box computer code and underlying physics or to adopt an investigation policy so as to enhance the input probabilistic model.

6 Conclusion

In this book chapter, we considered a simplified launcher stage fallout model to analyze, without loss of generality, the efficiency of the proposed methods. Our objective was to determine the most influential factors on the fallout and on its failure probability. For that purpose, we first apply a new scheme of estimation of moment independent sensitivity measures (δ-sensitivity measures) that has a low computational cost. These indices take the entire fallout distribution probability into account unlike classical Sobol's indices that focus on the distribution variance. We noticed in this test-case that the influence of the input "propellant mass perturbation at separation" was underestimated by Sobol's indices while it is the most influential factors according to δ-sensitivity measures. In a second part, we assume that the launcher stage fallout model is affected by a bi-level uncertainty and propose a numerical estimation strategy to estimate the predictive failure probability and its sensitivities w.r.t. the hyper-parameters of the prior distribution. This estimation strategy, called ARA/NAIS, relies on the use of an augmented space (ARA) coupled to a nonparametric adaptive importance sampling (NAIS) scheme. Thus, this strategy allows to estimate, with a better efficiency than CMC, both the predictive failure probability and its sensitivities by just post-processing the samples used to estimate the predictive failure probability. This study shows the benefits of using an ARA/NAIS strategy when the failure event becomes very rare, especially for complex models.

Acknowledgements The first and second author contributed equally to this work. The first two authors are currently enrolled in a PhD program, respectively, funded by Université Toulouse III—Paul Sabatier and co-funded by ONERA—The French Aerospace Lab and SIGMA Clermont. Their financial supports are gratefully acknowledged. The authors would like to thank Dr. Loïc Brevault (Research scientist at ONERA—The French Aerospace Lab) for having provided the launch vehicle fallout zone estimation code.

References

1. Beer, M., Ferson, S., Kreinovich, V.: Imprecise probabilities in engineering structures. Mech. Syst. Signal Process. **37**, 4–29 (2013)
2. Bilal, N.: Implementation of Sobol's method of global sensitivity analysis to a compressor simulation model. In: Proc. of the 22nd International Compressor Engineering Conference, Purdue (2014)
3. Borgonovo, E.: Measuring uncertainty importance: investigation and comparison of alternative approaches. Risk Anal. **26**(5), 1349–1361 (2006)
4. Borgonovo, E.: A new uncertainty importance measure. Reliab. Eng. Syst. Saf. **92**(6), 771–784 (2007)
5. Borgonovo, E., Plischke, E.: Sensitivity analysis: a review of recent advances. Eur. J. Oper. Res. **248**(3), 869–887 (2016)
6. Botev, Z.I., Grotowski, J.F., Kroese, D.P.: Kernel density estimation via diffusion. Ann. Stat. **38**(5), 2916–2957 (2010)
7. Bourinet, J.M.: Rare-event probability estimation with adaptive support vector regression surrogates. Reliab. Eng. Syst. Saf. **150**, 210–221 (2016)
8. Bourinet, J.M.: FORM sensitivities to distribution parameters with the Nataf transformation. In: Gardoni, P. (ed.) Risk and Reliability Analysis: Theory and Applications. In Honor of Prof. Armen Der Kiureghian, Springer Series in Reliability Engineering, pp. 277–302. Springer International Publishing, Cham (2017)
9. Bucklew, J.A.: Introduction to Rare Event Simulation. Springer, New York (2004)
10. Chabridon, V., Balesdent, M., Bourinet, J.M., Morio, J., Gayton, N.: Evaluation of failure probability under parameter epistemic uncertainty: application to aerospace system reliability assessment. Aerosp. Sci. Technol. **69**, 526–537 (2017)
11. Chabridon, V., Balesdent, M., Bourinet, J.M., Morio, J., Gayton, N.: Reliability-based sensitivity analysis of aerospace systems under distribution parameter uncertainty using an augmented approach. In: Proc. of the 12th International Conference on Structural Safety and Reliability (ICOSSAR'17), Vienna (2017)
12. Chabridon, V., Balesdent, M., Bourinet, J.M., Morio, J., Gayton, N.: Reliability-based sensitivity estimators of rare event probability in the presence of distribution parameter uncertainty. Reliab. Eng. Syst. Saf. **178**, 164–178 (2018)
13. Der Kiureghian, A.: Measures of structural safety under imperfect states of knowledge. J. Struct. Eng. ASCE **115**(5), 1119–1140 (1989)
14. Der Kiureghian, A.: Analysis of structural reliability under parameter uncertainties. Probab. Eng. Mech. **23**(4), 351–358 (2008)
15. Der Kiureghian, A., Dakessian, T.: Multiple design points in first and second-order reliability. Struct. Saf. **20**, 37–49 (1998)
16. Derennes, P., Morio, J., Simatos, F.: A nonparametric importance sampling estimator for moment independent importance measures. Reliab. Eng. Syst. Saf. (2018). https://doi.org/10.1016/j.ress.2018.02.009

17. Devroye, L., Gyorfi, L.: Nonparametric Density Estimation: The L1 View. Wiley, New York (1985)
18. Ditlevsen, O.: Generalized second moment reliability index. J. Struct. Mech. **7**(4), 435–451 (1979)
19. Ditlevsen, O., Madsen, H.O.: Structural Reliability Methods, Internet ed. 2.3.7. Technical University of Denmark, Lyngby (2007)
20. Dubourg, V.: Adaptive surrogate models for reliability analysis and reliability-based design optimization. Ph.D. thesis, Université Blaise Pascal – Clermont II (2011)
21. Homma, T., Saltelli, A.: Importance measures in global sensitivity analysis of nonlinear models. Reliab. Eng. Syst. Saf. **52**(1), 1–17 (1996)
22. Hoogendoorn, R., Mooij, E., Geul, J.: Uncertainty propagation for statistical impact prediction of space debris. Adv. Space Res. **61**(1), 167–181 (2018)
23. Hurtado, J.E.: Structural Reliability: Statistical Learning Perspectives. Lecture Notes in Applied and Computational Mechanics. Springer, Berlin (2004)
24. Iooss, B., Lemaître, P.: A review on global sensitivity analysis methods. In: Dellino, G., Meloni, C. (eds.) Uncertainty Management in Simulation-Optimization of Complex Systems: Algorithms and Applications, chap. 5, pp. 101–122. Springer, Boston (2015)
25. Iooss, B., Lemaître, P.: A review on global sensitivity analysis methods. In: Uncertainty Management in Simulation-Optimization of Complex Systems, pp. 101–122. Springer, Berlin (2015)
26. Lebrun, R., Dutfoy, A.: Do Rosenblatt and Nataf isoprobabilistic transformations really differ? Probab. Eng. Mech. **24**, 577–584 (2009)
27. Lemaire, M.: Structural Reliability. ISTE Ltd, London; Wiley, Hoboken (2009)
28. Lemaître, P., Sergienko, E., Arnaud, A., Bousquet, N., Gamboa, F., Iooss, B.: Density modification-based reliability sensitivity analysis. J. Stat. Comput. Simul. **85**(6), 1200–1223 (2015)
29. Limbourg, P., De Rocquigny, E., Andrianov, G.: Accelerated uncertainty propagation in two-level probabilistic studies under monotony. Reliab. Eng. Syst. Saf. **95**, 998–1010 (2010)
30. Millwater, H.R.: Universal properties of kernel functions for probabilistic sensitivity analysis. Probab. Eng. Mech. **24**, 89–99 (2009)
31. Morio, J., Balesdent, M.: Estimation of Rare Event Probabilities in Complex Aerospace and Other Systems: A Practical Approach. Woodhead Publishing, Elsevier, Cambridge (2015)
32. Pasanisi, A., De Rocquigny, E., Bousquet, N., Parent, E.: Some useful features of the Bayesian setting while dealing with uncertainties in industrial practice. In: Proc. of the 19th European Safety and Reliability Conference (ESREL), Prague (2009)
33. Pasanisi, A., Keller, M., Parent, E.: Estimation of a quantity of interest in uncertainty analysis: some help from Bayesian decision theory. Reliab. Eng. Syst. Saf. **100**, 93–101 (2012)
34. Ridolfi, G., Mooij, E.: Regression-Based Sensitivity Analysis and Robust Design, pp. 303–336. Springer International Publishing, Cham (2016)
35. Ronse, A., Mooij, E.: Statistical impact prediction of decaying objects. J. Spacecr. Rocket. **51**(6), 1797–1810 (2014)
36. Rosenblatt, M.: Remarks on a multivariate transformation. Ann. Math. Stat. **23**(3), 470–472 (1952)
37. Rubino, G., Tuffin, B.: Rare Event Simulation Using Monte Carlo Methods. Wiley, New York (2009)
38. Rubinstein, R.Y., Kroese, D.P.: Simulation and the Monte Carlo Method, 2nd edn. Wiley, New York (2008)
39. Sankararaman, S., Mahadevan, S.: Integration of model verification, validation, and calibration for uncertainty quantification in engineering systems. Reliab. Eng. Syst. Saf. **138**, 194–209 (2014)
40. Sobol, I.M.: Sensitivity estimates for nonlinear mathematical models. Math. Model. Comput. Experiments **1**(4), 407–414 (1993)
41. Sobol, I.M.: A Primer for the Monte Carlo Method. CRC Press, Boca Raton (1994)

42. Soize, C.: Uncertainty Quantification: An Accelerated Course with Advanced Applications in Computational Engineering. Interdisciplinary Applied Mathematics. Springer International Publishing, Cham (2017)
43. Sudret, B.: Uncertainty propagation and sensitivity analysis in mechanical models – Contributions to structural reliability and stochastic spectral methods. Habilitation à Diriger des Recherches, Université Blaise Pascal – Clermont II (2007)
44. Tokdar, S.T., Kass, R.E.: Importance sampling: a review. Wiley Interdiscip. Rev. Comput. Stat. **2**(1), 54–60 (2009)
45. Zhang, P.: Nonparametric importance sampling. J. Am. Stat. Assoc. **91**(435), 1245–1253 (1996)

Dynamic System Control Dispatch: A Global Optimization Approach

Giorgio Fasano

Abstract This work originates from research related to an optimal control dispatch problem in space: the problem in question is presented in detail in another chapter of this volume by Anselmi et al. (Control propellant minimization for the next generation gravity mission. In: Fasano G, Pintér JD (eds) Modeling and optimization in space engineering – state of the art and new challenges. Springer, New York, 2019). Here we discuss the general issue of dispatching the control of a dynamic system through a number of actuators, presenting a novel model development and algorithmic solution approach. A control law, expressed in terms of total force and torque demand, represents the operational scenario. This gives rise to a very challenging optimization problem, concerning the actuator accommodation and utilization. Following the model formulation, a dedicated heuristic approach–involving nonlinear and mixed integer linear programming–is proposed. The numerical results presented illustrate the efficiency of the methodology adopted.

1 Introduction

This study takes its inspiration from a space engineering application, namely, the demanding task of spacecraft attitude control. In this context, a dedicated controller has the task of determining, usually at a predetermined frequency, the overall control action, aimed at achieving (step by step) the desired system attitude. A number of thrusters are available to exert the overall force and torque as required. The control designer is presented with the difficult task of positioning and orienting these thrusters on the external surface of the spacecraft, since different layouts are expected to accommodate a significant performance range, in terms of propellant consumption of the entire mission.

G. Fasano (✉)
Thales Alenia Space, Turin, Italy
e-mail: giorgio.fasano@thalesaleniaspace.com

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_4

From a practical point of view, the possible choices concerning the thruster location are usually far less problematic than those relevant to their orientation, since the suitable placements are quite limited. At the same time, the thruster orientation – that is assumed to be kept fixed during the mission – is a major issue. Once on orbit, the requested spacecraft control has to be dispatched (in compliance with given operational rules) through the thrusters, minimizing the propellant consumption. The methodology presented here has been adopted for a real-world application, in the context of the Next Generation Gravity Mission studies (European Space Agency, ESA), discussed by Anselmi et al. [1], in this volume.

The relevant optimization context can be extended, *mutatis mutandis*, to a wide range of real-world high-tech scenarios, including applications in robotics and automation. Therefore our work is intended to provide a general point of view, independently from the current specific field of application. According to this general setting, a dynamic system is considered, assuming that its overall control is performed by a number of actuators.

The remainder of this chapter is structured as follows: Sect. 2 presents the actuator layout problem in terms of global optimization. Since this issue, in its general form, is extremely complex, the problem is redefined adopting a restricted version that limits the layout aspect to actuator orientation only (by supposing that their positions are assigned *a priori*). This specific problem is still an NP-hard problem: regarding the complexity theory, consult, e.g., Cenzer and Remmel [2], Dasgupta et al. [3], Goldreich [4], Harel and Feldman [5], and Rudich and Wigderson [6]. Section 3 outlines a heuristic global optimization (GO) approach aimed at finding satisfactory (albeit sub-optimal or approximate) solutions to the redefined problem. Regarding relevant GO aspects, consult, e.g., Floudas et al. [7], Floudas and Pardalos [8], Horst and Pardalos [9], Liberti and Maculan [10, 11], and Pintér [12, 13]. The overall methodology introduced here consists in partitioning this hard GO problem into two sub-problems that are significantly easier to handle. The first sub-problem focuses on actuator orientation, considering only a subset of control steps from the whole time span considered. The second sub-problem, carrying over the orientations of the actuators obtained as the results of the first one, looks into the final solution of the original problem, considering the entire analysis period (i.e., covering the entire time span).

Sections 4 and 5, respectively, are devoted to the two sub-problems, providing an in-depth discussion on the relevant mathematical models, formulated in terms of nonlinear programming (NLP), mixed integer (linear) programming (MI(L)P), and linear programming (LP). Regarding NLP, we refer, e.g., to Bertsekas [14], Bonnans et al. [15], Brinkhuis and Tikhomirov [16], Hillier and Lieberman [17], Mordecai [18], Nocedal and Wright [19], and Ruszczyński [20]. MI(L)P is discussed, e.g., by Chen et al. [21], Karlof [22], Jünger et al. [23], Hillier and Lieberman [17], Minoux [24], Nemhauser and Wolsey [25], Papadimitriou and Steiglitz [26], and Schrijver [27]. A selection of appropriate LP references are, e.g., Cottle [28], Dantzig and Thapa [29, 30], Gärtner and Matoušek [31], Padberg [32], Roos et al. [33], and Vanderbei [34]. Section 6 surveys possible extensions and applications in a real-world context. Section 7 provides some insight on relevant computational aspects.

2 Problem Statement

We consider a general rigid body system S over a given timeframe $[0, T]$. An appropriate S-based main orthogonal reference frame (O, x, y, z) is defined. This is the only reference frame used throughout this work, and all vectors shall be with respect to it, without referring to this convention. Considering that the overall control of S is executed at discrete time steps, the entire interval $[0, T]$ may be equipartitioned into a set of sub-intervals, of duration Δ each, delimited by $N_I + 1$ instants $i \in \{0, 1, \ \ldots \ , N_I\} = I$ (the number of time steps can be very large, due to control exercised at a high frequency). It is assumed that, at each time instant i, a force $\boldsymbol{F}_i = (F_{xi}, F_{yi}, F_{zi})$ and a torque $\boldsymbol{T}_\mathrm{i} = (T_{xi}, T_{yi}, T_{zi})$, representing the overall control request, have to be exerted on the system through a set of N_A actuators $r \in \{1, \ \ldots \ , N_A\} = A$, during the time sub-interval Δ.

In accordance with the above general framework, the following notations are introduced and used throughout in this study:

$\|\boldsymbol{w}\|$ is the Euclidean norm of vector $\boldsymbol{w}$;
I is the set of time instants, starting from 0, N_I is the last instant;
N_A is the number of actuators;
$F_i = (F_{xi}, F_{yi}, F_{zi})^T$ is the overall force (represented as a column vector), requested by the controller from the actuators at instant i;
$T_i = (T_{xi}, T_{yi}, T_{zi})^T$ is the overall torque (represented as a column vector), requested by the controller from the actuators at instant i;
$\boldsymbol{v}_r = (v_{rx}, v_{ry}, v_{rz})$ are the unit vectors, representing the orientation of each actuator r;
$v = \left(\left(v_{1x}, v_{1y}, v_{1z}\right)^T, \ldots, \left(v_{rx}, v_{ry}, v_{rz}\right)^T, \ldots, \left(v_{N_A x}, v_{N_A y}, v_{N_A z}\right)^T\right)$ is the (sub-)matrix whose columns are the column vectors associated with each $\boldsymbol{v}_r$ (the notation $v = \left(v_{1x}, v_{1y}, v_{1z}, \ldots, v_{rx}, v_{ry}, v_{rz}, \ldots, v_{N_A x}, v_{N_A y}, v_{N_A z}\right)$ is also utilized, depending on the context);
$\boldsymbol{f}_{ri} = (f_{rxi}, f_{ryi}, f_{rzi})$ is the force exerted by actuator r at instant i;
u_{ri} are, for each actuator r, the (Euclidean) norm $\|\boldsymbol{f}_{ri}\|$ of the force exerted at instant i, i.e.,

$$\boldsymbol{f}_{ri} = u_{ri}\boldsymbol{v}_r;$$

$$u = \big(u_{10}, \ldots, u_{r0}, \ldots, u_{N_A 0}, \ldots, u_{1i}, \ldots, u_{ri}, \ldots, u_{N_A i}, \ldots, u_{1N_I}, \ldots, u_{rN_I}, \ldots, u_{N_A N_I}\big);$$

$\boldsymbol{p}_r = (p_{rx}, p_{ry}, p_{rz})$ is, for each actuator r, the application-point vector of $\boldsymbol{f}_{ri}$, holding for all instants i;

$$p = \left(p_{1x}, p_{1y}, p_{1z}, \ldots, p_{rx}, p_{ry}, p_{rz}, \ldots, p_{N_Ax}, p_{N_Ay}, p_{N_Az}\right);$$

$\boldsymbol{p}_r \times \boldsymbol{v}_r = (q_{rx}, q_{ry}, q_{rz})$ is the cross product of $\boldsymbol{p}_r$ and $\boldsymbol{v}_r$;
$p \times v = \left(\left(q_{1x}, q_{1y}, q_{1z}\right)^T, \ldots, \left(q_{rx}, q_{ry}, q_{rz}\right)^T, \ldots, \left(q_{N_Ax}, q_{N_Ay}, q_{N_Az}\right)^T\right)$ is the (sub-)matrix whose columns are the column vectors associated with each cross product $\boldsymbol{p}_r \times \boldsymbol{v}_r$;
$\underline{U}_r, \overline{U}_r$ are, for each actuator r, the lower and upper bounds imposed on u_{ri}, respectively (for the sake of simplicity, it is assumed that both are time-independent);
$D_{vr} \subset \boldsymbol{R}^3$ is a compact domain delimited by specific (geometrical/operational) conditions on actuator r orientations;
$D_{pr} \subset \boldsymbol{R}^3$ is a compact domain delimited by specific (geometrical/operational) conditions on actuator r positions.

Applying the above notation, our modeling framework gives rise to the conditions expressed below (in a compact formalism and allowing for a slight abuse of notation):

$$\forall i \in I \qquad \begin{pmatrix} v \\ p \times v \end{pmatrix} \begin{pmatrix} u_{1i} \\ \cdots \\ u_{ri} \\ \cdots \\ u_{N_Ai} \end{pmatrix} = \begin{pmatrix} F_i \\ T_i \end{pmatrix}, \tag{1}$$

$$\forall r \in A \quad \|\boldsymbol{v}_r\| = 1, \tag{2}$$

$$\forall r \in A \quad \forall i \in I \quad u_{ri} \in \left[\underline{U}_r, \overline{U}_r\right], \quad \forall r \in A \quad \boldsymbol{v}_r \in D_{vr}, \quad \boldsymbol{p}_r \in D_{pr}. \tag{3}$$

Equation (1) expresses the assigned system control law that, in a more explicit (vector) formulation, reads as follows:

$$\forall i \in I \quad \sum_{r \in A} u_{ri} \boldsymbol{v}_r = \boldsymbol{F}_i,$$

$$\forall i \in I \quad \sum_{r \in A} \boldsymbol{p_r} \times (u_{ri} \boldsymbol{v_r}) = \boldsymbol{T_i}.$$

Equation (2) is a normalization condition, stating that each $\boldsymbol{v}_r$ is a unit-vector that determines the orientation of the forces exerted by the corresponding actuator. It is understood that, for each r, the lower bound $\underline{U}_r$ is always non-negative. Hence, the optimization problem can be stated as follows:

Choose the values for (parameters) p, v, as well as for (variables) u, minimizing a given cost function $f(p, v, u)$ (representing, e.g., the overall energy consumption) and subject to conditions (1), (2), and (3).

Obviously, the specific features characterizing D_{vr}, D_{pr}, and $f(p, v, u)$ strongly affect the actual nature and difficulty of the problem in question (that, in all cases considered here, is non-convex). As a first strong simplification, it is assumed that p consists of constant elements, i.e., for each r, D_{pr} is reduced to a single point, characterizing conditions (1) as a set of bilinear equations. Recall that a bilinear function (consisting in a special case of the quadratic class) has the general form $f(r, s) = A_r R + R^T Q S + A_s S$, where $r \in \boldsymbol{R}^{\mathrm{n}}$, $s \in \boldsymbol{R}^{\mathrm{m}}$, R and S are the corresponding column vectors, and A_r, A_s, and Q are the real matrices of dimension $1 \times n$, $1 \times m$, and $n \times m$, respectively. The handling of bilinear terms in an MIP model is discussed, e.g., by Floudas [35]. Furthermore, for each r, D_{vr} is assumed to be a three-dimensional interval, i.e., $\underline{V}_r \leq \left(v_{rx}, v_{ry}, v_{rz}\right)^T \leq \overline{V}_r$. Here $\underline{V}_r$ and $\overline{V}_r$ are the lower and upper bounds, respectively, expressed as column vectors: $\underline{V}_r \geq (-1, -1, -1)^T$ and $\overline{V}_r \leq (1, 1, 1)^T$. Define $f(p, v, u)$ as a separable function with respect to the variables u only. More precisely, with these specifications and simplifications, the original model reads as follows:

$$\min \sum_{\substack{r \in A \\ i \in I}} f_r\left(u_{ri}\right), \tag{4}$$

subject to

$$\forall i \in I \quad \begin{pmatrix} v \\ P \times v \end{pmatrix} \begin{pmatrix} u_{1i} \\ \cdots \\ u_{ri} \\ \cdots \\ u_{N_A i} \end{pmatrix} = \begin{pmatrix} F_i \\ T_i \end{pmatrix}, \tag{5}$$

$$\forall r \in A \quad v_{rx}^2 + v_{ry}^2 + v_{rz}^2 = 1, \tag{6}$$

$$\forall r \in A \quad \forall i \in I \; u_{ri} \in \left[\underline{U}_r, \overline{U}_r\right], \quad \underline{V}_r \leq \left(v_{rx}, v_{ry}, v_{rz}\right)^T \leq \overline{V}_r. \tag{7}$$

Here Eq. (2) has been substituted with the more convenient (equivalent) expressions (6). Furthermore, it is assumed that the cost associated with each actuator r

remains the same, independently from the time instant $i \in [0, T]$ considered. Bear also in mind that the convention of denoting the model variables with lower-case characters, and the constants with capital letters, has been adopted (in respect to this, $p \times v$ has been substituted with $P \times v$ in (5)). The same formal rule shall be followed in this chapter.

The specific occurrence (quite common in practice, at least in terms of an acceptable approximation), where the objective function appearing in (4) reduces to a linear one, is, for its simplicity, of particular interest. In this case (4) assumes the explicit expression shown below:

$$\min \sum_{\substack{r \in A \\ i \in I}} K_r u_{ri}, \tag{8}$$

where K_r are the positive constants, representing for each actuator r the associated cost per unit force.

In order to make the overall problem in question more representative of real-world scenarios, an appropriate enhancement is proposed. As a first refinement, the following restrictions are introduced:

$$\forall i \in \{0, 1, \ldots, N_I - 1\} \qquad | u_{r(i+1)} - u_{ri} | \leq L_r, \tag{9}$$

where L_r are the given positive constants. These inequalities state that the maximum allowable difference of the forces applied by the same actuator in two subsequent instants cannot exceed a given threshold. Constraints (9) represent for each r a Lipschitz condition expressed with respect to each Δ on each variable u_r, interpreted as a time-function.

In order to prevent the possible overworking of some particular actuators (presumably, among those with a lower cost), global conditions are introduced to limit the utilization of each actuator throughout the time period $[0, T]$. For this purpose, the following constraints are introduced:

$$\forall r \quad \sum_{i \in I} u_{ri} \leq \frac{J_r}{\Delta}, \tag{10}$$

where J_r represents, for each actuator r, a (technological) upper bound on the total impulse (scalar) admissible during the whole time span.

The present work mainly focuses on the formulation provided above, consisting of constraints (5)–(10) and objective function (8). This specific statement of the problem shall be referred to, in the following, as P. It is summarized below for the sake of convenience:

$$(P) \quad \min \sum_{\substack{r \in A \\ i \in I}} K_r u_{ri},$$

subject to

$$\forall i \in I \quad \begin{pmatrix} v \\ P \times v \end{pmatrix} \begin{pmatrix} u_{1i} \\ \cdots \\ u_{ri} \\ \cdots \\ u_{N_A i} \end{pmatrix} = \begin{pmatrix} F_i \\ T_i \end{pmatrix},$$

$$\forall r \in A \quad v_{rx}^2 + v_{ry}^2 + v_{rz}^2 = 1,$$

$$\forall r \in A \ \ \forall i \in \{0, 1, \ldots, N_I - 1\} \quad \mid u_{r(i+1)} - u_{ri} \mid \leq L_r,$$

$$\forall r \in A \quad \sum_{i \in I} u_{ri} \leq \frac{J_r}{\Delta},$$

$$\forall r \in A \ \ \forall i \in I \ \ u_{ri} \in \left[\underline{U}_r, \overline{U}_r\right], \quad \underline{V}_r \leq \left(v_{rx}, v_{ry}, v_{rz}\right)^T \leq \overline{V}_r.$$

As a further consideration on the overall statement of the problem, it should be observed that P (or its possible variations) can be solved exclusively on the implicit assumption that there exists (at least) a single set of N_A (time-independent) unit vectors allowing for a feasible solution of the assigned constraints, at any time instant. From a physical point of view, this means that it is always possible to identify a fixed orientation for the actuators to satisfy the control demand, instant by instant, over the entire time span, in compliance with all operational restrictions. In some real-world applications, this supposition may not be fully applicable. To overcome this shortcoming, a possible relaxation of the problem could be taken into account, by adding in (5) error variables defined within given tolerance ranges and readjusting, if necessary, the objective function adequately, e.g., by introducing the total error as a term to be minimized. Equation (5) may therefore be substituted with the following:

$$\forall i \in I \quad \begin{pmatrix} v \\ P \times v \end{pmatrix} \begin{pmatrix} u_{1i} \\ \cdots \\ u_{ri} \\ \cdots \\ u_{N_A i} \end{pmatrix} = \begin{pmatrix} F_i + \varepsilon_{Fi} \\ T_i + \varepsilon_{Ti} \end{pmatrix}, \tag{11}$$

$$\forall i \in I \quad -E_F \leq \varepsilon_{Fi} \leq E_F, \quad -E_T \leq \varepsilon_{Ti} \leq E_T. \tag{12}$$

Here $\varepsilon_{Fi} = (\varepsilon_{Fxi}, \varepsilon_{Fyi}, \varepsilon_{Fzi})^T$, $\varepsilon_{Ti} = (\varepsilon_{Txi}, \varepsilon_{Tyi}, \varepsilon_{Tzi})^T$, $E_F > 0$, and $E_T > 0$ are the chosen admissible levels of tolerance (expressed as column vectors). Albeit quite useful in practice, this approach shall not be further investigated in this work.

Let us also remark that although in most real-world scenarios the primary optimization target is to minimize the overall energy consumption, alternative objective functions could be considered, e.g., the overall difference in contribution among the actuators could be minimized (these aspects are also omitted here from further discussion).

3 A Heuristic Solution Perspective

Notwithstanding the fact that P is, actually, a simplified version of the overall problem outlined in Sect. 2, it still represents a very challenging numerical issue. Indeed, despite adopting a linear objective function, the relevant formulation relates to a non-convex quadratically constrained model, well known for being NP-hard. (Since any 0–1 integer problem can be formulated through a quadratically constrained model, quadratically constrained programming is, in general, NP-hard; similar considerations are valid, concerning the class of bilinear constraints.)

Problem complexity becomes even more evident, when large-scale instances are to be solved, as in the real-world context addressed by this work. To give an example, an operational scenario encompassing 5000 time instants gives rise to 30,000 bilinear equations, corresponding to (5).

In general, given N_I instants and N_A actuators, the resulting instance contains:

$6N_I$ bilinear equations, corresponding to (5);
N_A quadratic equations, corresponding to (6);
$(2N_I + N_A)$ linear equations, corresponding to (9) and (10), respectively;
$3N_A$ (continuous and bounded) variables v;
$N_A(N_I + 1)$ (continuous and bounded) variables u.

Therefore our study puts forward a heuristic methodology, aimed at providing approximate and sub-optimal solutions to P, or some possible variation of it, at a satisfactory level for a number of real-world applications. As a preliminary consideration, coping with this issue alone appears to be an extremely arduous task to achieve – except for very specific small-scale instances. The basic idea is to partition problem P into much easier sub-problems and to solve these (iteratively or recursively, if necessary).

The approach proposed takes advantage of the fact that our optimization problem has a very specific structure. First and foremost, the variables involved can be partitioned into two sets, whose cardinalities may be assumed to be significantly different from each other. The set corresponding to variables v has cardinality $3|A|$,

while the set associated with variables u has cardinality $|A||I|$: note that $|I| >> |A|$. In virtue of this feature, most of the nonlinear equations corresponding to (5) are bilinear, while the remaining constraints (of a limited number) are quadratic. If we fix variables v, then the resulting problem becomes linear, since all bilinear equations (5) reduce to linear and all quadratic relations (6) can be dropped. This suggests that two separate sub-problems could be considered: the first, called the *Reduced-Time-Domain* sub-problem (P_R), is aimed at finding a suitable set of values for the variables v, in order to make P linear (i.e., easy to solve); the second, referred to as the *Overall-Time-Domain* sub-problem (P_O), is this LP reduction.

More precisely, P_R mainly addresses the orientation of the actuators, i.e., the identification of sets of values for all variables v. For this purpose, limited subsets of instants (i.e., subsets of I), assumed to be representative of the whole time span, are taken into account. These can be generated either randomly or (more appropriately) through an *ad hoc* optimization process, selecting the instants on the basis of a best-representative criterion. (These aspects will not be discussed within this work).

On the other hand, P_O consists in optimizing the overall problem P, covering the entire set of instants I, once the variables v have been assigned the values obtained by solving P_R. The result thus obtained (if necessary by introducing a certain tolerance level with respect to Eq. (5)) is, in general, a sub-optimal solution of the original problem P. If this solution is not deemed satisfactory, then a further set of values for the all variables v is considered and the search process continues until a predefined stopping rule is satisfied. The generation of v can be obtained by solving a new P_R instance (based on a different subset of instants), or by meta-heuristic processes such as genetic or path-relinking algorithms.

Refinements of the current or final P_O solution may be carried out by applying local NLP or sequential linear programming (SLP), cf. e.g., Nocedal and Wright [36]. The overall search process applies, albeit heuristically, a global optimization logic. Specific models with different levels of approximation have therefore been conceived to tackle both sub-problems P_R and P_O. These shall be denoted and referred to as *Reduced-Time-Domain* (RTD) and *Overall-Time-Domain* (OTD) models.

4 Reduced-Time-Domain Models

As outlined in Sect. 3, sub-problem P_R is derived from P by replacing the set of instants I by a subset $\underline{I} \subset I$. The following general model form is assumed:

$$(P_R) \quad \min \sum_{\substack{r \in A, \\ i \in \underline{I}}} K_r u_{ri},$$

subject to

$$\forall i \in I \quad \begin{pmatrix} v \\ P \times v \end{pmatrix} \begin{pmatrix} u_{1i} \\ \cdots \\ u_{ri} \\ \cdots \\ u_{N_A i} \end{pmatrix} = \begin{pmatrix} F_i \\ T_i \end{pmatrix},$$

$$\forall r \in A \quad v_{rx}^2 + v_{ry}^2 + v_{rz}^2 = 1,$$

$$\forall r \in A \;\; \forall i \in I \;\; u_{ri} \in \left[\underline{U}_r, \overline{U}_r\right], \quad \underline{V}_r \leq \left(v_{rx}, v_{ry}, v_{rz}\right)^T \leq \overline{V}_r.$$

Note that in this case inequalities (9) and (10) have been eliminated. Conditions (9), indeed, are stated for all couples of subsequent instants of I and are not supposed to carry over to $\underline{I}$ (since it does not necessarily include pairs of subsequent instants). Inequalities (10) could be kept, provided that the terms J_r are properly reduced; however, these conditions are neglected for the sake of simplicity.

A further significant simplification of P_R is obtained when, for some r, a component $(v_{r\rho})$ of the relative orientation vector $\boldsymbol{v}_r$ is either non-negative or non-positive, i.e., $v_{r\rho} \in [0, 1]$ or $v_{r\rho} \in [-1, 0]$, $\rho \in \{x, y, z\}$. In this case, multiple solutions are implicitly excluded, e.g., if $v_{rx} \in [0, 1]$, then the solution $v_{rx} = -\sqrt{1 - v_{ry}^2 - v_{rz}^2}$ is not admissible.

GO approaches may be considered to handle sub-problem P_R. The first approach is to solve P_R directly using a global NLP solver, perhaps additionally providing a suitable initial solution. However, since even finding any quasi-feasible solution could be a challenging task, it is advisable to look for alternatives. The results obtained by these can be refined through a (not necessarily global) NLP algorithm. Two dedicated MILP models, aimed at finding approximate (sub-optimal) solutions to P_R, are illustrated hereinafter. Note that the level of approximation and the solution quality can be refined through iterative procedures that are, however, not discussed here.

4.1 Discretized Reduced-Time-Domain Models

A discretized formulation of sub-problem P_R, achieved in terms of an MILP model, is illustrated hereinafter and referred to as DRTD. In this setting, the variables v_{rx}, v_{ry}, and v_{rz} corresponding to all possible orientations of each actuator r are no longer supposed to be continuous but can only take a finite number of values (the non-trivial problem of generating a set of discretized orientations, as "evenly" as possible, can be related to that of the distribution of points on a unit sphere, e.g.,

[37]). The resulting sub-problem is therefore an approximation of P_R; evidently, the larger the number of these discretized values is, the more accurate the formulation becomes.

Let us consider the index set Ω, corresponding to all possible discretized orientations for the actuators: it is supposed, without loss of generality, that all actuators have the same number of admissible orientations. The set $\Omega_r = \{\boldsymbol{V}_{r1}, \ \ldots \ , \boldsymbol{V}_{r\omega}, \ \ldots \ , \boldsymbol{V}_{r\,|\,\Omega|}\}$ is associated with each r, in compliance with (7), by defining $\boldsymbol{V}_{r\omega} = (V_{r\omega x}, V_{r\omega y}, V_{r\omega z})$ as the unit-vector corresponding to orientation ω. Next, the binary variables $\delta_{r\omega} \in \{0, 1\}$ are introduced, with the following interpretation:

$\delta_{r\omega} = 1$ if r has orientation ω;
$\delta_{r\omega} = 0$ otherwise.

The continuous variables $u_{r\omega i} \in \left[0, \overline{U}_r\right]$ represent the force exerted by each actuator r, with respect to orientation ω, at each instant i. It is understood, as is shown in detail below, that if actuator r has orientation ω, then $\forall i \in \underline{I} \ \ \forall \omega' \neq \omega$ $u_{r\omega' i} = 0$.

The following mutually exclusive conditions hold, stating that each actuator r is assigned to exactly one orientation ω:

$$\forall r \in A \quad \sum_{\omega \in \Omega} \delta_{r\omega} = 1. \tag{13}$$

The following inequalities guarantee that each actuator r can exert a non-zero force, only in correspondence to the chosen orientation associated with the ω for which $\delta_{r\omega} = 1$:

$$\forall r \in A \quad \forall \omega \in \Omega \quad \sum_{i \in \underline{I}} u_{r\omega i} \leq \overline{U}_r \mid \underline{I} \mid \delta_{r\omega}. \tag{14}$$

Here $\overline{U}_r \mid \underline{I} \mid$ are set as *big M* values, cf. e.g., Williams [38]. Conditions (14) state that, if orientation ω is not selected for actuator r, then the variables $u_{r\omega i}$ are set to zero, for all instants $i \in \underline{I}$, i.e., the forces exerted by each actuator are directed along the corresponding orientation vector only.

The lower bounds $\underline{U}_r$ on the force applied by each r, for all instants of $\underline{I}$, are expressed by

$$\forall r \in A, \ \ \forall \omega \in \Omega, \ \ \forall i \in \underline{I} \quad u_{r\omega i} \geq \underline{U}_r \delta_{r\omega}. \tag{15}$$

The following vector equations represent the control law conditions (5):

$$\forall i \in \underline{I} \quad \sum_{\substack{r \in A \\ \omega \in \Omega}} u_{r\omega i} \boldsymbol{V}_{r\omega} = \boldsymbol{F}_i, \tag{16}$$

$$\forall i \in \underline{I} \quad \sum_{\substack{r \in A \\ \omega \in \Omega}} \boldsymbol{P}_r \times (u_{r\omega i} \boldsymbol{V}_{r\omega}) = \boldsymbol{T}_i . \tag{17}$$

An appropriate formulation of (16) and (17) corresponding to (11) and (12) could be also taken into consideration. Finally, the objective function corresponding to (8) is rewritten by directly substituting variables u_{ri} with $u_{r\omega i}$:

$$\min \sum_{\substack{r \in A \\ \omega \in \Omega \\ i \in \underline{I}}} K_r u_{r\omega i} . \tag{18}$$

A practically useful variant of the DRTD model derived above is briefly outlined here. For this purpose, for any actuator r, a one-to-one correspondence is established between Ω_r and the set $\Omega_{\phi r} \times \Omega_{\vartheta r}$ of the discretized polar and azimuthal angles $(\phi_{\mathrm{r}1}, \vartheta_{\mathrm{r}1}), \ \ldots \ , (\phi_{\mathrm{r}\omega}, \vartheta_{\mathrm{r}\omega}), \ \ldots \ , (\phi_{\mathrm{r}\,|\,\Omega|}, \vartheta_{\mathrm{r}\,|\,\Omega|})$ in the spherical coordinate system centered in the application-point $\boldsymbol{P}_r$ (applying the orientation of the main reference frame). This allows for a dramatic reduction of the binary variables involved, giving rise to a significantly enhanced DRTD model version. (As we know, advantageous reformulations of MI(L)P problems are frequently obtained by decreasing the number of 0–1 variables.) The substitution of $\delta_{r\omega}$ with $\delta_{r\phi}$ and $\delta_{r\vartheta}$ (also binary) is therefore adopted, by stating the condition:

$$\delta_{r\omega} = 1 \iff \left(\delta_{r\phi} = 1\right) \wedge \left(\delta_{r\vartheta} = 1\right),$$

where it is implied that $(\delta_{r\phi} = 0) \vee (\delta_{r\vartheta} = 0)$ otherwise.

Conditions (13)–(15) are hence rewritten as follows:

$$\forall r \in A \quad \sum_{\phi \in \Omega_\phi} \delta_{r\phi} = 1, \tag{19}$$

$$\forall r \in A \quad \sum_{\vartheta \in \Omega_\vartheta} \delta_{r\vartheta} = 1, \tag{20}$$

$$\forall r \in A \quad \forall \phi \in \Omega_\phi \quad \sum_{\substack{\vartheta \in \Omega_\vartheta \\ i \in \underline{I}}} u_{r\phi\vartheta i} \leq \overline{U}_r \mid \underline{I} \mid \delta_{r\phi}, \tag{21}$$

$$\forall r \in A \quad \forall \vartheta \in \Omega_\vartheta \qquad \sum_{\substack{\phi \in \Omega_\phi \\ i \in \underline{I}}} u_{r\phi\vartheta i} \leq \overline{U}_r \mid \underline{I} \mid \delta_{r\vartheta}, \tag{22}$$

$$\forall r \in A \;\; \forall \phi \in \Omega_\phi \;\; \forall \vartheta \in \Omega_\vartheta \;\; \forall i \in \underline{I}, \tag{23}$$

$$u_{r\phi\vartheta i} \geq \underline{U}_r \delta_{r\phi} + \underline{U}_r \delta_{r\vartheta} - \underline{U}_r.$$

Here the variables $u_{r\phi\vartheta i} \in \left[0, \overline{U}_r\right]$ replace the corresponding $u_{r\omega i}$. Analogous substitutions hold for (16)–(18).

4.2 *Relaxation of the Discretized Reduced-Time-domain Model*

This section discusses a possible relaxation of the DRTD model. This approach, aimed at finding approximate solutions to sub-problem P_R, is of use in practice, especially when $\mid \underline{I} \mid$ is quite large and, as such, likely to give rise to hard model instances. For this purpose, the joint use of two dedicated sub-models has been considered, denoted as DRTD-1 and DRTD-2, respectively. In DRTD-1, the bilinear terms $v_{rx}u_{ri}$, $v_{ryr}u_{ri}$, and $v_{rz}u_{ri}$ are substituted with the variables w_{rxi}, w_{ryi}, and w_{rzi}, respectively (assumed as independent). Conditions (5)–(7) are rewritten as follows:

$$\forall i \in \underline{I} \quad \sum_{r \in A} \boldsymbol{w}_{ri} = \boldsymbol{F}_i, \tag{24}$$

$$\forall i \in \underline{I} \quad \sum_{r \in A} \boldsymbol{P}_r \times \boldsymbol{w}_{ri} = \boldsymbol{T}_i, \tag{25}$$

$$\forall r \in A \;\; \forall i \in \underline{I} \quad \underline{U}_r^2 \leq w_{rxi}^2 + w_{ryi}^2 + w_{rzi}^2 \leq \overline{U}_r^2, \tag{26}$$

where $\boldsymbol{w}_{ri} = (w_{rxi}, w_{ryi}, w_{rzi})$.

Having adopted the variables w_{rxi}, w_{ryi}, and w_{rzi} instead of $v_{rx}u_{ri}$, $v_{ry}u_{ri}$, and $v_{rz}u_{ri}$, Eqs. (24) and (25), in addition to inequalities (26), represent for P_R a set of necessary conditions. The quadratic constraints (26) are separable and can easily be approximated by piecewise linear functions and handled by means of SOS2 (special ordered sets of type 2) relations (e.g., [38]). A further relaxation of P_R can be carried out by dropping the lower bounds in (26): these are replaced by $w_{rxi}^2 + w_{ryi}^2 + w_{rzi}^2 \leq \overline{U}_r^2$. The latter are convex and, as such, manageable, as piecewise linear functions, in an LP framework (e.g., [38]).

Since, for all instants $i \in \underline{I}$ and for each actuator r, the variables u_{ri} are non-negative (positive, if $\underline{U}_r > 0$), the signs of variables w_{rxi}, w_{ryi}, and w_{rzi} are determined by the actuator orientation. Consequently, the signs of these variables are kept unvaried for all instants $i \in \underline{I}$, as is expressed by the following conditions:

$$\forall r \in A \quad \forall i \in \underline{I} \;\; \rho = x, y, z,$$

$$\sigma_{r\rho}^{-} + \sigma_{r\rho}^{0} + \sigma_{r\rho}^{+} = 1, \tag{27}$$

$$w_{r\rho i} \geq \underline{U}_r \sigma_{r\rho}^{0} - \underline{U}_r, \tag{28a}$$

$$w_{r\rho i} \leq -\overline{U}_r \sigma_{r\rho}^{0} + \overline{U}_r, \tag{28b}$$

$$w_{r\rho i} \leq -\Delta_S \sigma_{r\rho}^{-} + \sigma_{r\rho}^{+} \overline{U}_r, \tag{28c}$$

$$w_{r\rho i} \geq \Delta_S \sigma_{r\rho}^{+} - \sigma_{r\rho}^{-} \underline{U}_r, \tag{28d}$$

$$\sum_{\rho} \sigma_{r\rho}^{0} \leq 2. \tag{29}$$

Here $\sigma_{r\rho}^{-}, \sigma_{r\rho}^{0}, \sigma_{r\rho}^{+} \in \{0, 1\}$, $\pm\Delta_S$ is a predefined tolerance with respect to zero and (29) states that at most two direction cosines of the same actuator r can be zero.

Since it is expected that, in general, the domain corresponding to constraints (24)–(29) contains a subset of solutions that are not feasible for P_R, additional conditions can be considered, in the perspective of a heuristic approach. The following linear (non-necessary) conditions are hence included:

$$\forall r \in A \;\; \forall i \in \underline{I} \;\; \rho = x, y, z \quad w_{r\rho i} = h_{r\rho} F_{\rho i} + \eta_{r\rho i}, \tag{30}$$

where $h_{r\rho}$ are the non-negative variables, $\eta_{r\rho i} \in [-E, E]$, and $E > 0$ is a parameter expressing a tolerance on these additional conditions. The variables $h_{r\rho}$ represent the contributions per unit force provided by each actuator r and are supposed to be approximately the same for all instants of $\underline{I}$. This assumption is deemed to be acceptable, as a first approximation, in a number of real-world scenarios; in any case, it may be adopted as a useful trick to approximate the solution feasibility by acting suitably on E.

In order to express objective function (8), as a function of the variables w_{rxi}, w_{ryi}, and w_{rzi}, the following second-order-cone constraints [39, 40] can be introduced:

$$w_{rxi}^2 + w_{ryi}^2 + w_{rzi}^2 \leq l_{ri}^2, \tag{31}$$

where l_{ri} are the non-negative variables. This way, the optimization target (8) is rewritten as

$$\min \sum_{\substack{r \in A \\ i \in \underline{I}}} K_r l_{ri}. \tag{32}$$

As a possible (albeit naïve) alternative, inequalities (31) can be tackled through piecewise linearization: in such a case, SOS2 conditions would be requested to approximate the terms l_{ri}^2, exclusively. It is nonetheless evident that, in the perspective of the heuristic approach adopted, instead of resorting to (31) and (32) objective function (8) could be replaced by the surrogate $\sum_{r \in A, i \in \underline{I}} K_r \left(w_{rxi} + w_{ryi} + w_{rzi}\right)$ to be minimized, based on the l_1 norm. This makes it significantly easier to deal with: a suitable trade-off between the adopted objective function and the parameter E could moreover be looked for by applying an iterative process.

The DRTD-1 model discussed above has the scope of providing realistic approximations for the bilinear terms $v_{rx}u_{ri}$, $v_{ry}u_{ri}$, and $v_{rz}u_{ri}$, in compliance with the assigned control law; the restrictions involving variables v_{rx}, v_{ry}, v_{rz}, and u_{ri}, respectively, have been neglected, however.

The DRTD-2 model, as is described hereinafter, is conceived in order to find approximate values for these variables (compliant with the given bounds), starting from the DRTD-1 (optimal) solutions, denoted as W_{rxi}^*, W_{ryi}^*, and W_{rzi}^*. By setting $v_{rx}u_{ri} = W_{rxi}^*$, $v_{ry}u_{ri} = W_{ryi}^*$, and $v_{rz}u_{ri} = W_{rzi}^*$, the following conditions, holding for $W_{rxi}^* \neq 0$, $W_{ryi}^* \neq 0$, and $W_{rzi}^* \neq 0$, are stated:

$$\forall r \in A \ \ \forall i \in \underline{I} \quad W_{ryi}^* v_{rx} = W_{rxi}^* v_{ry} - \eta_{rxyi}^- + \eta_{rxyi}^+, \tag{33a}$$

$$\forall r \in A \ \ \forall i \in \underline{I} \quad W_{rzi}^* v_{rx} = W_{rxi}^* v_{rz} - \eta_{rxzi}^- + \eta_{rxzi}^+, \tag{33b}$$

$$\forall r \in A \ \ \forall i \in \underline{I} \quad W_{rzi}^* v_{ry} = W_{rzi}^* v_{rz} - \eta_{rzi}^- + \eta_{ryzi}^+. \tag{33c}$$

Here the additional variables η represent possible errors (on the basis of the DRTD-1 results, some variables v_r are set a priori to 0 or 1). The following relaxation of condition (6) is considered:

$$\forall r \in A \quad v_{rx}^2 + v_{ry}^2 + v_{rz}^2 \leq 1. \tag{34}$$

This convex inequality, which can be approximated in terms of piecewise linear functions, without imposing SOS2 conditions, does not guarantee per se that, for

each r, $v_{rx}^2 + v_{ry}^2 + v_{rz}^2 = \|\boldsymbol{v}_r\|^2 = 1$. If $\|\boldsymbol{v}_r\| < 1$, then $\boldsymbol{v}_r$ is substituted with the normalized vector $\frac{\boldsymbol{v}_r}{\|\boldsymbol{v}_r\|}$ and the associated variables u_{ri} (corresponding to $v_{rx}u_{ri} = W_{rxi}$, and/or $v_{ry}u_{ri} = W_{ryi}$, and/or $v_{rz}u_{ri} = W_{rzi}$) are replaced with $\|\boldsymbol{v}_r\|u_{ri}$. This is always admissible when $\underline{U}_r = 0$; otherwise, a relaxation of the bound $\underline{U}_r$ might be implicit (by redefining $\underline{U}_r$ as $\underline{U}'_r = \min_{i \in \underline{I}} \left\{ \underline{U}_r, \|\boldsymbol{v}_r\| u_{ri} \right\}$).

The total error relative to the additional variables η of Eqs. (33a), (33b), (33c) is minimized:

$$\min \sum_{\substack{r \in A \\ i \in \underline{I}}} \left(\eta^-_{rxyi} + \eta^+_{rxyi} + \eta^-_{rxzi} + \eta^+_{rxzi} + \eta^-_{ryzi} + \eta^+_{ryzi} \right). \tag{35}$$

The DRTD-1 and DRTD-2 models, based on an overall relaxation of the corresponding DRTD, are aimed at providing approximate (sub-optimal) solutions to P_R that are expected to be at least nearly feasible. The results obtained with DRTD-2 can be refined either by solving P_R directly, as an NLP problem (taking the DRTD-2 results as starting solutions), or adopting the DRTD model. This can be used to find local optima, with respect to suitable (discretized) neighborhoods of the DRTD-2 solutions. Alternatively, these can be utilized by the OTD model directly (without further refinement, within the overall heuristic procedure outlined in Sect. 3).

5 Overall Time-Domain Model and Solution Refinement

The OTD model assumes the following linear formulation:

$$(P_O) \quad \min \sum_{\substack{r \in A \\ i \in I}} K_r u_{ri},$$

subject to

$$\forall i \in I \quad \begin{pmatrix} V^* \\ P \times V^* \end{pmatrix} \begin{pmatrix} u_{1i} \\ \cdots \\ u_{ri} \\ \cdots \\ u_{N_A i} \end{pmatrix} = \begin{pmatrix} F_i \\ T_i \end{pmatrix},$$

$$\forall i \in \{0, 1, \ldots, N_I - 1\} \quad | u_{r(i+1)} - u_{ri} | \leq L_r,$$

$$\forall r \in A \quad \sum_{i \in I} u_{ri} \leq \frac{J_r}{\Delta},$$

$$\forall r \in A \;\; \forall i \in I \;\; u_{ri} \in \left[\underline{U}_r, \overline{U}_r\right].$$

Here the terms V^* are given by the values obtained as (optimal) solutions of sub-problem P_R. Since the imposition of these values could jeopardize the feasibility of P_O, even if P is feasible per se, the relaxations (11) and (12) should be taken into consideration, with an appropriate adaptation of the optimization objective. If the solution obtained for P_O is deemed satisfactory, then this solution, together with the values set for the parameters V^*, can be accepted as the solution of P. Otherwise, the optimization process is carried on, searching for improved values for V^*, as per the overall heuristic approach outlined in Sect. 3.

The way the various phases are activated determines the specificity of the heuristic strategy adopted (once again, this non-trivial topic is not dealt with here). The additional optimization aimed at refining the values of V^* (possibly considering a larger subset of instants) can be carried out by solving directly P_R as an NLP problem. Here we take advantage of the available solution as an initial guess for the adopted local/global NLP algorithm. The DRTD model itself can be used (to perform local search) recursively, refining step-by-step, the discretization adopted for the vectors $\boldsymbol{v}_r$, while restricting the relative admissibility neighborhoods, coincident for each r with $\left[\underline{V}_r, \overline{V}_r\right]$ at the initial step.

Additionally, refinements of any solution of P_O may be explored. A first approach consists in adopting a large-scale NLP local optimization approach, exploiting the solution available at hand as an initial guess. As an interesting alternative, sequential linear programming (SLP) could be considered. This can be carried out by solving recursively a sequence of problem P approximations (through first-order Taylor expansion of each quadratic term) within appropriate *trust regions*. A case of particular interest occurs when for some r, one of their elements can be expressed as a function of the others (e.g., $v_{xr} = \sqrt{1 - v_{yr}^2 - v_{zr}^2}$). This circumstance allows the elimination of conditions (6).

6 Extensions and Related Problems

The control dispatch problem stated in Sect. 2 has been discussed focusing on the specific formulation of P. A number of its extensions could also be studied. Some extended problems are briefly dealt with in this section as illustrative examples. Anything but non-trivial another issue, certainly of interest in several real-world applications, is further taken into account. It concerns the control dispatch, in terms of actions to accomplish, step-by-step, within given (usually very strict) time limits.

6.1 Extended Problems

A first type of extensions concerns possible generalizations of objective function (8). Provided that it is reasonable to maintain its separability feature, it may be represented in the more general (not necessarily linear) form $\sum_{\substack{r \in A \\ i \in I}} f_r(u_{ri})$. Depending on the specific connotation of each term, the resulting function can be either convex or non-convex. Piecewise linear formulations may be adopted to approximate each nonlinear function, with the introduction of SOS2 restrictions, in case of non-convexity.

A further issue that could be of interest in some applications deals with the determination of the actuator positions, in addition to their orientations. This certainly represents a very challenging task. A possible approach consists in tackling the resulting non-convex NLP problem directly, by using a global optimization solver engine. To this purpose, problem P can be solved as a first step, by making suitable assumptions on the positions of the actuators. Its solutions are subsequently taken as an initial guess for the extended problem. As an alternative, a discretized set of values, admissible for the positions of the actuators, may be considered. As already pointed out, refinements of the solutions, also in this extended version, could be found either by large-scale local NLP or SLP: in both cases, the positions of the actuators should be considered as variables.

An additional and by no means easier extension might concern the problem of realizing a redundant system of actuators, up to satisfying, at each instant, the control demand, should any actuator break down. The difficulty of the problem becomes even greater when it is assumed that more than one actuator is not working and/or the actuator positions are not fixed. These aspects, in addition to the above mentioned extended problems, will be the subject of future research.

6.2 Step-by-Step Control Dispatch

The step-by-step control dispatch problem has to be solved at each instant, after receiving the control demand as input, in terms of overall force and torque. Conditions (9) are imposed on the current step, taking into account the solution of the previous step. The intersections of the resulting intervals for each u_r with those established by conditions (7) give rise for each actuator r, to the lower and upper bounds $\left[\underline{U}_{Sr}, \overline{U}_{Sr}\right]$. The obtained problem is referred to as P_S and stated as follows:

$$(P_S) \quad \min \sum_{r \in A} K_r u_{Sr},$$

subject to

$$\begin{pmatrix} V \\ P \times V \end{pmatrix} u_{Sr} = \begin{pmatrix} F_S \\ T_S \end{pmatrix},$$

$$\forall r \in A \quad u_{Sr} \in \left[\underline{U}_{Sr}, \overline{U}_{Sr}\right].$$

Here u_{Sr} are the forces associated with the actuators at the current instant, whereas F_S and T_S correspond to the overall force and torque requested; V and $P \times V$ are the given sub-matrices, corresponding to the given positions and orientation of the actuators. As it is immediately seen, P_S is an LP problem. It is assumed that the system of equations in P_S is consistent and has rank (6) and, as such, it can be solved directly by an LP algorithm [41, 42]. In a number of real-world applications this approach could nevertheless result in being inadequate, due to the execution time necessary that might exceed the limit imposed at any step. The Moore–Penrose pseudo-inverse [43–45] cannot even be utilized to obtain a feasible solution to P_S, due to the presence of the lower and upper bounds.

As a possible alternative, a naïve heuristic approach, aimed at providing fast sub-optimal and approximate solutions to P_S, is briefly outlined here. In order to find sub-optimal solutions to P_S rapidly, it is substituted with the convex nonlinear P_{SNL} shown below:

$$(P_{SNL}) \quad \min \sum_{r \in A} \left[K_r u_{Sr} + H_r (u_{Sr} - U_{0r})^2\right],$$

subject to

$$\begin{pmatrix} V \\ P \times V \end{pmatrix} u_{Sr} = \begin{pmatrix} F_{Sr} \\ T_{Sr} \end{pmatrix}.$$

Here for each r, $U_{0r} = \frac{\underline{U}_{Sr} + \overline{U}_{Sr}}{2}$ and $H_r > 0$ are penalty parameters that can properly be chosen, in order to meet the conditions $u_r \in \left[\underline{U}_{Sr}, \overline{U}_{Sr}\right]$, as established in P_S. Alternative and more refined selections for $U_0 = \left(U_{01}, \ldots, U_{0r}, \ldots, U_{0N_A}\right)$ could also be considered. This suggests that an iterative process can be designed to achieve an acceptable trade-off, between obtaining the actual optimal solution of P_S and the compliance with the relevant bounds. (In most cases, a first good estimation of the parameters H_r may be available a priori, e.g., derived from a previous statistical analysis.) At each iteration ι, a set of parameters $H_{r\iota}$ is chosen, either by appropriately increasing or decreasing their values with respect to the previous iteration step, and the corresponding problem P_{SNL} solved.

An analytical (i.e., closed-form) solution of P_{SNL} can be determined a priori, by making use of the multipliers $(\lambda_x, \lambda_x, \lambda_z, \mu_x, \mu_y, \mu_z)$, corresponding to equations $Vu_S = F_S$ and $(P \times V)u_S = T_S$, respectively. The Lagrangian $\mathcal{L}(u, \lambda, \mu)$ is

therefore introduced, with the conditions $\nabla_{u,\lambda,\mu}\mathcal{L}(u,\lambda,\mu) = 0$. This gives rise to the following (linear) equations:

$$\forall r \in A \quad K_r + 2H_r\,(u_{Sr} - U) + V_{rx}\lambda_x + V_{ry}\lambda_y + V_{rz}\lambda_z + (P \times V)_r\mu_x + (P \times V)_r\mu_y + (P \times V)_r\mu_z = 0,$$

$$\begin{pmatrix} V \\ P \times V \end{pmatrix} u_{Sr} = \begin{pmatrix} F_S \\ T_S \end{pmatrix}.$$

7 Computational Aspects

An experimental analysis (currently on-going) has been performed to date, mainly focusing on sub-problem P_R: in the framework considered, this represents a major task in terms of computational effort. A bird's-eye view of these aspects is provided hereinafter, focusing on a real-world application arising in a specific space engineering context (hence, technical details have to be kept confidential). Test instances of P_R have been solved both directly by an NLP (GO) approach and by the DRTD model. The following optimizers have been adopted for the purpose:

LGO [46–48];
IBM ILOG CPLEX (12.3, [49]).

LGO has been utilized as the global nonlinear solver, while CPLEX as the MILP solver. Tests have been performed on a personal computer, equipped with Core 2 Duo P8600, 2.40 GHz processor; 1.93 GB RAM; MS Windows XP Professional, Service Pack 2.

The tests carried out using LGO concerned instances involving 100 variables and 70 constraints. Nominal orientations for the actuators (based on symmetry considerations) were adopted as an initial guess. The solution quality resulted, on the whole, comparable to that obtained, for the same instances, by CPLEX with the DRTD model. The execution time requested was in the range of 7000–10,000 seconds (s), i.e., in about 2–3 h. An instance, involving up to 168 variables and 116 constraints, was solved in 14,000 s (i.e., about 4 h), but this dimension is plausibly very close to the tractability limit to date. (An advanced tuning of the LGO parameters is expected to significantly improve the solver performances; non-negligible enhancements are also foreseen, related to the choice of the initial guess.)

The DRTD model allowed for the handling of larger P_R instances (in terms of subset $\underline{I}$), with respect to those solved directly by the NLP optimizer; future research should be dedicated to a combined approach, by utilizing the DRTD model to generate the initial solutions. Some DRTD test results are outlined in Table 1, as illustrative examples. The instance dimensional aspects are there summarized. We report the relevant data with reference to the reduced model, as obtained after the execution of the CPLEX MIP pre-processing phase, as well as the computational

Table 1 Illustrative DRTD tests

Tests	Rows	Columns	0–1 variables	First sol. gap %	First sol. execution time (s)	Best sol. gap %	Best sol. execution time (s)
1_{DRTD}	6033	5905	108	17.39	8	17.39	8
$T2_{DRTD}$	6668	6542	116	22.82	7	14.37	7
$T3_{DRTD}$	6920	6794	116	41.35	15	11.95	104
$T4_{DRTD}$	7228	7102	118	29.23	5	13.12	369
$T5_{DRTD}$	5761	6129	248	48.43	227	23.25	1573
$T6_{DRTD}$	21,268	21,128	248	41.16	110	18.55	2116
$T7_{DRTD}$	35,260	35,048	248	48.98	227	23.21	3277
$T8_{DRTD}$	11,244	11,154	279	39.75	72	25.77	975
$T9_{DRTD}$	11,679	11,589	279	36.06	1311	19.26	1490
$T10_{DRTD}$	12,114	12,024	279	49.51	39	30.06	86

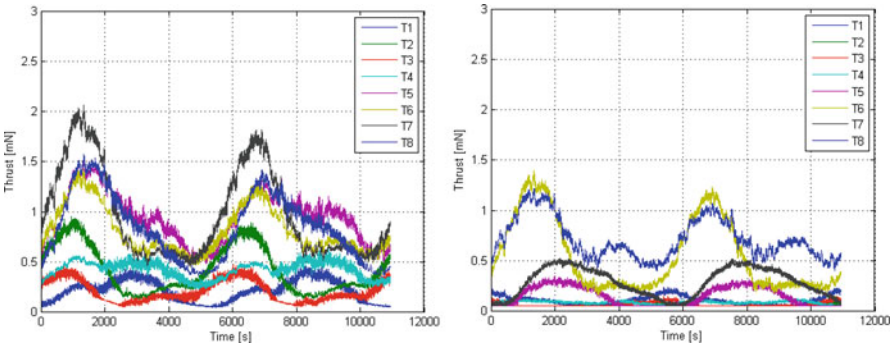


Figure 1 A real-world case study in space engineering

performance, attained with the imposition of a 1-h limit for each test, showing the time spent to reach the first and the best solutions found, respectively. The CPLEX gaps relevant to the first integer solution found and the last one, within the maximum allowed time, are further reported (these gaps are computed as: (best integer – best node)/(1e–10 + abs(best integer)) and expressed as percentages, see IBM, 2010).

Figure 1 is introduced in order to provide some hints concerning the real-world applicability of the overall methodology presented here. It relates to a case study arising in space engineering, relevant to a challenging gravimetry mission [1]. Two solutions, obtained for the same operational scenario, have been considered and are represented by the pictures on the left and right, respectively. Both report the force (thrust, expressed in millinewtons, mN) exerted by each of the eight actuators (thrusters) involved, during the given time span expressed in seconds. In the left-hand side figure, a symmetrical configuration of the thrusters has been imposed a priori, and the solution has been obtained through a dedicated simulator, cf. [1]. The solution on the right-hand side has been obtained by the joint use of the DRTD and OTD models. As shown by Figure 1, for the specific operational scenario

considered, the resulting non-symmetrical thruster configuration has provided a reduction of the overall propellant consumption by about 23%.

8 Concluding Remarks

This work, motivated by advanced space engineering applications, investigates the issue of dispatching the control of a dynamic system through a number of actuators. An appropriate control strategy – expressed in terms of total force and torque demand, occurring step by step during the entire time span relevant to a baseline scenario—represents the operational framework. The control distribution among the available actuators, in terms of forces requested step by step, is optimized, in compliance with an appropriate criterion such as the total energy consumption. This problem setting leads to a very challenging assignment, concerning actuator layout and utilization. Here we analyze the control dispatch of a general dynamic system, introducing a well-defined problem and provide a detailed description of the relevant mathematical models. Next, an ad hoc heuristic methodology, based on mathematical programming (NLP/GO, MILP, and LP) techniques is proposed, to attain satisfactory (albeit non-rigorous and, in most cases, sub-optimal) solutions for practice. A straightforward application of our approach to a real-world case study is discussed by Anselmi et al. [1]. Possible extensions and open problems are also highlighted, to identify future research objectives.

Acknowledgements I express my thanks to the referee of the present chapter for the constructive comments offered. I also wish to acknowledge J. Pintér for his meticulous observations and subtle suggestions providing significant enhancements, as well as J. Evans for her accurate perusal of the text.

References

1. Anselmi, A., Cesare, S., Dionisio, S., Fasano, G., Massotti, L.: Control propellant minimization for the next generation gravity mission. In: Fasano, G., Pintér, J.D. (eds.) Modeling and Optimization in Space Engineering – State of the Art and New Challenges. Springer, New York (2019)
2. Cenzer, D., Remmel, J.B.: Complexity, decidability and completeness. J. Symb. Log. **71**(2), 399–424 (2006)
3. Dasgupta, S., Papadimitriou, C., Vazirani, U.V.: Algorithms. McGraw Hill, Boston (2008)
4. Goldreich, O.: Computational Complexity: A Conceptual Perspective. Cambridge University Press, Cambridge (2008)
5. Harel, D., Feldman, Y.: Algorithmics: The Spirit of Computing. Addison-Wesley, Reading (2004)
6. Rudich, S., Wigderson, A. (eds.): Computational Complexity Theory. American Mathematical Society, IAS/Park City (2004)

7. Floudas, C.A., Akrotirianakis, I.G., Caratzoulas, S., Meyer, C.A., Kallrath, J.: Global optimization in the 21st century: advances and challenges for problems with nonlinear dynamics. Comput. Chem. Eng. **29**, 1185–1202 (2005)
8. Floudas, C.A., Pardalos, P.M. (eds.): Encyclopedia of Optimization. Kluwer Academic Publishers, Dordrecht (2001)
9. Horst, R., Pardalos, P.M. (eds.): Handbook of Global Optimization, vol. 1. Kluwer Academic Publishers, Dordrecht (1995)
10. Liberti, L., Maculan, N. (eds.): Global Optimization: From Theory to Implementation. Springer Science + Business Media, New York (2005)
11. Pardalos, P.M., Romeijn, H.E. (eds.): Handbook of Global Optimization, vol. 2. Kluwer Academic Publishers, Dordrecht (2002)
12. Pintér, J.D.: Global Optimization in Action. Kluwer Academic Publishers, Dordrecht (1996)
13. Pintér, J.D. (ed.): Global Optimization: Scientific and Engineering Case Studies. Springer Science + Business Media, New York (2006)
14. Bertsekas, D.P.: Nonlinear Programing, 2nd edn. Athena Scientific, Cambridge (1999)
15. Bonnans, F.J., Gilbert, C.J., Lemaréchal, C., Sagastizábal, C.A.: Numerical Optimization: Theoretical and Practical Aspects, 2nd edn. Springer, Berlin (2006)
16. Brinkhuis, J., Tikhomirov, V.: Optimization: Insights and Applications. Princeton University Press, Princeton (2005)
17. Hillier, F.S., Lieberman, G.J.: Introduction to Operations Research. McGraw-Hill, New York (2001)
18. Mordecai, A.: Nonlinear Programming: Analysis and Methods, 3rd edn. Springer, New York (2008)
19. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (1999)
20. Ruszczyński, A.: Nonlinear Optimization. Princeton University Press, Princeton (2006)
21. Chen, D.S., Batson, R.G., Dang, Y.: Applied Integer Programming: Modeling and Solution. Wiley, Hoboken (2011)
22. Karlof, J.K.: Integer Programming: Theory and Practice. CRC Press, Boca Raton (2006)
23. Jünger, M., Liebling, T.M., Naddef, D., Nemhauser, G., Pulleyblank, W.R., Reinelt, G., Rinaldi, G., Wolsey, L.A. (eds.): 50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art. Springer, Heidelberg (2009)
24. Minoux, M.: Mathematical Programming: Theory and Algorithms. Wiley, Paris (1986)
25. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley, New York (1988)
26. Papadimitriou, C.H., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Dover, Mineola (1998)
27. Schrijver, A.: Theory of Linear and Integer Programming. Wiley, New York (1998)
28. Cottle, R.W. (ed.): The Basic George B. Dantzig. Stanford Business Books/Stanford University Press, Stanford (2003)
29. Dantzig, G.B., Thapa, M.N.: Linear Programming 1: Introduction. Springer, New York (1997)
30. Dantzig, G.B., Thapa, M.N.: Linear Programming 2: Theory and Extensions. Springer, New York (2003)
31. Gartner, B., Matoušek, J.: Understanding and Using Linear Programming. Springer, Berlin (2006)
32. Padberg, M.: Linear Optimization and Extensions, 2nd edn. Springer, New York (1999)
33. Roos, C., Terlaky, T., Vial, J.: Interior Point Methods for Linear Optimization, 2nd edn. Springer, New York (2006)
34. Vanderbei, R.J.: Linear Programming: Foundations and Extensions. Springer, New York (2001)
35. Floudas, C.A.: Deterministic Global Optimization: Theory, Methods and Applications. Springer, New York (2000)
36. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, New York (2006)
37. Katanforoush, A., Shahshahani, M.: Distributing points on the sphere, I. Exp. Math. **12**(2), 199–209 (2003)

38. Williams, H.P.: Model Building in Mathematical Programming, 5th edn. Wiley, Hoboken (2013)
39. Ben-Tal, A., Nemirovski, A.: On polyhedral approximations of the second-order cone. Math. Oper. Res. **26**(2), 193–205 (2001)
40. Yuan, Z., Moreno, L.A., Fügenschuh, A., Kaier, A., Mollaysa, A., Schlobach, S.: Mixed integer second-order cone programming for the horizontal and vertical free-flight planning problem. Applied Mathematics and Optimization Series AMOS# 21, Hamburg, Germany, 2015
41. Bindel, D., Schlotterer, M, Theil, S.: Thruster actuation algorithms for scientific space missions. In: 56th International Astronautical Congress - Proceedings IAC-05-C1.P.10, Fukuoka, Japan, 2005
42. Vankov, A., Voloshinov, V.: Spacecraft thruster management subsystem design and analysis software tool. Tech-Report D3-TMF-ABS-1 1, D3 Group, GmbH, Berlin, Germany, 2003
43. Ben-Israel, A., Greville, T.N.: Generalized Inverses, New York (2003)
44. Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. Johns Hopkins, Baltimore (1996)
45. Stoer, J., Bulirsch, R.: Introduction to Numerical Analysis, 3rd edn. Springer, New York (2002)
46. Pintér, J.D.: LGO - a program system for continuous and Lipschitz optimization. In: Bomze, I.M., Csendes, T., Horst, R., Pardalos, P.M. (eds.) Developments in Global Optimization, pp. 183–197. Kluwer Academic Publishers, Dordrecht (1997)
47. Pintér, J.D.: Software development for global optimization. In: Pardalos, P.M., Coleman, T.F. (eds.) Global Optimization: Methods and Applications. American Mathematical Society, Providence (2009)
48. Pintér, J.D.: LGO – A Model Development and Solver System for Global-Local Nonlinear Optimization. User's Guide. PCS Inc., North York (2015)
49. IBM Corporation: ILOG CPLEX Optimizer. High performance mathematical optimization engines. IBM Corporation Software Group, NY 10589 U.S.A. WSD14044-USEN-01 (2010)

Choice of the Optimal Launch Date for Interplanetary Missions

Giancarlo Genta and P. Federica Maffione

Abstract Interplanetary missions are strictly dependent on the launch date. Mission planning requires a knowledge not only of the technological and cost constraints but also the study of the influence of the launch opportunity on the spacecraft performance and on the feasibility of the mission. Pork-chop plots are effective tools to design interplanetary missions, providing a graphical overview of the relationship between the fundamental parameters of the mission design, namely, the launch date, the duration and the energy requirements. In this way it is possible to evaluate the best timing to accomplish the mission under current constraints. Plots of a similar type can be drawn also for optimizing missions based on low thrust or on propellantless propulsion—like solar sails. The cost function described in these plots may be the square of the hyperbolic excess speed, the ΔV, the ratio between the initial mass and the payload mass or between the mass of the propulsion system and the payload or the cost function J used in the optimization of the trajectory. In case of two-way missions, it is possible to plot a cost function of the same type, by adding the values related to the forward and the backward journeys, as a function of the duration of the two legs of the travel, once the stay on the planet has been fixed. In this way it is possible to optimize also missions which are intrinsically two ways, like human exploration missions or sample return missions.

Abbreviations

a	Thrust/mass ratio
C_3	Square of V_∞
I, J	Cost function
I_s	Specific impulse
K	Tankage factor

G. Genta (✉) · P. F. Maffione
Politecnico di Torino, Torino, Italy
e-mail: giancarlo.genta@polito.it

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_5

m	Mass of the spacecraft
m_f	Final mass
m_i	Initial mass
m_L	Payload mass
m_p	Propellant mass
m_{ps}	Mass of the propulsion system
m_s	Structural mass
m_t	Mass of the thruster
m_{tank}	Mass of the tanks
m_w	Mass of the generator
$\mathbf{r}$	Vector defining the position of the spacecraft
R_E	Radius of the orbit of Earth
T	Spaceflight duration
$\mathbf{T}$	Thrust
t	Time
T_a	Arrival date
T_p	Time spent on the planet
T_s	Starting date
V_∞	Hyperbolic excess speed
α	Mass/power ratio of the power generator
γ	Optimization parameter
ΔV	Velocity increment
ε	Earth axis obliquity
η	Efficiency
CEV	Constant Exhaust Velocity
ICRF	International Celestial Reference Frame
IMLEO	Initial Mass in Low Earth Orbit
LEO	Low Earth Orbit
NEP	Nuclear Electric Propulsion
NTP	Nuclear Thermal Propulsion
SEP	Solar Electric Propulsion
VEV	Variable Exhaust Velocity
VASIMR®	Variable Specific Impulse Magnetoplasma Rocket

Subscripts

i	Inbound
o	Outbound

1 Introduction

The planets of our Solar System move in a complex way around the Sun, and this causes the performances of any interplanetary spacecraft to depend on the launch opportunity and, once the latter has been chosen, on the exact launch date.

The precision required in the computation of the spacecraft trajectory is such that the approximation obtained assuming that the planets move along circular and even elliptical orbits is not sufficient for the analysis of an interplanetary mission, and the knowledge of detailed ephemerides is required. While historically ephemerides consisted in tables where the positions of the celestial bodies of interest were reported as functions of time, modern ephemerides are based on mathematical models allowing to compute the position of the solar system bodies at any given time. Precise reference frames and time reference must then be stated: at present the ephemerides made available by NASA-JPL (https://ssd.jpl.nasa.gov/?ephemerides, https://naif.jpl.nasa.gov/pub/naif/toolkit_docs/C/req/frames.html#Selecting a Name) are based on the International Celestial Reference Frame (ICRF) and on the Julian time.

The ICRF is an inertial reference frame centred in the centre of mass of the solar system and whose axes are fixed to a set of 295 extra-galactic radio sources. The axes directions are assumed to be very close to that of the axes of the Earth-fixed J2000 frame, and so its x-axis points in the direction of the vernal equinox and the y-axis is contained in the Earth equatorial plane in its position at 12.00 of January 1st, 2000. For solar system navigation it is perhaps better to rotate the reference frame about its x-axis of the obliquity angle ε of the Earth axis, so that the x–y plane coincides with the ecliptic plane. An Earth-fixed reference frame with the axes so oriented is the ECLIPJ2000 frame.

The solar system is particularly simple, with respect to the many star systems in which the central star is multiple, in that the mass of the Sun is much greater, by orders of magnitude, than that of all other bodies. As a consequence, the trajectories of all bodies, natural and artificial, are very close to conic sections and, in particular, to ellipses.

If the spacecraft is propelled by a *high thrust* or *impulsive* system, i.e. a thruster which supplies a thrust much higher than all other forces acting on the spacecraft for a time which is much shorter that the travel time, the *patched conics* approach allows to compute the trajectory with a good approximation. The thrust is assumed to be similar to a Dirac's delta, i.e. an infinitely large force applied for an infinitesimal time, so that it results in an instant velocity variation ΔV and the trajectory is determined by the gravitational attractions only (*coasting arcs*). Moreover, each coasting arc is approximated by an arc of a conical section determined by the attraction of a single celestial body, that in whose *sphere of influence* the spacecraft lies, i.e. that exerting the largest gravitational acceleration. In this case, a spacecraft leaving planet A and aimed to planet B moves along an hyperbolic trajectory about the starting planet until it gets out of its sphere of influence, then follows

an heliocentric elliptical (or hyperbolic) trajectory until it enters the sphere of influence of the destination planet and finally follows a hyperbolic trajectory centred in planet B.

The thrust bursts are best applied at the beginning and at the end of the journey (in the latter case aerodynamic forces due to the atmosphere of the destination planet can be used), but usually the conic arcs are interrupted by (usually small) velocity variations aimed at correcting the overall trajectory. More complex trajectories can be split in a larger number of arcs, for instance, when a gravity assist manoeuvre is performed in the sphere of influence of another planet.

In case of low-thrust systems, i.e. thrusters which provide a thrust which is not much higher than the other forces acting on the spacecraft and is applied for long times, possibly up to the whole travel time, the same scheme can be applied by subdividing the trajectory in arcs performed in the sphere of influence of a single body. The propelled arcs are, in this case, no more parts of conical sections. Usually geometrically they look like arcs of spirals, which are neither logarithmic nor Archimedes spirals, and cannot be computed in closed form, so that the numerical integration of the trajectory becomes mandatory.

This *patched arcs* approach is thus the simplest, although approximate, way of computing the trajectory of a spacecraft in the solar system.[1] The present chapter will be based on this approach.

2 Choice of the *Best* Travel Dates

The problem of choosing the best start and arrival date for an interplanetary mission requires that a parameter, namely, a *cost function I* is stated. This parameter must be minimized to find the start and the arrival dates (T_s and T_a) or, which is the same, the start date and the travel duration (T_s and T) yielding the minimum value of I. Depending on the propulsion type and the mission goals, different formulations of the cost function I can be chosen. For instance, in case of an interplanetary travel starting from Low Earth Orbit (LEO) with the aim of bringing cargo to a planet, say Mars, the cost function may be the Initial Mass in Earth Orbit (IMLEO), which is strictly linked with the cost of the mission. In case of a human mission to the same planet, the IMLEO is not the only important parameter, since the duration T of the travel is in itself a parameter to be minimized and a compromise between T and the IMLEO must be looked for.

Another important difference is whether the mission under consideration is a one-way mission, e.g. launching a probe, delivering a cargo or performing a one-

[1] It must be expressly stated that this approach will be impossible when the problem of travelling in the system around a multiple star will be faced. If a mission like that studied in the Breakthrough Starshot project will aim to perform more than just a flyby in the Alpha Centaury system, even its preliminary study will need to go beyond the patched arcs assumption.

way colonization mission, or a two-way mission like a human exploration mission. In this case the parameters to be chosen are the starting and arrival dates of the outbound and inbound journeys (T_{so}, T_{ao}, T_{si} and T_{ai}) or the overall starting date, the duration of both journeys and the stay on the planet (T_{so}, T_o, T_p and T_i).

In two-way missions a distinction must be made: Is the return journey to be performed in the same or in a different launch opportunity? In the latter case, at least if the journey duration is much shorter than the synodic period of the destination planet with respect to the starting one, the outbound and the inbound journeys can be studied independently and it is not required to perform the optimization of the whole mission, while in the former case the whole mission must be optimized taking into consideration both journeys.

Consider the optimization of a one-way journey: the problem is to find the values of T_s and T_a (or of T_s and T) yielding the minimum of function $I(T_s, T_a)$ (or of function $I(T_s, T)$)—if such a minimum exists—or yielding a suitable compromise between minimizing I and minimizing T.

To solve this problem the value of I is computed for a large number of missions characterized by different values of T_s and T_a and the surface $I(T_s, T_a)$ is plotted, in particular in the form of a contour plot. From this plot it is possible to identify the dates yielding a minimum of I or, if such a minimum does not exist or leads to a value of the travel time which is unpractically long, searching for a reasonable trade-off between travel time and cost.

The fact that a minimum of the cost function may not exist, or may lead to an unacceptably long travel time, justifies the computationally heavy approach mentioned above instead of using more efficient optimization methods like those based on a gradient descent. What is essential in this case is to gain a good knowledge of the overall shape of the surface $I(T_s, T_a)$ to search for an adequate compromise between low cost function and low travel time.

The cost function can be computed for the interplanetary journey only, and in this case—if the planetary orbits are assumed to be circular and coplanar—there exists a single plot for a given pair of planets once the propulsion system has been chosen. If, on the contrary, the ellipticity and non-coplanarity of the planetary orbits is accounted for, the plot depends on the chosen launch opportunity.

Function $I(T_s, T_a)$ can be defined to take into account the whole journey, from the starting orbit around the first planet (or the planetary surface) to the arrival orbit about the second planet. In this case the plot can be obtained only after many details of the mission have been stated (starting and arrival orbits parameters, possible flybys, aerobraking manoeuvres, trajectory corrections, etc.) and the increase in accuracy is paid with a decrease in the generality of the results.

The computation of the plot is quite computationally intensive, since it requires to compute thousands (possibly hundred thousand or millions) trajectories. Each one of the trajectories may involve an optimization procedure, particularly in the case of low thrust, in which case not only the trajectory but also the thrust profile needs to be computed. The cost function may be the same cost function which will be then used to plot the final diagram. However, as it will be shown below,

even in cases in which the procedure is very computationally intensive, the choice of the launch and arrival dates can be performed in a few hours using a laptop computer.

The starting and arrival time can be expressed in terms of Julian dates or of Gregorian dates or, in a simpler way, as days after the closest opposition. This is perhaps the simplest way of stating the scales of the plot; since the mission usually starts before the opposition and arrives after, T_s and T_a are usually negative and positive, respectively.

In case of two-way travels, the two plots for the outbound and the inbound journeys $I_o(T_{so}, T_o)$ and $I_i(T_{si}, T_i)$ must be first computed. Then some relationships between the four independent variables have to be introduced, for instance, that yielding to an arbitrarily assumed time spent on the planet T_p and to a minimum value of the sum $I_o + I_i$ at fixed values of T_o and T_i and then the plot $(I_o + I_i)(T_o, T_i)$ is computed. From such a plot the values yielding to a minimum value of the total cost function or to the required compromise value can be identified.

3 Impulsive Thrust

The tool traditionally used to choose the starting and arrival dates for a one-way impulsive mission is the so-called *pork-chop plot*. It has been used since the 1970s, when it was introduced for the Voyager programme, in the preliminary stages of the mission design. Since Voyager required a higher accuracy compared to the previous missions as Mariner or Viking, the mission planners Roger Bourke and Charles Kohlhase needed to know exactly the position of the spacecraft at each instant. In particular, they tried to design a mission to be completed before the end of the 1981, maintaining a very high position precision in all mission phases [1]. They generated 10,000 possible trajectories from the pork-chop plot and then selected the best one that satisfied the mission objectives.

The classical pork-chop plot uses as cost function the square of the hyperbolic excess speed $C_3 = {V_\infty}^2$ as a function of T_s and T_a and thus takes into account only the energy of the trajectory when leaving the starting planet's sphere of influence, without accounting for the ΔV required to enter the arrival planet's sphere of influence or those related to the manoeuvres performed to leave the starting orbits or achieving the arrival one. This is in a way obvious because to take them into account would mean to have already taken many design decisions, like whether opting for propulsive or aerodynamic manoeuvres at arrival (in case of a planet with an atmosphere) or whether opting for a direct launch, or a perigee burn, etc.

In this regard, JPL developed MIDAS, a patched conic interplanetary trajectory optimization tool (http://trajectory.grc.nasa.gov/tools/midas.shtml). Then, using the data obtained from this programme, JPL published a Mission Design Handbook which provides the trajectory designers with contour plots for all Earth–Mars mission opportunities from 1991 to 2045 [2–5]. Once the launch window is chosen,

there are several types of pork-chop plot: the simplest one refers to ballistic trajectories without any deep space manoeuvres, the second one takes into account the addition of the deep space manoeuvres or of flybys of other planets. In this way it is possible to compare different mission architectures.

Plotting a pork-chop plot is relatively simple: the values of T_s and T_a are chosen, then from the ephemerides the positions of the starting and arrival planets are obtained and the orbital parameters of the trajectory between these two points can be computed, together with the initial and final velocity and thus the value of C_3 [6, 7].

If the planetary orbits were circular and coplanar, the surface $C_3(T_s, T_a)$ would have a single minimum corresponding to the Hohmann transfer, which in the case of Earth–Mars trajectories starts about 96 days before the opposition ($T_s \approx -96$ days) and has a duration of 258.73 days ($T_a \approx 163$ days). Since in an actual case the orbits of the planets must be considered at least elliptical and not coplanar, the pork-chop plot depends on the specific launch opportunity and usually the surface $C_3(T_s, T_a)$ has two minima, one related to type I missions (trajectories which intersect the orbit of the arrival planet the first time at the arrival time) and the other to type II missions (trajectories intersecting the orbit of the arrival planet once before the arrival time). Depending on the launch opportunity, the absolute minimum of C_3 may occur for type I or type II trajectories: for instance, in the 2035 opportunity the minimum energy required for a Earth–Mars type I trajectory ($C_3 = 10.28$ km^2/s^2) is lower than that required for type II ($C_3 = 17.78$ km^2/s^2) trajectories and the optimal transfer times are, respectively, 194 and 426 days.

The pork-chop plot is usually plotted for a limited range of starting dates around the optimal launch window, but it may be extended indefinitely to study what happens also in case of a launch well outside the optimal conditions. This may seem to have very little practical interest, but it may be useful for designing short-stay Mars missions or to deal with cases in which an advanced propulsion system—mostly of the Nuclear Thermal (NTP) type—allows to perform a mission to deliver badly needed cargo to an outpost or to allow the return of personnel from Mars in an emergency.

In case a wide range of start dates is expedient, to obtain a more compact plot, it is convenient to put the interplanetary transit time T instead of the arrival time T_a on the ordinate axis.

It is also possible to compute the total orbit-to-orbit ΔV, i.e. the sum of the velocity increments required to enter the interplanetary trajectory from the parking orbit plus that required to enter the orbit about the arrival planet from the former. In case the manoeuvres at the arrival planet are performed using the aerodynamic drag of the planetary atmosphere (aerobraking or aerocapture manoeuvres), the second ΔV is set to 0 or to a low value. Such a plot can be defined as a *ΔV-plot*.

Clearly, the pork-chop plot is much more general, since it doesn't require to state the orbits about the two planets. It is also more useful in the case of chemical propulsion, when the braking at the arrival planet is performing aerodynamically, while the ΔV-plot is required in the case of Nuclear Thermal Propulsion (NTP) when a propulsive braking manoeuvre is performed [8].

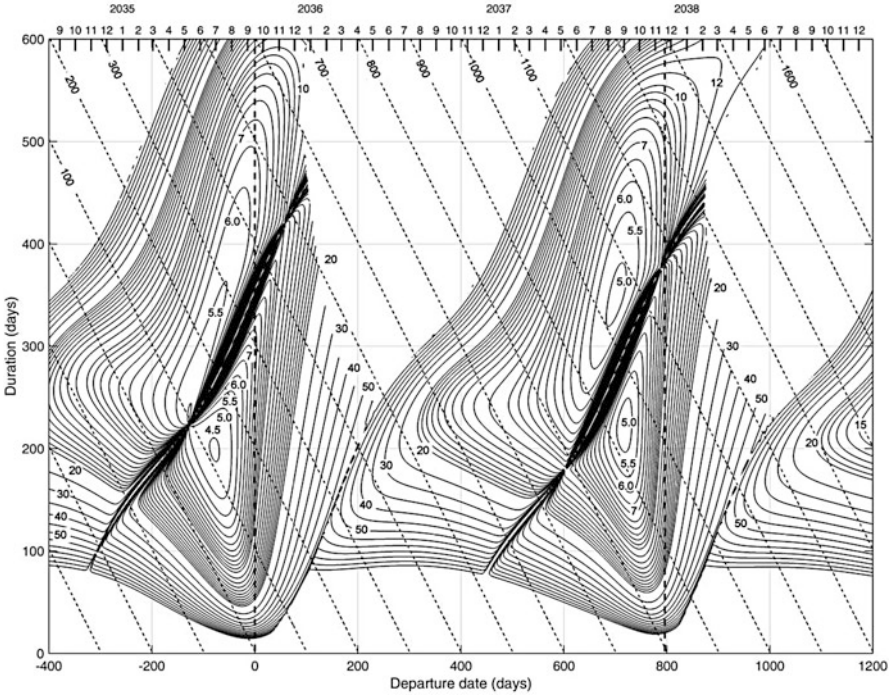


Figure 1 ΔV-plot for Earth–Mars transfers performed between mid-August 2034 and the end of December 2038. The ΔV is expressed in km/s and the time is in days

As an example, the ΔV-plot for Earth–Mars transfers performed between mid-August 2034 (400 days before the 2035 opposition) and the end of December 2038 (1200 after the same opposition) is shown in Figure 1. The departure orbit is circular and has an altitude of 900 km (a possible choice for NTP), while the arrival orbit is highly elliptical with a periareion of 400 km and an apoareion of 30,000 km.

The dotted lines are the arrival dates, expressed in days after the 2035 opposition.

The plot covers more than two complete synodic periods, and the differences between the two launch opportunities are clear. In the 2035 opportunity the optimal launch window is around 80 days before opposition and the minimum energy trajectory is of type I with a duration of about 200 days. A good choice is launching on June 27, 2035, which requires $\Delta V_1 = 3542$ m/s and $\Delta V_2 = 921$ m/s, with a total ΔV of 4468 m/s. A value of $C_3 = 10.30$ km^2/s^2 is obtained, which is lower than that required for a type II trajectory ($C_3 = 20.68$ km^2/s^2) with launch on May 5, 2035 and a total ΔV of 4977 m/s. It must be stated that a lower C_3 could be obtained for a longer type II trajectory (down to $C_3 = 17.78$ km^2/s^2), but in this case a higher total ΔV was obtained. In fact, the spacecraft would have a higher hyperbolic excess speed arriving to Mars, requiring much more energy to brake, unless aerobraking is used.

In the following launch opportunity, type II trajectories would be more convenient, particularly for what C_3 is concerned. If different starting or arrival orbits were chosen, or a non-propulsive manoeuvre were performed at arrival, the plot would have been different.

The plot was obtained using a purposely written MATLAB code, based on the Newton Raphson method [9], solving the Gauss problem and the ephemeris data downloaded from the Jet Propulsion Laboratory's online resources (http://ssd.jpl.nasa.gov/?planet_pos). The total number of trajectories computed is 240,000.

4 Low Thrust

The case of low thrust is increasingly more important, since a number of robotic planetary missions are propelled by electric thrusters—mostly ion thrusters—[10, 11] powered by solar arrays (Solar Electric Propulsion, SEP), often in conjunction with gravity assist manoeuvres, and in the future also Nuclear Electric Propulsion is much promising [12]. Since the early studies of human Mars missions, the possibility of using electric propulsion (mostly NEP, but also SEP) was considered [13, 14]. In this case the time spent in space becomes a very important parameter of the mission, and the use of gravity assist manoeuvres is seldom considered, except for performing missions which would be impossible without exploiting the gravitational field of a planet different from the departure or arrival planets (in case of Mars missions, the Venus' gravitational field).

Owing to the rising number of interplanetary mission planned for low-thrust electric propulsion and the importance of the travel time, there is a need for a plot similar to the pork-chop or the ΔV-plot, but for low, continuous thrust, trajectories which can be applied to NEP and SEP spacecrafts. A plot of this kind, usually referred to as the bacon plot, has recently been introduced [15]. In this case the useful mass carried to the destination planet $m_L + m_s$, obviously at equal initial mass m_i or at equal IMLEO, is assumed as a cost function to be maximized instead of minimized. Better, what is to be maximized is the ratio $(mL + m_s)/m_i$. To use a true cost function to be minimized, its reciprocal can be used: $I = m_i/(m_L + m_s)$.

The first difference between the pork-chop plot and the bacon plot is related to the mathematical approach used to compute them. While in case of impulsive mission the patched conics approach can be used and the trajectory can be obtained in closed form, in case of low-thrust mission the trajectory must be computed together with the thrust profile and different types of optimization techniques, both direct or indirect, can be used [16]. Many programmes aimed to compute the trajectory and the thrust profile of low-thrust missions are available.

The second difference is that in case of low thrust the contours are not closed, i.e. the cost function I has no minima and the longer is the travel time, the lower is the value of the cost function.

In case of low-thrust mission, it is also necessary to distinguish between Variable Ejection Velocity (VEV) and Constant Ejection Velocity (CEV) systems [17–20]. The simplest and most convenient approach is assuming that the system operates

at constant power (in the case of NEP) or at the maximum power allowed by the solar arrays (in the case of SEP) [21–24] and when the computed thrust profile requires a reduction of the thrust, this is obtained by increasing the specific impulse. This approach is described in detail in [25, 26] and leads to much less computationally intensive algorithms. However, it has a basic limitation: the maximum value of the specific impulse which is required, in particular in the middle of the interplanetary cruise, is much higher than the maximum value the thruster can actually reach, even if particularly suitable plasma thrusters, like the VASIMR® [12] (http://www.adastrarocket.com/aarc/VASIMR, http://www.nasa.gov/home/hqnews/2008/dec/HQ_08-332_VASIMR_engine.html), are used.

When the thrust must be reduced below the value corresponding to the maximum specific impulse, the simplest action is to keep the latter at the maximum possible value and to reduce the power, but this leads to a non-optimal control law and then to an increase of the cost function. A much better alternative is maintaining the power at its maximum possible value when the thruster is on, and switching off the thruster, letting the spacecraft to proceed with its engine off, for a certain time. The engine on-off times are among the parameters to be optimized, together with the thrust direction and intensity and the specific impulse.

The basic equations used for the optimization of the trajectory and of the thrust profile and for the optimal spacecraft mass breakdown are summarized in Appendix.

It must be expressly stated that the cost function J as defined by Eq. (11) or Eq. (27) has a meaning only in the case of a thruster with unlimited specific impulse. If on the contrary there are limitations on the maximum specific impulse of the thruster, a different definition of J is required (Eq. 30).

Since it is possible to compute the value of J once that the starting and arrival dates are stated, it is possible to plot the surface $J(T_s,T_a)$ or $J(T_s,T)$ also in this case. The dependence of the energy requirements of the interplanetary flight on the starting and arrival dates (or on the starting date and the travel time) can thus be summarized by a single contour plot, which is here indicated as *J-plot*.

The J-plot can be obtained using both direct and indirect optimization methods. Of course also stochastic methods can be used but, since a large number of trajectories must be computed and the problem is a constrained one, they become too computationally intensive and for this reason are not considered here. Here a Matlab tool developed by the authors and called IRMA (InteRplanetary Mission Analysis) and based on the approach proposed by Keaton in [25] is used. The boundary value problem is solved by using the BVP5C [27] Matlab routine based on the four-stage Lobatto IIIa formula and implemented as an implicit Runge–Kutta formula.

If direct methods are considered, the optimal control software Falcon.m, developed at the Institute of Flight System Dynamics of Munich, is used [28]. The problem is converted into a nonlinear programming problem using the trapezoidal transcription method and then solved using the numerical solver IPOPT (Interior Point OPTimizer) [29]. A more detailed discussion of the solution techniques can be found in [30, 31].

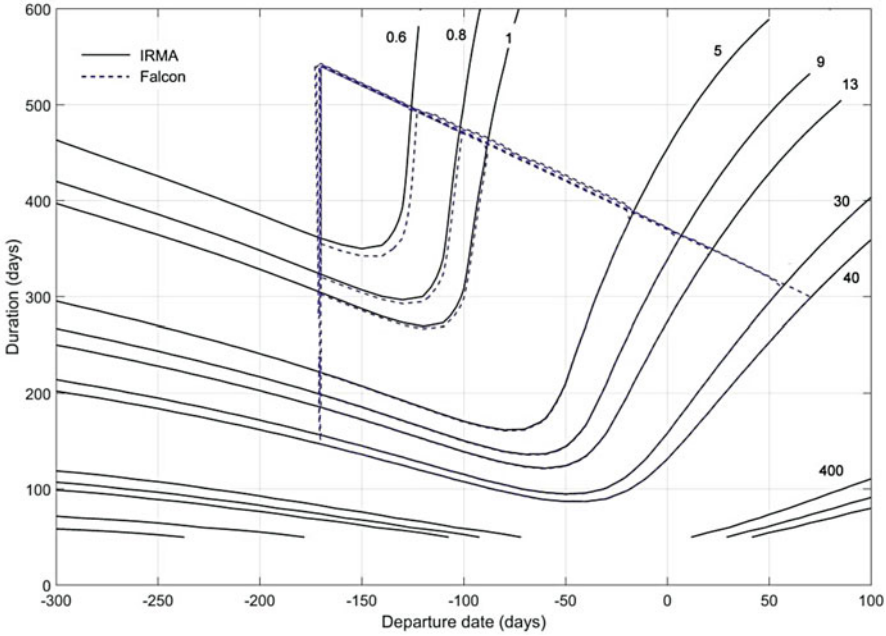


Figure 2 Ideal J-plot for NEP for Earth–Mars travels in the launch opportunity of 2035. J is expressed in m^2/s^3. Computation performed both using the IRMA and the FALCON.m codes. Note that in the zone included in the area between the two dashed lines, where the two codes give similar results, the lines for $J = 5$, 9, 13, 30 and 40 m^2/s^3 are completely superimposed, while those for $J = 0.6$, 0.8, 1 m^2/s^3 are very close to each other

A J-plot is computed using both IRMA and Falcon.m codes for Earth–Mars travels in the launch opportunity of 2035, with $\alpha = 10$ kg/kW, and maximum specific impulse $I_{smax} = 6000$ s is reported in Figure 2. The plot has been obtained by using the ephemerides developed by JPL (https://naif.jpl.nasa.gov/pub/naif/toolkit_docs/C/req/frames.html#Selecting a Name). The computation time is 1631 s for IRMA and 3988 s for Falcon.m. The two plots are almost coincident near the optimal launch window but, away from this region, FALCON.m finds local minima that are far from the global minimum (as the results from IRMA suggest). It would be necessary to optimize again the trajectory in these points, leading to a severe increase of computational time. It can be concluded that in this case the indirect method is more efficient than the direct one.

Also in the case of low thrust it is possible to obtain an orbit-to-orbit plot. However, while in the case of impulsive thrust, the duration of the first and the last phase, performed within the spheres of influence of the start and destination planets, can be neglected, in the present case they can take a substantial fraction of the total time. The relative durations of the three phases constitute a set of parameters which are essential in the overall optimization of each mission.

Consider the case in which the cost function J, defined either by Eq. (11), Eq. (28) or Eq. (31), is used. Let the durations of the three phases be T_1, T_2 and T_3, and the values of the parameter J be J_1, J_2 and J_3. The total duration and cost function are thus [26, 32]

$$T_{tot} = T_1 + T_2 + T_3, \quad J_{tot} = J_1 + J_2 + J_3. \tag{1}$$

Once the starting orbit, which may be either circular or elliptical, has been defined, it is possible to compute a function $J_1 = J_1(T_1)$, at least in an approximate way, by approximate algorithms (for circular orbits) or numerical integration (for elliptical orbits). If the spacecraft is powered by a solar array, the interruption of the thrust in the shadow of the planet can be accounted for [24]. In the same way also function $J_3 = J_3(T_3)$ is easily obtained.

The J-plot for the interplanetary cruise gives the function $J_2 = J_2\ (T_2,T_{s2})$. Since

$$T_{s2} = T_s + T_1, \tag{2}$$

for each value of T_s and T_{tot} it is possible to obtain a function $J_{tot}\ (T_1,T_3)$ and then the values of the first and third phases leading to the minimum value of J_{tot}. Repeating this computation for all the values of T_s and T_{tot}, a plot of the function $J_{tot}\ (T_s, T_{tot})$, namely, an orbit-to-orbit J-plot, is finally obtained.

Such a plot for a NEP spacecraft travelling from Earth to Mars in the 2035 and 2037 launch opportunities, characterized by $\alpha = 10$ kg/kW and a maximum specific impulse $I_{smax} = 6000$ s, is shown in Figure 3. The starting and arrival orbits are the same as those considered in Figure 1.

5 Propellantless Propulsion

The possibility of travelling in space without the need of propellant is extremely interesting. At present the only devices which allow to produce a thrust without the need of expelling a propellant are solar or magnetic sails. They produce very small thrust, but one which may be adequate for a number of robotic missions. With further technological improvements aimed to build very low mass sails, there is the possibility of extending their applicability to other types of missions.

Apart from solar sails, propelled by solar light, there is the possibility of using laser sails[2] and magnetic sails, propelled by solar wind. Still hypothetical (and much controversial) is the possibility of future propellantless propulsion systems like those studied in the NASA Breakthrough Propulsion Physics (BPP) project [34].

In all these systems it is possible to define a specific thrust, i.e. a ratio between the thrust supplied by the propulsion system and its total mass. In the case of solar

[2]Laser sails have been proposed also for interstellar journeys (see, for instance, [33]) and are at the base of the Breakthrough Starshot project (https://breakthroughinitiatives.org/initiative/3). Solar and laser sails are collectively referred to as light sails.

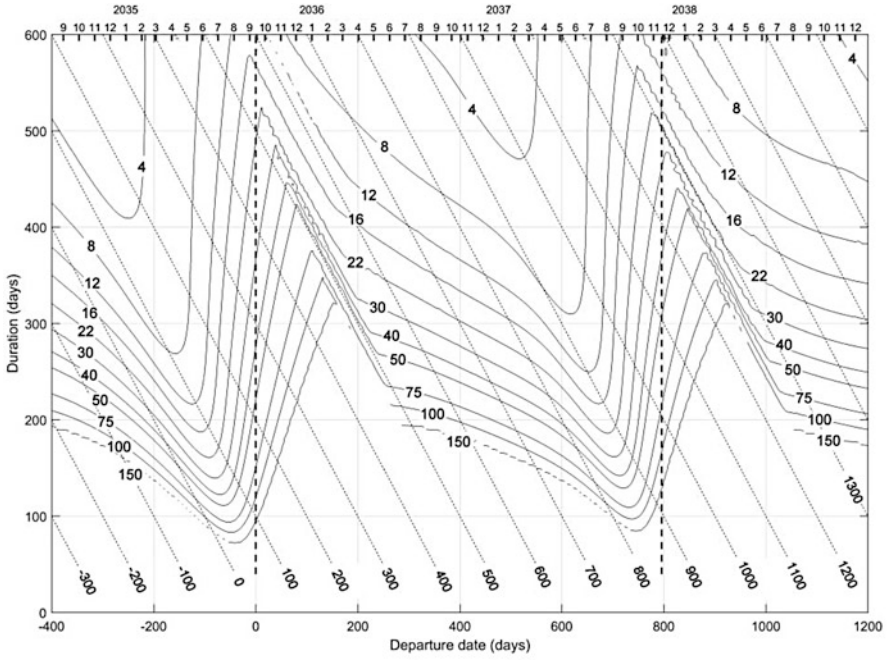


Figure 3 Orbit-to-orbit J-plot for a NEP spacecraft travelling from Earth to Mars in the 2035 and 2037 launch opportunities. $\alpha = 10$ kg/kW, maximum specific impulse $I_{smax} = 6000$ s. The starting and arrival orbits are the same as those considered in Figure 1

sails the specific thrust decreases with the square of the distance from the Sun. In case of sails of all types, the value of the thrust depends of the orientation of the sail, reducing to zero when the sail is in a plane containing the light source. Each trajectory, characterized by a starting and an arrival date, can be optimized by searching the direction of the thrust which minimizes the total mass of the propulsion system (or better the ratio between the propulsion system and the total mass of the spacecraft m_{ps}/m_i) [23].

A plot of $I = m_{ps}/m_i$ (T_s, T_a) can thus be obtained also in the case of propellantless systems, although this plot does not yet have a name.

6 Two-Way Journeys

As already stated, a two-way mission can be optimized as a whole; in particular, this is the case for human Mars missions when the two journeys are performed in the same launch opportunity (short-stay missions). Also in this case the optimal dates can be obtained by plotting a suitable contour plot.

In case of impulsive thrust, this plot is a *two-way ΔV-plot*.

The total ΔV is a function of four parameters, either the starting and arrival dates of both journeys (T_{so}, T_{ao}, T_{si} and T_a) or the starting date of the outbound journey (T_{so}), the duration of the inbound and outbound travels (T_i, T_o) and the stay on the planet (T_p):

$$\Delta V_{mission} = \Delta V_{mission}\,(T_{so}+T_{ao}+T_{si}+T_{ai}) = \Delta V_{mission}\left(T_{so} + T_o + T_i + T_p\right). \tag{3}$$

A way to optimize an impulsive mission with a given value of the time spent on the destination planet T_p is to state a range for the duration of the outbound and the inbound journeys T_o and T_i and to search the value of the start time for the outbound T_s yielding the minimum value of the total ΔV for each pair T_o, T_i. Proceeding in this way it is possible to obtain a contour plot in the plane (T_o, T_i): for instance, using the orbit-to-orbit ΔV-plots for the Earth–Mars and Mars–Earth journeys in the 2035 launch opportunity, from an 800 km circular LEO to an elliptical Mars orbit with a 320 km periaerion and a 35,000 km apoareion, and assuming a stay on Mars $T_p = 40$ days, the plot of Figure 4 has been obtained.

The surface has an absolute minimum (A) obtained with two type II trajectories, while minimum B is obtained with a type I trajectory outbound and a type II

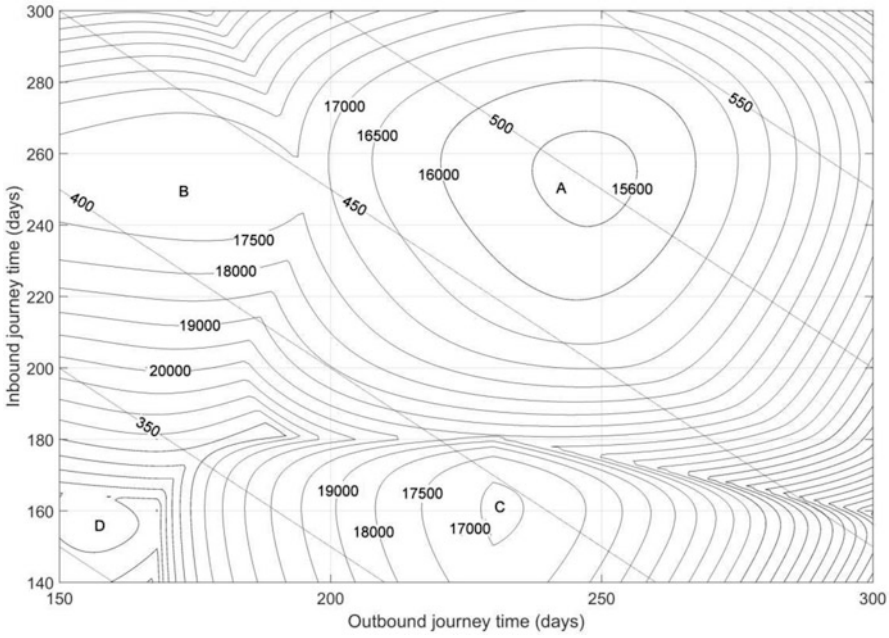


Figure 4 Contour plot of the total ΔV for a two-way Earth–Mars travel with a stay on Mars of 40 days performed in the 2035 launch opportunity

Table 1 Three near-optimum short-stay missions. The first is close to the absolute minimum, while the second is made by a short outbound journey followed by a much longer return; the opposite for the third. The last two lines deal with a mission contemplating a Venus flyby in the return journey. Times are in days, referred to the opposition date

		ΔV (m/s)	Starting date	Arrival date	Travel time	Total	ΔV_{miss} (m/s)	Type
Min. A	Outbound	7700	−260	−34	236	486	15,842	II
	Inbound	8142	6	266	250			II
Min. B	Outbound	5439	−122	38	160	410	17,172	I
	Inbound	11,733	78	328	250			II
Min. C	Outbound	9307	−296	−70	230	386	17,569	II
	Inbound	8262	−30	130	156			I
	Outbound	6425	−125	25	150	420	15,342	I
	Inbound	8917	65	335	270			Flyby

inbound; minimum C is obtained with a type II trajectory outbound and a type I inbound and minimum D with two type I trajectories.

Three cases of near-optimum short-stay missions (the D minimum is flat enough to make it immaterial to search for the exact minimum) are reported in detail in Table 1. Inserting a Venus flyby in the return leg of the journey allows to either reducing the total time spent in space or to reduce the total ΔV of the mission. The total ΔV is lower than those for mixed type I and type II missions, or the total time in space is much smaller than that of a mission contemplating both type II trajectories.

Things are different in case of low-thrust missions. From the two J-plots for forward and backward journeys, it is possible to obtain the two-way plot for the whole mission which does not contain closed lines. Such a plot for a stay on Mars of 40 days and the same conditions of Figure 2 are shown in Figure 5. There is no optimal mission, in the sense of a minimum energy mission; however, it is possible to identify an outbound and an inbound journey which give the minimum value of the sum of parameter J of the two-way journeys for a given total time in space.

Such conditions are identified by the points in which the lines for constant total time in space (parallel to the bisector of the second quadrant—if the scales are equal) are tangent to the contour lines.

For very short journeys there is only one of such points per contour line; accepting longer journeys there are two of such conditions, with the most favourable one being that with a longer outbound travel, then the best one is that with the longest travel in the inbound leg of the journey, and finally there is only one point of tangency. The two solutions for a total time in space of 400 days are shown in Table 2 (the most favorable one is that with a short outbound and long inbound travel).

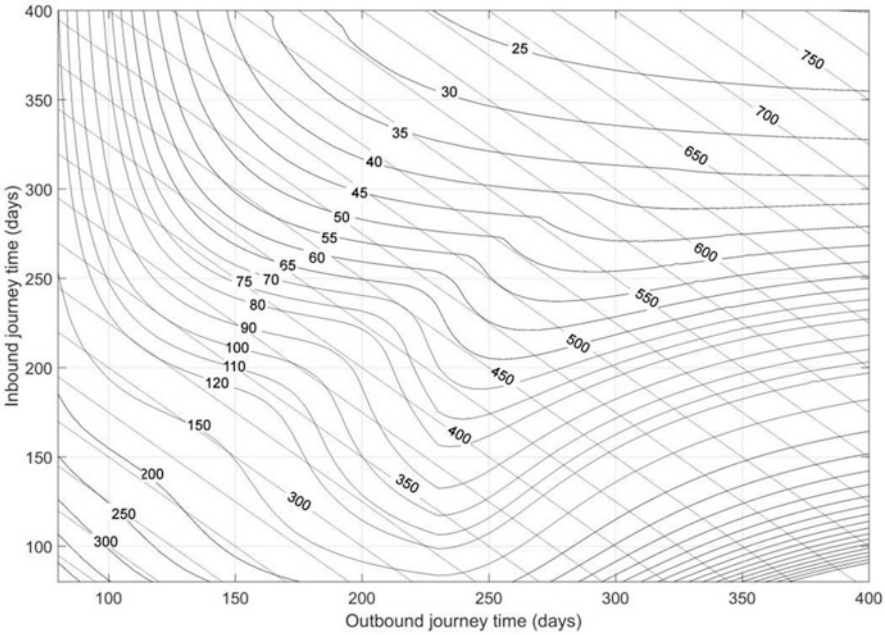


Figure 5 Contour plot of the surface $J_{miss}(T_o,T_i)$, for a two-way Earth–Mars travel (2037 launch opportunity) obtained from the J-plots of Figure 3. The values of J are expressed in m^2/s^3

Table 2 Two low-thrust short-stay (40 days) missions. The first has a slow outbound travel and a fast return journey, while the second the opposite. Time in days after the opposition

	J (m^2/s^3)	Starting date	Arrival date	Travel time	Total time
Outbound	36.802	−300	−60	240	400
Inbound	41.948	−20	140	160	
Outbound	29.621	−90	50	140	400
Inbound	42.760	90	350	260	

7 Examples

7.1 NTP Mission to Titan, Possibly with a Jupiter Flyby

Titan, the satellite of Saturn, is a very interesting destination and sending there a lander, with a rover [35] perhaps able to also sail in the methane lakes, is a priority. Reaching Saturn is not easy, and nuclear thermal (or electric) propulsion is a possibility—like it was for the Jupiter moons (JIMO) mission considered in the past and then cancelled. Reaching Saturn can be helped by performing a flyby of Jupiter, and the 2039 launch is a good choice for this type of trajectory. The example is performed using the following data: Trajectory data:

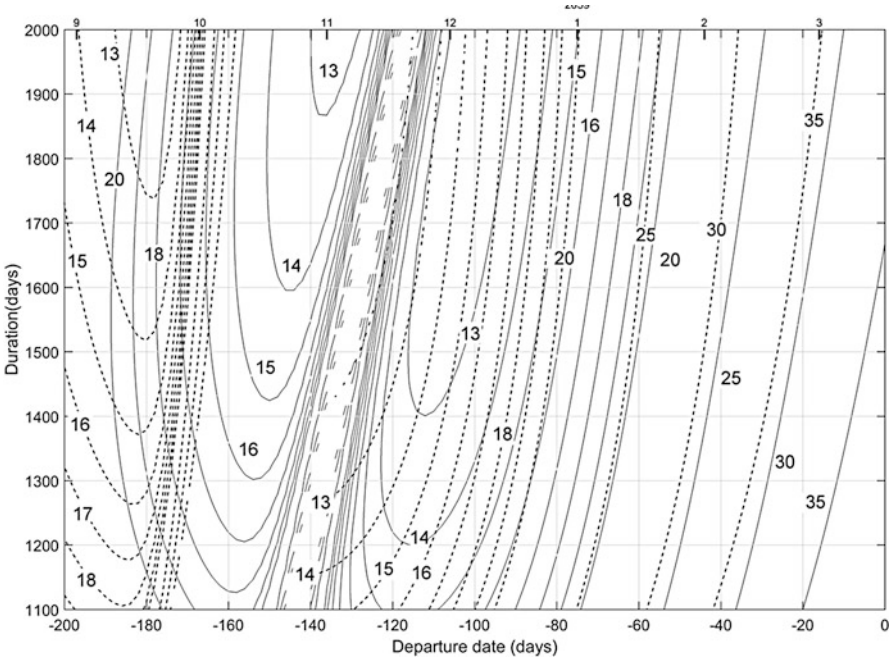


Figure 6 Superimposition of the ΔV-plots for orbit-to-orbit Earth–Saturn travels in the 2039 launch opportunity without (full lines) and with (dashed lines) a Jupiter flyby

- Starting orbit: Circular, 800 km (adequate for NTP) from Earth surface.
- Arrival orbit: Circular about Saturn, same radius as Titan orbit, 1,161,662 km (from which the spacecraft can be positioned in the Titan–Saturn L4 or L5 from where the lander can start).

Spacecraft data: NTP with $I_s = 850$ s.

The orbit-to-orbit ΔV-plots without (full lines) and with (dashed lines) a Jupiter flyby are reported in Figure 6: the ΔV is expressed in km/s and the dates are in days after the opposition (March 17, 2039). From the plot it is clear that performing a flyby it is possible to shorten (even if not dramatically) the duration of the mission at equal ΔV or decreasing the ΔV at equal duration.

A more thorough use of flybys would be to start, for instance, with an Earth or Venus flyby, perhaps a multiple one. This has, however, not been considered here since it would drastically increase the number of possibilities which need to be considered and also would drastically increase the travel time.

A possible starting date for a 1450-day mission without flyby is November 24, 2038 (113 days before opposition), with arrival date on November 13, 2042. The relevant speed increments are $\Delta V_1 = 8158.8$ m/s and $\Delta V_2 = 5088.7$ m/s ($\Delta V_{tot} = 13,347.5$ m/s).

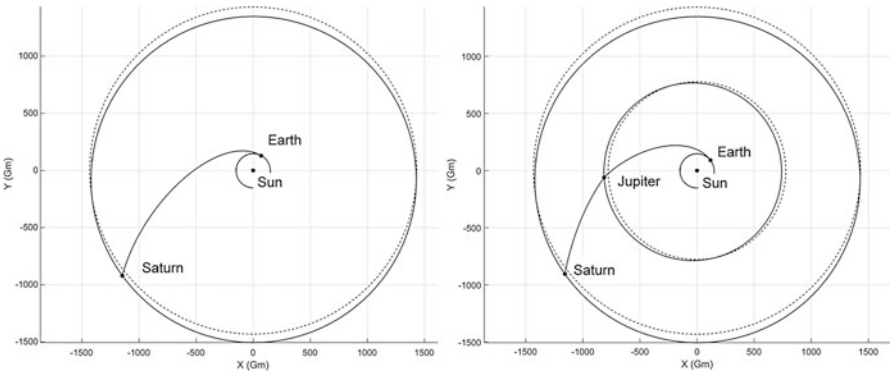


Figure 7 Trajectories for the Earth–Saturn mission without and with Jupiter flyby

Owing to the high specific impulse, the ratio $m_i/m_f = 4.9$.

In case of Jupiter flyby, a possible starting date for a 1450.0-day journey is October 31, 2038. The flyby occurs on June 29, 2040 (607.1 days after leaving Earth orbit) and the arrival date is October 20, 2042.

The flyby main parameters are: distance from the centre of Jupiter 3,561,954 km and deviation angle $\delta = 30.37°$. The relevant speed increments are $\Delta V_1 = 7161.3$ m/s and $\Delta V_2 = 5470.8$ m/s ($\Delta V_{tot} = 12,632.1$ m/s).

The ratio $m_i/m_f = 4.5$. A possible mass breakdown is final mass: 4.4 t, initial mass 19.8 t; propellant mass: 15,4 t.

A plot of the trajectories for the two cases is reported in Figure 7.

7.2 NEP Mission to Europa

Europa is one of the most interesting destinations in the solar system, and sending a lander, possibly with a rover, on its surface is an important mission which may be performed by a NEP robotic spacecraft. The problem of choosing the best launch date for starting such a mission in the 2039 launch opportunity is here dealt with, together with a first approximation mission design. Trajectory data:

- Starting orbit: Circular, 800 km from Earth surface.
- Arrival orbit: Circular about Jupiter, same radius as Europa orbit 670,900 km (from which the spacecraft can be positioned in the Europa-Jupiter L4 or L5 from where the lander can start).

Spacecraft data:

- Generator: Nuclear, with $\alpha = 5$ kg/kW $= 0.005$ kg/W.
- Thruster: VASIMR®, $H = 0.625$ ($\alpha_e = \alpha/H = 8$ kg/kW), $I_{smax} = 8000$ s.

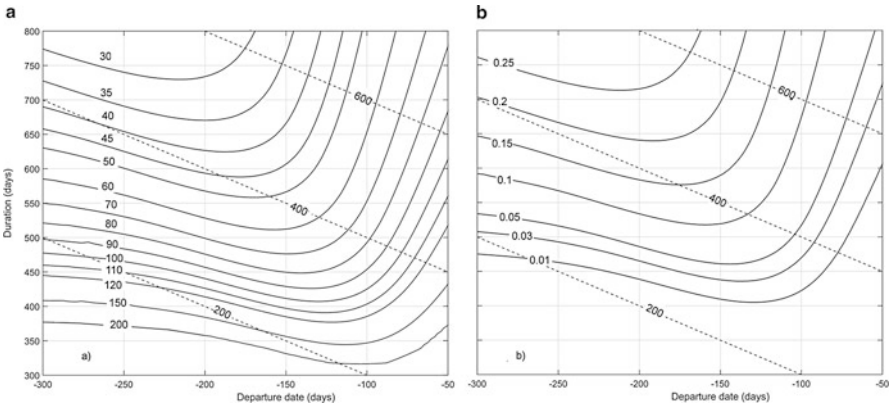


Figure 8 (**a**) Orbit-to-orbit J-plot and (**b**) bacon plot for a Jupiter-Europa mission in the 2039 launch opportunity

The J-plot for an orbit-to-orbit trajectory is first obtained together with a bacon plot (Figure 8; J is expressed in m^2/s^3; in the bacon plot ratio $(m_f\text{-}m_w)/m_i$ has been reported).

A reasonable choice for the starting date is August 7, 2038 (192 days before the opposition) which, together with a travel duration of 640 days, yields a value $(m_L + m_s)/m_i = 0.2$.

The first phase (geocentric) lasts 47.4 days and the spacecraft enters the interplanetary trajectory on September 23, 2038 (144.6 days before the opposition). The interplanetary travel lasts 521.6 days, with an arrival at Jupiter on February 26, 2040 (376.9 days after the opposition). The date at which the final orbit is reached is May 8, 2040 (448.0 days after the opposition) after 71.1 days spent spiralling about Jupiter (in this computation the gravitational attraction of Jupiter satellites has been neglected).

The trajectory of the spacecraft is shown in Figure 9.

The values of J are 5.528 m^2/s^3, 23.446 m^2/s^3, and 9.187 m^2/s^3 for the three phases, with a total $J = 38.161$ m^2/s^3. The overall value of γ is 0.553.

The time histories of the acceleration, the specific impulse, the thrust and the power or the jet are reported in Figure 10.

A possible mass breakdown of the spacecraft is the following:

• Initial mass	19,977 kg
• Payload plus structures (including the thrusters)	4000 kg
• Propellant mass	11,038 kg
• Power generator mass	4939 kg
• Generator power	988 kW

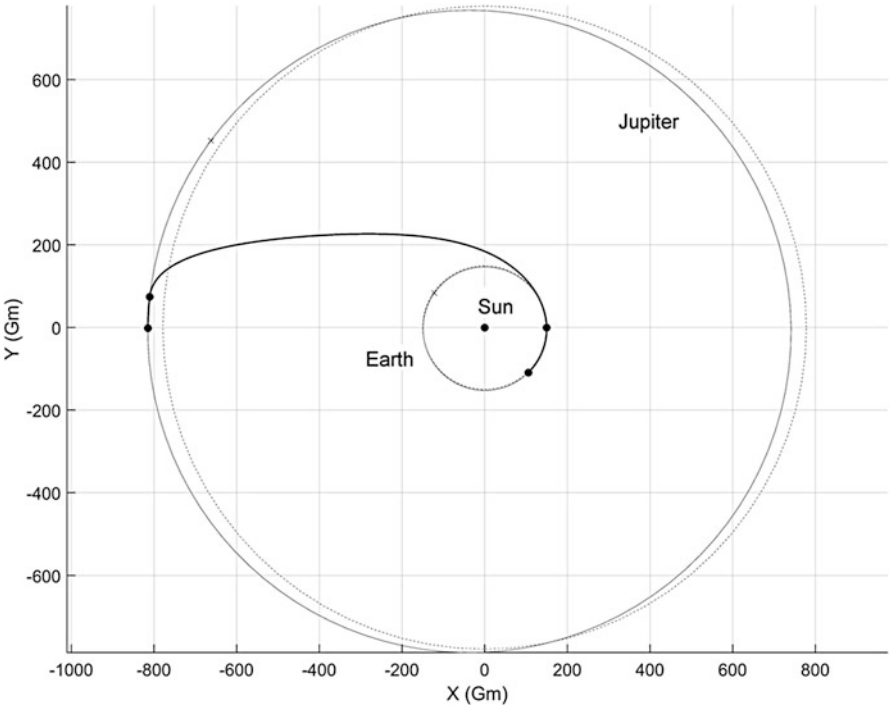


Figure 9 Trajectory of the Earth–Jupiter mission

A mission to Europa allowing to bring a payload of the order of about 1.5 t in less than 2 years with an IMLEO of 20 t is thus possible if a nuclear generator of moderate power with a specific mass of 5 kg/kW is available.

8 Conclusions

The choice of the launch date and of the travel duration for interplanetary missions is a delicate one, since it strongly affects the energy required for the mission, or, which is the same, the IMLEO and thus the cost.

The usual approach is defining a cost function linked in some way to the IMLEO, or better to the ratio m_i/m_L, where m_L is the mass of the payload carried to destination and plotting the surface which expresses the cost function versus the start and arrival dates, or better the start date and the journey duration. This plot, usually in the form of a contour plot, can be used to search for a minimum of the cost function, or better, to perform a trade-off between the cost function and the travel time—in particular in the case of human missions, in which minimizing the travel time is in itself a very important goal.

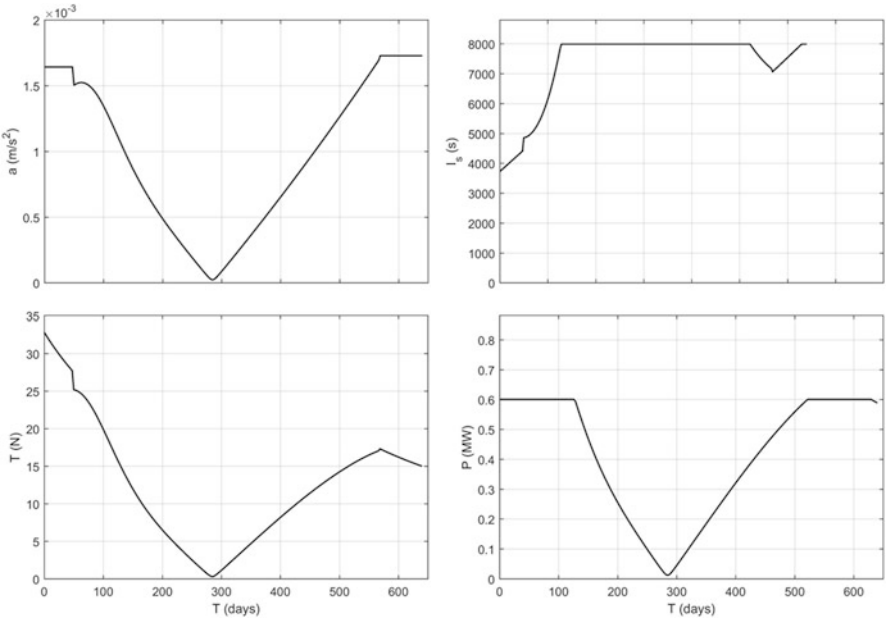


Figure 10 Time histories of the acceleration, the specific impulse, the thrust and the power of the jet

The resulting plots can be subdivided in two wide classes, depending on the type of propulsion used.

If the spacecraft is propelled by an impulsive system and most of the trajectory is performed without thrust, and thus its shape is made up by conic sections (if the two-body problem is assumed), the resulting surface has one or two minima (the contour lines are closed lines) and it is possible to define optimal conditions for the IMLEO. Since the zones about the minima are fairly flat, it is possible to search for a compromise between time and cost, but it is not convenient to move much from the optimal conditions.

The plots used in this case are called pork-chop plots (usually when only the hyperbolic excess speed needed to start the trajectory is considered), or ΔV-plot, when the total ΔV required to leave the parking orbit about the starting planet and to enter an orbit around the destination planet (or even to leave the surface of the former and to reach the surface of the latter) is considered.

On the contrary, if the spacecraft is propelled by a thruster which supplies a small thrust for the entire duration of the flight, or at least for a substantial part of it, the freedom of choosing the trajectory is much larger. This is the case of electric propulsion, but also of propellantless propulsion like solar or laser sails. Even if the two-body problem is solved, the trajectories are not conic sections (except for the coasting arcs which can be included), and a thrust programme must be defined together with the trajectory. The surface doesn't have minima (or maxima), but there is a constant decrease of the cost function with increasing travel time. A compromise

with short travel time and low IMLEO must always be searched and this depends on the mission to be designed. For instance, this allows to distinguish between cargo spacecraft, which can economically carry large cargos along slow trajectories, and passenger spacecraft, which can carry a reduced payload along fast trajectories.

The plots of this kind are called bacon plots (in which the cost function must be maximized) and J-plots. Also in this case it is possible to take into account not only the interplanetary part of the journey, which takes place in the sphere of influence of the Sun, but also the parts occurring within the spheres of influence of the starting and the arrival planets, to obtain orbit-to-orbit plots.

A third type of plot can then be introduced to study two-way journeys, in particular in the case in which they cannot be studied independently. Two-way journeys are important practically only in case of human missions, which at present are mostly aimed to Mars exploration. In the case of Mars, "long-stay missions" can be studied by considering independently the outbound and the inbound journeys, while in case of short-stay missions they must be dealt with in an integrated way. A total cost function can be defined, but this depends on four parameters, the start and arrival dates of both legs of the travel, or the start date, the two travel times and the stay on the planet. By fixing the stay time and searching for the start date leading to the smallest cost function, a plot in which the cost function is plotted against the two travel times can be obtained. A trade-off between the time spent in space and the cost of the mission, for a given stay time, can thus be performed.

A.1 Appendix: Optimization of a VEV Electric Spacecraft

A.1.1 NEP Spacecraft

Consider the case of an ideal VEV (Variable Exhaust Velocity) spacecraft—i.e. a spacecraft provided with a thruster which has no limitation in the specific impulse, powered by a nuclear generator.

The optimization of the trajectory and of the spacecraft can be performed separately as shown in [25] for the case in which the initial mass m_i is assumed to be

$$m_i = m_L + m_s + m_w + m_t + m_{ta} + m_p, \tag{4}$$

where m_L is the payload mass, m_s is the structural mass, m_w is the mass of the generator, m_t is the mass of the thruster, m_{ta} is the mass of the tanks and m_p is the mass of the propellant.

The power system can be assumed as made by a generator, a power conditioning system and a thruster (Figure 11b); the mass of the generator can be assumed to be proportional to the power it produces and that of the thruster to the power it receives from the ECU. Including the mass of the ECU into that of the generator, the total mass of the propulsion system:

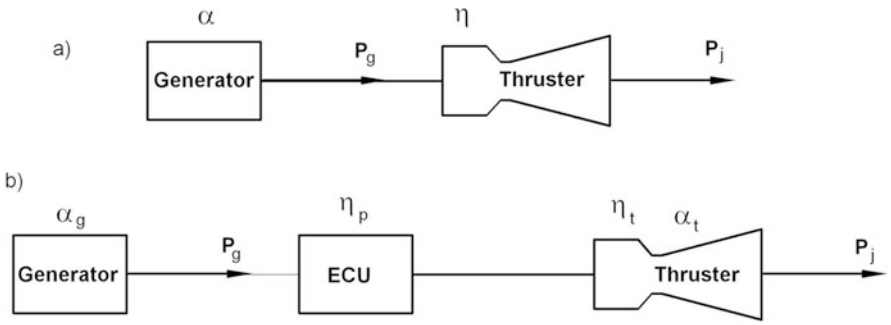


Figure 11 Sketch of the propulsion system. (**a**) simplified scheme; (**b**) more detailed scheme

$$m_{ps} = m_w + m_t = \alpha_g P_g + \alpha_t P_g \eta_p = \left(\frac{\alpha_g}{\eta_p \eta_t} + \frac{\alpha_t}{\eta_t} \right) P_j = \alpha_e P_j, \tag{5}$$

where P_j is the power of the jet and α_e, α_g and α_t are the effective specific masses of the power system, the generator and the thruster.

The power of the jet is linked with the thrust **T**, the exhaust velocity v_e and the propellant throughput $\dot{m}$ by the relationship

$$P_j = \frac{1}{2} \dot{m} v_e^2 = \frac{1}{2} T v_{ej}, \tag{6}$$

where P_j is the power of the jet and α_e is the effective specific mass of the generator.

The ratio between the thrust and the mass (both functions of time) is

$$\mathbf{a} = \frac{\mathbf{T}}{m} = -\frac{\dot{m} \mathbf{v}_e}{m}. \tag{7}$$

Solving Eq. (7) in v_e and substituting it in Eq. (6) it follows:

$$\frac{a^2}{2P_j} = -\frac{\dot{m}}{m^2} = -\frac{d}{dt}\left(\frac{1}{m}\right). \tag{8}$$

By integrating Eq. (8) from the starting moment 0 to the final time, it follows:

$$\frac{1}{m_f} = \frac{1}{m_i} + \int_0^T \frac{a(u)^2}{2P_j(u)} du = \frac{1}{m_i} + \frac{1}{2P_j} \int_0^T a(u)^2 du. \tag{9}$$

The second formulation is justified by considering that in the case of NEP the system can be regulated so that the power is maintained constant at its maximum value and the thrust is regulated by changing the specific impulse. By introducing Eq. (12) into Eq. (16) it follows:

$$\frac{1}{m_f} = \frac{1}{m_i} + \frac{1}{m_{ps}} \frac{1}{2} \alpha_e \int_0^T a(u)^2 du = \frac{1}{m_i} + \frac{\alpha_e J}{m_{ps}} = \frac{1}{m_i} + \frac{\gamma^2}{m_{ps}}, \tag{10}$$

where parameters J and γ are, respectively,

$$J = \frac{1}{2} \int_0^T a(u)^2 du; \gamma = \sqrt{\alpha_e J}. \tag{11}$$

The optimum trajectory and thrust profile are obtained by minimizing the integral J, which can be used as a cost function of the optimization problem. In [25] it is shown that the minimum problem can be expressed by a set of 6 s order scalar differential equations (in the three-dimensional case):

$$\begin{cases} \ddot{\mathbf{r}} = -\nabla U(\mathbf{r}, t) + \mathbf{a}(t) \\ \ddot{\mathbf{a}} = -\mathbf{a} \nabla [\nabla U(\mathbf{r}, t)] \end{cases}, \tag{12}$$

where U ($\mathbf{r}$,t) is the gravitational potential in which the spacecraft is moving. The boundary conditions are simply the positions $\mathbf{r}$ and the velocities $\dot{\mathbf{r}}$ at the initial and final instants.

Equation (10) can be rewritten as

$$\frac{m_f}{m_i} = \frac{1}{m_i \left(\frac{1}{m_i} + \frac{\gamma^2}{m_{ps}} \right)} = \frac{m_{ps}}{m_i} \frac{1}{\left(\frac{m_{ps}}{m_i} + \gamma^2 \right)}. \tag{13}$$

From Eq. (4) the final mass can be written as

$$m_f = m_L + m_s + m_w + m_t + m_{ta}. \tag{14}$$

The latter mass m_{ta} can be written as

$$m_{ta} = K m_p, \tag{15}$$

where the mass of the tank has been assumed to be proportional to the mass of the propellant through the so-called tankage factor K.

The propellant fraction is

$$\frac{m_p}{m_i} = 1 - \frac{m_f}{m_i} = 1 - \frac{m_{ps}}{m_i} \frac{1}{\left(\frac{m_{ps}}{m_i} + \gamma^2 \right)}, \tag{16}$$

and thus

$$\frac{m_L + m_s}{m_i} = \frac{m_{ps}}{m_i} \left[\frac{(1 + K)}{\left(\frac{m_{ps}}{m_i} + \gamma^2 \right)} - 1 \right] - K. \tag{17}$$

Equation (17) can be written as

$$y = x\left[\frac{(1+K)}{(x+\gamma^2)} - 1\right] - K, \tag{18}$$

where

$$y = \frac{m_L + m_s}{m_i}, x = \frac{m_{ps}}{m_i}.$$

To maximize the payload factor y, it is enough to search for a generator mass x such that

$$\frac{dy}{dx} = \frac{(1+K)}{(x+\gamma^2)} - 1 - x\frac{(1+K)}{(x+\gamma^2)^2} = 0, \tag{19}$$

i.e.

$$\frac{m_{ps}}{m_i} = \gamma\sqrt{1+K} - \gamma^2. \tag{20}$$

It then follows:

$$\frac{m_p}{m_i} = \gamma\sqrt{\frac{1}{1+K}}, \frac{m_{ta}}{m_i} = \gamma K\sqrt{\frac{1}{1+K}}, \tag{21}$$

$$\frac{m_L + m_s}{m_i} = \left(\sqrt{1+K} - \gamma\right)^2 - K = 1 + \gamma^2 - 2\gamma\sqrt{1+K}.$$

Clearly $(m_l + m_s)/m_i$ must be positive, which puts a limitation to γ:

$$\gamma < \sqrt{1+K} - \sqrt{K}, \tag{22}$$

which is only a theoretical minimum, since it corresponds to a case in which the spacecraft is made only of propellant, tank and propulsion system, with no allowance for payload and structure.

The mass breakdown as a function of γ is shown, for the case $K = 0.06$, in Figure 12. In this case the limitation is $\gamma < 0.7846$.

From the figure it is clear that minimizing γ leads to not only minimizing the propellant fraction but also minimizing the propulsion system lass and maximizing the payload (and structure) fraction.

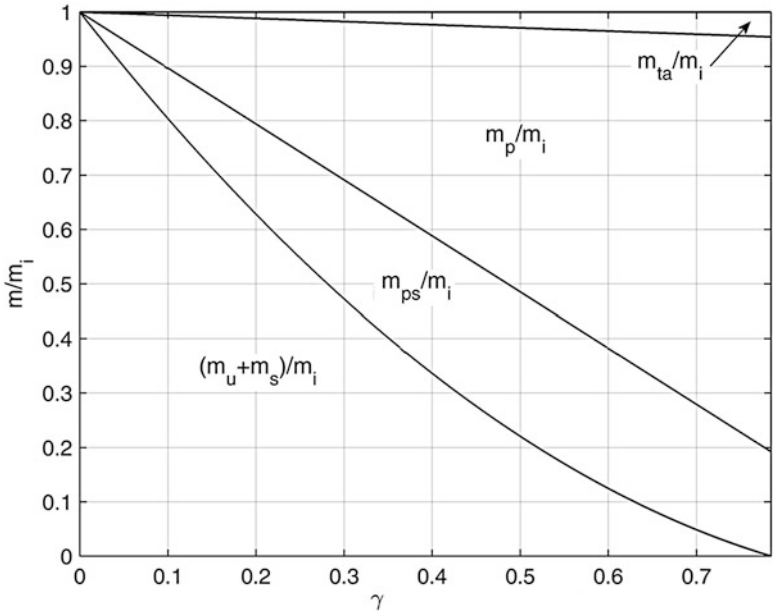


Figure 12 Mass breakdown as a function of γ for $K = 0.06$

A.1.2 SEP Spacecraft

Consider now an ideal VEV (variable exhaust velocity) spacecraft powered by a solar generator. The power generator supplies a power which depends on the distance from the Sun through a function $f(r)$:

$$P_g = P_0 f(r), \tag{23}$$

where P_0 is the power supplied at 1 AU from the Sun and function $f(r)$ is a function which takes a unit value when r = 1 AU. If the efficiency of the solar arrays were constant, function $f(r)$ would be

$$f(r) = \left(\frac{R_E}{r}\right)^2, \tag{24}$$

where R_E is the mean radius of Earth orbit. However, $f(r)$ decreases less than what could be obtained from Eq. (28) because the efficiency of the arrays increases with decreasing temperature and thus with increasing distance from the Sun. Other functions $f(r)$ have been suggested to account for this effect [20–22].

The power of the jet is thus linked with the mass of the generator as

$$P_j = m_w \frac{f(r)}{\alpha_{e0}}, \tag{25}$$

where α_{e0} is the effective specific mass of the propulsion system α_e at 1 AU from the Sun.

Operating as seen for the case of NEP, Eq. (10) becomes

$$\frac{1}{m_f} = \frac{1}{m_i} + \frac{1}{m_w}\frac{1}{2}\alpha_{e0}\int_0^T \frac{a(u)^2}{f(r)}du. \tag{26}$$

By introducing vector

$$\mathbf{q} = \frac{\mathbf{a}}{\sqrt{f(r)}}, \tag{27}$$

integral J can be written in the form

$$J = \frac{1}{2}\int_0^T q(u)^2 du, \tag{28}$$

and the differential equations yielding the trajectory and thrust profile become

$$\begin{cases} \ddot{\mathbf{r}} = -\nabla U(\mathbf{r}, t) + \mathbf{q}(t) f(r) \\ \ddot{\mathbf{q}} = -\mathbf{q}\nabla[\nabla U(\mathbf{r}, t)] \end{cases}. \tag{29}$$

Equations (29) can thus be used to solve the problem of the trajectory and the thrust profile in the case of a SEP spacecraft, exactly like Eq. (12), was used for a NEP spacecraft.

Equation (13) can be used again to obtain the final mass, provided that the correct value of γ, obtained from the value of J in Eq. (28), is introduced.

The optimized mass breakdown can be obtained in the same way seen in the previous section.

A.1.3 Limited Specific Impulse Spacecraft

In the previous two sections an ideal VEV spacecraft was considered. Actual systems usually cannot reach the very high value of the specific impulse which is required to optimize the thrust programme in the way seen above, i.e. a value high enough to reduce the thrust as required in the central part of the interplanetary trajectory, absorbing at the same time the maximum power which the power generator is able to supply.

The strategies which could be implemented are essentially two: reducing the power supplied to the thrusters so that the required thrust can be obtained with a possible value of the specific impulse and maintaining the power at its maximum value and then switching off the thruster for a suitable time, introducing a coasting arc in the trajectory. Both strategies cause a decrease of the performance of the system, since the generator is used below its potential capabilities for a certain period of the spaceflight.

The first approach has the advantage of not requiring to re-compute the trajectory, but it yields an overall performance lower than the second one. The second approach is better from the viewpoint of performances but requires to re-optimize both the trajectory and the thrust profile.

A secondary advantage is that the thrusters are not used for the whole travel time, reducing the required useful life of the device and, in the case of crewed missions, allowing the crew to perform maintenance and repair operations.

In both cases Eqs. (11) and (28) are no more valid, but a different definition of J can be introduced. After optimizing the trajectory, it is possible to compute the final mass of the spacecraft, and then the propellant mass fraction. At this point, the value of γ is readily obtained from the first Eq. (21)

$$\gamma = \sqrt{1+K}\frac{m_p}{m_i} \tag{30}$$

Since $\gamma = \sqrt{\alpha_e J}$, a value of parameter J can be obtained:

$$J = \frac{1+K}{\alpha_e}\left(\frac{m_p}{m_i}\right)^2. \tag{31}$$

References

1. Westwick, P.J.: Into the Black. Yale University Press, New Haven (2007)
2. Sergeyevsky, A.B.: Interplanetary Mission Design Handbook. Volume 1, Part 1: Earth to Venus Ballistic Mission Opportunities, 1991–2005 (1983). ntrs.nasa.gov
3. Sergeyevsky, A.B., Snyder, G.C., Cunniff, R.A.: Interplanetary Mission Design Handbook. Volume 1, Part 2: Earth to Mars Ballistic Mission Opportunities, 1990–2005 (1983). ntrs.nasa.gov
4. George, L.E., Kos, L.D.: Interplanetary Mission Design Handbook: Earth-to-Mars Mission Opportunities and Mars-to-Earth Return Opportunities 2009–2024 (1998). ntrs.nasa.gov
5. Burke, L.M.: Interplanetary Mission Design Handbook: Earth-to-Mars Mission Opportunities 2026 to 2045 (2010). ntrs.nasa.gov
6. Kemble, S.: Interplanetary Mission Analysis and Design. Springer, Chichester (2006)
7. Hintz, G.R.: Orbital Mechanics and Astrodynamics: Techniques and Tools for Space Missions. Springer, New York (2015)
8. Drake, B.G. (ed.): Mars Architecture Steering Group, Human Exploration of Mars, Design Reference Architecture 5.0 (and addendums), NASA Johnson Space Center (2009)

9. Shefer, A.: New method of orbit determination from two position vectors based on solving Gauss's equations. Sol. Syst. Res. **44**(3), 252–266
10. Choueiri, E.Y.: A critical history of electric propulsion: the first 50 years (1906–1956). J. Propuls. Power. **20**, 193–203 (2004)
11. Jahn, R.G., Choueiri, E.Y.: Electric Propulsion, Encyclopedia of Physical Science and Technology. Academic, New York (2002)
12. Diaz, F.C., Seedhouse, E.: To Mars and beyond, Fast!: How Plasma Propulsion Will Revolutionize Space Exploration. Springer, New York (2017)
13. Genta, G.: Next Stop Mars: The Why, How, and When of Human Missions. Springer, New York (2017)
14. Clark, J.S., George, J.A., Gefert, L., Doherty, M., Sefcik, R.: Nuclear Electric Propulsion: A Better, Safer, Cheaper Transportation System for Human Exploration of Mars, NASA technical memorandum 06406 (1994)
15. Woolley, R.C., Nicholas, A.K.: SEP mission design space for mars orbiters. In: AAS/AIAA Astrodynamics Specialist Conference, Vail (2015)
16. Ben-Asher, J.Z.: Optimal Control Theory with Aerospace Applications, AIAA Education Series. American Institute of Aeronautics and Astronautics, Reston (2010). ISBN: 978-1600867323
17. Marec, J.P.: Optimal Space Trajectories. Elsevier, New York (1979)
18. Langmuir, D.B.: Low-thrust flight. Constant exhaust velocity in field-free space, chap. 9. In: Seifert, H. (ed.) Space Technology. Wiley, New York (1959)
19. Edelbaum, T.M.: Optimal Space Trajectories. Analytical Mechanics Associates, Jericho (1969)
20. Irving, J.H.: Low-thrust flight. Variable exhaust velocity in gravitational fields, chap. 10. In: Seifert, H. (ed.) Space Technology. Wiley, New York (1959)
21. Circi, C.: Mars and mercury missions using solar sails and solar electric propulsion. J. Guid. Control. Dyn. **27**(3), 496–498 (2004)
22. Williams, S.N., Coverstone-Carroll, V.L.: Benefits of solar electric propulsion for the next generation of planetary exploration missions. J. Astronaut. Sci. **45**(2), 143–160 (1997)
23. Kim, M.: Continuous Low-Thrust Trajectory Optimization: Techniques and Applications. Virginia Polytechnic Institute and State University, Blacksburg (2005)
24. Genta, G., Maffione, P.F.: Low thrust interplanetary transfers: second approximation computation of Planetocentric phases. Advances in Aerospace Science and Technology. **1**, 100–107 (2017)
25. Keaton, P.W.: Low Thrust Rocket Trajectories, LA-10625-MS, Los Alamos (2002)
26. Genta, G., Maffione, P.F.: Optimal low-thrust trajectories for nuclear and solar electric propulsion. Acta Astronaut. **118**, 251–261 (2016)
27. Shampine, L.F., Reichelt, M.W., Kierzenka, J.: Solving Boundary Value Problems for Ordinary Differential Equations in MATLAB with bvp4c (2010). http://www.mathworks.com/bvp_tutorial
28. Rieck, M., Bittner, M., Gruter, B., Diepolder, J.: FALCON.m User Guide. Institute of Flight System Dynamics (Technical University of Munich, 2016). www.falcon-m.com
29. Wachter, A., Biegler, L.T.: On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)
30. Betts, J.T.: Practical Methods for Optimal Control and Estimation Using Nonlinear Programming. Second Edition, Advances in Design and Control. SIAM, Society for Industrial and Applied Mathematics, Philadelphia (2009)
31. Topputo, F., Zhang, C.: Survey of direct transcription for low-thrust space trajectory optimization with applications. In: Abstract and Applied Analysis. Hindawi Publishing Corporation, Cairo (2014)

32. Genta, G., Maffione, P.F.: Sub-Optimal Low-Thrust Trajectories for Human Mars Exploration, Atti dell'Accademia delle Scienze di Torino, Memorie Sc. Fis, vol. 38–39, pp. 87–126 (2014–2015)
33. Matloff, G.: The Starflight Handbook. Wiley, New York (1989)
34. White, H., March, P., et al.: Measurement of impulsive thrust from a closed radio-frequency cavity in vacuum. J. Propuls. Power. **1**, 12 (2016)
35. Genta, G., Genta, A.: Preliminary assessment of a small robotic rover for titan exploration. Acta Astronaut. **68**(5–6), 556–566 (2011)

Optimal Topological Design of a Thermal Isolator for a Monopropellant Space Thruster

Sebastián Miguel Giusti, Augusto Alejandro Romero, and Javier Eduardo Salomone

Abstract This work is focused on the study of the thermal-structural behavior of a thermal isolator device employed in a monopropellant thruster for space applications. Engines of this kind are widely used to perform attitude corrections in artificial satellites. Their operating principle is based on the catalytic decomposition of the fuel (hydrazine), producing gasification with a consequent heat generation. These gases are properly conducted to a nozzle to produce thrust. A couple of redundant solenoid *on-off* electro-valves, in a serial configuration, are used to control the fuel supply system. To avoid leak risk in this system, soft seals are also used. Duration and performance of this kind of engine rely on two main aspects. The first one is the number of cold ignitions. When the engine starts at low temperature conditions, the catalytic bed is subjected to a thermal transient (high gradient—hundreds of $^{\circ}C$/s) which generates a breakage of grains, causing low size particles to fill the inter-granular spaces, clogging the downstream gas flow. The second aspect to consider in the reduction of the life span is the loss of reliability in the soft seals used in the fuel supply system due to high temperature degradation. Such degradation can drive the module out of service, generate a catastrophic failure in the reactor, or lead to mission stoppage. A thermal isolator is used to protect the seals from a premature degradation due to thermal effects. Its structural design is optimized by using a novelty structural optimization methodology based on topological sensitivity analysis in this work. This analysis allows achieving the best structural configuration that minimizes the temperature field around the seals

S. M. Giusti (✉) · A. A. Romero
National Technological University, Regional Faculty of Córdoba (UTN/FRC), Research and Development Group in Applied Mechanics (GIDMA), Córdoba, Argentina

National Scientific and Technical Research Council (CONICET), Buenos Aires, Argentina
e-mail: sgiusti@frc.utn.edu.ar; aromero@frc.utn.edu.ar

J. E. Salomone
National Technological University, Regional Faculty of Córdoba (UTN/FRC), Research and Development Group in Applied Mechanics (GIDMA), Córdoba, Argentina
e-mail: jsalomone@frc.utn.edu.ar

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_6

and also the isolator weight. Finally, a thermal and structural evaluation of the monopropellant thruster is presented in order to validate the structural strength and integrity. Inertial forces due to high G's are considered in this analysis.

1 Introduction

Low thrust monopropellant engines are usually used in launch vehicles as reaction control systems, attitude and positioning control of low earth orbit (LEO) satellites, apogee insertion and north-south keeping of geosynchronous satellites, speed increment (ΔV) and retro-propulsion of spaceships, and as orbit and attitude correctors in artificial satellites. In this work is considered a 1.5 N thrust monopropellant engine to be used in a LEO class satellite.

The fuel used by the engine is hydrazine and is conducted by a platinum-iridium alloy pipe through the combustion chamber. The pipe is welded in the inlet zone to the wall of the chamber to fix and seal the interface. Inside the chamber, an injector distributes the propellant through the catalytic bed. When the fuel interacts with the catalyst, the breakage of the molecules will produce the following reaction:

$$1^{st}\text{stage} : 3N_2H_4 \rightarrow 4NH_3 + N_2 \tag{1}$$

$$2^{nd}\text{stage} : 3NH_3 \rightarrow 2N_2 + 6H_2 \tag{2}$$

In the first stage, the fuel dissociates into ammonia and nitrogen producing an exothermic reaction, while in the second stage the ammonia dissociates into nitrogen and hydrogen in an endothermic reaction.

The thruster was designed based on the following guidelines:

- High quantities of propulsion shots,
- Long life span,
- Redundancies to prevent catastrophic failure,
- High performance,
- Cold ignition avoidance,
- Prevent degradation of soft seals,
- Prevent monopropellant freezing.

Figure 1 shows a section view of the thruster, where it is possible to see the proximal valve, the thermal isolator, the soft seals, the Pt-Ir pipe, and the catalytic bed. The heat flow generated in the catalytic bed is dissipated by means of: (a) radiation to space, (b) conduction to the vehicle through the feeding pipe and the thermal isolator to the proximal valve. An elastomeric soft seal, which is highly sensitive to high temperatures, is located close to the proximal valve. This component will be protected by means of an optimal design of the thermal isolator. For the detailed design and analysis of the engine, the reader may refer to the works [7, 27–29]. In Figure 2 a detailed view of the soft seal region, indicated by a red dashed box in Figure 1, is presented.

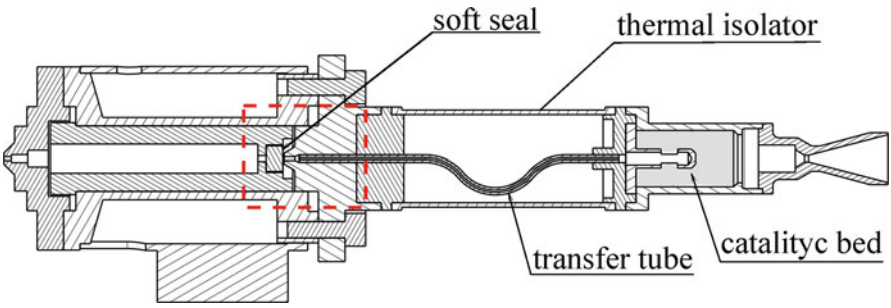
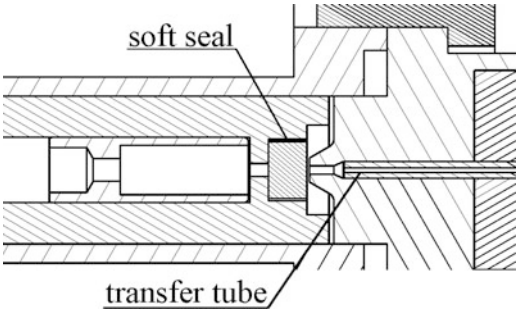


Figure 1 Section of monopropellant thruster

Figure 2 Detailed view of soft seal zone



The main objective of this work is to optimize the thermal isolator design in a way that maximizes the thermal dissipation by radiation and minimizes the thermal conduction towards the area next to the soft seals, keeping its weight within acceptable values. In addition, a mechanical and thermal simulation will be carried out in order to validate that the optimized component can withstand the environmental conditions and mechanical loads that result from space operations.

The thermal isolator optimization is performed by applying a novelty topological optimization technique based on topological sensitivity analysis and topological derivative concept [22]. The main characteristic of this technique is that it allows to establish the sensitivity to topological changes in every point of the design domain. In general, a topological change is represented by the introduction of a hole at the point where the sensitivity reaches the maximum or the minimum value, depending on the objective pursued with the optimization. In this sense, the notion of a topological change can be extended to study how two different materials interact between them. Therefore, it is possible to establish the sensitivity to a change in the constitutive properties of the underlying material at each point of the project domain.

The algorithm used to perform the topology optimization is based on the geometrical description of the domain via a level-set representation, see [23]. The topological derivative is used as a feasible descent direction in the algorithm, and the

procedure is described in detail in [3, 20]. Once the optimization procedure finishes and the optimal topology of the thermal isolator is obtained, a series of computational simulations will be executed to verify the thermal and structural behavior considering the operating conditions of the different configurations obtained.

This work is organized as follows. Section 2 introduces a theoretical background for the heat transfer problem. Section 3 describes the methodology to address the optimum design problem and the procedure to solve it. This section also explains topology optimization and topological derivatives general concepts. Section 4 shows the optimization procedure results and a thermal and structural validation are carried out for the different configurations obtained of the thermal isolator. The work ends in Sect. 5 where final remarks are presented.

2 Steady-State Heat Transfer Problem

Consider an isotropic three dimensional body Ω with a smooth enough boundary (surface) $\partial\Omega$ with outer normal denoted as n (see Figure 3). The thermal flux across the boundary $\partial\Omega$ is $q_n = q(\theta) \cdot n$, where θ is the temperature field. By considering a heat source b acting within the body Ω, the thermal equilibrium equation that governs the steady-state heat transfer problem is: Find the temperature field θ, such that

$$\operatorname{div} q(\theta) = b \text{in} \Omega. \tag{3}$$

By considering the Fourier's law, the thermal flux $q(\theta)$ can be stated as

$$q(\theta) = -k\nabla\theta, \tag{4}$$

where k is the thermal conductivity of the material. The above equation states that the thermal flux is opposite to the thermal gradient, and it will flow from a higher temperature zone to a lower temperature zone. Equation (3) must be solved taking into account the following boundary conditions:

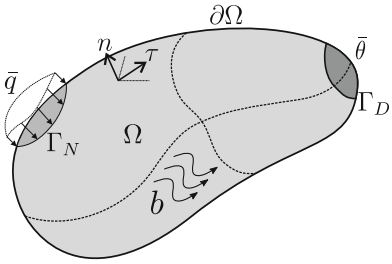


Figure 3 Representation of the steady-state heat transfer problem

$$\begin{cases} \theta & = \bar{\theta} \text{ on } \Gamma_D, \\ q(\theta) \cdot n & = \bar{q} \text{ on } \Gamma_N. \end{cases} \tag{5}$$

Being Γ_D the part of the boundary $\partial\Omega$ where the temperature θ is prescribed by $\bar{\theta}$. Analogously Γ_N denotes the part of the boundary $\partial\Omega$ where the flux $q(\theta)$ is prescribed by $\bar{q}$. In addition, these regions of the boundary must fulfill the following conditions, $\partial\Omega$=$\Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N$=$\emptyset$.

The set of Equations (3)–(5) is called the strong formulation of the steady-state heat transfer problem, see [25]. The total potential energy developed by the solid under the later conditions can be written as

$$\mathscr{J}(\theta) = \frac{1}{2}\int_{\Omega} k\nabla\theta \cdot \nabla\theta d\Omega + \int_{\Omega} b\theta d\Omega + \int_{\Gamma_N} \bar{q}\theta d\Gamma_N. \tag{6}$$

The associated variational formulation (or weak form) for the heat transfer problem presented in Equations (3)–(5) is: Find $\theta \in \mathscr{U}$, such that

$$\begin{aligned} &\int_{\Omega} q(\theta) \cdot \nabla\eta d\Omega = \int_{\Omega} b\,\theta d\Omega + \int_{\Gamma_N} \bar{q}\eta d\Gamma_N \qquad \forall\eta \in \mathscr{V}, \\ &\text{with} \quad q(\theta) = -k\nabla\theta\,. \end{aligned} \tag{7}$$

The functional set $\mathscr{U}$ and the functional space $\mathscr{V}$ are given by

$$\mathscr{U} := \{\phi \in H^1(\Omega;\mathbb{R}^3) : \phi|_{\Gamma_D} = \bar{\theta}\} \text{ and } \mathscr{V} := \{\phi \in H^1(\Omega;\mathbb{R}^3) : \phi|_{\Gamma_D} = 0\}, \tag{8}$$

Hilbert's space, denoted as $H^1(\Omega;\mathbb{R}^3)$, is defined as

$$H^1(\Omega;\mathbb{R}^3) : \left\{\phi \in \Omega \subset \mathbb{R}^3; \left(\int_{\Omega} \phi^2 + (\nabla\phi)^2\, d\Omega\right)^{1/2} \leq c < \infty\right\} \tag{9}$$

where c is an arbitrary constant. The choice of this function space ensures that both the discrete solution of the variational equation (7) and its gradient are bounded by the constant c. For other heat transfer models by considering, for instance, anisotropic medium, radiation effects, see [8, 25] and for a detailed explanation on functional spaces see [19].

3 Structural Topological Optimization

As it was mentioned at the beginning, the thermal isolator is used to prevent the overheating of the soft seals near the proximal valve. The working principle is the following: first it generates a thermal conductive resistance between the *hot side* of the space thruster (near the nozzle) and the *cold side* (near the proximal valve) and

second, it works as a heat radiant emitter. Therefore, to obtain an optimal design for the thermal isolator it is necessary to minimize the thermal conduction in a particular direction and also maximize the thermal dissipation by radiation. These conditions can be expressed mathematically as

$$\Psi(\Omega,\theta) = \frac{1}{2}\int_{\Omega} k\nabla\theta\cdot\nabla\theta d\Omega + \int_{\Gamma_N}\bar{q}\theta d\Gamma_N - \lambda|\Omega|, \tag{10}$$

where $|\Omega|$ is used to denote the measure of the domain Ω. Note that the two first terms in (10) represent the total potential energy associated with the stationary heat conduction problem introduced in Equation (6) without a thermal source b, and the last one represents the volume of the material to be optimized, which is penalized by the parameter λ. This penalization parameter λ allows in the optimization procedure the introduction of a linear constraint on the final volume of the thermal isolator v_f. This technique is called *penalty function method* or *sequential unconstrained minimization*, see [21, 24]. The minimization of the proposed functional meets with the proposed objectives of this work, since:

1. the minimization of the total potential energy will define the topology of the reference configuration Ω that conduces less amount of heat towards the soft seals,
2. under the same hypothesis mentioned above, the problem of maximizing heat dissipation towards the outer space in form of radiation is equivalent to maximizing the exposed surfaces (represented by the term $\lambda|\Omega|$).

Therefore, there is a competition to obtain the optimal topology of the domain Ω between these two factors: the first part would try to minimize the conduction surface, and at the same time, the last term would attempt to maximize the exposed surface. Finally, the optimization problem can be stated as:

$$\begin{cases} \underset{\Omega}{\text{Minimize}} \quad \Psi(\Omega,\theta), \\ \text{Subjected to } \theta \text{ being the solution of (7)}, \end{cases} \tag{11}$$

3.1 Topological Derivative Concept

The topological sensitivity analysis gives the topological asymptotic expansion of a shape functional with respect to an infinitesimal singular domain perturbation. The main term of this expansion is a scalar field called topological derivative. In order to introduce these concepts, let us consider the open bounded domain $\Omega \in \mathbb{R}^3$, which is subjected to a non-smooth perturbation in a small region $B_\epsilon(\widehat{x})$ of size ϵ with center at an arbitrary point $\widehat{x} \in \Omega$. Thus, introducing a characteristic function $\chi = \mathbb{1}_\Omega$, associated with the unperturbed domain, it is possible to define the characteristic function associated with the topological perturbed domain χ_ϵ.

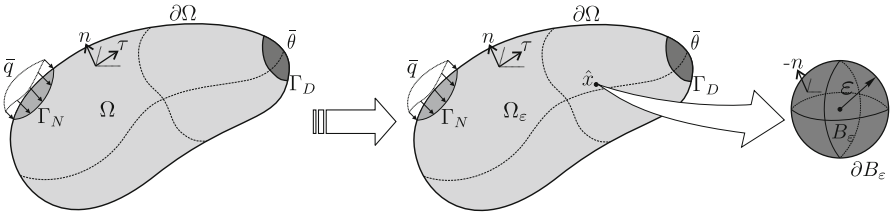


Figure 4 Topological derivative concept

Particularly, if the topological perturbation is an inclusion, we have $\chi_\epsilon(\widehat{x}) = \mathbb{1}_\Omega - (1-\gamma)\mathbb{1}_{\overline{B_\epsilon(\widehat{x})}}$, where $\gamma \in \mathbb{R}^+$ is the contrast parameter in the material property of the medium, see Figure 4. Then we assume that a given shape functional $\psi(\chi_\epsilon(\widehat{x}))$, associated with the topological perturbed domain, admits the following topological asymptotic expansion:

$$\psi(\chi_\epsilon(\widehat{x})) = \psi(\chi) + f(\epsilon)T_D\psi(\widehat{x}) + o(f(\epsilon)) \,, \tag{12}$$

where $\psi(\chi)$ is the shape functional associated with the unperturbed domain Ω, $f(\epsilon)$ is a function such that $f(\epsilon) \to 0^+$, with $\epsilon \to 0^+$. Equation (12) can be interpreted as a Taylor expansion, which first term measures the sensitivity of a topological change in the domain Ω. A function $\widehat{x} \mapsto T_D\psi(\widehat{x})$ is the so-called topological derivative of ψ in the point $\widehat{x}$. Thus, the topological derivative can be seen as a first order correction factor over $\psi(\chi)$ to approximate $\psi(\chi_\epsilon(\widehat{x}))$. In fact, after rearranging (12), we have

$$\frac{\psi(\chi_\epsilon(\widehat{x})) - \psi(\chi)}{f(\epsilon)} = T_D\psi(\widehat{x}) + \frac{o(f(\epsilon))}{f(\epsilon)} \,. \tag{13}$$

Taking the limit $\epsilon \to 0^+$ in the above expression,

$$T_D\psi(\widehat{x}) = \lim_{\epsilon\to 0^+} \frac{\psi(\chi_\epsilon(\widehat{x})) - \psi(\chi)}{f(\epsilon)} \,. \tag{14}$$

Note that the shape functionals $\psi(\chi_\epsilon(\widehat{x}))$ and $\psi(\chi)$ are associated with domains with different topologies. Then, to calculate the limit $\epsilon \to 0^+$ in (14) it is necessary to perform an asymptotic expansion of the functional $\psi(\chi_\epsilon(\widehat{x}))$ with respect to the parameter ϵ.

Expression (14) represents the topological sensitivity of the shape functional $\psi(\chi)$ due to the introduction of a singular perturbation in an arbitrary point $\widehat{x} \in \Omega$. Historically, the topological derivative concept was rigorously introduced in [30]. Since then, this concept has been widely used in several research areas and engineering applications, see, for instance, the works [1, 6, 11–13, 15, 17, 18, 26, 34] and the book [22].

Theorem 1 *For the shape functional stated in Equation* (10) *and considering a sphere of radius* ϵ *as topological perturbation, the analytical expression of the topological derivative* (14) *is given by:*

$$T_D\Psi(\hat{x}) = -3k\frac{(1-\gamma)}{(2+\gamma)}\nabla\theta(\hat{x})\cdot\nabla\theta(\hat{x}) - \lambda \ , \tag{15}$$

where γ *is the contrast coefficient and defines a relation between the base material and the perturbation material.*

Proof The reader interested in the proof of this result may refer to [2, 22].

The topological derivative (15) will be used as a feasible descent direction in a computational framework for topology optimization.

3.2 Optimization Procedure

The optimization procedure is based on the representation of the domain in a bi-material fashion. The topology is identified by the hard material distribution (denoted as Ω^h) and the inclusions of soft material (denoted as Ω^s) are used to mimic the holes. The constitutive properties of these regions are characterized by the thermal conductivity k, and the phase contrast γ^* so that, we have

$$k(x) = \begin{cases} k \ \forall x \in \Omega^h \\ \gamma^* k \ \forall x \in \Omega^s \end{cases} . \tag{16}$$

Based on the approach presented above, note that the constitutive properties of the domains Ω^h and Ω^s are correlated with the contrast parameter γ^* (see Figure 5).

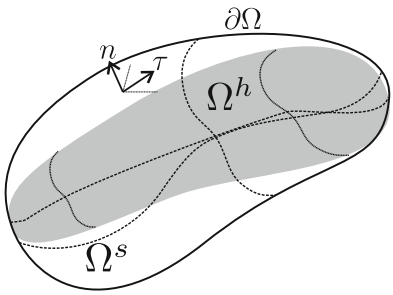


Figure 5 Bi-material distribution in the domain Ω

Therefore, the optimization problem (11) can be re-written as:

$$\begin{cases} \text{Minimize} \quad \Psi(\Omega,\theta) = \frac{1}{2}\int_{\Omega} k\nabla\theta\cdot\nabla\theta d\Omega + \int_{\Gamma_N}\bar{q}\theta d\Gamma_N - \lambda|\Omega^h|, \\ \Omega^h \subset \Omega \\ \text{Subjected to Find } \theta \in \mathscr{U} \text{ such that :} \\ \qquad \int_{\Omega} q(\theta)\cdot\nabla\eta d\Omega = \int_{\Gamma_N}\bar{q}\eta d\Gamma_N \quad \forall\eta\in\mathscr{V}. \end{cases} \tag{17}$$

It should be stressed that the design variable in problem (17) is the topology of the domain Ω^h. Hence, by considering the exact topological sensitivity information provided by the topological derivative (15) it is possible to construct a numerical optimization procedure to tackle the problem.

Then, the computation of (15) is carried out by using $\gamma = \gamma^*$ if $x \in \Omega^h$; and $\gamma = 1/\gamma^*$ if $x \in \Omega^s$. Having made the previous consideration and in order to solve the optimization problem (17), we use the topology optimization algorithm proposed by Amstutz and Andrä [3]. The procedure relies on a level-set domain representation [23] and the approximation of the topological optimality conditions by a fixed point iteration. The topological derivative (15) is used as a feasible descent direction to minimize the cost function $\Psi(\Omega,\theta)$. This class of algorithm has been successfully used in research areas related to topological optimization such as microstructure of materials [5], load-bearing structures and flow through porous media [3], load-bearing structures subjected to point-wise stress constraint [4, 6], and advanced multi-physics devices [11, 14, 16]. For completeness, the algorithm is outlined in the following. For further details we refer to the works [3] and [20].

By considering the level-set domain representation, the strong (or hard) material is characterized by a function $\Phi \in L^2(\Omega)$ such that

$$\Omega^h = \{x \in \Omega, \Phi(x) < 0\}\,, \tag{18}$$

whereas the weak (or soft) material domain is defined by

$$\Omega^s = \{x \in \Omega, \Phi(x) > 0\}\,. \tag{19}$$

Lebesgue spaces denoted by $L^2(\Omega)$ are defined as

$$L^2(\Omega;\mathbb{R}^3) : \left\{\phi \in \Omega \subset \mathbb{R}^3; \left(\int_{\Omega}\phi^2\, d\Omega\right)^{1/2} \leq c < \infty\right\} \tag{20}$$

where c is an arbitrary constant. The norm associated with this space is

$$\|\phi\|_{L^2(\Omega)} = \left(\int_{\Omega}\phi^2\, d\Omega\right)^{1/2} \tag{21}$$

and its corresponding inner product is stated as

$$\langle \phi_1, \phi_2 \rangle = \int_{\Omega} \phi_1 \phi_2 d\Omega . \tag{22}$$

Now, let us consider the topological derivative $T_D\Psi$. According to [3], a sufficient condition of *local optimality* of problem (11) for the class of perturbations consisting of spherical inclusions is

$$T_D\Psi(x) > 0 \quad \forall x \in \Omega \ . \tag{23}$$

To devise a level-set-based algorithm whose aim is to produce a topology that satisfies (23) it is convenient to define the function

$$g(x) = \begin{cases} -T_D^h\Psi(x) \text{ if } x \in \Omega^h \\ \ \ T_D^s\Psi(x) \text{ if } x \in \Omega^s \end{cases} . \tag{24}$$

With the above definition and Equations (18)–(19) it can be easily established that the sufficient condition (23) is satisfied if the following equivalence relation between the functions g and the level-set Φ holds

$$\exists\, \tau > 0 \quad \text{s.t} \quad g = \tau\, \Phi \ , \tag{25}$$

or, equivalently,

$$\varphi := \arccos \left[\frac{\langle g, \Phi \rangle_{L^2(\Omega)}}{\|g\|_{L^2(\Omega)} \, \|\Phi\|_{L^2(\Omega)}} \right] = 0 \ , \tag{26}$$

where φ is the angle between the vectors g and Φ in $L^2(\Omega)$. Starting from a given level-set function $\Phi_0 \in L^2(\Omega)$ which defines the chosen initial guess for the optimum topology, the algorithm proposed in [3] produces a sequence $(\Phi_i)_{i\in\mathbb{N}}$ of level-set functions that provides successive approximations to the sufficient condition for optimality (25). The sequence satisfies

$$\begin{aligned} &\Phi_0 \in L^2(\Omega) \ , \\ &\Phi_{i+1} \in \text{co}(\Phi_i, g_i) \qquad \forall i \in \mathbb{N} \ , \end{aligned} \tag{27}$$

where $\text{co}(\Phi_i, g_i)$ is the convex hull of $\{\Phi_i, g_i\}$. In the current algorithm the initial guess Φ_0 is normalized. With $\mathcal{S}$ denoting the unit sphere in $L^2(\Omega)$, the algorithm is explicitly given by

$$\begin{aligned} &\Phi_0 \in \mathcal{S} \ , \\ &\Phi_{i+1} = \frac{1}{\sin\varphi_i} \left[\sin((1-\kappa_i)\varphi_i)\Phi_i + \sin(\kappa_i\varphi_i) \frac{g_i}{\|g_i\|_{L^2(\Omega)}} \right] , \end{aligned} \tag{28}$$

where $\kappa_i \in [0, 1]$ is a step size determined by a line-search in order to decrease the value of the cost functional Ψ. The iterative process is stopped when for some iteration the obtained decrease in Ψ is smaller than a given numerical tolerance. If, at this stage, the optimality condition (25), (26) is not satisfied to the desired degree of accuracy, i.e., if $\varphi_{i+1} > \epsilon_\varphi$, where ϵ_φ is a pre-specified convergence tolerance, then a uniform mesh refinement of the structure is carried out and the procedure is continued.

Based on the above description, the main steps of the algorithm can be summarized as follows:

1. Choose an initial level-set function by defining the initial guess for the optimal structure domain.
2. Define the domains Ω^h and Ω^s according to (18) and (19).
3. Define the constitutive properties for the finite elements in each domain Ω^h and Ω^s according to (16).
4. Obtain the discretized field θ by solving the problem (7) by using the standard finite element method (FEM), see [33].
5. Compute the topological derivative field (15) at the Gauss point of the finite element and perform a standard nodal averaging procedure.
6. Obtain the function $g(x)$ according to (24) by using the nodal values of the topological derivative and compute the φ angle with Equation (26).
7. Update the level-set function Φ_{i+1} according to (28) and update the domains Ω^h and Ω^s by considering Equations (18) and (19).
8. Check convergence $\varphi_{i+1} \leq \epsilon_\varphi$. If True: Exit. If False: go to 3.

A general flowchart for the algorithm is shown in Figure 6.

3.3 Computational Optimization Model

The thermal isolator is made of INCONEL 600 (UNS N06600), where the constitutive properties exhibit a dependence with the temperature (see Figure 7). These dependencies are not considered for the optimization procedure. The constitutive properties considered for this analysis are estimated on a basis of an operating temperature at 500 °C. Under this consideration, the thermal conductivity is given by: $k = 22.1$ W/m °C .

The project domain for the optimization procedure of the thermal isolator consists of a cylindrical tube with a length of 24.7 mm, 10.65 mm inner diameter, and 11.75 mm outer diameter, see Figure 8.

The boundary conditions established on stationary state during the operation time were based on the following considerations:

- Prescribed temperature $\bar{\theta} = 330$ K on the thermal isolator/proximal valve interface, due to existence of a device that controls this temperature.

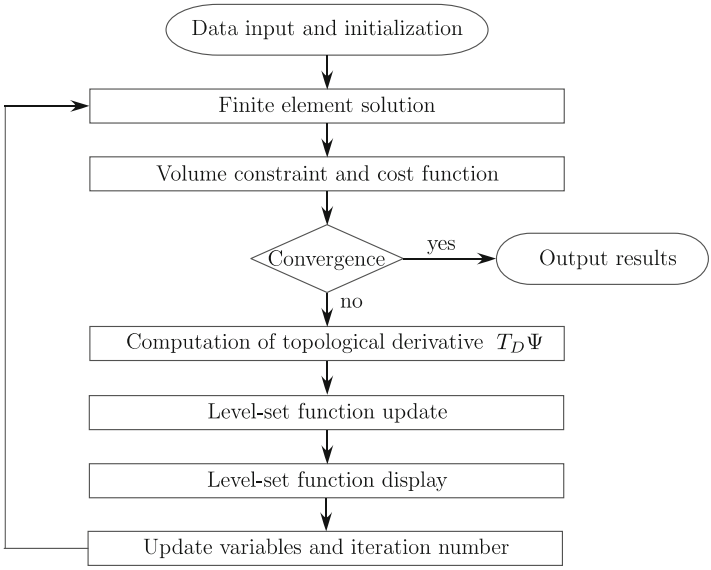


Figure 6 Flowchart of the optimization algorithm

- Thermal flux $\bar{q} = 28\,\text{W/mm}^2$ in the contact area between the engine and the catalytic bed. This flux was determined by considering the heat generated by the chemical reaction of the hydrazine.

Figure 9 presents the design domain and applied boundary conditions used for the topological optimization procedure.

In Figure 10 a selected set of optimal configurations obtained with the optimization algorithm for two volume fractions v_f is shown. Notice that to obtain the desired volume fraction v_f at the end of the optimization procedure, the penalization parameter λ must be enforced manually in each case. In Figure 10 gray color zones represent material INCONEL 600, while blue zones indicate holes. It can be seen that, although both configurations are optimal solutions from the thermal point of view, the results lack structural viability for volume fractions lower than 50%, since disconnected material zones appear.

In both cases, it can be seen how the optimization algorithm minimizes the conductive surface and at the same time maximizes the radiant surface.

4 Computational Simulations for Operational Conditions

The optimal topologies obtained from the optimization procedure will be validated under operational conditions in this section. In order to avoid stress concentration zones and constructive issues, the optimal solutions obtained from the algorithm

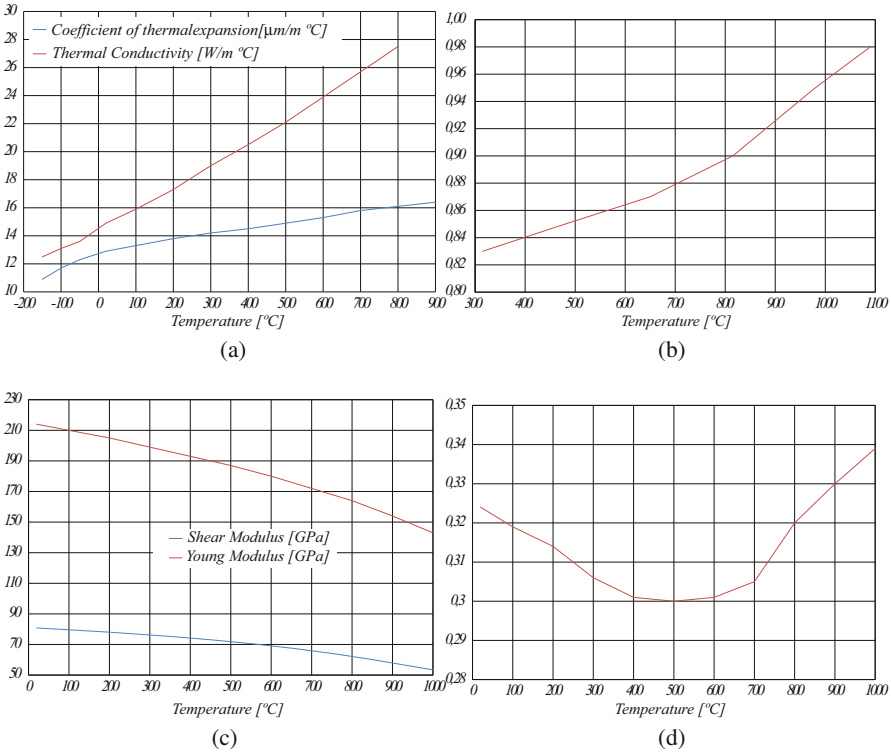


Figure 7 INCONEL 600 constitutive properties. (**a**) Thermal properties. (**b**) Emissivity. (**c**) Mechanical properties. (**d**) Poisson's ratio

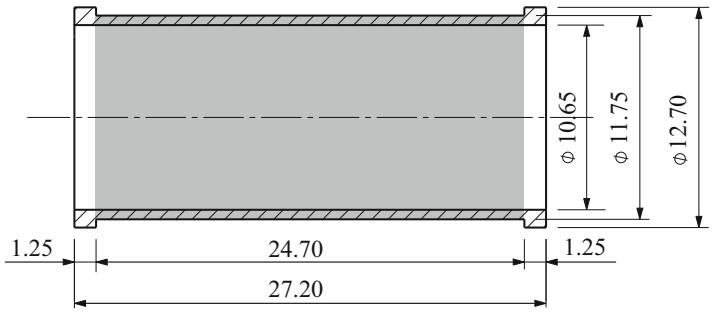


Figure 8 Domain for the optimization procedure in gray—dimensions in mm

described in the previous section (see Figure 10) need to be interpreted from an engineering point of view. This interpretation requires to smooth the corners of the holes to be drilled. After that, we proceed to model several configurations of the thermal isolator (each one with a different volume fraction). This is done in

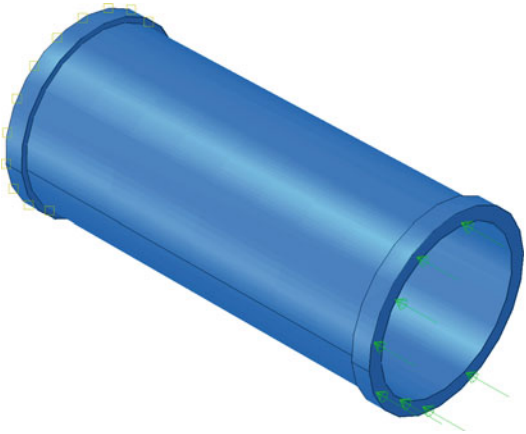


Figure 9 Project domain and its boundary conditions

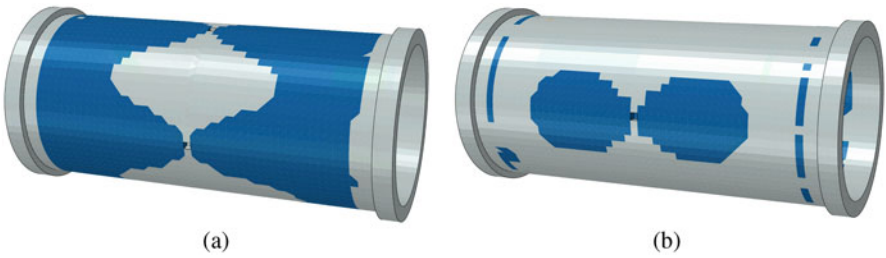


Figure 10 Optimal configuration results. (**a**) Optimal configuration $v_f = 47\%$. (**b**) Optimal configuration $v_f = 80\%$

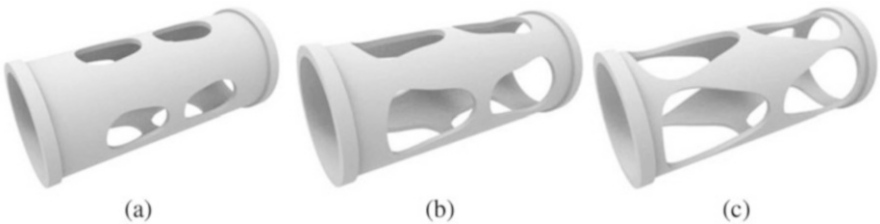


Figure 11 Final topological designs for the thermal isolator. (**a**) $vf = 80\%$. (**b**) $vf = 65\%$. (**c**) $vf = 50\%$

order to study the effect that each configuration of the thermal isolator has in the whole thermal behavior of the engine during operating conditions. Also a structural validation by using computational simulation under operational conditions will be carried out.

Figure 11 shows thermal isolator designs based on the optimal topological configurations for volume fractions of 80%, 65%, and 50%.

Table 1 Number of nodes and elements of used meshes

Volume fraction v_f	Nodes	Elements
1.00	78,226	359,453
0.80	82,272	372,617
0.65	81,016	368,001
0.50	84,935	385,481

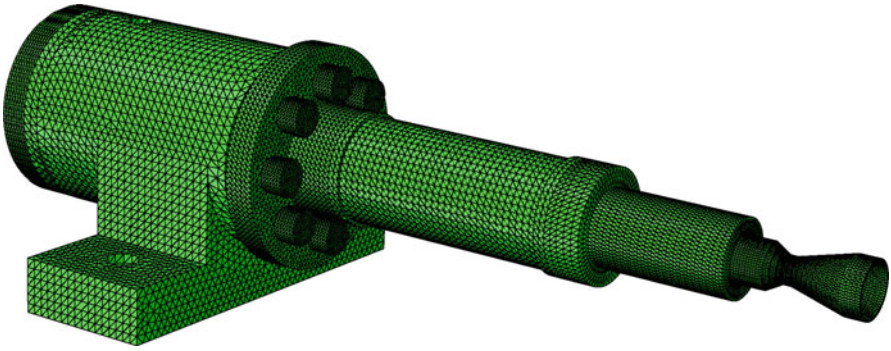


Figure 12 Finite element mesh of the *reference* configuration of the thermal isolator

All configurations were discretized using 4-node-tetrahedral elements, and the same mesh is used to carry out both thermal and structural simulations. The number of nodes and elements used for each configuration is listed in Table 1 and the *reference* configuration mesh is shown in Figure 12.

4.1 Heat Transfer Validation

A heat transfer simulation of the whole assembly is performed in order to validate the optimal configuration under operating conditions. The constitutive properties of the material are presented in Figure 7. To determine the heat loss, the following reference temperature values are assumed: $\theta_{surf} = 313\,\text{K}$ and $\theta_{vac} = 15\,\text{K}$, where θ_{surf} and θ_{vac} are the external surface and the outer space temperature, respectively, see Figure 13. A view factor of f_v =1 is adopted to model the heat loss due to radiation while it will be assumed that only the faces exposed to the outer space will take part in the process.

The net energy ΔH released during the chemical reaction presented early depends fundamentally on the percentage of ammonia dissociated which can be calculated as (see [9, 10, 31, 32]):

$$\Delta H = 3484 - (1910X)\ \text{J/g} \tag{29}$$

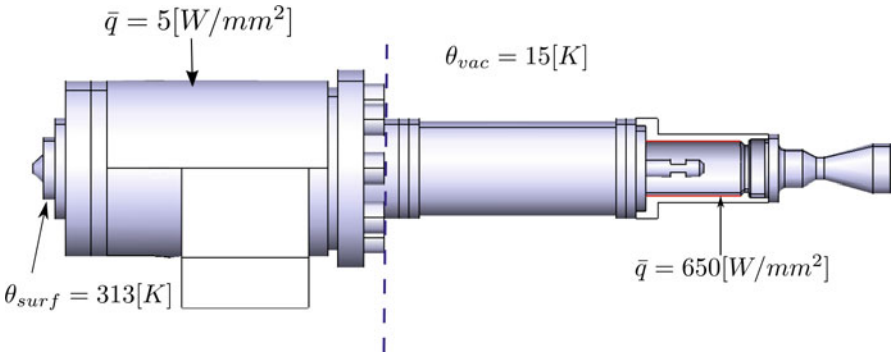


Figure 13 Model to be simulated with its boundary conditions

where X is used to denote the percentage of ammonia dissociated. The amount of ammonia dissociated depends on the length of the catalytic bed and can reach 40% for *high performance thrusters*. Therefore the following is obtained:

$$\Delta H = 2720 \text{ J/g} \tag{30}$$

The nominal fuel flow, denoted as $\dot{m}$, of the engine under study is 0.8 g/s; then the heat flow Q generated within the catalytic bed is

$$Q = \Delta H \dot{m} = 2176 \text{ W}. \tag{31}$$

The heat generated by the chemical reaction presented in Equations (29)–(31) will be considered as a thermal flux $\bar{q}$ distributed in the internal surface of the catalytic bed.

The boundary conditions were determined based on the following considerations:

1. a temperature of 313 K on the left surface, since the operation of this type of engine is by pulses and the analysis will correspond to a stationary state;
2. a thermal flux $\bar{q} = 650 \text{ W/mm}^2$ is adopted on the exposed surface of the catalytic bed.
3. a thermal flux $\bar{q} = 5 \text{ W/mm}^2$ on the upper part of the engine;
4. an exterior temperature of 15 K on the isolator zone and the nozzle.

The model of the monopropellant thruster to simulate (with its boundary conditions) is shown in Figure 13.

Figures 14, 15, and 16 illustrate the results of the simulation with the original design of the thermal isolator. For comparative reasons, the results of this simulation will be taken as a reference solution.

From Figure 16, it can be seen that the temperature in the seal housing area is in the range of 163–178 °C, with the critical temperature being 200 °C.

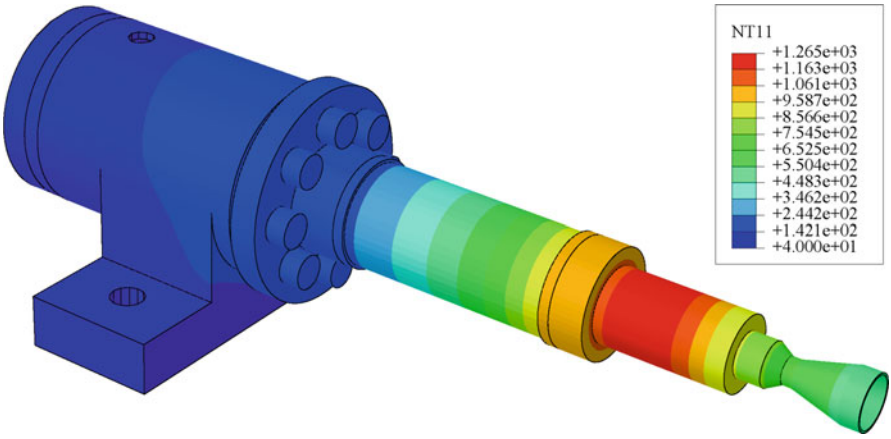


Figure 14 Full view. Maximum (in red): $\theta = 1265\,^{\circ}\text{C}$—minimum (in blue): $\theta = 40\,^{\circ}\text{C}$

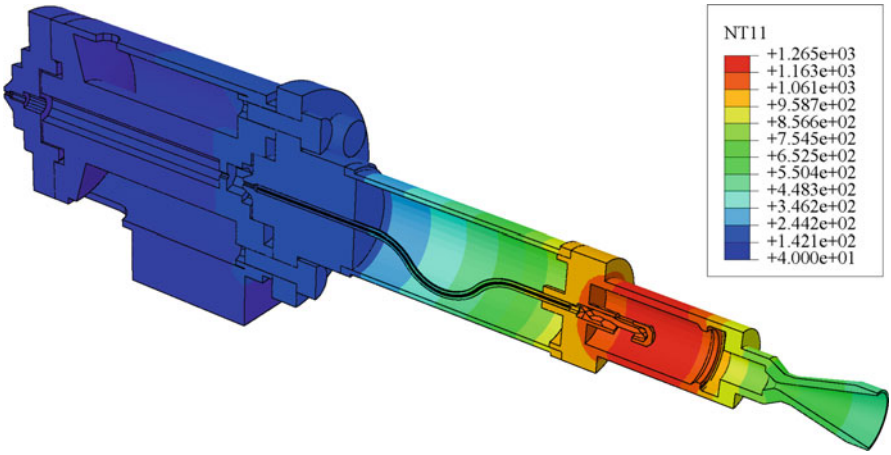


Figure 15 Section view. Maximum (in red): $\theta = 1265\,^{\circ}\text{C}$—minimum (in blue): $\theta = 40\,^{\circ}\text{C}$

In Figures 17, 18, and 19, the temperature distribution obtained by considering an optimal configuration isolator with 80% volume fraction is shown.

The maximum temperature reached for the case of a thermal isolator with $v_f = 80\%$ of material decreased to 159.3 °C, being 66 °C lower than the maximum obtained with the reference solution, see Figure 14. Also the temperature in the seal housing is below 120 °C.

Figures 20, 21, and 22 show the results of the assembly simulation containing the 65% volume fraction of the original configuration of the isolator. For this case, the maximum temperature reached is 132.8 °C, being 92 °C below the reference solution. The temperature in the housing seal area is near to 100 °C, approximately 38% lower than the reference solution, see Figure 22.

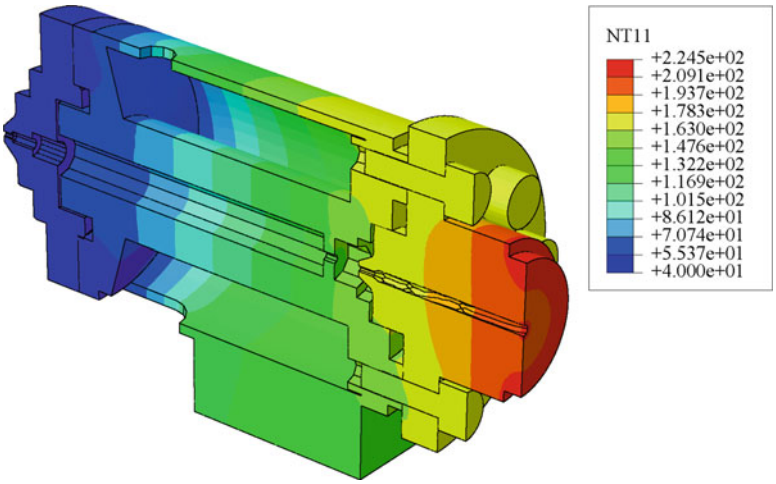


Figure 16 Soft seal zone. Maximum (in red): $\theta = 224.5\,^{\circ}\text{C}$—minimum (in blue): $\theta = 40\,^{\circ}\text{C}$

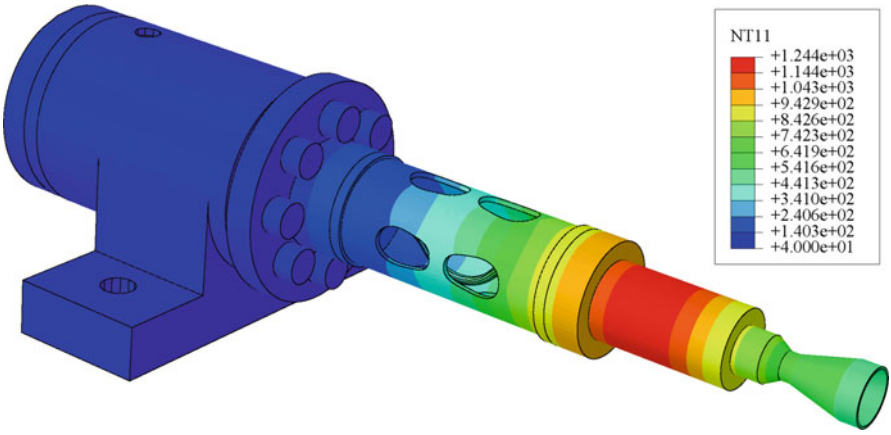


Figure 17 Full view with thermal isolator with $v_f = 80\%$. Maximum (in red): $\theta = 1244\,^{\circ}\text{C}$—minimum (in blue): $\theta = 40\,^{\circ}\text{C}$

Finally, the results obtained for a thermal isolator with 50% of the original amount of material are presented in Figures 23, 24, and 25. The simulation shows that the temperature on the interest zone is 85 °C (78°C below the reference configuration), which represents a 48% decrease, see Figure 25.

The obtained surface temperature profile of the whole assembly is presented in Figure 26, where all the optimized configurations have been considered and compared with the reference solution.

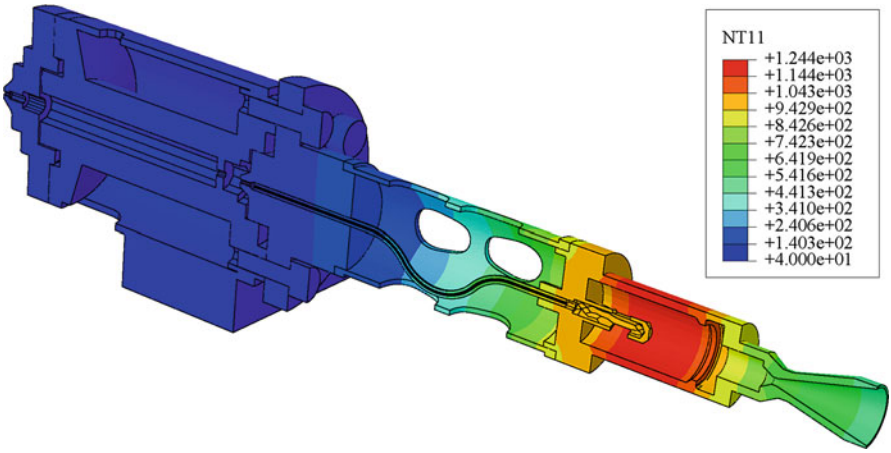


Figure 18 Section view with thermal isolator with $v_f = 80\%$. Maximum (in red): $\theta = 1244\,^{\circ}\text{C}$—minimum (in blue): $\theta = 40\,^{\circ}\text{C}$

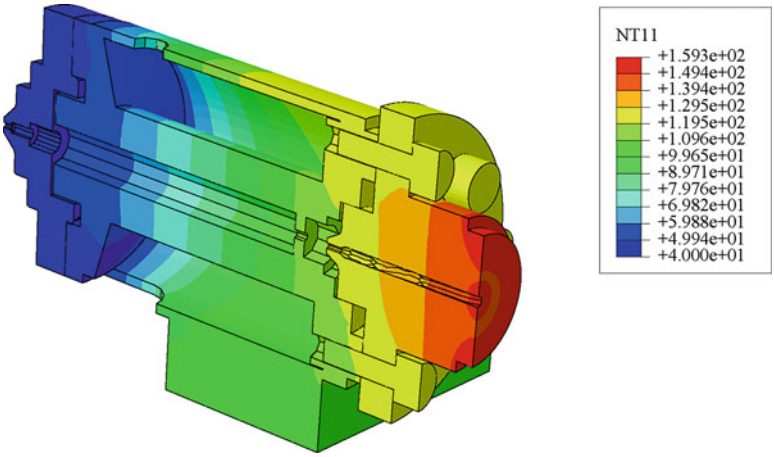


Figure 19 Soft seal zone. Maximum (in red): $\theta = 159.3\,^{\circ}\text{C}$—minimum (in blue): $\theta = 40\,^{\circ}\text{C}$

From the simulations carried out for the three cases of volume fraction, it is clear that when the amount of material decreases the temperature on the seal housing region does too. Therefore the main objective for the topological optimization of the thermal isolator was fulfilled. The maximum temperature reached in the region of interest in all configurations does not exceed 120 °C, which results satisfactory since the soft seals will not run the risk of premature degradation.

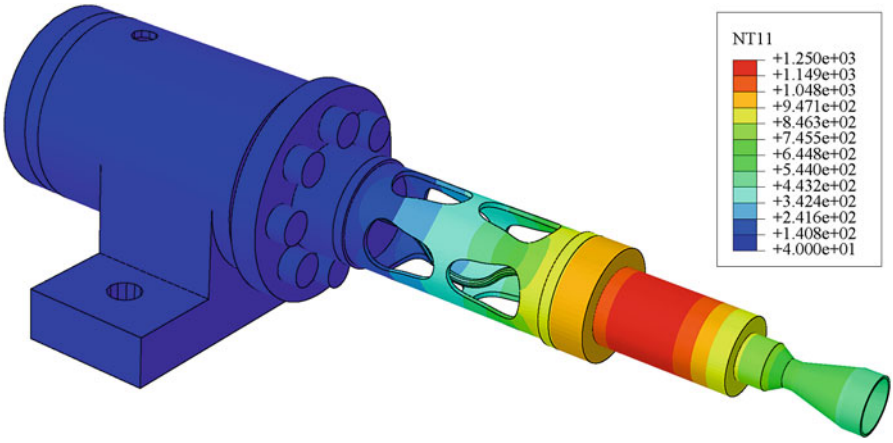


Figure 20 Full view with thermal isolator with $v_f = 65\%$. Maximum (in red): $\theta = 1250\,^{\circ}\text{C}$—minimum (in blue): $\theta = 40\,^{\circ}\text{C}$

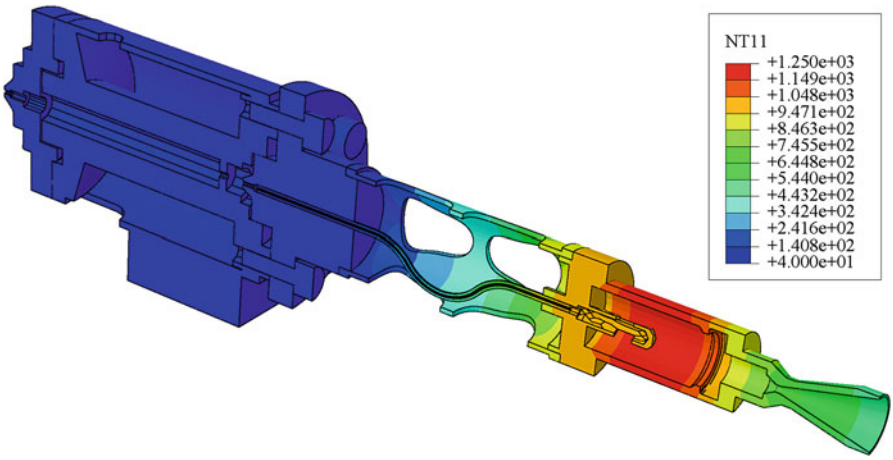


Figure 21 Section view with thermal isolator with $v_f = 65\%$. Maximum (in red): $\theta = 1250\,^{\circ}\text{C}$—minimum (in blue): $\theta = 40\,^{\circ}\text{C}$

4.2 Structural Validation

Since these types of components need to meet certain structural strength criteria, as an integral part of a more complex mechanical system, the mechanical behavior of the thruster must be analyzed in this section. Loads generated during the operation of the monopropellant thruster are quite low, about 1.5 N approximately, and the generated stresses could be negligible. However, during launch stage, the thermal isolator must withstand a static load equivalent to 42 times the gravity load. The

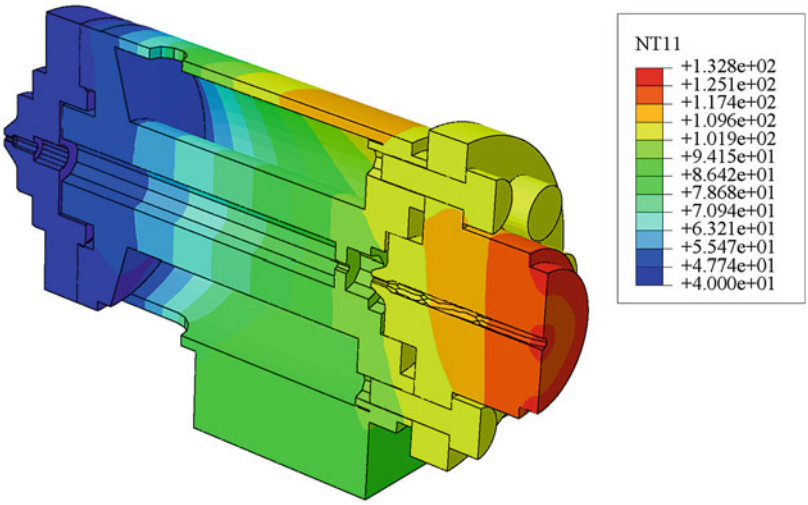


Figure 22 Soft seal zone. Maximum (in red): $\theta = 132.8\,°\text{C}$—minimum (in blue): $\theta = 40\,°\text{C}$

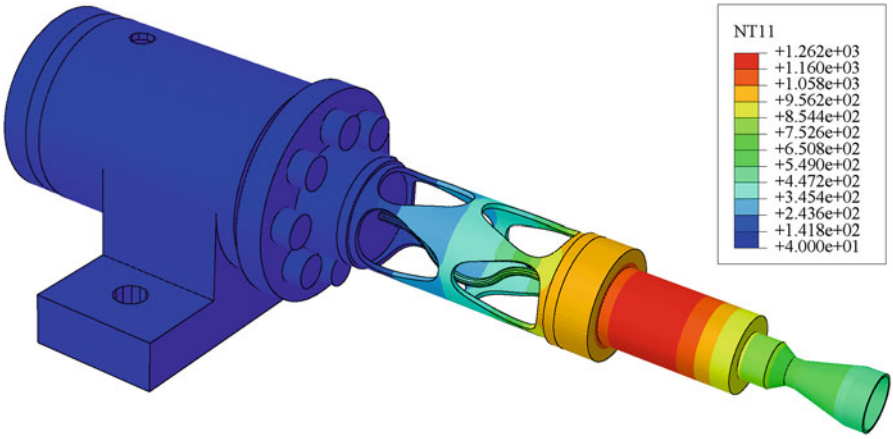


Figure 23 Perspective view with thermal isolator with $v_f = 50\%$. Maximum (in red): $\theta = 1262\,°\text{C}$— minimum (in blue): $\theta = 40\,°\text{C}$

assembly shall be simulated under the above conditions for two different positions of the load vector:

- Load case (I): vertical direction.
- Load case (II): horizontal direction.

The boundary conditions considered for the computational structural simulations can be seen in Figure 27.

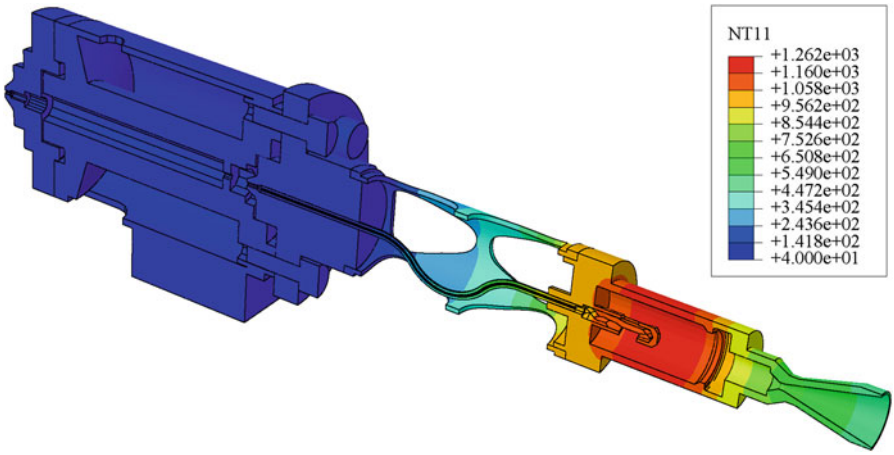


Figure 24 Section view with thermal isolator with $v_f = 50\%$. Maximum (in red): $\theta = 1262\,^{\circ}\text{C}$—minimum (in blue): $\theta = 40\,^{\circ}\text{C}$

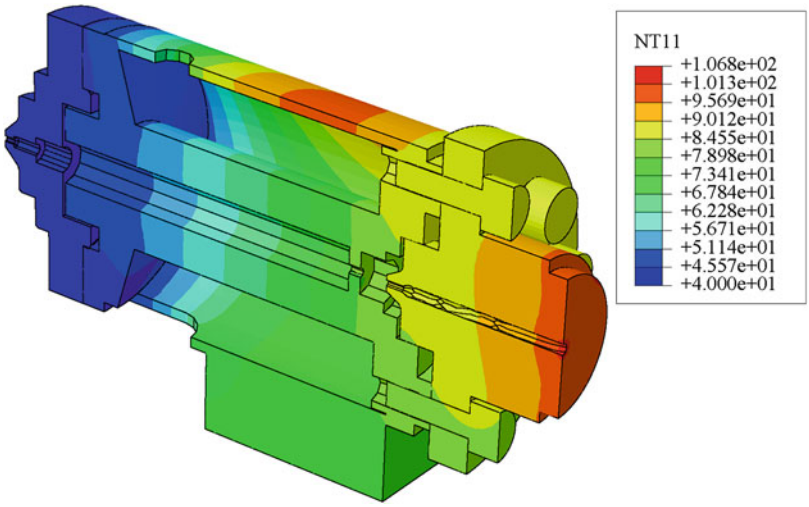


Figure 25 Soft seal zone. Maximum (in red): $\theta = 106.8\,^{\circ}\text{C}$—minimum (in blue): $\theta = 40\,^{\circ}\text{C}$

From these simulations, it can be analyzed the impact of the optimized topology in the structural behavior of the assembly. Although the simulation of the structural behavior was carried out with the complete assembly, only the stress distribution corresponding to the thermal isolator will be shown. In Figures 28, 29, and 30, the results of the simulations for the three optimized configurations are presented. The von Mises failure criterion will be considered for the structural validation. For this criterion, the von Mises stress should not exceed the limit value of the yield stress of INCONEL600, namely $F_y = 280\,\text{MPa}$.

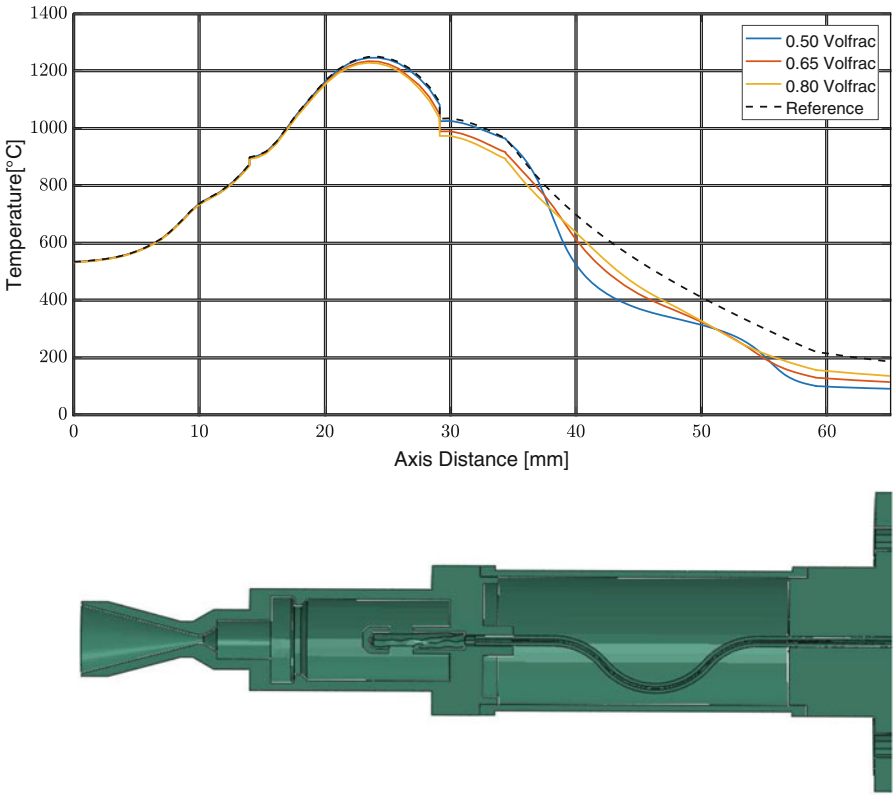


Figure 26 Surface temperature profile of the evaluated configurations

From the results shown in Figures 28, 29, and 30, it can be observed that the von Mises stress values reached are below the yield stress of the material. The structural strength safety factors for each configuration of the thermal isolator are: 38 ($v_f = 80\%$), 17 ($v_f = 65\%$), and 6.6 ($v_f = 50\%$).

5 Final Remarks

The topological optimization and numerical simulation of a thermal isolator of a monopropellant thruster for space applications have been presented. The topological optimization was carried out by considering the thermal behavior of the component. The used optimization procedure is based on the concept of topological derivative.

The objective function was formulated to minimize the heat transfer towards the soft seal of the thruster. For this purpose, minimizing the heat conduction of the thermal isolator and, simultaneously, maximizing the dissipation towards the outer

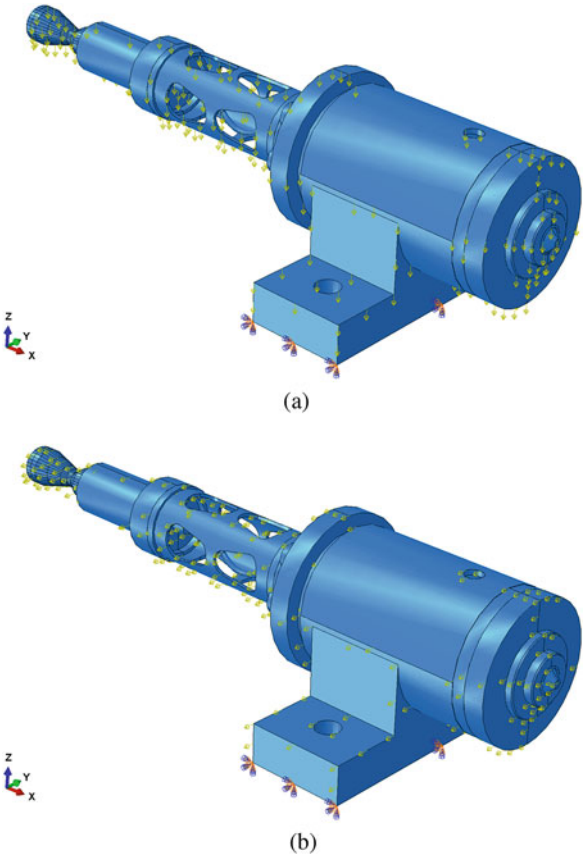


Figure 27 Boundary conditions for the structural validation. (**a**) Load case (I). (**b**) Load case (II)

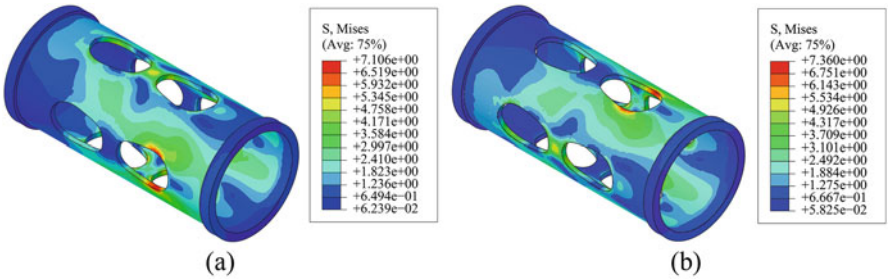


Figure 28 von Mises stress (in MPa) with thermal isolator with $v_f = 80\%$. (**a**) Load case (I). Maximum (red): 7.10—minimum (blue) 6.23×10^{-4}. (**b**) Load case (II). Maximum (red): 7.36—minimum (blue) 5.82×10^{-2}

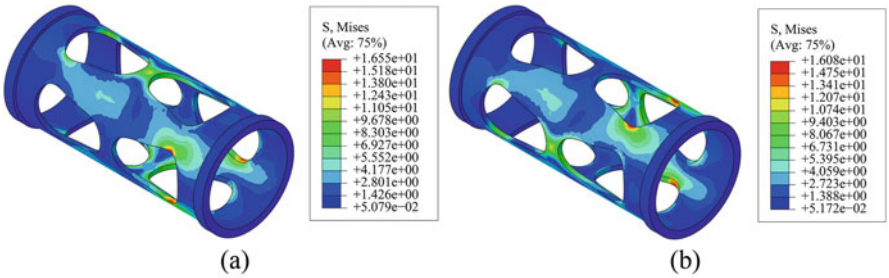


Figure 29 von Mises stress (in MPa) with thermal isolator with $v_f = 65\%$. (**a**) Load case (I). Maximum (red): 16.50—minimum (blue) 5.07×10^{-2}. (**b**) Load case (II). Maximum (red): 16.08—minimum (blue) 5.17×10^{-2}

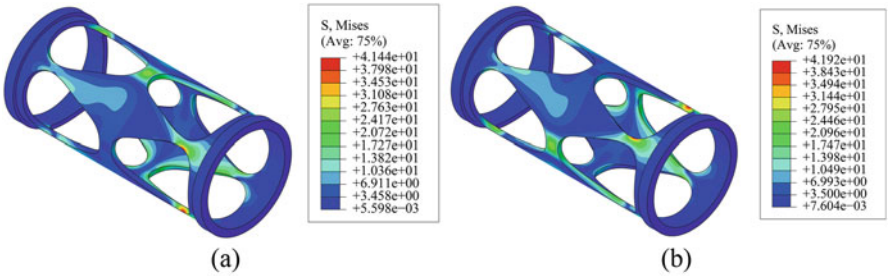


Figure 30 von Mises stress (in MPa) with thermal isolator with $v_f = 50\%$. (**a**) Load case (I). Maximum (red): 41.44—minimum (blue) 5.59×10^{-3}. (**b**) Load case (II). Maximum (red): 41.92—minimum (blue) 7.60×10^{-3}

space in the form of radiation has been proposed. From the optimization procedure it was obtained that the lower volume fraction of material is 50%. Optimal designs with values less than 50% result in thermal isolators that are unviable from the constructive point of view.

From the thermal simulations under operating conditions, the obtained optimal configurations for the isolator fulfill the requirements of reducing the thermal flux from the catalytic bed. The obtained temperature values in the proximal zone of the soft seal are below the operative limit value. In the worst case, configuration with $v_f = 80\%$, the developed temperature is 11% lower than the value of the reference configuration (without optimization).

The results of the structural simulations in all the optimized configurations are satisfactory. The adopted failure criterion based on the von Mises stress is adequate for this kind of material. Also, the obtained stress distribution indicates that the maximum values of von Mises stress is lower than the established limit value for INCONEL600, with a minimum safety factor of 6.6.

Finally, from the present analysis, it is possible to conclude that the design of this type of engine is driven by the thermal dissipation criterion, rather than the structural behavior.

Acknowledgements This research was partially supported by PID-UTN (Research and Development Program of the National Technological University, Argentina) and CONICET (National Council for Scientific and Technical Research, Argentina). The supports of these agencies are gratefully acknowledged.

References

1. Allaire, G., Jouve, F., Van Goethem, N.: Damage and fracture evolution in brittle materials by shape optimization methods. Chin. J. Comput. Phys. **230**(12), 5010–5044 (2011)
2. Amstutz, S.: Sensitivity analysis with respect to a local perturbation of the material property. Asymptot. Anal. **49**(1–2), 87–108 (2006)
3. Amstutz, S., Andrä, H.: A new algorithm for topology optimization using a level-set method. J. Comput. Phys. **216**(2), 573–588 (2006)
4. Amstutz, S., Novotny, A.A.: Topological optimization of structures subject to von Mises stress constraints. Struct. Multidiscip. Optim. **41**(3), 407–420 (2010)
5. Amstutz, S., Giusti, S.M., Novotny, A.A., de Souza Neto, E.A.: Topological derivative for multi-scale linear elasticity models applied to the synthesis of microstructures. Int. J. Numer. Methods Eng. **84**, 733–756 (2010)
6. Amstutz, S., Novotny, A.A., de Souza Neto, E.A.: Topological derivative-based topology optimization of structures subject to Drucker-Prager stress constraints. Comput. Methods Appl. Mech. Eng. **233–236**, 123–136 (2012)
7. Cagliolo, C.M., Salomone, J.E., Jazni, J.E., Lagier, J.E.: Analysis of a thermal behaviour of the fuel supply pipe in a low thrust monopropellant engines. In: 4th Argentinian Congress in Space Technology (CATE), Buenos Aires (2007)
8. Çengel, Y.A., Ghajar, A.J.: Heat and Mass Transfer. Fundamentals and Applications, 4th edn. McGraw Hill, New York (2011)
9. Emmons, D.L.: Design criteria manual for long-life monopropellant engines. Technical Report AFRPL-TR-77-6, Rocket Research Company, Redmond (1977)
10. Gagosian, J.: Hydrazine handbook. Technical Report, Rocket Research Company, Redmond (1993)
11. Giusti, S.M., Novotny, S.M.: Design of bi-metallic devices based on the topological derivative concept. Mech. Res. Commun. **65**, 1–8 (2015)
12. Giusti, S.M., Novotny, A.A., de Souza Neto, E.A., Feijóo, R.A.: Sensitivity of the macroscopic elasticity tensor to topological microstructural changes. J. Mech. Phys. Solids **57**(3), 555–570 (2009)
13. Giusti, S.M., Novotny, A.A., de Souza Neto, E.A.: Sensitivity of the macroscopic response of elastic microstructures to the insertion of inclusions. Proc. R. Soc. A Math. Phys. Eng. Sci. **466**, 1703–1723 (2010)
14. Giusti, S.M., Mello, L.M.A., Silva, E.C.N.: Piezoresistive device optimization using topological derivative concepts. Struct. Multidiscip. Optim. **50**, 453–464 (2014)
15. Giusti, S.M., Ferrer, A., Oliver, J.: Topological sensitivity analysis in heterogeneous anisotropic elasticity problem. Theoretical and computational aspects. Comput. Methods Appl. Mech. Eng. **311**, 134–150 (2016)
16. Giusti, S.M., Mróz, Z., Sokolowski, J., Novotny, A.A.: Topology design of thermomechanical actuators. Struct. Multidiscip. Optim. **55**, 1575–1587 (2017)
17. Hintermüller, M., Laurain, A.: Multiphase image segmentation and modulation recovery based on shape and topological sensitivity. J. Math. Imaging Vision **35**, 1–22 (2009)
18. Hintermüller, M., Laurain, A., Novotny, A.A.: Second-order topological expansion for electrical impedance tomography. Adv. Comput. Math. **36**(2), 235–265 (2012)
19. Lions, J.L., Magenes, E.: Problèmes aux limites non homogènes et applications, vols. 1–2. Dunod, Paris (1968)

20. Lopes, C.G., Santos, R.B., Novotny, A.A.: Topological derivative-based topology optimization of structures subject to multiple load-cases. Latin Am. J. Solids Struct. **12**, 834–860 (2015)
21. Nash, S.G., Griva, I., Sofer, A.: Linear and Nonlinear Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2009)
22. Novotny, A.A., Sokołowski, J.: Topological Derivatives in Shape Optimization. Interaction of Mechanics and Mathematics. Springer, Berlin (2013)
23. Osher, S., Sethian, J.A.: Front propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys. **78**, 12–49 (1988)
24. Rao, S.S.: Engineering Optimization. Theory and Practice, 4th edn. Wiley, New Jersey (2009)
25. Salençon, J.: Handbook of Continuum Mechanics. General Concepts Thermoelasticity. Springer, Berlin (2001)
26. Sales, V., Novotny, A.A., Rivera, J.M.: Energy change to insertion of inclusions associated with the Reissner-Mindlin plate bending model. Int. J. Solids Struct. **59**, 132–139 (2013)
27. Salomone, J.E., Giusti, S.M., Sanna, A.D., Micheloud, P.D., Manavella, L.: Study and optimization of structural components of low thrust monopropellants engine. In: 3th Argentinian Congress in Mechanical Engineering (CAIM), Buenos Aires (2012)
28. Salomone, J.E., Lagier, J.E., Cova, W.J.D., Gonzalez, G.J., Yasielski, R.: Development of a monopropellant thruster of 1.5n to control the orbit and attitude of satellites. In: 4th Argentinian Congress in Space Technology (CATE), Buenos Aires (2007)
29. Salomone, J.E., Giusti, S.M., Sanna, D.A., Gonzalez, G.J., Cova, W.J.D.: Thermostructural simulation and optimization of a monopropellant engine for space application. In: 6th Argentinian Congress in Space Technology (CATE), La Punta (2011)
30. Sokołowski, J., Żochowski, A.: On the topological derivative in shape optimization. SIAM J. Control Optim. **37**(4), 1251–1272 (1999)
31. Sutton, G.P.: Rocket Propulsion Elements. Wiley, New York (1963)
32. Sutton, D.: Hydrazine thrusters for space applications. J. Br. Interplanet. Soc. **25**, 537–551 (1972)
33. Taylor, R.L., Zienkiewicz, O.C.: The Finite Element Method. Volume 1: The Basis, 5th edn. Butterworth-Heinemann, Oxford (2000)
34. Van Goethem, N., Novotny, A.A.: Crack nucleation sensitivity analysis. Math. Methods Appl. Sci. **33**(16), 197–1994 (2010)

Evidence-Based Robust Optimization of Pulsed Laser Orbital Debris Removal Under Epistemic Uncertainty

Liqiang Hou, Massimiliano Vasile, and Zhaohui Hou

Abstract An evidence-based robust optimization method for pulsed laser orbital debris removal (LODR) is presented. Epistemic type uncertainties due to limited knowledge are considered. The objective of the design optimization is set to minimize the debris lifetime while at the same time maximizing the corresponding belief value. The Dempster–Shafer theory of evidence (DST), which merges interval-based and probabilistic uncertainty modeling, is used to model and compute the uncertainty impacts. A Kriging based surrogate is used to reduce the cost due to the expensive numerical life prediction model. Effectiveness of the proposed method is illustrated by a set of benchmark problems. Based on the method, a numerical simulation of the removal of Iridium 33 with pulsed lasers is presented, and the most robust solutions with minimum lifetime under uncertainty are identified using the proposed method.

1 Introduction

The vast majority of the objects larger than 1 cm diameter in low-Earth orbit consists of space debris and remnants of larger man-made objects [14, 16]. Some of them will drop out of orbit and finally re-enter the atmosphere due to atmospheric effects, but most of them may stay in the orbit for numbers of years if no operation of debris removal is implemented. In the past decades, the amount of space

L. Hou (✉)
Xian Satellite Control Center, Xian, China
e-mail: houliqiang2008@gmail.com

M. Vasile
Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow, UK
e-mail: massimiliano.vasile@strath.ac.uk

Z. Hou
School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_7

debris orbiting Earth has grown exponentially. The growing space debris seriously threatens the safety of spacecrafts on orbit, especially in the near-Earth space, in which the probability of collision with debris increases steadily. To tackle the issues, researchers proposed a series of methods to reduce the threat of debris by collision avoidance maneuver and active debris removal. Recent proposals to reduce the amount of debris include electrical currents, slingshots, solar sails, electro dynamic tethers, etc. Some of these proposals are still hypothetical, some of them are operational and will be implemented in the near future. Among them the pulsed laser orbital debris removal (LODR) has attracted the attention of many researchers, whose feasibility has been demonstrated in some recent missions.

The ground-based laser orbital debris removal (LODR) system delivers a pulsed laser on the target object to deorbit the target. The pulsed laser is delivered at a pre-determined angle such that the plasma jet generated by the laser slows down the target's velocity and lowers its perigee altitude. A typical project using LODR includes the Orion Project, conducted by NASA and the USAF in 1995–1996. In [13], effects of irregular shapes on the laser debris removal are investigated. In [14], a method of debris collision avoidance using laser radiation is proposed. Simulation results show that promising results of range displacement for the LEO debris objects could be achieved.

In this chapter, a robust optimization method for determining the targets of pulsed laser orbital debris removal (LODR) under uncertainty is presented. Due to limited knowledge, actual values of the debris characteristics parameters, e.g., mass, dimensions, and ablation rate, may not be available before the design optimization. In most cases, the information on the uncertainties are given in the form of interval distributions. To minimize the impact of the uncertainties, an evidence-based robust design optimization method can be used. The robust design optimization takes into account the epistemic uncertainties, optimizes the LODR performance, and maximizes its belief value [22]. A numerical lifetime predictor is used to evaluate lifetimes of the debris before and after the laser pulses.

Robust design optimization is complex and expensive. In design optimization of LODR, the uncertainties, such as the dimensions and masses of the debris, are hard to be modeled using the conventional continuous distribution model. With DST, both the epistemic and aleatory uncertainty can be properly modeled [6, 27]. However, the number of focal elements needed to be explored in the design optimization grows exponentially with the dimension of the uncertain space, and soon becomes prohibitively expensive if a number of uncertainties are involved [21]. Another problem in the design optimization relates to lifetime prediction of the debris in the LODR. The high accuracy numerical debris lifetime prediction model is computationally expensive, and is repetitively used during the search of the optimal solutions. Therefore, a strategy that reduces the computational cost due to the numerical lifetime prediction, while maintaining the accuracy, is required.

The evidence-based robust design optimization of LODR is formulated as a multi-objective optimization problem (MOP). Unlike the conventional design optimization, with the evidence-based robust optimization, a step-like Pareto front should be obtained. Therefore, some new algorithms and techniques for the design optimization should be developed. Accordingly, the remainder of the chapter is

structured as follows: Sect. 2 presents the general framework of design optimization of pulsed LODR. In Sect. 3, a multi-objective optimization (MOO) algorithm with Tchebysheff decomposition and Gaussian Predictor is developed. A Kriging based surrogate is used to reduce the cost due to the numerical debris lifetime prediction. Discussions of the strategy to incorporate the evidence computation into the design optimization are presented in this section too. Section 4 presents a numerical simulation of a LODR example problem under epistemic uncertainties, and finally Sect. 5 concludes the book chapter.

2 Design Optimization of Laser Orbital Debris Removal (LODR)

2.1 Pulsed LODR

Figure 1 shows schematically the geometry of laser–target interaction of pulsed LODR and debris. Series of multi-kJ laser pulses are delivered at the target debris to decelerate the debris velocity and finally re-enter it into the atmosphere. The velocity variation obtained with the pulse laser is [13, 17]

$$\Delta v = \frac{\eta_0 C_m \Phi}{\mu} \tag{1}$$

where η_0 is the efficiency factor for the combined effects of improper thrust direction, target shape, target tumbling, etc., μ is the target areal mass density, Φ is

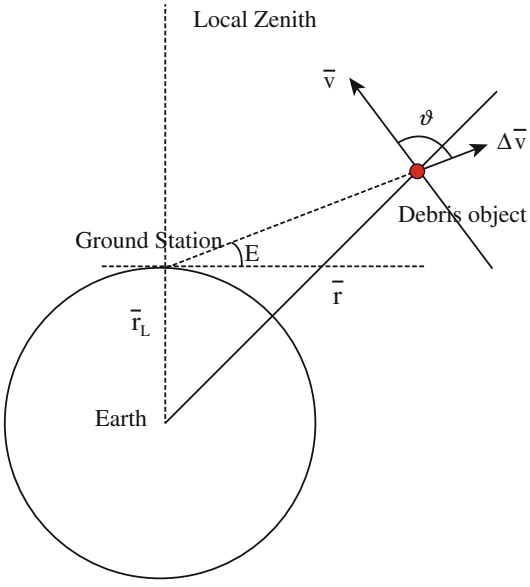


Figure 1 Geometry of laser–target interaction of pulsed LODR: the laser pulses are delivered at the target to lower its perigee altitude. During the LODR, the angle ϑ is greater than a specified value

the laser fluence, and C_m is the mechanical coupling coefficient to the laser pulse energy and defined as [17, 20]

$$C_m = \frac{p\tau}{\Phi} = \frac{p}{I} \quad (\mathrm{PaW^{-1}m^{-2}} \quad \text{or} \quad \mathrm{N/W}) \tag{2}$$

where p is the ablation pressure delivered to the target, I is the laser intensity, τ is the duration of the pulse, and Φ is the laser fluence.

The laser pulses are delivered at the target when the LODR station finds the target, and the angle between the debris velocity and the laser beam

$$\vartheta = \arccos\left(\frac{|(\mathbf{r} - \mathbf{r}_l) \cdot \mathbf{v}|}{|\mathbf{r} - \mathbf{r}_l| \times |\mathbf{v}|}\right) \tag{3}$$

is greater than a specified value, e.g., 100° , where $\mathbf{r}_l$ is the position of LODR station, and $\mathbf{r}$ and $\mathbf{v}$ are the debris position and velocity, respectively.

The Δv obtained with Equation (1) is aligned with laser beam, and the velocity variation in radial–transversal coordinates $\Delta\mathbf{v} = [\Delta v_r, \Delta v_\theta, \Delta v_n]$ due to the laser pulse can be computed as

$$\begin{bmatrix} \Delta v_r \\ \Delta v_\theta \\ \Delta v_n \end{bmatrix} = \mathbf{M}_{oi} \frac{(\mathbf{r} - \mathbf{r}_l)}{|\mathbf{r} - \mathbf{r}_l|} \Delta\mathbf{v} \tag{4}$$

where $\mathbf{M}_{oi}$ is the transformation matrix from the earth-centered inertial (ECI) reference frame to the radial–transversal coordinate system, and $\mathbf{r}_l$ is the position of the ground station in ECI frame.

Given the velocity variation in radial–transversal coordinates, variations of the orbital elements due to the velocity increment can be computed ([19], Ch.12). Repeating the process until the LODR is stopped, the debris orbital elements after the LODR can be obtained. The orbit propagator can then be used to predict the debris lifetimes after the LODR.

2.2 Debris Lifetime Prediction

A numerical orbit propagator model can be used to predict the debris lifetime after LODR. In the numerical model, influences of the gravitational and non-gravitational forces, including non-spherical Earth, third-bodies, atmospheric drag, solar radiation pressure (SRP), etc., are considered.

The acceleration due to solar radiation pressure is given by

$$\mathbf{a} = -C_r \frac{A}{m} \frac{K\varphi_s}{c} \left(\frac{1}{R_{AU}}\right)^2 \mathbf{r}_s \tag{5}$$

where C_r, ϕ_s, and c are the coefficient of reflectivity, solar flux at 1 AU, and the speed of light, respectively. K represents the percentage of the sun, as seen from the object (usually 1.0), R_{AU} is the distance from the object to the sun in AU, and $\mathbf{r}_s$ is the unit position vector of the sun, as seen from the debris.

The acceleration due to atmospheric drag is given by

$$\mathbf{D} = \frac{1}{2} C_D \frac{A}{m} \rho v^2 \frac{\mathbf{v}}{|\mathbf{v}|} \tag{6}$$

where C_D, A, and m are the drag coefficient, cross-sectional area, and mass, respectively. $\mathbf{v}$ is the velocity of the debris relative to the atmosphere, and ρ is the atmospheric density at the location of the debris.

The drag coefficient is a dimensionless quantity which describes the interaction of the atmosphere with the surface material. The coefficient depends on several factors, including the debris shape, orientation, composition, and temperature of the atmosphere, etc. A crude approximation is $C_D = 2.2$ for compactly shaped debris [14]. A variable step-size numerical integrator is used to integrate the high fidelity dynamic equations. Error control techniques are used to manage the integration step size, increasing the computational efficiency, while preserving accuracy [19].

Although some numerical techniques, such as the variable step-size numerical integrator, can be used for predicting the lifetimes of debris, the overall computational cost of the optimization process is expensive because one needs to compute the debris lifetimes and evaluate uncertainty impacts repetitively during the design optimization. Therefore, the cost will be huge if the numerical lifetime predictor is implemented directly in the design optimization.

3 Evidence-Based Robust Design Optimization

3.1 Epistemic Uncertainties and Evidence Computation

The uncertainties of debris parameters include the cross-sectional area and masses. Uncertainties of laser characteristic parameters, e.g., efficiency factor and coupling coefficient, should be taken into account too. Due to a lack of knowledge, accurate distribution of the uncertainties cannot be given in advance of the design optimization. The estimated values may be from different sources, and have different belief levels too. Therefore, in this work, an evidence-based tool is used to quantify the uncertainties. Based on the evidence theory, an evidence-based robust design optimization method is proposed.

In evidence theory, beliefs of uncertain parameters $\mathbf{u}$ are supposed to be within various intervals given by the experts. The level of confidence an expert has on an elementary proposition E on the set $\mathscr{U}$ is quantified using the basic probability assignment (BPA). The BPA $m(E)$ satisfies the three following axioms [1, 6]:

Table 1 Example of BPA structure of the interval-based evidence sets

Parameter	Expert	Estimated uncertain intervals ($[U_l, U_u]$)	$BPA(U)$
Mass (Kg)	A	[1.3 , 1.5]	0.7
	B	[1.2 , 1.7]	0.2
	C	[1.4 , 1.8]	0.1
Ablation rate (Kg/J)	A	$[76 \cdot 10^{-9}, 80 \cdot 10^{-9}]$	0.3
	B	$[60 \cdot 10^{-9}, 78 \cdot 10^{-9}]$	0.5
	C	$[65 \cdot 10^{-9}, 85 \cdot 10^{-9}]$	0.2

Estimates of the intervals of the debris mass and ablation rate are given by three experts. Corresponding confidence levels or BPAs of the intervals are listed in the table. The BPA values show the experts' confidence in the proposition of the uncertain intervals

$$\begin{cases} m(E) \geq 0, & \forall E \in \mathscr{U} \\ m(\emptyset) = 0, & \text{and} \\ \sum\limits_{E \in \mathscr{U}} m(E) = 1 \end{cases} \tag{7}$$

Table 1 shows an example of BPA structure of uncertainties of LODR. The physical properties of debris, average mass, and ablation rate can differ considerably from one debris to the other. In this table, estimates of uncertain intervals of the debris mass and ablation rate are given in evidence form by three experts. The confidence levels represent the perception that experts have on the specified physical properties of debris:

- Expert A is quite optimistic on his/her estimate interval of the mass. Therefore he/she assigns a high confidence of 70% to the proposition that the mass will be between 1.3 and 1.5 kg; on the other hand, the expert is less confident about the estimate of the ablation rate and assigns a confidence level of 30% that the interval of the ablation rate is between $76 \cdot 10^{-9}$ and $80 \cdot 10^{-9}$.
- While expert B, due to the data available and accuracy levels of the instruments, assigns a comparatively low confidence of 20% to the proposition that the mass will be between 1.2 and 1.7 kg. In contrast, he/she is quite confident on his/her estimate of the laser ablation rate, and the probability assignment of the proposition that the ablation rate is between $60 \cdot 10^{-9}$ and $78 \cdot 10^{-9}$ is set to 50%.
- As for expert C, his/her estimates of the interval of the mass and ablation rate are [1.4, 1.8] and $[65 \cdot 10^{-9}, 85 \cdot 10^{-9}]$, respectively, with the confidence levels of the mass 10% and ablation rate 20%.

Written in mathematical terms, the statements of the estimated intervals given by the three experts can be expressed as Table 1 shows.

An element of $\mathbf{U} \in \mathbb{R}^n$ that has a non-zero BPA is named a focal element (FE). When more than one parameter is considered uncertain (e.g., u_1 and u_2), the BPA of the uncertain space defined by the Cartesian product of single uncertain intervals is the product of the BPA of each interval

$$m(u_1, u_2) \in [a_1, b_1] \times [a_2, b_2] = m(u_1 \in [a_1, b_1]) m(u_2 \in [a_2, b_2]) \tag{8}$$

Two measures quantifying the epistemic uncertain impacts, the belief and plausibility of proposition A, over the frame of discernment $\mathbf{U}$, can then be defined as

$$bel(A) = \sum_{FE \subset A, FE \in \mathbf{U}} m(FE) \tag{9}$$

$$pl(A) = \sum_{FE \bigcap A \neq \emptyset, FE \in \mathbf{U}} m(FE) \tag{10}$$

where $m(FE)$ is the BPA value of the focal element FE, and

$$pl(A) = 1 - bel(\bar{A}) \tag{11}$$

Plausibility and belief can be viewed as the upper and lower probabilities of possible values.

Figure 2 shows a BPA structure of the interval type data given by two experts. Focal elements of the BPA structure, generated by the Cartesian product of single uncertain intervals, are computed and listed in the lower part of the table. The intersection of two intervals is defined by the maximum of the two lower bounds and the minimum of the two upper bounds. With calculation of Dempster's rule, using Equations (8)–(11), cumulative belief and plausibility value of the proposition that the data is less than a specified value can be obtained. Figure 3 shows the belief and plausibility values. As the figure shows, the belief represents confidence that the proposition is true, while the plausibility represents the confidence that the proposition is possible.

Table 2 shows an example of the BPA structure associated to the uncertain parameters for a Gaussian distribution. The evidence data set consists of statistics data values of the debris and ablation rate, and three experts express their own opinion of the estimated intervals of the mean value and standard deviation. BPA

A / Focal Elements / B		Interval	m	Interval	m
		[1, 4]	0.5	[3, 5]	0.5
Interval	m				
[1,4]	0.333333	[1,4]	0.16667	[3,4]	0.16667
[2,5]	0.333333	[2,4]	0.16667	[3,5]	0.16667
[3,6]	0.333333	[3,4]	0.16667	[3,5]	0.16667

Figure 2 BPAs of the interval type data given by two experts. Corresponding focal elements are generated using Equation (8). Intersection of two intervals is defined by the maximum of the two lower bounds and the minimum of the two upper bounds corresponding to the intersection

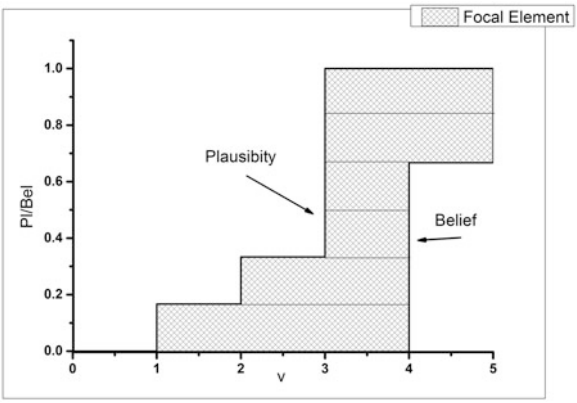


Figure 3 Calculation of Dempster's rule using expression equations (8)–(11). Cumulative plausibility and belief values of the proposition that the data is less than a specified value are computed by adding up the masses associated to the focal elements that partially or totally support the proposition. BPA structure of the data is from Figure 2

Table 2 Another example of BPA structure

Parameter	Description	μ	σ_l	σ_u	BPA
u(kg/J)	Ablation rate	$80 \cdot 10^{-9}$	$10 \cdot 10^{-10}$	$20 \cdot 10^{-10}$	0.40
		$78 \cdot 10^{-9}$	$15 \cdot 10^{-10}$	$22 \cdot 10^{-10}$	0.30
		$72 \cdot 10^{-9}$	$12 \cdot 10^{-10}$	$17 \cdot 10^{-10}$	0.30
m_0(kg)	Average mass [12]	1.30	0.05	0.10	0.30
		1.50	0.05	0.12	0.20
		1.42	0.05	0.08	0.50

This time, the BPA structure consists of estimated values, deviations, and corresponding BPAs. Three experts express their opinion of the estimated interval of the mean value and standard deviation of the uncertain mass and ablation rate

values are assigned to each data set to show the expert's confidence level of the estimated data set. In the table, if $\mu_1^i = \ldots = \mu_m^i$, the parameter μ^i can be removed because the parameter is a deterministic parameter. The table now consists of intervals of σ and associated BPA values. Therefore, the table can be seen as a variation of Table 1. Belief and plausibility of the uncertain intervals can be computed in a way similar to Table 1.

3.2 Formulation of Evidence-Based Robust Design Optimization

Consider an optimization problem under uncertainty

$$\min_{\mathbf{d}\in\mathbf{D},\mathbf{u}\in\mathbf{U}} f(\mathbf{d},\mathbf{u}) \tag{12}$$

where $f(\mathbf{d}, \mathbf{u})$ is the objective function, $\mathbf{d} \in \mathbf{D} \subset \mathbb{R}^n$ are the design variables, and $\mathbf{u} \in \mathbf{U}$ are the interval-based set uncertainties with BPA values. Distribution of the uncertainties is given in the form of BPA structure as Table 1 shows.

To obtain the optimal robust design solutions, one needs to optimize the cost function and maximize its belief under uncertainties at the same time. Therefore, the problem can be formulated as a multi-objective optimization problem (MOP) [6, 27]

$$\begin{cases} \max\limits_{v \in \mathbb{R}, \mathbf{d} \in \mathbf{D}, \mathbf{u} \in \mathbf{U}} Bel(f(\mathbf{d}, \mathbf{u}) < v) \\ \min\limits_{v \in \mathbb{R}} v \end{cases} \tag{13}$$

where v is the threshold to be minimized, and Bel is the belief value.

Note that the evidence metric for the uncertainty impacts is related to the uncertain space only. Therefore, one can decompose the uncertain box first, recombine the focal elements, and compute BPA values of the new focal elements in advance of the optimization. A sampling based approach can be used for this purpose. First, a set of intervals are generated using the sampling approach, new focal elements are then generated and combined using the evidence rules. The number of focal elements is set to be equal to the population size of the design variables, and each focal element is associated to one individual. Following Equations (9) and (11), belief and plausibility of $f(\mathbf{d}, \mathbf{u}) < v$ can be computed as [10, 11]

$$Bel(f(\mathbf{d}, \mathbf{u}) < v) = \sum_{FE \in \mathbf{U} | \forall \mathbf{u} \in FE, f(\mathbf{d}, \mathbf{u}) < v} m(FE) \tag{14}$$

and

$$Pl(f(\mathbf{d}, \mathbf{u}) < v) = \sum_{FE \in \mathbf{U} | \exists \mathbf{u} \in FE, f(\mathbf{d}, \mathbf{u}) < v} m(FE) \tag{15}$$

where FE is the new focal element in the uncertain space. An evidence-based analysis tool is used to sample and recombine the focal elements [3, 18].

3.3 MOO with Tchebysheff Decomposition and Proportional Orthogonal Decomposition

Consider a MOP as follows:

$$\min_{\mathbf{x} \in \mathbf{D}} F(\mathbf{x}) = [f_1(\mathbf{x}), \ldots f_m(\mathbf{x})]^T \quad x \in \mathbf{D} \subset \mathbb{R}^n \tag{16}$$

where $\mathbf{x}$ is the design variable and $f_i(x) \in \mathbb{R}$ with $i =\in \{1, 2, \ldots, m\}$ are the objective functions.

To search the optimal Pareto solutions of Equation (16), the genetic MOO algorithms, e.g., NSGA-II [8], MOPSO, and MOEA/D [26], etc., can be used. However, the computational cost can be expensive as it may take hundreds of iterations of the MOOs to search for the optimal solutions. To resolve this issue, in this work, a new high efficiency MOO with decomposition is developed. The new MOO works as follows.

With the Tchebysheff decomposition, the MOP of Equation (16) can be formulated into a set of scalar single-objective optimization problems [2, 7]

$$\min_{\mathbf{x}\in\mathbf{D}} g\left(f(\mathbf{x}), \mathbf{w}, \mathbf{z}\right) = \min_{\mathbf{x}\in\mathbf{D}} \max_{i=1,\ldots,m} w_i(f_i(\mathbf{x}) - z_i) \tag{17}$$

where $g \in \mathbb{R}$ is the Tchebysheff metric, $\mathbf{w} \in \mathbb{R}^m$ is the weight vector , and $z \in \mathbb{R}^m$ denotes the reference point

$$z_i = \min_{\mathbf{x}\in\mathbf{D}} f_i(\mathbf{x}) \qquad (i = 1, \ldots, m) \tag{18}$$

Initialize the population $\mathbf{x}_j \in \mathbb{R}^n$, $j = 1, \ldots, n_{pop}$ randomly, where n_{pop} is the population size. For each $\mathbf{x}_j^{(i)} \in \mathbf{X}^{(i)}$, where $\mathbf{X}^{(i)}$ is the population at ith iteration, compute the Tchebysheff metrics $g_j^{(i)}$, $j = 1, \ldots, n_{pop}$ using Equation (18). Select the individuals using the Tchebysheff metric, a set of non-dominated individuals can be obtained. Note that during the selection, some individuals may have the optimal Tchebysheff metrics w.r.t more than one weight vectors.

Suppose population $\mathbf{X}^{(i)} = \{\mathbf{x}_j^{(i)} \in^n | j = 1, \ldots, n_{pop}\}$ can be described using a set of distribution parameters $\sigma^{(i)} = \{\sigma_j^{(i)}(\mathbf{x}) \in | j = 1, \ldots, d \leq n_{pop}\}$, new distribution parameters $\sigma^{(i)}$ of the potential solutions based on previous $\sigma^{(i-1)}$ and Tchebysheff metric values $\mathbf{g}^{(i-1)} = \{g_j^{(i-1)} \in \mathbb{R} | j = 1, \ldots, n_{pop}\}$ can be predicted as

$$\sigma^{(i)} = \sigma^{(i-1)} + \mathbf{K}(\mathbf{g}, \sigma)\left(\mathbf{g}^{(i)} - \mathbf{g}^{(i-1)}\right) \tag{19}$$

where $\mathbf{K}(\mathbf{g}, \sigma)$ is the feedback matrix, and new individuals $\mathbf{x}_j^{(i)} \in \mathbf{X}^{(i)}$ are generated using $\sigma^{(i)}$.

The key now is to estimate adaptively the distribution parameter σ and feedback matrix $\mathbf{K}(\mathbf{g}, \sigma)$ with the previous $\sigma^{(i-1)}$ and metric values $\mathbf{g}^{(i-1)}$. To tackle the problem, a proportional orthogonal decomposition (POD, also called Principal component analysis (PCA) [23, 25]) and Gaussian estimator based technique are proposed.

Suppose in the ith generation, the individuals are $\mathbf{x}_j^{(i)}$, a confidence ellipsoid that contains all the possible candidate individuals can be constructed. The distribution parameters σ can be computed using proportional orthogonal decomposition (POD,

also called Principal component analysis). Given data set $\mathbf{X} = \{\mathbf{x}_j \in \mathbb{R}^n | j = 1, \ldots, n_{pop}\}$ of the individuals, kernel matrix of $\mathbf{X}$ can be computed as

$$\mathbf{C} = \sum_{j=1}^{n_{pop}} (\mathbf{x}_j - \mathbf{x}_c)(\mathbf{x}_j - \mathbf{x}_c)^T \tag{20}$$

where $\mathbf{x}_c$ is the barycenter of $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_{npop}$. The jth principal component in the direction of the eigenvector associated with the jth largest eigenvalue is

$$\xi_j = \sqrt{\frac{\lambda_j}{\sum_{j=1}^{M} \lambda_j}} \mathbf{v}_j \tag{21}$$

where $\mathbf{v}_j \in^n$ and $\lambda_j \in \mathbb{R}$ are the eigenvectors and eigenvalues of $\mathbf{C}$, respectively. Candidate individuals in the ellipsoid can then be generated at random uniformly as the weighed sum of the principal components

$$\mathbf{x}_{new} = \sum_{j=1}^{d} \left[\sigma_j \left(\text{rand}(1, 1) - 0.5\right) \xi_j + \mathbf{x}_{ref_j}\right] \tag{22}$$

where $\sigma_j \in \mathbb{R}$ is the magnitude of the ellipsoids semi-axis in the direction of $\xi_i \in \mathbb{R}^n$, and $\mathbf{x}_{ref_j}$ is the vector of the population centroid projected on ξ_i

$$\mathbf{x}_{ref_j} = \frac{\mathbf{x}_c \cdot \xi_j}{|\xi_j|^2} \xi_j \tag{23}$$

Changing $\sigma = \{\sigma_j | j = 1, \ldots, d \leq n\}$ and $\mathbf{x}_{ref}$, the size and location of the ellipsoid can be varied.

In Equation (22), the number of principal components d that generates new design variables can be less than the dimension of design space, n, as those principal components that make little contributions to the function evaluations can be neglected and removed [23–25].

Figure 4 schematically shows the proposed techniques using principal analysis. The population of the design variables in the design space and corresponding front in the criteria space are shown in the upper panel of the figure. Principal components of the population are listed in the figure too. Given the data set of distribution parameters σ, new individuals can be generated using Equations (20)–(23). To obtain the optimal parameters σ, correlation between the data set σ and Tchebysheff metrics $\mathbf{g}$ needs to be modeled. The lower panel of the figure shows the correlation between the data set of $\mathbf{g}$ and σ established a priori, using a Gaussian type model.

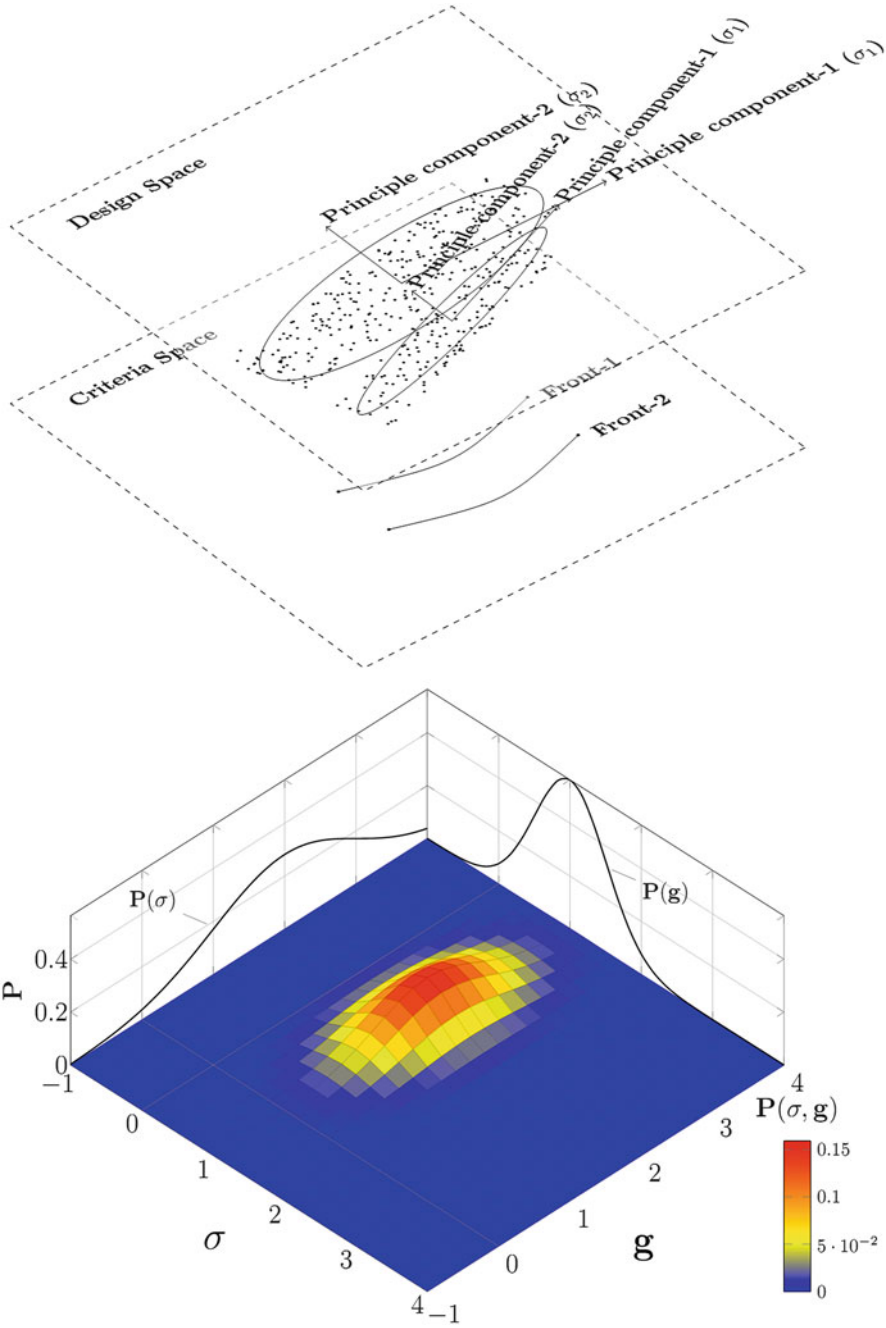


Figure 4 Principal analysis and Tchebysheff metrics of the bi-objective optimization problem. In the upper panel, principal components of the population of design variables and corresponding fronts in the criteria space are presented. Fronts are pushed forwards by varying the control parameters σ of principal components (Equations (20)–(23)). The lower part of the figure correlates the control parameters σ to the data set of the metric values **g** using a Gaussian type model. Control parameters σ to generate new individuals can be determined using the previous data set of σ and the metric values of **g**

Now the problem becomes that of determining the parameters $\sigma^{(i)}$ of the new ellipsoid of the solutions given $\mathbf{g}^{(i)}$, formerly $\sigma^{(i-1)}$ and $\mathbf{g}^{(i-1)}$. Suppose $\sigma^{(i-1)}$ and $\mathbf{g}^{(i-1)}$ are jointly Gaussian, i.e.,

$$\begin{bmatrix} \sigma \\ \mathbf{g} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_\sigma \\ \mu_\mathbf{g} \end{bmatrix}, \begin{bmatrix} \Sigma_\sigma & \Sigma_{\sigma\mathbf{g}} \\ \Sigma_{\sigma\mathbf{g}} & \Sigma_\mathbf{g} \end{bmatrix} \right) \tag{24}$$

where μ_σ and $\mu_\mathbf{g}$ are the mean value of parameters σ and $\mathbf{g}$, $\Sigma_\mathbf{g}$, $\Sigma_{\sigma\mathbf{g}}$ are the co-variance of σ and $\mathbf{g}$, respectively, then the conditional distribution of σ given $\mathbf{g} = [g_i]$ is normal and has [4]

$$\mu_{\sigma|\mathbf{g}} = \mu_\sigma + \Sigma_{\sigma\mathbf{g}}\Sigma_\mathbf{g}^{-1}(\mathbf{g} - \mu_\mathbf{g}) \tag{25}$$

with the variance

$$\Sigma_{\sigma|\mathbf{g}} = \Sigma_\sigma - \Sigma_{\sigma\mathbf{g}}\Sigma_\mathbf{g}^{-1}\Sigma_{\sigma\mathbf{g}}^T \tag{26}$$

The estimator makes sense even when $\mathbf{g}$ and σ are not jointly Gaussian, and $\Sigma_\sigma - \Sigma_{\sigma\mathbf{g}}\Sigma_\mathbf{g}^{-1}\Sigma_{\sigma\mathbf{g}}^T \leq \Sigma_\sigma$, i.e., the covariance of the estimation error is always less than the previous covariance of [15].

The new data set of $\sigma = \{\sigma_j | j = 1, \ldots, d\}$ can then be generated as

$$\sigma = \mathcal{N}(\mu_{\sigma|\mathbf{g}}, \Sigma_{\sigma|\mathbf{g}}) \tag{27}$$

Putting the data set σ into Equation (22), new individuals can be generated. Implementing the steps Equations (20)–(27) till the termination conditions (e.g., the number of iterations, MSE of $\mathbf{g}$) are met, the optimal solutions can be obtained.

Figure 5 shows the simulation results of the new algorithm, POD-MOO, and the conventional evolutionary algorithm, NSGA-II. The test function is bi-objective ZDT1 with 30 design variables. Population sizes of POD-MOO and NSGA-II are both set to 100, and in the POD-MOO, only 5 principal components are used to

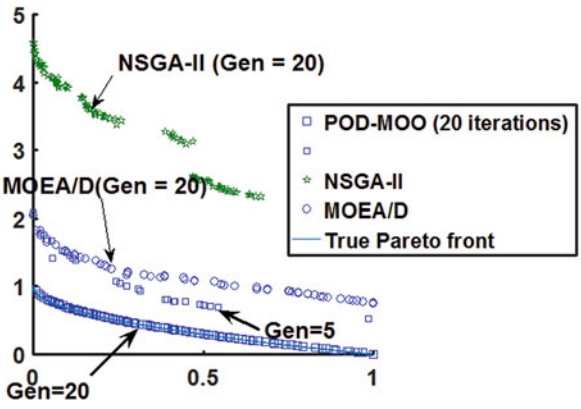


Figure 5 POD-MOO vs. NSGA-II and MOEA/D. Test function: ZDT1. Iterations: 20. Population size: 200. Number of design variables:30. Function evaluations of both algorithms: around 4000

generate new individuals. Crossover probability and mutation probability of NSGA-II are set to $p_c = 0.9$ and $p_m = 1/n$, respectively, where n is the number of decision variables. The distribution indices of SBX and polynomial mutation in NSGA-II are both set equal to 20. It appears that after 20 iterations, the solutions obtained using POD-MOO are quite close to the true Pareto set. As for the popular MOO algorithm, NSGA-II, it may take up to hundreds of iterations before the optimal Pareto solutions are obtained.

Figure 6 shows the experimental results of the standard bi-objective ZDT series benchmarks: ZDT2, ZDT3, ZDT6, and ZDT4.[1] Population size of the MOO is set to

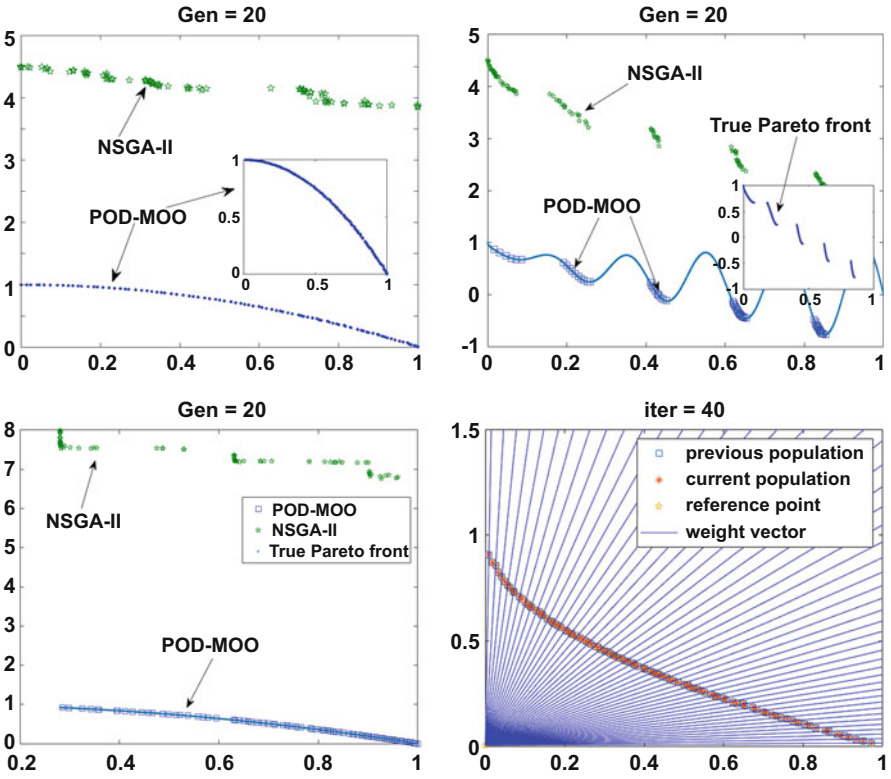


Figure 6 Tests of POD-MOO on standard benchmarks of ZDT2(top-left), ZDT3 (top-right), ZDT6 (bottom-left), and ZDT4 (bottom-right). Population size: 200. Reference vectors of the decomposed bi-objective optimization ($\mathbf{w}$ in Equation (17)) are listed in the test of ZDT4. Although the benchmarks have different Pareto fronts, in each test, the MOO takes dozens of iterations to converge to the true Pareto front

[1]Program code of POD-MOO and experimental simulations of the bi-objective and three objective benchmarks are available at https://sites.google.com/site/adloptimization/moo-with-principle-component-analysis.

200. Reference vectors of the decomposed bi-objective optimization ($\mathbf{w}$ in Equation (17)) are listed in the test of ZDT4. Although the benchmarks have different Pareto fronts, in each test, the MOO takes dozens of iterations to converge to the true Pareto front. In contrast, if the conventional genetic algorithms, e.g., NSGA-II, are used, it may take hundreds of iterations for the algorithms to search the true Pareto fronts.

3.4 Surrogate Assisted Optimization

To reduce the computational cost of the numerical model, the Kriging based surrogate can be used. The Kriging surrogate acts like an interpolator. First, a set of sample data and corresponding responses of the numerical model are put into the surrogate to train the surrogate. New data at the untested point can then be predicted using the Kriging surrogate.

The surrogate is constructed as follows: Consider a set of sample individuals $\mathbf{X}$ with the responses of the numerical model, $\mathbf{y}$, the Kriging prediction at point $\mathbf{x}$ can be constructed as a mean function plus a variance [5, 9]

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \psi(\mathbf{x}, \mathbf{X})\Psi^{-1}(\mathbf{X}, \mathbf{X})(\mathbf{y} - \mathbf{1}\hat{\mu}) \tag{28}$$

with the predictive mean and variance

$$\hat{\mu} = \frac{\mathbf{1}^T \Psi^{-1}(\mathbf{X}, \mathbf{X})\mathbf{y}}{\mathbf{1}^T \Psi^{-1}(\mathbf{X}, \mathbf{X})\mathbf{1}} \tag{29}$$

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\mu)^T \Psi^{-1}(\mathbf{y} - \mathbf{1}\mu)}{n} \tag{30}$$

where $\mathbf{1}$ is the $n \times 1$ column vector of ones. $\psi(\mathbf{x}, \mathbf{X})$ and $\Psi(\mathbf{X}, \mathbf{X}')$ are the correlation vector and correlation matrix with

$$\psi(\mathbf{x}, \mathbf{x}') = \exp\left(-\sum_{i=1}^{p_i} \theta_i |\mathbf{x}_i - \mathbf{x}_i^{'}|^{p_i}\right) \tag{31}$$

where θ and p are the hyper parameters that can be determined using the maximum likelihood method [5, 9].

The mean squared error of the prediction at $\mathbf{x}$ is

$$\hat{s}^2(\mathbf{x}) = \sigma^2 \left[1 - \psi^T \Psi^{-1} \psi + \frac{1 - \mathbf{1}^T \Psi^{-1} \psi}{\mathbf{1}^{\mathbf{T}} \Psi^{-1} \mathbf{1}}\right] \tag{32}$$

To update the surrogate during the optimization, some more sample data can be used to train the surrogate further. In this work, an expected improvement (EI) based infill strategy is used to update the surrogate.

Suppose $\mathcal{N}\left(\hat{y}(\mathbf{x}), \hat{s}^2(\mathbf{x})\right)$ is the Kriging prediction for the objective function $y(\mathbf{x})$, and the minimum of $y(\mathbf{x})$ over all evaluated points in population $\mathbf{X}$ is y_{min}, then the expected improvement of an untested point $\mathbf{x}$ is [5]

$$EI(\mathbf{x}) = E\left[\max\{y_{min} - y(\mathbf{x}), 0\}\right] \tag{33}$$

and can be computed as

$$E\left[I(\mathbf{x})\right] = \left[y_{min} - \widehat{y}(\mathbf{x})\right] \Phi\left(\frac{y_{min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) + \hat{s}(\mathbf{x})\phi\left(\frac{y_{min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) \tag{34}$$

New infill points can then be determined by exploring the design space using the infill criterion, and those points that have maximum EI values can be sampled as the infill points to improve the surrogate.

3.5 *Surrogate Assisted Evidence-Based Robust Design Optimization*

The design optimization consists of two loops. In the inner loop, the Kriging based surrogate of the expensive numerical lifetime predictor is constructed, while in the outer loop the new MOO algorithm is implemented. The surrogate reads the debris area-to-mass ratio (AMR), semi-axis, eccentricity, argument of perigee, and mean anomaly and computes the debris lifetime till the decay altitude reaches 65 km. Data of the numerical lifetime predictor are Latin-hypercube sampled and put into the surrogate to train the surrogate.

The evidence computation samples the uncertain space and decomposes the uncertain space into a series of subspaces. Uncertain parameters and the design variables are put into the surrogate to compute the lifetime. With the predicted lifetimes, evidence of the lifetimes can be computed using the sampling approach as Equations (14) and (15) show. The MOO of Equations (17)–(27) is then used to optimize the objective functions of the belief and lifetimes. Figure 7 shows the flowchart of the proposed evidence-based robust design optimization.

4 Numerical Simulation

Consider the robust design optimization problem of the pulsed LODR.[2] The target debris is set to Iridium 33, and the optimization objective is set to maximize under uncertainty the difference in the sum of the lifetimes of 10 debris objects. The orbital

[2]Part of program code of the robust POD-MOO and numerical simulation of LODR are available at https://sites.google.com/site/adloptimization.

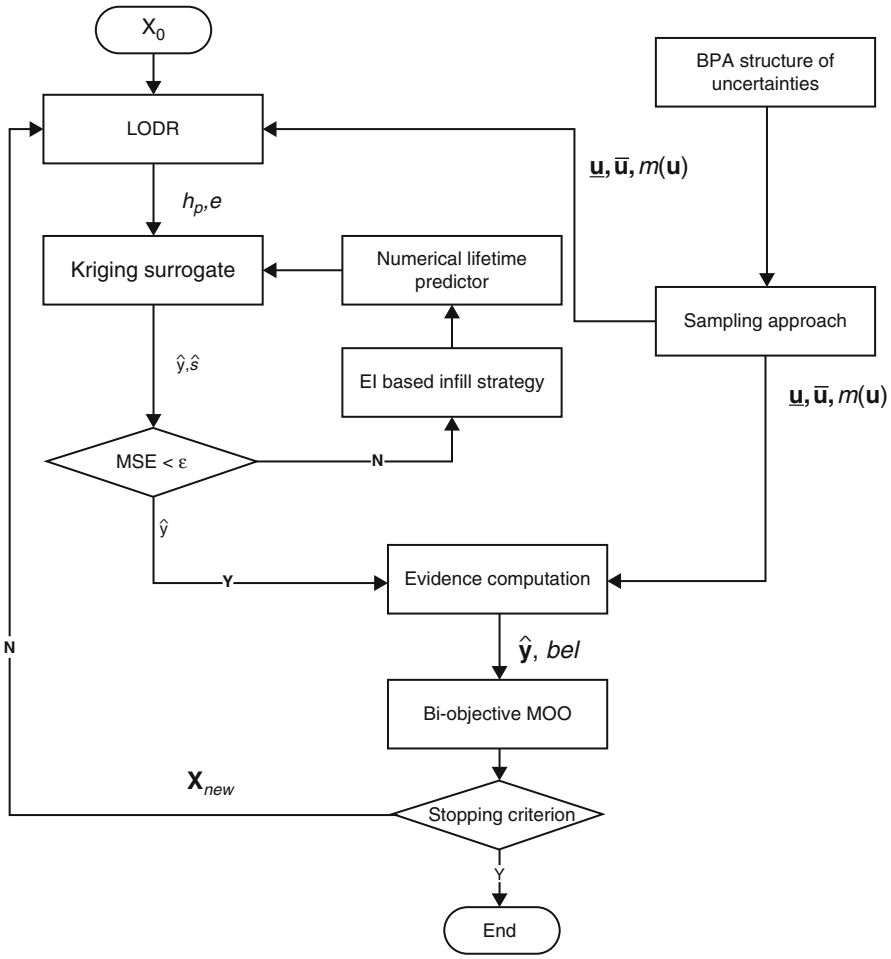


Figure 7 Flowchart of the evidence-based robust design optimization. The design optimization consists of two loops. In the inner loop, the surrogate is constructed and updated with the new infill points , while in the outer loop, the bi-objective optimization is implemented to optimize debris lifetimes and their belief values

elements of the debris are from the CelesTrak database (TLEs of the debris are available at http://celestrak.com/NORAD/elements/iridium-33-debris.txt). Table 3 shows the parameters of the pulsed laser.

In the numerical propagator of the debris lifetime prediction, solar radiation pressure, atmospheric drag, non-spherical Earth, and gravity from the third body are taken into account. The Jacchia 1970 atmospheric density model is used, and the decay altitude is set to 65 km. In the numerical lifetime predictor, a Runge–Kutta–Fehlberg 7th order integrator with 8th order error control is used. The drag coefficient is set to 2.2, and the coefficient of reflectivity is set to 0.28 [14]. Note that

Table 3 Parameters of the pulsed LODR

Parameter	Value	Unit
Number of impulse	4750	–
Laser pulse frequency	100	Hz
Laser pulse interval	100	ns between pulses
Laser power	1000	J/pulse

Table 4 BPA structure of uncertainties

Parameter	Description	μ	σ_l	σ_u	BPA
C_m(NS/J)	Coupling coefficient	$2.0 \cdot 10^{-5}$	$0.8 \cdot 10^{-7}$	$1.5 \cdot 10^{-7}$	0.60
		$2.8 \cdot 10^{-5}$	$0.5 \cdot 10^{-7}$	$1.0 \cdot 10^{-7}$	0.30
		$2.0 \cdot 10^{-5}$	$0.2 \cdot 10^{-7}$	$1.2 \cdot 10^{-7}$	0.10
u(kg/J)	Ablation rate	$80 \cdot 10^{-9}$	$10 \cdot 10^{-10}$	$20 \cdot 10^{-10}$	0.40
		$78 \cdot 10^{-9}$	$15 \cdot 10^{-10}$	$22 \cdot 10^{-10}$	0.30
		$72 \cdot 10^{-9}$	$12 \cdot 10^{-10}$	$17 \cdot 10^{-10}$	0.30
m_0(kg)	Average mass [12]	1.30	0.05	0.10	0.30
		1.50	0.05	0.12	0.20
		1.42	0.05	0.08	0.50
r_0(cm)	Debris average radius	12.33	1.05	1.50	0.50
		11.40	0.85	1.02	0.20
		13.52	0.75	0.98	0.30
C_r	Coefficient of reflectivity	0.20	0.05	0.10	0.20
		0.23	0.07	0.12	0.30
		0.32	0.05	0.08	0.50
C_d	Drag coefficient	2.20	0.05	0.10	0.30
		2.23	0.05	0.12	0.20
		2.32	0.05	0.08	0.50

The uncertainties are of the LODR and debris parameters, and supposed to be estimated by three experts

both values are closely related to the shape of the debris, and can vary with different debris. Regarding the LODR characteristic parameters, the coupling coefficient is $C_m = 2.0 \cdot 10^{-5}$ Ns/J for the ideal case of aluminum, and the ablation rate is equal to $\mu = 80 \cdot 10^{-9}$ kg/J [20]. Both the values could vary with the laser parameters such as intensity, wavelength, pulse length irradiated to the debris material, and surface conditions [20]. The average mass of a single Iridium 33 debris object is set equal to $m_0 = 1.327$ kg [12]. A BPA structure for the parameters is presented (Table 3) to show the uncertainty effects. Three groups of estimated values and standard deviations are listed in the table, assuming that the uncertainties follow a Gaussian distribution (Table 4).

Most of the orbital eccentricities of the debris are less than 0.01, and the laser impulses are delivered within a short period of time (47 s in the simulation); therefore, the accumulated velocity increment with a constant delivered angle can be used to compute the orbital elements after LODR. The station is set to Svalbard with

latitude equal to 78.21670°. To simplify the computation, the velocity is computed in advance using the parameters provided in Table 3, and an equivalent constant ϑ, 120°, is used (Figure 1).

All debris is associated to an ID No. 1-466. The design vectors are then the ID numbers of the debris and the objective is set to maximize under uncertainty the difference in the sum of the lifetimes of 10 debris objects. Writing in mathematical form, the problem can be formulated as

$$\min_{\mathbf{x}\in\mathbf{D},\mathbf{u}\in\mathbf{U}} f(\mathbf{x},\mathbf{u}) = \sum_{i=1}^{10} L(\mathbf{x}_i,\mathbf{u}) \tag{35}$$

where $\mathbf{x} \in \mathbb{N}$ are the indices of debris from 1 to 466, $\mathbf{u} \in \mathbf{U}$ is the uncertain parameters, and $L(\mathbf{x}_i,\mathbf{u})$ is the predicted lifetime of $\mathbf{x}_i$ debris.

The population size of robust POD-MOO is set to 200. Focal elements and corresponding masses are combined and approximated in advance to reduce the computational cost. The number of focal elements is set to be equal to the population size. A data set of numerical lifetime prediction with perigee altitude $h_P \in [250\,\text{km}, 600\,\text{km}]$ and eccentricity $e \in [0, 0.3]$ is Latin-hypercube sampled and put into Kriging to initialize the surrogate. The sample size is set to 240. In each iteration, 5 solutions with maximum EI values are selected as the infill points, and put into the Kriging to update the surrogate.

Figure 8 shows the objective values of the optimal solutions after 50 iterations. Among the solutions, two solutions A and B with the belief of 0.865 and 0.625, respectively, are selected. Table 5 shows the debris IDs of solution A and B and their evidence levels. The belief of solution A is 0. 865 for a sum of the post-LODR lifetimes of 32.41 years, while the belief of solution B is 0. 625/24.07 years.

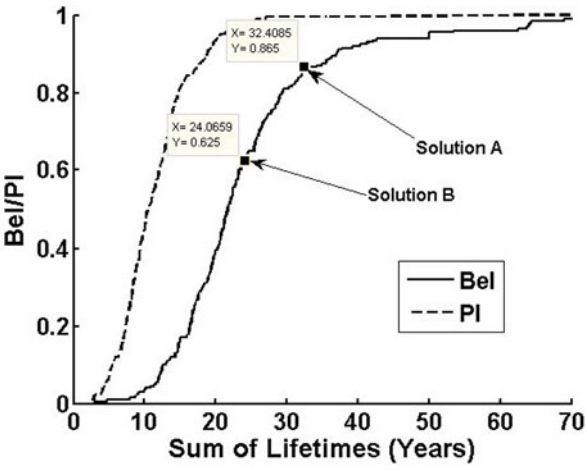


Figure 8 Solution A and B: target debris, lifetimes, and belief values

Table 5 Robust optimal solution A and B: target debris, lifetimes before/after LODR, and evidence level of the optimal solutions

Solution	Debris	Sum (before/after, years)	Belief/plausibility
A	283, 115, 9, 31, 3, 75, 425, 192, 112, 52	796.95/32.41	0.865/0.995
B	45, 133, 109, 308, 15, 275, 297, 144, 82, 40	631.18/24.07	0.625/0.975

Table 6 Solution C: target debris and lifetimes before and after LODR (assuming all uncertainties are set to zero and using a deterministic optimizer)

Debris	Before (years)	After (years)	Sum (before/after, years)
142	8.32	0.07	117.50/1.558
194	10.97	0.16	
168	29.01	0.44	
207	7.60	0.11	
133	9.82	0.18	
335	8.17	0.11	
142	8.32	0.07	
281	8.96	0.03	
196	22.63	0.32	
159	3.67	0.04	

To verify the robust solutions, a genetic based deterministic optimizer is used to optimize directly the objective function with $C_m = 2.0 \cdot 10^{-5}$, $\mu = 80 \cdot 10^{-9}$ kg/J, and $m_0 = 1.327$ kg, respectively. As for coefficient of reflectivity, drag coefficient, and average radius of the debris, the values of $C_r = 0.28$, $C_d = 2.2$, and $r_0 = 12.33$ cm are used [12, 14].

Table 6 shows the deterministic optimal solution (Solution C) using the time consuming numerical predictor. The data is in good agreement with the proposed method and is highly consistent with the robust calculations when the uncertainty impacts are unconsidered. In Figure 8, the minimum lifetimes of the curves are about 1.5 years with nearly zero belief (the belief is 0.04) in which the non-deterministic effects can be ignored, while the result of the deterministic approach is 1.558 years. The two sets of calculations are quite close. As can be seen, in addition to providing a deterministic calculation result, the proposed method can provide the confidence levels of the calculation results under non-deterministic conditions. All of these provide convenience for decision makers to optimize their designs.

5 Conclusion

In this work, an early stage robust analysis tool for the LODR design is presented. In the design optimization, both the epistemic uncertainty and aleatory uncertainty are considered. The evidence-based analysis tool is used to quantify the evidence level of the optimal solutions. Compared to the optimization method using conventional

optimization algorithms, the method proposed in this work provides not only the solutions that could be implemented in the LODR, but also the maximum belief values that the solutions can be achieved.

To resolve the problems due to the evidence computation and the expensive numerical model, a high efficiency MOO, using the Tchebysheff decomposition strategy and Gaussian predictor, is proposed. The MOO outperforms the conventional algorithms in a series of standard test functions. A surrogate is used to reduce the cost due to the expensive model. Evidence computations of the uncertainty impacts are integrated into the optimization via the sampling approach. At each iteration, the number of objective function evaluations is twice the population size. Data from the surrogate are used instead of the numerical expensive data, and an EI based infill strategy is used to update the surrogate.

In the simulation, the debris fragments are supposed to be spherical and face-on planes. As the lifetimes of the debris can vary greatly with the relevant parameters, a more detailed analysis of the objects properties, such as the realistic shapes and tumbling effects, should be taken into account if an actual LODR is implemented. The robust optimization method presented in this chapter provides an early stage robust analysis tool for the LODR in order to determine the potential candidates. Impacts due to the attitude and shape of the debris will be taken into account in future works.

References

1. Agarwal, H., Renaud, J.E., Preston, E.L., Padmanabhan, D.: Uncertainty quantification using evidence theory in multidisciplinary design optimization. Reliab. Eng. Syst. Saf. **85**(1), 281–294 (2004)
2. Asafuddoula, M., Ray, T., Sarker, R.: A decomposition-based evolutionary algorithm for many objective optimization. IEEE Trans. Evol. Comput. **19**(3), 445–460 (2015)
3. Auer, E., Luther, W., Rebner, G., Limbourg, P.: A verified Matlab toolbox for the Dempster-Shafer theory. In: Workshop on the Theory of Belief Functions (2010)
4. Balakrishnan, N.: Continuous Multivariate Distributions. Wiley, London (2006)
5. Couckuyt, I., Dhaene, T., Demeester, P.: ooDACE toolbox: a flexible object-oriented Kriging implementation. J. Mach. Learn. Res. **15**, 3183–3186 (2014)
6. Croisard, N., Vasile, M., Kemble, S., Radice, G.: Preliminary space mission design under uncertainty. Acta Astronaut. **66**(5), 654–664 (2010)
7. Deb, K.: Multi-objective optimization. In: Search Methodologies, pp. 403–449. Springer, Berlin (2014)
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
9. Forrester, A., Keane, A., et al.: Engineering Design via Surrogate Modelling: A Practical Guide. Wiley, London (2008)
10. Hou, L., Pirzada, A., Cai, Y., Ma, H.: Robust design optimization using integrated evidence computationł with application to orbital debris removal. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 3263–3270. IEEE, Piscataway (2015)
11. Hou, L., Tan, W., Ma, H.: Multi-fidelity design optimization under epistemic uncertainty. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 4452–4459. IEEE, Piscataway (2016)

12. Kelso, T., et al.: Analysis of the iridium 33-cosmos 2251 collision. Adv. Astronaut. Sci. **135**(2), 1099–1112 (2009)
13. Liedahl, D., Rubenchik, A., Libby, S.B., Nikolaev, S., Phipps, C.R.: Pulsed laser interactions with space debris: target shape effects. Adv. Space Res. **52**(5), 895–915 (2013)
14. Mason, J., Stupl, J., Marshall, W., Levit, C.: Orbital debris–debris collision avoidance. Adv. Space Res. **48**(10), 1643–1655 (2011)
15. Moon, T.K., Stirling, W.C.: Mathematical Methods and Algorithms for Signal Processing, vol. 1. Prentice Hall, New York (2000)
16. Phipps, C., Albrecht, G., Friedman, H., Gavel, D., George, E., Murray, J., Ho, C., Priedhorsky, W., Michaelis, M., Reilly, J.: Orion: Clearing near-earth space debris using a 20-kw, 530-nm, earth-based, repetitively pulsed laser. Laser Part. Beams **14**(1), 1–44 (1996)
17. Phipps, C.R., Baker, K.L., Libby, S.B., Liedahl, D.A., Olivier, S.S., Pleasance, L.D., Rubenchik, A., Trebes, J.E., George, E.V., Marcovici, B., et al.: Removing orbital debris with lasers. Adv. Space Res. **49**(9), 1283–1300 (2012)
18. Rebner, G., Auer, E., Luther, W.: A verified realization of a Dempster–Shafer based fault tree analysis. Computing **94**(2), 313–324 (2012)
19. Roy, A.E.: Orbital Motion. CRC Press, West Palm Beach (2004)
20. Schall, W.O.: Laser radiation for cleaning space debris from lower earth orbits. J. Spacecr. Rocket. **39**(1), 81–91 (2002)
21. Vasile, M.: Robust mission design through evidence theory and multiagent collaborative search. Ann. N. Y. Acad. Sci. **1065**(1), 152–173 (2005)
22. Vasile, M.: A behavioral-based meta-heuristic for robust global trajectory optimization. In: IEEE Congress on Evolutionary Computation, CEC 2007, pp. 2056–2063. IEEE, Piscataway (2007)
23. Volkwein, S.: Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling. Lecture Notes. vol. 4(4). University of Konstanz, Konstanz (2013)
24. Xiao, D., Fang, F., Buchan, A.G., Pain, C.C., Navon, I.M., Du, J., Hu, G.: Non-linear model reduction for the Navier–Stokes equations using residual DEIM method. J. Comput. Phys. **263**, 1–18 (2014)
25. Xie, D., Xu, M., Dowell, E.H.: Proper orthogonal decomposition reduced-order model for nonlinear aeroelastic oscillations. AIAA J. **52**(2), 229–241 (2014)
26. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
27. Zuiani, F., Vasile, M., Gibbings, A.: Evidence-based robust design of deflection actions for near earth objects. Celest. Mech. Dyn. Astron. **114**, 107–136 (2012)

Machine Learning and Evolutionary Techniques in Interplanetary Trajectory Design

Dario Izzo, Christopher Iliffe Sprague, and Dharmesh Vijay Tailor

Abstract After providing a brief historical overview on the synergies between artificial intelligence research, in the areas of evolutionary computations and machine learning, and the optimal design of interplanetary trajectories, we propose and study the use of deep artificial neural networks to represent, on-board, the optimal guidance profile of an interplanetary mission. The results, limited to the chosen test case of an Earth–Mars orbital transfer, extend the findings made previously for landing scenarios and quadcopter dynamics, opening a new research area in interplanetary trajectory planning.

1 Introduction

The use of artificial intelligence (AI) techniques for the design of interplanetary trajectories, in particular in their early design phases, has been proposed and studied by numerous scientists over the past few decades. Among the AI techniques deployed to help the optimisation of spacecraft trajectories are evolutionary algorithms [15, 22, 23, 28, 31, 35, 48, 52], machine learning techniques [2, 4, 27, 33], evolutionary neuro-controllers [6, 7], tree search methods [17, 18, 23, 27, 50] to only name a few widely studied methods. While the synergies between research in AI and in trajectory design were becoming increasingly apparent, the field of AI as a whole experienced a renaissance in the second decade of the millennium, delivering exciting developments that are of significance to many scientific fields. Certainly the area of machine learning is one that is powering such an AI renaissance, in particular with deep learning techniques [47] becoming fundamental to define

D. Izzo (✉) · D. V. Tailor
European Space Agency, Noordwijk, The Netherlands
e-mail: dario.izzo@esa.int; dharmesh.tailor@live.co.uk

C. I. Sprague
KTH Royal Institute of Technology, Stockholm, Sweden
e-mail: sprague@kth.se

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_8

new benchmarks in the most diverse applications. It thus seems likely that also the optimisation of interplanetary trajectories, already receptive of AI methods in general, is going to benefit from the vast amount of new knowledge being produced in the context of AI research.

In this chapter we first briefly provide an overview on the state-of-the art of the use of evolutionary techniques (Sect. 2) and machine learning techniques (Sect. 3) when it comes to optimising interplanetary trajectories, and we then introduce a novel idea concerning the application of deep learning for the on-board representation of the optimal guidance profile for an interplanetary probe, extending previous work made on planetary landing scenarios [42–44].

2 Evolutionary Algorithms for Trajectory Planning: An Overview

Evolutionary algorithms are a class of global optimisation techniques that make use of heuristic rules, often inspired but not limited to natural paradigms such as Darwinian evolution, to search for optimal solutions in discontinuous, rugged landscapes. As such they are a good match to solve interplanetary trajectory optimisation problems where the planetary geometry defines a quite complex solution landscape already in simple cases. Already in 1985 a genetic algorithm (GA) [28] was proposed and studied in the context of interplanetary trajectory optimisation concluding that

> ...a considerable effort is still needed for developing efficient schemes using genetic algorithms. However, they appear to offer an entirely original way for solving a large class of [interplanetary trajectory] global optimisation problems ...

Given the CPU power available at the time, it is only natural that this pioneering work complained about efficiency. In the following decade the "entirely original way" was consolidated and more successful applications of GAs for both low-thrust propelled spacecraft [40] and impulsive thrust strategies [15] were deployed. Studies on the use of some form of GA, also leveraging on the ever increasing computational power available, continued (see for example [3, 9, 12, 14, 41]) showing how the original intuition, back in 1985, was one rich of consequences. It was indeed relatively consequential to substitute the GA with any other form of stochastic optimiser, so that in 2004 the European Space Agency performed a series of studies, in collaboration with academia, to test and compare several global optimisers on interplanetary trajectory problems [10, 22, 34]. It then became clear how other evolutionary algorithms were offering alternative, and in many ways superior, choices to help the preliminary phases of trajectory design. Among them, differential evolution (DE) and particle swarm optimisation (PSO) were identified and benchmarked on several chemical propulsion problems opening the way to further independent confirmations of their performances [32, 35, 37, 48, 52, 55].

Interestingly, in 2013, a self-adaptive differential evolution algorithm was used to plan the grand tour of the Jupiter moons that was awarded the golden Humies medal [23] for "human-competitive results produced by genetic and evolutionary computation". Many other evolutionary approaches have been proved to be of use by various researchers during the early 2000s, including simulated annealing (SA) [32] and ant colony optimisation (ACO) [5, 39]. More recently, covariance matrix evolutionary strategy (CMA-ES) was shown to be potentially outperforming self-adaptive DE on a class of transfers [24].

While the no free lunch theorem [54] guarantees that no evolutionary approach is better than any other on average, restricting the class of optimisation problems to those representing interplanetary transfers allows to identify the most useful approaches. To this end, a set of problems called GTOP database [53] (a sort of open trajectory gym) was created and made available to the scientific community and is still the subject of active research [1, 4, 20, 45, 49, 51]. Some of the interplanetary trajectory problems (i.e. Messenger and Cassini2) in the GTOP database were also used during the CEC2011 competition attracting the attention of the larger scientific community of evolutionary computations (see [12] for the competition winner).

Most of the research mentioned so far considers continuous optimisation problems with a single objective and box constraints. In its most general case, though, the problem encountered in interplanetary trajectory design is multi-objective, with nonlinear constraints and, possibly, integer decision variables. Genetic approaches to multi-objective trajectory design were benchmarked already in 2005 [31] followed by deeper studies on non-dominated sorting genetic algorithm (NSGA-II) that included also, as integer variables, the planetary fly-by sequence [9]. Multi-objective versions of PSO have also been considered early on [30]. The more modern multi-objective evolutionary algorithm by decomposition (MOEA/D) was later identified as a most performing technique in applicable cases [24]; in the same work, several constraint handling techniques including co-evolution and immune systems were tested.

Whenever the representation of an interplanetary trajectory requires integer and continuous variables, the resulting optimisation problem (a MINLP) typically becomes intractable also for evolutionary approaches and while some results have been obtained, for example using the ACO paradigm [5, 46], a more convincing approach is to consider the continuous part and the integer part of the problem separately and architect some optimisation scheme tackling the two problems with different nested techniques (bi-level optimisation). Evolutionary approaches based on GAs [13, 25, 26] or ACO [50] have been used in bi-level optimisation schemes, often coupled with smart tree search strategies such as beam search (BS) [17, 50], Monte Carlo tree search (MCTS) [18], multi-objective beam search (MOBS) [27], or lazy race tree search (LRTS) [23] to take care of the integer part.

Evolutionary approaches to interplanetary trajectory planning have proven their worth beyond any criticism and while they are still not as widely used by the aerospace industry as they could, it is likely that we will see a larger penetration in the industrial sector in the upcoming years, as the interest in artificial intelligence

methods powered by deep learning and machine learning will put also evolutionary techniques in the spotlight.

3 Machine Learning and Interplanetary Trajectories

The use of machine learning (ML) algorithms to aid the design of interplanetary trajectory is not as widely researched as that of evolutionary techniques and is limited to fewer works. The reasons are to be found in the less obvious applicability of these methods to the problems encountered in the design of interplanetary trajectories and in the lack of data sets produced and made available by the aerospace community. In this section we try to summarise the ideas that have so far been proposed.

During the optimisation of an interplanetary transfer, as in any optimisation task, a large number of solutions are computed and assessed to inform the search for better candidates. Typically all these design points are discarded and lost after they have been used to produce new promising search directions. The idea of applying supervised learning on such a data set in order to build a model improving the further selection of initial guesses to guide successive evolutionary runs was proposed and tested [4] on some of the trajectory problems in the GTOP database [53] using support vector machines (SVM). Following a similar reasoning, a ML model can be trained, using the points sampled during the evolution, in order to construct a surrogate model of the trajectory worth, which then avoids expensive evaluations of the objective function [2]. Building surrogate models is particularly relevant when the interplanetary mission fitness requires a high degree of computational resources such as in the case of optimal low-thrust transfers. In that case a surrogate model approximating the final optimal transfer mass enables to quickly search for good launch and arrival epochs, as well as planetary body sequences, e.g. in the case of multiple asteroid or debris rendezvous missions [19, 33]. Unsupervised learning techniques such as clustering or nearest neighbours have also been used to select the target of transfers in multiple asteroid rendezvous missions, upon proper definition of a metric coping with the orbital nonlinearities [27], or to define new box bounds and hence focus successive evolutionary runs in promising areas of the search space (cluster pruning [21]).

A second field where ML algorithms have been used in the context of research in interplanetary trajectory design methods is that of the on-board representation of the optimal guidance profile. Already in 2004 [6] the idea was put forward of using machine learning to learn a representation of the optimal spacecraft guidance profile. In later years the technique was studied exclusively in the context of evolutionary neuro-controllers and optimal control. More recently, neuro-evolution has been substituted with supervised learning (back-propagation), the artificial neural network structure has become deeper, and focus has been shifted to the on-board real time computation of guidance profiles [42–44]. These last works

have all been limited to landing problems and the applicability of these ideas to interplanetary trajectories has not been studied. In the next section we will try to contribute closing this gap showing how the optimal guidance profile of a phase-less Earth–Mars transfer can also be satisfactorily represented by a deep artificial neural network.

4 On-Board Optimal Guidance via a Deep Network

In planetary landing problems, the possibility to generate optimal guidance profiles on-board has been studied and suggested to significantly enhance a vehicles landing accuracy [8, 11]. The resulting algorithms need to run on a radiation-hardened flight processor, that is on a significantly slower processor than those available on modern desktops and one having significant architectural differences. As a rule of thumb, a space qualified processor is an order of magnitude slower than the average desktop processor. The resulting CPU load on the spacecraft has been estimated in a practical scenario [11] to be 0.7s for the computation of one optimal action. To the same end, an alternative is offered by deep feed-forward neural networks trained on the ground to approximate the optimal control and used on-board in real time. The on-board computational effort associated to this architecture is that of one forward pass of the network which, though deep, is not necessarily large [42, 43]. Extending on previous studies on landing and quadcopter dynamics, the feasibility of such a scheme for interplanetary low-thrust transfers is studied here. The same overall scheme is used [43]:

- Step 1: We solve thousands of optimal control problems using Pontryagin's maximum principle. We store, with some time sampling, the obtained solutions in a data set.
- Step 2: We train deep feed-forward neural networks on the data set to learn the optimal control structure.
- Step 3: We use, on-board, the trained network to compute the optimal feedback.

With respect to the previously studied case of planetary landings the methodology is, essentially, unchanged and it is of great interest to see how it can be applied on a radically different problem having larger dimensionality, different nonlinearities and, arguably, a more complex structure.

A Short Note on Notation

We use boldface symbols (e.g. $\boldsymbol{\lambda}$, $\mathbf{r}$, etc.) to denote vector quantities, while scalars or the vector norm will be indicated by normal symbols (e.g. λ, r, etc.). Unit vectors

will be indicated with a small hat (e.g. $\hat{\mathbf{i}}$) and time derivatives with a dot (e.g. $\dot{x}$). We will sometime use the asterisk as a superscript to indicate optimal quantities (e.g. u^*).

4.1 Spacecraft Dynamics

The motion of a spacecraft equipped with a constant specific impulse nuclear electric low-thrust propulsion system can be described in some heliocentric, inertial, frame using Cartesian coordinates by the equations:

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= -\frac{\mu}{r^3}\mathbf{r} + c_1 \frac{u(t)}{m}\hat{\mathbf{i}}_{\mathbf{u}}(t) \\ \dot{m} &= -c_2 u(t) \end{aligned} \tag{1}$$

where $\mathbf{r}$, $\mathbf{v}$ and m denote the spacecraft position, velocity and mass (also denoted as spacecraft state $\mathbf{x}$), μ is the gravitational parameter of the Sun, c_1 is the maximum thrust achievable by the on-board propulsion system and $c_2 = c_1/I_{sp}g_0$. We have denoted with I_{sp} the propulsion specific impulse and with g_0 the Earth gravity constant at sea level. The control variables $u(t)$ and $\hat{\mathbf{i}}_{\mathbf{u}}(t)$ (or $\mathbf{u}(t)$ for brevity) describe the thrust level (here also called throttle) and its direction and are constrained as follows $|u(t)| \leq 1$ and $|\hat{\mathbf{i}}_{\mathbf{u}}(t)| = 1, \forall t \in [t_1, t_2]$.

We are interested in the problem of finding the controls $u(t)$ and $\hat{\mathbf{i}}_{\mathbf{u}}(t)$ to minimise the functional:

$$J(t_1, t_2, u(t)) = \alpha \int_{t_1}^{t_2} u(t)dt + (1-\alpha)\int_{t_1}^{t_2} u^2(t)dt \tag{2}$$

and able to steer the spacecraft from some initial point $x_1 \in \mathscr{S}_1$ to some final point $x_2 \in \mathscr{S}_2$ where the sets $\mathscr{S}_1$ and $\mathscr{S}_2$ are some closed subsets (hypersurfaces) of the state space. Note that the functional J is parameterised by the continuation parameter $\alpha \in [0, 1]$ which weights two contributions corresponding to mass optimality ($\alpha = 1$) and quadratic control optimality ($\alpha = 0$).

We now derive the, known, necessary optimality conditions for the above stated problem applying the maximum principle from Pontryagin (note that we have stated a minimisation problem, hence the conditions are actually slightly different from the ones originally derived in Pontryagin work [38]).

4.2 Minimisation of the Hamiltonian $\mathscr{H}$

We start by introducing seven auxiliary functions defined in $t \in [t_1, t_2]$, the co-states, which we will denote with $\boldsymbol{\lambda}_r$,$\boldsymbol{\lambda}_v$ and λ_m, or, for brevity, $\boldsymbol{\lambda}$. We then introduce the Hamiltonian function:

$$\mathscr{H}(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}) = \boldsymbol{\lambda}_r \cdot \mathbf{v} + \boldsymbol{\lambda}_v \cdot \left(-\frac{\mu}{r^3}\mathbf{r} + c_1 \frac{u}{m}\hat{\mathbf{i}}_{\mathbf{u}}\right) - \lambda_m c_2 u + \alpha u + (1-\alpha)u^2 \quad (3)$$

which, following Pontryagin theory, needs to be minimised by our controls along an optimal trajectory. Isolating the relevant part of the Hamiltonian that depends on the control $\hat{\mathbf{i}}_{\mathbf{u}}$ we have that $\mathscr{H} = \ldots + c_1 \frac{u}{m}\boldsymbol{\lambda}_v \cdot \hat{\mathbf{i}}_{\mathbf{u}} + \ldots$, which allows to conclude that, since m, u and c_1 are all positive numbers, the thrust direction must be in the opposite direction of $\boldsymbol{\lambda}_v$ for $\mathscr{H}$ to be minimised, more formally:

$$\hat{\mathbf{i}}^*_{\mathbf{u}} = -\frac{\boldsymbol{\lambda}_v}{\lambda_v} \quad (4)$$

Substituting this expression back into the Hamiltonian we get

$$\mathscr{H}(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}) = \boldsymbol{\lambda}_r \cdot \mathbf{v} - \frac{\mu}{r^3}\boldsymbol{\lambda}_v \cdot \mathbf{r} - c_1 \frac{u}{m}\lambda_v - \lambda_m c_2 u + \alpha u + (1-\alpha)u^2$$

which we now need to minimise with respect to u. Isolating the relevant part of $\mathscr{H}$, i.e. the terms that depend on u, we have

$$\mathscr{H} = \ldots + (1-\alpha)u^2 + u\left(\alpha - \frac{c_1}{m}\lambda_v - \lambda_m c_2\right) + \ldots$$

which is a convex parabola that will take its minimal value in $u = \frac{\frac{c_1}{m}\lambda_v + \lambda_m c_2 - \alpha}{2(1-\alpha)}$. Since $u \in [0, 1]$, we get the final expression for an optimal u^*:

$$u^* = \min\left[\max\left(\frac{\frac{c_1}{m}\lambda_v + \lambda_m c_2 - \alpha}{2(1-\alpha)}, 0\right), 1\right] \quad (5)$$

Note that in the corner case $\alpha = 1$, which corresponds to a mass-optimal control, the above expression results to be singular and the minimiser of the Hamiltonian can be conveniently rewritten introducing the switching function

$$S_{sw} = c_1\lambda_v + m c_2 \lambda_m - m\alpha$$

as

$$u^* = \begin{cases} 1 \text{ if } & S_{sw} > 0 \\ 0 \text{ if } & S_{sw} < 0 \end{cases} \quad (6)$$

4.3 The Co-state Equations

The states $\mathbf{x}(t)$ and co-states $\boldsymbol{\lambda}(t)$ must be a solution to the set of differential equations that are elegantly written using the Hamiltonian formalism as

$$\begin{aligned}\dot{\mathbf{r}} &= \frac{\partial \mathscr{H}}{\partial \boldsymbol{\lambda}_r}, \;\; \dot{\boldsymbol{\lambda}}_r = -\frac{\partial \mathscr{H}}{\partial \mathbf{r}} \\ \dot{\mathbf{v}} &= \frac{\partial \mathscr{H}}{\partial \boldsymbol{\lambda}_v}, \;\; \dot{\boldsymbol{\lambda}}_v = -\frac{\partial \mathscr{H}}{\partial \mathbf{v}} \\ \dot{m} &= \frac{\partial \mathscr{H}}{\partial \lambda_m}, \; \dot{\lambda}_m = -\frac{\partial \mathscr{H}}{\partial m}\end{aligned}$$

while for the first three equations it is trivial to get back the expressions in Equation (1), the co-states equations do require some extra work, in particular when deriving the gravity gradient. Let us then start deriving the co-state equations by computing $\frac{\partial \mathscr{H}}{\partial \mathbf{r}}$ starting from the expression in Equation (3). We have

$$\frac{\partial \mathscr{H}}{\partial \mathbf{r}} = -\boldsymbol{\lambda}_v \cdot \nabla \left(\frac{\mu}{r^3} \mathbf{r} \right)$$

where the symbol ∇ denotes the gradient operator. It is easier to compute the expression regrouping as follows:

$$\begin{aligned}-\frac{1}{\mu} \frac{\partial \mathscr{H}}{\partial \mathbf{r}} &= \nabla \left(\boldsymbol{\lambda}_v \cdot \frac{1}{r^3} \mathbf{r} \right) = \\ &= \frac{1}{r^3} \nabla \left(\boldsymbol{\lambda}_v \cdot \mathbf{r} \right) + \left(\boldsymbol{\lambda}_v \cdot \mathbf{r} \right) \nabla \left(\frac{1}{r^3} \right) = \\ &= \frac{\boldsymbol{\lambda}_v}{r^3} - 3 \left(\boldsymbol{\lambda}_v \cdot \mathbf{r} \right) \frac{\nabla r}{r^4} = \\ &= \frac{\boldsymbol{\lambda}_v}{r^3} - 3 \left(\boldsymbol{\lambda}_v \cdot \mathbf{r} \right) \frac{\mathbf{r}}{r^5}\end{aligned}$$

We may thus write the corresponding Hamilton equation as

$$\dot{\boldsymbol{\lambda}}_r = -\frac{\partial \mathscr{H}}{\partial \mathbf{r}} = \mu \frac{\boldsymbol{\lambda}_v}{r^3} - 3\mu \left(\boldsymbol{\lambda}_v \cdot \mathbf{r} \right) \frac{\mathbf{r}}{r^5}$$

The remaining two equations are then also easily obtained as

$$\dot{\boldsymbol{\lambda}}_v = -\frac{\partial \mathscr{H}}{\partial \mathbf{v}} = -\boldsymbol{\lambda}_r$$

and

$$\dot{\lambda}_m = -\frac{\partial \mathscr{H}}{\partial m} = c_1 u \frac{\boldsymbol{\lambda}_v \cdot \hat{\mathbf{i}}_{\mathbf{u}}}{m^2}$$

The final system of differential equations describing the state and co-states evolution along an optimal trajectory may be now summarised:

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = -\frac{\mu}{r^3}\mathbf{r} - c_1 \frac{u^*(t)}{m} \frac{\boldsymbol{\lambda}_v}{\lambda_v} \\ \dot{m} = -c_2 u^*(t) \\ \dot{\boldsymbol{\lambda}}_r = \mu \frac{\boldsymbol{\lambda}_v}{r^3} - 3\mu \left(\boldsymbol{\lambda}_v \cdot \mathbf{r}\right) \frac{\mathbf{r}}{r^5} \\ \dot{\boldsymbol{\lambda}}_v = -\boldsymbol{\lambda}_r \\ \dot{\lambda}_m = -c_1 u^*(t) \frac{\lambda_v}{m^2} \end{cases} \tag{7}$$

4.4 The Two-Point Boundary Value Problem

Following the maximum principle, we know that there exist initial values for the co-states $\boldsymbol{\lambda}(t_1) = \boldsymbol{\lambda}_1$ such that an optimal trajectory steering the system from a starting state $\mathbf{x}_1 \in \mathscr{S}_1$ to a final state $\mathbf{x}_2 \in \mathscr{S}_2$ will be found integrating the system of equations stated in Equation (7) from the initial condition $\mathbf{x}_1, \boldsymbol{\lambda}_1$, for a time $t_2 - t_1$ leading to a final condition $\mathbf{x}_2, \boldsymbol{\lambda}_2$. Since we also want to find optimal values for $\mathbf{x}_1 \in \mathscr{S}_1$, $\mathbf{x}_2 \in \mathscr{S}_2$ and t_2 we need some additional conditions. The problem of finding the optimal t_2 (t_1 can be assumed to be 0 as our system is autonomous) is solved by adding a condition on the Hamiltonian:

$$\mathscr{H}(\mathbf{x}(t_2), \boldsymbol{\lambda}(t_2), \mathbf{u}^*(t_2)) = 0 \tag{8}$$

To optimally select $\mathbf{x}_1 \in \mathscr{S}_1$ and $\mathbf{x}_2 \in \mathscr{S}_2$, instead, we will need to add some conditions, called transversality conditions and derived in the next section.

4.5 Transversality Conditions

Transversality conditions ensure that the initial and final states are selected optimally in the allowed sets $\mathscr{S}_1$ and $\mathscr{S}_2$. They can be elegantly written introducing the vectors $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ belonging to the hyperplane tangent to the hypersurfaces $\mathscr{S}_1$ and $\mathscr{S}_2$ in $\mathbf{x}_1$ and $\mathbf{x}_2$. For all such vectors it must be that

$$\boldsymbol{\lambda}(t_1) \cdot \boldsymbol{\theta}_1 = 0$$

and equivalently,

$$\boldsymbol{\lambda}(t_2) \cdot \boldsymbol{\theta}_2 = 0$$

Let us now derive explicitly these relations in the case of the interplanetary transfer dynamics. We start focusing on the terminal condition $\mathbf{x}_2 \in \mathscr{S}_2$. Typically the final mass of the spacecraft is not specified and we may, in this case, express $\mathscr{S}_2$ as $\mathbf{x}_2 = [\mathbf{r}_2, \mathbf{v}_2, c]$ with c being a free parameter and $\mathbf{r}_2 and \mathbf{v}_2$ being at this stage considered as fixed. Trivially, in this case, $\boldsymbol{\theta}_2 = [\mathbf{0}, \mathbf{0}, \theta_2]$ and the transversality condition for a free final mass is thus

$$\boldsymbol{\lambda}_m(t_2) = 0 \tag{9}$$

More work is needed to derive transversality conditions in the case where also $\mathbf{r}_2$ *and* $\mathbf{v}_2$ are not fixed but constrained to belong to some manifold. A typical situation would be, for example, to leave some of the final Keplerian orbital elements as free. In this case, in order to find $\boldsymbol{\theta}_2$, i.e. the vector tangent to the defined manifold, we need to write a parameterisation of such manifold, that is an expression of $\mathbf{r}_2$ and $\mathbf{v}_2$ as a function of the desired orbital parameter. Using the Keplerian elements $a, e, i, \omega, \Omega, E$ we may use well-known relations to establish such a parameterisation:

$$\begin{aligned} \mathbf{r}_2 &= \mathbf{R} \begin{bmatrix} a(\cos E - e) \\ a\sqrt{1-e^2}\sin E \\ 0 \end{bmatrix} \\ \mathbf{v}_2 &= \sqrt{\tfrac{\mu}{a^3}} \tfrac{1}{1-e\cos E} \mathbf{R} \begin{bmatrix} -a\sin E \\ a\sqrt{1-e^2}\cos E \\ 0 \end{bmatrix} \end{aligned} \tag{10}$$

where $\mathbf{R}$ is the rotation matrix from the orbital reference frame to the inertial defined as

$$\mathbf{R} = \begin{bmatrix} \cos\Omega\cos\omega - \sin\Omega\sin\omega\cos i & -\cos\Omega\sin\omega - \sin\Omega\cos\omega\cos i & \sin\Omega\sin i \\ \sin\Omega\cos\omega + \cos\omega\sin\omega\cos i & -\sin\Omega\sin\omega + \cos\Omega\cos\omega\cos i & -\cos\Omega\sin i \\ \sin\omega\sin i & \cos\omega\cos i & \cos i \end{bmatrix}$$

the tangent vector to the resulting manifold will then be simply obtained deriving the above expression with respect to the chosen parameter.

For example, let us take the case of a free final anomaly. This corresponds to a transfer to some final target orbit regardless of the phase. Deriving the above expressions with respect to E, or in this case equivalently t, we get the transversality condition:

$$\boldsymbol{\lambda}_r \cdot \mathbf{v}_2 - \boldsymbol{\lambda}_v \cdot \frac{\mu}{r_2^3}\mathbf{r}_2 = 0 \tag{11}$$

Similarly if we leave the semi-major axis a as free, the conditions are easily obtained as

$$2\boldsymbol{\lambda}_r \cdot \mathbf{r}_2 - \boldsymbol{\lambda}_v \cdot \mathbf{v}_2 = 0 \tag{12}$$

Similar expressions are derived in [36] using true anomaly and not the eccentric anomaly and using a different method.

5 Test Case Description (Nominal Trajectories)

As a test case, we consider a low-thrust Earth–Mars orbital transfer. A spacecraft having mass $m_0 = 1000$ (kg) and equipped with a propulsion system able to deliver a constant $T_{max} = 0.3$ (N) with a specific impulse of $I_{sp} = 2500$ (s) starts its transfer from Earth's orbit to reach, after a time of flight Δt, Mars' orbit. Both starting and target orbits are considered Keplerian. The optimal transfer strategy is found by solving the resulting TPBVP where we seek the time of flight Δt, departure eccentric anomaly E_0, arrival eccentric anomaly E_f and departure co-state variables λ_0 that satisfy the boundary conditions, the dynamics Equation (7), the transversality condition on free time Equation (8), free mass Equation (9) and free anomalies Equation (12).

The TPBVP is solved first for quadratic control optimality (QOC - $\alpha = 0$). Once the optimal trajectory is found, the problem is then solved with gradually increasing α until mass-optimal control (MOC) is achieved $\alpha = 1$, at which point both solutions (nominal trajectories visualised in Figure 1) are stored as

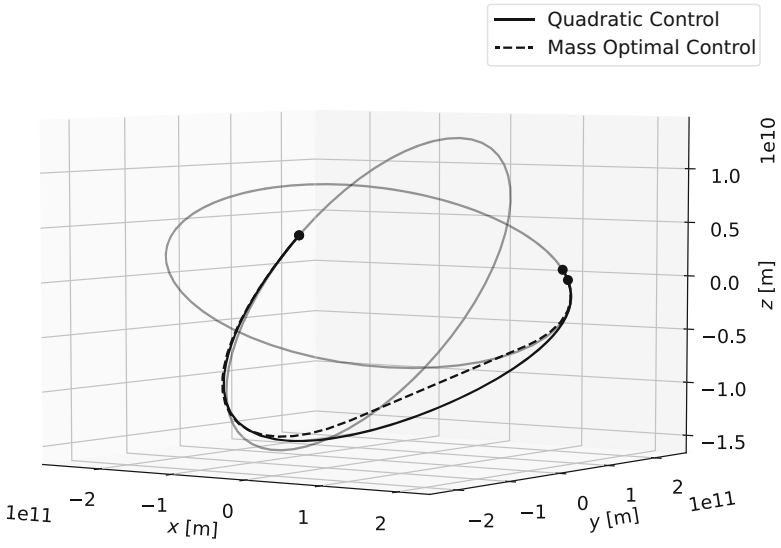


Figure 1 The two nominal trajectories considered

$$\mathbf{T}_{nom} = \left\{ (\mathbf{x}, \boldsymbol{\lambda}, t)_{nom,j} \right\}, j = 0, \ldots, J_{nom}$$

where the grid points are determined by the adaptive integrator used. This will result in more points allocated in areas where the dynamics gradient is higher, which are also areas where we would want to have more training samples, and is thus considered as an appropriate mechanism.

6 Generating the QOC Data Set

As we want to train a deep neural architecture to represent the optimal control around our nominal trajectories, we need to generate a large data set describing the functional relationship that is to be learned. Such a functional relationship is the optimal state feedback ($\mathbf{x} \rightarrow \mathbf{u}^*$) which can be computed along single trajectories neighbouring the nominal ones at the cost of solving the resulting TPBVPs. In order to make use of previously computed solutions (starting from the nominal trajectories) to help the convergence of the TPBVP solver, a continuation technique is then employed where new TPBVPs are created using as initial states those of the nominal trajectories perturbed along a random walk and as initial co-states those computed from previous steps. This allows to generate efficiently the database continuing one starting solution, i.e. the nominal trajectory. The TPBVP is solved by means of a simple single shooting method. The pseudo algorithm used to fill the data set is given in Algorithm 1 which is run starting from ten different points equally spaced along the nominal trajectory and using $\alpha = 0$. Let the final data set

$$\mathscr{T}_{\text{QOC}} = \{\mathbf{T}_i\}, i = 0, \ldots, I \tag{13}$$

contain I trajectories

$$\mathbf{T}_i = \left\{ (\mathbf{x}, \boldsymbol{\lambda}, t)_{i,j} \right\}, j = 0, \ldots, J_i \tag{14}$$

sampled in J_i nodes (as determined by the outcome of an adaptive step numerical integration) each recording the states $\mathbf{x}$, the co-states $\boldsymbol{\lambda}$ and the times t. We refer to this data set as QOC (quadratic optimal control).

6.1 Generating the MOC Data Set

A second data set containing mass-optimal trajectories is obtained via continuation over the parameter α (homotopy). Iterating through the initial states and co-states in $\mathscr{T}_{\text{QOC}}$, an attempt is made to solve, starting from that guess, directly a mass-optimal control ($\alpha = 1$). If the resulting trajectory is feasible, it is stored and the next data set entry is considered. If the trajectory is not feasible, the homotopy

Algorithm 1 Random walk

procedure RANDOM WALK($\mathbf{x}_{nom,j}, \boldsymbol{\lambda}_{nom,j}, \Delta t_{nom,j}, E_{f_{nom}}, \alpha, \overline{\gamma}, n$)
  $\mathscr{T} \leftarrow \{\}$ ▷ Set of optimal trajectories
  $\mathbf{x}_0 \leftarrow \mathbf{x}_{nom,j}$ ▷ Set initial state as nominal
  $\boldsymbol{\lambda}_0 \leftarrow \boldsymbol{\lambda}_{nom,j}$ ▷ Set initial co-state as nominal
  $\Delta t \leftarrow \Delta t_{nom,j}$ ▷ Set time of flight as nominal
  $E_f \leftarrow E_{f_{nom}}$ ▷ Set arrival eccentric anomaly as nominal
  $\gamma \leftarrow \overline{\gamma}$ ▷ Nominal perturbation percentage
  for $i \leftarrow 1, \ldots, n$ **do** ▷ n random perturbations
    $\boldsymbol{\beta} \leftarrow U(-1, 1) \in \mathbb{R}^7$ ▷ Vector of random uniformly distributed numbers
    $\mathbf{x}_1 \leftarrow \mathbf{x}_0 + \mathbf{x}_0 \odot \boldsymbol{\beta}\gamma$ ▷ Perturb state in random direction by γ
    $(\boldsymbol{\lambda}_1, \Delta t_1, E_{f_1}) \leftarrow \text{TPBVP}(\mathbf{x}_1, \boldsymbol{\lambda}_0, \Delta t, E_f, \alpha)$
    if successful **then** ▷ If the TPBVP is successfully solved
      Update $\mathscr{T}$ ▷ Save successful trajectory
      $\mathbf{x}_0 \leftarrow \mathbf{x}_1$ ▷ Accept perturbed state
      $\boldsymbol{\lambda}_0 \leftarrow \boldsymbol{\lambda}_1$ ▷ Accept new computed co-states
      $\Delta t_0 \leftarrow \Delta t_1$ ▷ Accept new computed time of flight
      $E_{f_0} \leftarrow E_{f_1}$ ▷ Accept new computed eccentric anomaly
      $\gamma \leftarrow (\gamma + \overline{\gamma})/2$ ▷ Increase perturbation size
    else ▷ If solution is unfeasible
      $\gamma \leftarrow \gamma/2$ ▷ Decrease perturbation size and solve again
    end if
  end for
  return $\mathscr{T}$ ▷ Return set of optimal trajectories
end procedure

parameter is decreased, and the optimisation is reattempted. If successful, the homotopy parameter is saved as the current best and its current value is increased. If, instead, the TPBVP solver is not converging, the homotopy parameter is decreased, and the algorithm continues. The pseudo code describing the homotopy method is outlined in Algorithm 2 and results in a new data set $\mathscr{T}_{\text{MOC}}$ containing mass-optimal trajectories and visualised in Figure 2.

7 Learning Details and Network Architecture

We consider for both of the data sets, $\mathscr{T}_{\text{MOC}}$ and $\mathscr{T}_{\text{QOC}}$, the optimal state-action pairs $(\mathbf{x}, \mathbf{u})$ and we build a deep model of the relation $\mathbf{u}(\mathbf{x})$, i.e. the optimal state feedback. The dimension of the values we have to model is $D = 3$ as they represent a three dimensional vector (i.e. the thrust vector). We choose to represent such a vector in its polar coordinates so that

$$\mathbf{u} = u\,[\sin\theta\cos\phi, \sin\theta\sin\pi, \cos\theta]$$

where θ is the polar angle, ϕ is the azimuth angle and u is the throttle magnitude.

Algorithm 2 Homotopy

procedure HOMOTOPY($\mathbf{x}_{QOC,0}, \boldsymbol{\lambda}_{QOC,0}, \Delta t_{QOC}, E_{f_{QOC}}, \alpha_{\text{tol}}$)
  $\alpha \leftarrow 1$ ▷ Try to solve for mass-optimal at first
  loop
    $(\boldsymbol{\lambda}, \Delta t, E_f) \leftarrow$ TPBVP($\mathbf{x}_{QOC,0}, \boldsymbol{\lambda}_{QOC,0}, \Delta t_{QOC}, E_{f_{QOC}}, \alpha$)
    if successful **then** ▷ If the TPBVP is successfully solved
      $\alpha^\star \leftarrow \alpha$ ▷ Current best α
      $(\boldsymbol{\lambda}_{QOC,0}, \Delta t_{QOC}, E_{f_{QOC}}) \leftarrow (\boldsymbol{\lambda}, \Delta t, E_f)$ ▷ $\mathbf{x}_0^\star$ doesn't change
      if $\alpha < \alpha_{\text{tol}}$ **then**
        $\alpha \leftarrow (1 + \alpha)/2$ ▷ Increase α
      else if $\alpha \geq \alpha_{\text{tol}}$ **then** ▷ If α is close to 1
        $\alpha \leftarrow 1$
      else if $\alpha = 1$ **then** ▷ If trajectory is feasible and mass-optimal
        return T ▷ Break the loop and return trajectory
      end if
    else
      $\alpha \leftarrow (\alpha + \alpha^\star)/2$ ▷ Decrease α and retry
    end if
  end loop
end procedure

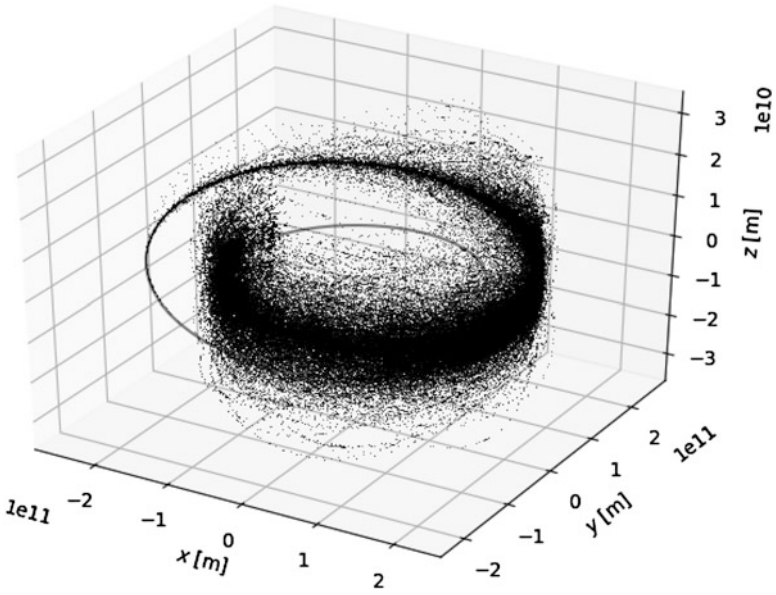


Figure 2 Visualisation of the MOC data set containing 308,579 optimal state-control pairs around the nominal trajectory

In previous work [42], it was found that a feed-forward, fully connected neural network with 3 hidden layers and 32 units/layer could satisfactorily represent the optimal guidance profile for a problem with simpler dynamics. Starting from that knowledge, and from the fact that we are here considering a higher dimensional

case with a higher degree of complexity, we experimented with deeper and wider networks. The result of our investigation indicated that peak performance could be obtained by a neural network with four hidden layers and 200 units per layer. Larger networks either matched this performance or performed worse. We settled on the rectified linear unit (ReLU) as the activation function for the hidden layers. In addition, we used a hyperbolic tangent activation function for the output layer. Thus we scaled the targets in the data set to the range $[-1, 1]$. We initialised the network's weights using the heuristic in [16] in which random values are sampled from a uniform distribution close to zero. Biases were initialised to zero. We also normalised our input data such that features had a mean of zero and standard deviation of one; this helped to speed up the optimisation process. We used the Adam training algorithm [29], a variant of stochastic gradient descent. Weights were updated based on a minibatch of 64 training examples. For the performance metric we use the commonly used mean squared error loss function (MSE). A 10% split is used to define training and validation data. We used an adaptive learning rate starting from an initial value of 10^{-3}. The learning rate was reduced by a factor of 10 when we observed a plateau in the training loss for a period greater than 10 epochs. Furthermore, we stopped training when we observed a plateau in the training loss for greater than 50 epochs ('early stopping'). The plateau was defined to be a decrease in loss less than 10^{-4}. We experimented with different values for the hyperparameters described above before settling on the final configuration.

8 Results

We train different networks to predict the three outputs both concurrently or separately and on both $\mathscr{T}_{\text{MOC}}$ and $\mathscr{T}_{\text{QOC}}$ data sets. We indicate each network using superscripts and subscripts. For example, $\mathcal{N}^{MOC}_{\phi,\theta}$ indicates the network trained to model the variables ϕ and θ in the $\mathscr{T}_{\text{MOC}}$ data set. We report in Table 1 the obtained validation loss in the cases tested. We note that the best results for the output variables are obtained from different models indicating the absence of a clear

Table 1 MSE for neural networks trained on the MOC and QOC data sets

	MSE			
Network	u	ϕ	θ	Epochs
$\mathcal{N}^{QOC}_{u,\phi,\theta}$	0.0178	**0.0690**	**0.0870**	453
$\mathcal{N}^{QOC}_{\phi,\theta}$	–	0.0755	0.0907	461
$\mathcal{N}^{QOC}_{u}$	**0.0075**	–	–	247
$\mathcal{N}^{QOC}_{\phi}$	–	0.0698	–	394
$\mathcal{N}^{QOC}_{\theta}$	–	–	0.0976	503

	MSE			
Network	u	ϕ	θ	Epochs
$\mathcal{N}^{MOC}_{u,\phi,\theta}$	**0.7250**	0.0255	0.0337	794
$\mathcal{N}^{MOC}_{\phi,\theta}$	–	0.0212	**0.0289**	316
$\mathcal{N}^{MOC}_{u}$	0.7740	–	–	721
$\mathcal{N}^{MOC}_{\phi}$	–	**0.0204**	–	248
$\mathcal{N}^{MOC}_{\theta}$	–	–	0.0305	324

The networks differ on the output dimension

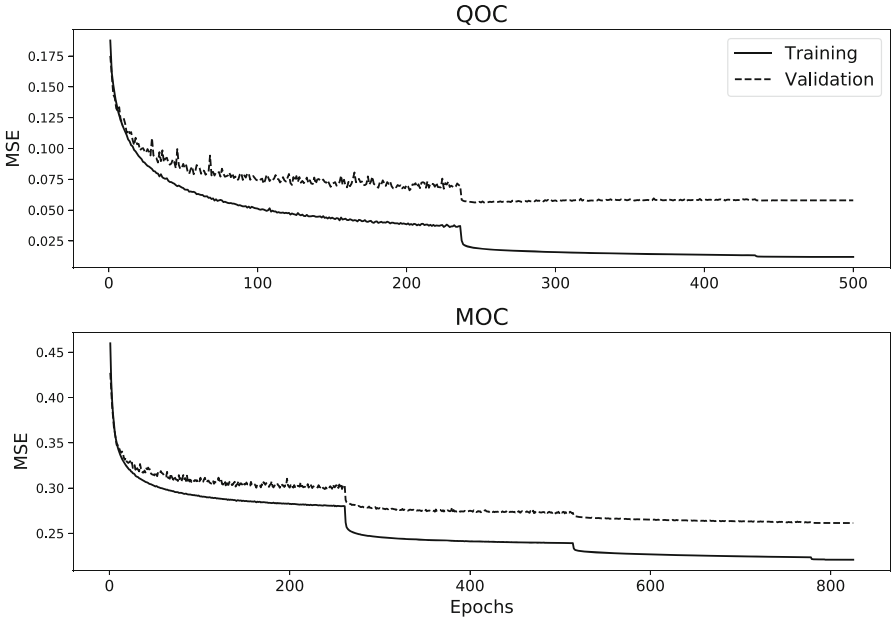


Figure 3 Loss history for the networks $\mathcal{N}_{u,\phi,\theta}^{QOC}$ and $\mathcal{N}_{u,\phi,\theta}^{MOC}$. Note the jumps correspond to the learning rate being reduced

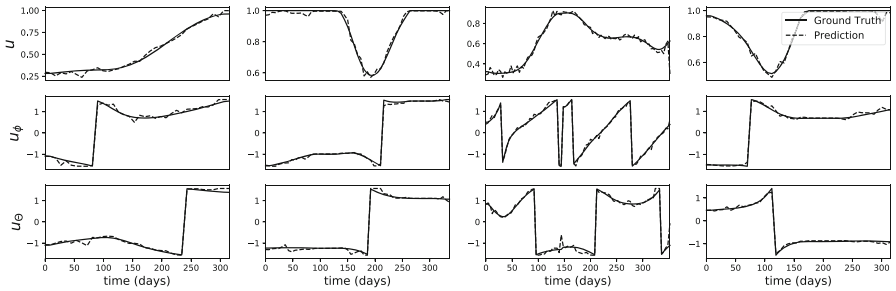


Figure 4 Network $\mathcal{N}_{u,\phi,\theta}^{QOC}$ predictions on four different transfers from the QOC test set

trend to be exploited, while one could expect that predicting the entire thrust vector at once would bring advantages as the network would be able to share some of the weights as part of the necessary computations. We also report in Figure 3 the loss during the training for the representative case of the $\mathcal{N}_{u,\phi,\theta}^{MOC}$ and $\mathcal{N}_{u,\phi,\theta}^{QOC}$ networks showing a relatively standard trend. We finally show four different optimal transfers and plot alongside them the optimal values against the predicted network values. The results are visualised in Figures 4 and 5.

It appears that a deep network is able to represent the optimal guidance structure of the Earth–Mars transfer quite satisfactorily introducing errors that are, on

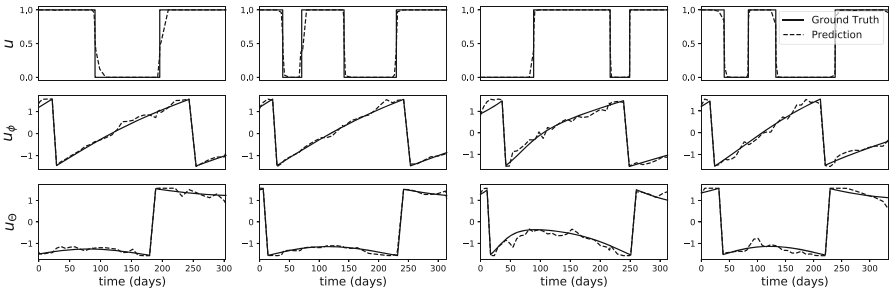


Figure 5 Network $\mathcal{M}_{u,\phi,\theta}^{MOC}$ predictions on four different transfers from the MOC test set

average, rather small. One should also keep in mind that the actual quantitative value of the losses can be improved by further fine-tuning of the hyperparameters used in training the models. In general, we observe the prediction of the throttle u is better attained in the QOC case. This is not surprising as the structure of the optimal control in the MOC case is highly discontinuous being a bang-bang control. We further note that data points close to the switching points are very difficult to predict correctly and when erroneously predicted result in big contributions to the overall loss. These points form a majority in the data sets; a consequence of the time grid defined by the numerical integrator for the TPBVP shooting method solver. This is the reason for the MSE on the u prediction to be of a greater magnitude in the case of the MOC networks, while the visualised plot of the predictions for the same cases is actually returning a much better scenario.

A definitive assessment on the capability of the trained deep models to steer correctly the spacecraft has to be conducted, in a similar fashion to what done in [42], by integrating forward the whole spacecraft dynamics considering the deep network predictions as a control and will be part of a separate future publication.

References

1. Addis, B., Cassioli, A., Locatelli, M., Schoen, F.: A global optimization method for the design of space trajectories. Comput. Optim. Appl. **48**(3), 635–652 (2011)
2. Ampatzis, C., Izzo, D.: Machine learning techniques for approximation of objective functions in trajectory optimisation. In: Proceedings of the IJCAI-09 Workshop on Artificial Intelligence in Space, pp. 1–6 (2009)
3. Biesbroek, R.G., Ancarola, B.P.: Optimization of launcher performance and interplanetary trajectories for pre-assessment studies. In: IAF Abstracts, 34th COSPAR Scientific Assembly (2002)
4. Cassioli, A., Di Lorenzo, D., Locatelli, M., Schoen, F., Sciandrone, M.: Machine learning for global optimization. Comput. Optim. Appl. **51**(1), 279–303 (2012)
5. Ceriotti, M., Vasile, M.: MGA trajectory planning with an ACO-inspired algorithm. Acta Astronaut. **67**(9), 1202–1217 (2010)

6. Dachwald, B.: Low-thrust trajectory optimization and interplanetary mission analysis using evolutionary neurocontrol. Ph.D. thesis, Doctoral thesis, Universität der Bundeswehr München Fakultät für Luft-und Raumfahrttechnik (2004)
7. Dachwald, B., Ohndorf, A.: Global optimization of continuous-thrust trajectories using evolutionary neurocontrol. In: Fasano, G., Pinter, J. (eds.) Modeling and Optimization in Space Engineering - 2018. Springer, Basel (2019)
8. de Croon, G., Izzo, D.: Real-time landing based on optimality principles and vision. In: 23rd International Symposium on Space Flight Dynamics (ISSFD) (2012)
9. Deb, K., Padhye, N., Neema, G.: Interplanetary trajectory optimization with swing-bys using evolutionary multi-objective optimization. In: International Symposium on Intelligence Computation and Applications, pp. 26–35. Springer, Berlin (2007)
10. Di Lizia, P., Radice, G.: Advanced global optimisation for mission analysis and design. Final Report Ariadna id 04/4101 (2004)
11. Dueri, D., Açıkmeşe, B., Scharf, D.P., Harris, M.W.: Customized real-time interior-point methods for onboard powered-descent guidance. J. Guid. Control. Dyn. **40**, 197–212 (2016)
12. Elsayed, S.M., Sarker, R.A., Essam, D.L.: GA with a new multi-parent crossover for solving IEEE-CEC2011 competition problems. In: IEEE Congress on Evolutionary Computation (CEC), 2011, pp. 1034–1040. IEEE, Piscataway (2011)
13. Englander, J.: Automated trajectory planning for multiple-flyby interplanetary missions. University of Illinois at Urbana-Champaign (2013)
14. Gad, A., Abdelkhalik, O.: Hidden genes genetic algorithm for multi-gravity-assist trajectories optimization. J. Spacecr. Rocket. **48**(4), 629–641 (2011)
15. Gage, P., Braun, R., Kroo, I.: Interplanetary trajectory optimization using a genetic algorithm. J. Astronaut. Sci. **43**(1), 59–76 (1995)
16. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010)
17. Grigoriev, I., Zapletin, M.: Choosing promising sequences of asteroids. Autom. Remote. Control. **74**(8), 1284–1296 (2013)
18. Hennes, D., Izzo, D.: Interplanetary trajectory planning with Monte Carlo tree search. In: IJCAI, pp. 769–775 (2015)
19. Hennes, D., Izzo, D., Landau, D.: Fast approximators for optimal low-thrust hops between main belt asteroids. In: IEEE Symposium Series on Computational Intelligence (SSCI), 2016, pp. 1–7. IEEE, Piscataway (2016)
20. Islam, S.M., Das, S., Ghosh, S., Roy, S., Suganthan, P.N.: An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. IEEE Trans. Syst. Man Cybern. B Cybern. **42**(2), 482–500 (2012)
21. Izzo, D.: Global optimization and space pruning for spacecraft trajectory design. Spacecr. Trajectory Optim. **1**, 178–200 (2010)
22. Izzo, D., Becerra, V.M., Myatt, D.R., Nasuto, S.J., Bishop, J.M.: Search space pruning and global optimisation of multiple gravity assist spacecraft trajectories. J. Glob. Optim. **38**(2), 283–296 (2007)
23. Izzo, D., Simões, L.F., Märtens, M., De Croon, G.C., Heritier, A., Yam, C.H.: Search for a grand tour of the Jupiter Galilean moons. In: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, pp. 1301–1308. ACM, New York (2013)
24. Izzo, D., Hennes, D., Riccardi, A.: Constraint handling and multi-objective methods for the evolution of interplanetary trajectories. J. Guid. Control. Dyn. **38**, 792–800 (2014)
25. Izzo, D., Simoes, L.F., Yam, C.H., Biscani, F., Di Lorenzo, D., Addis, B., Cassioli, A.: GTOC5: results from the European Space Agency and University of Florence. Acta Futura **8**, 45–55 (2014)
26. Izzo, D., Getzner, I., Hennes, D., Simões, L.F.: Evolving solutions to TSP variants for active space debris removal. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 1207–1214. ACM, New York (2015)

27. Izzo, D., Hennes, D., Simões, L.F., Märtens, M.: Designing complex interplanetary trajectories for the global trajectory optimization competitions. In: Space Engineering, pp. 151–176. Springer, Berlin (2016)
28. Janin, G., Gomez-Tierno, M.: The genetic algorithms for trajectory optimization. In: Stockholm International Astronautical Federation Congress (1985)
29. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2014). CoRR abs/1412.6980. http://arxiv.org/abs/1412.6980, 1412.6980
30. Lavagna, M.R.: Multi-objective pso for interplanetary trajectory design. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, pp. 175–175. ACM, New York (2007)
31. Lee, S., von Allmen, P., Fink, W., Petropoulos, A., Terrile, R.: Multi-objective evolutionary algorithms for low-thrust orbit transfer optimization. In: Genetic and Evolutionary Computation Conference (GECCO 2005) (2005)
32. Luo, Y.Z., Tang, G.J., Zhou, L.N.: Simulated annealing for solving near-optimal low-thrust orbit transfer. Eng. Optim. **37**(2), 201–216 (2005)
33. Mereta, A., Izzo, D., Wittig, A.: Machine learning of optimal low-thrust transfers between near-earth objects. In: International Conference on Hybrid Artificial Intelligence Systems, pp. 543–553. Springer, Berlin (2017)
34. Myatt, D., Becerra, V.M., Nasuto, S.J., Bishop, J.: Advanced global optimisation for mission analysis and design. Final Report Ariadna id 04/4101 (2004)
35. Olds, A.D., Kluever, C.A., Cupples, M.L.: Interplanetary mission design using differential evolution. J. Spacecr. Rocket. **44**(5), 1060–1070 (2007)
36. Pan, B., Chen, Z., Lu, P., Gao, B.: Reduced transversality conditions in optimal space trajectories. J. Guid. Control. Dyn. **36**, 1289–1300 (2013)
37. Pontani, M., Conway, B.A.: Particle swarm optimization applied to space trajectories. J. Guid. Control. Dyn. **33**(5), 1429–1441 (2010)
38. Pontryagin, L.S., Boltyanskii, V., Gamkrelidze, R., Mishchenko, E.F.: The Mathematical Theory of Optimal Processes. Interscience, New York (1962)
39. Radice, G., Olmo, G.: Ant colony algorithms for two-impulse interplanetary trajectory optimization. J. Guid. Control. Dyn. **29**(6), 1440 (2006)
40. Rauwolf, G.A., Coverstone-Carroll, V.L.: Near-optimal low-thrust orbit transfers generated by a genetic algorithm. J. Spacecr. Rocket. **33**(6), 859–862 (1996)
41. Rogata, P., Di Sotto, E., Graziano, M., Graziani, F.: Guess value for interplanetary transfer design through genetic algorithms. Adv. Astronaut. Sci. **114**, 613–627 (2003)
42. Sánchez-Sánchez, C., Izzo, D.: Real-time optimal control via deep neural networks: study on landing problems (2016). arXiv preprint arXiv:161008668
43. Sánchez-Sánchez, C., Izzo, D., Hennes, D.: Learning the optimal state-feedback using deep networks. In: IEEE Symposium Series on Computational Intelligence (SSCI), 2016, pp. 1–8. IEEE, Piscataway (2016)
44. Schiavone, G., Izzo, D., Simões, L.F., De Croon, G.C.: Autonomous spacecraft landing through human pre-attentive vision. Bioinspir. Biomim. **7**(2), 025,007 (2012)
45. Schlueter, M.: MIDACO software performance on interplanetary trajectory benchmarks. Adv. Space Res. **54**(4), 744–754 (2014)
46. Schlueter, M., Erb, S.O., Gerdts, M., Kemble, S., Rückmann, J.J.: MIDACO on MINLP space applications. Adv. Space Res. **51**(7), 1116–1131 (2013)
47. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015)
48. Sentinella, M.R., Casalino, L.: Hybrid evolutionary algorithm for the optimization of interplanetary trajectories. J. Spacecr. Rocket. **46**(2), 365 (2009)
49. Simões, L.F., Izzo, D., Haasdijk, E., Eiben, A.E.: Self-adaptive genotype-phenotype maps: neural networks as a meta-representation. In: International Conference on Parallel Problem Solving from Nature, pp. 110–119. Springer (2014)
50. Simões, L.F., Izzo, D., Haasdijk, E., Eiben, A.: Multi-rendezvous spacecraft trajectory optimization with beam P-ACO. In: European Conference on Evolutionary Computation in Combinatorial Optimization, pp. 141–156. Springer, Berlin (2017)

51. Stracquadanio, G., La Ferla, A., De Felice, M., Nicosia, G.: Design of robust space trajectories. In: SGAI Conference, pp. 341–354 Springer, Berlin (2011)
52. Vasile, M., Minisci, E., Locatelli, M.: Analysis of some global optimization algorithms for space trajectory design. J. Spacecr. Rocket. **47**(2), 334 (2010)
53. Vinkó, T., Izzo, D.: Global optimisation heuristics and test problems for preliminary spacecraft trajectory design. Eur Space Agency, Adv Concepts Team, ACT Tech Rep, id: GOHTPPSTD (2008)
54. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997)
55. Yao, W., Luo, J., Macdonald, M., Wang, M., Ma, W.: Improved differential evolution algorithm and its applications to orbit design. J. Guid. Control. Dyn. **41**, 1–8 (2017)

Real-Time Optimal Control Using TransWORHP and WORHP Zen

Matthias Knauer and Christof Büskens

Abstract In many industrial applications solutions of optimal control problems are used, where the need for low computational times outweighs the need for absolute optimality. For solutions of fully discretized optimal control problems we propose two methods to approximate the solutions of problems with modified parameter values in real-time by using sensitivity derivatives.

We use TransWORHP to transcribe an optimal control problem to a sparse nonlinear programming problem, which will be solved using our NLP solver WORHP. For this nominal solution sensitivity derivatives can be computed with respect to any system parameter using WORHP Zen. On NLP level, the sensitivity derivatives allow to perform correction steps for changes in the system parameters. This concept can be transferred to discretized optimal control problems using, e.g., the sensitivity derivatives of the boundary condition or of the discretized differential equations. The quality and applicability of both methods are illustrated by a trajectory planning problem in the context of the planar restricted problem of three bodies. In both methods the sensitivity derivatives can be used to give numerical validations of the theoretically expected convergence behaviour.

1 Introduction

Numerical solution methods for optimal control problems are used widely and successfully in many academic and industrial applications. The quality of a computed optimal solution depends on the used mathematical model and on the knowledge available at the start of the optimization process. If the solution is applied to operate a real system, perturbations will always occur. The initial state might be different than expected, or the environment might have changed since the solution

M. Knauer (✉) · C. Büskens
Universität Bremen, Zentrum für Technomathematik, Bremen, Germany
e-mail: knauer@math.uni-bremen.de; bueskens@math.uni-bremen.de

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_9

was computed. Of course, an updated optimal control problem can be solved now to handle the new situation.

In many cases, however, the solution of the updated optimal control problem would take too much time, even if the old solution could be used as a good initial guess. Different strategies have been developed to generate feasible trajectories within limited computation time:

Feedback control. Based on the dynamic system, a Riccati controller can be used to compensate for occurring perturbations and bring the states back on track, see [11].

Model predictive control. By considering only a finite time horizon for the optimal control problem, computation times can be reduced. In this method, any type of changing constraints can be considered, see [8, 10].

Sensitivity updates. Only cheap matrix–vector multiplications are needed to update the solution using sensitivity derivatives, see [5].

In this chapter, we will propose two methods for sensitivity updates for fully discretized optimal control problems. Using an example we will discuss the applicability and quality of both methods. We use TransWORHP to solve optimal control problems and WORHP Zen to generate the accompanying sensitivity derivatives. Both tools were developed especially for the NLP solver WORHP.

2 Optimal Control

How do I have to operate a system to guide it from an initial state to a final state without overstressing? This question, which applies to a wide range of problems (from landing trajectories of spaceships to medication of patients), is the template for optimal control problems:

$$\begin{aligned} \min_{x,u,t_f} \ & \phi(x(t_f)) + \int_0^{t_f} f_0(x(t), u(t))\, dt \\ \text{s.t.} \quad & \dot{x}(t) = f(x(t), u(t)), \quad t \in [0; t_f] \\ & x(0) = x_0 \\ & x(t_f) = x_f \\ & g(x(t), u(t)) \leq 0, \quad t \in [0; t_f] \end{aligned} \tag{1}$$

Here, a control function $u \in C_p^0([0; t_f], \mathbb{R}^{n_u})$ and a state function $x \in C_p^1([0; t_f], \mathbb{R}^{n_x})$ have to be determined for a fixed or free final time t_f. The dynamic behaviour of the system is modelled using a function $f \in C^1(\mathbb{R}^{n_x} \times \mathbb{R}^{n_u}, \mathbb{R}^{n_x})$. $C^j(I, \mathbb{R}^n)$ denotes the class of j times continuously differentiable functions from a domain $I \subset \mathbb{R}^k$ to $\mathbb{R}^n$, and $C_p^j(I, \mathbb{R}^n) \subset C^{j-1}(I, \mathbb{R}^n)$ the class of j times piecewise continuously differentiable functions.

For simplicity, let's for now allow only completely given initial and final states x_0 resp. x_f. The control $u(t)$ and the state $x(t)$ have to comply with inequality constraints using the function $g \in C^1(\mathbb{R}^{n_x} \times \mathbb{R}^{n_u}, \mathbb{R}^{n_g})$.

The optimal solution $(x^\star(t), u^\star(t))$ has to minimize the objective function consisting of the final term $\phi \in C^1(\mathbb{R}^{n_x}; \mathbb{R})$ and the integral of $f_0 \in C^1(\mathbb{R}^{n_x}, \mathbb{R}^{n_u})$. Note that the integral term can always be omitted by introducing a new differential equation

$$\dot{x}_+(t) = f_0(x(t), u(t)), \quad x_+(0) = 0 \tag{2}$$

and using $\phi(x(t_f)) + x_+(t_f)$ in the objective. Hence, we can assume $f_0 \equiv 0$ in the following.

2.1 Discretization

The infinite dimensional optimal control problem (1) can be approximated by a nonlinear programming problem (NLP) consisting of a large, but only finite number of parameters. See [2] or also [13] for a comparison of different methods. The so-called direct methods replace the continuous time interval $[0; t_f]$ by discrete grid points

$$0 = t_1 \le t_2 \le \cdots \le t_\ell = t_f, \quad \ell \in \mathbb{N} \tag{3}$$

Likewise, the control function $u(t)$ and the state function $x(t)$ are replaced by vectors of discrete values

$$u = (u^1, \ldots, u^\ell)^T, \quad x = (x^1, \ldots, x^\ell)^T \tag{4}$$

where $u^i \approx u(t_i)$ and $x^i \approx x(t_i)$.

Finally, the trapezoidal method is applied to the system of ordinary differential equations in (1), resulting in the discretized version of the optimal control problem, where $h_i := t_{i+1} - t_i$, $i = 1, \ldots, \ell - 1$ and $f^i = f(x^i, u^i)$:

$$\begin{aligned} \min_{x,u,t_f} \ & \phi(x^\ell) \\ \text{s.t.} \quad & x^{i+1} = x^i + h_i \tfrac{f^i + f^{i+1}}{2}, \quad i = 1, \ldots, \ell - 1 \\ & x^1 = x_0 \\ & x^\ell = x_f \\ & g(x^i, u^i) \le 0, \quad i = 1, \ldots, \ell \end{aligned} \tag{5}$$

For full discretization the discretized variables x and u can be grouped to a vector of optimization variables $z = (x, u) \in \mathbb{R}^{(n_x + n_u)\cdot \ell}$. In case of a free final time, t_f is also inserted into the vector z. The (in-)equality constraints of (5) can be grouped together as well. Alternatively, for shooting methods the discretized variables x are

only considered in the vector of optimization variables z for selected shooting points and are computed using Runge–Kutta schemes elsewhere. In both cases, (5) was led back to an NLP with n optimization variables and m constraints. Using the function $F \in C^1(\mathbb{R}^n, \mathbb{R})$ for the objective, and collecting all m_e equality and $m - m_e$ inequality constraints in $G \in C^1(\mathbb{R}^n, \mathbb{R}^m)$, we get:

$$\begin{aligned} \min_{z \in \mathbb{R}^n} \; & F(z) \\ \text{s.t.} \; & G_i(z) = 0, \quad i = 1, \ldots, m_e \\ & G_i(z) \leq 0, \quad i = m_e + 1, \ldots, m \end{aligned} \tag{6}$$

2.2 Numerical Solution

Sequential quadratic programming methods (SQP) can be used to solve (6) efficiently. Starting from an initial guess they approximate the solution of the nonlinear problem by a sequence of solutions of quadratic problems (QP). The ESA NLP solver WORHP implements a sparse SQP method to solve systems with millions of variables under millions of constraints, see [6]. Obviously, also for smaller problems the sparsity of the derivative matrices can be exploited to keep computation times small. Fully discretized problems (5) naturally yield highly structured NLPs.

To solve an optimal control problem (1) the software library TransWORHP generates the finite dimensional problem (5) and uses WORHP to solve it, see [9]. Different transcription schemes are available in TransWORHP:

Full discretization. The user can currently choose between Euler's method, the trapezoidal rule and Hermite–Simpson.

Multiple shooting. All discretized control variables, but only selected state variables are optimized. This method can be used with any Runge–Kutta method or Runge–Kutta–Fehlberg method for integration.

Pseudospectral methods. The discretized control and state variables are interpreted as control points of a polynomial of high order resulting in smooth functions.

TransWORHP was especially designed to solve problems consisting of multiple phases. The user can define a set of optimal control problems and connect their discretized versions to one large NLP.

3 Parametric Nonlinear Programming

If the functions F and G in (6) additionally depend on a parameter $p \in \mathbb{N}^{n_p}$, a parametric nonlinear programming problem is formulated. After an optimal solution was found for a fixed $p = p_0$, it is known, which constraints are active or inactive.

Without restriction the inactive inequality constraints can be removed from the problem, and only $m_a \leq m$ active constraints G^a remain:

$$\begin{aligned} &\min_{z \in \mathbb{R}^n} \; F(z, p) \\ &\text{s.t.} \;\; G_i^a(z, p) = 0, \quad i = 1, \ldots, m_a \end{aligned} \tag{7}$$

For an optimal solution of (7) for a fixed $p = p_0$, the tool WORHP Zen can be used to compute sensitivity derivatives $\frac{dz}{dp}(p_0)$. The proof of the sensitivity theorem, as given, for example, by Fiacco et al. [7], shows how the sensitivity derivatives can be computed without measurable computational cost by solving this system:

$$\begin{pmatrix} \nabla_z^2 L & \nabla_z G^{a\,T} \\ \nabla_z G^a & 0 \end{pmatrix} \cdot \begin{pmatrix} \frac{dz}{dp}(p_0) \\ \frac{d\lambda}{dp}(p_0) \end{pmatrix} = - \begin{pmatrix} \nabla_{zp} L \\ \nabla_p G^a \end{pmatrix} \tag{8}$$

Here, the Lagrange multipliers $\lambda \in \mathbb{R}^{m_a}$ and the Lagrange function

$$L(z, \lambda, p) := F(z, p) + \lambda^T G^a(z, p) \tag{9}$$

are used. If perturbations occur, i.e. $p \neq p_0$, the sensitivity derivatives can be used for real-time approximations of the solution of a perturbed problem:

$$z^{[1]} := z(p_0) + \frac{dz}{dp}(p_0) \cdot (p - p_0) \tag{10}$$

After this *pre-correction step* the constraint $G_i^a(z^{[1]}) \leq 0$ might not be fulfilled anymore and $z^{[1]}$ has to be considered as an infeasible solution. However for a sequence of *post-correction steps*, we additionally consider linear perturbations $q \in \mathbb{N}^m$ in the constraints of (7):

$$\begin{aligned} &\min_{z \in \mathbb{R}^n} \; F(z, p) \\ &\text{s.t.} \;\; G_i^a(z, p) + q = 0, \quad i = 1, \ldots, m_a \end{aligned} \tag{11}$$

For $q = 0$ the problems (7) and (11) coincide. The measured violation of the constraints, q, can be reduced iteratively for $i = 1, 2, 3, \ldots$:

$$z^{[i+1]} := z^{[i]} + \frac{dz}{dq}(0) \cdot q \tag{12}$$

Please note that after solving (11) the sensitivity derivatives $\frac{dz}{dq}(0)$ can be computed as in (8) for a simpler right-hand side, as $\nabla_{zq} L = 0$ and $\nabla_q (G^a + q) = I$.

Some important properties of these correction steps are proven in [4].

Theorem 1 (Convergence of Repeated Correction Method) *Let the necessary and sufficient optimality conditions of first and second order be fulfilled, and* $F, G^a \in C^3$.

Then there exists $U(p_0)$, $v \in \mathrm{Ker}(\nabla_z G^a(z(p_0), p_0))$, $\|v\| = \mathcal{O}(\|p - p_0\|^2)$ *such that for all* $p \in U(p_0)$, $i = 2, 3, 4, \ldots$:

$$\begin{aligned} \|z(p) - z^{[i]}\| &= \|v\| + \mathcal{O}(\|p - p_0\|^3) \\ \|F(z(p), p) - F(z^{[i]}, p)\| &= \mathcal{O}(\|p - p_0\|^3) \\ \|G^a(z^{[i]}, p)\| &= \mathcal{O}(\|p - p_0\|^{i+1}) \end{aligned} \tag{13}$$

The last equation ensures the convergence to a feasible solution, as

$$\|G^a(z^{[\infty]}, p)\| = 0 \tag{14}$$

4 Real-Time Optimal Control

Sensitivity derivatives for parametric optimal control problems with a perturbation parameter p in any function of (1) can be straightforwardly determined, at least when considering the discretized version (5). Further, the results of Theorem 1 can be transferred to optimal control problems and the sensitivity derivatives can be used for iteratively reducing errors in boundary conditions or path constraints while preserving optimality [3]. After solving the fully discretized optimal control problem for a nominal value $p = p_0$ and preparing the sensitivity differentials $\frac{dz}{dp}(p_0)$, approximative solutions for perturbations $p \neq p_0$ can be computed in real-time.

For comprehensibility, we restrict the presentation of the following methods to perturbations in the initial point and want to ensure that boundary conditions in the final point hold.

4.1 Using Sensitivity Derivatives of Boundary Condition

This method can be applied to fully discretized optimal control problems as well as to shooting methods:

$$\begin{aligned} \min_{x,u,t_f} \ & \phi(x^\ell) \\ \text{s.t.} \quad & x^{j+1} = x^j + h_j \tfrac{f^j + f^{j+1}}{2}, \quad j = 1, \ldots, \ell - 1 \\ & x^1 = x_0 + p \\ & x^\ell = x_f + q \\ & g(x^j, u^j) \leq 0, \quad j = 1, \ldots, \ell \end{aligned} \tag{15}$$

Initialize iteration counter $i := 1$
Update control to compensate initial perturbation:

$$u^{[1]} := u(p_0) + \frac{du}{dp}(p_0) \cdot (p - p_0)$$

Compute state $x^{[1]}$ by integration using $u^{[1]}$
Measure final state deviation:

$$\Delta q^{[i]} := x^{[i],\ell} - x_f$$

while $\|\Delta q^{[i]}\|_\infty > \varepsilon$ **do**
  Update control:

$$u^{[i+1]} := u^{[i]} + \frac{du}{dq}(0) \cdot \Delta q^{[i]}$$

  Compute state $x^{[i+1]}$ by integration using $u^{[i+1]}$
  Update final state deviation $\Delta q^{[i+1]}$
  Update iteration counter $i \leftarrow i + 1$
end

Algorithm 1: Real-time optimal control using sensitivity derivatives of boundary condition

To handle perturbations p at the initial point in the pre-correction step, the sensitivity derivative $\frac{du}{dp}(p_0) \in \mathbb{R}^{(n_u \cdot \ell) \times n_x}$ is needed. To ensure feasibility of the final point in the post-correction steps, an additional perturbation q is introduced to compute the sensitivity derivative $\frac{du}{dq}(0) \in \mathbb{R}^{(n_u \cdot \ell) \times n_x}$.

Algorithm 1 uses the derivatives $\frac{du}{dp}(p_0)$ and $\frac{du}{dq}(0)$ to generate an approximative solution of the perturbed optimal control problem in real-time.

To integrate the state in Algorithm 1 starting from $x^{[i],1} = x_0 + p$, the trapezoidal method has to be implemented. However, this requires the solution of implicit equations.

4.2 Using Sensitivity Derivatives of Discretized Differential Equations

This method only applies to fully discretized optimal control problems:

$$\begin{aligned} \min_{x,u,t_f} \ & \phi(x^\ell) \\ \text{s.t.} \ & x^{j+1} = x^j + h_j \tfrac{f^j + f^{j+1}}{2} + q^j, \quad j = 1, \ldots, \ell - 1 \\ & x^1 = x_0 + p \\ & x^\ell = x_f \\ & g(x^j, u^j) \le 0, \quad j = 1, \ldots, \ell \end{aligned} \tag{16}$$

Initialize iteration counter $i := 1$
Update control and state to compensate initial perturbation:

$$u^{[1]} := u(p_0) + \frac{du}{dp}(p_0) \cdot (p - p_0)$$

$$x^{[1]} := x(p_0) + \frac{dx}{dp}(p_0) \cdot (p - p_0)$$

Measure deviations in discretized differential equations:

$$\Delta q^{[i]} := \left(x^{[i],j+1} - x^{[i],j} - h_j \frac{f^{[i],j} + f^{[i],j+1}}{2} \right)_{j=1,\dots,\ell-1}$$

while $\|\Delta q^{[i]}\|_\infty > \varepsilon$ **do**
Update control and state:

$$u^{[i+1]} := u^{[i]} + \frac{du}{dq}(0) \cdot \Delta q^{[i]}$$

$$x^{[i+1]} := x^{[i]} + \frac{dx}{dq}(0) \cdot \Delta q^{[i]}$$

Update deviation in discretized differential equations $\Delta q^{[i+1]}$
Update iteration counter $i \leftarrow i + 1$
end

Algorithm 2: Real-time optimal control using sensitivity derivatives of discretized differential equations

To handle perturbations p at the initial point in the pre-correction step, the sensitivity derivatives $\frac{du}{dp}(p_0) \in \mathbb{R}^{(n_u \cdot \ell) \times n_x}$ and $\frac{dx}{dp}(p_0) \in \mathbb{R}^{(n_x \cdot \ell) \times n_x}$ are needed. After the pre-correction step the final point is still feasible, as the optimization variable x^ℓ is forced to equal x_f and hence the sensitivity of x^ℓ with respect to any perturbations is zero. However, the equations containing the discretized differential equations will not hold anymore. To enforce them in the post-correction steps, an additional perturbation $q = (q^j)_{j=1,\dots,\ell-1}$ is used to provide the sensitivity derivatives $\frac{du}{dq}(0) \in \mathbb{R}^{(n_u \cdot \ell) \times (n_x \cdot (\ell-1))}$ and $\frac{dx}{dq}(0) \in \mathbb{R}^{(n_x \cdot \ell) \times (n_x \cdot (\ell-1))}$.

Algorithm 2 uses the derivatives $\frac{du}{dp}(p_0)$, $\frac{dx}{dp}(p_0)$, $\frac{dx}{dq}(0)$ and $\frac{du}{dq}(0)$ to generate an approximative solution of the perturbed optimal control problem in real-time only using matrix–vector multiplications. We used the abbreviation $f^{[i],j} = f(x^{[i],j}, u^{[i],j})$. In case of a free final time t_f, the derivatives $\frac{dt_f}{dp}(p_0)$ and $\frac{dt_f}{dq}(0)$ have to be considered analogously for both methods. This is necessary, as the final time is expected to change for perturbed problems.

5 Numerical Results

The functionality of Algorithms 1 and 2 will be illustrated by the restricted problem of three bodies [1], which we expand to an optimal control problem.

5.1 Restricted Problem of Three Bodies

Two bodies with finite masses m_1 and m_2 are orbiting around their center of mass. The restricted problem of three bodies studies the motion of a test body with infinitesimal mass in the gravitational field of the masses m_1 and m_2. In particular the test body does not influence the other bodies.

The standard scaling of this problem is also suitable for optimization:

- The masses of the two bodies are scaled such that their sum is 1. Without restriction let $m_2 = \mu \leq \frac{1}{2}$. Then we get $m_1 = 1 - \mu$.
- The constant distance between the finite masses is $a = 1$.
- The time scale is chosen such that the gravitational constant is $G = 1$.
- The mean motion n of the masses depends on their period T and hence

$$n = \frac{2\pi}{T} = \sqrt{G\frac{(1-\mu)+\mu}{a^3}} = 1 \tag{17}$$

The coordinate system to describe the motion of the test body is defined such that the origin is located in the center of mass of the system, and the axes are rotating together with the masses in the x–y-plane such that the x-axis always goes through the masses m_1 and m_2. For simplicity, we will only consider the planar restricted problem of three bodies. The fixed positions of the masses m_1 and m_2 in the rotating coordinate system are $(x_1, 0)$ and $(x_2, 0)$, respectively. We will denote the distance of the test body to the mass m_i by

$$r_i = \sqrt{(x - x_i)^2 + y^2}, \quad i = 1, 2 \tag{18}$$

Then, the motion of the test body is subject to this system of second order differential equations:

$$\begin{aligned} \ddot{x} - 2\dot{y} - x &= -(1-\mu)\frac{x - x_1}{r_1^3} - \mu\frac{x - x_2}{r_2^3} \\ \ddot{y} + 2\dot{x} - y &= -(1-\mu)\frac{y - y_1}{r_1^3} - \mu\frac{y - y_2}{r_2^3} \end{aligned} \tag{19}$$

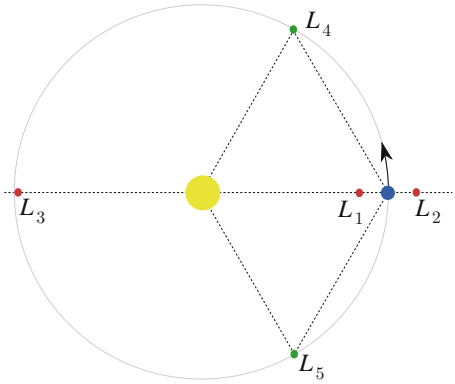


Figure 1 Lagrangian points L_1–L_5 in the restricted problem of three bodies

To ensure that the center of mass is in the origin, $x_1 = -\mu$ and $x_2 = 1 - \mu$ have to be fulfilled. The Lagrangian points L_i are the five stationary solutions of (19). L_1, L_2 and L_3 are located on the x-axis. L_4 and L_5 form equilateral triangles with the masses m_1 and m_2, as shown in Figure 1. If $\mu < \frac{1}{2} - \sqrt{\frac{23}{108}}$ it can be shown that L_4 and L_5 are stable in the linearized system.

As exemplary optimal control problem we will discuss the controlled motion of the test body.

5.2 *Optimal Control for a Spaceship*

As an illustrative example the task is to generate trajectories for a spaceship flying from a point p_0 close to L_5 to a point p_f close to L_4:

$$p_0 = (0.5, -0.866), \quad p_f = (0.5, 0.866) \tag{20}$$

The acceleration achieved by the thrusters is considered as the vector of control functions $u = (u_x, u_y)$ in the optimal control problem. Here, the control u_x is acting in the direction of x and the control u_y is acting in the direction of y.

The vector of state functions $x = (p_x, p_y, v_x, v_y)$ collects the positions in directions x and y and their velocities, which we have to introduce when writing (19) as a first order system. To generate a maneuver between two rest positions, we set

$$v_0 = (0, 0), \quad v_f = (0, 0) \tag{21}$$

In the notation of (1), let us define the trajectory planning problem as follows:

$$
\begin{aligned}
\min_{x,u,t_f} \; & t_f + \omega \int_0^{t_f} u_x^2 + u_y^2 \, dt \\
\text{s.t.} \qquad & \dot{p}_x = v_x \\
& \dot{p}_y = v_y \\
& \dot{v}_x = 2v_y + p_x - \frac{(1-\mu)(p_x - x_1)}{r_1^3} - \frac{\mu(p_x - x_2)}{r_2^3} + u_x \\
& \dot{v}_y = -2v_x + p_y - \frac{(1-\mu)p_y}{r_1^3} - \frac{\mu p_y}{r_2^3} + u_y \\
& (p_x, p_y)(0) = p_0 \\
& (p_x, p_y)(t_f) = p_f \\
& (v_x, v_y)(0) = v_0 \\
& (v_x, v_y)(t_f) = v_f
\end{aligned}
\tag{22}
$$

The objective is to minimize the process time t_f as well as the control input (or 'energy') as an integral term. For simplicity we use the weighting parameter $\omega = 1$. The mass of the smaller body is $\mu = 0.01$.

5.3 Reference Solution with TransWORHP

In Figure 2 the unperturbed solution is shown in blue for $\ell = 101$ discrete points. The task was to find a trajectory from p_0 to p_f. The spaceship can gain velocity by being attracted to the larger mass m_1, close to the origin. The resulting process time is $t_f = 2.46672$ and $E = 1.13076$ is a measure for the control input.

If the initial position is perturbed—in our example we modify the position in direction y by 0.1 and the velocity in the same direction by 0.1—the final position will not be reached if the same control is applied, as the red curve (perturbed integration) in Figure 2 shows. The slightly smaller initial distance to the masses accelerates the spaceship more than planned during optimization.

However, solving the optimal control problem from the perturbed initial position and velocity

$$
\widetilde{p}_0 = (0.5, -0.866 + 0.1), \quad \widetilde{v}_0 = (0, 0 + 0.1) \tag{23}
$$

we will fulfil the final condition again, as shown in green in Figure 2. As the distance from the perturbed initial point to the final point is shorter, and as the perturbed initial velocity points in the right direction, we get a faster process with $t_f = 2.22978$ and need less energy $E = 1.04937$. The controls for the unperturbed and perturbed solution are shown in Figure 3.

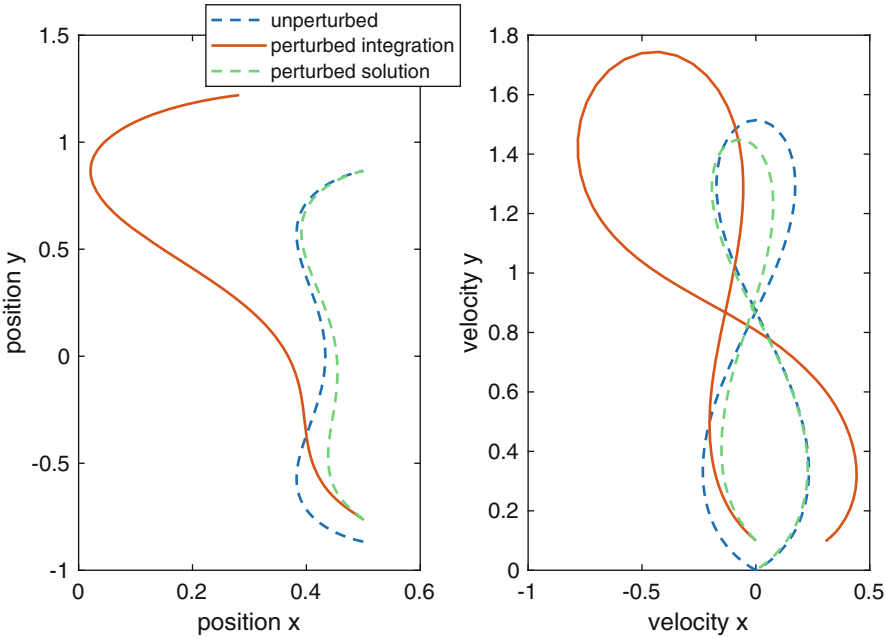


Figure 2 Position and velocity for the optimal solution from the unperturbed and the perturbed initial position

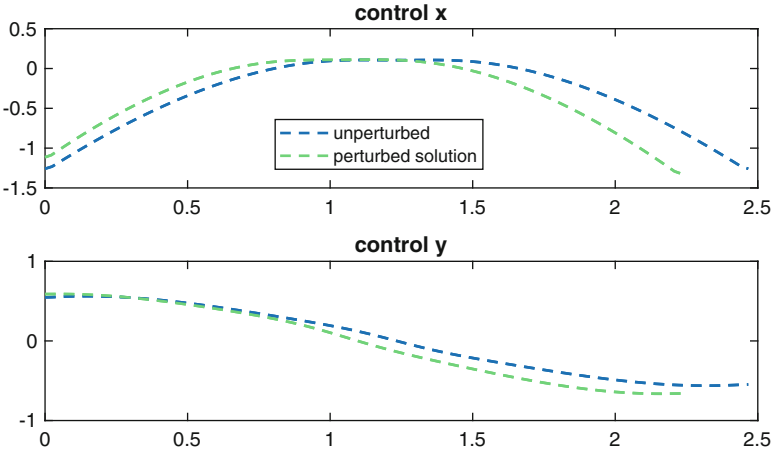


Figure 3 Controls for the optimal solution from the unperturbed and the perturbed initial position

All solutions for (22) have been computed with TransWORHP. Full discretization with the trapezoidal method requires $n = (n_x + n_u)\ell + 1 = 708$ variables in the NLP. Note that $n_x = 5$, as the integral term in the objective function, is implemented as an additional differential equation. One optimization variable is used for the free

Table 1 Dependency pattern for differential equations as used by TransWORHP

		States					Controls	
		p_x	p_y	v_x	v_y	E	u_x	u_y
Differential equations	$\dot{p}_x$			×				
	$\dot{p}_y$				×			
	$\dot{v}_x$	×	×		×		×	
	$\dot{v}_y$	×	×	×				×
	$\dot{E}$						×	×

final time. After optimization $m = n_x(\ell - 1) = 500$ equality constraints for the discretized differential equations have to be fulfilled.

The objective function only consists of two NLP variables. Hence the gradient only has two non-zero entries out of n (with the values 1 and ω). The Jacobian of the constraints could store up to $n \cdot m = 354{,}000$ entries. However, due to the discretization in (5) each constraint only depends on two adjacent points. Table 1 shows that each differential equation only depends on few states and controls. This pattern can be used to provide a Jacobian with only 3900 non-zero entries. We use finite differences to compute them (although it would also be possible analytically). The Hessian also takes advantage of the fact that constraints only depend on two adjacent points. Using finite differences again, we only have to compute 3536 out of $\frac{n(n+1)}{2} = 250{,}986$ entries.

We used a reasonable initial guess for the process time $t_f = 2.5$ and for the controls

$$u_x = 0.5 - 0.38t, \quad u_y = -0.6(t - 1.3)^2$$

and provided an initial guess for the states by integration. An optimal solution was found using TransWORHP in 10 NLP iterations with a precision of 10^{-6} for feasibility and optimality. The high level of sparsity allowed the computation in approximately 5 s on a standard pc using standard WORHP settings.

5.4 Real-Time Solution Using WORHP Zen

Instead of solving the optimal control problem again in the case of a perturbation, we will now apply the algorithms of Sect. 4 to approximate the optimal solution in less time.

In the method described in Sect. 4.1 the pre-correction step considering the measured perturbation in the initial position brings the spaceship closer to the final position, but still does not reach it. The final position in direction y is almost reached, whereas in direction x there is a big gap, see the yellow curves in Figure 4. After some post-correction steps, however, the desired final position is reached precisely (purple curves). In Table 2 the iterative behaviour of the objective function

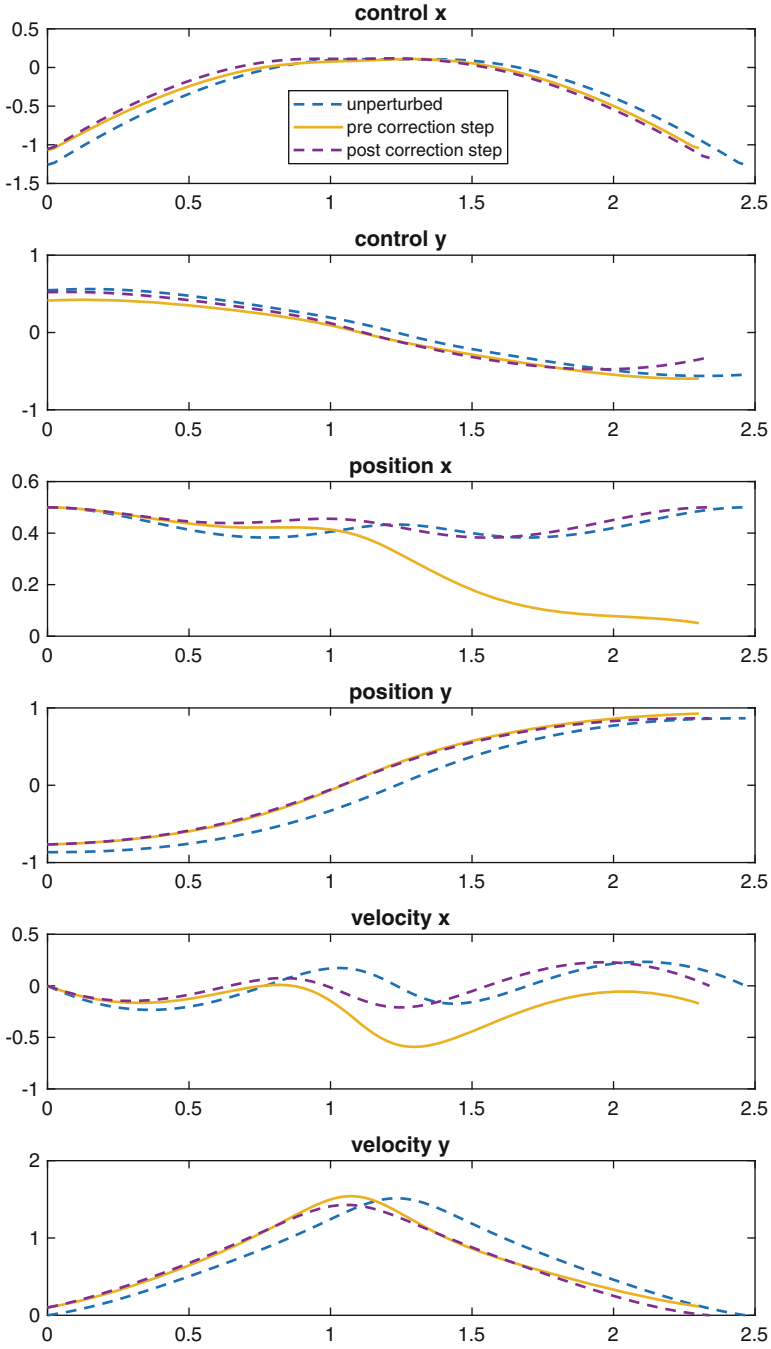


Figure 4 Real-time optimal control for the spaceship using sensitivity derivatives of boundary condition

Table 2 Process time t_f, energy term and final state deviation for real-time optimal control for the spaceship using sensitivity derivatives of boundary condition

Iteration	t_f	E	$\|\Delta q^{[i]}\|_\infty$
0	2.46672	1.13076	0.5256863
1	2.30186	0.77673	0.4980371
2	2.34467	1.19887	0.1383581
3	2.33899	0.83768	0.0231891
4	2.33976	0.84755	0.0030709
5	2.33917	0.84558	0.0015629
6	2.33919	0.84594	0.0001827
7	2.33917	0.84585	0.0000756
8	2.33917	0.84586	0.0000132
9	2.33917	0.84585	0.0000041
10	2.33916	0.84585	0.0000009

and the constraints during the correction steps is shown. The free process time t_f is also adjusted during the iteration. Starting with an initial velocity towards our target results in a shorter process time. Iteration 0 is the solution computed for the perturbed initial position. In iteration 1 the pre-correction step was applied, and the following iterations refer to the post-correction steps.

On the other side, the pre-correction step of the method described in Sect. 4.2 already reaches the final position and velocity exactly as shown in Figure 5. However, the updated controls and states do not fulfil the system of discretized differential equations anymore, as shown in Figure 6. This gap can be closed during the post-correction steps. In Figure 6 after the fourth post-correction step the deviation can't be distinguished from zero anymore.

The process time is also adjusted for this method. Comparing the results for both methods in Tables 2 and 3 we get a smaller process time for the second method, at a higher energy cost.

5.5 Comparison with Convergence Theorem

In Theorem 1 an exponential convergence rate for the constraints (13) was shown. This can be observed in the last column of Table 2 and in Figure 7 for the method described in Sect. 4.1. In every iteration, the measured deviation in the final point is reduced by the same magnitude.

Analogously, Table 3 and Figure 8 visualize the convergence rate for the method described in Sect. 4.2. A remarkable fact when using the post-correction steps: solutions can be found which hold the system of differential equations with higher accuracy than the original solution, where only 10^{-6} was demanded.

Furthermore, Theorem 1 states in (13) that the objective function maintains optimality depending on the deviation $\|p - p_0\|^3$. To show this numerically, we use

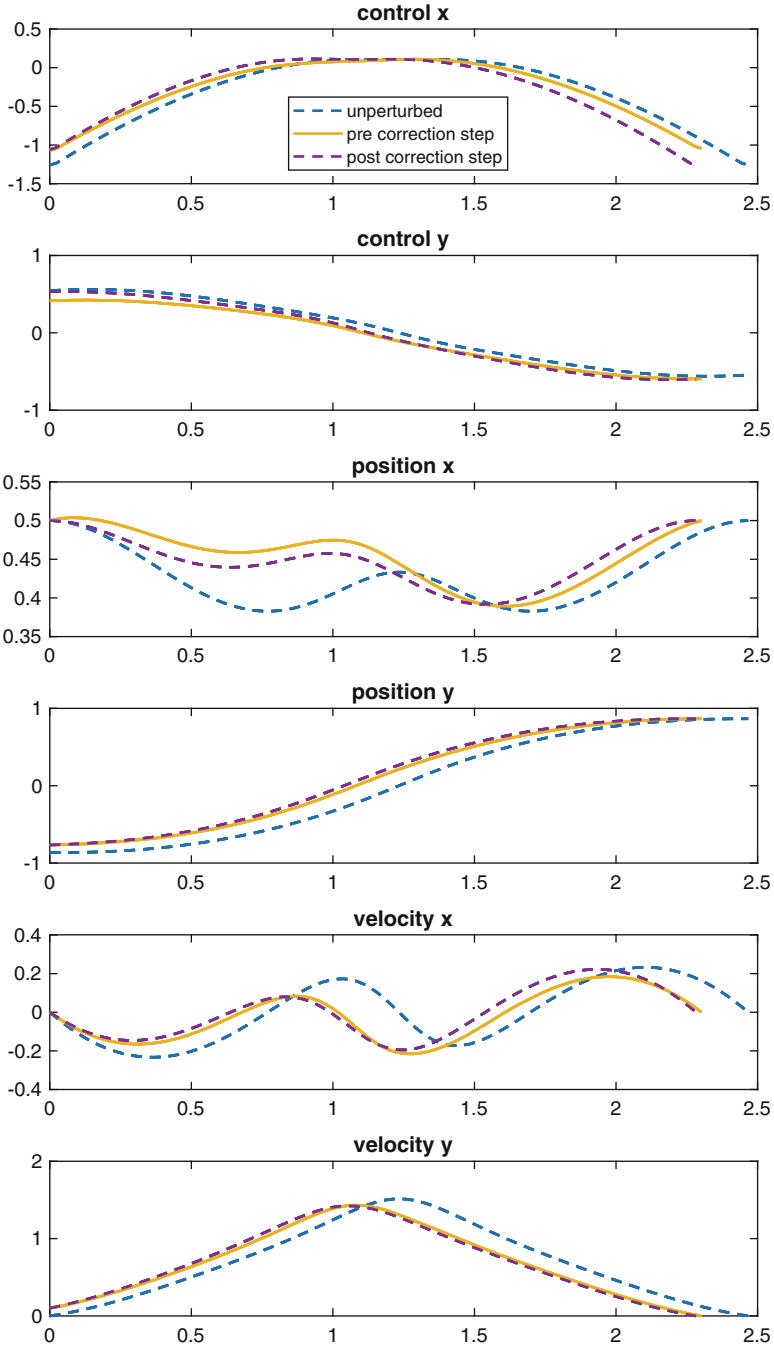


Figure 5 Real-time optimal control for the spaceship using sensitivity derivatives of discretized differential equation

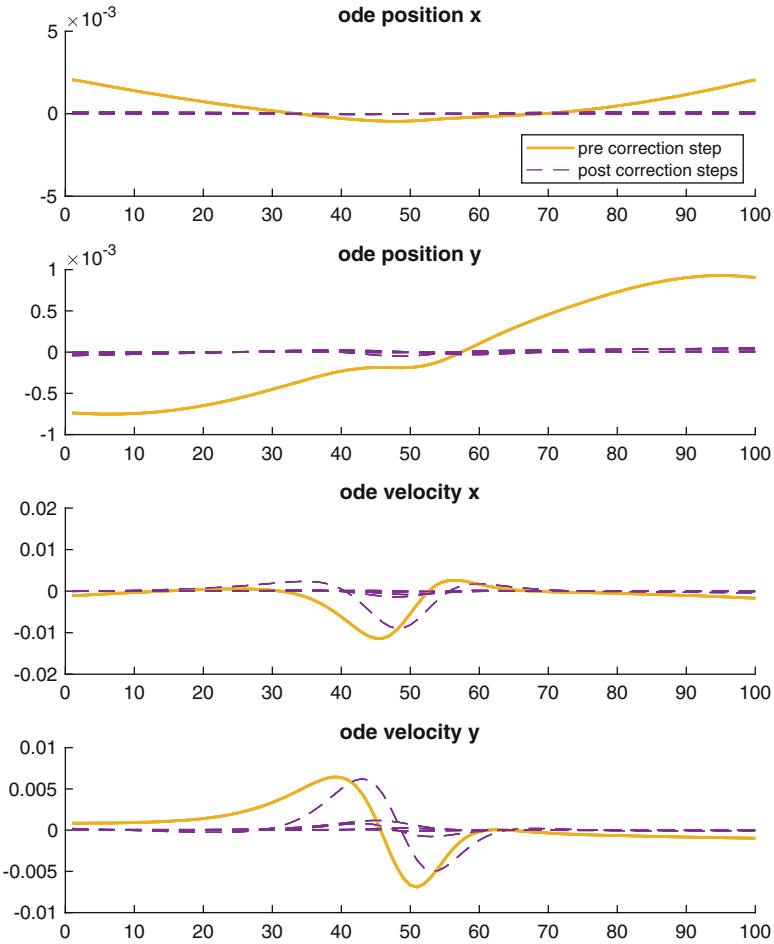


Figure 6 Deviation $\Delta q^{[i],j}$ in discretized differential equations for $j = 1, \ldots, \ell - 1$ ($\ell = 101$) during correction steps

the same unperturbed solution as in Sect. 5.3 together with the sensitivity derivatives and consider perturbations $\Delta y_0 \in [-0.4, 0.4]$ in the initial position of p_y.

For the method described in Sect. 4.1, Figure 9 shows that the feasible solution after the post-correction steps is a good approximation of the perturbed optimal solution for a large range of Δy_0. For the objective function (sum of process time and energy) a cubic behaviour was expected. However, the individual components of the objective function act differently.

As expected, smaller perturbations require less iterations. The number of iterations is reflected also directly in the computational time of 0.5–2 s using MATLAB. Obviously, the trapezoidal method takes the largest part of the high computational cost.

Table 3 Process time t_f, energy term and deviations in discretized differential equations for real-time optimal control for the spaceship using sensitivity derivatives of discretized differential equation

Iteration	t_f	E	$\|\Delta q^{[i]}\|_\infty$
0	2.46672	1.13076	0.1414214
1	2.30186	0.77673	0.0113876
2	2.29206	0.90077	0.0089981
3	2.28513	0.92540	0.0011791
4	2.28319	0.93618	0.0013948
5	2.28328	0.93942	0.0002601
6	2.28275	0.94005	0.0001871
7	2.28292	0.94058	0.0000506
8	2.28280	0.94056	0.0000245
9	2.28284	0.94065	0.0000086
10	2.28282	0.94063	0.0000033
11	2.28282	0.94065	0.0000015
12	2.28282	0.94065	0.0000006

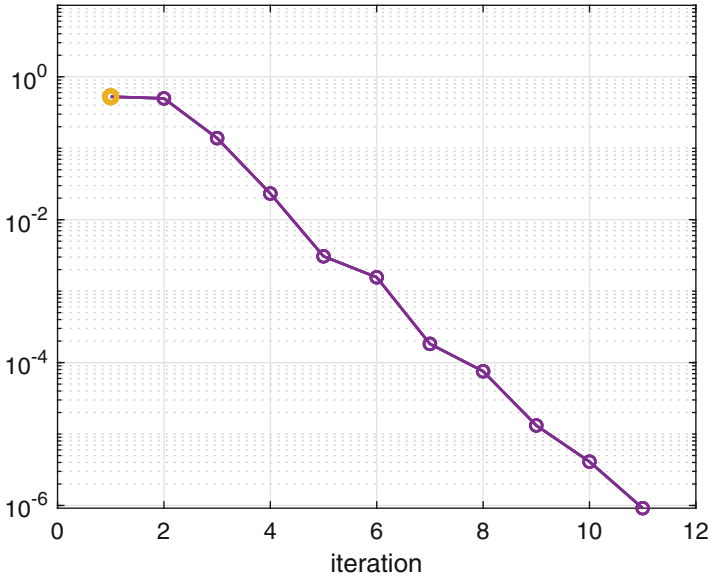


Figure 7 Convergence of $\|\Delta q^{[i]}\|_\infty$ in final point during correction steps

Analogously Figure 10 shows the results for the method described in Sect. 4.2. The post-correction steps yield better approximations for the objective function than for the first method. Even if the number of iterations is higher (for larger deviations) than for the first method, the computational time of under 0.06 s (using MATLAB again) is remarkably smaller as the trapezoidal method is not called any more.

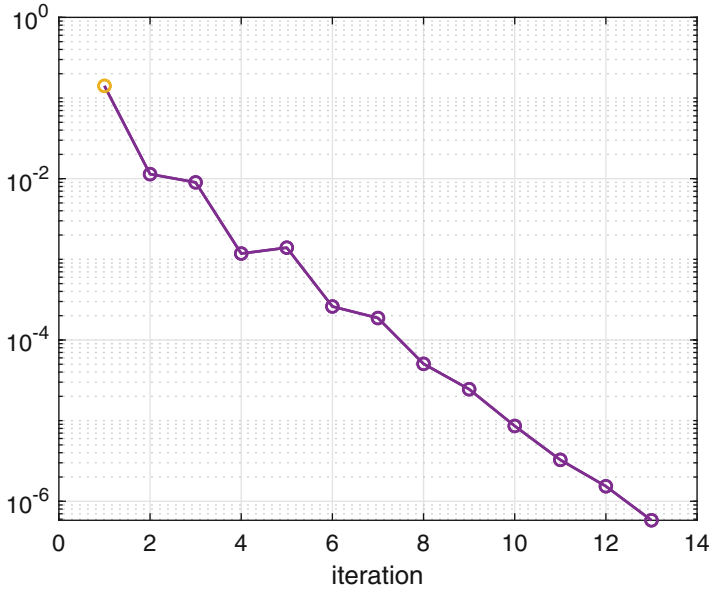


Figure 8 Convergence $\|\Delta q^{[i]}\|_\infty$ in discretized differential equations during correction steps

6 Conclusion

Sensitivity derivatives can be used to generate sub-optimal solutions of perturbed optimal control problems. Constraints, which have been active in the solution of the unperturbed problem, can be iteratively fulfilled. The optimality of the sub-optimal solution depends on the amount of perturbation.

We presented two methods to ensure feasibility in the post-correction steps: First, by using sensitivity derivatives of the boundary condition only small matrices of sensitivity derivatives were needed. However, the states have to be integrated for updated controls to evaluate the deviation. For implicit integration methods as the trapezoidal method, this would require costful operations. Second, by using sensitivity derivatives of discretized differential equations larger matrices of sensitivity derivatives were needed. Apart from function evaluations of f this method only uses matrix–vector multiplications to update the states and controls.

If only selected states have to be transferred to a motor controller, the pre-correction step of the second method already provides an acceptable solution.

As active constraints on the controls would stay active after the correction steps, optimality can't be guaranteed any more. On the other hand, the presented algorithms would not handle it, if inactive constraints on the controls would get active. However, by trimming the controls to the feasible region, the algorithm can be adjusted.

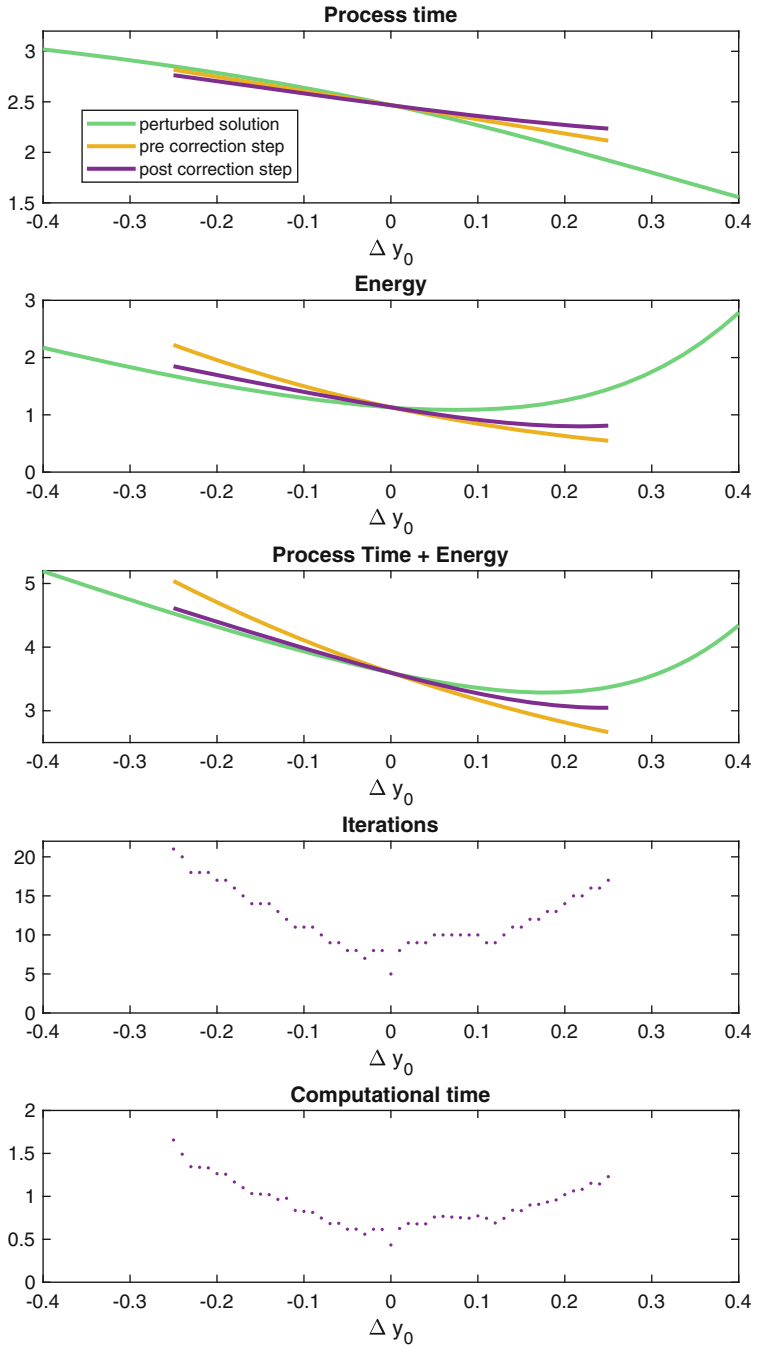


Figure 9 Approximation of the objective function for the perturbed trajectory using sensitivity derivatives of boundary condition

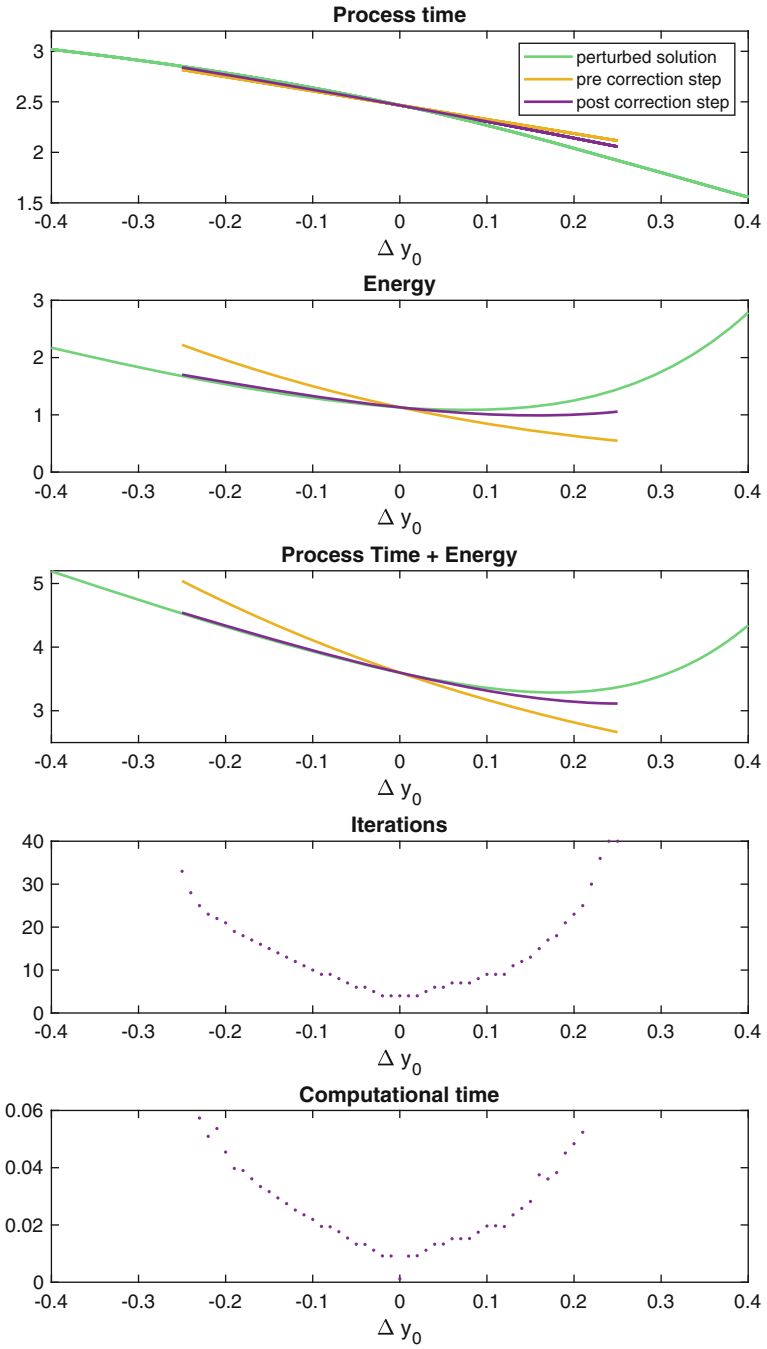


Figure 10 Approximation of the objective function for the perturbed trajectory using sensitivity derivatives of discretized differential equation

The hidden potential of sensitivity derivatives is still large. To improve handling of state perturbations at any time during a manoeuver in the pre-correction step, [12] applied similar techniques to efficient on-board optimization for a Mars entry. However, large data sets for sensitivity derivatives are needed in any of these methods.

References

1. Battin, R.H.: An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition. AIAA Education Series (1999)
2. Betts, J.T.: Practical Methods for Optimal Control Using Nonlinear Programming. SIAM, Philadelphia (2001)
3. Büskens, C.: Optimierungsmethoden und Sensitivitätsanalyse für optimale Steuerprozesse mit Steuer- und Zustandsbeschränkungen. Ph.D. thesis, Universität Münster, Institut für Numerische Mathematik, Münster (1998)
4. Büskens, C.: Real-time optimization and real-time optimal control of parameter-perturbed problems. Habilitation, Universität Bayreuth (2002)
5. Büskens, C., Maurer, H.: SQP-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control. J. Comput. Appl. Math. **120**, 85–108 (2000)
6. Büskens, C., Wassel, D.: The ESA NLP solver WORHP. In: Modeling and Optimization in Space Engineering, pp. 85–110. Springer, New York (2013)
7. Fiacco, A.V.: Introduction to Sensitivity and Stability Analysis in Nonlinear Programming. Mathematics in Science and Engineering, vol. 165. Academic Press, New York (1983)
8. Grüne, L., Pannek, J.: Nonlinear Model Predictive Control: Theory and Applications. Springer, London (2011)
9. Knauer, M., Büskens, C.: From WORHP to TransWORHP. In: Proceedings of the 5th International Conference on Astrodynamics Tools and Techniques (2012)
10. Knauer, M., Büskens, C.: Processing user input in tracking problems using model predictive control. IFAC-PapersOnLine **50**(1), 9846–9851 (2017)
11. Roenneke, A.J., Cornwell, P.J.: Trajectory control for a low-lift re-entry vehicle. J. Guid. Control. Dyn. **16**(5), 927–933 (1993)
12. Seelbinder, D.: On-board trajectory computation for mars atmospheric entry based on parametric sensitivity analysis of optimal control problems. Ph.D. thesis, Universität Bremen (2017)
13. Topputo, F., Zhang, C.: Survey of direct transcription for low-thrust space trajectory optimization with applications. Abstr. Appl. Anal. **2014**, 15 pp. (2014). Article ID 851720

Theory and Applications of Optimal Finite Thrust Orbital Transfers

L. Mazzini and M. Cerreto

Abstract The main author proposed a mission for the first time with a LEO to GEO orbital transfer for telecommunication application (GovSatCom). This mission will allow the use of small launchers to bring, at a lower cost, satellites of medium and large size at Geostationary orbit. This motivated the authors to develop a mathematical model in order to find the optimal thrust strategy for very long orbital transfers of satellites with electric thrusters. During the transfer, the satellite is supposed capable to steer the thrust vector in any direction. To solve the optimization problem, an averaging technique has been adopted. The authors discussed and solved this problem including the J_2 and eclipse effects. Moreover some external constraints are included in the problem in order to avoid simulations with unrealistic orbital transfers (i.e., too low perigee altitude). Referring to the papers already published by the authors, this last one is a synthetic review of the theory and the applications. After a mathematical introduction of the theoretical notions, new numerical results are presented.

Nomenclature

0	Subscript of the initial orbit
f	Subscript of the final orbit
e	Eccentricity
a	Semi-major axis (km)
g	Perigee anomaly from the ascending node
I	Inclination
h	Right ascension of the ascending node (RAAN)
u	Anomaly from the descending node
v	Anomaly from the perigee

L. Mazzini (✉) · M. Cerreto (✉)
Thales Alenia Space, Rome, Italy
e-mail: leonardo.mazzini@thalesaleniaspace.com; marco.cerreto@thalesaleniaspace.com

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_10

r	Satellite range, dimensional (km) or non-dimensional $r = r/a_0$
p	Satellite range rate (km/s)
G	$\sqrt{\mu\alpha(1-e^2)}$, momentum (km^2/s) or non-dimensional $G = \sqrt{\alpha(1-e^2)/\alpha_0}$
w	Orbital rate integrated in time defined by $\dot{w} = G/r^2 = \sqrt{\mu\alpha(1-e^2)/r^2}$
H	Angular momentum normal to the equatorial plane (km^2/s) or non-dimensional $H = H/\sqrt{\mu\alpha_0}$
Y_1	l/r
Y_2	w derivative of Y_1
μ	Gravitational constant (km^3/s^2)
A	3-vector, acceleration vector (km/s^2)
X	6-vector, non-dimensional orbit state
L	6-vector, non-dimensional orbital costate
X	8-vector, non-dimensional orbital state
L	8-vector, non-dimensional orbital costate
l	3-vector, firing direction cosines in orbital frame
B	6,3-matrix, maneuver matrix
M	Satellite mass (kg)
m	Mass flow rate (kg/s)
V_e	Gas exhaust velocity (km/s)
f	Non-dimensional control force
V	Non-dimensional characteristic velocity
GEO	Geostationary Earth Orbit
MEO	Medium Earth Orbit
LEO	Low Earth Orbit
GTO	Geostationary Transfer Orbit
RAAN	Right Ascension of the Ascending Node

1 Introduction

The use of electric propulsion systems for satellite orbital transfers is a fundamental topic in space applications. For a given mission and payload application, the electric thrusters, due to the high value of the specific impulse, significantly reduce the satellite launch mass with respect to chemical propulsion system. However, the electric propulsion system provides a very low thrust, thus the orbital transfers can last many months: for this reason an optimal thrust strategy computation is necessary in order to minimize the time of the transfer or the propellant.

The optimal control theory for orbital transfers involves two main objectives: minimize the transfer time (defined by the authors "T_{min} problem") or minimize the propellant consumption (defined by the authors "V_{min} problem"). In particular, considering a T_{min} problem, the thrusters are always switched on during the orbital

transfer, except when the satellite is in eclipse phase. In the V_{min} problem the optimal thrust strategy calculates the firing arcs where it is convenient to switch on the thrusters in order to save propellant mass.

The optimization problem can be treated with two different techniques (see also [3]). The direct method involves controlling the discrete force vector in many nodes and, for very long orbital transfers, this results in a great number of control variables in the numeric algorithm. This method is often simplified using part of the features of the extremal solutions derived from the Maximum Principle (see [7] and, for a modern introduction, [2]).

The indirect approach looks for the solution using the first order necessary conditions of extremality from the Maximum Principle, which results in a two boundary value problem. The problem does not result in searching for the minimum of a function directly, but consists in solving a two boundary value problem "shooting" the correct boundary conditions with an algorithm, which is not globally convergent in such a highly nonlinear context (see modern papers like [6, 7] and [2]).

In this paper the indirect method is used with simplified solutions for initializing the "shooting" algorithm. The usual approach is to find approximate solutions implementing a control parametrization, then averaging the dynamical equations and finally using the Maximum Principle to optimize the parametrized control.

The authors apply the averaging to the "exact" Hamiltonian and use the averaging theorem (see Sect. 2.3) that state the ε closeness with the exact optimal solution in a ε^{-1} non-dimensional timescale, where ε is the non-dimensional control force maximum value. The averaging techniques can provide fast and converging computing methods for long orbital transfers (see also [7]).

The mathematical model developed for the finite thrust optimization takes into account the perturbations as J_2 and the eclipses. In particular, during the eclipse phases the thrusters must be switched off thus a constraint is necessary in the optimization process. The conditions produced by the Earth's shadow can be represented mathematically as the effect of a non-smooth mixed state control constraint (see also [4]). This constraint produces a discontinuity in the costate through an eclipse function (E) which is negative when the satellite is in eclipse and positive if it is in sunlight.

Thousands of cases representing all possible scenarios for applications have been studied by the authors, some of them are reported in the numerical section focusing on the perturbations and eclipse effects.

Furthermore, the study of the relationship between the mission ΔV cost and the final right ascending node is reported. The use of low-thrust optimization allows to deliver in a single launch multiple satellites in different orbital planes (constellation deployment), this requires the computation of the evolution of the ΔV as a function of the final right ascending node. As presented in [11] it is possible to obtain different extremal solutions starting from the same initial conditions, one with a higher ΔV than the other. The introduction of the J_2 in this analysis increases the problem complexity and requires the introduction of the perigee constraint to

avoid unrealistic collision with the Earth. Two different orbital transfers with a strong change of RAAN are reported in the numerical section in order to discuss the obtained results.

Finally, in the numerical section a V_{min} problem is reported in order to show how the mission ΔV cost decreases, increasing the available transfer time.

2 Mathematical Model

The mathematical model adopted uses a representation in a non-dimensional state, which greatly simplifies the usual equations. It is possible to obtain the invariant non-dimensional vector that defines the state of the orbit, always starting from the Keplerian elements (see [9] and [8]). Following [8] the authors use coordinates derived from the Hill parameters. In this way they permit to set the system in a suitable form for direct averaging, without ε series.

For what concern the derivation of the controlled dynamics of the state, it is necessary to introduce the equations of the satellite dynamics under the effect of external accelerations. Define the vector A that contains the radial, bi-normal, and normal components A_r, A_t, A_n of the acceleration:

$$A_\alpha = l_\alpha \frac{|\dot{m}|V_e}{M}; \qquad \frac{dM}{dt} = -|\dot{m}|; \qquad \alpha = r, t, n \tag{1}$$

where $\dot{m}$ is the propellant mass flow rate, M is the satellite mass, V_e is the exhaust velocity, and l is the unitary vector of the local flight triad.

The Hill variables were introduced into artificial satellite theory by Izsak, and they are effective in describing satellite short-term perturbations. In these variables the maneuver equations are

$$\begin{aligned}
\dot{r} &= p \\
\dot{p} &= \frac{G^2}{r^3} - \frac{\mu}{r^2} + A_r \\
\dot{G} &= rA_t \\
\dot{u} &= \frac{G}{r^2} - r\cos(I)\,\frac{\sin(u)A_n}{G\,\sin(I)} \\
\dot{H} &= r\,\cos(I)A_t - r\,\sin(I)\,\cos(u)A_n \\
\dot{h} &= r\,\frac{\sin(u)}{G\,\sin(I)}A_n
\end{aligned} \tag{2}$$

At this point a new independent variable is introduced, the anomaly w that is defined by:

$$\dot{w} = \frac{G}{r^2} \Rightarrow dt = r^2 \frac{dw}{G} \tag{3}$$

The vector A multiplied by $\frac{dt}{dw}$ provides the velocity variation per variation of anomaly w:

$$A_\alpha^w = l_\alpha \frac{\dot{m} V_e r^2}{MG} \tag{4}$$

New variables Y_1 and Y_2 are introduced in place of r and p to obtain an expression for the in-plane free dynamics similar to that of an oscillator. Such a feature will later permit to easily introduce as states the invariant amplitudes of the oscillator. Letting $Y_1 = \frac{1}{r}$, $\frac{dY_1}{dw} = Y_2$ and differentiating these two equations with respect to w, it follows

$$\frac{dY_2}{dw} = -Y_1 + \frac{\mu}{G^2} + \frac{Y_2}{GY_1} A_t^w - \frac{A_r^w}{G} \tag{5}$$

Then, all the variables are made non-dimensional using a_0, the semi-major axis of the final orbit, by letting $Y_1^{ad.} = Y_1 a_0$, $Y_2^{ad.} = Y_2 a_0$, $A_\alpha^{ad.} = \frac{A_\alpha^w}{\sqrt{\frac{\mu}{a_0}}}$, $G^{ad.} = \frac{G}{\sqrt{\mu a_0}}$, and $H^{ad.} = \frac{H}{\sqrt{\mu a_0}}$. The resulting equations in non-dimensional variables are formally identical to the dimensional ones with $\mu = 1$. The non-dimensional vector A is

$$A_\alpha^{ad.} = l_\alpha \frac{\dot{m} V_e r^2}{MG\sqrt{\frac{\mu}{a_0}}} = l_\alpha e^{\frac{V}{V_e}} f \frac{\sqrt{\frac{a_0}{a(1-e^2)}}}{Y_1^2} \tag{6}$$

where

- f is the positive non-dimensional control thrust $f = \frac{\dot{m} V_e}{M_0 \frac{\mu}{a_0^2}}$, limited by a small parameter ε: $f \leq \varepsilon$
- M_0 is the initial launch mass;
- V is the characteristic velocity;
- V_e is the gas exhaust velocity;
- Y_1 is the non-dimensional satellite inverse range.

The multiplicative term in the expression of the acceleration is $\frac{\sqrt{\frac{a_0}{a(1-e^2)}}}{Y_1^2} = \frac{d\tau}{dw}$, where $\tau = t\sqrt{\frac{\mu}{a_0^3}}$ is the non-dimensional time (from now on, the non-dimensional variables will be always used without the superscript $ad.$).

A new non-dimensional invariant state vector X, based on the non-dimensional Hill parameters and the Keplerian elements, can now be introduced as:

$$\begin{aligned} X_1 &= (Y_1 - X_3)\ \cos(u) - Y_2 \sin(u) = eX_3 \cos(g) \\ X_2 &= (Y_1 - X_3)\ \sin(u) + Y_2 \cos(u) = eX_3 \sin(g) \\ X_3 &= \frac{1}{G^2} = \frac{a_0}{a(1-e^2)} \\ X_4 &= \frac{H}{G} = \cos(I) \\ X_5 &= h \\ X_6 &= u - w = g + \nu - w \end{aligned} \tag{7}$$

where a is the semi-major axis, a_0 is the reference semi-major axis (typically the semi-major axis of the final orbit), e is the eccentricity, I is the inclination, g is perigee anomaly from the ascending node, ν is the anomaly from the perigee, w is anomaly rate integrated in time from t_0 defined by $\dot{w} = \frac{\sqrt{\mu a(1-e^2)}}{r^2}$. These parameters have been used in references [9, 10] to develop the full theory of the optimal transfer, and they offer quite simple maneuver equations although affected by singularities for orbit at null eccentricity and inclination.

In order to avoid the abovementioned singularities, the theory has been evolved using a new coordinate system (see [11]). Calling h the right ascension, the new non-dimensional invariant state vector $\mathbf{X}$ is defined as:

$$\begin{aligned} \mathbf{X}_1 &= e\frac{a_0}{a\ (1-e^2)} \cos(g+h) = e\mathbf{X}_3 \cos(g+h) \\ \mathbf{X}_2 &= e\frac{a_0}{a\ (1-e^2)} \sin(g+h) = e\mathbf{X}_3 \sin(g+h) \\ \mathbf{X}_3 &= \frac{a_0}{a\ (1-e^2)} \\ \mathbf{X}_4 &= \tan\left(\frac{I}{2}\right) \cos(h) \\ \mathbf{X}_5 &= \tan\left(\frac{I}{2}\right) \sin(h) \\ \mathbf{X}_6 &= h + g + \nu - w \end{aligned} \tag{8}$$

The initial conditions can be set such that $g_0 = 0$, $w_0 = 0$, $u_0 = 0$, $I_0 = 90°$ and the initial orbits are parametrized only in the eccentricity e_0, so that:

$$\begin{aligned} \mathbf{X}_{01} &= \frac{e_0}{1 - e_0^2} \\ \mathbf{X}_{02} &= 0 \\ \mathbf{X}_{03} &= \frac{1}{1 - e_0^2} \\ \mathbf{X}_{04} &= 0 \\ \mathbf{X}_{05} &= 0 \\ \mathbf{X}_{06} &= 0 \end{aligned} \tag{9}$$

all the possible elliptic final orbits are defined by a choice of the first five state variables such that:

$$\sqrt{\mathbf{X}_{f1}^2 + \mathbf{X}_{f2}^2} < \mathbf{X}_{f3}$$

Each state variable is constant if the external accelerations are null. The state variables $\mathbf{X}_1$ and $\mathbf{X}_2$ can be considered as amplitudes of the non-dimensional inverse range,

$$Y_1 = \mathbf{X}_1 \cos(u) + \mathbf{X}_2 \sin(u) + \mathbf{X}_3 \qquad Y_2 = -\mathbf{X}_1 \sin(u) + \mathbf{X}_2 \cos(u)$$

At this point, the maneuver equations in the independent variable w are defined as:

$$\frac{d\mathbf{X}}{dw} = B(\mathbf{X}, u = w + \mathbf{X}_6)A \tag{10}$$

where $B(\mathbf{X}, u = w + \mathbf{X}_6)$ is a 6×3 matrix that functionally depends on the first four entries of $\mathbf{X}$: $\mathbf{X}_1$, $\mathbf{X}_2$, $\mathbf{X}_3$, and $\mathbf{X}_4$.

The independence on the fifth entry is due to symmetry reasons while the dependence on the sixth entry takes place through the specific 2π-periodic argument: $u = w + \mathbf{X}_6$. Using the new definition of the state vector, the elements of the matrix $\tilde{B} = Y_1 B$ are

$$\tilde{B}_{11} = (\mathbf{X}_3 + \mathbf{X}_1 \cos(u) + \mathbf{X}_2 \sin(u)) \sin(u) \sqrt{\mathbf{X}_3}$$

$$\tilde{B}_{12} = \left(\frac{\mathbf{X}_2}{2} \sin(2u) - \mathbf{X}_1 \sin^2(2u) + 2\mathbf{X}_3 \cos(u) \right) \sqrt{\mathbf{X}_3}$$

$$\tilde{B}_{13} = (\mathbf{X}_5 \cos(u) - \mathbf{X}_4 \sin(u)) \mathbf{X}_2 \sqrt{\mathbf{X}_3}$$

$$
\begin{aligned}
\tilde{B}_{21} &= -\cos(u)\sqrt{\mathbf{X}_3}(\mathbf{X}_3 + \mathbf{X}_1\cos(u) + \mathbf{X}_2\sin(u)) \\
\tilde{B}_{22} &= \sqrt{\mathbf{X}_3}(-\mathbf{X}_2\cos^2(u) + 2\mathbf{X}_3\sin(u) + \mathbf{X}_1\sin(u)\cos(u)) \\
\tilde{B}_{23} &= (-\mathbf{X}_5\cos(u) + \mathbf{X}_4\sin(u))\mathbf{X}_1\sqrt{\mathbf{X}_3} \\
\tilde{B}_{32} &= -2\sqrt{\mathbf{X}_3^3} \\
\tilde{B}_{43} &= \frac{1}{2}\cos(u)(\mathbf{X}_4^2 + \mathbf{X}_5^2 + 1)\sqrt{\mathbf{X}_3} \\
\tilde{B}_{53} &= \frac{1}{2}\sin(u)(\mathbf{X}_4^2 + \mathbf{X}_5^2 + 1)\sqrt{\mathbf{X}_3} \\
\tilde{B}_{63} &= (-\mathbf{X}_5\cos(u) + \mathbf{X}_4\sin(u))\sqrt{\mathbf{X}_3} \\
\tilde{B}_{31} &= \tilde{B}_{33} = \tilde{B}_{41} = \tilde{B}_{42} = \tilde{B}_{51} = \tilde{B}_{52} = \tilde{B}_{61} = \tilde{B}_{62} = 0
\end{aligned}
\tag{11}
$$

In the orbital control problems, the quantity to be minimized is the non-dimensional characteristic velocity:

$$
V = \int_{w_0}^{w_f} |A| \, dw \tag{12}
$$

or the total transfer time $T = \int_{w_0}^{w_f} \frac{d\tau}{dw}\, dw$.

The dynamic system that will be used for the orbital transfer optimization includes Equation (10) and the time, because it is necessary to represent the Sun's motion. In order to obtain the perturbation structure of the system in the parameter ε, time will be represented in the state with the variable $T = \varepsilon\tau$, ε being the maximum possible value for the non-dimensional control force f. The state $(\mathbf{X}, V, T)$ evolves as:

$$
\begin{aligned}
\frac{d\mathbf{X}}{dw} &= f\, B(\mathbf{X}, w + \mathbf{X}_6)\, \underline{l}\, e^{\frac{V}{Ve}}\, \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} \\
\frac{dV}{dw} &= f\, e^{\frac{V}{Ve}}\, \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} \\
\frac{dT}{dw} &= \varepsilon\, \frac{\sqrt{\mathbf{X}_3}}{Y_1^2}
\end{aligned}
\tag{13}
$$

where the controls are f and $\underline{l}$.

The non-dimensional control force is limited by the following inequality: $0 < f \leq \varepsilon$. With this setting it is introduced the pre-Hamiltonian $\tilde{H}$ and the costate $\mathbf{L}$ ($\mathbf{L} \in R^6, L_V, L_T$), following the standard Maximum Principle method [2], where L_V and L_T are, respectively, the costate components referred to the velocity and the time.

$$\tilde{H} = f(\mathbf{L}^T B(\mathbf{X}, u)\underline{l} + L_V)e^{\frac{V}{V_e}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2} + \varepsilon L_T \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} \tag{14}$$

The Hamiltonian is obtained by maximizing $\tilde{H}$ in the controls $\underline{l}$, f:

$$\begin{aligned} H &= \sup_{f \in [0,\, \varepsilon],\, \underline{l} \in |\underline{l}| = 1} \tilde{H} \\ H &= \varepsilon\kappa(S)S(\mathbf{X}, \mathbf{L}, L_V, u)e^{\frac{V}{V_e}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2} + \varepsilon L_t \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} \end{aligned} \tag{15}$$

where the functions S and $\kappa(S)$ have been defined as:

$$\begin{aligned} S(\mathbf{X}, \mathbf{L}, L_V, u) &= \sqrt{\mathbf{L}^T B(\mathbf{X}, u)B(\mathbf{X}, u)^T \mathbf{L}} + L_V \\ \begin{cases} \kappa(S) &= 1, \qquad \forall S > 0 \\ \kappa(S) &= 0, \qquad \forall S < 0 \end{cases} \end{aligned} \tag{16}$$

The control action, as derived by maximizing the pre-Hamiltonian, is

$$\begin{aligned} \underline{l} &= \frac{B(\mathbf{X}, u)^T \mathbf{L}}{\sqrt{\mathbf{L}^T B(\mathbf{X}, u)B(\mathbf{X}, u)^T \mathbf{L}}} \\ f &= \varepsilon\, \kappa(S) \end{aligned} \tag{17}$$

The control action f is not defined when $S(\mathbf{X}, \mathbf{L}, L_V, u) = 0$ but this happens in a single isolated point, so the state and costate curves are unique, unless there is a specific and non-generic case of a singular arc which can be studied using higher order derivatives. When the state and costate curves cross the surface $S(\mathbf{X}, \mathbf{L}, L_v, u) = 0$ there is an on/off switching of the thrusters, therefore S is called the *Switching function*.

The necessary conditions that any extremal satisfies are expressed by the following system of differential equations derived from the Hamiltonian:

$$
\begin{aligned}
\frac{d\mathbf{X}}{dw} &= \varepsilon\kappa(S)\frac{B(\mathbf{X},u)B(\mathbf{X},u)^T\mathbf{L}}{\sqrt{\mathbf{L}^T B(\mathbf{X},u)B(\mathbf{X},u)^T\mathbf{L}}}e^{\frac{V}{V_e}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}\\
\frac{dV}{dw} &= \varepsilon\kappa(S)e^{\frac{V}{V_e}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}\\
\frac{dT}{dw} &= \varepsilon\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}\\
\frac{d\mathbf{L}}{dw} &= -\varepsilon\kappa(S)\frac{\partial\sqrt{\mathbf{L}^T B(\mathbf{X},u)B(\mathbf{X},u)^T\mathbf{L}}}{\partial\mathbf{X}}e^{\frac{V}{V_e}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2} - \varepsilon\left(\kappa(S)Se^{\frac{V}{V_e}} + L_T\right)\frac{\partial\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}}{\partial\mathbf{X}}\\
\frac{dL_V}{dw} &= -\varepsilon\kappa(S)S\frac{e^{\frac{V}{V_e}}}{V_e}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}\\
\frac{dL_T}{dw} &= 0
\end{aligned}
\tag{18}
$$

2.1 Boundary Conditions

Concerning the orbital transfer optimization, the boundary conditions are reported in this section. In particular the presented system of 16 differential equations requires 16 boundary conditions, plus the final w free that is another condition, for a total of 17 boundary conditions.

Typically the initial and final state $\mathbf{X}_0$ and $\mathbf{X}_f$ are known:

$$
\begin{aligned}
&\mathbf{X}_k(0) = \mathbf{X}_{0k} \qquad for\ k = 1, .., 6\\
&\mathbf{X}_k(w_f) = \mathbf{X}_{fk} \qquad for\ k = 1, .., 5
\end{aligned}
\tag{19}
$$

The time and cost at starting anomaly are null:

$$
V(0) = T(0) = 0 \tag{20}
$$

There are two transversality conditions: the first derived from the fact that $\mathbf{X}_6(w_f)$ is not specified because the authors are not interested in the final anomaly, the second derived by the fact that w_f is free:

$$
\begin{aligned}
\mathbf{L}_6(w_f) &= 0\\
H(w_f) &= 0
\end{aligned}
\tag{21}
$$

Finally the additional transversality conditions depend on the specific optimization problem, since there are two different possible goals, for T_{min} and V_{min} problems these conditions are

$$\begin{aligned} T_{min} &: L_V(w_f) = 0, \qquad L_T(w_f) = -1 \\ V_{min} &: L_V(w_f) = -1, \qquad T(w_f) = T_f \end{aligned} \tag{22}$$

It is possible to verify that under the minimum time problem conditions, the Switching function is always positive definite and the optimal solution has a continuous firing. In fact, the derivative of L_V is always negative semi-definite and the final condition $L_V(w_f) = 0$ ensures that $L_V(w_f) \geq 0$ which implies that the Switching function is always semi-positive.

2.2 Eclipse δ Layer

The entire optimization tool described here is adopted for low-thrust orbital transfers referring to satellites with electric propulsion system. The thrusters require a high amount of energy that cannot be fed by batteries alone during the eclipse phases: for this reason, the thrusters are usually switched off at these times with a significant change in the orbital transfer optimization process. To include also the eclipse problem, then, in the optimization algorithm, an eclipse function $E(\mathbf{X}, T, w, \varepsilon)$ is inserted that is negative during the shadow (eclipse) and positive during the period of sunlight. The constraint placed in f previously, that provided $f \in [0, \varepsilon]$, is then replaced by $f \in [0, \varepsilon\kappa(E(\mathbf{X}, T, w, \varepsilon))]$. In this case it is used as a non-smooth mixed control-state constraint (see [1]). But function E is also included in Equation (13) by multiplying the $\mathbf{X}$ and V state derivatives by $\kappa(E)$. This, however, leads to no longer being able to use the excellent control theory, since the equations system is transformed into a non-Lipschitz system. To overcome this problem, instead of κ, a κ_δ function is inserted, where the δ subscript identifies a small positive real number, and thus the κ_δ function is defined as:

$$\kappa_\delta(E) = \begin{cases} 1, & \forall E > \delta \\ \frac{E}{\delta}, & 0 < E < \delta \\ 0, & \forall E < 0 \end{cases} \tag{23}$$

Having restored the weakest regularity conditions to apply the Maximum Principle [2], it is possible to obtain the necessary conditions for extremality for any positive $\delta > 0$.

When treating problems without the eclipse constraint, the solution and the ΔV cost depends only on f and on the ratio between the engine thrust and the

initial satellite launch mass. The Hamiltonian is determined maximizing the pre-Hamiltonian in the controls (see also [1]):

$$H = \sup_{f \in [0,\ \varepsilon\kappa_\delta(E)],\ \underline{l} \in |\underline{l}| = 1} \tilde{H} = \varepsilon\kappa(S)\kappa_\delta(E) S(\mathbf{X}, \mathbf{L}, L_V, u) e^{\frac{V}{V_e}} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} + \varepsilon L_t \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} \tag{24}$$

where the functions S and $\kappa(S)$ have been defined in Equation (16).

The control action, as derived by maximizing the pre-Hamiltonian taking into account the eclipse functions, is

$$\begin{aligned} \underline{l} &= \frac{B(\mathbf{X}, u)^T \mathbf{L}}{\sqrt{\mathbf{L}^T B(\mathbf{X}, u) B(\mathbf{X}, u)^T \mathbf{L}}} \\ f &= \varepsilon\, \kappa(S)\, \kappa_\delta(E) \end{aligned} \tag{25}$$

The necessary conditions that any extremal satisfies are expressed by the following system of differential equations derived from the Hamiltonian with respect to the state and the costate:

$$\begin{aligned} &\frac{d\mathbf{X}}{dw} = \frac{\partial H}{\partial \mathbf{L}}; \qquad \frac{dV}{dw} = \frac{\partial H}{\partial L_V}; \qquad \frac{dT}{dw} = \frac{\partial H}{\partial L_T} \\ &\frac{d\mathbf{L}}{dw} = -\frac{\partial H}{\partial \mathbf{X}}; \qquad \frac{dL_V}{dw} = -\frac{\partial H}{\partial V}; \qquad \frac{dL_T}{dw} = -\frac{\partial H}{\partial T} \end{aligned} \tag{26}$$

The extremal equations are

$$\begin{aligned} \frac{d\mathbf{X}}{dw} &= \varepsilon\kappa(S)\kappa_\delta(E) \frac{B(\mathbf{X}, u) B(\mathbf{X}, u)^T \mathbf{L}}{\sqrt{\mathbf{L}^T B(\mathbf{X}, u) B(\mathbf{X}, u)^T \mathbf{L}}} e^{\frac{V}{V_e}} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} \\ \frac{dV}{dw} &= \varepsilon\kappa(S)\kappa_\delta(E) e^{\frac{V}{V_e}} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} \\ \frac{dT}{dw} &= \varepsilon \frac{\mathbf{X}_3}{Y_1^2} \\ \frac{d\mathbf{L}}{dw} &= -\varepsilon\kappa(S)\kappa_\delta(E) \frac{\partial \sqrt{\mathbf{L}^T B(\mathbf{X}, u) B(\mathbf{X}, u)^T \mathbf{L}}}{\partial \mathbf{X}} e^{\frac{V}{V_e}} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} \\ &\quad - \varepsilon \left(\kappa(S)\kappa_\delta(E) S e^{\frac{V}{V_e}} + L_T \right) \frac{\partial \frac{\sqrt{\mathbf{X}_3}}{Y_1^2}}{\partial \mathbf{X}} \\ &\quad - \varepsilon\kappa(S)\kappa(E) S \frac{d\kappa_\delta(E)}{dE} \frac{\partial E}{\partial \mathbf{X}} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} e^{\frac{V}{V_e}} \end{aligned} \tag{27}$$

$$\frac{dL_V}{dw} = -\varepsilon\kappa(S)\kappa_\delta(E)S\frac{e^{\frac{V}{V_e}}}{V_e}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}$$

$$\frac{dL_T}{dw} = -\varepsilon\kappa(S)\kappa(E)S\frac{d\kappa_\delta(E)}{dE}\frac{\partial E}{\partial T}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}e^{\frac{V}{V_e}}$$

Defining $U_\delta = (\mathbf{X}, T, w)|_0 \leq E(\mathbf{X}, T, w, \varepsilon) \leq \delta$, the term $\frac{d\kappa_\delta(E)}{dE} = \frac{1}{\delta}$ if $(\mathbf{X}, T, w) \in U_\delta$ and $\frac{d\kappa_\delta}{dE} = 0$ if
$(\mathbf{X}, T, w) \notin U_\delta$. The set U_δ will be called the δ layer.

2.2.1 Integrating in the δ Layer

In this section, it is introduced a specific symbol to extend the state $(\mathbf{X}, V, T)$ and the costate $(\mathbf{L}, L_V, L_T)$ vector. In this way the velocity and the time factors are directly inserted in the problem. They are denoted by $\tilde{\mathbf{X}}$, $\tilde{\mathbf{L}}$, therefore, $\tilde{\mathbf{X}}_7 = V$, $\tilde{\mathbf{X}}_8 = T$, $\tilde{\mathbf{L}}_7 = L_V$, $\tilde{\mathbf{L}}_8 = L_T$. With these symbols it is possible to represent the system of Equation (27) in the U_δ layer as:

$$\begin{aligned}
\frac{d\tilde{\mathbf{X}}}{dw} &= \varepsilon f_0(\tilde{\mathbf{X}}, w) + \varepsilon f_1(\tilde{\mathbf{X}}, \tilde{\mathbf{L}}, w)\frac{E}{\delta} \\
\frac{d\tilde{\mathbf{L}}}{dw} &= -\varepsilon g_0(\tilde{\mathbf{X}}, \tilde{\mathbf{L}}, w) - \varepsilon g_1(\tilde{\mathbf{X}}, \tilde{\mathbf{L}}, w)\frac{E}{\delta} - \varepsilon g_2(\tilde{\mathbf{X}}, \tilde{\mathbf{L}}, w)(\frac{1}{\delta}) \\
\frac{dE}{dw} &= \partial_{\tilde{\mathbf{x}}}E(\tilde{\mathbf{X}}, w)\frac{d\tilde{\mathbf{X}}}{dw} + \partial_w E(\tilde{\mathbf{X}}, w)
\end{aligned} \tag{28}$$

where $f_k, g_k, k = 0 \ldots 2$, are suitable regular functions of the arguments and do not depend on δ. It is considered now a point $P_0 \in U_\delta \otimes R^8 \otimes R = (\tilde{\mathbf{X}}_0, w_0, \tilde{\mathbf{L}}_0, E_0)$ for the system of Equation (28), that is a boundary point of U_δ such that $E(\tilde{\mathbf{X}}_0, w_0) = E_0 = 0$. This point P_0 may be an eclipse entry point if $\frac{dE}{dw} < 0$, or eclipse exit point if $\frac{dE}{dw} > 0$ or an eclipse critical point if $\frac{dE}{dw} = 0$. In order to integrate from P_0 in the system of Equation (28), it is introduced new variable $s = \frac{w-w_0}{\delta}$ and $\mathbf{E} = \frac{E}{\delta}$. With such positions the system becomes

$$\begin{aligned}
\frac{d\tilde{\mathbf{X}}}{ds} &= \varepsilon f_0(\tilde{\mathbf{X}}, w_0 + s\delta)\delta + \varepsilon f_1(\tilde{\mathbf{X}}, \tilde{\mathbf{L}}, w_0 + s\delta)\mathbf{E}\delta \\
\frac{d\tilde{\mathbf{L}}}{ds} &= -\varepsilon g_0(\tilde{\mathbf{X}}, \tilde{\mathbf{L}}, w_0 + s\delta)\delta - \varepsilon g_1(\tilde{\mathbf{X}}, \tilde{\mathbf{L}}, w_0 + s\delta)\mathbf{E}\delta - \varepsilon g_2(\tilde{\mathbf{X}}, \tilde{\mathbf{L}}, w_0 + s\delta) \\
\frac{d\mathbf{E}}{ds} &= \partial_{\tilde{\mathbf{x}}}E(\tilde{\mathbf{X}}, w_0+s\delta)(\varepsilon f_0(\tilde{\mathbf{X}}, w_0+s\delta)+\varepsilon f_1(\tilde{\mathbf{X}}, \tilde{\mathbf{L}}, w_0+s\delta)\mathbf{E})+\partial_w E(\tilde{\mathbf{X}}, w_0+s\delta)
\end{aligned} \tag{29}$$

In order to calculate the behavior of the system when crossing the δ layer, this system must be integrated in s from $\mathbf{E} = 0$ to $\mathbf{E} = 1$. For the regularity of the f_k, g_k, $k = 0, 1, 2$, the second member of the system can be developed in a series of δ. Developing also the solution on powers of δ, and introducing the series in the system of Equation (29) the solution is

$$\begin{aligned}
\frac{d\tilde{\mathbf{X}}}{ds} &= 0 \\
\frac{d\tilde{\mathbf{L}}}{ds} &= -\varepsilon g_2(\tilde{\mathbf{X}}_0, \tilde{\mathbf{L}}, w_0) \\
\frac{d\mathbf{E}}{ds} &= \partial_{\tilde{\mathbf{x}}} E(\tilde{\mathbf{X}}_0, w_0)(\varepsilon f_0(\tilde{\mathbf{X}}_0, w_0) + \varepsilon f_1(\tilde{\mathbf{X}}_0, \tilde{\mathbf{L}}, w_0)\mathbf{E} + \partial_w E(\tilde{\mathbf{X}}_0, w_0)
\end{aligned} \tag{30}$$

If the system of Equation (30) is specialized with the f_k, g_k, $k = 0, 1, 2$, specific of the system of Equation (27), it is found:

$$\begin{aligned}
&\frac{d\mathbf{X}}{ds} = 0; \qquad \frac{dV}{ds} = 0; \qquad \frac{dT}{ds} = 0 \\
&\frac{d\mathbf{L}}{ds} = -\varepsilon\kappa(S) S \frac{\partial E}{\partial \mathbf{X}} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} e^{\frac{V}{Ve}}; \qquad \frac{dL_V}{ds} = 0; \qquad \frac{dL_T}{ds} = -\varepsilon\kappa(S) S \frac{\partial E}{\partial T} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} e^{\frac{V}{Ve}} \\
&\frac{d\mathbf{E}}{ds} = \varepsilon\, \partial_T E \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} + \varepsilon \mathbf{E} \frac{\frac{\partial E^T B(\mathbf{X},u) B(\mathbf{X},u)^T \mathbf{L}}{\partial \mathbf{X}}}{\sqrt{\mathbf{L}^T B(\mathbf{X},u) B(\mathbf{X},u)^T \mathbf{L}}} e^{\frac{V}{Ve}} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} + \frac{\partial E}{\partial w}
\end{aligned} \tag{31}$$

where $\mathbf{X} = X_0$, $V = V_0$ $T = T_0$, $u = u_0$, $L_V = L_{V0}$. In addition, the term $\partial_{\mathbf{X}} E^T B(\mathbf{X}, u)^T \mathbf{L}$ is null, since during an impulsive maneuver the satellite coordinates remain constant, for the following property:

$$\frac{\partial y_k(X, w)}{\partial X_j} \frac{dX_j}{dw} = \frac{\partial y_k(X, w)}{\partial X_j} B_{j,n} = 0 \tag{32}$$

Therefore, any function that depends on $\mathbf{X}$ through the satellite position, like the eclipse function, has the gradient in the null space of the maneuver matrix.

The second member of Equation (31) in $\mathbf{E}$ is linear in s. Therefore $\mathbf{E}$ is easily integrated in $\mathbf{E} = sE_w$, where $E_w = \left(\partial w E + \varepsilon \partial T\, E \sqrt{\frac{\mathbf{X}_3}{Y_1^2}}\right)\Big|_{P_0}$. The system exists the δ layer at $s = \frac{1}{E_w}$. Now $\mathbf{L}$ and L_T are integrated between $s = [0 \ldots \frac{1}{E_w}]$.

If $(\mathbf{X}, T, w)$ is constant in the δ layer, then also $\frac{\partial E}{\partial \mathbf{X}} = \frac{\partial E}{\partial \mathbf{X}}\Big|_{P_0}$ is constant, it follows that $\mathbf{L} = L_0 + \rho(s) \frac{\partial E}{\partial \mathbf{X}}\Big|_0$, $\rho(s)$ being some scalar function. Introducing this last position in the definition of the Switching function and considering the property

described in Equation (32), it is possible to define that the Switching function $S = S(X_0, L_0)$ is constant in the δ layer. The costate is easily integrated in the δ layer (remember that $s = \frac{\mathbf{E}}{E_w}$):

$$\begin{aligned} \mathbf{L} &= L_0 - \varepsilon \left(\kappa(S) S \frac{\partial E}{\partial \mathbf{X}} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} e^{\frac{V}{V_e}} \right) \Big|_0 \frac{\mathbf{E}}{E_w} \\ L_T &= L_{T0} - \varepsilon \left(\kappa(S) S \frac{\partial E}{\partial T} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} e^{\frac{V}{V_e}} \right) \Big|_0 \frac{\mathbf{E}}{E_w} \end{aligned} \tag{33}$$

The jump of the costate in the δ layer is calculated as the variation from $\mathbf{E} = 0$ to $\mathbf{E} = 1$. This is the variation from eclipse zone to the sunlight. In terms of variation in the direction of increasing w, an additional sign is considered which is positive outwards the eclipse and negative when inwards. So the jump calculated as a variation of the costate in the $\Delta w > 0$ direction can be obtained by applying the modulus to E_w, which is negative at the eclipse entry and positive at the eclipse exit:

$$\begin{aligned} \Delta \mathbf{L} &= -\frac{\varepsilon \left(\kappa(S) S \frac{\partial E}{\partial \mathbf{X}} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} e^{V/V_e} \right) \Big|_0}{|E_w|} \\ \Delta L_T &= -\frac{\varepsilon \left(\kappa(S) S \frac{\partial E}{\partial T} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} e^{V/V_e} \right) \Big|_0}{|E_w|} \end{aligned} \tag{34}$$

The points where $E_w = 0$, where the derivative of the eclipse function becomes null and it is possible to pass from a set of eclipse orbits to fully sunlight ones or vice versa, will be called critical points. At the critical points, the costate jumps go to infinity. Such a singularity will also affect the averaged equations that will be analyzed later.

The jump of the orbital costate $\Delta \mathbf{L}$ is in the null space of the maneuver matrix $B(\mathbf{X}, u)$ as it is expressed in Equation (32). This implies the continuity of the Switching function and the continuity of the derivative of the state across the eclipse entry and exit points.

The final equations with eclipse are modified as follows:

$$\frac{d\mathbf{X}}{dw} = \varepsilon \kappa(S) \kappa(E) \frac{B(\mathbf{X}, u) B(\mathbf{X}, u)^T \mathbf{L}}{\sqrt{\mathbf{L}^T B(\mathbf{X}, u) B(\mathbf{X}, u)^T \mathbf{L}}} e^{\frac{V}{V_e}} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2}$$

$$\frac{dV}{dw} = \varepsilon \kappa(S) \kappa(E) e^{\frac{V}{V_e}} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2}$$

$$\frac{dT}{dw} = \varepsilon \frac{\sqrt{X_3}}{Y_1^2}$$

$$\begin{aligned}
\frac{d\mathbf{L}}{dw} &= -\varepsilon\kappa(S)\kappa(E)\frac{\partial\sqrt{\mathbf{L}^T B(\mathbf{X},u)B(\mathbf{X},u)^T\mathbf{L}}}{\partial\mathbf{X}}e^{\frac{V}{Ve}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2} \\
&\quad -\varepsilon\left(\kappa(S)\kappa(E)Se^{\frac{V}{Ve}}+L_T\right)\frac{\partial\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}}{\partial\mathbf{X}}-\varepsilon\kappa(S)S\frac{\partial E}{\partial\mathbf{X}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}e^{\frac{V}{Ve}}\frac{\delta(w-w_e)}{|E_w|} \\
\frac{dL_V}{dw} &= -\varepsilon\kappa(S)\kappa(E)S\frac{e^{V/V_e}}{V_e}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2} \\
\frac{dL_T}{dw} &= -\varepsilon\kappa(S)S\frac{\partial E}{\partial T}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}e^{\frac{V}{Ve}}\frac{\delta(w-w_e)}{|E_w|}
\end{aligned} \tag{35}$$

where $\delta(w-w_e)$ is the Dirac function at the eclipse entrance $w_e = w_{entry}(\mathbf{X},T)$ and at the eclipse exit $w_e = w_{entry}(\mathbf{X},T)$ anomalies of any orbit. In this case it is considered a strictly decreasing sequence of $\delta_n \to 0$.

This sequence generates, using Equation (27) and a set of boundary conditions, a series of δ_n-regularized extremal solutions. The succession of objective function generated by these δ_n-regularized extremal solution will be decreasing because for any $\delta_a > \delta_b$ the optimal solution with the δ_a-regularized will always be in the set of the feasible solutions for the δ_b-regularized optimal problem. The sequence of objective functions will then be convergent, because it is naturally limited from below. It is possible to deduce that there exists a δ-regularized optimal solution as close as desired to this objective function limit. If, in addition, there exists a solution of the system of Equation (35) with the same boundary conditions, then it will be the limit of the δ_n-regularized extremal solutions and it will have as cost the limit of the objective function series.

2.3 Averaging

It is possible to note that in the system of Equations (27) and (35) the term $\varepsilon = \frac{\dot{m}V_e}{\frac{M_0}{\mu a_0^2}}$ is always present. In these applications, ε is considered as a small parameter in the range of $\approx 10^{-3}$, 10^{-4}. The second members of systems of Equations (27) and (35) are periodic functions of the independent variable. Therefore, the system is set in the perturbation standard form (see [13], Chapter 1.6). It is possible to use the averaging technique to implement an approximate solution. Here it is applied the averaging on a suitable δ - regularized system of Equation (27) for a given $\delta > 0$, then the averaged equations are derived and finally δ is zero. In order to do this an averaging operator is introduced

$$\begin{aligned}\mathcal{E}_w[H_\delta(\mathbf{X},\mathbf{L},w)] &= \frac{1}{2\pi}\int_0^{2\pi} H_\delta(\mathbf{X},\mathbf{L},w)\,dw \\ \mathcal{E}_w[H_\delta(\mathbf{X},\mathbf{L},w)] &= \overline{H}_\delta(\mathbf{X},\mathbf{L})\end{aligned} \tag{36}$$

It is possible to note that the averaging is performed in the independent variable w. The state and the costate, including the variable T which is the slow time used to describe the Sun motion, are considered fixed during the averaging integral.

$$\overline{H}_\delta(\mathbf{X},\mathbf{L}) = \varepsilon\mathcal{E}_w\left[\kappa(S)S\kappa_\delta(E)\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}\right]e^{\frac{V}{V_e}} + \varepsilon L_T\mathcal{E}_w\left[\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}\right] \tag{37}$$

The average Hamiltonian is then independent of w and X_6. The average equations can be obtained using the $\mathcal{E}_w[.]$ on the second members of the system of Equations (27) and (35) or deriving the Hamiltonian flow from the averaged Hamiltonian:

$$\begin{aligned}\frac{d\mathbf{X}}{dw} &= \frac{\partial\overline{H}_\delta}{\partial\mathbf{L}} \\ \frac{d\mathbf{L}}{dw} &= -\frac{\partial\overline{H}_\delta}{\partial\mathbf{X}}\end{aligned} \tag{38}$$

In this case $\frac{\partial\overline{H}_\delta}{\partial\mathbf{X}_6} = 0$. When $\delta \to 0$, the resulting averaged equations are

$$\begin{aligned}\frac{d\mathbf{X}}{dw} &= \varepsilon\mathcal{E}_w\left[\kappa(S)\kappa(E)\frac{B(\mathbf{X},u)B(\mathbf{X},u)^T\mathbf{L}}{\sqrt{\mathbf{L}^T B(\mathbf{X},u)B(\mathbf{X},u)^T\mathbf{L}}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}\right]e^{\frac{V}{V_e}} \\ \frac{dV}{dw} &= \varepsilon\mathcal{E}_w\left[\kappa(S)\kappa(E)e^{\frac{V}{V_e}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}\right]e^{\frac{V}{V_e}} \\ \frac{dT}{dw} &= \varepsilon\mathcal{E}_w\left[\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}\right] \\ \frac{d\mathbf{L}}{dw} &= -\varepsilon\mathcal{E}_w\left[\kappa(S)\kappa(E)\frac{\partial\sqrt{\mathbf{L}^T B(\mathbf{X},u)B(\mathbf{X},u)^T\mathbf{L}}}{\partial\mathbf{X}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}\right]e^{\frac{V}{V_e}} \\ &\quad -\varepsilon\mathcal{E}_w\left[\kappa(S)\kappa(E)Se^{\frac{V}{V_e}} + L_T\right]\frac{\partial\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}}{\partial\mathbf{X}} \\ &\quad -\varepsilon\left(\kappa(S)S\frac{\partial E}{\partial\mathbf{X}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}e^{\frac{V}{V_e}}\frac{1}{2\pi E_w}\right)\Big|_{w_{entry}}^{w_{exit}}\end{aligned} \tag{39}$$

$$\frac{L_V}{dw} = -\varepsilon \mathcal{E}_w \left[\kappa(S)\kappa(E) S \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} \right] \frac{e^{\frac{V}{V_e}}}{V_e}$$

$$\frac{L_T}{dw} = -\varepsilon \left(\kappa(S) S \frac{\partial E}{\partial T} \frac{\sqrt{\mathbf{X}_3}}{Y_1^2} e^{\frac{V}{V_e}} \frac{1}{2\pi E_w} \right) \Big|_{w_{entry}}^{w_{exit}}$$

If in the Hamiltonian instead of $\kappa(E)$ it is inserted $\kappa(E_\delta)$, it is obtained an averaged Hamiltonian similar to that presented in [12], where the effects of the mixed control-state constraint, which are represented by the costate jumps averaged over one period, are not analyzed. This approach will be called *Heuristic*: in certain missions, including the costate jumps in the averaged equations produces a significant difference in the optimized solution. These equations can be associated with the boundary conditions of Equations (17) and (21) where $H(w_f)$ is substituted by $\overline{H}(w_f)$ and (22) depending on the specific problem. The exploitation of the system of Equation (39) is numerically faster than the solution given by the system of Equation (35) and, using the proximity of the two trajectories, the solution of the system of Equation (35) can be easily solved numerically starting from the averaged one.

This implies the calculations of the short terms of the solution. To do this, the initial costate found by the average solution, is changed of a ε small quantity in order to satisfy terminal conditions at a final w_f which is close to the one of the "averaged extremal." The terminal condition on the average Hamiltonian $\overline{H}(w_f) = 0$ guarantees that the close by extremal has a null Hamiltonian $H(w_f^*) = 0$ at some point of the same orbit w_f^*.

2.4 Orbital Perturbations

Another improvement in the calculation is applied taking into account the effects of the orbital perturbations (as the J_2 effects). It represents an important problem especially for long transfer. In particular, the rotation of the node has a significant effect on LEO to MEO/GEO transfers. When a perturbation creates a node it should be left rotating within the boundary conditions so to not consume propellant and keeping it fixed. Such a condition is expressed by setting $L_{f5} = 0$ and keeping X_{f5} free. Mathematically, the perturbation effects can be considered, by adding a term to the Hamiltonian (see [10] and [9]). If $A_p(X, w)$ is a perturbation field such that the dynamic equation (10) becomes

$$\frac{d\mathbf{X}}{dw} = B(\mathbf{X}, u = w + \mathbf{X}_6)(A + A_p(\mathbf{X}, w)) \tag{40}$$

then the Hamiltonian of the optimal control problem is

$$H = \varepsilon\kappa(S)S\kappa(E)e^{\frac{V}{Ve}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2} + \varepsilon L_T\frac{\sqrt{\mathbf{X}_3}}{Y_1^2} + \mathbf{L}^T B(\mathbf{X},u)A_p(\mathbf{X},w) \tag{41}$$

In the averaged version it is introduced $\mathcal{E}_w[B(\mathbf{X},u)A_p(\mathbf{X},w)] = \overline{\mathbf{X}}_p(\mathbf{X})$.

The extremal equations for the averaged case that are modified in the presence of perturbations are

$$\begin{aligned}
\frac{d\mathbf{X}}{dw} &= \varepsilon\mathcal{E}_w\left[\kappa(S)\kappa(E)\frac{B(\mathbf{X},u)B(\mathbf{X},u)^T\mathbf{L}}{\sqrt{\mathbf{L}^T B(\mathbf{X},u)B(\mathbf{X},u)^T\mathbf{L}}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}\right]e^{\frac{V}{Ve}} + \overline{\mathbf{X}}_p(\mathbf{X}) \\
\frac{d\mathbf{L}}{dw} &= -\varepsilon\mathcal{E}_w\left[\kappa(S)\kappa(E)\frac{\partial\sqrt{\mathbf{L}^T B(\mathbf{X},u)B(\mathbf{X},u)^T\mathbf{L}}}{\partial\mathbf{X}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}\right]e^{\frac{V}{Ve}} \\
&\quad -\varepsilon\mathcal{E}_w\left[\kappa(S)\kappa(E)Se^{\frac{V}{Ve}} + L_T\right]\frac{\partial\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}}{\partial\mathbf{X}} \\
&\quad -\varepsilon\left(\kappa(S)S\frac{\partial E}{\partial\mathbf{X}}\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}e^{\frac{V}{Ve}}\frac{1}{2\pi E_w}\right)\Big|_{w_{entry}}^{w_{exit}} - \frac{\partial\mathbf{L}^T\overline{\mathbf{X}}_p(\mathbf{X})}{\partial\mathbf{X}}
\end{aligned} \tag{42}$$

Since the perturbation action of the J_2 effects is taken into account during the computation, it is necessary to study the J_2 potential, that is:

$$F_2 = -J_2\frac{\mu}{r}\left(\frac{R}{r}\right)^2\left(\frac{3\sin(\theta)^2 - 1}{2}\right) \tag{43}$$

The three accelerations are calculated from the gradient of the potential dividing it by $\frac{\mu}{a_0^2}$ to obtain non-dimensional accelerations (A_r, A_t, and A_n). Then, applying the averaging: $\overline{\mathbf{X}}_p(\mathbf{X}) = \mathcal{E}_w\left[B(\mathbf{X},u)A\frac{\sqrt{\mathbf{X}_3}}{Y_1^2}\right]$. This can be done by quadratures, calling $j_2 = \left(\frac{J_2R^2}{a_0^2}\right)$ it is obtained

$$\overline{\mathbf{X}}_{p1} = -\frac{3}{2}j_2\frac{\mathbf{X}_2\mathbf{X}_3^2}{(\mathbf{X}_4^2+\mathbf{X}_5^2+1)^2}(3\mathbf{X}_4^4 + 3\mathbf{X}_5^4 - 6\mathbf{X}_4^2 - 6\mathbf{X}_5^2 + 6\mathbf{X}_4^2\mathbf{X}_5^2 + 1)$$

$$\overline{\mathbf{X}}_{p2} = \frac{3}{2}j_2\frac{\mathbf{X}_1\mathbf{X}_3^2}{(\mathbf{X}_4^2+\mathbf{X}_5^2+1)^2}(3\mathbf{X}_4^4 + 3\mathbf{X}_5^4 - 6\mathbf{X}_4^2 - 6\mathbf{X}_5^2 + 6\mathbf{X}_4^2\mathbf{X}_5^2 + 1)$$

$$\overline{\mathbf{X}}_{p3} = 0$$

$$\overline{\mathbf{X}}_{p4} = -\frac{3}{2}\, j_2 \frac{\mathbf{X}_5\, \mathbf{X}_3^2}{\mathbf{X}_4^2 + \mathbf{X}_5^2 + 1} (\mathbf{X}_4^2 + \mathbf{X}_5^2 - 1)$$

$$\overline{\mathbf{X}}_{p5} = \frac{3}{2}\, j_2 \frac{\mathbf{X}_4\, \mathbf{X}_3^2}{\mathbf{X}_4^2 + \mathbf{X}_5^2 + 1} (\mathbf{X}_4^2 + \mathbf{X}_5^2 - 1) \tag{44}$$

$$\overline{\mathbf{X}}_{p6} = 3\, j_2 \frac{\mathbf{X}_3^2}{(\mathbf{X}_4^2 + \mathbf{X}_5^2 + 1)^2} (\mathbf{X}_4^4 + \mathbf{X}_5^4 - \mathbf{X}_4^2 - \mathbf{X}_5^2 + 2\, \mathbf{X}_4^2\, \mathbf{X}_5^2)$$

The costate equation terms can be obtained with a further derivation in the state as the expression $\partial L^T \frac{\overline{\mathbf{X}}_p}{\partial \mathbf{X}}$.

2.5 Introduction of Inequality Constraints on the State Along the Path

Considering some specific cases of GTO to MEO orbital transfer that require big plane changes in the presence of J_2, some solutions appear unrealistic, since the perigee is lower than the Earth radius. For this reason, a barrier function has been introduced in order to avoid the satellite-Earth collision.

Since the optimal control solution is found by maximizing the Hamiltonian H, the cost function has to penalize (i.e., reduce) the value of H when the state approaches the inadmissible area of its domain. Calling H_0 the Hamiltonian of the original problem without constraints, the Hamiltonian of the constrained problem is expressed as:

$$H(\mathbf{X}, \mathbf{L}) = H_0(\mathbf{X}, \mathbf{L}) - P \cdot W(r(\mathbf{X})) \tag{45}$$

where

- P is the weight of the barrier
- $r(\mathbf{X} > 0)$ represents the inequality constraint and is a function of the state which equals 0 on the boundary of the inadmissible domain;
- $W(r(\mathbf{X}))$ is the *barrier function*, defined as a function such that:
  - $W(r(\mathbf{X})) \geq 0$ for $r(\mathbf{X}) > 0$ (strictly admissible zone)
  - $W(r(\mathbf{X})) \to +\infty$ for $r(\mathbf{X}) \to 0$ (boundary of the inadmissible zone)

The optimal equations of motion (27) are then modified as:

$$\begin{aligned} \dot{\mathbf{X}} &= \frac{dH}{d\mathbf{L}} = \frac{dH_0}{d\mathbf{L}} \\ \dot{\mathbf{L}} &= -\frac{dH}{d\mathbf{X}} = -\frac{dH_0}{d\mathbf{X}} + P \cdot \frac{dW}{dr} \frac{dr}{d\mathbf{X}} \end{aligned} \tag{46}$$

3 Numerical Results

All the theory outlined above is applied to find numerical solutions, through the code developed by the authors and named *SOFTT* (Space Optimal Finite Thrust Transfer). Using SOFTT was possible to resolve many mission analyzes for different proposals such as Galileo Second Generation.

In this section some interesting results are reported

- Minimum time problem for orbital transfers to achieve MEO and GEO starting from LEO and GTO. In these cases both the J_2 and the eclipse effects are considered in the optimization.
- Analysis of the relationship between the launch mass and the minimum transfer time to achieve the GEO starting from an elliptical GTO.
- Minimum mass consumption problem (V_{min}) for a GTO to GEO orbital transfer. This study is reported to show how the ΔV mission cost increases as function of the available transfer time. For the V_{min} problem the solution convergence is very slow thus the J_2 and eclipse effects are not considered only to minimize the computational time.
- Constellation deployment application, with the study of the evolution of the ΔV mission cost as function of the final right ascending node. The introduction of the J_2 effects increases the problem complexity and the perigee altitude constraint effects are explained. In this study too, the solution convergence is very slow thus eclipse effects are not considered in order to analyze the problem regardless of the Sun longitude.

3.1 Typical Orbital Transfers to MEO

In this section two minimum time orbital transfers to achieve MEO are reported. The first mission starts from a circular LEO while the second starts from an elliptic GTO.

Referring to the Galileo Second Generation proposal, the orbital Keplerian parameters, the satellite, and the propulsion system characteristics are the following:

- LEO: $a = 7378\,\text{km}$; $e = 0$; $I = 56°$;
- GTO: $a = 24{,}475\,\text{km}$; $e = 0.7292$; $I = 6°$; $g = 178°$;
- MEO: $a = 29{,}601\,\text{km}$; $e = 0$; $I = 56°$.

- Launch mass = 1500 kg;
- Thrust = 0.6 N;
- I_{sp} = 1500 s.

In these cases the eclipse effects are considered. Many figures are reported in order to compare the main differences with and without eclipse effects. Furthermore the J_2 effects are always considered.

3.1.1 LEO to MEO Orbital Transfer

Considering the first mission, from circular LEO to circular MEO, the orbital transfer results are reported in Table 1 in terms of transfer time, mission ΔV, and required propellant mass. It possible to note a significant variation on the transfer time ($\approx$12%) and a smaller change of the mission ΔV. These are expected results due to the presence of the perturbations: in particular taking into account the eclipse presence, the transfer time increases due to the fact that during the eclipse phases the thrusters are switched off. The variation on the mission ΔV is due to the J_2 presence.

The following Figures 1 and 2 show the evolution of the Keplerian parameters in terms of semi-major axis (Figure 1 left), eccentricity (Figure 1 right), inclination (Figure 2 left), and right ascending node (Figure 2 right).

Looking at the evolution of the semi-major axis (Figure 1 left) it is clear how the transfer time increases due to the introduction of the eclipse effects but the behavior

Table 1 Orbital transfer results with and without eclipse effects

Initial Orbit	*Target Orbit*	*Eclipse*	*Time [days]*	*ΔV [km/s]*	*ΔM [kg]*
LEO	MEO	No	94.2	3.681	331.9
		Yes	105.8	3.720	335.1

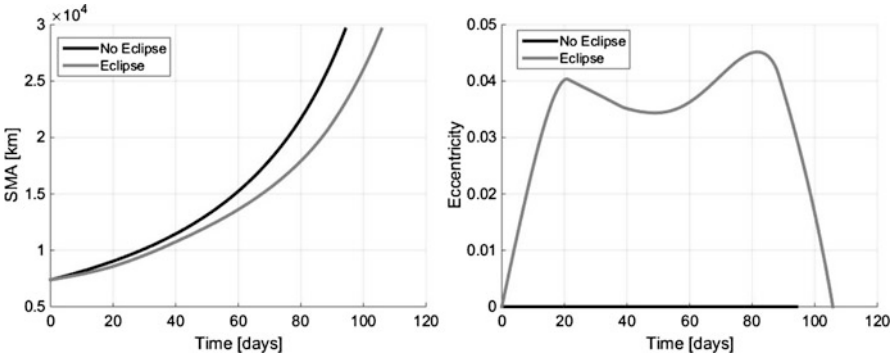


Figure 1 Evolution of the semi-major axis (left) and the eccentricity (right) with and without eclipse effects

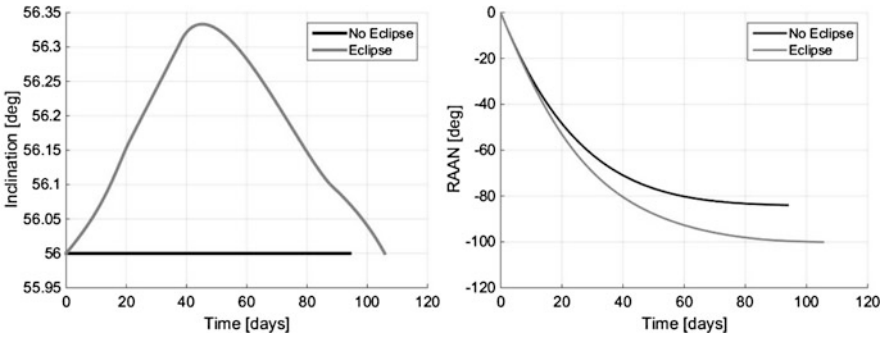


Figure 2 Evolution of the inclination (left) and the right ascending node (right) with and without eclipse effects

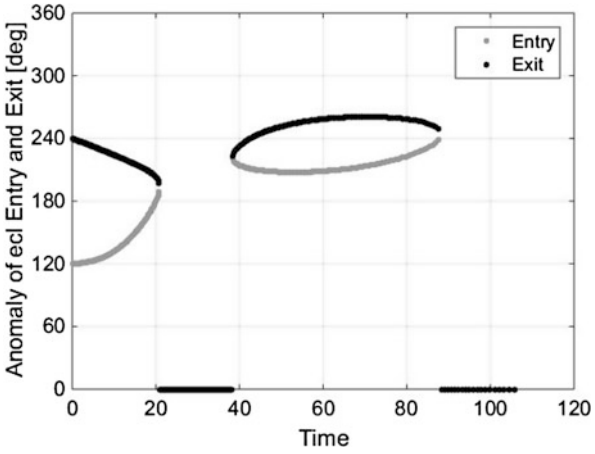


Figure 3 Evolution of the eclipse entry/exit anomaly during the orbital transfer

remains practically the same. The evolution of the eccentricity (Figure 1 right) and the inclination (Figure 2 left) are practically the same.

Figure 3 shows the evolution of the eclipse anomaly entry (grey line) and exit (black line) during the orbital transfer.

Finally Figure 4 shows the evolution of the square acceleration components in radial, normal, and bi-normal direction, obtained during the orbital transfer: with eclipse effects (left) and without them (right). In the right figure (no eclipse) it is possible to note that all components are equal to zero except the ortho-radial that is equal to one. This means that in this case the thrust direction is tangential to the orbit, like a classic spiral transfer. Taking into account the eclipse effects, the fire vector components show some differences: in particular, in the last part of the transfer the ortho-radial component too is different from zero to maintain the final orbit circular.

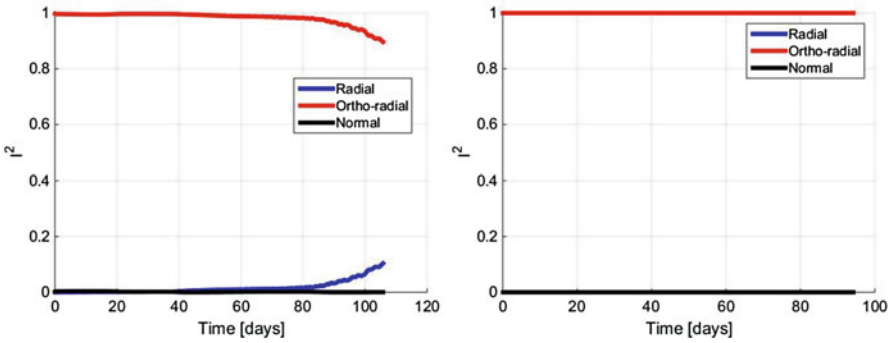


Figure 4 Evolution of the square acceleration components with (left)and without (right) eclipse effects during the orbital transfer

3.1.2 GTO to MEO Orbital Transfer

Considering the second mission, from elliptic GTO to circular MEO, the orbital transfer results are reported in Table 2 in terms of transfer time, mission ΔV, and required propellant mass.

In this case the variation on the transfer time is lower ($\approx$2%) than the previous case ($\approx$12%).

Figures 5 and 6 show the evolution of the Keplerian parameters in terms of semi-major axis (Figure 5 left), eccentricity (Figure 5 right), inclination (Figure 6 left), and right ascending node (Figure 6 right).

As introduced in Table 2, looking at these figures it is possible to confirm how the introduction of the eclipse does not influence the evolution of the orbital parameters.

Figure 7 shows the evolution of the eclipse anomaly entry (grey line) and exit (black line) during the orbital transfer, while Figure 8 shows the evolution of the square acceleration components with (left) and without (right) eclipse effects.

The evolution of the accelerations is practically the same justifying the behavior of the Keplerian parameters that are very similar considering or not the eclipse effects.

Table 2 Orbital transfer results with and without eclipse effects

Initial Orbit	*Target Orbit*	*Eclipse*	*Time [days]*	*ΔV [km/s]*	*ΔM [kg]*
GTO	MEO	No	88.9	3.448	313.3
		Yes	90.3	3.435	312.3

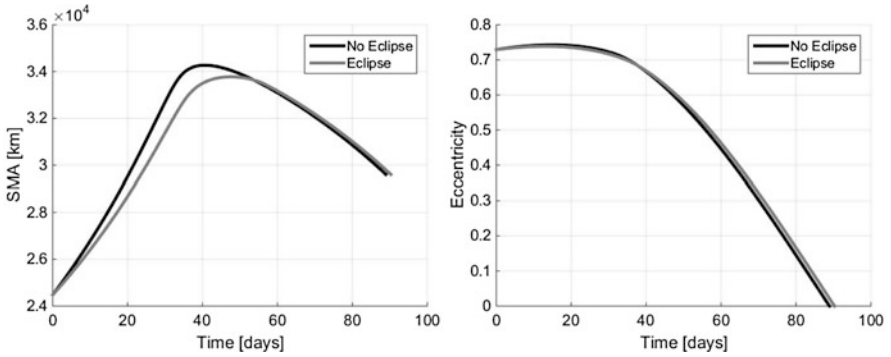


Figure 5 Evolution of the semi-major axis (left) and the eccentricity (right) with and without eclipse effects

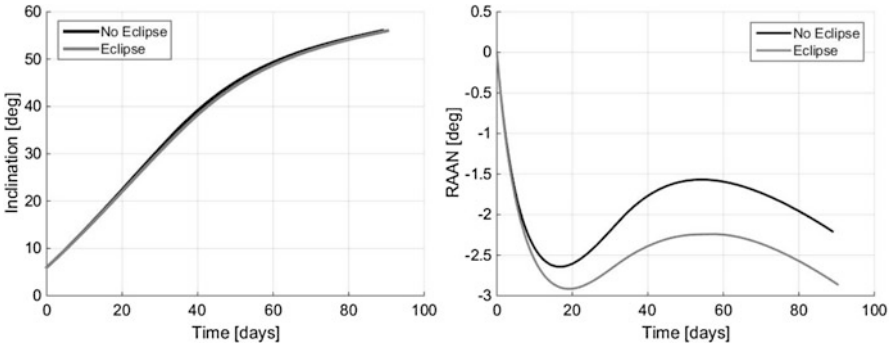
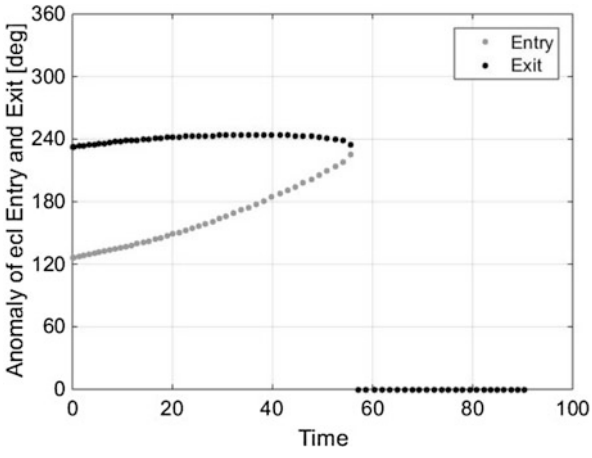


Figure 6 Evolution of the inclination (left) and the right ascending node (right) with and without eclipse effects

Figure 7 Evolution of the eclipse entry/exit anomaly during the orbital transfer



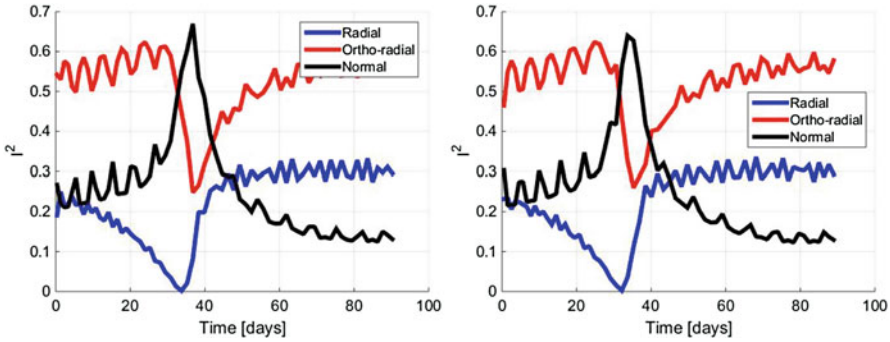


Figure 8 Evolution of the square acceleration components with (left) and without (right) eclipse effects during the orbital transfer

3.2 Typical Orbital Transfers to GEO

Two different orbital transfers to achieve GEO are examined. The first mission starts from a circular LEO while the second starts from an elliptic GTO.

Referring to the Galileo Second Generation proposal, the orbital Keplerian parameters, the satellite, and the propulsion system characteristics are the following:

- LEO: $a = 7378\,\text{km}$; $e = 0$; $I = 5°$;
- GTO: $a = 24{,}475\,\text{km}$; $e = 0.7292$; $I = 6°$; $g = 178°$;
- GEO: $a = 42{,}164\,\text{km}$; $e = 0$; $I = 0°$.

- Launch mass = 1500 kg;
- Thrust = 0.6 N;
- I_{sp} = 1500 s.

In these cases too the eclipse effects are considered case by case in order to underline the main differences between the different analysis. Otherwise the J_2 effects are always considered.

3.2.1 LEO to GEO Orbital Transfer

Considering the first mission, from circular LEO to circular GEO, the orbital transfer results are reported in Table 3 in terms of transfer time, mission ΔV, and required propellant mass. It is possible to note a significant variation on the transfer time ($\approx$14%) and a smaller change of the mission ΔV cost. As seen in Sect. 3.1.1 the transfer time variation was expected due to the presence of the eclipse, while the variation on the mission ΔV cost is due to the J_2 presence.

Figures 9 and 10 show the evolution of the Keplerian parameters in terms of semi-major axis (Figure 9 left), eccentricity (Figure 9 right), and inclination

Table 3 Orbital transfer results with and without eclipse effects

Initial Orbit	Target Orbit	Eclipse	Time [days]	ΔV [km/s]	ΔM [kg]
LEO	GEO	No	108.2	4.316	381.3
		Yes	123.5	4.332	382.5

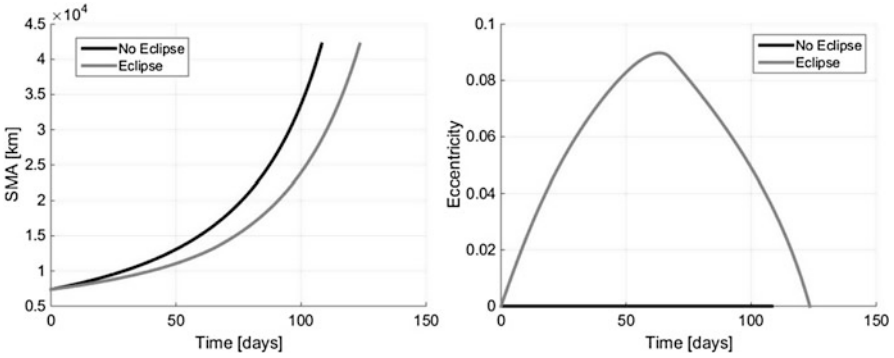
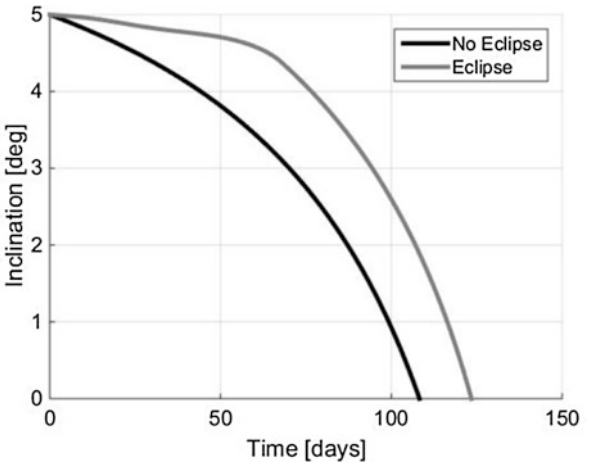


Figure 9 Evolution of the semi-major axis (left) and the eccentricity (right) with and without eclipse effects

Figure 10 Evolution of the inclination with and without eclipse effects



(Figure 10). Looking at the evolution of the semi-major axis (Figure 9 left) it is possible to note how the transfer time increases due to the introduction of the eclipse effects but the behavior remains practically the same. The evolution of the eccentricity (Figure 9 right) and the inclination (Figure 10) are quite different but the absolute values have small changes.

Figure 11 shows the evolution of the eclipse anomaly entry (grey line) and exit (black line) during the orbital transfer.

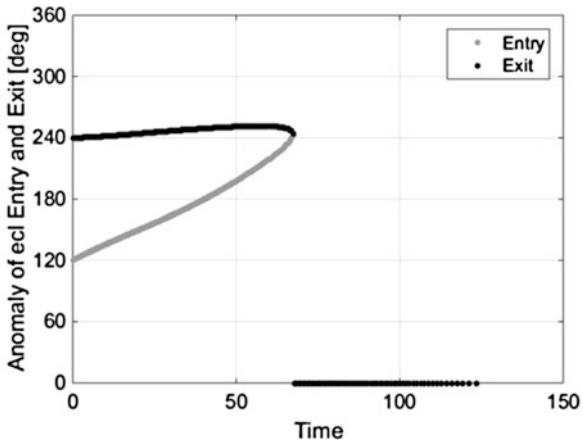


Figure 11 Evolution of the eclipse entry/exit anomaly during the orbital transfer

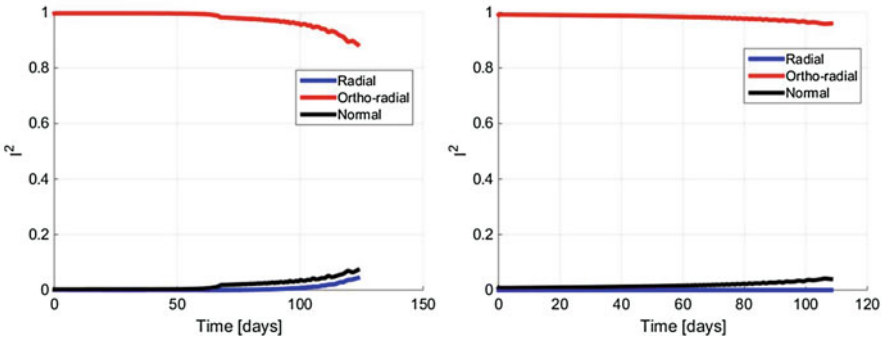


Figure 12 Evolution of the square acceleration components with (left) and without (right) eclipse effects during the orbital transfer

Finally Figure 12 shows the evolution of the square acceleration components in radial, normal, and bi-normal direction, obtained during the orbital transfer: with eclipse effects (left) and without them (right).

Taking into account the eclipse effects or not the fire vector components show some differences but practically the evolution is quite similar. With respect to Sect. 3.1, now the normal component too is different from zero, justifying an out of plane maneuver in order to correct the orbital inclination.

3.2.2 GTO to GEO Orbital Transfer

Considering the second mission, from elliptical GTO to circular GEO, the numerical results are reported in Table 4. Looking at Table 4, in this case too, the transfer time remains the same ($\approx$1%) due to the lower eclipse period with respect to the case in Sect. 3.1.

Table 4 GTO to GEO orbital transfer resulting using the software SOFTT

Initial Orbit	Target Orbit	Eclipse	Time [days]	ΔV [km/s]	ΔM [kg]
GTO	GEO	No	59.1	2.200	208.3
		Yes	59.7	2.173	206.0

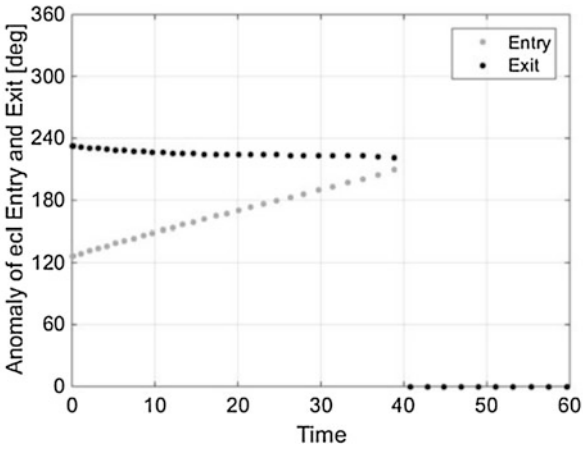


Figure 13 Evolution of the eclipse entry/exit anomaly during the orbital transfer

The comparison between the evolution of the Keplerian parameters and the accelerations development with and without eclipse effects is not interesting due to the same behavior of these parameters. For this reason, figures that represent the Keplerian elements and the accelerations are not reported. Figure 13 shows the evolution of the eclipse anomaly entry (grey line) and exit (black line) during the orbital transfer.

3.3 Different Launch Mass for Typical Orbital Transfer: GTO to GEO

Considering a minimum time orbital transfer, in this section a typical GTO to GEO orbital transfer is considered for different launch masses.

The initial and final orbital Keplerian parameters for this analysis are the following:

- GTO: $a = 24{,}475\,\text{km}$; $e = 0.7292$; $I = 6°$; $g = 178°$;
- GEO: $a = 42{,}164\,\text{km}$; $e = 0$; $I = 0°$.

The electrical propulsion system considered has the following characteristics:

- Thrust = 0.3 N;
- I_{sp} = 1500 s.

Different values of the initial launch mass are considered: 1500, 1650, 1800, and 2000 kg.

Both the eclipse and J_2 perturbations effects are considered.

The numerical orbital transfer results are reported in Table 5 in terms of transfer time, ΔV mission cost, and mass consumption.

Looking at this table it is possible to note how the transfer time increases as a function of the initial launch mass: this is an expected result because maintaining the same thrust value and increasing the launch mass, the acceleration decreases then the transfer time results higher.

Otherwise the ΔV mission cost does not change due to the fact that the Keplerian parameters are always the same. Figures 14 and 15 show the evolution of the Keplerian parameters in terms of semi-major axis (Figure 14 left), eccentricity (Figure 14 right), and inclination (Figure 15).

Table 5 Orbital transfer results for different values of initial launch mass

Initial Orbit	*Target Orbit*	*Lanch Mass [kg]*	*Time [days]*	*ΔV [km/s]*	*ΔM [kg]*
GTO	GEO	1500	119.2	2.182	206.7
		1650	131.1	2.183	227.5
		1800	143.0	2.185	248.3
		2000	158.9	2.186	276.1

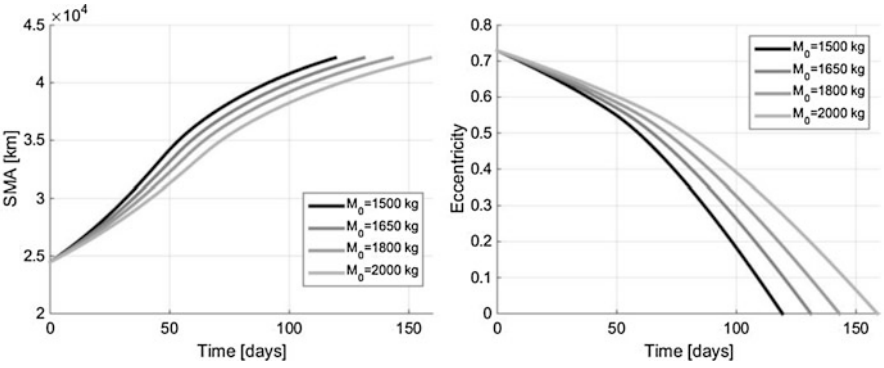


Figure 14 Evolution of the semi-major axis (left) and the eccentricity (right) with and without eclipse effects

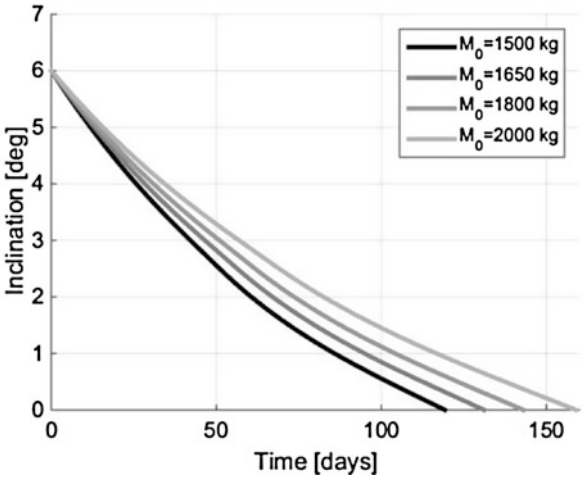


Figure 15 Evolution of the inclination with and without eclipse effects

Looking at the evolution of the Keplerian parameters, it is possible to note how the transfer time increases as function of the initial launch mass, but the behavior is practically the same.

Figures that represent the evolution of the eclipse entry/exit anomaly and the accelerations are not reported.

3.4 Minimum Velocity Orbital Transfer from GTO to GEO

Here a different optimization approach is used with respect to the previous Sects. 3.1, 3.2, and 3.3, the optimization was done minimizing the transfer time. In this section an analysis of elliptical GTO to circular GEO orbital transfer is reported minimizing the mass consumption (V_{min} problem).

Considering the minimum mass consumption, SOFTT estimates the arcs where it is convenient to switch on the thrusters during the orbital transfer. The longer is the available transfer time, the lower is the ΔV cost of the mission due to the longer time with thrusters switched off.

In this kind of analysis the convergence is very slow. SOFTT is able to compute the thrust strategy including the perturbations effects but in this paper, the eclipse and the J_2 effects are not considered, only in order to save computational time.

For this analysis the initial and final Keplerian parameters are the following:

- GTO: $a = 24{,}475\,\text{km}$; $e = 0.7292$; $I = 6°$; $g = 178°$;
- GEO: $a = 42{,}164\,\text{km}$; $e = 0$; $I = 0°$.

The satellite and the electrical propulsion system characteristics data are

- Launch mass = 2000 kg;
- Thrust = 0.3 N;
- I_{sp} = 1500 s.

Table 6 reports the orbital transfer results in terms of ΔV mission cost and mass consumption as function of the available transfer time.

Figure 16 shows the evolution of the ΔV mission cost as function of the available transfer time.

Table 6 Orbital transfer results for different values of the available transfer time

Time [days]	*ΔV [km/s]*	*ΔM [kg]*
157.6	2.200	277.7
167.7	2.010	255.4
177.7	1.917	244.3
187.8	1.849	236.2
197.6	1.797	230.0
207.6	1.757	225.1
217.6	1.724	221.1
227.6	1.698	218.0
237.5	1.676	215.4
247.6	1.658	213.1
257.6	1.643	211.2
267.7	1.629	209.6
277.6	1.618	208.2

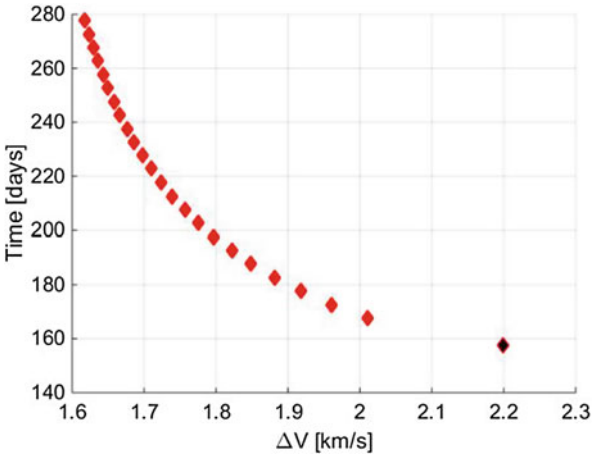


Figure 16 Evolution of the ΔV mission cost as function of the available transfer time

Starting from the T_{min} solution (black point), each solution was found with an iterative method, increasing step by step the available transfer time and using the initial costate of the previous solution.

Figure 16 shows how the ΔV mission cost decreases with higher available transfer time: the total velocity reaches an asymptotic value that represents the minimum possible velocity expendable, the impulsive maneuver.

In order to show the evolution of the Keplerian parameters, four transfer times are selected: 157 (that corresponds to the T_{min} solution), 197, 227, and 277 days.

Figures 17 and 18 show the evolution of the Keplerian parameters in terms of semi-major axis (Figure 17 left), eccentricity (Figure 17 right), and inclination (Figure 18) increasing the available transfer time.

Figure 19 shows the evolution of the Switching function on three particular orbits for T_{min} case (left) and for V_{min} case at fixed time 277 days (right). It is possible to note that in the T_{min} case (Figure 19 left) the Switching function is always positive,

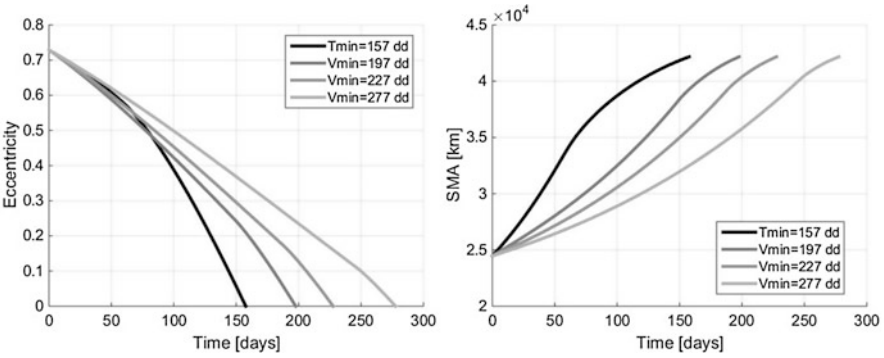


Figure 17 Evolution of the semi-major axis (left) and the eccentricity (right) for different values of the available transfer time

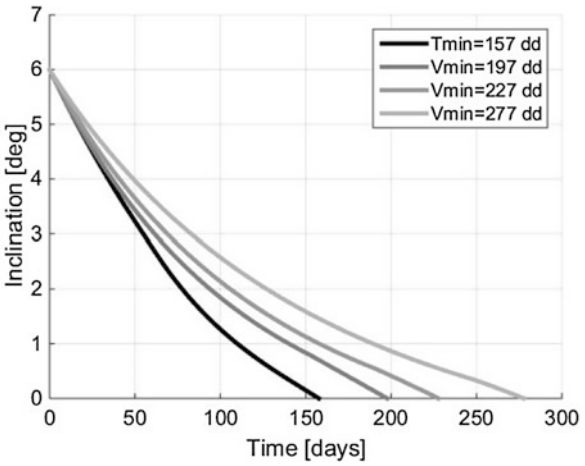


Figure 18 Evolution of the inclination for different values of the available transfer time

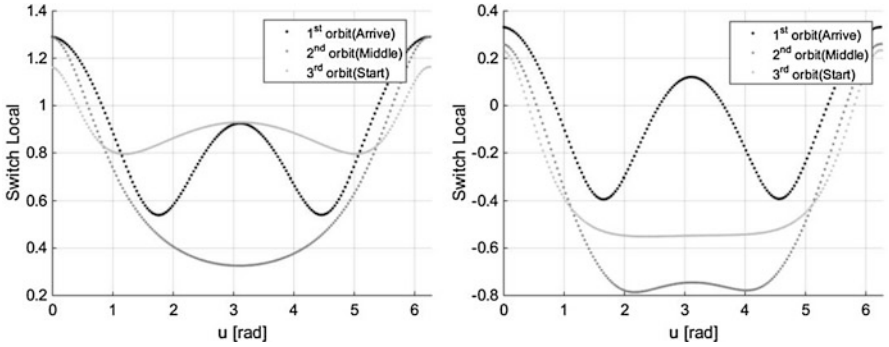


Figure 19 Switching function for T_{min}(left) and V_{min}(right) solutions: $S > 0$ thrusters are on, $S < 0$ thrusters are off

meaning that the thrusters were always switched on during the transfer. Otherwise in the V_{min} case (Figure 19 right) the Switching function can be positive or negative, meaning that the thrusters were switched on only on the arcs where it was convenient to fire.

3.5 Orbital Transfer to MEO Delta-RAAN Analysis

The constellation deployment consists in launching more than one satellite at the same time with the same launcher and achieve different planes characterized by different right ascension values.

For this study the approach was to compute the evolution of the ΔV mission cost as function of the change of the delta-RAAN which is defined as the difference between the final RAAN and the initial or separation RAAN.

Two different analyses are reported later: a GTO to MEO and a LEO to MEO delta-RAAN mission analyses. In both of them the eclipse effects are not included, in order to analyze a problem independent on the initial Sun longitude (see [11] for more details).

Otherwise the J_2 presence is considered due to the relevant effects on the orbital transfers with strong change of the right ascending node.

The initial (GTO) and final (MEO) Keplerian parameters are the following:

- GTO: $a = 24{,}475\,\text{km}$; $e = 0.7292$; $I = 178°$;
- MEO: $a = 29{,}601\,\text{km}$; $e = 0$; $I = 56°$.

The satellite and the propulsion system characteristics are

- Launch mass = 1800 kg;
- Thrust = 0.240 N;
- I_{sp} = 1500 s.

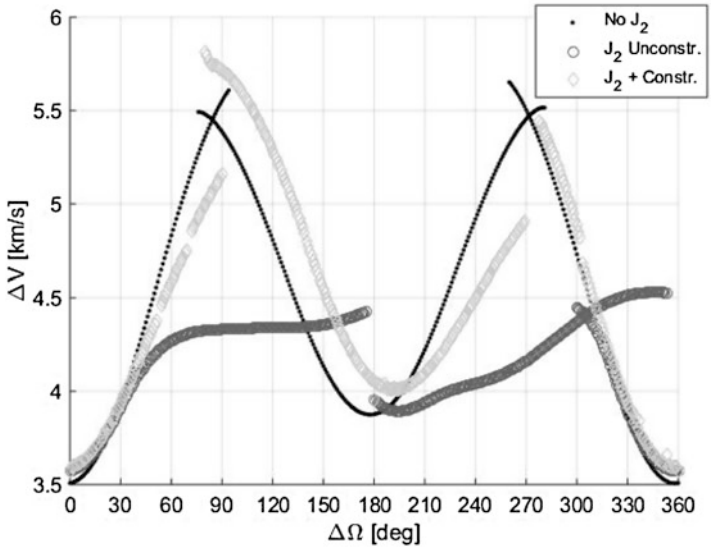


Figure 20 Evolution of the ΔV mission cost as function of the change of the delta-RAAN

Figure 20 shows the evolution of the ΔV mission cost as function of the change of the delta-RAAN. In particular three different lines are reported: the black one represents the ΔV cost without considering the J_2 effects, the dark grey line represents the ΔV cost with J_2 effects, and the light grey line represents the ΔV cost considering the J_2 effects and the perigee altitude constraint.

As reported in [11] Figure 20 shows the possibility to have two different solutions for the same initial and final conditions, one with a ΔV higher than the other, just changing the initial costate L_0. In this particular case the second order necessary and sufficient conditions should be adopted to verify which solution is a local minimizer (see [5]).

Furthermore the introduction of the J_2 changes the ΔV evolution breaking the symmetry with respect to the delta-RAAN and providing unrealistic solutions that consider an impossible collision with the Earth. For this reason the perigee altitude constraint was introduced in the optimal strategy computation.

The same analysis was done for a LEO to MEO orbital transfer, studying the evolution of the ΔV mission cost as function of the delta-RAAN.

The Keplerian parameters are the following:

- LEO: $a = 7978\,\text{km}$; $e = 0$; $I = 56°$;
- MEO: $a = 29{,}601\,\text{km}$; $e = 0$; $I = 56°$.

The satellite and the propulsion system characteristics are

- Launch mass = 1625 kg
- Thrust = 0.180 N
- I_{sp} = 3829 s

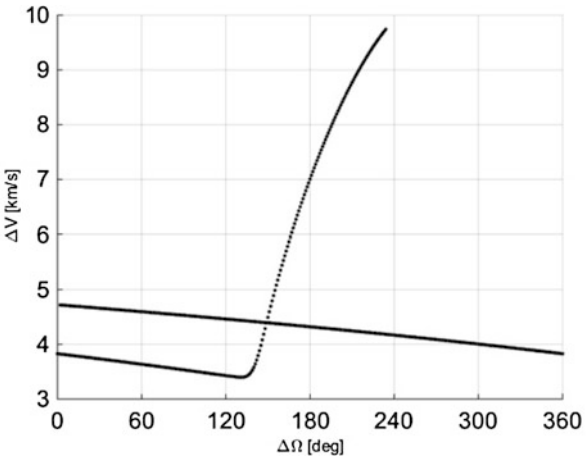


Figure 21 Evolution of the ΔV mission cost as function of the change of the delta-RAAN

As reported in [11], Figure 21 shows the evolution of the ΔV mission cost as function of the delta-RAAN. In this case too it is possible to note some areas where there is more than one solution with different ΔV, one higher than the other.

See [11] for more details.

4 Conclusions

The electrical propulsion application on the telecom satellites is going to be one of the most interesting topics of the international scenario. Many companies are investing many millions of euros in order to design and develop electrical satellites for future applications in space (telecommunication, Earth observation, navigation, etc.). Electrical propulsion means lower launch masses than the chemical systems, but implies very long orbital transfers to achieve the target orbit. In this context it is fundamental the computation of an optimal thrust strategy in order to minimize the orbital transfer time or the propellant consumption.

In this paper the authors have shown an optimal mathematical model based on an indirect method. This method was implemented in a code called *SOFTT* able to compute the optimal thrust strategy for every orbital transfer taking into account the perturbations, J_2, and eclipse.

Many numerical results are presented starting from simple orbital transfers arriving to solve more difficult problems such as constellation deployment.

A future interesting application of the optimal control theory on the orbital transfers is the possibility to implement an optimizer directly on board the satellite, this requires the capability to measure the satellite orbital status autonomously. Even if technical difficulties still exist to make such autonomous estimation on board, this research is well motivated by the cost of maintaining a LEOP ground segment operational during such long transfers.

The authors are now working to develop a "on board real time trajectory optimization" software, named *AUTONAV*. It will provide the possibility to design smart and autonomous satellites able to compute continuously the optimal trajectory from an estimated satellite orbital status. This approach shall enable for a new generation of electrical, smart, and autonomous satellites independent from the ground stations contacts and commands and will be particularly effective in the case of constellation deployments.

References

1. Chachuat, B.: Non Linear and Dynamic Optimization, from Theory to Practice. No. La-Teaching-2007-001, Ecole Polytechnique Federale de Lausanne, Lausanne (2007)
2. Clarke, F.H.: Functional Analysis Calculus of Variations and Optimal Control. Springer, Berlin (2013)
3. Conway, B.A.: Spacecraft Trajectory Optimization. Cambridge University Press, Cambridge (2010)
4. Ferrier, C., Epenoy, R.: Optimal control for engines with electro-ionic propulsion under constraint of eclipse. Acta Astronaut. **48**(4), 181–192 (2001)
5. Jo, J.W., Prussing, J.E.: Procedure for applying second order conditions in optimal control problems. J. Guid. Control. Dyn. **23**(2), 241–250 (2000)
6. Kechichian, J.A.: Analytic representations of optimal low thrust transfer in circular orbits. In: Conway, B.A. (eds.) Spacecraft Trajectory Optimization, pp. 139–177. Cambridge University Press, Cambridge (Chapter 6).
7. Kluever, C.A.: Low thrust trajectory optimization using orbital averaging and control parametrization. In: Conway, B.A. (eds.) Spacecraft Trajectory Optimization, pp. 112–138. Cambridge University Press, Cambridge (2010) (Chapter 5)
8. Mazzini, L.: Time open orbital transfer in a transformed Hamiltonian setting. J. Guid. Control Dyn. **36**(5), 1364–1374 (2013)
9. Mazzini, L.: Finite thrust orbital transfers. Acta Astronaut. **100**, 107–128 (2014)
10. Mazzini, L.: Flexible Spacecraft Dynamics, Control and Guidance. Springer, Rome (2015)
11. Mazzini, L., Cerreto, M.: Applications of optimal finite thrust orbital transfers. Aerotecnica Missili Spazio. J. Aerosp. Sci. Technol. Syst. **96**(1), 3–15 (2017)
12. Sackett, L., Malchow, H.L., Edelbaum, T.N.: Solar Electric Geocentric Transfer with Attitude Constraints: Analysis. NASA-CR-134927. pp. 112–137 (1975)
13. Sanders, J.A., Verhulst, F., Murdock, J.: Averaging Methods in Nonlinear Dynamical Systems. vol. 59, 2nd edn., Springer, New York (2000)

Collection Planning and Scheduling for Multiple Heterogeneous Satellite Missions: Survey, Optimization Problem, and Mathematical Programming Formulation

Snezana Mitrovic-Minic, Darren Thomson, Jean Berger, and Jeff Secker

Abstract This chapter introduces novel integrated management of multiple heterogeneous satellite missions for the purpose of intelligence collection. The focus is on optimization of acquisition planning and scheduling for various missions including single satellites and satellite constellations. The relevant optimization problem and its mathematical programming formulation that allow multiple area coverage plans for each acquisition request, as well as consideration of the quality measures of coverage plan, strip, and imaging opportunity, are presented. The chapter consists of a multi-mission planning system overview, a survey of relevant literature, a definition of the integrated acquisition scheduling optimization problem and its mathematical programming formulation.

1 Introduction

This chapter presents a novel optimization problem recently arising in the satellite industry. With the latest developments in Earth observation missions and the constant increase in the demand for remote sensing imagery, the need for integrated scheduling of multiple heterogeneous satellite missions is becoming eminent. The need is twofold: from the perspective of the organizations that operate an increased number of satellite missions and sell the imagery, and from the perspective of

S. Mitrovic-Minic · D. Thomson
MDA Systems Ltd., Richmond, BC, Canada
e-mail: snezanam@mdacorporation.com

J. Berger (✉)
DRDC, Valcartier Research Centre, Quebec City, QC, Canada
e-mail: jean.berger@drdc-rddc.gc.ca

J. Secker
DRDC Ottawa, Ottawa, ON, Canada

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_11

governments that own several missions and consider ordering access time windows or imagery from commercial satellite providers.

This chapter reports on an initial study of the integrated resource management of satellites and satellite constellations for intelligence gathering. The remainder of the document is organized as follows:

- This section presents the context and the motivation for the study.
- Section 2 provides a survey of relevant literature.
- Section 3 introduces the optimization problem statement for multi-mission acquisition scheduling.
- Section 4 introduces models for multi-satellite (multi-mission) acquisition scheduling.
- Section 5 provides conclusions followed by the appendix that contains a summary of the literature survey.

1.1 Motivation

The number of Earth observation satellites has been growing steadily, but at the same time the demand for satellite imagery has been increasing at even a higher pace. The satellite imagery is nowadays a crucial part of many areas of human life including meteorology, agriculture, environmental monitoring, energy explorations, map charting, ice monitoring, security, search-and-rescue, and disaster relief operations.

The first artificial satellite Sputnik 1 was launched on October 4, 1957, followed soon after, in 1960, by the first Earth observation satellite CORONA. In November 2017, there were 4635 satellites orbiting Earth [32], out of which 1738 were operational. Most operational satellites are either communication satellites (742) or Earth observation satellites (596). In terms of their altitude, more than half of the operational satellites (61.6%) are in low Earth orbit (LEO), including the International Space Station and the Hubble Space Telescope, orbiting the Earth at the distance of several 100–2000 km. Most of the other satellites (30.6%) are in the geostationary orbit (GEO), orbiting the Earth at the altitude of around 36,000 km.

Earth observation satellites most commonly acquire images using optical cameras or SAR (synthetic aperture radar) sensors. Image acquisition planning and scheduling was initially performed manually, but as demand for satellite imagery has increased, researchers and mission managers have successfully improved the utilization of single satellites and small satellite constellations by applying operations research and optimization methodologies to satellite operations.

Nowadays, many governments and commercial vendors manage more than one satellite mission or satellite constellation. For example, a commercial vendor Planet operates 191 satellites belonging to several missions. Overcoming separate order desks for each satellite mission and the silo-effect in their management would be beneficial to all parties. A centralized order desk and integrated management of

multiple-satellite missions would simplify operations, increase satellite utilization, and it would result in increased performance of intelligence collection. It would allow more imaging, satisfying customer needs faster, and potentially increasing overall operational performance by 25–40% [14, 30].

The imminent motivation for the preliminary study project, whose findings are presented in this chapter, arises from the need for:

- Integrated utilization of RADARSAT-2 and RCM constellation.
- Enhanced resource management of RCM (RADARSAT Constellation Mission).
- Enrichments for search-and-rescue decision support.
- Surveillance support for a variety of domains including vast Canadian territories in the North and in the Arctic.

1.2 Multi-Mission Planning System Overview

The multi-mission planning and scheduling system (MiMPS) allows a user to enter an order for satellite imagery through a centralized order desk. The user can specify explicitly which satellite or a set of satellites can be used for satisfying the order, or the user can provide the request specifications and the system would be given the freedom to select an appropriate available satellite to acquire the images. By avoiding several entries to different order desks, and by avoiding situations of being rejected due to already occupied resources, the user's needs would be satisfied faster and better. If the best satellite is available, it will acquire the image. If this satellite is not available, the second-best choice would be selected and the user will get their order fulfilled instead of rejected.

The MiMPS system mandate [28] includes:

1. Receiving the acquisition requests or intelligence, surveillance, and reconnaissance (ISR) tasks from a variety of users.
2. Scheduling the satellites to best meet the needs of users and the requirements of their image acquisition requests.
3. Generating optimized de-conflicted schedules for the satellites.
4. Utilizing the multi-mission satellite resources, commonly oversubscribed, in the best manner such that the mission constraints are satisfied.

The MiMPS system consists of the commercial satellite image acquisition planning system (CSIAPS) and the multi-mission schedule optimizer. The MiMPS context is shown in Figure 1. Planners and analysts from various organizations submit orders as intelligence, surveillance, and reconnaissance (ISR) tasks or as image acquisition requests through the multi-mission order handling system (MiOHS). Simple examples of ISR tasks are: "Extract shoreline for GIS foundation layers," "Detect ships in open ocean," and "Extract terrain slope and surface roughness for trafficability." The order priorities are consolidated in the MiOHS based on the user/organization priority and the order priority. The orders are then

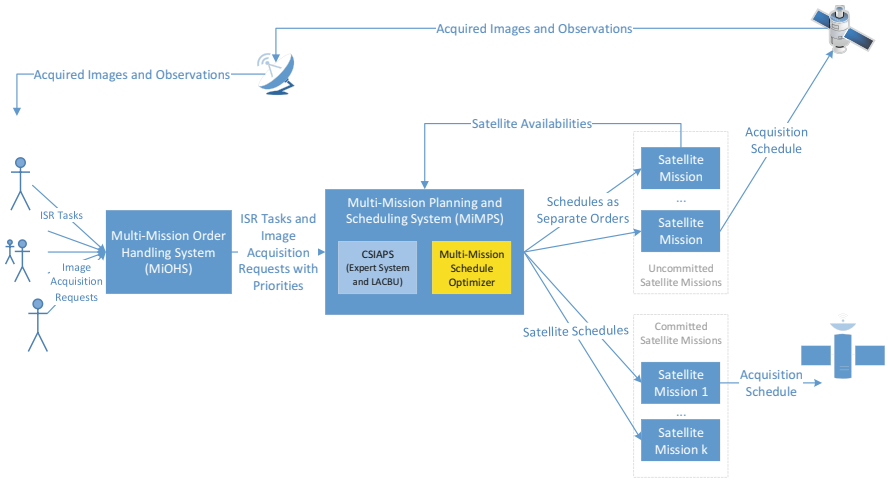


Figure 1 Multi-mission planning and scheduling system (MiMPS) context

sent to MiMPS where each ISR task is translated into image acquisition request(s) using the CSIAPS guidance expert system [33, 35]. On a regular basis, the MiMPS requests a report on the satellite availability from satellite missions. Based on these reports, the MiMPS generates coverage plans for the area of interest (AOI) of each image acquisition request. The AOI coverage plan is built by the CSIAPS' large area collection build-up tool (LACBU). The image acquisition requests and the corresponding AOI coverage plans are sent to multi-mission acquisition schedule optimizer to generate optimized de-conflicted schedules, one for each satellite.

The MiMPS system is designed to support different operational scenarios of satellite missions, and as such, differentiates between committed satellite missions and uncommitted satellite missions. The committed satellite mission is fully controlled by the MiMPS system. The entire acquisition schedule the MiMPS system generates will be sent and acquired by the mission's satellites. The uncommitted satellite mission will receive separate orders (image acquisition requests) generated from the MiMPS acquisition schedule. If the uncommitted satellite mission rejects the acquisition request, the MiMPS system will trigger re-planning and re-scheduling.

2 Literature Survey

This section provides a survey of scientific literature relevant to multi-mission planning and scheduling. It is a summary of the survey presented in Mitrovic-Minic et al. [28] and it includes manuscripts on image acquisition scheduling for a single satellite or a satellite constellation.

Most of the literature covers centralized scheduling for deterministic and static problems. The papers tackle:

- Optical satellites or SAR satellites.
- Single satellite or a satellite constellation.
- Single-orbit or multi-orbit scheduling.
- Scheduling horizons from an hour to a week.
- Imaging of spot targets and imaging of large polygons.
- One, two, or three sensors/cameras: for potential stereo-imaging.

As per Bensana et al. [3], the satellite image acquisition scheduling problem deals with serving observation tasks with satellite resources. Most often, it deals with selecting opportunities to acquire as many images as possible within a planning horizon while considering image priorities and requirements and satisfying satellite constraints. The problem can be categorized as a version of the multi-dimensional knapsack problem and as such it is NP-hard.

Solution approaches proposed for solving the image acquisition scheduling problem are:

- Heuristics including variety of greedy construction algorithms.
- Metaheuristics including simulated annealing (originally introduced in 1983), tabu search (originally proposed in 1989), genetic algorithms (initially proposed in 1995), and squeaky wheel algorithms (first presented in 1999).
- Constraint programming.
- Exact optimization methods.
- Bounding procedures.

Comprehensive surveys of image acquisition scheduling are provided in [6, 30, 38, 39]. Verfaillie's PhD thesis [39] contains an annotated bibliography of 110 papers on satellite scheduling. The problems include: ground planning of Earth observation satellite activities (40 papers), onboard planning of Earth observation satellite activities (31 papers), cooperation of ground and onboard planning (3 papers), planning of Earth observation satellite constellation activities (25 papers), fair sharing of the use of Earth observation satellites (6 papers), and six PhD theses. Verfaillie [38] is a tutorial presentation on scheduling optical and radar satellites prepared by ONERA (French Aerospace Lab) research group.

Several *exact algorithms* have been proposed. Bensana et al. [3] reported on scheduling for French satellite SPOT—an electro-optical satellite with multi-spectral capability and three cameras that have to be scheduled in parallel for some requests. The requests included mono images and stereo images. Gabrel and Vanderpooten [12] formulated the SPOT scheduling problem as a multi-criteria path problem over an acyclic graph. Their algorithm was an adaptation of the label-setting shortest path algorithm for generating all efficient paths, followed by an interactive session for selection of the best sequence. The optimization criteria included demand satisfaction, maximization of the number of high-priority shots, and minimization of satellite use to prevent wear. For one satellite over four orbits, the algorithm run time was less than an hour. Hall and Magazine [16] proposed a

dynamic programming algorithm with a bounding procedure by either Lagrangian relaxation or relaxation of some constraints. They solved instances with 200 targets.

A selection of the most efficient *heuristics* and *metaheuristics* is presented in the following paragraphs.

Nelson [30] in her PhD thesis proposes a column generation approach for solving the image acquisition scheduling problem for a satellite constellation for a given set of spot targets. The approach is a combination of exact methods (integer and linear programming) and heuristics, and it contains three steps: cluster, route, and schedule. The clustering step groups imaging targets; the routing step uses column generation, where each satellite and each orbit are treated as separate subproblems to determine which clusters would be imaged by which satellite; and the scheduling step schedules each satellite separately using time–space networks and heuristics.

The radar imaging satellites are considered with two-dimensional electronic scanning array (ESA). Moving from one target to another within the ESA's field of regard (EFOR) is almost instantaneous. Dwell time is the time needed to image a target. It can also be called the processing time of the imaging task. The SAR satellite constellation considered in this research has ten SAR imaging satellites in five equally spaced orbital planes. The satellite orbits are circular at an altitude of 1000 km and an inclination of 60°. The orbit time is 104 min. An average time window within which a target has to be imaged is 3 min.

Each satellite has limited power (20 min of imaging time per orbit) and different resolution modes. Higher resolution modes require longer imaging time. The satellite can image to the left or to the right from its track. The time between imaging two targets within the same EFOR is zero. The time required to image the target is randomly assigned to between 5 s and 45 s. The problem instances included 8000 targets and scheduling horizons of two and four orbits. Computational study results state that 768 clusters were formed for 4200 targets in 8.5 min. Column generation took 108 min. The time–space network generated for scheduling with 5-s time increments resulted in a mixed-integer programming formulation with ten million variables and 45,000 constraints. Scheduling of all four revolutions took 47 min and 328 targets were scheduled, almost all high priority with only two medium and one low.

In addition, Nelson [30] provides comprehensive literature survey including the following excerpts: Verfaille et al. [18] and Lemaitre [40] proposed advanced versions of constraint-programming algorithm. Pemberton [31] recognized that over-constrained scheduling problems, such as the satellite scheduling problem, can be challenging for constraint-programming methods. Wolfe et al. [43] defined the problem as a window-constrained packing problem. Bianchessi et al. [4] and Cordeau and Laporte [5] achieved near-optimal solution using a tabu search algorithm for scheduling optical satellites. The infeasible solutions were also explored by relaxing time window constraints. Muraoka et al. [29] proposed complex ranking function consisting of 12 elements, [31] used priority segmentation and solved each subproblem to optimality by branch and bound, and Frank et al. [10] proposed greedy hill-climbing with stochastic variations to escape local optima and with ranking that used priority and contention. Globus et al. [14] and Globus et al. [15] compared 13 variations of genetic algorithm, simulated annealing, stochastic hill-

climbing, squeaky wheel, and iterated sampling on optical satellite constellation and concluded that the simulated annealing performed the best. Two solved problems are scheduling a single satellite with a cross-track slewable sensor and scheduling a two-satellite constellation with the similar sensor. The schedule horizon is 1 week. Lin et al. [21] decomposed the problem into subproblems using Lagrangian relaxation, solved the subproblems by linear search, and generated a feasible schedule by a greedy heuristic. Lin and Chang [22] proposed a hybrid algorithm which included Lagrangian relaxation with a tabu search based feasibility adjustment.

Lamaitre et al. [19] researched scheduling of agile Earth observing satellites, including French Pleiades. Most often, an agile satellite has a non-mobile onboard sensor and the whole satellite maneuvers for image acquisitions on two or three axes (roll, pitch, and yaw). The authors analyzed the scheduling problem of imaging spot targets and large polygon areas. A piecewise-linear function is introduced for measuring the satisfaction of a request with large polygon area. The memory and energy constraints are considered. The scheduling horizon is one orbit and there is one opportunity for each image. For solving the track selection and scheduling problem, the authors propose: greedy algorithm, dynamic programming algorithm, constraint programming, and local search algorithm (with insertion and removal moves). The algorithms were tested on six real-world problem instances with 70–375 requests (within an orbit). There were 12–87 stereo-images, 17–346 spot targets, and 4–133 polygon targets. There were in total 212–1068 opportunities for candidate images, with 5–9 s average imaging time. Stopping criteria for the algorithms was 2 min, and the local search heuristic achieved best performance.

Systems toolkit (STK) [8, 9] has a component for satellite-task scheduling called STK scheduler. The tool implements two algorithms: a sequential algorithm and a neural algorithm. The sequential algorithm is a greedy algorithm with backtracking capabilities and with selectable ranking functions. The neural algorithm, inspired by neural networks, assigns a probability to each satellite-task assignment and changes the probability iteratively with the aim of increasing the objective function.

Bianchessi et al. [4] proposed a tabu search metaheuristic algorithm for solving a multi-satellite multi-orbit image acquisition scheduling problem for optical agile satellite. The authors modeled the French satellite constellation Pleiades, which was launched in 2008. Each imaging request has a priority and belongs to one user group. The imaging area of a request may consist of several equal-width strips. Each image strip is of a varying length that determines the duration of imaging. Sometimes the strips of one request have to be acquired one after another (in time) and one next to each other (in space)—without acquiring another strip in between. Mono and stereo images are also considered. The stereo-image request is considered satisfied only if both images are taken. The objective function is a weighted sum of the normalized utilities associated with the different users. The utility of a user is defined as the sum of the profits associated with the (possibly partially) satisfied user's requests. Order weighted average is used to ensure fairness. The objective function is linear with respect to the proportion of the polygon's area being acquired as opposed to the piecewise-linear convex used in Cordeau and Laporte [5] and Malladi et al. [23].

In addition, one direction of acquisition is allowed for each strip instead of two (forward and backward) as in Cordeau and Laporte [5].

The tabu search metaheuristic the authors proposed has the capability to explore infeasible solutions during the search. The time window constraints are relaxed and relevant penalties are added to the objective function. The multiplier of the time window violations is a self-adjusting parameter. Six types of moves define the neighborhoods, and several elaborate intensification and diversification procedures are proposed.

A computational study was done on 13 problem instances provided by CNES (Centre National d'Etudes Spatiales). The schedule horizon was 24 h long and there were four user groups. The instances have 3000–4000 requests, with 100–120 high-priority requests, around 2500 twin requests, and around 500 polygon requests with around 2000 strips. The stopping criteria for the algorithm was the 2-h running time, and the solutions achieved were within 3% of the upper bound, for most of the instances.

Hwang et al. [17] proposed a genetic algorithm for scheduling multiple agile satellites. The authors also provided ways to calculate opportunity time windows and a function for calculating the imaging time using Cauchy partial differential equations (PDF) instead of Gaussian PDF. The initial scheduling problem is divided into subproblems based on satellite orbits. The uncorrelated parallel machine scheduling problems are solved where each single track (orbit) is a machine. The target assignment to tracks is performed by mission track coordination.

Dishan et al. [6] propose a rolling-horizon based heuristic for solving the dynamic scheduling problem for agile satellites. The heuristic is greedy where ranking function includes dynamic problem aspects, including waiting time. The authors also propose a mixed-integer programming (MIP) formulation of the problem. Each request is a strip and has a priority, imaging duration, roll angle and pitch angle, arrival time, and a deadline. Each strip has several opportunities. The schedule horizon has 24 h, and the rolling horizon is 2 h long. The computational study compared different ranking schemas on 12 problem instances with 4–6 satellites and 100–400 requests.

Wu et al. [44] propose adaptive simulated annealing aggregated with dynamic task clustering strategy. The major contributions of the paper include: adaptive temperature control, tabu-list based revisiting avoidance technique, and the intelligent combination of two neighborhoods. The "adaptive" refers to the choice between two neighborhood structures: the swap (insertion and removal) neighborhood, and the task migration neighborhood. The task migration neighborhood attempts to move a scheduled task that has multiple opportunities to another orbit. The choice between the two neighborhoods is based on the probabilities of the success of each neighborhood in previous iterations. A computational study was performed on two groups of problem instances: "The first group could represent the normal situation. The second group reflects some emergent situations such as an earthquake, forest fire, flooding, and military conflict." The problem instances include four satellites, scheduling horizon of 1 day, and 100–1000 target requests. Run times were from 2 min to 1 h, for small and large problem instances, respectively.

There were 200–2500 opportunities and 5–30% of targets were not scheduled. The proposed algorithm achieved the best performance compared to typical ant colony optimization, tabu search metaheuristic, genetic algorithm, and a highest-priority first schedule algorithm.

Augenstein [1] describes a decision-support system for scheduling Skybox agile optical satellites that can slew. The author proposes modeling the problem as a maximum weight longest path in a precedence (sequence) graph of the targets (images) that can be imaged by the satellite. Each target has a priority, an imaging time, and one opportunity to be imaged. The schedule horizon is one orbit. The problem is polynomially solvable when the precedence graph is acyclic. The interactive aspect of the solver consists of the capability that a user can force an insertion of an image in the schedule or a removal of an image from the schedule.

Xiaolu et al. [45] propose a decomposition algorithm for scheduling multiple agile satellites that can roll. The requests are for imaging spot targets and the problem is modeled as observation scheduling. Each observation images a strip that can contain several merged spot targets. The authors divide the problem in task assignment and task merging. The task assignment deals with assigning a task to one of the potential observation windows (strips). The task merging merges the tasks assigned to one observation window for each satellite and each orbit separately. The algorithm alternates between the task assignment and the task merging until the stopping criteria is met. The task assignment is solved by ant colony optimization. The task merging, modeled as the maximal covering location problem, is solved by a dynamic programming algorithm. The authors also provide the mixed-integer programming (MIP) formulation of the observation scheduling problem that includes energy and memory considerations as well as maximal continual imaging time of the satellite.

For the computational study, a series of problem instances were generated inspired by the Solomon's construction of the problem instances for the vehicle routing problem with time windows (VRPTW). Three distributions were considered: uniform, collective (clustered), and mixed. There are 63 problem instances with 100–600 imaging requests and 2–8 satellites. There are 200–4300 imaging opportunities. The proposed decomposition algorithm is compared with a simulated annealing algorithm developed earlier by the authors. The new algorithm achieved considerable improvements of around 25% and up to 50% for some instances. The algorithm has been used in a satellite scheduling software that has been operational in the China Satellite Management Center.

Tangpattanakul et al. [34] propose iterated indicator-based multi-objective local search (IBMOLS) algorithm for scheduling image acquisitions of agile satellites. The objective is to maximize profit and to simultaneously ensure fairness. The fairness is represented as minimizing the maximum profit difference between the users. The satellite can roll, pitch, and yaw. The imaging targets are spot targets and rectangle targets. They are all transformed into strips, where a spot target consists of one strip and a rectangle can consist of several strips. There are also stereo requests. The gain part of the profit function for a rectangle request is a convex piecewise-linear function and it depends on the acquired area.

The IBMOLS is an iterated search algorithm that in each iteration generates a population P of solutions from which a set A of non-dominated solutions is selected. Within each iteration, the following process is repeated as long as A keeps changing: the local search is applied to each solution from P and the non-dominated set A is updated. When A remains unchanged, the iteration is complete and the set PO of Pareto optimal solutions is updated by adding solutions from A. Then a next iteration of the IBMOLS algorithm starts with the creation of new P set generated partially from PO set by removing ¾ of the scheduled strips from each solution. The algorithm stops when a stopping criterion is satisfied. The local search step generates a neighborhood of each solution in P and when a better solution is found it replaces the worst solution in P. The best improvement strategy is used as opposed to the first improvement strategy used in an earlier study by the same authors. The algorithm was tested on the modified instances of ROADEF 2003 challenge. The proposed algorithm is compared to the random-key genetic algorithm proposed in the authors' earlier paper. Most of the results show that the IBMOLS algorithm obtains better solutions within a shorter running time.

Wang et al. [41] propose an exact optimization algorithm and several heuristic algorithms for solving image acquisition scheduling problem for optical satellites with uncertainty of clouds. The paper also provides a novel non-linear mathematical programming model. The satellite can slew (roll), but it cannot pitch. The allocation of a task to multiple resources is allowed to achieve a higher probability of acquiring an image without a cloud cover. The problem is divided into a master problem and multiple subproblems—one for each satellite and orbit. Based on the solutions of the subproblems, all feasible solutions of the master problem are listed and an optimal solution is selected. The non-linear profit function is transformed into a piecewise-linear convex function of the acquired surface. The authors consider the memory and the energy capacities for each orbit to be constants. For the practical-size problem instances the authors propose five heuristics for solving the subproblems and generating the set of feasible solutions out of which in the master problem one solution will be selected.

Experiments by simulation on three satellites were conducted to verify that the proposed method performs better than previous studies under uncertainties of clouds. The authors used 90 medium-size problem instances and 60 large-size problem instances. The medium-size instances have up to 60 tasks in nine orbits and up to 30 tasks in 21 orbits. The large-size instances have up to 160 tasks in 21 and 42 orbits. The achieved results were compared to the CPLEX-based approach proposed by Liao and Chang [20].

Malladi et al. [23, 24] studies deal with scheduling RCM (RADARSAT Constellation Mission) that is a satellite constellation of three identical non-agile SAR satellites. Each satellite can image only to the right of its nadir. The satellite can image using one of the several modes characterized by different swath widths and resolutions. For certain modes, the satellite can make images with one of the several different incidence angles. The time satellite needs to switch between two incidence angles is almost instantaneous. Malladi et al. [23] propose a mixed-integer programming (MIP) model and solve the problem using CPLEX. Malladi et al.

[24] solve the problem using graph-based models and a model-based metaheuristic (matheuristic). Each request is characterized with a geographic area, priority, mode, repeat cycle, and a deadline. Each large area is divided into strips. The width of each strip depends on the mode of the request. Each satellite may have several opportunities for imaging a strip. Each opportunity has a duration and an incidence angle. In the schedule, each strip has to be acquired at most once.

The objective function is a piecewise-linear function. Two versions of the piecewise-linear function are tested. A computational study was performed using generated problem instances based on the statistical analysis of potential practical yearly orders. Encouraging results have been achieved. The computational study used seven problem instances with the scheduling horizon of 1 day. Optimal solutions were generated within several minutes. The number of strips was between 1300 and 10,000.

The computational study also indicated that the energy and thermal constraints are not tight nor bounding. Thus, a further research study was conducted on the relaxed scheduling problem. The relaxed problem is a special novel version of the maximum clique problem, called the cluster-restricted maximum weight clique problem [24]. A compatibility graph is devised that captures compatibilities between the pairs of opportunities. Each opportunity is a node of the compatibility graph, and an edge between two opportunities indicates that the pair is not in conflict. Furthermore, all opportunities of one strip represent a cluster and there is no edge between two nodes in one cluster. The acquisition scheduling problem is thus converted into the problem of finding the special maximum weight clique in the compatibility graph. The computational study using the proposed mixed-integer programming (MIP) model and the model-based metaheuristic (matheuristic) was performed on the problem instances generated by an adaptation of well-known DIMACS and BOSHLIB benchmark instances for the maximum weight clique problem. There are 80 DIMACS instances and 36 BOSHLIB instances with between 400 and 1500 nodes. The nodes were randomly partitioned into 25 clusters. The proposed model-based metaheuristic achieved encouraging results when compared to CPLEX.

3 Optimization Problem Statement

3.1 Context

3.1.1 Problem Type

We assume that the acquisition scheduling problem is deterministic and centralized. Both static and dynamic environments are considered. Dynamic environment is handled by repeated scheduling sessions whose start is triggered by arrival of request(s). The dynamic environment allows for dealing with near real-time tasking, urgent requests, adaptive sensing, re-tasking, and re-planning.

3.1.2 Satellite Types

We consider the heterogeneous satellites and satellite constellations for Earth observations. In terms of agility, we consider agile and non-agile satellites. An agile satellite can yaw, pitch, and roll (slew). Pitching allows the satellites to look forward and backward along their direction of travel. Rolling allows the satellite to look left and right, perpendicular to their direction of travel. Yawing changes the heading of the satellite, although not necessarily their direction of travel. The time needed to change the satellite (sensor) positioning between imaging two strips is called the maneuver time.

A satellite that cannot yaw, nor pitch, nor roll is a non-agile satellite. A sensor on a non-agile satellite produces an image as satellite moves along its track. The starting time of any candidate image is the time at which the satellite flies over the beginning of the area to be acquired. Consequently, the compatibilities between the images and the opportunities can be pre-computed.

A satellite generates energy using solar panels, stores it in batteries, and uses it for maneuvering, imaging, and communicating with ground stations.

Some non-agile satellites can image using different incidence angles (Figure 2). A non-agile satellite with a SAR (synthetic aperture radar) sensor is typically capable of imaging in many different beam modes. A beam mode is characterized by a resolution, field of view, noise characteristics, and the number of beams that need to be activated to generate an image. For certain beam modes, only a subset of beams available on the SAR sensor need to be activated. Thus, by activating two different subsets of the satellite beams, two different incidence angles can be used to make an image. The change of the incidence angle represents the change of the subset of active beams and it is almost instantaneous. The change of incidence angle results also in the change of geographic footprint the satellite can image in its pass.

3.1.3 Sensor Types

Each Earth observation satellite has a sensor of a certain type including optical camera, SAR (synthetic aperture radar), infrared, multi-spectral, and hyper-spectral. The sensors are able to acquire images with various modes, resolutions (from hundreds of meters to less than 1 m), and different swath widths (from several kilometers to several hundreds of kilometers). For the purpose of acquisition scheduling, in this chapter we assume that each satellite has a single sensor.

Most of optical cameras and SAR sensors generate a one-dimensional image at each time instant. A two-dimensional image is created by imaging for a specified duration of time while the satellite passes over the area. This imaging duration is called imaging time, or *observation duration*, or dwell time. The time window when the imaging can be performed is called an observation time window, visibility window, or an *opportunity time window*. It is the time interval within which the sensor "sees" the target. For a non-agile satellite, with a one-dimensional imaging device, the observation duration is equal to the length of the opportunity time window.

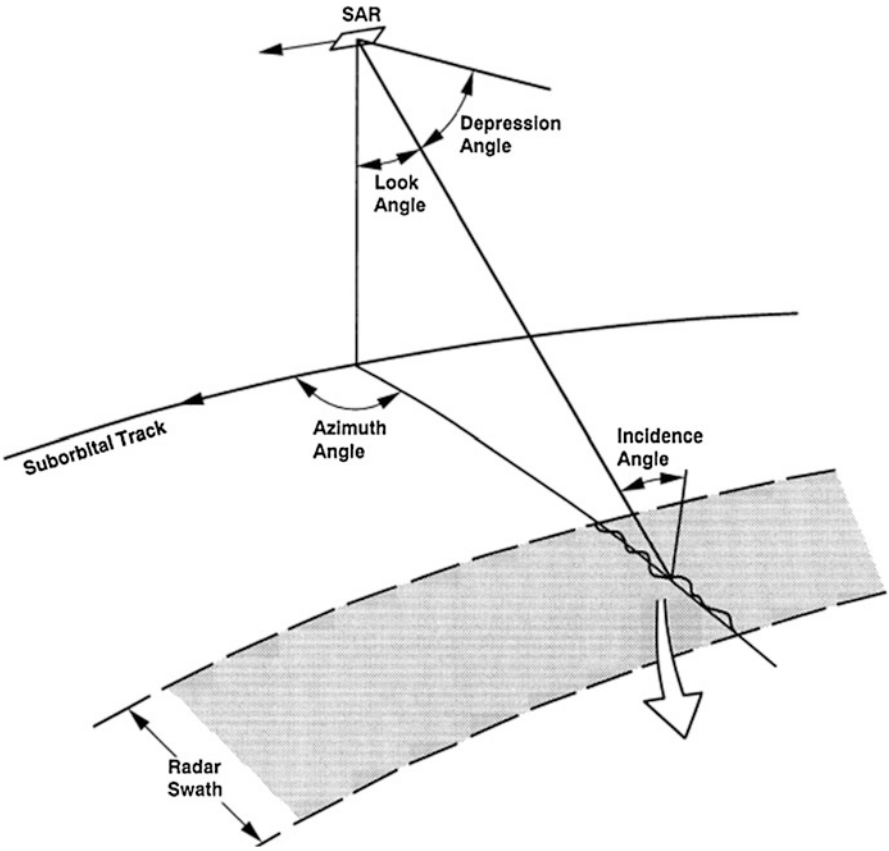


Figure 2 Look angle and incidence angle [7]

For an agile satellite, the observation window can be anywhere within the opportunity time window. Therefore, there may be several observation opportunities for imaging a strip within one satellite pass. If the satellite maneuvering is sufficiently fast, the agile satellite may even allow stereo observations within one orbit (Figure 3).

Most often in image acquisition scheduling, approximation(s) of maneuver time between two successive observations is used, since the maneuver (transition) time is non-linear and depends on the time at which the transition is initiated (Figure 4).

3.2 Multi-Mission Collection Scheduling

Multi-mission collection scheduling deals with scheduling of satellite resources to satisfy the acquisition requests and satellite constraints and to optimize pre-specified optimization criteria. More precisely, for a given:

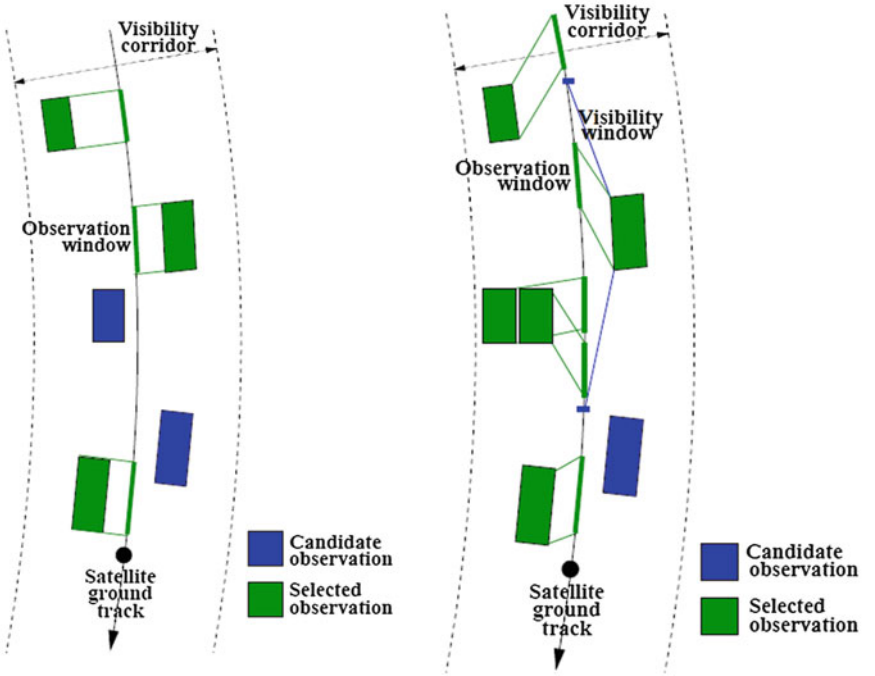


Figure 3 Non-agile satellite vs. agile satellite: opportunity time window and observation duration [38]

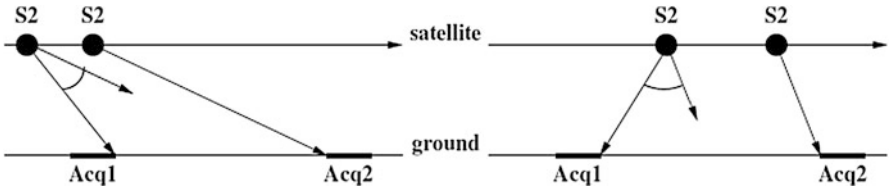


Figure 4 Maneuver time between two imaging targets depends on the time at which the maneuver is triggered [38]

- Set of intelligence, surveillance, and reconnaissance (ISR) tasks with specific requirements.
- Set of image acquisition requests with specific requirements.
- Set of satellites (satellite constellations) and satellite sensor assets with specific performance capabilities.
- Collection tasking objectives and constraints.
- Scheduling horizon.

The *Multi-Satellite Collection Scheduling Problem* determines the satellite schedules that serve the requests, optimize the collection scheduling objectives and

asset utilization, and satisfy the constraints. Solving the multi-mission collection scheduling problem consists of the following steps:

- Step 1: Translation of the ISR imaging task to image acquisition request(s), including the compatibility evaluation between sensor capabilities and ISR task requirements.
- Step 2: Generating the coverage plans, for a given list of satellites, for the geographic area of interest (AOI) to be imaged as defined in the image acquisition request.
- Step 3: Generating schedules by solving the integrated multi-mission acquisition scheduling problem.

This chapter focuses mostly on Step 3—solving the optimization problem dealing with integrated multi-mission image acquisition scheduling. This scheduling problem is solved by the multi-mission schedule optimizer system (Figure 1). The first two steps are done by CSIAPS (commercial satellite image acquisition planning system). The first step is performed by the CSIAPS guidance expert system that translates an ISR imaging task to one or more image acquisition requests [33, 35]. The second step is done by the CSIAPS large area collection build-up tool (LACBU) that generates the AOI coverage plans.

3.3 CSIAPS

The commercial satellite image acquisition planning system (CSIAPS) has the following capabilities:

- Automatically selects the best satellite(s) and sensor mode(s) for a specified ISR imaging, based on the knowledge of subject matter experts that has been encoded into inference rules and ranking algorithms.
- Creates an AOI coverage plan for the automatically selected set of satellites.

The first capability is supported by the CSIAPS guidance expert system. The second capability is supported by the CSIAPS LACBU tool in the following manner:

1. For an AOI and a selection of satellites (provided by the CSIAPS guidance expert system or by an intelligence analyst), CSIAPS generates all the opportunities that represent the intersection of satellites' swaths and the AOI. Only compatible satellites and satellite modes are used. The number of opportunities can be in the thousands.
2. From the generated pool of opportunities, the CSIAPS LACBU tool selects the opportunities to generate the AOI coverage plan. The current algorithm is an iterative approach that repeatedly selects the strip and opportunity that covers the most uncovered area of the AOI until the AOI is covered to a given tolerance (e.g., 98%). The coverage plan can consists of the strips from more than one

satellite. All opportunities that comprise a coverage plan are designed to be strips. For each strip, the corresponding opportunities are selected and marked as the opportunities to image the strip.

The CSIAPS can generate several AOI coverage plans for one ISR task or image acquisition request. For example, one coverage plan could be derived from only the ascending orbits, another from only the descending orbits, and a third derived from the opportunities that are weighted by their collection date and time.

In MiMPS, CSIAPS pre-processes the ISR tasks one by one and generates the set of image acquisition requests with AOI coverage plans. Then, the multi-mission acquisition schedule optimizer system generates a schedule for each satellite from the multi-mission systems optimizing the pre-specified optimization criteria.

3.4 Multi-Mission Image Acquisition Scheduling

This section focuses on the models and algorithms for the multi-mission (satellite) image acquisition schedule optimizer system (Figure 1) whose purpose is to generate acquisition schedules for the set of given satellites that serve acquisition requests, optimize the asset utilization, and satisfy the existing constraints. An *image acquisition request* is characterized by the following specifications:

- Area of interest (AOI)—the geographic area with location, size, and shape.
- Request time window within which the AOI has to be imaged.
- Request priority: priority of the user and importance of the task.
- Satellite(s) or sensor(s), potentially with preference values.
- Resolution(s), mode(s), polarization(s), potentially with preference values.
- Range of incidence angles.
- Look direction(s), for the case of satellite that can be left-looking or right-looking.
- Pass direction: ascending pass (from South to North) and/or descending pass (from North to South).
- Repeated imaging, or a standing order (e.g., daily, weekly repeat cycle).

Initially, each standing order is converted to a set of image acquisition requests whose time windows do not overlap and satisfy request repeat cycle. Further, according to the request specifications, CSIAPS builds the AOI coverage plan(s). Thus, the input to multi-mission acquisition schedule optimizer system contains at least one AOI coverage plan for each image acquisition request. Each coverage plan consists of a set of strips that may overlap (Figure 5). Each strip is a portion of the satellite swath—the footprint the satellite can image. The strips of one coverage plan may belong to different satellites (depicted in Figure 5 by different colors). Further, each strip can have one or more opportunities to be acquired. Each opportunity allows for the acquisition of the entire strip and it satisfies the request specifications.

Therefore, the multi-mission image acquisition scheduling problem deals with selecting the opportunities that would acquire as many images as possible within the

Figure 5 An AOI coverage plan with overlapping strips

schedule horizon while considering the acquisition request priorities and constraints. The problem is a selective scheduling problem since the satellite acquisition scheduling is most often oversubscribed and it is not possible to satisfy all the requests. The *schedule horizon* is the duration of the schedule. The resource constraints for a satellite include: (a) energy/fuel capacity, (b) thermal constraints (instrument temperature) that often translates to the maximal continuous imaging duration, (c) memory/data storage capacity, and (d) the maximal imaging time per orbit.

Processing and exploitation of satellite imagery is often sensitive to the particular imagery characteristics. As a result, for a single request only one AOI coverage plan should be selected to be acquired. Potential dependencies between the coverage plan strips further stress the need for that selection. For example, all strips in one coverage plan are on ascending passes of a satellite.

The number of opportunities to acquire a strip depends on the request time window and on the agility of the satellites compatible with the request specifications. The following often holds:

- Non-agile satellite: There is at least one opportunity to acquire a strip in one revisit cycle of a satellite. The revisit cycle is the duration of time the satellite needs to come over exactly the same point on Earth. The revisit cycle for a satellite is typically several days or few weeks. For example, the revisit cycle time for RADARSAT-2 satellite is 24 days.
- Agile satellites: In one satellite orbit there could be many opportunities to acquire a strip. The satellite agility allows many different imaging start times within the opportunity time window. There could also be several orbits within the satellite revisit cycle in which a strip could be acquired.

Each AOI coverage plan can be assigned a quality measure indicating how well the plan meets the request specifications. It may include the following aspects: (a) compatibility or preference of the selected satellites to the request specifications,

(b) the percentage of the covered AOI, and (c) the amount of overlap among the strips. Each strip of the AOI coverage plan is characterized by the satellite, the geographic footprint (portion of a satellite swath), and a quality measure of the strip (for example, indicating which portion of the AOI the strip covers). Each opportunity of a strip is characterized by the time window within which the strip acquisition can start and the quality of the opportunity. The opportunity quality may indicate how close the opportunity incidence angle is to the preferred incidence angle defined in the request specifications. Note that the geographic footprint of each opportunity is equal to the geographic footprint of the strip.

Further pre-processing could be considered to divide the strips at the strip intersections to reduce the size of the imaging strips that are in conflict.

4 Models for Multi-Mission Acquisition Scheduling

This section provides a mathematical programming formulation for the optimization problem dealing with acquisition scheduling for multiple heterogeneous satellite missions. The model is a mixed-integer programming (MIP) formulation and it models integrated management of agile and non-agile satellites. First, the notation and the decision variables are listed, followed by the optimization criteria and the MIP model.

4.1 Notation and Decision Variables

This section provides notation and decision variables used for the MIP model.

4.1.1 Notation

H planning horizon [0, T]
Γ set of non-agile satellites
Π set of agile satellites
R set of image acquisition requests
p_r priority of image acquisition request r
$[R_r, D_r]$ time window of request r. It is the time interval within which the request has to be acquired. The opportunities for the strips associated with request r are within the request time window
Φ^r set of the AOI coverage plans for request r
A^r area size of the AOI of request r
A_c^r sum of the areas of all strips of AOI coverage plan c of request r. Note that this may be larger than the total area $A^r$of the AOI of request r since the strips may overlap
A_j area size of strip j

φ_c^r quality measure on how well the specifications of request r are satisfied with the coverage plan c

I set of all strips of all requests in R

I_c^r strips that belong to the AOI coverage plan $c \in \Phi^r$. $\left|I_c^r\right|$ is the number of strips in the plan

I^r set of all strips of request r. $I = \bigcup_{r \in R} I^r$, $I^r = \bigcup_{c \in \Phi^r} I_c^r$.

P set of all opportunities within H for all strips in I of the requests in R

T_s set of all opportunities of satellite s for all strips of requests in R. Thus, $P = \bigcup_{s \in \Gamma \cup \Pi} T_s$

P_j set of all opportunities of strip j

P_j^s set of all opportunities to image strip j by satellite s, $P_j^s = P_j \cap T_s$

P^r set of all opportunities of request r. $P^r = \bigcup_{j \in I^r} P_j$, $P = \bigcup_{r \in R} P^r$

T_{si-} set of opportunities of satellite s whose time window is after, or overlap with, the time window of opportunity i

T_{si+} set of opportunities of satellite s whose time window is before, or overlap with, the time window of opportunity i

T_s^k set of all opportunities of satellite s in orbit k for all strips of requests in R

c_i^s cost of acquiring (imaging) the strip associated with the opportunity i using satellite s

d_i^s duration of imaging the strip associated with the opportunity i using satellite s

$[E_i, L_i + d_i^s]$ time window of opportunity i is the time interval within which the associated strip could be acquired (Figure 6)

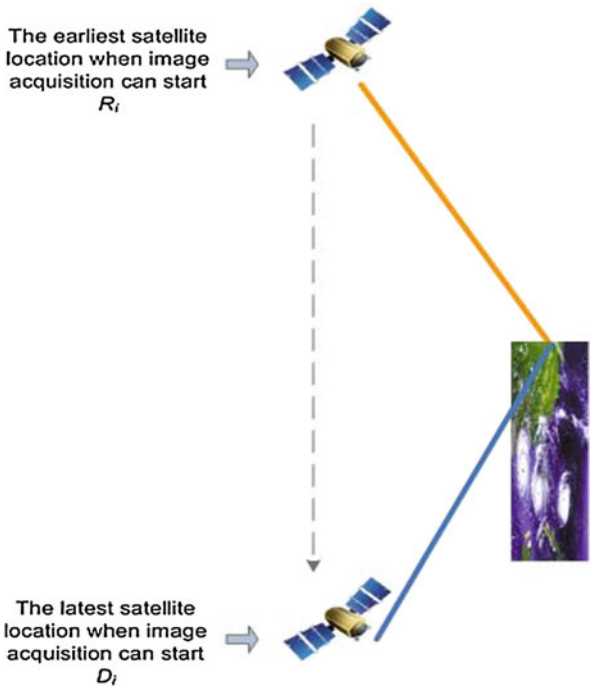


Figure 6 The opportunity time window $[E_i, L_i + d_j]$

q_{ir}^s opportunity quality indicating how well request r specifications are satisfied if opportunity i is scheduled

C set of pairs of the opportunities that are in conflict and cannot be both imaged by one satellite

π_r total acquired area of request r. In the case of a dynamic scheduling problem, this may include the area scheduled in the current scheduling session and an area π'_r scheduled in the previous scheduling sessions

τ_1 maximal imaging time allowed for a satellite during an orbit

τ_2 average acquisition time per orbit allowed for a satellite over the planning horizon

κ^s number of orbits of satellite s in the planning horizon

t_{ij}^s maneuver time of satellite s required between the end of imaging strip associated with opportunity i and the beginning of imaging strip associated with opportunity j

0^s,$n^s + 1$ initial and final position (look angle) of the camera head for satellite s. Since the schedule time horizon is [0, T], then the time window of the initial camera head position [E_{0^s}, L_{0^s}] is [0,0] and the time window of the final camera head position [E_{n^s+1}, L_{n^s+1}] is [T,T]

β^s power usage per 1 s of imaging activity for satellite s

γ^s power usage per 1 s of maneuvering time for satellite s

Ξ^s total power available within the schedule horizon

δ_j^s memory consumption per 1 s of imaging opportunity j by satellite s

Ψ^s total memory available within the schedule horizon on satellite s

ω_1, ω_2 the objective function weights

4.1.2 Decision Variables

The decision variables for the proposed MIP model are:

$$z_c^r = \begin{cases} 1, \text{ AOI coverage plan c is selected to satisfy request } r \\ 0, \text{ otherwise} \end{cases}$$

$$y_i^s = \begin{cases} 1, \text{ if opportunity } i \text{ of satellite } s \text{ is scheduled} \\ 0, \text{ otherwise} \end{cases}$$

$$x_{ij}^s = \begin{cases} 1, \text{ opportunity } i \text{ is scheduled immediatelly before opporunity j on satellite s} \\ 0, \text{ otherwise} \end{cases}$$

$u_i^s =$ start time of strip acquistion associated with opportunity i on satellite s

4.2 Optimization Criteria

This section presents an initial set of possible optimization criteria. The problem is solved as single-objective optimization problem where the objective function is a weighted sum of selected optimization criteria.

4.2.1 Request Fulfillment, Priorities, and Image Quality

For the simplicity, assume that there is one AOI coverage plan for each request. The optimization criteria could be to maximize the utility of the acquired images with respect to the image acquisition requests. The utility for each request is a product of the *request priority* and the *proportion of the acquired area* [23]:

$$Max \sum_{r \in R} p_r * f\left(\frac{\pi_r}{A^r}\right) \tag{1}$$

where π_r is the area acquired and A^r is the total area of all the strips of the AOI coverage plan. Function f can be $f(x) = x$ or an increasing piecewise-linear function. The latter *encourages acquiring* the strips of already partially acquired requests [23]. When π_r is expanded:

$$Max \sum\nolimits_{r \in R} p_r * f\left(\frac{\sum_{j \in I^r} A_j \sum_{i \in P_j} y_i}{A^r}\right) \tag{2}$$

The *quality of an opportunity* q_{ir}^s can encompass the following factors: (1) how well the satellite sensor capabilities satisfy the request specifications, and (2) how well the mode and the geometry parameters associated to the opportunity satisfy the request specifications. Thus, the expanded optimization criteria is

$$Max \sum_{r \in R} p_r * f\left(\frac{\sum_{j \in I^r} A_j \sum_{s \in \Gamma \cup \Pi} \sum_{i \in P_j^s} q_{ir}^s y_i}{A^r}\right) \tag{3}$$

Since an opportunity has the same geographic footprint as the strip, the strip opportunities on non-agile SAR satellites are most often associated to the satellites from a same constellation (that are homogenous and follow the same orbits). In this situation, the difference in the opportunity quality would be caused by different incidence angles, if allowed. Another cause of a difference might be a partially faulty sensor on one satellite in a constellation. For agile optical satellites, the opportunities of one strip could be associated with different satellites or satellite constellations and the difference in the quality could arise from the difference between the satellite missions. Also, for optical satellites the differences between the opportunities might be related to the difference in illumination and time of day.

Introducing the *quality measure* φ_c^r of the AOI coverage plan c for request r further refines the optimization criteria to

$$Max \sum_{r \in R} p_r * \sum_{c \epsilon \Phi^r} \varphi_c^r * f \left(\frac{\sum_{j \in I_c^r} A_j \sum_{s \in \Gamma \cup \Pi} \sum_{i \in P_j^s} q_{ir}^s y_i}{A_c^r} \right) \quad (4)$$

4.2.2 Monetary Cost

If the upper bound of the monetary cost is not known, the optimization criteria may include its minimization:

$$Min \sum_{r \in R} \sum_{c \epsilon \Phi^r} \sum_{j \in I_c^r} \sum_{s \in \Gamma \cup \Pi} \sum_{i \in P_j^s} c_i^s y_i \quad (5)$$

However, note that this should not be the only optimization criteria since it would result in an empty schedule.

4.2.3 Dynamic Problem and Rolling Horizons

When solving a dynamic scheduling problem using rolling horizons, a series of subsequent scheduling sessions will be performed. The scheduling horizons of two consecutive scheduling sessions may overlap. The total acquired area of request r consists of the areas that scheduled in this scheduling session plus the areas that were scheduled in all previous scheduling sessions π'_r, up to the start time of the current scheduling horizon. Thus instead of (I) we use [23]

$$\pi_r = \pi\prime_r + \sum_{j \in I^r} A_j \sum_{i \in P_j} y_i$$

4.3 Mathematical Programming Formulation

The mathematical programming formulation of the image acquisition scheduling problem for the integrated management of multiple heterogeneous satellite missions is a mixed-integer programming (MIP) model. The portion of the model dealing with non-agile satellites is a special case of a maximum clique problem [24]. The portion of the model dealing with agile satellites is a special case of the selective multiple traveling salesman problem with time windows (m-TSPTW). The satellite resource constraints are similar to the m-TSPTW constraints handling the maximal route duration and the maximal total service time of a route. The corresponding entities of the m-TSPTW and the multi-satellite image acquisition scheduling problem are: (1) cities (nodes) are the opportunities, (2) service times

are the observation times, (3) time windows are the opportunity time windows, (4) each salesman is one satellite, and (5) distances between cities are the maneuver times.

The proposed MIP model can handle:

- Multiple missions, including multiple satellites and multiple constellations.
- Heterogeneous satellites.
- Any combination of non-agile and agile satellites.
- Multiple AOI coverage plans for one acquisition request.
- Multiple strips in an AOI.
- Multiple opportunities for a strip.

$$Max\ \omega_1 \sum_{r\in R} p_r \sum_{c\epsilon\Phi^r} \varphi_c^r\ f\left(\frac{\sum_{j\in I_c^r} A_j \sum_{s\in\Gamma\cup\Pi}\ \sum_{i\in P_j^s} q_{ir}^s y_i^s}{A_c^r}\right) - \omega_2 \sum_{r\in R}\sum_{c\epsilon\Phi^r}\sum_{j\in I_c^r}\sum_{s\in\Gamma\cup\Pi}\sum_{i\in P_j^s} c_i^s y_i^s \tag{6}$$

Subject to:

$$\sum_{s\in\Pi\cup\Gamma}\sum_{i\in P_j^s} y_i^s \le 1,\ j \in I_c^r, c\epsilon\Phi^r, r\epsilon R \tag{7}$$

$$y_i^s + y_j^s \le 1, \forall\{i, j\} \in C, s \in \Gamma \tag{8}$$

$$\sum_{j\in I_c^r}\sum_{s\in\Pi\cup\Gamma}\sum_{i\in P_j^s} y_i^s \le z_c^r * \mid I_c^r \mid, c\epsilon\Phi^r, r\epsilon R \tag{9}$$

$$\sum_{c\epsilon\Phi^r} z_c^r \le 1, \forall r\epsilon R \tag{10}$$

$$\sum_{k\in\{0^s\}\cup T_{si+}} x_{ki}^s = y_i^s, \forall i \in T_s, \forall s \in \Pi \tag{11}$$

$$\sum_{j\in\{n^s+1\}\cup T_{si-}} x_{ij}^s = y_i^s, \forall i \in T_s, \forall s \in \Pi \tag{12}$$

$$\sum_{j\in T_s\cup\{n^s+1\}} x_{0^s j}^s = 1, \forall s \in \Pi \tag{13}$$

$$\sum_{j \in T_s \cup \{0^s\}} x_{j,n^s+1} = 1, \forall s \in \Pi \tag{14}$$

$$\sum_{i \in T_s} \sum_{j \in T_s} x_{ij}^s = \sum_{j \in T_s} y_i^s - 1, \forall s \in \Pi \tag{15}$$

$$\left(u_i^s + d_i^s\right) + t_{ij}^s - \left(1 - x_{ij}^s\right) M \leq u_j^s, 0^s \leq i, j \leq n^s + 1, \forall s \in \Pi \tag{16}$$

$$u_{n^s+1}^s \leq T, \forall s \in \Pi \tag{17}$$

$$\sum_{j \in T_s \cup \{0^s, n^s+1\}} d_j^s y_j^s + \sum_{j \in T_s \cup \{0^s\}} \sum_{j \in T_s \cup \{n^s+1\}} t_{ij}^s x_{ij}^s \leq T, \forall s \in \Pi \tag{18}$$

$$E_i^s \leq u_i^s \leq L_i^s, 0^s \leq i \leq n^s + 1, \forall s \in \Pi \tag{19}$$

$$y_0^s = y_{n^s+1}^s = 1, \forall s \in \Pi \cup \Gamma \tag{20}$$

$$\sum_{i \in T_s^k} d_i^s y_i^s \leq \tau_1^s, \forall k \in \left\{1, \ldots, \kappa^s\right\}, \forall s \in \Gamma \cup \Pi \tag{21}$$

$$\sum_{i \in T_s} d_i^s y_i^s \leq \kappa^s * \tau_2^s, \forall s \in \Gamma \cup \Pi \tag{22}$$

$$\beta^s \sum_{j \in T_s \cup \{0^s, n^s+1\}} d_j^s y_j^s + \gamma^s \sum_{j \in T_s \cup \{0^s\}} \sum_{j \in T_s \cup \{n^s+1\}} t_{ij}^s x_{ij}^s \leq \Xi^s, \forall s \in \Pi \tag{23}$$

$$\sum_{j \in T_s \cup \{0^s, n^s+1\}} \delta_j^s * d_j^s * y_j^s \leq \Psi^s, \forall s \in \Gamma \cup \Pi \tag{24}$$

$$y_i^s \in \{0, 1\}, 1 \leq i \leq n^s, \forall s \in \Pi \cup \Gamma \tag{25}$$

$$x_{ij}^s \in \{0, 1\}, 0^s \leq i, j \leq n^s + 1, \forall s \in \Pi \tag{26}$$

$$z_c^r \in \{0, 1\}, c \epsilon \Phi^r, r \epsilon R \tag{27}$$

$$u_i^s \in R, 1 \leq i \leq n^s, \forall s \in \Pi \tag{28}$$

where M in Constraint (16) is an arbitrarily large number, greater or equal to $\max_{i,j} \left\{ u_i + d_i + t_{ij} \right\}$. It states that $x_{ij}^s = 1 \Longrightarrow \left(u_i^s + d_i^s\right) + t_{ij}^s \leq u_j^s$. Note also that $x_{ij}^s = 1 \Longrightarrow y_i^s = 1 \bigwedge y_j^s = 1$.

Objective function (6) is a weighted sum of two objective criteria: maximizing the fulfillment of the high-priority requests with the AOI coverage plans of high quality consisting of the high quality opportunities and minimizing the dollar cost of the schedule.

Constraint (7) assures that at most one opportunity for each strip is acquired. Constraint (8) assures that no two opportunities that are in conflict are to be imaged. Set C contains all opportunity pairs that are in conflict. Two opportunities are in conflict if: (a) they belong to the same satellite and (b) their (opportunity) time windows overlap.

Constraint (10) states that at most one AOI coverage plan is selected to be scheduled for each acquisition request. Constraint (9) assures that only the opportunities for acquiring the strips of the selected AOI coverage plan are scheduled. Constraints (11) and (12) state that, for each satellite, before and after of each scheduled opportunity there is another scheduled opportunity, or the initial and final camera head positions. Constraints (13) and (14) state that for each satellite at least two positions (dummy opportunities) are scheduled: the initial camera head position and the final camera head position. Constraint (15) provides relationship between two sets of the problem variables x and y. Constraint (16) represents the sub-tour elimination constraints. Constraint (17) states that for each satellite the camera head has to be back at its final position before the end of schedule horizon T. Constraint (19) guarantees that each strip acquisition starts within the strip's opportunity time window. Constraint (20) makes sure that the initial and the final camera head positions are included in the schedule, for each satellite.

The proposed model accounts for heterogeneous satellites in terms of the resource constraints, where each satellite has different parameters and resource bounds. Constraints (21) and (22) deal with thermal requirements arising from imaging activities. Constraint (21) states that the imaging duration per orbit cannot exceed τ_1 of each satellite s. Constraint (22) states that the average imaging duration per orbit, over the planning horizon that has κ orbits, cannot exceed τ_2 for each satellite s. Constraint (23) handles the power constraints and states that the power is used for imaging activity and for the maneuver activity and it is proportional to the duration of those activities. Parameter Ξ^s is the total power available within the time horizon T on satellite s, β^s is the power usage per 1 s of time of imaging activity, and γ^s is the power usage per 1 s of maneuvering activity of the satellite. Constraint (24) handles memory requirements. Ψ^s is the total memory available within the time horizon T on satellite s, and $\delta_j^s$is the memory consumption per 1 s of image acquisition by satellite s for opportunity j.

Constraints (25), (26), and (27) are binary variable constraints, whereas constraint (28) states that the start time for the acquisitions can have non-integral values.

In the proposed model, the constraints (8) and (11)–(28) could be partitioned into separate subproblems each of which deals with scheduling one satellite. The only

linking expressions are the objective function and Constraints (7), (9), and (10). Thus, decomposition solution approaches could be used, including Dantzig–Wolfe decomposition.

The MIP model can be used for solving the scheduling problem for multiple heterogeneous satellite missions using a general purpose MIP solver (CPLEX or Gurobi). In the case of large-scale practical problems dealing with large satellite constellations, the MIP model is a basis for designing a model-based metaheuristics (matheuristic) such as variable intensity neighborhood search (VINS) [25–27].

5 Conclusion

This chapter provides a description of a novel optimization problem in satellite industry that arises from the need for integrated utilization of multiple-satellite missions for the purpose of collection of surveillance products. The chapter also presents a survey of the image acquisition scheduling literature and a mathematical programming formulation of the integrated scheduling problem. Novelties of the presented optimization scheduling problem compared to the literature include dealing with:

- Multiple-satellite missions.
- Heterogeneous satellites, in terms of agility and in terms of sensor types.
- Multiple AOI coverage plans for one image acquisition request.
- Quality of coverage plans, strips, and imaging opportunities.

Integrated multi-mission collection scheduling of heterogeneous satellites would result in enhanced intelligence collection beneficial to many applications including maritime surveillance and ship detection, land surveillance of remote unpopulated areas, environmental sensing, and search-and-rescue operations.

The design and development of the MiMPS system with the multi-mission acquisition schedule optimizer system for generating optimized and de-conflicted schedules has the potential to considerably increase the overall operational performance and utilization of satellite resources. The proposed MIP model can be used for solving the scheduling for multiple heterogeneous satellite missions using a general purpose MIP solver or a model-based metaheuristics (matheuristic).

A.1 Appendix: Survey Summary

Table 1 summarizes the literature survey presented in this chapter. The manuscripts are grouped by the solution approach in the following manner: exact methods, heuristics and metaheuristics, constraint programming, and bounds. Within each group, the manuscripts are listed chronologically.

Table 1 Summary of the literature on image acquisition scheduling for multi-satellite collection scheduling

Authors	Year	Satellite(s)	Method	Solution quality	Further description
Exact algorithms					
Hall and Magazine [16]	1994	1 satellite	*Dynamic programming* algorithm with a bounding by either Lagrangian relaxation or relaxation of some constraints		Problem instances: 200 targets
Gabrel and Vanderpooten [12]	2002	1 satellite, 4 orbits, non-agile	Model: multi-criteria path problem over an acyclic graph. Algorithm: *the label-setting shortest path algorithm* for generating all efficient paths, followed by an interactive session	Less than an hour running time	Objective function: Max the number of high-priority shots, max demand satisfaction, min satellite use to prevent wear
Heuristics and metaheuristics					
Bensana et al. [3]	1996	1 satellite, 1 orbit, multiple orbits, non-agile	Depth-first *branch and bound*, *Russian Doll search*, best-first branch and bound, a *greedy* algorithm, and t*abu* search. Integer linear programming (ILP) and valued constraint satisfaction problem (VCSP) formulations	*Tabu search* performed best in general, Russian Doll search performed best for the one orbit case but failed in six of the seven instances in the multiple orbit cases	Problem instances: 13 instances (1 satellite, 1 orbit), and 7 instances (1 satellite, multiple orbits) SPOT 5 with 3 cameras. Mono-image and stereo-image requests
Gabrel et al. [13]	1997	1 satellite, non-agile	An *approximate* method with discrete times, an approximate method with cont. time and *greedy* algorithm based on the *longest path* algorithm, and an exact method *branch and bound* approach with a depth-first search strategy	The *continuous-time approximation* algorithm averaged within 7% of optimal. The discretized time algorithm performed very close to the continuous approximate one	
Muraoka et al. [29]	1998	1 satellite	*Greedy*: one by one request is scheduled		Ranking function consists of 12 elements ASTER—advanced spaceborne thermal emission and reflectance radiometer—a multi-spectral imaging radiometer, USA and Japan

(continued)

Table 1 (continued)

Authors	Year	Satellite(s)	Method	Solution quality	Further description
Pemberton [31]	2000	1 satellite	Priority *segmentation*: each subproblem solved to optimality by *branch and bound*	*Priority segmentation* better than greedy algorithm that sorts tasks by priority and schedules them one by one	
Wolfe and Sorensen [43]	2000	1 satellite, 2-satellite constellation	*Hill-climbing*, look-ahead algorithm, *genetic* algorithm with look-ahead Defined the problem as a window-constrained packing problem	*Genetic* algorithm performed the best. Needed longer run times. Look-ahead algorithm may be more practical for large problem instances	Schedule horizon: 1 week
Frank et al. [10]	2001	1 satellite	*Greedy* hill-climbing search with *stochastic* variations to escape local optima. Algorithm based on *constraint-based interval planning*		Ranking: priority and contention (how many opportunities a request has)
Vasquez and Hao [36]	2001	1 satellite, non-agile	*Tabu* search together with "logic-constrained" knapsack formulation	*Tabu* search better than tabu search of [3]	
Lemaitre et al. [19]	2002	1 satellite, 1 orbit, *Agile* satellite	*Greedy* algorithm, *dynamic programming* algorithm, *constraint programming*, and *local search* algorithm. Greedy and dynamic algorithm ignored stereo-images	Only mono-images: dynamic programming approach was the best performer; stereo-images included: *local search* method were the best	Problem instances: 6 problem instances. Included stereo-images
Globus et al. [14]	2003	1 satellite, 2-satellite constellation *Agile* Optical, included: access, slew, and dwell times	*Genetic* algorithm, *simulated* annealing, *stochastic hill-climbing*, *iterated sampling*	*Simulated* annealing was the best	Problem instances: 4200 imaging targets with priorities 1–5 Objective function: Minimize weighted sum of unscheduled image priorities, total slew time, sum of slew angles of scheduled images Considered: on-board data storage and downlink

Globus et al. [15]	2004	2- or 3-satellite constellation, optical satellites	*Genetic* algorithm (13 variations), *simulated* annealing, stochastic hill-climbing, *squeaky wheel*	*Simulated* annealing was the best, closely followed by hill-climbing. Genetic algorithm was the worst	Problem instances: 10 instances, 2100*m targets with priorities, where m is the number of satellites Objective function: minimize weighted sum of unscheduled image priorities, mean slew time, mean of off-nadir pointing angles of scheduled images
Cordeau and Laporte [5]	2005	1 satellite, 1 orbit *Agile* satellite	*Tabu* search. Allowed infeasibility by relaxing the time window constraints Upper bound: column generation procedure solving the linear program relaxation	*Tabu search* achieved *near-optimal* solutions. Compared to upper bound	Problem instances: ROADEF 2003 competition Objective function: piecewise-linear Convex with respect to the proportion of the polygon's area being acquired
Lin et al. [21]	2005	1 satellite	*Lagrangian relaxation* and *linear search*	*Lagrangian relaxation* better when compared to simple tabu search	China/Taiwan ROCSAT II (FORMOSAT II)
Lin and Chang [22]	2005	1 satellite	Hybrid approach: *Lagrangian relaxation* with a *tabu* search based feasibility adjustment	*Hybrid* better than [21]	
Bianchessi et al. [4]	2007	2-satellite constellation Optical, *Agile*	*Tabu* search partly based on [5]	The same as [5]	Schedule horizon: 24 h Objective function: max weighted sum of utilities assigned to users; linear with respect to the proportion of the polygon's area being acquired. Priorities PLEIADES—French satellites Consecutiveness of strips from one polygon request
Wang et al. [42]	2009	3-satellite constellation, multiple orbits	*Tabu* search heuristic embedded in ILOG dispatcher and a *greedy*-based algorithm called "conflict-avoided heuristic"	*Tabu* search resulted in better solutions. Running time for their largest instance, the greedy took 50 s, while tabu search took 3499 s	

(continued)

Table 1 (continued)

Authors	Year	Satellite(s)	Method	Solution quality	Further description
Hwang et al. [17]	2010	Multiple-satellite constellation	*Genetic* algorithm. Problem simplified by a division in limited number of single-orbit scheduling problems.	Not provided	Problem instances: 20 targets Objective function: Multi-criteria FORMOSAT 5, Taiwan
Nelson [30]	2012	2-satellite constellation, 2 orbits, SAR satellite constellation	Three-step approach: cluster-route-schedule. Clustering groups imaging targets. Routing step uses *column generation*, where each satellite in one orbit is treated as separate sub-problem to determine the clusters; the scheduling step schedules each satellite separately using *time–space networks* and heuristics.	768 clusters formed	Problem instances: 8250 targets 4271 could be imaged in schedule horizon of 208 min = two-satellite revolutions
Wu et al. [44]	2013	4-satellite constellation, multiple orbits, semi-agile (only rolls)	Adaptive *simulated annealing* with dynamic clustering This algorithm was compared to ant colony, tabu search, genetic algorithm, simulated annealing with static clustering, and highest-priority first algorithm	Adaptive *simulated annealing* with dynamic clustering	Problem instances: 100–1000 targets, spot target tasks Priorities Schedule horizon: 1 day Objective function: Max profit
Xiaolu et al. [45]	2014	Multiple-satellite constellation, multiple orbits, semi-agile (only rolls)	Decomposition algorithm: one subproblem is assignment of a task to an opportunity and the other subproblem is merging of tasks into an observation. The problems are solved iteratively until stopping criteria are met. MIP is proposed for scheduling observations for collection of satellites	The proposed *decomposition* algorithm better than simulated annealing: around 25% better and up to 50% for some instances Operational in the China Satellite Management Center.	Problem instances: random instances: uniform, collective, and mixed. 63 instances: 100–600 requests, 2–8 satellites, and 200–4300 opportunities Objective function: max profit Energy and memory are considered Max continuous imaging time is considered
Malladi et al. [23, 24]	2015, 2017	Multiple-satellite constellation, multiple orbits, semi-agile (only roll)	Model-based metaheuristic (matheuristic). MIP. Also, modeling as a new cluster-restricted maximum weight cliques problem	Compared to CPLEX Real-world problems instances. Also BOSHLIB and DIMACS benchmarks	Objective function: max priority

Tangpattanakul et al. [34]	2015	Agile includes stereo requests and polygon requests	Iterated indicator-based multi-objective local search		Problem instances: modified ROADEF 2003 Objective function: multi-criteria: max profit + max fairness (by minimizing the maximal difference between the user profits)
Wang et al. [41]	2015	Multiple-satellite constellation, multiple orbits, semi-agile (only roll)	Exact: decomposition with master problem solved by enumeration and subproblems by dynamic programming Heuristics: for generating feasible solution to subproblems. Master still solved by enumeration	Compared to Liao and Chang [20] model solved by CPLEX. Generated better solutions	Problem instances: three satellites. Number of requests from 10 to 160 Planning horizon: from 6 h to 24 h Objective function: max expected profit. Uncertainty of clouds
Constraint programming (CP)					
Verfaille et al. [40]	1996	1 satellite, 1 orbit, non-agile	*Russian Doll algorithm* to improve search for an opt solution in CP; depth-first search branch and bound	*Russian Doll* algorithm solved all instances to optimality, while branch and bound did not solve any	Problem instances: 8 instances, 30 min time limit
Lemaitre et al. [18]	2000	1 satellite *Agile* satellite	*Constraint programming* (OPL Studio Framework), local search (LS)	*Local search* algorithm was better than CP	Problem instances: 7 instances Stop criteria: CP: 5 min, LS: 2 min
Bounds					
Vasquez and Hao [37]	2003	1 satellite, multiple orbits, SPOT 5, non-agile	Using *tabu* search the problem is partitioned into subproblems, which are solved exactly by an iterative enumeration algorithm	Equal or better than [11]. Upper bounds less than 3% from lower bounds	

(continued)

Table 1 (continued)

Authors	Year	Satellite(s)	Method	Solution quality	Further description
Gabrel and Murat [11]	2003	1 satellite, 1 orbit, SPOT 5, 3 cameras, on-board memory, non-agile	*Vertex-path* formulation with a *column generation* procedure	Bounds generally within 10% of optimality compared to [3] 2006: show that the bounds are tighter than linear relaxation	Problem instances: up to 300 targets
Benoist and Rottembourg [2]	2004	*Agile* satellites	*Prize collecting traveling salesman problem with time windows*, enhanced with valid inequalities based on task interval reasoning and a Russian Doll search approach	With valid inequalities: 22% gap. With Russian Doll search: 12% gap, took several hours	Problem instances: 20 instances with 300–500 targets
Bianchessi et al. [4]	2007	2-satellite constellation, optical, *Agile*	*Column generation* based on set partitioning. Branch and price	The same as [5]	Schedule horizon: 24 h Objective function: max weighted function of utilities assigned to users; linear with respect to the proportion of the polygon's area being acquired. Priorities PLEIADES—French satellites Consecutiveness of strips from one polygon request

References

1. Augenstein, S.: Optimal scheduling of Earth-imaging satellites with human collaboration via directed acyclic graphs. In: The Intersection of Robust Intelligence and Trust in Autonomous Systems: AAAI Spring Symposium, 2014
2. Benoist, T., Rottembourg, B.: Upper bounds of the maximal revenue of an earth observation satellite. 4OR: Quart. J. Belgian, French and Italian Oper. Res. Soc. **2**(3), 235–249 (2004)
3. Bensana, E., Verfaillie, G., Agnese, J.C., Bataille, N., Blumstein, D.: Exact and inexact methods for the daily management of an earth observation satellite. In: Proceedings of the 4th International Symposium on Space Mission Operations and Ground Data Systems (SpaceOps-96), Munich, Germany, 1996
4. Bianchessi, N., Cordeau, J.-F., Desrosiers, J., Laporte, G., Raymond, V.: A heuristic for the multi-satellite, multi-orbit and multi-user management of earth observation satellites. Eur. J. Oper. Res. **177**, 750–762 (2007)
5. Cordeau, J.-F., Laporte, G.: Maximizing the value of an earth observation satellite orbit. J. Oper. Res. Soc. **56**(8), 962–968 (2005)
6. Dishan, Q., Chuan, H., Jin, L., Manhao, M.: A dynamic scheduling method of earth-observing satellites by employing rolling horizon strategy. Sci. World J. **2013**, 1–11 (2013)
7. Farr, T.G.: Chapter 5: Radar interactions with geologic surface. In Guide to Magellan Image Interpretation, NASA and Jet Propulsion Laboratory, California Institute of Technology, 1993
8. Fisher, W.: The optiwise corporation deconfliction scheduler algorithms (as used in STK/Scheduler). Optwise (2004)
9. Fisher, W.A., Herz, E.: A flexible architecture for creating scheduling algorithms as used in STK scheduler. White paper, Optwise Corporation and Orbit Logic Incorporated, 2013
10. Frank, J., Jonsson, A., Morris, R., Smith, D.: Planning and scheduling for fleets of earth observing satellites. In: Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SAIRAS-01), Montreal, Canada, 2001
11. Gabrel, V., Murat, C.: Mathematical programming for earth observation satellite mission planning. In: Ciriani, T., Fasano, G., Gliozzi, S., Tadei, R. (eds.) Operations Research in Space and Air, pp. 103–122. Kluwer, Dordrecht (2003)
12. Gabrel, V., Vanderpooten, D.: Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an earth observing satellite. Eur. J. Oper. Res. **139**, 533–542 (2002)
13. Gabrel, V., Moulet, A., Murat, C., Paschos, V.: A new single model and derived algorithms for the satellite shot planning problem using graph theory concepts. Ann. Oper. Res. **69**, 115–134 (1997)
14. Globus, A., Crawford, J., Lohn, J., Pryor, A.: Scheduling earth observing satellites with evolutionary algorithms. In: Proceedings of the 1st International Conference on Space Mission Challenges for Information Technology (SMC-IT-03), Pasadena, CA, 2003
15. Globus, A., Crawford, J., Lohn, J., Morris, R.: A comparison of techniques for scheduling earth observing satellites. In: Proceedings of the 16th Conference on Innovative Applications of Artificial Intelligence (IAAI-04), San Jose, CA, 2004
16. Hall, N., Magazine, M.: Maximizing the value of a space mission. Eur. J. Oper. Res. **78**, 224–241 (1994)

17. Hwang, F.-T., Yeh, Y.-Y., Li, S.-Y.: Multi-objective optimization for multi-satellite scheduling system. In: Proceedings of 31st Asian Conference on Remote Sensing 2010 (ACRS 2010), Hanoi, 1–5 Nov (2010)
18. Lemaitre, M., Verfaillie, G., Jouhaud, F., Lachiver, J.M., Bataille, N.: How to manage the new generation of agile earth observation satellites. In: Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space, pp. 1–10 (2000)
19. Lemaitre, M., Verfaillie, G., Jouhaud, F., Lachiver, J.-M., Bataille, N.: Selecting and scheduling observations of agile satellites. Aerosp. Sci. Technol. **6**, 367–381 (2002)
20. Liao, D., Chang, Y.: Imaging order scheduling of an Earth observation satellite. IEEE Trans. Syst. Man Cybern. C Appl. Rev. **37**(5), 794–802 (2007)
21. Lin, W., Liao, D., Liu, C., Lee, Y.: Daily imaging scheduling of an earth observation satellite. IEEE Trans. Syst. Man Cybern. **35**(2), 213–223 (2005)
22. Lin, W.C., Chang, S.C.: Hybrid algorithms for satellite imaging scheduling. In: 2005 IEEE International Conference on Systems, Man and Cybernetics, Waikoloa, 10–12 Oct 2005
23. Malladi, K.T., Mitrovic-Minic, S, Karapetyan, D., Punnen, A.P.: Satellite constellation image acquisition problem: a case study. In: Fasano G., Pinter J.D. (eds.) Space Engineering: Modelling and Optimization with Case Studies. Springer (2016)
24. Malladi, K.T., Mitrovic-Minic, S., Punnen, A.P.: Cluster restricted maximum weight clique problem: algorithms and empirical analysis. Comput. Oper. Res. **85**, 113–128 (2017)
25. Mitrovic-Minic, S., Punnen, A.P.: Very large-scale variable neighborhood search for the generalized assignment problem. Journal of Interdisciplinary Mathematics. **11**(5), 653–670 (2008)
26. Mitrovic-Minic, S., Punnen, A.P.: Variable intensity local search. Annals of Information Systems. **10**, 245–252 (2009)
27. Mitrovic-Minic, S., Punnen, A.P.: Local search intensified: very large-scale variable neighborhood search for the multi-resource generalized assignment problem. Discret. Optim. **6**, 370–377 (2009)
28. Mitrovic-Minic, S., Berger, J., Thomson, D.: Collection Planning Management: Multi-Satellite Collection Scheduling. MDA, CR, DRDC-RDDC-2016-C309 (2016)
29. Muraoka, H., Cohen, R., Ohno, T., Doi, N.: ASTER observation scheduling algorithm. In: Proceedings of the 5th International Symposium on Space Mission Operations and Ground Data Systems (SpaceOps-98), Tokyo, Japan, 1998
30. Nelson, F.N.: Scheduling optimization for imagery satellite constellations using column generation. PhD Thesis, Department of Systems Engineering and Operations Research, Volgenau School of Engineering, George Mason University (2012)
31. Pemberton, J.: Towards scheduling over-constrained remote sensing satellites. In: Proceedings of the 2nd NASA International Workshop on Planning and Scheduling for Space, pp. 84–89, San Francisco, CA, 2000
32. Pixalytics. https://www.pixalytics.com/sats-orbiting-earth-2017/ (2017)
33. Secker, J., Robson, M., Rowe, J., Vachon, P.W.: Automated acquisition planning for commercial satellite imagery. In: IEEE International Conference on Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006, pp. 3279–3282 (2006)
34. Tangpattanakul, P., Jozefowiez, N., Lopez, P.: A multi-objective local search heuristic for scheduling earth observations taken by an agile satellite. Eur. J. Oper. Res. **245**, 542–554 (2015)
35. Vachon, P.W., Secker, J., Werle, D.: Remote sensing in support of Arctic intelligence: Development of sensor selection rules. DRDC Ottawa TM 2007–109 (2007)
36. Vasquez, M., Hao, J.K.: A logic-constrained knapsack formulation and a Tabu algorithm for the daily photograph scheduling of an earth observation satellite. Journal of Computational Optimization and Applications. **20**(2), 137–157 (2001)
37. Vasquez, M., Hao, J.K.: Upper bounds for the SPOT 5 daily photograph scheduling problem. J. Comb. Optim. **7**, 87–103 (2003)

38. Verfaillie, G.: Planning and scheduling activities for earth surveillance and observation satellites: a constraint-based perspective. In: ICAPS 2013 Summer School, Perugia, 4–7 June 2013
39. Verfaillie, G., Lemaitre, M.: Planning and scheduling activities for earth surveillance and observation satellites: a constraint-based perspective. In: ICAPS Tutorial 2006, Cumbria, 2006
40. Verfaillie, G., Lemaitre, M., Schiex, T.: Russian doll search for solving constraint optimization problems. In: Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96), pp. 181–187. Portland 1996
41. Wang, J., Demeulemeester, E., Qiu, D., Liu, J.: Exact and inexact scheduling algorithms for multiple earth observation satellites under uncertainties of clouds. In: Technical Chapter KBI_1514, KU Leuven, Faculty of Economics and Business, Preprint submitted to EJOR on 1 July 2015
42. Wang, P., Tan, Y., Reinelt, G.: A comparison of heuristic methods for scheduling Earth observing satellites fleet. In: Proceedings of 2009 International Conference on Information Technology and Computer Science, Kiev, 25–26 July 2009
43. Wolfe, W., Sorensen, S.: Three scheduling algorithms applied to the earth observing systems domain. Manag. Sci. **46**(1), 148–168 (2000)
44. Wu, G., Wang, H., Li, H., Pedrycz, W., Qiu, D., Ma, M., Liu, J.: An adaptive simulated annealing-based satellite observation scheduling method combined with a dynamic task clustering strategy. Comput. Res. Repos. (2014). https://arxiv.org/abs/1401.6098
45. Xiaolu, L., Baocun, B., Yingwu, C., Feng, Y.: Multi satellites scheduling algorithm based on task merging mechanism. Appl. Math. Comput. **230**, 687–700 (2014)

Single-Stage-to-Orbit Space-Plane Trajectory Performance Analysis

Erwin Mooij

Abstract The development of fully reusable launch systems has been the topic of many studies since the 1960s. Over the years, several aspects of both so-called single- and two-stage-to-orbit space planes have provided many interesting research topics. Amongst others, the constrained trajectory optimisation has proven to be a challenging subject. In this chapter, an inverse-dynamics approach is combined with trajectory optimisation and analysis, by discretising a representative (vertical-plane) ascent trajectory into a number of flight segments, and by parametrising the guidance in terms of flight-path angle as a function of altitude. When the individual guidance parameters are varied, the effect on performance indices payload mass and integrated heat load can be analysed. This can subsequently lead to a refinement of the trajectory. To do so with limited effort, design-of-experiment techniques are used. It is shown that with this relatively simple simulation scheme, combined with variance analysis and response-surface methodology, the insight in the trajectory dynamics can be increased. Alternatively, this method can be used as refinement to an otherwise (local) optimum trajectory. It is stressed, though, that the application of design of experiments to the ascent-trajectory problem cannot replace numerical optimisation. Finally, the impact of using thrust-vector control as a means to (partially) trim the vehicle shows significant fuel savings and should therefore be included in the optimisation process.

E. Mooij (✉)
Delft University of Technology, Faculty of Aerospace Engineering, Delft, The Netherlands
e-mail: e.mooij@tudelft.nl

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_12

1 Introduction

In the ongoing effort to reduce launch costs, the current development focuses on the reusability of the first stage of conventional launchers, such as the Falcon 9.[1] Studies on the reusability of second stages are ongoing, and it will probably not be for long that this will be flight tested [1].

The development of fully reusable launch systems has been the topic of many studies since the 1960s, but the enormous cost involved in the technology development has until now put a stop to most system developments. Over the years, several aspects of both so-called single-stage-to-orbit (SSTO) and two-stage-to-orbit (TSTO) space planes have provided many interesting research topics. Amongst others, the constrained trajectory optimisation has proven to be a challenging subject. As for SSTO launchers, the ascent is the most strenuous flight phase; in fact, the design and sizing is largely determined by the ascent mission goals [2]. Because of the very small performance margin, the guidance should be based on fuel-optimal trajectories to ensure mission success [3, 4]. Also frequent on-board reoptimisation and high-gain perturbation guidance that rapidly responds to perturbations and therefore closely tracks the optimal trajectory will likely be required [5].

For TSTO vehicles, on the other hand, the requirements are less demanding, since the burden of achieving mission success is divided over two vehicles. Last but not least, since it is unlikely that SSTO and TSTO vehicles will go all the way to orbit on the very first mission, also flight test-type missions should be considered. Therefore, guidance must also be provided for more conventional flight modes, e.g., lower-performance climbs and descents, altitude holds, and achieving and maintaining course to area navigation targets.

New developments in guidance algorithms for space planes commonly start from the notion that the algorithms should result in optimal (or near-optimal) trajectories. For instance, [6] and [7] apply the singular-perturbation technique to achieve a rapid, near-optimal trajectory generation and guidance scheme for space planes (zero-order solution). Van Buren and Mease [8] extended this model to a first-order solution and included feedback control. Drawback of these methods is that they lean heavily on trajectory optimisation. Alternatively, one can draw from guidance-system development in the field of re-entry, e.g., the predictor–corrector guidance developed by Lu [9], which will then have to be extended with thrust forces.

Since the main focus of this study is not the development of robust guidance algorithms, a relatively simple alternative can be found in a so-called *inverse-dynamics* approach, which can be described as follows. Given a differential equation of motion that relates the time derivative of a state variable to some external force, this time derivative can be prescribed such that the external forces (and therefore also the controlling variables) can be determined, instead of the usual way of computing the

[1] De Selding, P.B., "SpaceX to launch SES-10 on previously flown Falcon 9 this year", *Spacenews*, August 30, 2016. http://spacenews.com/spacex-to-launch-ses-10-satellite-on-reused-falcon-9-by-years-end/. Accessed October 5, 2017.

time derivative for a given value of the external force (hence the term *inverse*). This approach is very much suitable to the foreseen trajectory segmentation and using the dynamics parameters as design parameters to analyse the complete trajectory.

For actual guidance-system design, the inverse-dynamics approach has been applied by Lu [10] and proved to be quite effective in solving the difficult problem of trajectory optimisation, and by Hess et al. [11] who developed a generalised technique for inverse simulation applied to aircraft manoeuvres. Finally, Morrio et al. [12] used flatness theory combined with non-linear dynamic inversion that led to a longitudinal guidance law that is in-flight self-adaptive to any feasible hypersonic trajectory. It only requires a limited set of design parameters, because of its analytical formulation.

In this chapter, the inverse-dynamics approach is combined with trajectory optimisation and analysis, by discretising a representative (vertical-plane) ascent trajectory into a number of flight segments, and by parametrising the guidance in terms of flight-path angle as a function of altitude. When the individual guidance parameters are varied, the effect on performance indices, such as payload mass and integrated heat load, can be analysed, which can subsequently lead to a refinement of the trajectory. To do so with limited effort, design-of-experiment techniques are used. In this way, insight in the trajectory dynamics can be obtained in an easy and fast way. And, once a consolidated methodology has been established, it may assist in local refinement of an otherwise optimal trajectory obtained from a global optimisation process, or may provide insight in the robustness of the trajectory if small changes in trajectory parameters occur.

The layout of the rest of this chapter is as follows. In Sect. 2, the simulation model is introduced, consisting of the reference vehicle, the flight-dynamics model, and the guidance system. In the following Sect. 3 the parametric design and analysis method taken from the field of design of experiments is discussed. This methodology is successively applied to the sub-optimisation and analysis of the vertical-ascent flight of an SSTO space plane, for which the (constrained) reference trajectory has been predefined in terms of flight-path angle and flight-path angle rate (Sect. 4). Included in the analysis is the application of thrust-vector control to reduce the induced drag due to elevon deflections. Section 5 concludes this chapter with some final remarks.

2 Simulation Model

2.1 Reference Vehicle

The winged-cone configuration (WCC) reference vehicle, also known as the Langley accelerator, is a generic, horizontal take-off, SSTO configuration that can be used for point-mass as well as batch and real-time six-degrees-of-freedom simulations [13]. In the past, the WCC has been successfully used in many optimisation and control-system design studies, see, for instance, [8, 10, 14–16]. Even though design details concerning structure and propulsion system are lacking, and the performance

of the vehicle is far too optimistic, the advantage of using this vehicle lies in the fact that a complete aero-propulsive database is publicly available. As it is difficult to find complete and consistent databases of other, more realistic space-plane concepts, the WCC is chosen to be the reference vehicle for setting up the performance-analysis methodology. Below, the aerodynamic, propulsion and mass models will be briefly described. More details as well as the numerical values of the complete database are provided by Shaughnessy et al. [13].

The WCC has a dry mass of 58,968 kg and is composed of an axisymmetric 5° half-angle conical forebody, a cylindrical engine nacelle section and a cone frustum nozzle [15]. The wing has a leading-edge sweep of 78° and is set at 0° incidence and dihedral. The wing is a 4% thick diamond airfoil. Elevons are located at the trailing edge of the wing with their hinge line perpendicular to the fuselage centre line, and positive deflections are with the trailing edge down. The vertical tail is a 4% thick diamond airfoil with a leading-edge sweep angle of 70° and includes a rudder with a hinge line at the 75% chord position measured from the leading edge. Positive rudder deflections are with the trailing edge left. The canards have a 6% thick symmetrical 65A series airfoil, are deployed only at subsonic speeds and are kept at 0° incidence relative to the fuselage centre line. Positive canard deflections are with the trailing edge down. The take-off mass of the vehicle is 136,079 kg. The main geometric characteristics can be found in Figure 1 and Table 1.

The aerodynamic database was generated with the aerodynamic preliminary analysis system (APAS), a collection of algorithms to compute aerodynamic coefficients in the subsonic, supersonic and hypersonic velocity range. Also the control effectiveness and dynamic derivatives are output. Data are given as a function of Mach number, M, angle of attack and sideslip, α and β, the angular roll, pitch and

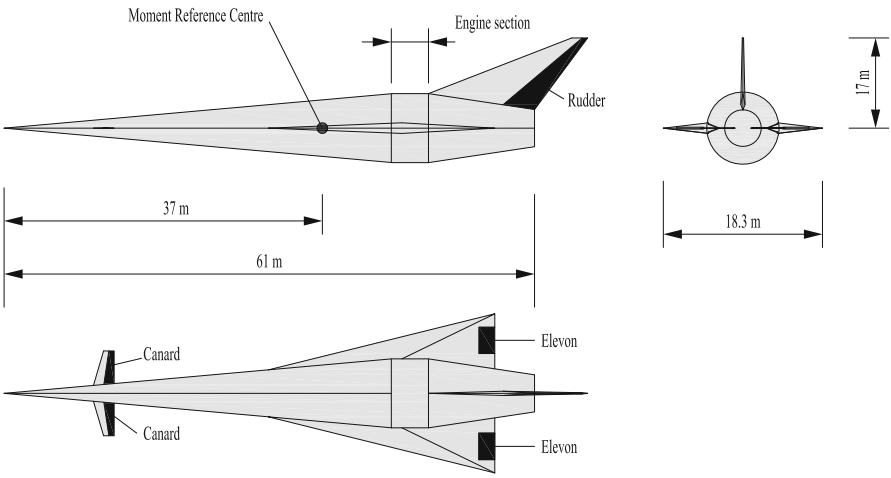


Figure 1 Geometry of the winged-cone configuration

Table 1 WCC geometric characteristics

Fuselage length	61.0 m
Cone half angle	5°
Cylinder radius	3.9 m
Cylinder length	3.9 m
Boat-tail half angle	9°
Boat-tail length	12.2 m
Moment reference centre from nose	37.8 m
Wing reference area	334.7 m^2
Aspect ratio	1.0
Span	18.3 m
Mean aerodynamic chord	24.4 m
Elevon area	8.6 m^2
Chord	2.2 m
Rudder area	15.0 m^2
Span	7.0 m
Chord (relative to tail)	25%
Canard area	14.3 m^2
Aspect ratio	5.48
Span	10.2 m

yaw rate, p, q and r, and control-surface deflection angles, and are defined with respect to the (fixed) moment reference centre.

The propulsion model was developed using a two-dimensional forebody, inlet and nozzle code with a one-dimensional combustor code. The thrust and specific impulse of the all air-breathing engine, T and I_{sp}, were determined as functions of M, dynamic pressure, $\bar{q}$, and fuel-equivalence ratio, ϕ_T. The effects of α, β, p, q, r and control-surface deflections on T and I_{sp} are assumed to be negligible for the current configuration of the WCC.

From the available documentation it is not clear what the geometry of the propulsion system is like. However, the Langley accelerator is known to have engine modules all around the cylindrical engine nacelle section. In principle, each of the inlets is equally effective at $\alpha = 0°$, since only in that case there is a symmetrical flow and none of the modules are shielded from the flow by the fuselage. Although the major part of the flight of the WCC will be at small α (and preferably at $\beta = 0°$), it is not known to what extent the assumption of independency on both α and β holds.

The vehicle mass model is based on a rigid structure and distributed fuel. No fuel slosh is considered. The mass properties, i.e., the total mass, the x-location of the centre of mass (c.o.m.) and the moments of inertia vary due to the consumption of fuel. They are modelled as a function of the current total mass. Note that the products of inertia are assumed to be negligible, even though the use of the inertia tensor is restricted to simulating rotational motion, which is not considered here.

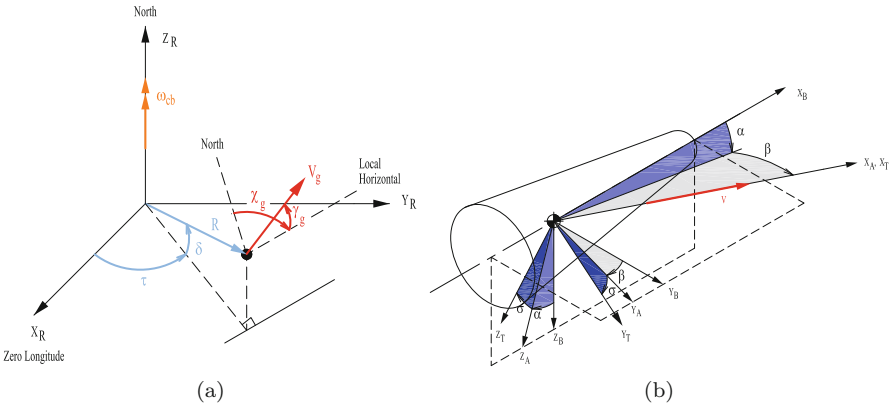


Figure 2 Definition of state variables. (**a**) Position and velocity. (**b**) Attitude

2.2 *Flight Dynamics*

To describe the flight dynamics we use the position and velocity definition in spherical coordinates, see Figure 2a. The position is defined by the distance $R = R_e + h$ (sum of Earth radius and altitude), longitude τ and latitude δ, whereas the velocity is expressed by its modulus, the groundspeed V and two direction angles, i.e., flight-path angle γ with respect to the local horizon, and heading, χ. For a flight to the north, the latter is defined as $\chi = 0°$ whereas $\chi = 90°$ for a flight to the east. The attitude of a vehicle, or, in mathematical terms, the orientation of the body-fixed reference frame with respect to the trajectory reference frame, is expressed by the so-called aerodynamic angles, i.e., the angle of attack α, the angle of sideslip β and the bank angle σ, see also Figure 2b. For the flight considered, the angle of sideslip is always $\beta = 0°$.

For a large portion of the mission, the ascent flight of an SSTO space plane can be assumed to take place in a single, vertical frame. As a consequence, in the absence of (horizontal) manoeuvres, the vehicle can fly with zero bank angle, yielding the maximum lift, L, in the vertical plane. Any control over the vertical motion will thus be achieved by angle-of-attack modulation, and could possibly be supported by using thrust-vector control (TVC). Nominally, the thrust force, T, will contribute to the vertical acceleration proportional to the (positive) flight-path angle.

To simplify the equations of motion, it is further assumed that this vertical plane coincides with the equatorial plane. With the spherical coordinates used, this means that the flight heading is equal to $\chi = 90°$ with $\dot{\chi} = 0°/\text{s}$, whereas the latitude remains constant at $\delta = 0°$. This assumption will effectively decouple the vertical from the horizontal motion and reduces the equations of motion to [17]:

$$\dot{V} = -\frac{D}{m} + T\cos(\alpha + \varepsilon_T) - g\sin\gamma + \omega_{cb}^2 R\sin\gamma \tag{1}$$

$$V\dot{\gamma} = \frac{L}{m} + T\sin(\alpha + \varepsilon_T) - g\cos\gamma + 2\omega_{cb}V + \frac{V^2}{R}\cos\gamma + \omega_{cb}^2 R\cos\gamma \quad (2)$$

$$\dot{R} = \dot{h} = V\sin\gamma \quad (3)$$

In the above equations D and L are the aerodynamic drag and lift in N, g is the gravitational acceleration in m/s^2 and ω_{cb} is the rotational rate of the Earth in rad/s. T is the *total thrust*, i.e., the sum of impulse and pressure thrust. The Coriolis effect due to variable mass is neglected, as it will only be non-zero in case the vehicle is rotating, and then only at a high rate.

It is noted that when a robust guidance system is to be developed the full set of equations of motion shall be used, to account for all non-linearities and asymmetries in the flight dynamics. As the focus of this study is to show the capabilities of parametric design and analysis, in combination with the use of a simple guidance system the chosen assumptions are assumed to be justified.

2.3 Guidance

Starting point for describing the guidance model is the notion that a reference trajectory is available, i.e., γ as a function of, for instance, h, V or M. This γ-profile is to be followed as closely as possible, thereby responding to deviations by controlling α. This so-called gamma-alpha steering is based on an inverse-dynamics approach, i.e., a prescribed $\dot{\gamma}$ yields α_c by solving the corresponding differential equation for γ. Furthermore, flying at the maximum values of the flight-path constraints maximum dynamic pressure, axial load and heat flux (so-called constraint tracking) will be considered, since that will result in principle in an optimal trajectory [8]. The flight-path constraints will be controlled by adjusting the thrust, which is discussed in more detail in the next subsection.

For the small angles of attack encountered during the major part of hypersonic flight, after inspecting the aerodynamic database a linear lift curve can be assumed, i.e., $L \approx (C_{L_0} + C_{L_\alpha}\alpha)\bar{q}S_{ref}$ [13]. If also the thrust elevation is small, then $\sin(\alpha + \varepsilon_T)$ can be approximated by $\alpha + \varepsilon_T$. After rearranging terms, a functional relationship between α and $\dot{\gamma}$ is obtained from Equation (2):

$$\alpha = K_{\gamma 1}\dot{\gamma} + K_{\gamma 2} + K_{\gamma 3} \quad (4)$$

with

$$K_{\gamma 1} = \frac{mV}{T + C_{L_\alpha}\bar{q}S_{ref}} \qquad K_{\gamma 2} = -\frac{T\varepsilon_T - mg\cos\gamma + C_{L_0}\bar{q}S_{ref}}{T + C_{L_\alpha}\bar{q}S_{ref}}$$

$$K_{\gamma 3} = \frac{2\omega_{cb} + \frac{V}{R}\cos\gamma}{T + C_{L_\alpha}\bar{q}S_{ref}}$$

In Equation (4) $K_{\gamma 1}$ is the actual $\dot{\gamma}$-to-α gain, whereas $K_{\gamma 2}$ can be seen as the trim value of α at which the thrust and aerodynamic forces balance the component of the gravity normal to the velocity vector, and thus at which $\dot{\gamma}$ is zero. The $K_{\gamma 3}$ term represents the influence of the Coriolis and centrifugal acceleration.

What is left is to compute a commanded $\dot{\gamma}$, which drives the actual γ to its reference value, γ_c. This is done in a general manner by applying a combination of proportional, integral and derivative control, the so-called PID family of control laws. Consider a time-varying error $e_\gamma(t)$, i.e., the difference between γ_c and γ. In proportional control, the output of the controller is simply related to its input by a proportional constant, $u(t) = K_{\gamma p}e_\gamma(t)$. Hence, a large *current* error signal will result in a large corrective action. The integral control law is represented by $u(t) = K_{\gamma i}\int_0^t e_\gamma(t)\mathrm{d}t$, with its output proportional to the accumulation of the *past* error. This can be very effective when the error has the same sign for most of the time. The derivative control law, finally, relates its output to the time derivative of the error signal, i.e., $u(t) = K_{\gamma d}\frac{\mathrm{d}e_\gamma}{\mathrm{d}t}$. A large slope of the error will give a large corrective action, implying that this control law anticipates on a large *future* error. With slowly varying errors, this control law is not very effective, but overall it improves stability. The flight-path angle steering loop is thus formulated as:

$$\dot{\gamma}_c = K_{\gamma p}e_\gamma + K_{\gamma i}\int_0^t e_\gamma \mathrm{d}t + K_{\gamma d}\frac{\mathrm{d}e_\gamma}{\mathrm{d}t}, \text{ with } e_\gamma = \gamma_c - \gamma \tag{5}$$

Without detailed analysis, the gains that give a satisfactory response are $K_{\gamma p} = 2.5$, $K_{\gamma i} = 1.8$ and $K_{\gamma d} = 0.05$ for Mach numbers lower than $M = 2$, $K_{\gamma p} = 2.0$, $K_{\gamma i} = 0.05$ and $K_{\gamma d} = 0.2$ for $2 < M \leq 4$, $K_{\gamma p} = 1.0$, $K_{\gamma i} = 0.9$ and $K_{\gamma d} = 0.01$ for $4 < M \leq 6$, and $K_{\gamma p} = 6.0$, $K_{\gamma i} = 2.0$ and $K_{\gamma d} = 0.2$ otherwise.

2.4 Throttle Control

To ensure mission success, it is important that the so-called flight-path constraints maximum dynamic pressure, maximum axial acceleration and maximum heat flux are satisfied. These constraints are, amongst others, a function of the velocity and since the velocity is directly dependent on the magnitude of the thrust, the throttle setting can be used to regulate the velocity. Successively, a feedback control law to regulate each of the three mentioned constraints will be developed that is easy to implement and is robust enough to account for sudden changes in flight path.

The throttle-control law for regulating $\bar{q}$ is derived from the expression of $\dot{\bar{q}}$, which is given by

$$\bar{q} = \frac{1}{2}\rho V^2 \quad \Rightarrow \quad \dot{\bar{q}} = \rho V \dot{V} + \frac{V^2}{2}\dot{\rho} \tag{6}$$

Since the trajectory of the space plane is shallow ($\gamma \approx 0$), flown at α close to zero, for guidance purposes $\dot{V}$ can be approximated from Equation (1):

$$\dot{V} = \frac{T - D}{m} \tag{7}$$

Furthermore, the atmospheric density is assumed to be exponential so that the density variation with time can be written as

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = \frac{\mathrm{d}\rho}{\mathrm{d}h}\frac{\mathrm{d}h}{\mathrm{d}t} = -\frac{\rho}{H_s}\dot{h} \tag{8}$$

where H_s is the density scale height ($H_s = 7050\,\mathrm{m}$ for the Earth's atmosphere). Substitution of Equations (7) and (8) into Equation (6) yields

$$\dot{\bar{q}} = \frac{\rho V}{m}T - \frac{\rho V}{m}D - \frac{\rho V^2}{2H_s}\dot{h} \tag{9}$$

For a flight along the dynamic-pressure constraint, the above rate is zero yielding the commanded thrust, T_c, to maintain equilibrium. However, to compensate for variations in $\bar{q}$ corrective terms must be added to T_c. Considering a PI regulator, T_c can be written as

$$T_c = D + \frac{mV}{2H_s}\dot{h} + K_{\bar{q},p}e_{\bar{q}} + K_{\bar{q},i}\int_0^t e_{\bar{q}}\mathrm{d}t \tag{10}$$

with $e_{\bar{q}} = \bar{q} - \bar{q}_{max}$. Substituting the above equation into Equation (9) gives an expression for the rate of change of the error in $\bar{q}$, i.e.,

$$\dot{e}_{\bar{q}} = \frac{\rho V}{m}\left(K_{q,p}e_{\bar{q}} + K_{\bar{q},i}\int_0^t e_{\bar{q}}\mathrm{d}t\right) \tag{11}$$

To determine the feedback gains, the above equation is Laplace transformed to the s-domain:

$$\left(s^2 - \frac{\rho V}{m}K_{\bar{q},p}s - \frac{\rho V}{m}K_{\bar{q},i}\right)e_{\bar{q}}(s) = 0 \tag{12}$$

which yields the following second-order characteristic equation:

$$s^2 + 2\zeta_{\bar{q}}\omega_{\bar{q}} + \omega_{\bar{q}}^2 = 0 \tag{13}$$

Combining the two equations yields for the proportional and integral gain:

$$K_{\bar{q},p} = -\frac{2m}{\rho V}\zeta_{\bar{q}}\omega_{\bar{q}} \quad \text{and} \quad K_{\bar{q},i} = -\frac{m}{\rho V}\omega_{\bar{q}}^2 \tag{14}$$

By specifying the damping coefficient, $\zeta_{\bar{q}}$, and natural frequency, $\omega_{\bar{q}}$, the (dynamics-dependent) gains can easily be determined.

In a similar manner, a throttle-control law is derived for tracking a maximum stagnation-point heat flux, approximated by Chapman's equation:

$$\dot{Q} = \frac{c_{\dot{Q},1}}{\sqrt{r_N}}\sqrt{\frac{\rho}{\rho_0}}\left(\frac{V}{V_c}\right)^{c_{\dot{Q},2}} \tag{15}$$

where $c_{\dot{Q},1} = 1.06584 \cdot 10^8\,\mathrm{W/m^{3/2}}$ and $c_{\dot{Q},2} = 3$ are constants, r_N is a characteristic radius, $\rho_0 = 1.225\,\mathrm{kg/m^3}$ is the Earth's atmospheric density at sea level and V_c is the local circular velocity with respect to the rotating frame. Taking the time derivative of Equation (15) results in

$$\ddot{Q} = \dot{Q}\left(c_{\dot{Q},2}\frac{\dot{V}}{V} - \frac{\dot{h}}{2H_s}\right) = \dot{Q}\left(c_{\dot{Q},2}\frac{T-D}{mV} - \frac{\dot{h}}{2H_s}\right) \tag{16}$$

T_c is written as the sum of an equilibrium value, derived from Equation (16), and the PI compensation terms:

$$T_c = D + \frac{mV}{2c_{Q,2}}\frac{\dot{h}}{H_s} + K_{\dot{Q},p}e_{\dot{Q}} + K_{\dot{Q},i}\int_0^t e_{\dot{Q}}\mathrm{d}t \tag{17}$$

with $e_{\dot{Q}} = \dot{Q} - \dot{Q}_{max}$. Substituting T_c in Equation (16) yields an expression for the rate of change of the error in heat flux. After Laplace transforming the resulting equation, and equating it with a second-order characteristic equation similar to Equation (13), the following proportional and integral gains are obtained:

$$K_{\dot{Q},p} = -\frac{2mV}{c_{\dot{Q},2}\dot{Q}}\zeta_{\dot{Q}}\omega_{\dot{Q}} \quad \text{and} \quad K_{\dot{Q},i} = -\frac{mV}{c_{\dot{Q},2}\dot{Q}}\omega_{\dot{Q}}^2 \tag{18}$$

Finally, given the axial acceleration and its time derivative by

$$n_a = \frac{\dot{V}}{g_0} \quad \Rightarrow \quad \dot{n}_a = \frac{\ddot{V}}{g_0} = \frac{\dot{T}-\dot{D}}{mg_0} - \frac{T-D}{m^2 g_0}\dot{m} \tag{19}$$

Table 2 Parameter values for guidance and throttle control

Parameter	Value
ζ_q	0.7
ω_q	0.75 rad/s
$\zeta_{\dot{Q}}$	0.7
$\omega_{\dot{Q}}$	0.075 rad/s
ζ_{n_a}	0.7
ω_{n_a}	0.075 rad/s

with g_0 the acceleration due to gravity at the equatorial sea level ($g_0 = 9.798\,\mathrm{m/s^2}$ for Earth), T_c is written as

$$T_c = D + \frac{\dot{m}}{m}(\dot{T} - \dot{D}) + K_{n,p} e_{n_a} + K_{n,i} \int_0^t e_{n_a} \mathrm{d}t \tag{20}$$

with $e_{n_a} = n_a - n_{a_{max}}$. For $K_{n,p}$ and $K_{n,i}$ it follows:

$$K_{n,p} = \frac{2m^2 g_0}{\dot{m}} \zeta_{n_a} \omega_{n_a} \quad \text{and} \quad K_{n,i} = \frac{m^2 g_0}{\dot{m}} \omega_{n_a}^2 \tag{21}$$

The applied values for the damping ratio and natural frequency to give a good response and an acceptable overshoot are summarised in Table 2.

3 Parametric Design and Analysis

Generally, in a sensitivity analysis or a design exploration one wants to cover the full experimental region with a minimum number of simulations. When no details on the functional behaviour of the response parameters are available, it is important to obtain information from the entire design space. Therefore, design points should be "evenly spread" over the entire region. Without degrading the method, one tends to reach out to a Monte-Carlo simulation. Because the design space can be sampled at random and once sufficient runs have been executed, one can get a reliable insight in the mean performance and associated standard deviation. However, the keyword here is *sufficient*. It might take a significant number of runs to obtain statistical confidence, and if the runs are time consuming to prepare and/or execute a Monte-Carlo simulation might not be the most efficient approach.

A potential alternative comes from the field of design of experiments,[2] where with a relatively low number of simulations significant insight in the system

[2]Various alternatives to design of experiments exist, e.g., Latin hypercube sampling [18] or optimised hypercube sampling [19]. Even though potentially efficient methods, it was decided to

Table 3 Orthogonal array L_8 with 7 factors (A–G) on two levels; "−1" represents the normalised minimum value and "1" the maximum one

Design nr	A	B	C	D	E	F	G
1	−1	−1	−1	−1	−1	−1	−1
2	−1	−1	−1	1	1	1	1
3	−1	1	1	−1	−1	1	1
4	−1	1	1	1	1	−1	−1
5	1	−1	1	−1	1	−1	1
6	1	−1	1	1	−1	1	−1
7	1	1	−1	−1	1	1	−1
8	1	1	−1	1	−1	−1	1

performance can be obtained. This methodology was earlier used for screening experiments and shape optimisation of re-entry test vehicles [20], and would also be suitable to analyse the reference trajectory of the SSTO space plane. Rather than executing a full factorial design, i.e., varying one parameter at a time and executing all combinations, rapidly leads to a large number of simulations,[3] it is preferred to do a so-called fractional factorial design. A good candidate is the Taguchi method [21, 22], which makes use of *orthogonal arrays* to define parameter-setting combinations. Matrix orthogonality, in this context, should be considered in the combinatorial sense, namely: for any pair of columns all combinations of parameter levels occur an equal number of times, the so-called *balancing property* [22]. In the field of design of experiments, a parameter (a design variable, sensitivity parameter, uncertainty, etc.) is commonly known as a *factor*. Similarly, the performance of the system under study (or, equivalently, the deviation from a set point, a constraint value, or anything that says something about the system's behaviour) is called the *response* of the system.

Taguchi [21] has derived many orthogonal arrays, most of them based on two- or three-level factors, which are commonly used in practical applications. As an example, the so-called L_8 array is given in Table 3 (note that the index "8" indicates the number of rows, or, similarly, the number of designs/experiments). Seven two-level factors (A through G), with levels −1 (normalised minimum value) and 1 (normalised maximum value) are varied over eight experiments. For columns 1 and 2, the 4 possible combinations of factor levels, i.e., (−1,− 1), (−1,1), (1, − 1) and (1,1), occur in experiments (1,2), (3,4), (5,6) and (7,8), respectively. Note that in a full factorial design 2^7(=128) experiments would be required. It is also mentioned that the L_8 design is non-collapsing.[4]

use a fully deterministic approach that is simple to implement and allows for a structured variance analysis of the results.

[3] Variation of k parameters with two (three) possible values, also called levels, results in a total of 2^k (3^k) combinations.

[4] Two design points are said to *collapse* when one of the design parameters has (almost) no influence on the function value and the two designs differ only in this parameter. As a consequence

Although there are many orthogonal arrays available from literature, the selection of the proper orthogonal array is not trivial. This is particularly true if many factors are included and there are potential interactions (an *interaction* between two factors is said to exist, when a variation in the first factor results in a different variation of the response for each level of the second factor). Then, the column assignment can be complicated and it is possible that not all of the factors and interactions can be studied in one go. The columns that are assigned to interactions are available from so-called interaction tables [22]. For the array given in Table 3, if one assigns two factors to columns A and B, then a potential interaction is linked with column C. Similarly, column A and D are linked to E, whereas B and D are linked to column F. The particular interaction tables follow from the mathematical derivation of the corresponding orthogonal array, and is not a trivial process. Discussing the interactions in detail does not serve a purpose here, so we leave it to the given references that provide a lot of background information.

A statistical description of a number of N observations of a response can be given by the mean response, $\bar{y}$, and its standard deviation, σ, defined by:

$$\bar{y} = \frac{1}{N}\sum_{j=1}^{N} y_j = \frac{T}{N} \quad \sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2 \tag{22}$$

where $T = \sum_{j=1}^{N} y_j$ is the total sum. To identify the relative "strength" of factors to the total variation of a response, analysis of variance (ANOVA) can be done, simplified by the fact of using orthogonal arrays [21]. The sum of the squared deviation from this mean (or the total variation in the set of observations) is represented by the *total sum of squares*, S_T.

$$S_T = \sum_{i=1}^{N}(y_i - \bar{y})^2 = \sum_{i=1}^{N}\left(y_i^2 - 2y_i\bar{y} + \bar{y}^2\right) = \sum_{i=1}^{N} y_i^2 - 2\bar{y}\sum_{i=1}^{N} y_i + N\bar{y}^2 \tag{23}$$

Because there is an equal number of experiments n_k at each of the levels for one factor x_j when they are varied according to an orthogonal array, it is easy to compute the factor sum of squares, S_j. The sum of squares S_j for factor x_j on levels $x_{j,k}$ is the sum of all level variations:

$$S_j = \frac{1}{n_L}\frac{\left(\sum^{N} y(x_j^+) - \sum^{N} y(x_j^-)\right)^2}{n_k} \tag{24}$$

this means that effectively the same point is evaluated twice, and for deterministic simulation models this is not a desirable situation.

with each sum adding all responses at level k ($+1$ or -1) together (n_k out of the total of N samples); n_L is the number of levels ($n_L = 2$ for the linear variations considered in this study). This factor sum of squares can be used to assess the response sensitivity to a particular factor variation by determining the corresponding relative contribution to S_T:

$$P_j = \frac{S_j}{S_T} * 100\% \tag{25}$$

The error sum of squares S_E is simply the difference between S_T and the sum of all S_j ($j = 1, \ldots, k$).

An alternative way to study the factor effects is by fitting a response surface through the data, and by comparing the relative contribution of each of the terms. For example, to study quadratic effects, a second-order response surface may be used:

$$\eta = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{ij} x_i x_j \tag{26}$$

for which the coefficients β can be estimated by minimising a quadratic error criterion, resulting in the well-known method of least squares. Solution of the resulting problem can conveniently be done by, for instance, singular value decomposition. In Equation (26), the mean value of the particular response is represented by β_0, whereas the relative values of the other coefficients give the sensitivity of this mean response to a variation in the individual factors and interactions.

4 Vertical Ascent to Orbit

4.1 Initial Trajectory

At the beginning of this study, an all-up reference trajectory for an SSTO space plane was not readily available, nor was there an opportunity to use optimisation software to generate one. Based on experience gained by Marée et al. [23], it is possible, however, to roughly define a trajectory based on available literature and then to further refine the trajectory using Taguchi's orthogonal arrays. This implies that a sensitivity analysis is executed where the key parameters of the trajectory are the factors, and the payload mass delivered to orbit or the integrated heat load are the responses. The outcome of the analysis will be a sub-optimal trajectory *for the given definition of the trajectory parameters*.

A trajectory can be defined by specifying the flight-path angle as a function of altitude. By comparing guidance and optimisation studies by Hattis and Malchow [14], Powell et al. [15], Van Buren and Mease [8], and Lu [10], it was found that

the ascent trajectories include a take-off segment consisting of a pull-up manoeuvre towards a large γ between 25° and 30° in the subsonic range, and then falls back to values close to zero not to violate the trajectory constraints and for subsequent flight acceleration. Van Buren and Mease [8] showed that tracking the trajectory constraints $\bar{q}$, n_a and $\dot{Q}$ results in a near minimum-fuel trajectory for the applied vehicle (WCC) and propulsion models. Powell et al. [15] stated that $\bar{q}$ is the most stringent constraint, and Lu [10] applied both $\bar{q}$ and $\dot{Q}$ as trajectory constraints. To be in line with literature the three mentioned constraints will be applied and the importance of each of these constraints will be judged later. Moreover, the trajectory design will be based on the trajectory discussed by Powell et al. [15]. The design logic is as follows.

In principle, the WCC flies with full throttle to optimise propulsion-system performance. By varying ϕ_T (under- and overfueling) less or more thrust can be generated, if required. Constraint tracking could thus be done by varying ϕ_T, but since the propulsion model has been implemented for $\phi_T = 1$, the engines will be throttled directly. A maximum δ_T of 100% is used, which implies that the maximum thrust could be higher if overfueling is allowed. The γ-profile is specified as a function of h and includes segments with constant γ_c or constant $\dot{\gamma}_c$. The commanded γ is input to the gamma-alpha guidance system as discussed in Sect. 2.3, giving a commanded α.

By trial and error we come to a definition of the segments such that a circular LEO at 120 km altitude can be reached (see Table 4 for the trajectory segmentation).

Table 4 Initial reference trajectory

Nr	Guidance law	Parameter	Segment boundary	ΔT (sec)	Δm_{fuel} (kg)
1	Constant normal load	$n_{z,c} = 1.5$	$\gamma_1 = 25.0°$	14	849
2	Constant flight-path angle rate	$\dot{\gamma}_{c,2} = -0.2°/s$	$\gamma_2 = 5.0°$	101	6208
3	Constant flight-path angle rate	$\dot{\gamma}_{c,3} = -0.075°/s$	$\gamma_3 = 3.5°$	18	824
4	Constant flight-path angle rate	$\dot{\gamma}_{c,4} = -0.025°/s$	$\gamma_4 = 1.5°$	94	3686
5	Constant flight-path angle	$\gamma_{c,5} = 1.5°$	$M_5 = 7.0$	114	5049
6	Constant flight-path angle rate	$\dot{\gamma}_{c,6} = -0.025°/s$	$\gamma_6 = 0.5°$	61	3041
7	Constant flight-path angle	$\gamma_{c,7} = 0.5°$	$M_7 = 10.0$	103	4721
8	Constant flight-path angle rate	$\dot{\gamma}_{c,8} = -0.01°/s$	$\gamma_8 = 0.25°$	46	2062
9	Constant flight-path angle	$\gamma_{c,9} = 0.25°$	$M_9 = 14.0$	229	10,651
10	Constant flight-path angle rate	$\dot{\gamma}_{c,10} = -0.001°/s$	$\gamma_{10} = 0.12°$	151	7932
11	Constant flight-path angle	$\gamma_{c,11} = 0.12°$	$V_{11} = 7000$ m/s	905	19,879
12	*Pull-up manoeuvre*				
	Initial	$\alpha_{i,12} = 6.0°$	$\alpha_{12} = 0.0°$	83	767
	Transition	$\bar{q}_{12} = 2500$ N/m^2	–	–	–
	Final	$\alpha_{f,12} = 0.0°$	–	–	–
13	Constant angle of attack	$\alpha_c = 0.0°$	$\gamma_{13} = 0.0°$	316	0
14	Circularisation	$h_{circ} = 120$ km	$h_{circ} = 120$ km	0	7740

Transition from one segment to the other is triggered by reaching, for instance, a predefined γ, M or V. The final pull-up manoeuvre is started at $V \approx 7000$ m/s, a value close to values found in the literature, and is performed at $\delta_{T,max} = 120\%$. After reaching a specified $\bar{q}$ the vehicle is commanded to fly at $\alpha = 0°$, which corresponds with the minimum drag configuration, and the engines are switched off. A coast phase will bring the vehicle in orbit. Obviously, it will not have the proper final state since no targeting guidance is used. Powell et al. [15] guarantee this final state by applying a predictor–corrector guidance system as outer loop. It determines the proper conditions for starting the final pull-up, such that only minor corrections have to be applied by the circularisation engine. Since we do not have a targeting guidance system at our disposal, the final altitude is considered to be the only final state and circularisation of the target orbit is added as a mass penalty. This penalty is approximated by Tsiolkovsky's equation, using $I_{sp} = 465$ s for the circularisation engine [15]. After circularising the orbit, γ will be zero and V will be the local circle velocity at 120 km.

The trajectory represents a trimmed flight in the equatorial plane, starting with $V = 170$ m/s. Trim is guaranteed by the canards ($M \leq 0.9$) and elevons ($M > 0.9$). TVC is not considered for the reference trajectory, although the influence of TVC on fuel-mass consumption will be studied later. When $\bar{q}$ is low, i.e., during the final ascent to orbit, $\alpha = 0°$ and trim can be restricted to corrective control by reaction-control jets. The applied guidance system is the already mentioned gamma-alpha steering, with constraint tracking by the throttle laws of Sect. 2.3.

The trajectory constraints are $\bar{q}_{max} = 95{,}000\,\text{N/m}^2$, $\dot{Q}_{\max} = 8000\,\text{kW/m}^2$ at the leading edge of the wing ($r_N = 0.1$ m), and $n_{a,max}$ due to propulsion and aerodynamic forces of 1 g_0. These values are commonly applied to SSTO space planes, see the previously mentioned references. The resulting initial reference trajectory consists of the segments with corresponding parameters and fuel-mass consumption, as indicated in Table 4.

The resulting trajectory has been visualised in Figure 3, with a total flight time of $t_f = 2234$ s. It shows that the vehicle first flies along the dynamic-pressure constraint—although early in the flight this could be improved, provided that the axial-acceleration constraint is not violated—followed by flying along the heat-flux constraint, which kicks in at higher velocities. The powered ascent phase is characterised by a relatively flat trajectory; the major increase in altitude takes place after the pull-up, with an unpowered coasting phase. The circularisation manoeuvre is executed when the flight-path angle is zero, which, in this case, is at an altitude larger than 120 km. This means that effectively first the apogee height is lowered to 120 km, after which the velocity is increased to the local circle velocity.

Before we come to optimising the initial reference trajectory, some remarks must be made. The final relative velocity is $V_f = 6879.4$ m/s, a rather large difference with the local circle velocity ($V_c = 7358.2$ m/s). A properly selected guidance scheme might therefore reduce the total required fuel mass. The circularisation manoeuvre required an inertial ΔV of 479 m/s ($\Delta m_{fuel} = 7740$ kg, including the fuel for lowering the apogee) as compared with $\Delta V = 162$ m/s ($\Delta m_{fuel} =$

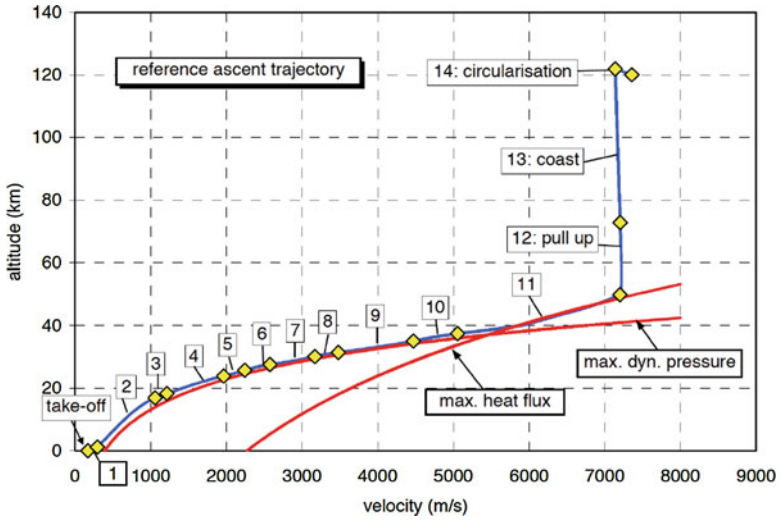


Figure 3 The ascent reference trajectory, including the dynamic-pressure constraint of $\bar{q}_{max} = 95{,}000\,\mathrm{N/m^2}$ and the heat-flux constraint of $\dot{Q}_{max} = 8000\,\mathrm{kW/m^2}$. The flight-segment boundaries are represented by diamonds

2487 kg), mentioned by Powell et al. [15]. The final mass after circularisation is 62,668 kg, which—after subtracting the vehicle's dry mass—means a maximum payload of 3700 kg.

Previously, it was stated that the performance of a space plane is marginal at best, and that small deviations from the nominal design could easily lead to a negative payload and thus mission failure. Here, however, we could relatively easily find a successful trajectory with a positive payload, although not as much as the commonly specified target payload of 7000 kg. The reason may be found in a too optimistic modelling of the WCC, in the sense that the performance of the propulsion system is overrated with respect to the vehicle's dry mass.

During the simulations it was found that the discrete jumps in flight-path angle commands caused peaks in the commanded angle of attack, especially after the initial pull-up to $\gamma = 25°$ and the successive commanded flight-path angle rate. To avoid these peaks—and during the sensitivity analysis sometimes even instabilities—any $\dot{\gamma}_c$ that is input to the guidance law will first be converted to a γ_c and subjected to the guidance law after which the resulting rate command is constrained to twice its input value.

4.2 Trajectory Analysis

As can be seen from Table 4, there are 28 parameters that define the trajectory. However, not all of them are suitable to be varied: (1) the γ-boundary of a segment preceding a constant-γ segment should in principle be equal to γ_c to prevent additional discontinuities, (2) α_f of the pull-up should be equal to α_c of segment #13 *and* zero, because $\alpha = 0°$ corresponds with the minimum-drag configuration which is already optimal, (3) the boundary altitude of the last segment is the (fixed) target altitude. Omitting these leaves 20 parameters.

Marée et al. [23] discuss the shaping process of an air-breathing first stage and rocket-powered second stage of a TSTO space plane by varying the important trajectory parameters according to an orthogonal array. For 13 selected trajectory parameters, a three-level L_{36} array taken from [21] was applied. Note that in principle, an L_{27} array would have sufficed for 13 factors; however, 9 simulations were added for the sake of error analysis. The outcome of the sensitivity analysis was that the payload-to-orbit varied between -1890 and $+5820$ kg. Analysis of variance of the payload mass indicated that the contribution of the quadratic terms was only 0.45%. Unfortunately, no effort was made to assess the influence of interactions although the results implied that this influence was small.

In line with previous experience with Taguchi's orthogonal arrays, we will start the analysis with a two-level array to assess the linear effects of the trajectory parameters. Nineteen parameters require the L_{32} array, leaving 12 columns for interactions. However, if it can indeed be assumed that there are no interactions, the number of simulations can be reduced if beforehand it can be decided that some parameters are not so important. On the other hand, we only want to indicate the potential of this way of trajectory shaping. Therefore, it is decided to use the L_{16} array. This means that 15 out of the 19 parameters will be varied. The four parameters that are discarded are the $\dot{\gamma}_c$ of segments #2, #3, #4 and #10, because the trajectory shape can also be influenced by the corresponding γ-boundaries.

It can be expected that not all of the combinations of trajectory parameters will result in actual trajectories all the way up to orbit. This has already been solved by incorporating the circularisation manoeuvre. The stop criterion for all simulations will be a zero flight-path angle after the coasting phase. In any case, one burn will be applied to raise or lower the final altitude and another burn for circularisation, as was discussed in the previous sub-section. The parameter variations are selected quite arbitrarily, but should guarantee that the vehicle can reach orbit. They have been defined as follows.

With the above parameter variations, not all of the trajectories are feasible ones, though, but this has been done intentionally to show a point later in this section. Two aspects came to light: three trajectories led to a crash of the vehicle,[5] and one trajectory gave a negative payload mass after circularisation. The latter is explained

[5]The simulation stops after 3000 s, which is about 700 s more than the flight duration of the reference trajectory, so not reaching $h = 120$ km after this time indicates a vehicle "crash".

A: $n_{z,c} = 1.5 \pm 0.15$
B: $\gamma_1 = 25.0° \pm 1.0°$
C: $\gamma_2 = 5.0° \pm 0.25°$
D: $\gamma_3 = 3.5° \pm 0.25°$
E: $\gamma_4 = \gamma_{c,5} = 1.5° \pm 0.15°$
F: $M_5 = 7.0 \pm 0.5$
G: $\gamma_6 = \gamma_{c,7} = 0.5° \pm 0.05°$
H: $M_7 = 10.0 \pm 0.5$
I: $\dot{\gamma}_{c,8} = -0.01°/\text{s} \pm 0.005°/\text{s}$
J: $\gamma_8 = \gamma_{c,9} = 0.25° \pm 0.05°$
K: $M_9 = 14.0 \pm 0.5$
L: $\gamma_{10} = \gamma_{c,11} = 0.12° \pm 0.02°$
M: $V_{11} = 7000\,\text{m/s} \pm 50\,\text{m/s}$
N: $\alpha_{i,12} = 6.0° \pm 0.5°$
O: $\bar{q}_{12} = 2500\,\text{N/m}^2 \pm 1000\,\text{N/m}^2$

as follows. The way the simulation is executed allows the propulsion system to keep thrusting, even though the fuel mass has depleted. This has been done to widen the possibilities to reach space and study the main performance effects. Using more fuel than is available will automatically lead to a negative payload. Keeping the three crash trajectories in the responses, the variance analysis will be influenced, because these have a relatively large weight on both payload mass (which will be too high, because not all fuel would have been used) and integrated heat load (too low, since the heat flux acting on the vehicle will be significantly lower, which dominates the longer flight duration). The same is true for a negative payload mass, although this effect is smaller, because the trajectory is "more feasible" and it concerns only a single case. It is noted that when relatively more negative payloads are present, the predictive power of the response surface of the payload reduces. This is attributed to the larger spread of the responses, which is a form of non-linearity not captured by the currently considered linear factor variation. (The use of response surfaces will be explained in more detail later in this section).

For a more robust analysis that would lead to an improvement of the trajectory it is therefore required to have feasible trajectories with preferably positive payload masses only. To achieve this the factor variation has to be lowered to 75%. One should be aware, though, that reducing the factor variation obviously reduces the payload and integrated heat-load variation. Furthermore, also the factor contribution to the total variance may change, which would confirm the non-linearity of the problem.

Executing the 16 simulations resulted in the payload masses as presented in Table 5. There, also the integrated heat loads are listed. The higher this heat load is, the higher the mass of the thermal protection system, which would in a proper design process have a negative impact on the available payload mass. After inspecting the results it is obvious that the minimum-fuel trajectory corresponds with the minimum heat-load trajectory, but only in this particular case. Normally, these two responses are conflicting, as will be shown below. Depending on the maximum allowable value of the latter, it should be included in any trajectory optimisation process.

Focusing on the payload mass a large variation can be seen, i.e., it ranges from 732 to 6584 kg (the integrated heat load shows less of a variation). This variation follows a normal distribution, as shown in Figure 4. To assess the relative influence of the trajectory parameters on the payload mass, a variance analysis is performed using Equations (22) through (26). In Figure 5, the linear factor contribution to the

Table 5 Payload mass and total heat load for sensitivity analysis (extrema are highlighted in bold)

Nr	$m_{payload}$ (kg)	Q (MJ/m^2)	Nr	$m_{payload}$ (kg)	Q (MJ/m^2)
1	**732**	**11,526**	9	1602	11,433
2	4650	8798	10	3291	8818
3	3942	9907	11	1886	9466
4	4574	9447	12	2883	10,694
5	4684	9556	13	3975	10,606
6	5752	9798	14	4206	9395
7	**6584**	**8736**	15	4420	8936
8	3117	11,189	16	2713	11,319

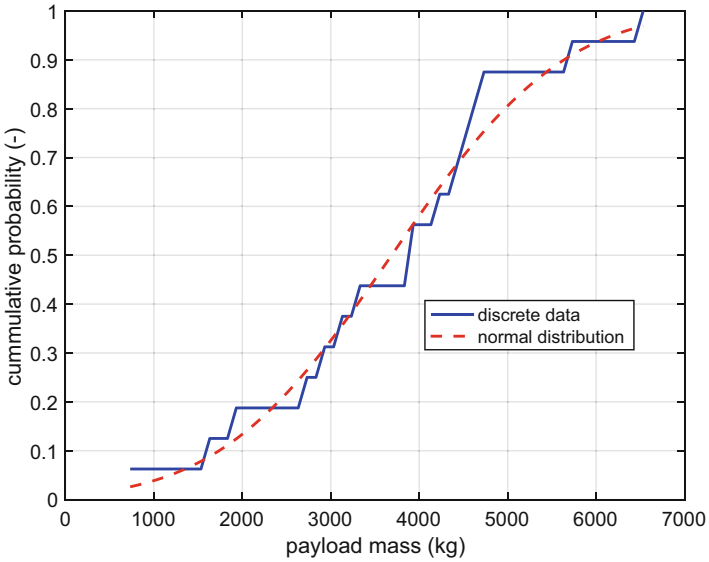


Figure 4 Cumulative probability distribution of the payload-mass variation: discrete data versus normal distribution with $\mu = 3688$ kg and $\sigma = 1521$ kg

total variance, Equation (24), has been plotted, for both the payload mass and the total heat load. Since all columns of the orthogonal array were assigned to factors, no column (or, equivalently, degree of freedom) was left to address the error variance. Note that any variation due to interactions is confounded with the factor variance, although the contribution of the interaction will be smaller than the corresponding main effects. As a result of the analysis with the payload mass as objective, we find that four factors including possible interactions contribute to about 82% of the variation in payload mass,[6] i.e.,

[6]Only two factors determine 90% of the variation in integrated heat load, i.e., parameters that determine the shape of the second part of the trajectory where thermal loading is larger.

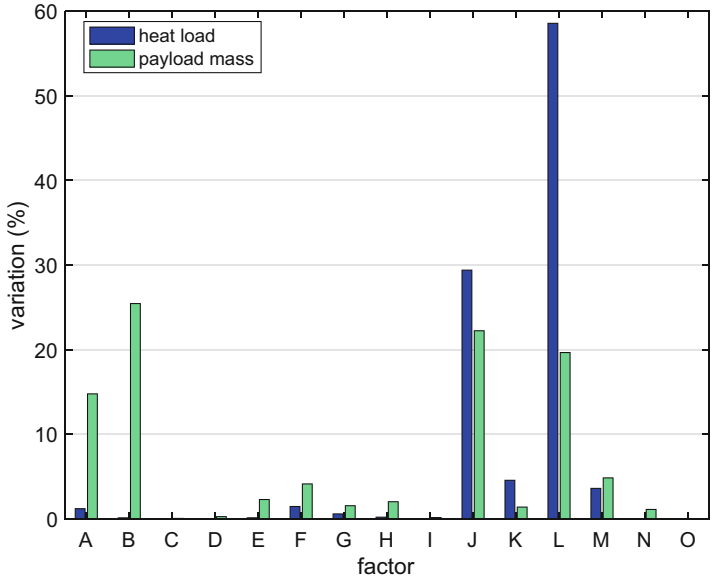


Figure 5 The contributions of the linear terms of variation per factor, in percents of the total variation, to the payload mass and total heat load

1. the normal load at take-off, factor A (14.8%),
2. the flight-path angle at the end of the initial pull-up, factor B (25.4%),
3. the commanded flight-path angle of segment 9, factor J (22.2%) and
4. the commanded flight-path angle of segment 11, factor L (19.7%).

The contribution of the other factors is smaller, though not negligible, apart from factors C, D, I and O that have an influence smaller than 1%. Note that only the *relative* influence of the factors is given. Since the factors are not equally varied it cannot be concluded that the trajectory is more sensitive to one factor than to another. To verify that, the payload mass is plotted as a function of the factors related to their mean value (Figure 6), i.e., the mean value of the fuel mass at the minimum level of a factor and the maximum level. Connecting these two values gives gradient information, and shows which of the factors have had the most impact on the variance of the responses. For ease of inspection, in Figure 7 the factor-response gradients of the four largest contributors (A, B, J, L) have been plotted. From this figure it is clear that the factor that contributes most to the total variation (B) also has the steepest gradient, but that the factor that comes close (J) has a much shallower gradient. Moreover, it is also obvious that a larger value of factor A has a decrement of the payload mass as result, while for the other three factors an increase leads to an increment.

The two ways of graphically presenting the results can be of great help while trying to interpret the results of a sensitivity analysis. In case the factor levels are

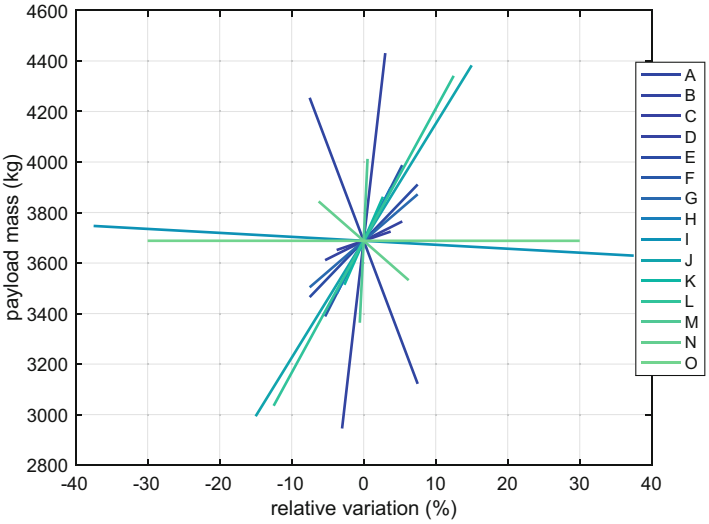


Figure 6 Relative factor influence on the variation of payload mass. The factor variation is expressed with respect to the corresponding nominal value

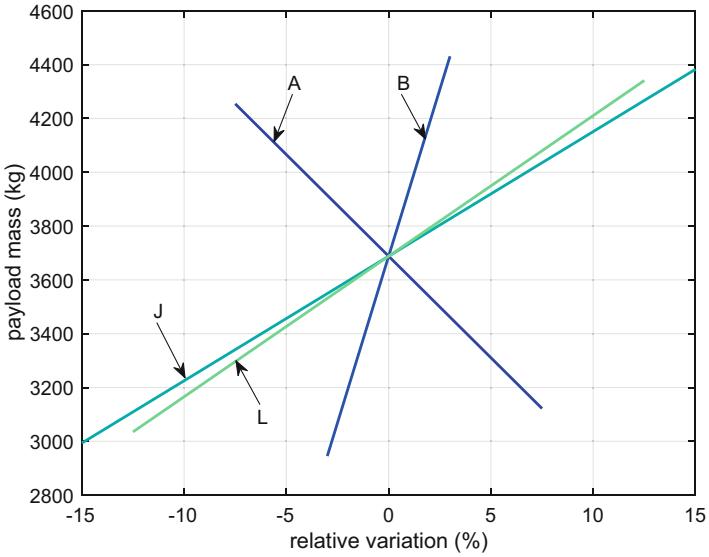


Figure 7 The influence on the payload mass of the four factors that contribute most to the total variation. The factor values are expressed with respect to their nominal values

given, then the factor contribution to the total variation directly gives the major factors, and relative differences between the factors are easily found from the bar graph. On the other hand, in case the factor levels are chosen quite arbitrarily

studying the response lines can give a lot of extra insight, especially when the factors are varied over more than two levels. These results can then be used to change individual factors to try and improve the responses.

To assess the influence of the simulation size on the selected responses and to have some columns available for interaction analysis to confirm that they can indeed be ignored, a second batch of simulations is executed with twice the number of runs. For the 15 factors an L_{32} array is used with a column assignment of alternating a factor with an empty column: A, empty, B, empty, C, empty, etc. ANOVA of the results shows that the contribution of the empty columns (i.e., column interactions) to the total variation is very small: 0.2% for the payload mass and 1.1% for the integrated heat load.

An aspect that came forward by comparing the results of the two analyses is that there are small changes in factor contribution to the total variation, although they remain small in this particular case. Major factor effects are still major factor effects, but in case of smaller effects there are some small shifts in the order of the factor contributions. The conclusion is that major factor effects can always be extracted from the results, but that the absolute importance cannot be fully assessed. This means that one cannot label a certain factor with a $p\%$ influence on the total variation. In particular, this holds for minor factor influences.

Concerning the extrema, the maximum-payload trajectory has $m_{payload} = 6353\,\text{kg}$, which is 131 kg less than for the L_{16}, with $Q = 9221\,\text{MJ/m}^2$, whereas the minimum heat-load trajectory has $Q_{min} = 8242\,\text{MJ/m}^2$ with $m_{payload} = 5367\,\text{kg}$. As mentioned earlier, the two "optimal" trajectories are definitely not the same, confirming the conflicting objectives.

Finally, the L_{16} and L_{32} analyses resulted in two different means (3688 and 3647 kg) and standard deviations (1521 and 1413 kg) for the payload mass. Evidently the problem of trajectory (sub)optimisation is a non-linear problem that emerges when different (but similar) factor combinations are used. Of importance is, of course, the question whether one mean is better than the other, or whether the two means differ significantly from one another. One can imagine that a mean gets more accurate with an increasing number of samples. In that sense, the mean obtained by the L_{32} analysis would be better. An even more global question is: are the two distributions of the payload masses consistent with each other or are they different? Statistical tests are available that can address these questions in a formal manner. For instance, the generally accepted two-sample Kolmogorov–Smirnov test can be applied to continuous distributions to check whether they are consistent or different, as to find out whether the two sets of orthogonal factor combinations that are in principle normally distributed result in the same distribution. Carrying out the test accepts the null hypothesis at the default 5% significance level and confirms that the distributions are consistent. Apparently, only the number of simulations has an influence on the mean and variance of a selected response.

Before moving on to finding the optimal payload mass, the earlier results of the batch with "crash" trajectories are revisited. Figure 8 shows the factor contribution to not only the payload mass and integrated heat load, but also to the final altitude. It is stressed that since there are only three crash trajectories with an altitude different

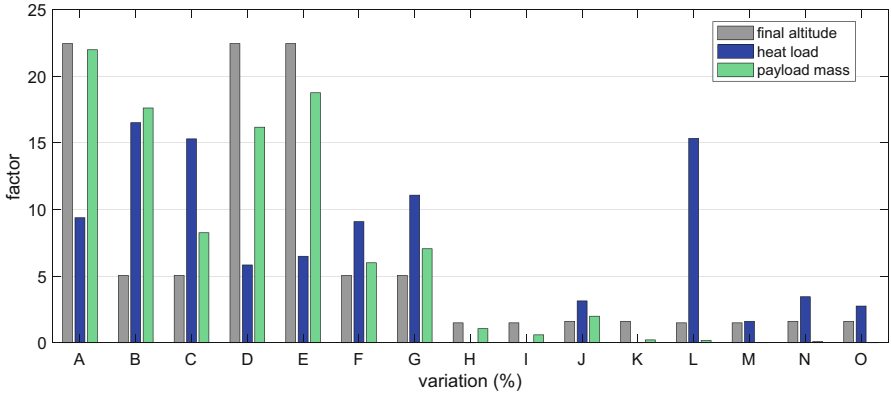


Figure 8 Factor contributions in percents of the total variation to the payload mass and total heat load, "crash" trajectories

from $h_f = 120\,\text{km}$, the variance analysis can only provide a rough indication. But it can be seen that all three responses are driven by the factors that determine the first half of the trajectory. Closer inspection may lead to the conclusion that the vehicle wants to climb too fast, and is therefore having too low dynamic pressure and thus lift that can counteract the large weight early in the flight. As a result the vehicle would "sink" back to lower altitudes before it can correct itself, and is then accelerating at an altitude where the atmospheric density induces a too large drag for the thrust to compensate. This first flight phase also determines the thermal loading later during the flight, when the velocity is too large for the given altitude and thus results in a larger heat flux. In conclusion, this kind of results can be used for failure analysis, in a similar way that the earlier, feasible results are meant to be used to study how to increase the payload mass, and/or reduce the integrated heat load. The fidelity of the ANOVA increases when there is more spread in the results, i.e., when there are no clustered groups of response values.

Another way of analysing the results is a response-surface approach. When the first-order response surface[7] has been applied, which for the payload mass is computed, we find, using normalised factors:

$$\begin{aligned} m_{payload} = \; & 3688.1 - 566.3 n_{z,c} + 743.1\gamma_1 + 36.6\gamma_2 + 76.7\gamma_3 + 223.0\gamma_{c,5} + \\ & 299.5 M_5 + 184.1\gamma_{c,7} + 210.1 M_7 + 58.8\dot{\gamma}_{c,8} + 694.6\gamma_{c,9} + \\ & 174.2 M_9 + 653.1\gamma_{c,11} + 324.3 V_{11} - 155.9\alpha_{i,12} - 0.1\bar{q}_{12} \end{aligned} \tag{27}$$

[7] A response surface with only linear terms assumes no interactions or higher-order effects in the response. This may not be true for all ranges of factor variation, but to do so allows for a comparison with the results of ANOVA. Also, the linear factor variation (minimum and maximum values only) allows for a fast analysis during conceptual trajectory design.

Unfortunately, it is not possible to compute the confidence interval for the coefficients in the response surface, because of the zero degrees of freedom for the error. Therefore, this surface should only be seen as a means to get an indication of the extremes in the design region, and not as a final model. Moreover, we included all factors in the model, whereas in principle those factors with a small contribution could be omitted. However, maybe also some interaction terms should be included in the final first-order response surface. Since it is not intended to actually optimise the trajectory, but merely improve the performance of the WCC to an acceptable level, a refinement of the response surface is not pursued.

Comparing the coefficients in the response surface with the factor contribution, a similar relative importance is found, as shown in Figure 6. Computing the maximum payload mass with Equation (27) is simple:

$$\mathbf{x_{opt}} = (-1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1)^{\mathrm{T}}$$

with $m_{payload,max} = 8089\,\mathrm{kg}$. However, it turns out that a verification simulation with this factor combination results in a payload mass of 7007 kg, which is about 12.5% less. This is not the best of results, yet it does not mean the response surface cannot be used. Optimising the response surface has still provided those factor settings that have increased the payload mass with almost 500 kg, compared to the best value in Table 5. On the other hand, it shows that the response surface has to be used with caution and it has for sure confirmed the importance of the verification run. This is particularly true when a hypersurface needs to be estimated with limited to no excess data points.

In that respect, doing the same calculation with the data obtained from the L_{32} analysis provides a similar response surface, but with an optimal payload mass of 7748 kg (about 4% lower).[8] This is a better result, although the difference is still around 10%. This may be caused by "curvatures" in the response distribution that are not properly captured by the response surface. In case the factor variation is relatively large, this may cause some issues with the predictive power of the surface in between the data points. To test this hypothesis, the factor ranges are reduced to 50% of the original values. Running both L_{16} and L_{32} analysis, computing the response surfaces and predicting the optimum yields 6511 kg (L_{16}) and 6418 kg (L_{32}), for otherwise the same $\mathbf{x_{opt}}$. The verification run gives $m_{payload} = 6270\,\mathrm{kg}$, which is clearly much closer.

When constructing the response surface for the integrated heat load (data listed in Table 5 for the L_{16}-batch) and evaluating this for the same $\mathbf{x_{opt}}$ that maximises $m_{payload}$, the predicted heat load is 8227 MJ/kg, whereas the verification run results

[8] A batch with factor variations over three levels, allowing to include quadratic effects, has been executed using the L_{81} orthogonal array. This array requires 81 simulations for a maximum of 40 independent factors. With the same column assignment as for the L_{32}-batch, the response surface gives a maximum payload mass of 7895 kg when both linear and quadratic terms are included, and 7899 kg with only linear terms. This analysis confirms the consistency of the approach and shows indeed that quadratic terms have a marginal effect for fitting the surface *through* the data points.

in 8548 MJ/kg. The L_{32}-batch provides an even better prediction, i.e., 8492 MJ/kg. The objective space for the integrated heat load is apparently much flatter than the one for the payload mass.

It is noted that the application of a first-order response surface is straightforward, and may not seem to contribute much to the analysis. However, the same approach can easily be extended to higher-order response surfaces, for which it is less trivial which factor combination will give the optimum value. This may be particularly useful when evaluating the simulation model is CPU-expensive, and the response surface can assist in finding the correct search direction at minimal computational cost.

4.3 Sub-Optimal Trajectory

In this sub-section the final reference trajectory will be presented, but only after two adjustments are made in the final pull-up and circularisation manoeuvre. During the final pull-up the propulsion system is producing 20% extra thrust. This is reduced by 10% to come closer to the original propulsion model. Furthermore, the circularisation manoeuvre is still taking about 6700 kg of fuel, which is rather large for an actual mission. Therefore, it should be decreased, which can be achieved by increasing the final pull-up velocity, since there is a one-to-one relationship between this velocity and the required ΔV. The ΔV for circularisation mentioned by Powell et al. [15] is 162 m/s. We aim at $\Delta V \approx 260$ m/s. In that case, the pull-up velocity should be 7170 m/s. Furthermore, the initial pull-up angle of attack is set to 3.5° to reduce the risk of having trim problems, although this would also reduce the final altitude for which additional fuel is required to raise this to 120 km. For this new trajectory, the consumed fuel mass is 68,592 kg, of which 4265 kg is used for the pull-up, leaving a payload mass of 8517 kg. It might be disputed whether such a high pull-up velocity is feasible. Since the model seems to be capable of achieving that velocity, this is not studied any further. The final reference trajectory will be discussed in more detail below.

To begin the discussion on the reference trajectory, one can observe that the trajectory constraints are not violated. Note that the n_a-constraint is not plotted in Figure 9; this constraint is only briefly active during the initial flight phase, which would not show in this graph. But, this constraint is an important one as will become clear later. The $\bar{q}$-constraint seems only to be followed for a short while. It is obvious that during the major part of the trajectory, the vehicle climbs only up to an altitude of about 50 km, whereas V increases from 170 m/s to 7170 m/s. During the powered pull-up an additional 13 km is gained whereas V remains almost constant. A major part of the remaining altitude is covered during the coasting phase.

Compared to the initial reference trajectory, the final trajectory shows a much smoother transition from $\bar{q}_{max}$-tracking to $\dot{Q}_{max}$-tracking. As a result, the throttle control is much smoother as well. Moreover, it is confirmed that after the pull-up at

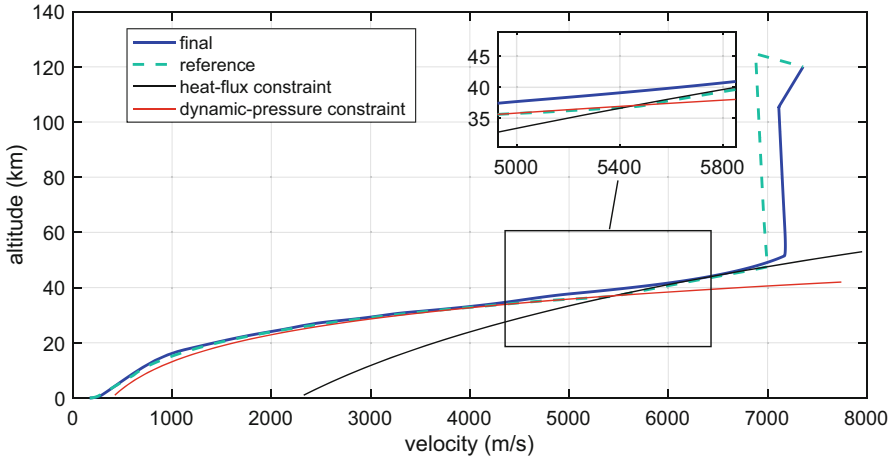


Figure 9 Altitude-velocity profile for the final and initial reference trajectory

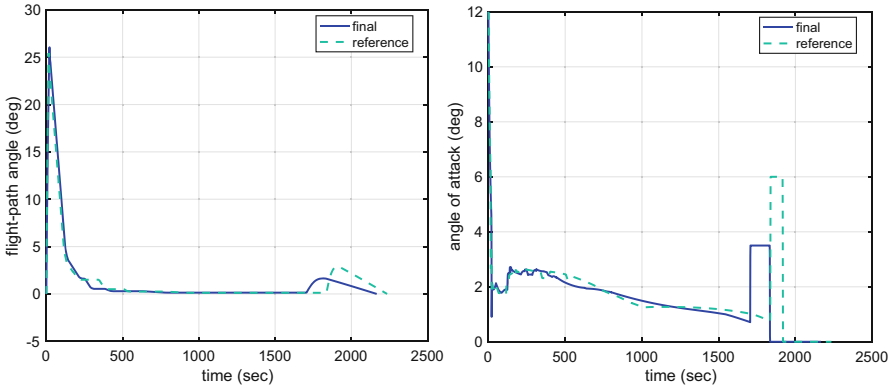


Figure 10 Flight-path angle and angle of attack versus time for the final and initial reference trajectory

the lower α_c, the final altitude of 120 km is not reached. An orbit raising is required, compared to an orbit lowering for the initial trajectory. The pull-up is initiated 100 s, earlier, but lasts longer due to the lower lift.

The trajectory was specified as a γ-profile. This profile is depicted in the left plot of Figure 10, where for both the initial and final reference trajectory γ as a function of time is plotted. Apparent is the large γ during the initial climb, which peaks in the subsonic flight regime. γ rapidly decreases to very small values in order to lock on to the trajectory constraints as good as possible. During the subsequent semi-cruise flight, the vehicle can accelerate while gaining altitude only slowly, such that after about 1700 s the pull-up velocity is reached. The pull-up shows as an increase in γ, decreasing again during the (unpowered) coasting phase. At $h = 112$ km, γ

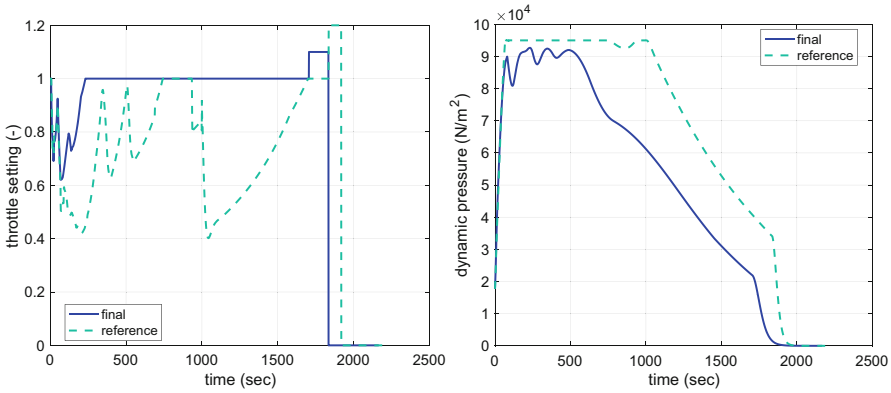


Figure 11 Throttle setting and dynamic pressure versus time for the final and initial reference trajectory

is zero and circularisation takes place. In Figure 10 it can also be seen that major differences between the initial and final γ-profile can be found at the beginning and end of the trajectory.

Figure 10, right, plots α_c versus t. The angle of attack is high during take-off ($\alpha = 12°$) to provide sufficient lift. It reaches a local minimum in the transonic region, which is also mentioned by [24]. Briefly, α_c increases, but the general trend is a decreasing angle of attack as a result of an increasing centrifugal relief. The velocity in the final reference trajectory is higher compared with the initial one at corresponding flight times, such that α of the final trajectory is generally lower. Discontinuities can be found whenever a sudden change in γ is commanded. At $t \approx 1700$ s, α is commanded to be 3.5° to provide sufficient lift for the final pull-up. Note the difference between the initial and final reference trajectory.

In Figure 11 (left), δ_T is plotted versus flight time. The minimum δ_T for the final trajectory is about 0.6, whereas for the initial trajectory this was 0.4. The difference is explained by the two γ-profiles: the final trajectory is a steeper one such that more of the available propulsion power is used to gain height so the vehicle can accelerate more while not violating the constraints. After 200 s of flight, δ_T reaches it maximum, which means that apparently the vehicle can fly with full power without violating the $\bar{q}$-constraint. For the initial trajectory this is not true: δ_T is smaller than 1 up to $t \approx 750$ s, with sharp peaks when a smaller γ is commanded. This means a decrease in ascent rate so that also the acceleration rate must be decreased.

This reasoning is confirmed by Figure 11 (right), where $\bar{q}$ is plotted versus t. It can clearly be seen that the initial trajectory is locked on the $\bar{q}$-constraint of 95 kPa for the major part of the first 1000 s of flight, whereas $\bar{q}$ of the final trajectory comes close to the constraint value, but does not quite reach it. It was stated before that flying along $\bar{q}_{max}$ results in a sub-optimal trajectory, which means that for the final trajectory this is not the case. And still, it is the more fuel-efficient one. Since the many optimisation studies published in literature all state

that a flight along the maximum allowable $\bar{q}$ is optimal, we can only conclude that despite that fact the current trajectory is only the best available for the given trajectory segmentation, parameter variations and applied models. It does not imply that this trajectory cannot be further improved, but for that we may need to refine the segmentation and extend the number of examined factors. One conclusion is obvious, and quite straightforward: the Taguchi method cannot replace a numerical trajectory optimisation method. On the other hand, by doing a Taguchi analysis one can increase the insight in the influence of the trajectory parameters on a selected response.

Returning to Figure 11, in the first 200 s of flight δ_T is not maximal, while still the dynamic pressure is not at its maximum. In that region, the axial acceleration constraint is active, preventing the vehicle from accelerating faster to cope with the increase in altitude. Thus, the vehicle's initial path is too steep for the air-breathing propulsion system. This is easily verified if one compares the corresponding γ profiles for the initial and final reference trajectory. It is interesting to wonder what would happen if the throttle control is released from the acceleration constraint. In that case, $\bar{q}_{max}$ is reached but due to the large γ and the relatively high velocity, the vehicle reaches higher altitudes too soon and it does not have enough climbing power to maintain the acceleration. Therefore, a strong drop in $\bar{q}$ is the result, the vehicle starts descending again and finally crashes. This clearly shows the difference between a conventional rocket and an air-breathing space plane. Whereas the former can accelerate fast enough to reach higher altitudes to minimise the drag, the latter must remain at lower altitudes and accelerate at high $\bar{q}$ to maximise the performance of the propulsion system. To conclude these figures, at $t \approx 1700$ s the pull-up shows as $\delta_T = 1.1$ for the final reference trajectory and a sharp decrease of $\bar{q}$. The rapid increment in h and thus decline in ρ result in this decrease.

In Figure 12 (left), $\dot{Q}$ is plotted as a function of t. It is clear that this constraint is an important one since a considerable part of the initial trajectory is flown along this

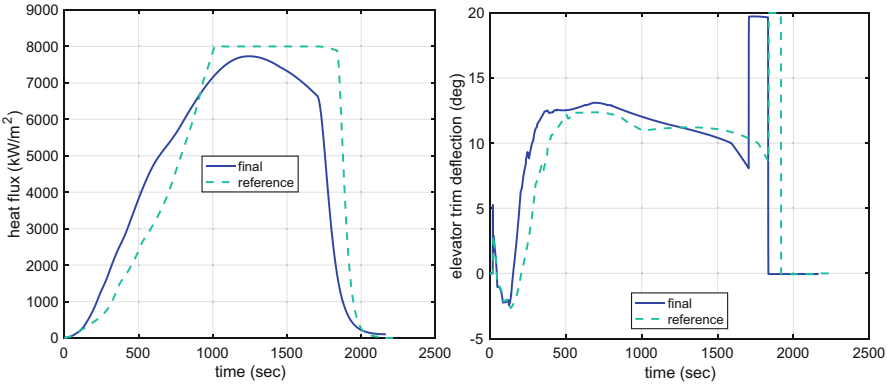


Figure 12 Heat flux and elevator-trim deflection versus time for the final and initial reference trajectory

constraint. For the final one, the vehicle follows the constraint more smoothly, and therefore does not reach $\dot{Q}_{max}$ before it enters the pull-up. However, the constraint drives this portion of the flight. It is therefore not sufficient to take only the $\bar{q}$-constraint into account as was suggested by Powell et al. [15], but to include $\dot{Q}_{\max}$ as well. The integrated heat load is for both trajectories almost the same, i.e., the higher velocities of the final trajectory are compensated for by the longer flight time of the initial trajectory.

Figure 12 (right), finally, shows the deflection angles of the elevons to trim the vehicle (at subsonic speeds trim is solely guaranteed by the canards; during take-off ($\alpha = 12°$), $\delta_{c,trim} = 5°$). The history shows a minimum near $M = 2 (t \approx 100\,\text{s})$ and exhibits a change in vehicle trim as a result of combined changes in aerodynamic loading and a shift in the location of the c.o.m. due to fuel consumption, an effect confirmed by Hattis et al. [24]. There is, of course, a strong relation between α and δ_e to compensate for the induced pitch moment. The maximum δ_e is found shortly after the local α_{max}. Considering the fact that $\delta_{e,max} = 20°$, trimming the vehicle with elevons alone puts a heavy load on them, which does not leave so much margin for manoeuvring and compensating deviations in α_c. For this reason, it is wise to assess other means of vehicle trim (see Sect. 4.4). Due to the decreasing α, also $\delta_{e,trim}$ decreases. The final pull-up results in a deflection of the elevons close to their limit value. Since this pull-up manoeuvre is not optimal in any sense, we cannot draw strong conclusions, but state that this manoeuvre should be carefully examined, since it might create control problems. During the coasting at $\alpha = 0°$, the pitch moment is zero, and thus also $\delta_{e,trim}$.

4.4 Thrust-Vector Control Used for Trim

The last topic to be discussed is the already mentioned large trim deflection of the elevons. Two reasons are apparent to decrease the required deflection angles, i.e., it leaves a larger margin for manoeuvring and the induced trim drag can be reduced to save fuel mass. The first reason has already been shown to be a valid one; the second one is relatively easy to illustrate, although some remarks must be made.

In Figure 13, the drag and lift have been plotted for the final reference trajectory, including the induced trim drag and lift. Both the trim drag and lift form a substantial part of the vehicle drag and lift, i.e., up to 50% for the drag and 60% for the lift (excluding the pull-up manoeuvre). The current reference trajectory has been defined such that a large part of the external forces is used to increase the altitude of the space plane. This part consists mainly of the lift force, so it is easy to understand that for this steep ascent trajectory the lift will not be sufficient if the contribution of the elevons is removed from the external-force vector. Verifying this notion indeed results in a crash of the vehicle due to insufficient lift. To avoid the problem of finding a new reference trajectory for an untrimmed vehicle, the influence of trim drag is assessed by flying a trimmed trajectory and by putting the trim drag to zero. It follows that 4044 kg less fuel is required to reach the target orbit, about 45% of the

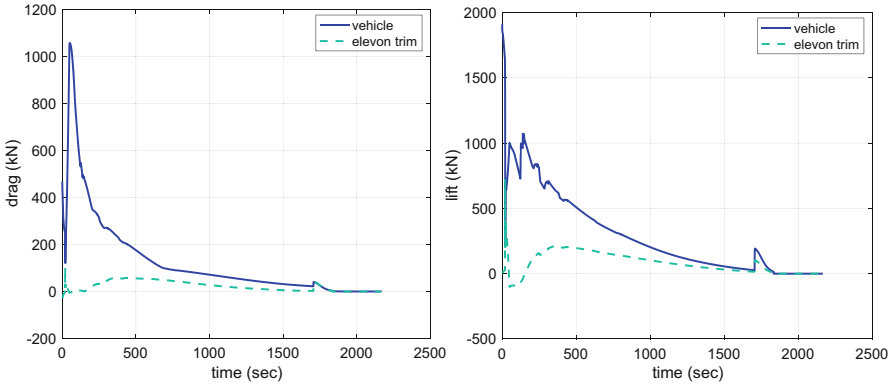


Figure 13 Vehicle drag and lift and elevon-trim drag and lift as a function of flight time

payload mass of 8517 kg! Conclusions that can be drawn from these observations is that in performance analysis, the influence of trim drag and lift should both be included. Moreover, the trim drag requires quite some extra fuel, which makes it worthwhile looking for alternatives.

One alternative that can be studied is the use of TVC. Of course, this will make the propulsion system more complicated (and possibly heavier), but in principle fuel-mass savings are possible. The thrust is the basis for the acceleration of the vehicle and contributes to the lift force. When the thrust vector is deflected such that the vertical component gives rise to a compensating pitch moment, two side effects are introduced. First, there is larger contribution to the lift force, which may result in a smaller α and therefore a decrease of the global drag force. Second, the horizontal thrust component for acceleration becomes smaller, such that it will take a larger δ_T (or, equivalently, a longer flight time) to reach the pull-up velocity and consequently a larger fuel consumption. Obviously, the use of TVC should be included in the trajectory optimisation process.

Since the WCC flies with maximum throttle for a large part of the current trajectory, the use of TVC may lead to problems in the sense that there is not enough accelerating power left. To apply thrust-vector control we need to define the so-called centre of thrust (c.o.t.), i.e., the location where the resulting thrust vector is acting on the vehicle. Since there is no information available on the defined propulsion model, the following is assumed. The actual engine section is considered to be the cylindrical part of the WCC, recall Figure 1. The cone frustum nozzle is also part of the propulsion system, since the exhaust-gas expansion is an indispensable part of the thrust generation. Therefore, the c.o.t. must be located somewhere on this nozzle. For computational purposes it is assumed that this is at one third of from the end of the nozzle, i.e., 19.1 m from the moment reference centre, and, since the engine section is axisymmetric, effectively at the X-axis. Furthermore, a limit of $\pm 25°$ will be put on the thrust-vector angle, although it is not implied that this limit is a feasible one.

Table 6 The influence of TVC on the fuel mass

M_{TVC} (%)	m_{fuel} (kg)	Δm_{fuel} (kg)
0	69,350	0
5	68,142	1208
10	67,692	1658
15	67,516	1834
20	67,378	1972
25	67,292	2058
30	67,240	2110
35	67,242	2108
40	67,285	2065
45	67,349	2001
50	67,421	1929

The technical implementation of TVC will not be addressed, nor the influence of TVC on propulsion-system performance, which—in a worst-case scenario—could actually render TVC infeasible. The practical implementation of the trim rule will be that below $M = 0.9$ the canards are used, and above $M = 0.9$ a predefined percentage of the pitch moment is compensated for by the elevons, and the remaining part by TVC. In case the limit value of the TVC angle is reached, the remaining moment is also compensated for by the elevons. Since full TVC is not possible, we will study values of 50% down to 0% of the pitching moment that is in principle addressed by TVC. In all cases, the WCC reaches the pull-up velocity at about the same altitude and after almost similar flight times, albeit the elevon-only trim case takes longer. The results of the simulations in terms of fuel mass and fuel savings are listed in Table 6.[9]

From the results it is obvious that the use of TVC can save a substantial amount of fuel. By assigning 30% of the trim moment to TVC, the total m_{fuel} to reach the pull-up velocity is reduced by 2110 kg. When the trim parameters are studied, i.e., δ_e and thrust elevation ε_T in Figure 14, we see the difference between the all-elevon-trim, and 5%-, 15%-, 30%- and 45%-thrust-elevation-trim trajectories. It is clear that the maximum δ_e reduces from about 13° down to about 8°. Overall, the decrease of δ_e is between 2° and 8°. This reduction is significant and it is evident that the manoeuvring capability has increased. In the profile of δ_e some sharp peaks are observed. Again, these peaks are the result of the crude reference-trajectory definition. Transition from one $\dot{\gamma}_c$ to another results in discrete jumps in α (see also Figure 10) and thus $\delta_{e,trim}$.

[9]The use of TVC interacted with the flight-path angle steering loop, Equation (5), and some high-frequency, small-amplitude oscillations were induced in the commanded angle of attack. This led to oscillations in the thrust-elevation angle. Without doing a redesign, the gains were set to $K_{\gamma p} = 2.0$, $K_{\gamma i} = 1.8$ and $K_{\gamma d} = 0$ for the complete trajectory, which solved the problem. However, it was observed that changing the gains has a noticeable effect on the fuel mass, so in a future design the gains should be optimised.

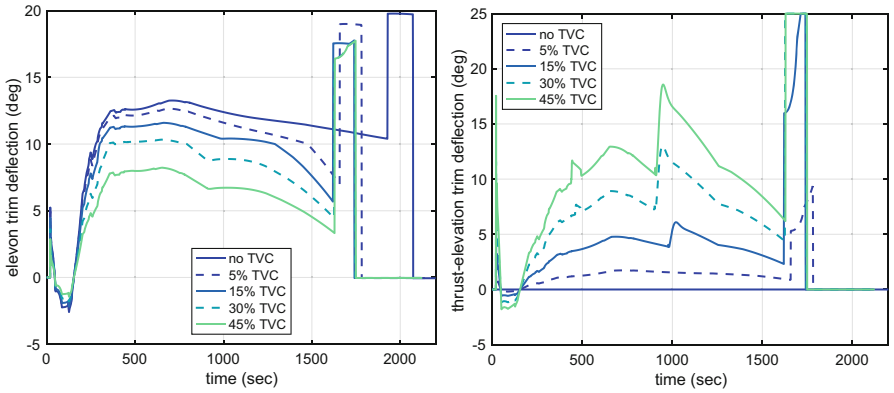


Figure 14 Elevator and thrust-elevation trim deflection versus time for different TVC percentages

Regarding $\varepsilon_{T,trim}$, at activation ($M = 0.9$) $\varepsilon_{T,trim} = 17°$ (at 45% TVC). Up to $t \approx 900$ s it decreases to 10°, with some transitions that are linked with the particular M-α profile flown. Also in this case, the variations are related to transitions from one flight segment to the other. Just after $t \approx 900$ s, due to the combination of propulsive efficiency and hypersonic effects a rather sudden decrease in T occurs. It is obvious that a larger deflection of the thrust vector is required to compensate for the pitch moment. This can clearly be seen in Figure 14, right. After reaching deflections close to its maximum of about 20°, $\varepsilon_{T,trim}$ slowly decreases to 6° due to the influence of M and α on the pitch moment.

From the above results it would be easy to conclude that TVC is a good means to decrease the total fuel mass and to increase the manoeuvrability capability with the elevons. As was indicated before, the technical aspects of TVC have not been addressed at all. Doing so might lead to completely different conclusions. Also the influence of the thrust-vector orientation on the flight mechanics (jet damping) will have to be studied. However, this coarse performance study has indicated that the use of TVC should at least be studied in more detail.

5 Concluding Remarks

The simulation and analysis of the powered winged-cone configuration includes closed-loop guidance based on inverse dynamics for the ascent to orbit. Moreover, a closed-loop throttle control law to track the trajectory constraints (maximum dynamic pressure, axial acceleration and heat flux) completes the guidance and control system. Trim can be guaranteed by using either elevons or thrust-vector control. The 3-degrees-of-freedom reference trajectory is computed by dividing the trajectory into a number of flight segments and by specifying the flight-path angle profile for each of these segments. The trajectory is sub-optimised with respect to payload mass by treating the segment parameters as design variables and by doing

a Taguchi analysis. A payload-mass variation between 732 and 6584 kg was found, which indicates that a proper selection of the trajectory is very important to ensure mission success. The minimum-fuel trajectory, obtained from the L_{32}-analysis, does not correspond to the minimum heat-load trajectory, which takes up almost 1000 kg more fuel. Therefore, the maximum allowable heat load should be included in any trajectory optimisation process. The relative ease with which trajectories could be found that could reach orbit seems to indicate that the performance of the propulsion system is overrated with respect to the vehicle's dry mass.

Analysis of variance of the results indicated that the influence of interactions, as well as quadratic effects, is small for this particular design problem of finding a reference trajectory. Redoing the design with double the number of simulations by using the L_{32} instead of the L_{16} array gave different means (3688 and 3647 kg) and standard deviations (1521 and 1413 kg) for the selected response, the payload mass. However, conducting a Kolmogorov–Smirnov test indicates that the two distributions of the payload masses are consistent. On the other hand, there also appeared to be a minor shift in factor contribution to the total variation, which means that major factor effects can be extracted from the results, but that the absolute importance should be assessed with caution.

The Taguchi method cannot replace a numerical trajectory optimisation method. However, by doing a Taguchi analysis for a given trajectory segmentation one can increase the insight in the influence of the trajectory parameters on a selected response.

Using the elevons for trim results in a relatively large contribution to the drag and lift. Therefore, for an accurate performance analysis the influence of trim drag and lift should both be included. Moreover, the trim drag increases the consumed fuel mass substantially. Using TVC to generate part of the compensating trim moment results in a significant fuel reduction. However, when major part of the flight is flown with full throttle the trajectory has to be redesigned (or re-optimised) to compensate for the loss of accelerating power. The technical feasibility of TVC was not included in this study.

With the assumptions made for the selected flight-dynamics model, simplified guidance-system development, and two-level factor variation the results have shown the potential of the design and analysis method. Next steps in the research could be, of course, extensions and improvements of each of these aspects. The principal approach does not have to be changed, though.

References

1. Dinardi, A., Capozzoli, P., Shotwell, G.: Low-cost launch opportunities provided by the falcon family of launch vehicles. In: The Fourth Asian Space Conference. Taipei (2008)
2. Cribbs, D.: Performance uncertainty analysis for NASP. In: 2nd International Aerospace Planes Conference. Orlando (1990)
3. Lovell, C.K., Schmidt, P.: Mission performance of hypersonic vehicles. In: AIAA Guidance Navigation and Control Conference. Baltimore (1995)

4. Schmidt, D.K., Lovell, T.A.: Mission performance and design sensitivities for hypersonic airbreathing vehicles. J. Spacecr. Rocket. **34**(2), 158–164 (1997)
5. Wagner, A., Thevenot, R.: STS 2000: a reference air-breathing SSTO. In: AIAA 3rd International Aerospace Planes Conference. Orlando (1991)
6. Calise, A.J., Flandro, G.A., Corban, J.E.: Trajectory Optimization and Guidance Law Development for National Aerospace Plane Applications. NASA CR-182994. NASA, Washington DC (1988)
7. Corban, J.E., Calise, A.J., Flandro, G.A.: Rapid near-optimal aerospace plane trajectory generation and guidance. J. Guid. Control. Dyn. **14**(6), 1181–1190 (1991)
8. Van Buren, M.A., Mease, K.D.: Aerospace plane guidance using time-scale decomposition and feedback linearisation. J. Guid. Control. Dyn. **14**(6), 1166–1174 (1992)
9. Lu, P.: Entry guidance: a unified method. J. Guid. Control. Dyn. **37**(3), 713–728, (2014)
10. Lu, P.: Inverse dynamics approach to trajectory optimization for an aerospace plane. J. Guid. Control. Dyn. **16**(4), 726–732 (1995)
11. Hess, R.A., Gao, C., Wang, S.H.: Generalized technique for inverse simulation applied to aircraft maneuvers. J. Guid. Control. Dyn. **14**(5), 920–926 (1991)
12. Morio, V., Cazaurang, F., Vernis, P.: Flatness-based hypersonic reentry guidance of a lifting-body vehicle. Control. Eng. Pract. **17**(5), 588–596 (2009)
13. Shaughnessy, J.D., Zane Pinckney, S., McMinn, J.D., Cruz, C.I., Kelley, M.L.: Hypersonic vehicle simulation model: Winged-Cone configuration. NASA Technical Memorandum 102610, Langley Research Center, Hampton, 1990
14. Hattis, P.D., Malchow, H.D.: Evaluation of some significant issues affecting trajectory and control management for air-breathing hypersonic vehicles. In: AIAA Fourth International Aerospace Planes Conference. Orlando (1992)
15. Powell, R.W., Shaugnessy, J.D., Cruz, C.I., Naftel, J.C.: Ascent performance of an air-breathing horizontal-takeoff launch vehicle. J. Guid. Control. Dyn. **14**(4), 834–839 (1991)
16. Gregory, I.M., Chowdhry, R.S., McMinn, J.D., Shaughnessy, J.D.: Hypersonic vehicle model control law development using H_∞ and μ-synthesis. NASA Langley Research Center, Hampton (1994)
17. Vinh, N.X.: Optimal Trajectories in Atmospheric Flight. Elsevier, Amsterdam (1981)
18. Iman, R.L.: Latin hypercube sampling. In: Encyclopedia of quantitative risk analysis and assessment. Wiley, New York (2008)
19. Park, J.S.: Optimal Latin-hypercube designs for computer experiments. J. Statist. Plann. Inference **39**(1), 95–111 (1994)
20. Mooij, E.: Re-entry test vehicle configuration selection and analysis. In: Fasano, G., Pintr, J.D. (eds.) Space Engineering. Modeling and Optimization with Case Studies, pp. 199–235. Springer, Berlin (2016)
21. Taguchi, G.: System of Experimental Design. Engineering Methods to Optimise Quality and Minimise Costs, 2nd edn, vol. 1. UNIPUB/Kraus International Publications, White Plains (1988)
22. Phadke, M.S.: Quality Engineering Using Robust Design. Prentice-Hall, Englewood Cliffs (1989)
23. Marée, A.G.M., Mooij, E., Zandbergen, B.T.C.: Space-plane analysis: the generation of a reference trajectory. Report LR-749, Delft University of Technology, Faculty of Aerospace Engineering, 1994
24. Hattis, P., Malchow, H., Shaughnessy, J., Chowdry, R.: Integrated trajectory and control analysis for generic hypersonic vehicles. In: 3rd International Aerospace Planes Conference, AIAA-1991-5052 (1991)

Ascent Trajectory Optimization and Neighboring Optimal Guidance of Multistage Launch Vehicles

Guido Palaia, Marco Pallone, Mauro Pontani, and Paolo Teofilatto

Abstract Multistage launch vehicles are employed to place spacecraft and satellites in their operational orbits. If the rocket aerodynamics and propulsion are modeled appropriately, optimization of their ascent trajectory consists in determining the coast duration and the thrust time history that maximize the final mass at injection. This research derives all the necessary conditions for ascent path optimization of a multistage launch vehicle. With reference to an existing rocket, the indirect heuristic method is then applied, for the numerical determination of the overall ascent trajectory. An effective approach is used with the intent of satisfying the path constraint related to the maximum dynamical pressure in the atmospheric phase. Then, the recently introduced, implicit-type variable-time-domain neighboring optimal guidance is applied to the upper stage powered arc, for the purpose of obtaining the corrective control actions in the presence of nonnominal flight conditions. The guidance approach at hand, based on the second-order analytical conditions for optimality, proves to be rather effective (in terms of propellant budget), and guarantees very accurate orbit injection in spite of perturbations.

1 Introduction

Multistage launch vehicles are commonly employed to place spacecraft in their operational orbits. Several characteristics, i.e., mass distribution and time variation, propulsion, and aerodynamics, affect the overall performance of the ascent vehicle of interest. Thus, it is apparent that accurate modeling is a central issue and an essential prerequisite for trajectory optimization. This research uses the Scout, a launch vehicle of reduced size used in the past, as the reference multistage rocket, because of availability of its basic propulsion and aerodynamics characteristics.

G. Palaia · M. Pallone · M. Pontani (✉) · P. Teofilatto
Sapienza University of Rome, Rome, Italy
e-mail: palaia.1164882@studenti.uniroma1.it; pallone.1420138@studenti.uniroma1.it; mauro.pontani@uniroma1.it; paolo.teofilatto@uniroma1.it

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_13

In general, the numerical solution of aerospace trajectory optimization problems is not trivial and has been pursued with different approaches in the past. However, only a relatively small number of publications are concerned with trajectory optimization of multistage launch vehicles [2, 4, 6, 7, 10, 12, 13, 15–18, 27, 29, 30, 35]. Calise et al. [4] and Gath and Calise [7] proposed and applied a hybrid analytic/numerical approach, based on homotopy and starting with the generation of the optimal solution in a vacuum. They adopted the approximate linear gravity model, and the same did Lu and Pan [13] and Lu et al. [12], who applied a multiple-shooting method to optimizing exoatmospheric trajectories composed of two powered phases separated by a coast arc. Weigel and Well [35] used a similar indirect, multiple-shooting approach to analyze and optimize the ascent trajectories of two launch vehicles with splash-down constraints. Miele [17] developed and applied the indirect multiple-subarc gradient restoration algorithm to optimizing the two-dimensional ascending trajectory of a three-stage rocket in the presence of dynamical and control constraints. The previously cited works [4, 7, 12, 13, 17, 35] make use of indirect algorithms and require a considerable deal of effort for deriving the analytical conditions for optimality and for the subsequent programming and debugging. Furthermore, these methods can suffer from a slow rate of convergence and an excessive dependence on the starting guess. This difficulty has been occasionally circumvented through homotopy [4, 7, 15], but this adds further complexity to the solution process. Other papers deal with direct numerical techniques applied to multistage rocket trajectory optimization. Roh and Kim [30] used a collocation method for optimizing the performance of a four-stage rocket, whose two-dimensional trajectory was assumed to be composed of three thrust phases and a coast arc of specified duration. Collocation was also employed by Jamilnia and Naghash [10], with the additional task of determining the optimal staging, and by Martinon et al. [15], for the purpose of validating the numerical results attained through indirect shooting. The latter paper refers to the Ariane V launch vehicle and is specifically devoted to investigating singular arcs. Recently, heuristic techniques have become established as suitable approaches for solving optimal control problems. Unlike deterministic methods, they do not suffer from locality of the results and do not require any starting guess. The main disadvantages of heuristic algorithms are the lack of any convergence proof and the capability of yielding only a near optimal solution, if a particular representation for control variables is adopted.

This research is concerned with an approach based on the joint use of swarming theory and the necessary conditions for optimality. This methodology circumvents the previously mentioned disadvantages of using heuristic approaches, while retaining their advantageous feature of not requiring any starting guess to generate an optimal solution. The particle swarm optimization technique, which is used in this study, is a heuristic population-based optimization method inspired by the natural motion of bird flocks. The population migrates toward different regions of the search space, taking advantage of the mechanism of information sharing that affects the overall swarm dynamics.

This work considers the optimization of the two-dimensional ascent path of a four-stage launch vehicle, whose trajectory is composed of the following thrust phases and coast arcs:

1. first stage propulsion,
2. second stage propulsion,
3. third stage propulsion,
4. coast arc (after the third stage separation), and
5. fourth stage propulsion.

An aerodynamic constraint related to the dynamical pressure must be enforced in order to preserve the vehicle structural integrity.

Precision of orbit injection, which occurs at burnout of the upper stage, represents a crucial issue, and potentially affects the subsequent phases of spaceflight, because corrective maneuvers may be needed if orbit insertion takes place with unsatisfactory accuracy. It is thus apparent that a feedback control law, or, equivalently, a closed-loop guidance algorithm, is needed for the upper stage, with the final aim of driving it toward the desired injection conditions, in the presence of deviations from nominal flight conditions. Traditionally, two different approaches to guidance exist. Adaptive (or explicit) algorithms stem from the idea of re-defining the flight trajectory at the beginning of each guidance interval, at which an updated trajectory (from the current to the target final condition) is computed [3, 32]. Perturbative (or implicit) algorithms consider the perturbations from a specified nominal trajectory, and define the feedback control corrections aimed at maintaining the vehicle in the proximity of the nominal path [9, 11, 33]. Neighboring optimal guidance (NOG) is a perturbative guidance concept that relies on the analytical second-order optimality conditions, in order to find the corrective control actions in the neighborhood of the reference path. This is an optimal trajectory that satisfies the first- and second-order optimality conditions. In general, the neighboring optimal path originates from a perturbed state and is associated with the minimization of the second differential of the objective function. Several time-varying gain matrices, referring to the nominal trajectory, are defined, computed offline, and stored in the onboard computer. Only a limited number of works have been devoted to studying neighboring optimal guidance [1, 5, 8, 31, 36]. A common difficulty encountered in implementing the NOG consists in the fact that the gain matrices become singular while approaching the final time. As a result, the real-time correction of the time of flight can lead to numerical difficulties so relevant to cause the failure of the guidance algorithm.

This work outlines the recently introduced [22–24], general-purpose variable-time-domain neighboring optimal guidance algorithm (VTD-NOG). Some fundamental, original features of VTD-NOG allow overcoming the main difficulties related to the use of former NOG schemes, in particular the occurrence of singularities and the lack of an efficient law for the iterative real-time update of the time of flight. This is achieved by adopting a normalized time domain, which leads to defining a novel updating law for the time of flight, a new termination criterion, and a new analytical formulation for the sweep method.

VTD-NOG is applied to the ascent path of the upper stage of the Scout launch vehicle, in the presence of nonnominal flight conditions, i.e., errors on the initial velocity and position at ignition. Monte Carlo simulations are performed with the intent of testing effectiveness (in terms of propellant required for corrective actions) and accuracy at injection in spite of perturbations.

In the end, with regard to the launch vehicle of interest, the research that follows has the main objectives of (i) providing accurate modeling of propulsion and aerodynamics, (ii) formulating the ascent trajectory optimization problem by deriving all the necessary conditions for optimality, (iii) determining the optimal ascent path that maximizes the payload mass while enforcing the aerodynamic constraint, and (iv) applying VTD-NOG for the upper stage powered arc, in order to ascertain effectiveness and accuracy at injection, in the presence of nonnominal flight conditions.

2 Launch Vehicle Modeling

The four-stage rocket that is being considered is the Scout launch vehicle, which is a rocket designed to place small satellites into low Earth orbit. It has specified structural, propulsive, and aerodynamic characteristics, and it is represented in Figure 1.

Figure 1 Scout rocket geometry

2.1 Mass Distribution and Propulsion

The mass distribution of the launch vehicle can be described in terms of masses of subrockets: subrocket 1 is the entire rocket with all the four stages, subrocket 2 is the launch vehicle after separation of the first stage, subrocket 3 is the launch vehicle after separation of the first two stages, and subrocket 4 is represented by the last stage only. Let $m_0^{(i)}$ denote the initial mass of the subrocket i; this mass is composed of a structural mass $m_S^{(i)}$, a propellant mass $m_P^{(i)}$, and a payload mass $m_U^{(i)}$:

$$m_0^{(i)} = m_S^{(i)} + m_P^{(i)} + m_U^{(i)} \tag{1}$$

For the first three subrockets $m_U^{(i)}$ (i=1,2,3) coincides with the initial mass of the subsequent subrocket (i.e., $m_U^{(i)} = m_0^{(i+1)}$). The mass distribution for the Scout rocket is presented in Table 1. Figure 2 portrays the mass variation [34] due to the propellant usage for the four stages. The instantaneous propellant mass is obtained through linear interpolation.

The propulsive characteristic of the launch vehicle can be described in terms of thrust magnitude $T^{(i)}$, where i is the stage number. The thrust is obtained through a linear interpolation of the experimental thrust data [34], which are given at discrete times. Figure 3 portrays the thrust curves for each motor, whose burnout time is t_{Bi}.

Table 1 Mass distribution for the first three subrockets [34]

i	$m_S^{(i)}$ (kg)	$m_P^{(i)}$ (kg)
1	1736	12,810
2	915	3749
3	346	1173
4	33	281

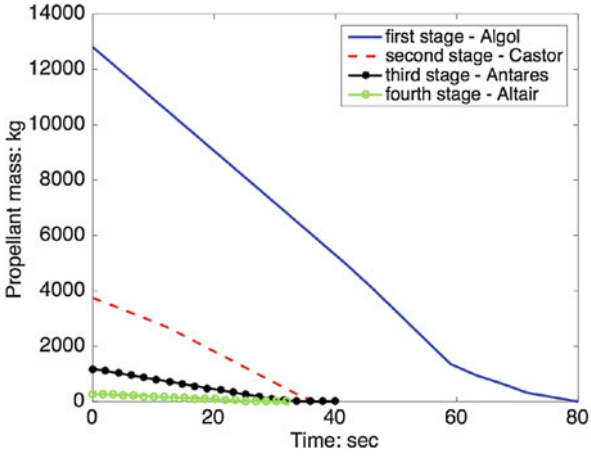


Figure 2 Propellant mass for the four stages

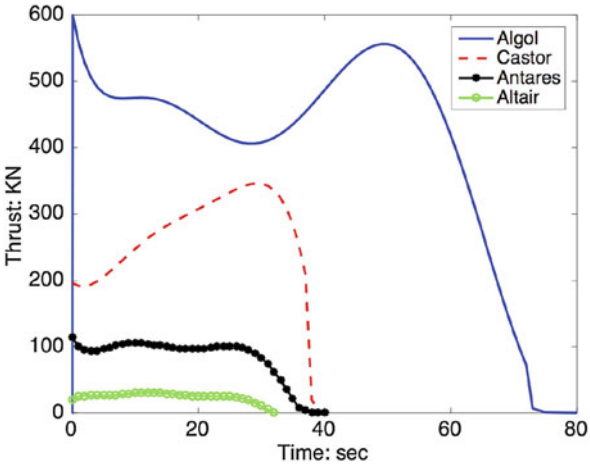


Figure 3 Thrust time histories for the four stages

Table 2 Aerodynamic surfaces for the first three subrockets

i	1	2	3
$S^{(i)}$ (m^2)	1.026	0.487	0.458

2.2 Aerodynamics

As two-dimensional motion is being considered, the aerodynamic force is assumed to be composed of two terms: lift force L and drag force D. Given the aerodynamic surface S, the atmospheric density ρ, the speed relative to the Earth atmosphere v, and the lift and drag coefficients C_L and C_D, the two components are

$$L = \frac{1}{2} C_L^{(i)}(\alpha, M) S^{(i)} \rho v^2 \quad \text{and} \quad D = \frac{1}{2} C_D^{(i)}(\alpha, M) S^{(i)} \rho v^2 \tag{2}$$

where for each subrocket i the coefficients C_L and C_D depend on the Mach number M and the aerodynamic angle of attack α. The aerodynamic surfaces used in the computation are the cross surfaces and are listed in Table 2.

Aerodynamic modeling is composed of two steps:

1. derivation of drag and lift coefficients at several Mach numbers and angles of attack, and
2. bivariate cubic spline interpolation of the aerodynamic parameters as functions of Mach number and angle of attack.

The aerodynamic coefficients (at discrete points) of the Scout rocket were obtained using the methodology described by Mangiacasale [14], which was applied to the first three subrockets. The domain for (α, M) is defined as

$$-10 \text{ deg} \leq \alpha \leq 10 \text{ deg}$$

$$0 \leq M \leq 10$$

The piecewise bicubic interpolation (performed on a regular grid of points (α_h, M_k)) produces a smooth surface that passes through all the data points and has continuous partial derivatives. The generic bicubic spline function C_η is defined over the domain $[\alpha_h, \alpha_{h+1}] \times [M_k, M_{k+1}]$

$$C_\eta(\alpha, M) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} (\alpha - \alpha_h)^i (M - M_k)^j \tag{3}$$

with $\eta = L, D$. A Matlab routine is employed to determine the values of the 16 coefficients a_{ij} that uniquely identify $C_\eta(\alpha, M)$. After patching all the bicubic portions, the overall surface is continuous (up to second-order derivatives) over the entire domain of (α, M).

An example of the aerodynamic interpolation in the supersonic range is portrayed in Figure 4, where the C_D and C_L coefficients along with their derivatives are illustrated. Despite only the supersonic range is shown, the interpolation of the aerodynamic coefficients and their derivatives yields continuous functions over all the domain. Over 120 km of altitude, where the fourth stage usually flies, the atmosphere is rarefied and the aerodynamic forces can be neglected.

3 Ascent Trajectory Optimization

The objective of ascent trajectory optimization is in finding the path that leads the rocket to reaching a prescribed circular orbit while minimizing propellant expenditure. The four-stage launch vehicle is modeled as a point mass, in the context of a two-degree-of-freedom problem.

3.1 Equations of Motion

The rocket motion is described more easily in a rotating (i.e., non-inertial) reference frame. The Earth centered Earth fixed (ECEF) frame represents a reference system that rotates with the Earth and has the origin in its center. The ECEF system rotates with a speed $\omega_E = 7.292115 \times 10^{-5}$ sec^{-1} with respect to an inertial Earth-centered frame (ECI), denoted with $(\hat{c}_1, \hat{c}_2, \hat{c}_3)$. Both frames share the same origin O. The unit vector $\hat{c}_1$ is aligned with the vernal axis and the vector $\hat{c}_3 = \hat{k}$ points toward the planet rotation axis and is positive northward. Therefore $\boldsymbol{\omega}_E$ represents the (vector) rotation rate of the ECEF frame with respect to the ECI frame. The unit vector $\hat{i}$ intersects the Greenwich meridian at all times. The ECEF frame is associated with $(\hat{i}, \hat{j}, \hat{k})$, which form a right-handed, time-dependent sequence of unit vectors. As the reference Greenwich meridian rotates with rotation rate ω_E, its angular position (with respect to the ECI frame) is identified by the Greenwich sidereal time $\theta_g(t)$.

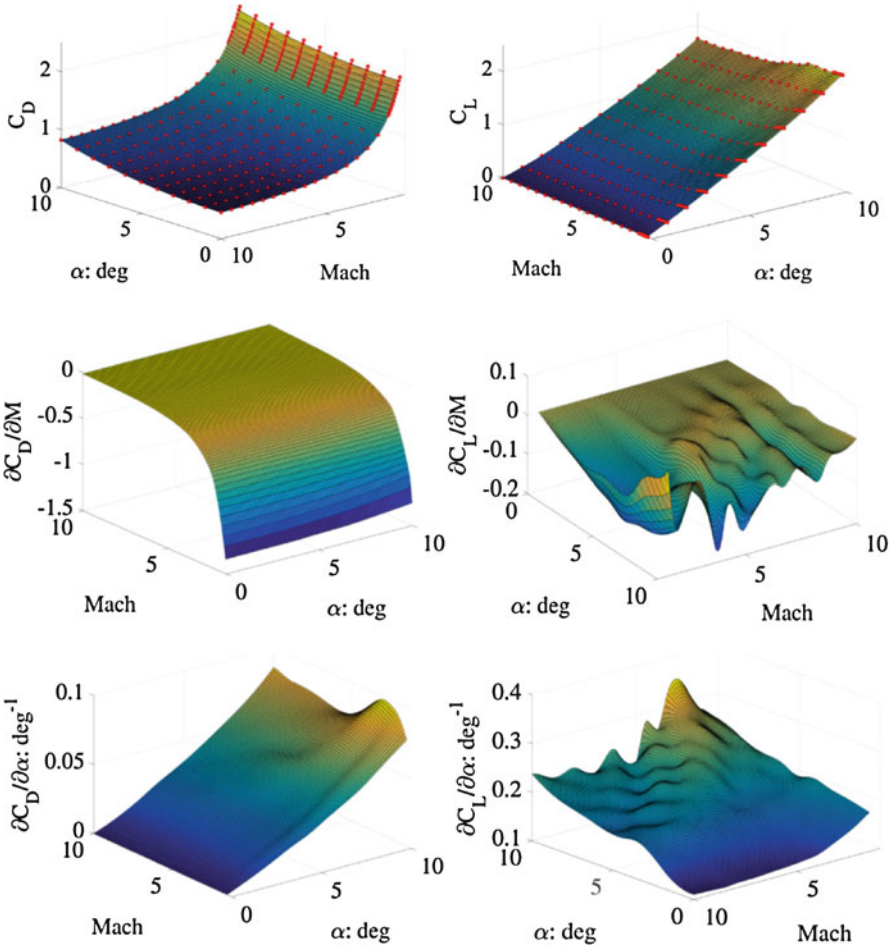


Figure 4 Bivariate cubic spline interpolation of the C_D and C_L coefficients along with their derivatives, for the first subrocket in supersonic flight

Moreover, as the entire trajectory lies on the equatorial plane, the flight path angle γ suffices to identify the velocity direction. The instantaneous position is defined through r (=$|\mathbf{r}|$) and the geographical longitude ξ. From the inspection of Figure 5, it is apparent that

$$\mathbf{v} = v[\sin\gamma \quad \cos\gamma][\hat{r} \quad \hat{E}]^T \tag{4}$$

The overall aerodynamic force $\mathbf{A}$ is conveniently written in the $(\hat{n}, \hat{v}, \hat{h})$ frame (with $\hat{v}$ aligned with $\mathbf{v}$) as the sum of the lift and drag forces

$$\mathbf{A} = L\hat{n} - D\hat{v} \tag{5}$$

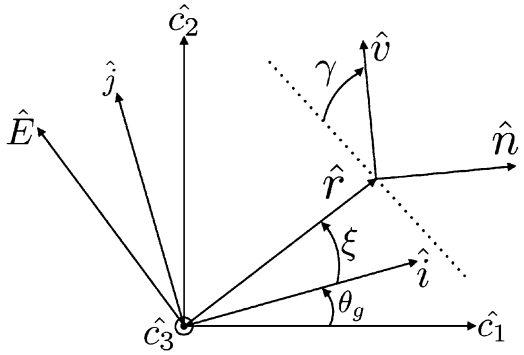


Figure 5 $(\hat{r}, \hat{E})$ frame, $(\hat{n}, \hat{v})$ frame, and related angles

Omitting the superscript *(i)*, for each subrocket the equations of motion are

$$\dot{r} = v \sin\gamma \tag{6}$$

$$\dot{\xi} = \frac{v \cos\gamma}{r} \tag{7}$$

$$\dot{\gamma} = \frac{T}{mv} \sin\alpha_T + \left(\frac{v}{r} - \frac{\mu_E}{r^2 v}\right) \cos\gamma + \frac{L}{mv} + 2\omega_E + \frac{\omega_E^2 r}{v} \cos\gamma \tag{8}$$

$$\dot{v} = \frac{T}{m} \cos\alpha_T - \frac{\mu_E}{r^2} \sin\gamma - \frac{D}{m} + \omega_E^2 r \sin\gamma \tag{9}$$

where μ_E (=398600.4 km^3/sec^2) is the Earth gravitational parameter and m is the instantaneous mass of the ascent vehicle. Specifically, for each subrocket i

$$m^{(i)}(t) = m_S^{(i)} + m_U^{(i)} + m_P^{(i)}(t) \tag{10}$$

where the time-varying $m_P^{(i)}(t)$ is portrayed in Figure 2, $m_S^{(i)}$ is reported in Table 1, $m_U^{(i)} = m_0^{(i+1)} = m_P^{(i+1)} + m_S^{(i+1)} + m_U^{(i+1)}$ $(i = 1, 2, 3)$, and $m_U^{(4)}$ is the payload mass.

As the thrust vector is assumed to be coplanar with the position and velocity vectors **r** and **v**, the angle α_T suffices to define its direction, which is taken clockwise from **v**. The state **x** is continuous across stage separations, which occur at times t_{b1}, t_{b2}, and t_{b3}. Finally, the fourth stage trajectory is assumed to be composed of two phases:

- a coast arc $(t_{b3} \le t \le t_{co})$, where $T = 0$, and
- a thrust phase $(t_{co} \le t \le t_f)$, where the time history of T is reported in Figure 3.

3.2 Formulation of the Problem

Minimizing the propellant needed for a rocket to reach a prescribed final operational orbit is a typical ascent trajectory optimization problem. This can be solved by finding the optimal control time history and the optimal thrust and coast durations.

The desired operational orbit is assumed to be circular and equatorial. If R_E (= 6378.136 km) denotes the Earth radius, the following conditions at orbit injection must hold:

$$r_f = R_E + 800 \text{ km} \qquad \gamma_f = 0 \qquad v_f = \sqrt{\frac{\mu_E}{r_f}} - \omega_E r_f \tag{11}$$

The rocket is launched from an equatorial site and toward the East direction, with an initial flight path angle of 86 deg. As the final orbit is equatorial as well, the ascent path lies entirely on the equatorial plane, because any out-of-plane motion would imply a useless waste of propellant. It is worth remarking that the launch vehicle dynamics is not affected by the initial location along the equatorial line. Moreover, the equations of motion (6), (8), and (9) do not depend on ξ, as well as the final conditions at injection (11). These circumstances imply that the state vector includes only r, v, and γ, i.e., $\mathbf{x}$ (= $[r\ \gamma\ v]^T$). The control vector includes α_T, i.e., $\mathbf{u} = \alpha_T$.

The initial conditions at liftoff are

$$r_i = R_E \qquad \gamma_i = 86 \text{ deg} \qquad v_i = 0 \tag{12}$$

The initial mass of the launch vehicle corresponds to the configuration with a payload of 150 kg. As the thrust duration for each of the first three stages is specified and equal to the nominal thrust duration, minimizing the propellant needed to reach the final orbit is equivalent to minimizing the thrust duration of the upper stage. Any propellant mass saving can be regarded as an additional mass available for the payload (other than the nominal 150 kg). Hence, the objective function is

$$J = k_J (t_f - t_{co}) \tag{13}$$

where k_J is an arbitrary positive constant.

If q is the dynamic pressure ($q = \frac{1}{2}\rho v^2$), for the rocket structural integrity the following constraint must be satisfied at all times:

$$q|\alpha| \leq 10^5 \text{ Pa} \cdot \text{deg} \tag{14}$$

3.3 First-Order Necessary Conditions for a Local Extremal

A common approximating assumption for rocket trajectory optimization is the alignment of the thrust direction with the rocket longitudinal axis. This circumstance implies that

$$\alpha_T(t) = \alpha(t) \tag{15}$$

Hence, the optimization problem consists in finding the optimal $\alpha(t)$ such that J is minimized and $q|\alpha| \leq 10^5$ Pa·deg at any time.

To obtain the necessary conditions for an optimal solution, a Hamiltonian H and a function of the boundary condition Φ are introduced as

$$\begin{aligned} H := {} & \lambda_r v \sin\gamma + \lambda_\gamma \frac{T}{mv}\sin\alpha + \lambda_\gamma\left(\frac{v}{r} - \frac{\mu_E}{r^2 v}\right)\cos\gamma + \lambda_\gamma \frac{L}{mv} + 2\lambda_\gamma \omega_E \\ & + \lambda_\gamma \frac{\omega_E^2 r}{v}\cos\gamma + \lambda_v \frac{T}{m}\cos\alpha - \lambda_v \frac{\mu_E}{r^2}\sin\gamma - \lambda_v \frac{D}{m} + \lambda_v \omega_E^2 r \sin\gamma \end{aligned} \tag{16}$$

$$\begin{aligned} \Phi := {} & \nu_1 (r_0 - R_E) + \nu_2 (\gamma_0 - \gamma_i) + \nu_3 (v_0 - v_i) \\ & + \nu_4 \left[r_f - (R_E + 800\ \text{km})\right] + \nu_5 \gamma_f + \nu_6 \left[v_f - (v_{If} - \omega_E r_f)\right] + k_J (t_f - t_{co}) \end{aligned} \tag{17}$$

where $\boldsymbol{\lambda}$ ($\equiv [\lambda_r \ \ \lambda_\gamma \ \ \lambda_v]^T$) and $\boldsymbol{\nu}$ ($\equiv [\nu_1 \ \ \nu_2 \ \ \nu_3 \ \ \nu_4 \ \ \nu_5 \ \ \nu_6]^T$) represent, respectively, the adjoint variable conjugate to the dynamics equations (6)–(9) and to the boundary conditions (11). The necessary conditions for optimality yield the following adjoint equations for the costate $\boldsymbol{\lambda}$:

$$\begin{aligned} \dot{\lambda}_r = -\Bigg[& \frac{\lambda_\gamma v \cos\gamma}{r^2} + \frac{2\lambda_\gamma \mu_E \cos\gamma}{r^3 v} + \frac{\lambda_\gamma}{v}\frac{\partial}{\partial r}\left(\frac{L}{m}\right) + \frac{\lambda_\gamma \omega_E^2 \cos\gamma}{v} \\ & + \frac{2\lambda_v \mu_E \sin\gamma}{r^3} - \lambda_v \frac{\partial}{\partial r}\left(\frac{D}{m}\right) + \lambda_v \omega_E^2 \sin\gamma \Bigg] \end{aligned} \tag{18}$$

$$\begin{aligned} \dot{\lambda}_\gamma = -\Bigg[& \lambda_r v \cos\gamma - \frac{\lambda_\gamma v \sin\gamma}{r} + \frac{\lambda_\gamma \mu_E \sin\gamma}{r^2 v} - \frac{\lambda_\gamma \omega_E^2 r \sin\gamma}{v} \\ & - \frac{\lambda_v \mu_E \cos\gamma}{r^2} + \lambda_v \omega_E^2 r \cos\gamma \Bigg] \end{aligned} \tag{19}$$

$$\begin{aligned} \dot{\lambda}_v = -\Bigg\{ & \lambda_r \sin\gamma - \frac{\lambda_\gamma T \sin\alpha}{mv^2} + \frac{\lambda_\gamma \cos\gamma}{r} + \frac{\lambda_\gamma \mu_E \cos\gamma}{r^2 v^2} \\ & + \lambda_\gamma \left[\frac{1}{v}\frac{\partial}{\partial v}\left(\frac{L}{m}\right) - \frac{L}{mv^2}\right] - \frac{\lambda_\gamma \omega_E^2 r \cos\gamma}{v^2} - \lambda_v \frac{\partial}{\partial v}\left(\frac{D}{m}\right) \Bigg\} \end{aligned} \tag{20}$$

in conjunction with the respective initial conditions

$$\lambda_{r,0} = -\nu_1 \quad \lambda_{\gamma,0} = -\nu_2 \quad \lambda_{v,0} = -\nu_3 \tag{21}$$

and final conditions

$$\lambda_{r,f} = \nu_4 \quad \lambda_{\gamma,f} = \nu_5 \quad \lambda_{v,f} = \nu_6 \tag{22}$$

The lift and drag acceleration derivatives may be rewritten as

$$\frac{\partial}{\partial r}\left(\frac{\eta}{m}\right) = \frac{\partial}{\partial r}\left(\frac{\frac{1}{2}\rho S C_\eta v^2}{m}\right) = \frac{1}{2} C_\eta \frac{S}{m} v^2 \frac{\partial \rho}{\partial r} - \frac{1}{2}\rho \frac{S}{m} v^2 \frac{\partial C_\eta}{\partial r} \tag{23}$$

$$\frac{\partial}{\partial v}\left(\frac{\eta}{m}\right) = \frac{\partial}{\partial v}\left(\frac{\frac{1}{2}\rho S C_\eta v^2}{m}\right) = \frac{v \rho S C_\eta}{m} + \frac{\frac{1}{2}\rho S v^2}{m v_S} \frac{\partial C_\eta}{\partial M} \tag{24}$$

where

$$\frac{\partial C_\eta}{\partial r} = \frac{\partial C_\eta}{\partial M}\frac{\partial M}{\partial r} = -\frac{\partial C_\eta}{\partial M}\frac{v}{v_S^2}\frac{\partial v_S}{\partial r} \tag{25}$$

and v_S is the sound speed, whereas $\eta = L, D$.

Due to the Weierstrass–Erdmann corner conditions [9] the adjoint variables are continuous across successive subarcs

$$\lambda_{in}^{(j+1)} = \lambda_{fin}^{(j)} \quad (j = 1, 2, 3, 4) \tag{26}$$

and the following relation that involves the Hamiltonian at the unknown time t_{co} must hold:

$$H_{fin}^{(4)} - H_{in}^{(5)} + \frac{\partial \Phi}{\partial t_{co}} = 0 \quad \Rightarrow \quad H_{fin}^{(4)} - H_{in}^{(5)} - k_J = 0 \tag{27}$$

The final time t_f is not specified, hence the following transversality condition must be satisfied:

$$\frac{\partial \Phi}{\partial t_f} + H_f^{(5)} = 0 \quad \Rightarrow \quad k_J + H_{fin}^{(5)} = 0 \tag{28}$$

The two equality constraints in Equations (27) and (28) can be replaced by a single equality constraint

$$H_{fin}^{(4)} - H_{in}^{(5)} + H_{fin}^{(5)} = 0 \tag{29}$$

in conjunction with the following inequality constraint:

$$H_{fin}^{(5)} = -k_J < 0 \tag{30}$$

In addition, the optimal control α^* can be expressed as a function of the costates through the Pontryagin minimum principle [28]

$$\alpha^* = \arg\min_{\alpha} H \tag{31}$$

This condition leads to determining the optimal value of α, either equal to the extremal values $(\alpha_{min}, \alpha_{max})$ or satisfying

$$\frac{\partial H}{\partial \alpha} = 0 \implies \lambda_\gamma \frac{T}{mv}\cos\alpha + \lambda_\gamma \frac{1}{mv}\frac{\partial L}{\partial \alpha} - \lambda_v \frac{T}{m}\sin\alpha - \frac{\lambda_v}{m}\frac{\partial D}{\partial \alpha} = 0 \tag{32}$$

where

$$\frac{\partial \eta}{\partial \alpha} = \frac{1}{2}\rho S V^2 \frac{\partial C_\eta}{\partial \alpha} \qquad (\eta = L,\ D) \tag{33}$$

The analytical conditions for optimality allow transforming the optimal control problem into a two-point boundary-value problem, where the unknowns are the initial values of the adjoint variables $\left(\lambda_{r,0}\ \ \lambda_{\gamma,0}\ \ \lambda_{v,0}\right)$, as well as the times t_{co} and t_f.

3.4 Method of Solution

For the problem at hand, a particle swarm algorithm (PSO) is run with the aim of determining a set of 5 unknown parameters that lead the dynamical system to satisfying the three final conditions stated by Equation (11), and the two conditions stated by Equations (29) and (30). Satisfying these conditions corresponds to minimizing the objective function J.

The PSO technique is a population-based method, where the population is represented by a swarm of N particles. Each particle is associated with a position vector and with a velocity vector. The initial population is randomly generated by introducing N particles, whose positions and velocities are (stochastically) uniformly distributed in the respective search spaces. Each particle represents a possible solution to the problem, and corresponds to a specific value of the objective function. The expressions for position and velocity update determine the swarm evolution toward the location of the globally optimal position, which corresponds to the globally optimal solution to the problem of interest. The central idea underlying the method is contained in the formula for velocity updating. This formula includes three terms with stochastic weights: the first term is the so-called inertial component

and for each particle is proportional to its velocity in the preceding iteration; the second component is termed the cognitive component, directed toward the personal best position, i.e., the best position experienced by the particle; and finally the third term is the social component, directed toward the global best position, i.e., the best position yet located by any particle in the swarm. The algorithm terminates when the maximum number of iterations N_{IT} is reached. The position vector of the best particle is expected to contain the optimal values of the unknown parameters, which correspond to the global minimum of J. In the scientific literature, this method has already been successfully applied to space trajectory optimization problems [19–21, 25, 26].

The parameter set employed in this problem is represented by the unknown initial values of the costate variables, as well the unknown coast duration ΔT_{co} and thrust duration ΔT_{b4} of the upper stage:

$$\left\{\lambda_{r,0}, \lambda_{\gamma,0}, \lambda_{v,0}, \Delta T_{co}, \Delta T_{b4}\right\}$$

In order to force the dynamical system to satisfying the four equality constraints stated by Equations (11) and (29) and the inequality constraint stated by Equation (30), an auxiliary objective function is introduced:

$$\begin{aligned} J' = & p_1 \left(r_f - R_E - 800\,\text{km}\right)^2 + p_2 \gamma_f^2 + p_3 \left(v_f - \sqrt{\frac{\mu_E}{r_f}} + \omega_E r_f\right)^2 \\ & + p_4 \left(H_{fin}^{(4)} - H_{in}^{(5)} + H_{fin}^{(5)}\right)^2 \end{aligned} \tag{34}$$

where the weights p_1, p_2, p_3, and p_4 affect the convergence of the method, and are set, respectively, to 100, 1, 5, and 1 (by trial-and-error). If the inequality constraint (30) is violated, a fictitious infinite value to the objective function J' is assigned.

In summary, the method of solution is based on the following points:

(a) as stated by Equation (31), the optimal control law α can be expressed as a function of the costates through the Pontryagin minimum principle. In order to find, at each time step, the value of α that minimizes the Hamiltonian function H, the stepwise regula falsi is employed, applied to small intervals (1 degree) for the angle α. This iterative method is very robust and usually guarantees a considerably fast convergence rate, which is a fundamental requisite when solving a computationally expensive problem like the one considered in this work. The regula falsi algorithm is used in order to identify the roots of Equation (32), in each subinterval (1 degree) in which the domain of α is divided. H is then evaluated at these points and its values are compared with the values of H at the boundaries of each subinterval, leading to the value of α for which H is minimized. The angle of attack α depends on time in a domain, whose extremals are always kept within the interval (-10 deg, 10 deg) while the rocket is in the atmosphere. More precisely, at a generic time t, the

value of α that minimizes the Hamiltonian function H is sought in the interval $(-(q\alpha)_{max}/q, (q\alpha)_{max}/q)$, where q is the dynamic pressure and $(q\alpha)_{max}$ is the maximum allowed value for the product $q|\alpha|$ below which the rocket structural integrity is preserved;

(b) the state equations (6)–(9) and the adjoint equations (18)–(20) are integrated numerically up to the final time t_f, with initial conditions partially specified and in part guessed as components of each particle;

(c) the five boundary conditions stated by Equations (11), (29), and (30) (four equality constraints and a single inequality constraint) are checked.

Enforcement of J' (Equation 34), in conjunction with the necessary conditions developed in the previous section, guarantees both feasibility and optimality.

3.5 Optimal Ascent Trajectory

The optimal ascending trajectory is determined by employing canonical units: the distance unit (DU) is the Earth radius, whereas the time unit (TU) is such that $\mu_E = 1\ \mathrm{DU}^3/\mathrm{TU}^2$. Thus, DU = 6378.136 km and TU = 806.8 sec. The PSO optimizer is used to find the unknown parameters. All the simulations are performed in the most favorable dynamical conditions, i.e., equatorial trajectory and launch toward the East direction.

The swarming optimizer is employed with the following settings: $N = 100$ (number of particles) and $N_{IT} = 300$ (number of iterations). It is worth remarking that the initial values of the Lagrange multipliers can be sought in the interval $[-1, 1]$, because the instantaneous optimal α^* is invariant under scaling of all the adjoints by a positive constant. Hence, the optimal values of the unknown parameters are sought in the following ranges:

$$-1 \leq \lambda_{r,0} \leq 1$$

$$-1 \leq \lambda_{\gamma,0} \leq 1$$

$$-1 \leq \lambda_{v,0} \leq 1$$

$$450\,\mathrm{sec} \leq \Delta T_{co} \leq 650\,\mathrm{sec}$$

$$10\,\mathrm{sec} \leq \Delta T_{b4} \leq 32\,\mathrm{sec}$$

The time histories of the optimal state, optimal control, and of $(q|\alpha|)$ obtained at the end of the optimization process are portrayed in Figure 6. The optimal coast and thrust durations are the following:

$$\Delta T_{co} = 556.40\,\mathrm{sec} \qquad \Delta T_{b4} = 25.81\,\mathrm{sec}$$

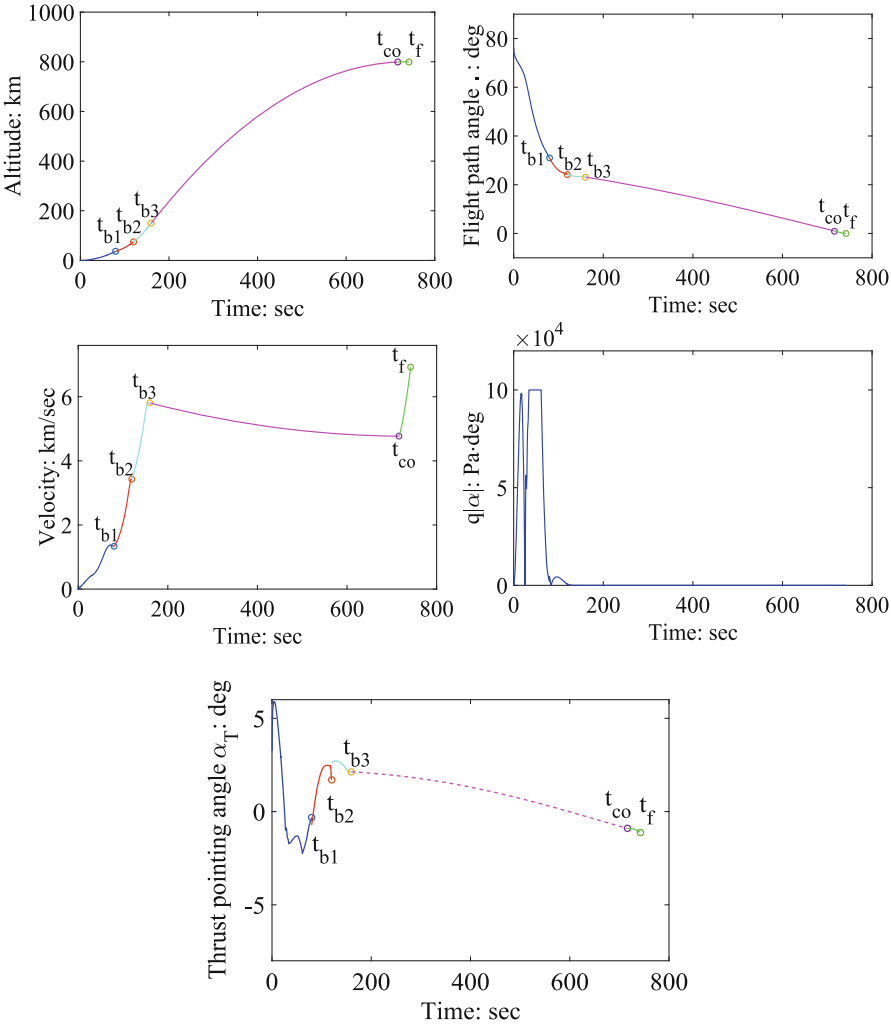


Figure 6 Optimal state variables, optimal control, and $q|\alpha|$ time histories

The optimal thrust duration ΔT_{b4} of the fourth stage is approximately 6 s less than the nominal thrust duration (32 s), which corresponds to using all the onboard propellant. This means that the target orbit can be reached by saving 32 kg of propellant mass. This is equivalent to boarding a payload with a supplementary mass of 32 kg, in addition to the nominal payload mass (150 kg).

4 Variable-Time-Domain Neighboring Optimal Guidance

The problem of driving the launch vehicle along the specified (nominal) optimal path requires defining the corrective actions aimed at compensating nonnominal flight conditions. This means that a feedback control law, or, equivalently, a closed-loop guidance algorithm, is to be defined, on the basis of the current state of the vehicle, evaluated at prescribed sampling times.

The recently introduced [22, 23], iterative variable-time-domain neighboring optimal guidance (VTD-NOG) uses the optimal trajectory as the reference path, with the final intent of determining the control correction at each sampling time $\{t_k\}_{k=0,\ldots,n_S}$. These are the times at which the displacement between the actual trajectory, associated with $\mathbf{x}$, and the nominal trajectory, corresponding to $\mathbf{x}^*$, is evaluated, to yield

$$d\mathbf{x}_k = \mathbf{x}(t_k) - \mathbf{x}_k^*(t_k) \tag{35}$$

The total number of sampling times, n_S, is unspecified, whereas the actual time interval between two successive sampling times is given and denoted with Δt_S, $\Delta t_S = t_{k+1} - t_k$. It is apparent that a fundamental ingredient needed to implement VTD-NOG is the formula for determining the overall time of flight $t_f^{(k)}$ at time t_k. This is equivalent to finding the time-to-go $\left(t_f^{(k)} - t_k\right)$ at t_k. Section 4.1 addresses this issue.

As a preliminary step, in the time interval $[t_0, t_f]$ a single-stage dynamical system obeys the vector equation

$$\dot{\mathbf{x}} = \tilde{\mathbf{f}}(\mathbf{x}, \mathbf{u}, \tilde{\mathbf{a}}, t) \tag{36}$$

where $\mathbf{a}$ is a vector of unknown time-independent parameters. Using the dimensionless normalized time τ defined as

$$\tau := \frac{t}{t_f} \quad \Rightarrow \quad \tau_0 \equiv 0 \leq \tau \leq 1 \equiv \tau_f \tag{37}$$

Equation (36) can be rewritten as

$$\dot{\mathbf{x}} = \tilde{\mathbf{f}}(\mathbf{x}, \mathbf{u}, \tilde{\mathbf{a}}, t_f \tau) \tag{38}$$

Let the dot denote the derivative with respect to τ hence forward in Sect. 4. If $\mathbf{a} := \begin{bmatrix} \tilde{\mathbf{a}}^T & t_f \end{bmatrix}^T$, the state equations (38) are finally expressed as

$$\dot{\mathbf{x}} = t_f \tilde{\mathbf{f}}(\mathbf{x}, \mathbf{u}, \tilde{\mathbf{a}}, t_f \tau) =: \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{a}, \tau) \tag{39}$$

Introduction of the vector **a** implies the fulfillment of a further necessary condition [22], which involves the time-varying adjoint variable $\boldsymbol{\mu}$.

4.1 Fundamentals

The fundamental principle that underlies the VTD-NOG scheme consists in finding the control correction $\delta\mathbf{u}(\tau)$ in the generic interval $[\tau_k, \tau_{k+1}]$ such that the second differential of J is minimized

$$\begin{aligned} d^2J = \int_{\tau_k}^{1} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \\ d\mathbf{a} \end{bmatrix}^T \begin{bmatrix} H_{\mathbf{xx}} & H_{\mathbf{xu}} & H_{\mathbf{xa}} \\ H_{\mathbf{ux}} & H_{\mathbf{uu}} & H_{\mathbf{ua}} \\ H_{\mathbf{ax}} & H_{\mathbf{au}} & H_{\mathbf{aa}} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{u} \\ d\mathbf{a} \end{bmatrix} d\tau \\ + \begin{bmatrix} d\mathbf{x}_k \\ d\mathbf{x}_f \\ d\mathbf{a} \end{bmatrix}^T \begin{bmatrix} \Phi_{\mathbf{x}_k\mathbf{x}_k} & \mathbf{0}_{n\times n} & \mathbf{0}_{n\times p} \\ \mathbf{0}_{n\times n} & \Phi_{\mathbf{x}_f\mathbf{x}_f} & \Phi_{\mathbf{x}_f\mathbf{a}} \\ \mathbf{0}_{p\times n} & \Phi_{\mathbf{a}\mathbf{x}_f} & \Phi_{\mathbf{aa}} \end{bmatrix} \begin{bmatrix} d\mathbf{x}_k \\ d\mathbf{x}_f \\ d\mathbf{a} \end{bmatrix} \end{aligned} \tag{40}$$

while holding the first-order expansions of the state equations and the related final conditions. Minimizing the objective (40) is equivalent to solving the accessory optimization problem, defined in the interval $[\tau_k, 1]$. Solving this problem involves the linear expansion of the state and costate equations and the related boundary conditions about the optimal trajectory.

Several analytical developments [22] lead to defining the updating law for the overall time of flight. This relation, not reported for the sake of brevity, derives directly from the natural extension of the accessory optimization problem to the time interval $[\tau_k, 1]$. Moreover, if $dt_f^{(k)}$ denotes the correction on t_f^* evaluated at τ_k, then $t_f^{(k)} = t_f^* + dt_f^{(k)}$. As the sampling interval Δt_S is specified, the general formula for τ_k is

$$\tau_k = \sum_{j=0}^{k-1} \frac{\Delta t_S}{t_f^{(j)}} \tag{41}$$

The overall number of intervals n_S is found at the first occurrence of the following condition, associated with the termination of VTD-NOG:

$$\sum_{j=0}^{n_S-1} \frac{\Delta t_S}{t_f^{(j)}} \geq 1 \qquad \Rightarrow \qquad \tau_{n_S} = 1 \tag{42}$$

The introduction of the normalized time τ is of crucial importance, because all the gain matrices remain defined in the normalized interval [0,1] and cannot become singular. Moreover, the limiting values $\{\tau_k\}_{k=0,\ldots,n_S-1}$ are dynamically calculated at

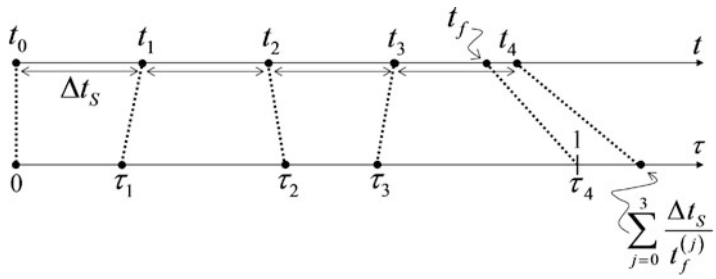


Figure 7 Illustrative sketch of the relations between the two time domains

each sampling time using Equation (41), while the sampling instants in the actual time domain are specified and equally spaced (Figure 7). Also the termination criterion (42) has a logical, consistent definition, and corresponds to the upper bound of the interval [0,1], to which τ is constrained.

In general, the definition of a neighboring optimal path requires the numerical backward integration of the sweep equations [9] in order to determine several time-varying gain matrices. However, the use of a variable time domain requires finding new sweep equations [22] for these matrices

$$\dot{\hat{\mathbf{S}}} = -\hat{\mathbf{S}}\mathbf{A} + \hat{\mathbf{S}}\mathbf{B}\hat{\mathbf{S}} + \left[\hat{\mathbf{S}}\mathbf{D}\boldsymbol{\alpha}^{-1} + \mathbf{W}\mathbf{F}\boldsymbol{\alpha}^{-1} + \mathbf{E}\boldsymbol{\alpha}^{-1}\right]\mathbf{m}^T - \mathbf{W}\mathbf{E}^T - \mathbf{W}\mathbf{D}^T\hat{\mathbf{S}} - \mathbf{C} - \mathbf{A}^T\hat{\mathbf{S}} \tag{43}$$

$$\dot{\mathbf{R}}^T = \mathbf{R}^T\mathbf{B}\hat{\mathbf{S}} - \mathbf{R}^T\mathbf{A} - \mathbf{R}^T\mathbf{B}\mathbf{W}\mathbf{m}^T \tag{44}$$

$$\dot{\mathbf{m}}^T = -\mathbf{m}^T\mathbf{A} + \mathbf{m}^T\mathbf{B}\hat{\mathbf{S}} - \mathbf{m}^T\mathbf{B}\mathbf{W}\mathbf{m}^T - \mathbf{E}^T - \mathbf{D}^T\hat{\mathbf{S}} + \mathbf{D}^T\mathbf{W}\mathbf{m}^T \tag{45}$$

$$\dot{\mathbf{Q}} = -\mathbf{R}^T\mathbf{B}\mathbf{W}\mathbf{n}^T \tag{46}$$

$$\dot{\mathbf{n}} = -\mathbf{R}^T\left(\mathbf{D} + \mathbf{B}\mathbf{W}\boldsymbol{\alpha}\right) \tag{47}$$

$$\dot{\boldsymbol{\alpha}} = \mathbf{D}^T\mathbf{W}\boldsymbol{\alpha} - \mathbf{F} - \mathbf{m}^T\mathbf{B}\mathbf{W}\boldsymbol{\alpha} - \mathbf{m}^T\mathbf{D} \tag{48}$$

The gain matrices involved in the sweep method, i.e., $\mathbf{S}$, $\hat{\mathbf{S}}$, $\mathbf{R}$, $\mathbf{Q}$, $\mathbf{n}$, $\mathbf{m}$, and $\boldsymbol{\alpha}$, can be integrated backward in two steps:

1. in $[\tau_{sw}, 1]$ the equations of the classical sweep method [9, 22], with the respective boundary conditions, are used.
2. in $[0, \tau_{sw}]$ Equations (43)–(48) are used; $\mathbf{R}$, $\mathbf{Q}$, $\mathbf{n}$, $\mathbf{m}$, and $\boldsymbol{\alpha}$ are continuous across the time τ_{sw}, whereas $\hat{\mathbf{S}}$ is given by $\hat{\mathbf{S}} := \mathbf{S} - \mathbf{U}\mathbf{V}^{-1}\mathbf{U}^T$; in this work τ_{sw} is set to 0.99.

The analytical developments leading to Equations (43)–(48) are described in full detail in [22] together with the related boundary conditions, as well as the definitions of matrices **A**, **B**, **C**, **D**, **E**, **F**, **U**, **V**, and **W**.

In short, adoption of a normalized, variable time domain has the beneficial effect of (i) deriving the updating law from optimal control theory, (ii) defining a consistent termination criterion, and (iii) constraining the gain matrices to an interval where they do not become singular. As a side effect, the modified sweep equations (43)–(48), more complicated than the usual counterparts (holding for neighboring optimal paths), must be integrated numerically.

4.2 Preliminary Offline Computations and Algorithm Structure

The implementation of VTD-NOG requires several preliminary computations that can be completed offline and stored in the onboard computer. First of all, the optimal trajectory is to be determined, together with the related state, costate, and control variables, which are assumed as the nominal ones. In the time domain τ these can be either available analytically or represented as sequences of equally spaced values, e.g., $\mathbf{u}_i^* = \mathbf{u}^*(\tau_i)$ ($i = 0, \ldots, n_D$; $\tau_0 = 0$ and $\tau_{n_D} = 1$). However, in the presence of perturbations, VTD-NOG determines the control corrections $\delta\mathbf{u}(\tau)$ in each interval $[\tau_k, \tau_{k+1}]$, where the values $\{\tau_k\}$ never coincide with the equally spaced values $\{\tau_i\}$ used for $\mathbf{u}_i^*$. Hence, regardless of the number of points used to represent the control correction $\delta\mathbf{u}(\tau)$ in $[\tau_k, \tau_{k+1}]$, it is apparent that a suitable interpolation is to be adopted for the control variable $\mathbf{u}^*$ (provided that no analytical expression is available). In this way, the value of $\mathbf{u}^*$ can be evaluated at any arbitrary time in the interval $0 \leq \tau \leq 1$. For the same reason also the nominal state $\mathbf{x}^*$ and costate $\boldsymbol{\lambda}^*$ need to be interpolated. If a sufficiently large number of points are selected (e.g., $n_D = 1001$), then piecewise linear interpolation is a suitable option. The successive step is the analytical derivation of the matrices

$$\left\{ \begin{array}{l} \mathbf{f}_{\mathbf{x}}, \mathbf{f}_{\mathbf{u}}, \mathbf{f}_{\mathbf{a}}, H_{\mathbf{xx}}, H_{\mathbf{xu}}, H_{\mathbf{x\lambda}}, H_{\mathbf{xa}}, H_{\mathbf{ux}}, H_{\mathbf{uu}}, H_{\mathbf{ua}}, H_{\mathbf{u\lambda}}, H_{\mathbf{ax}}, H_{\mathbf{au}}, H_{\mathbf{aa}}, H_{\mathbf{a\lambda}}, \\ \boldsymbol{\psi}_{\mathbf{x}_f}, \boldsymbol{\psi}_{\mathbf{x}_0}, \boldsymbol{\psi}_{\mathbf{a}}, \boldsymbol{\Phi}_{\mathbf{x}_0\mathbf{x}_0}, \boldsymbol{\Phi}_{\mathbf{x}_0\mathbf{a}}, \boldsymbol{\Phi}_{\mathbf{x}_f\mathbf{x}_f}, \boldsymbol{\Phi}_{\mathbf{x}_f\mathbf{a}}, \boldsymbol{\Phi}_{\mathbf{a}\mathbf{x}_f}, \boldsymbol{\Phi}_{\mathbf{aa}} \end{array} \right\} \tag{49}$$

Then, they are evaluated along the nominal trajectory and linearly interpolated, as well as **A**, **B**, **C**, **D**, **E**, and **F**, whose expressions are reported in [24]. Subsequently, the two-step backward integration of the sweep equations described in Sect. 4.1 is performed, and yields the gain matrices $\hat{\mathbf{S}}$, **R**, **m**, **Q**, **n**, and $\boldsymbol{\alpha}$, using also the analytic expressions of **W**, **U**, and **V**. The linear interpolation of all the matrices not yet interpolated concludes the preliminary computations.

On the basis of the optimal reference path, at each time τ_k the VTD-NOG algorithm determines the time of flight and the control correction. More specifically, after setting the actual sampling time interval Δt_S, at each τ_k ($k = 0, \ldots, n_S - 1$; $\tau_0 = 0$) the following steps implement the feedback guidance scheme [22]:

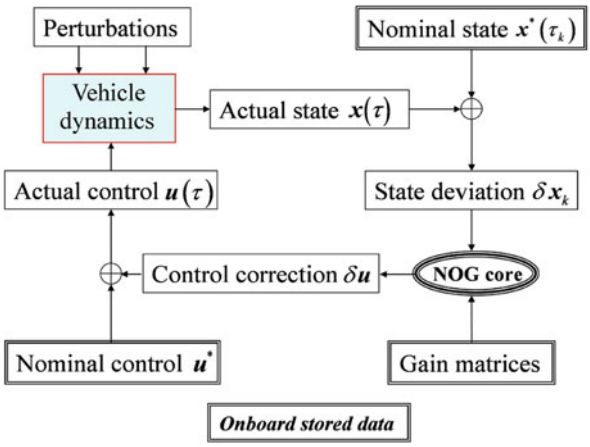


Figure 8 Block diagram of VTD-NOG

1. Evaluate $\delta \mathbf{x}_k$.
2. Calculate the correction $dt_f^{(k)}$ and the updated time of flight $t_f^{(k)}$.
3. Calculate the limiting value τ_{k+1}.
4. Evaluate $\delta \boldsymbol{\lambda}_k$.
5. Integrate numerically the linear differential system for the displacements $\delta \mathbf{x}$, $\delta \boldsymbol{\lambda}$, and $\delta \boldsymbol{\mu}$.
6. Determine the control correction $\delta \mathbf{u}(\tau)$ in $[\tau_k, \tau_{k+1}]$ by means of the following relation [22]:

$$\delta \mathbf{u} = -H_{\mathbf{uu}}^{-1} \left(H_{\mathbf{ux}} \delta \mathbf{x} + H_{\mathbf{ua}} \delta \mathbf{a} + H_{\mathbf{x\lambda}} \delta \boldsymbol{\lambda} \right) \tag{50}$$

7. Points 1–6 are repeated after increasing k by 1, until Equation (42) is satisfied.

Figure 8 portrays a block diagram that illustrates the feedback structure of VTD-NOG. At each sampling time τ_k the nominal state $\boldsymbol{x}^\star(\tau_k)$ is compared with the actual state $\boldsymbol{x}(\tau_k)$, and the state deviation $\delta \boldsymbol{x}_k$ is evaluated. This enters the NOG core, where previous steps 2–6 are completed. The time-varying gain matrices, computed offline and stored onboard, are used in the NOG core as well. The control correction $\delta \boldsymbol{u}(\tau)$ is added to the nominal control $\boldsymbol{u}^\star(\tau)$, and the resulting actual control $\boldsymbol{u}(\tau)$ is employed to drive the launch vehicle in the interval $[\tau_k, \tau_{k+1}]$.

5 Guidance of the Upper Stage Ascent Path

The VTD-NOG algorithm is applied to the minimum-propellant ascent path of the upper stage of the Scout, with the aim of driving the rocket to the target circular orbit, in the presence of nonnominal flight conditions. Due to perturbations from the nominal conditions, the actual upper stage ascending path is not a two-dimensional path as the nominal optimal trajectory, which means that the rocket motion does not take place entirely on the equatorial plane.

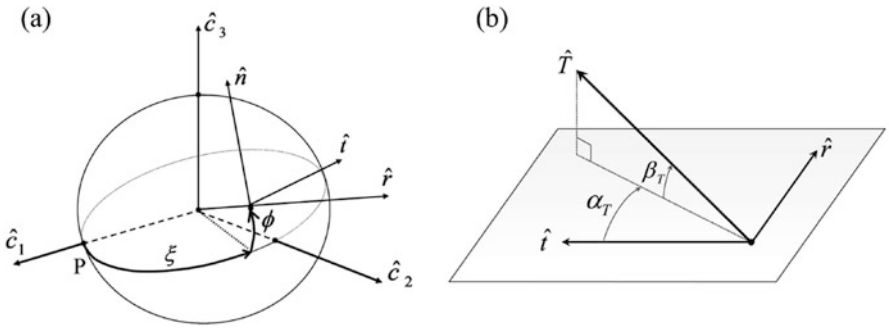


Figure 9 Reference frames **(a)**, and thrust angles α_T and β_T **(b)**

The upper stage motion can be described in a convenient inertial reference frame, aligned with the sequence of unit vectors $(\hat{c}_1, \hat{c}_2, \hat{c}_3)$ and with center coincident with the center of the Earth. The upper stage position can be identified by the following three variables: radius r, absolute longitude ξ, and latitude ϕ, portrayed in Figure 9a. The upper stage inertial velocity can be projected into the rotating reference frame $(\hat{r}, \hat{t}, \hat{n})$, where $\hat{r}$ is aligned with the position vector $\boldsymbol{r}$ and $\hat{t}$ is parallel to the $(\hat{c}_1, \hat{c}_2)$ plane, namely, the equatorial plane. The related components are denoted with (v_r, v_t, v_n) and termed, respectively, radial, transverse, and normal velocity component. Hence, the state vector of the rocket is the following:

$$\mathbf{x} := [r \;\; \xi \;\; \phi \;\; v_r \;\; v_t \;\; v_n]^T \tag{51}$$

The upper stage ascending trajectory is controlled by acting on the thrust direction, defined by the in-plane angle α_T and the out-of-plane angle β_T, both illustrated in Figure 9b. The thrust direction $\hat{T}$ is always aligned with the rocket longitudinal axis. The control vector is defined as

$$\mathbf{u} := [\alpha_T \;\; \beta_T]^T \tag{52}$$

The three-dimensional ascending path of the upper stage is governed by the following equations:

$$\dot{r} = v_r \tag{53}$$

$$\dot{\xi} = \frac{v_t}{r \cos\phi} \tag{54}$$

$$\dot{\phi} = \frac{v_n}{r} \tag{55}$$

$$\dot{v}_r = -\frac{\mu_E}{r^2} + \frac{v_t^2 + v_n^2}{r} + \frac{T}{m}\sin\alpha_T \cos\beta_T \tag{56}$$

$$\dot{v}_t = \frac{v_t}{r}\left(v_n \tan\phi - v_r\right) + \frac{T}{m}\cos\alpha_T \cos\beta_T \tag{57}$$

$$\dot{v}_n = -\frac{v_t^2}{r}\tan\phi - \frac{v_r v_n}{r} + \frac{T}{m}\sin\beta_T \tag{58}$$

Equations (53)–(58) can be written in the form of vector state equations

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\ \mathbf{u},\ t) \tag{59}$$

Using Equations (38)–(39) this expression can be rewritten in terms of the normalized time τ, in order to apply VTD-NOG.

5.1 Nominal Path

As the guidance algorithm uses the inertial variables to determine the necessary control correction at each sampling time, also the nominal state, costate, and control (which identify the two-dimensional minimum-propellant ascending path) have to be expressed in terms of inertial variables. According to the Bellman optimality principle, any subarc of an optimal path is itself optimal. This means that the last propulsive arc of the minimum-propellant ascending path is an optimal arc. Hence, a new optimization process, which employs inertial variables, is performed, in order to find the optimal upper stage ascending path. The optimal arc lies on the equatorial plane, and it is described in terms of the state variables r, ξ, v_r, and v_t, while the latitude ϕ, the normal velocity v_n, and the thrust out-of-plane angle β_T are equal to zero.

The optimization problem aims at finding the optimal thrust time history $\alpha_T(t)$, which leads the rocket to reach its target orbit, given the state at ignition of the fourth stage:

$$r_i = R_E + 798.984\,\text{km} \quad , \quad v_{r,i} = 0.096\ \text{km/sec} \quad , \quad v_{t,i} = 5.291\,\text{km/sec} \tag{60}$$

The final state variables must satisfy the following conditions:

$$r_f = R_E + 800\,\text{km} \quad , \quad v_{r,f} = 0 \quad , \quad v_{t,f} = \sqrt{\frac{\mu_E}{r_f}} \tag{61}$$

The optimal arc determined through the new optimization process corresponds, with great accuracy, to the last propulsive arc of the entire optimal ascending path

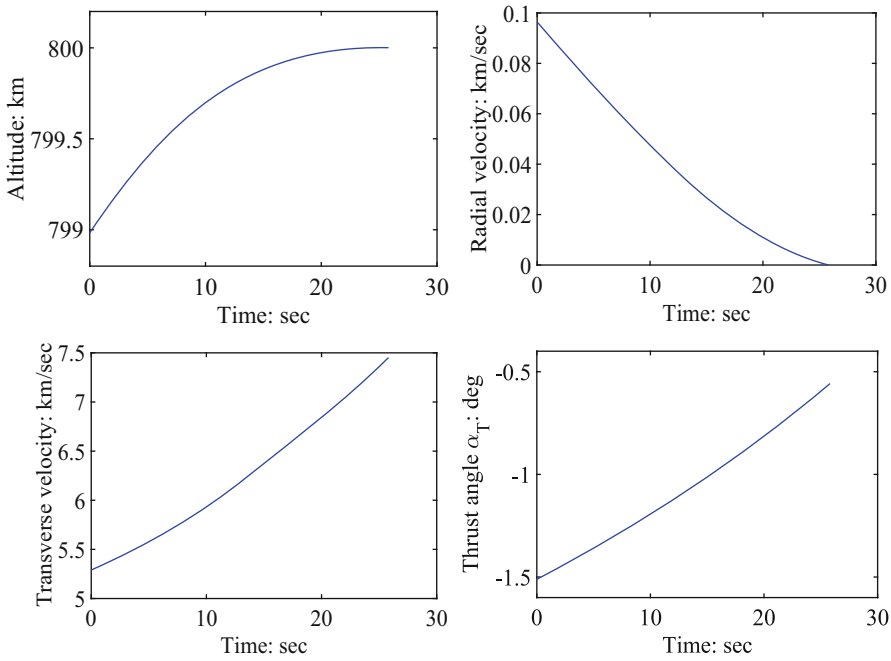


Figure 10 Optimal inertial state and control variables

presented in Sect. 3.5. The deviations between the final state variables and the respective desired values are

$$\Delta r_f = 0.983 \text{ m}\,, \quad \Delta v_{r_f} = -0.012 \text{ m/sec}\,, \quad \Delta v_{t_f} = -0.018 \text{ m/sec} \tag{62}$$

The optimal inertial state and control variables are portrayed in Figure 10.

5.2 Perturbed Paths

The neighboring optimal guidance algorithm proposed in this work is applied to the upper stage orbit injection, in the presence of nonnominal flight conditions arising from errors on the initial conditions. Several Monte Carlo simulations are run, with the intent of obtaining some useful statistical information on the accuracy of VTD-NOG, in the presence of the previously mentioned deviations, which are simulated stochastically.

More specifically, for the initial conditions errors on altitude and declination are assumed, with Gaussian distributions, zero mean values and standard deviations

$$r_0^{(\sigma)} = 1.5 \text{ km} \qquad \phi_0^{(\sigma)} = 0.009 \text{ deg} \tag{63}$$

Moreover, errors on the three components of velocity are considered, with zero mean value and standard deviation equal to 10 m/sec, i.e.,

$$v_{r0}^{(\sigma)} = 10 \text{ m/sec} \qquad v_{t0}^{(\sigma)} = 10 \text{ m/sec} \qquad v_{n0}^{(\sigma)} = 10 \text{ m/sec} \tag{64}$$

At the end of VTD-NOG, two statistical quantities are evaluated, i.e., the mean value and the standard deviation for all the outputs of interest.

A Monte Carlo campaign (MC) is performed, assuming errors on the initial state. The campaign includes 100 numerical simulations, adopting a time interval $\Delta t_s = 0.5$ sec. Figures 11, 12, 13 depict the state and control variables obtained during the simulations that form the Monte Carlo campaign. Table 3 reports the statistics on the errors at injection, where $\Delta \bar{\chi}_f$ is the mean value of the displacement from the desired final value for variable χ, $\chi_f^{(\sigma)}$ is the related standard deviation, $\bar{t}_f$ is the mean value of the time of flight, and $t_f^{(\sigma)}$ is its standard deviation. Inspection of this table reveals that VTD-NOG guarantees orbit injection with very satisfactory accuracy, in the presence of the previously mentioned nonnominal flight conditions.

As a final remark, the runtime of VTD-NOG on an Intel i5-3570k at 3.40 GHz does not exceed the nominal time of flight, and this guarantees that NOG can be implemented in real time.

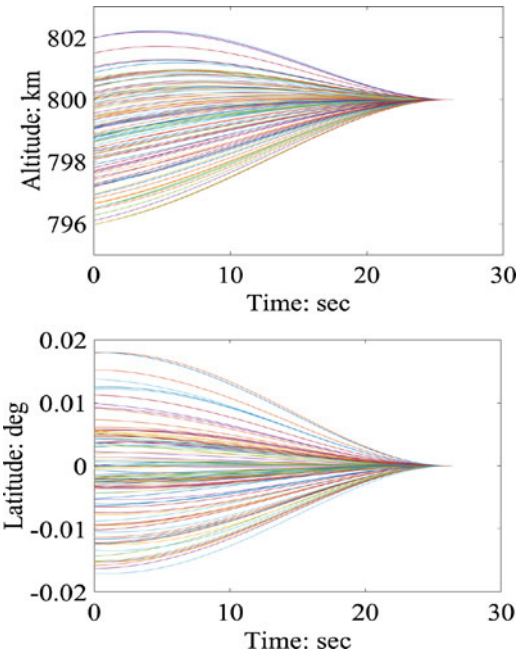


Figure 11 Altitude and latitude time histories obtained in the MC campaign

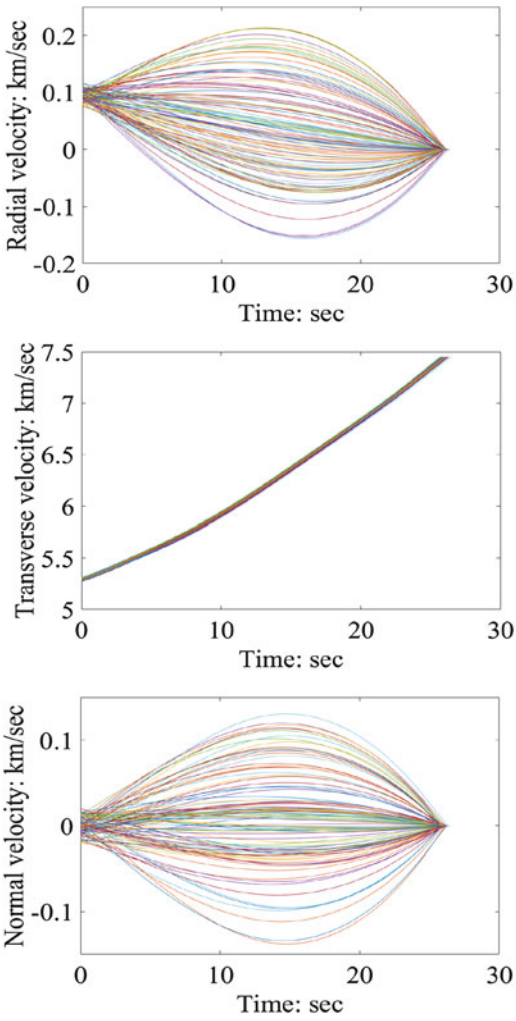


Figure 12 Inertial velocity components time histories obtained in the MC campaign

6 Concluding Remarks

Multistage launch vehicles have been repeatedly used and are currently employed to place spacecraft and satellites in their operational orbits. This chapter is focused on the ascent trajectory optimization and neighboring optimal guidance of multistage launch vehicles, with special focus on an existing rocket, i.e., the Scout. Accurate modeling of the rocket aerodynamics and propulsion, on the basis of realistic or experimental data, represents a prerequisite for path optimization. This work describes and applies suitable interpolating techniques with this intent. Ascent trajectory optimization is then addressed through the joint use of the necessary conditions for optimality, derived for the entire trajectory, and a heuristic technique,

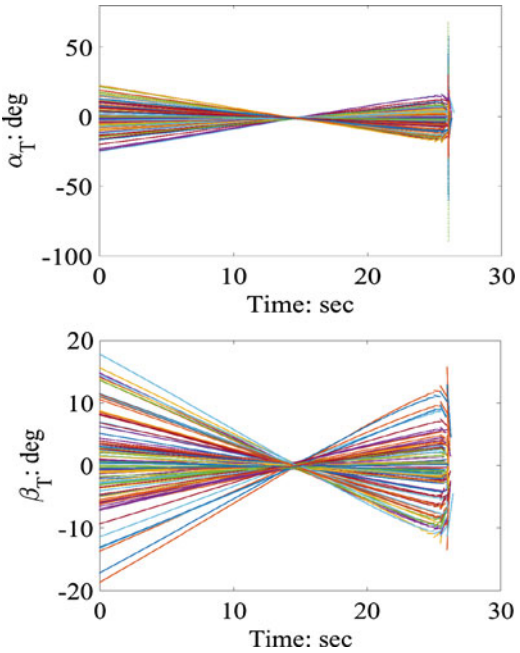


Figure 13 Thrust angles time histories obtained in the MC campaign

Table 3 Statistics on the final state components and the time of flight

Statistics		
$\Delta\bar{r}_f$	(m)	9.6e−4
$r_f^{(\sigma)}$	(m)	1.5e−5
$\Delta\bar{\phi}_f$	(deg)	−6.7e−9
$\phi_f^{(\sigma)}$	(deg)	7.1e−8
$\Delta\bar{v}_{rf}$	(m/sec)	−3.5e−5
$v_{rf}^{(\sigma)}$	(m/sec)	2.6e−4
$\Delta\bar{v}_{tf}$	(m/sec)	−5.1e−4
$v_{tf}^{(\sigma)}$	(m/sec)	6.3e−4
$\Delta\bar{v}_{nf}$	(m/sec)	−2.9e−5
$v_{nf}^{(\sigma)}$	(m/sec)	1.7e−4
$\bar{t}_f$	(sec)	25.9
$t_f^{(\sigma)}$	(sec)	0.2

i.e., the particle swarm algorithm. The structural constraint related to the dynamic pressure is enforced at all times during the ascent path. The resulting optimal trajectory satisfies all the analytical conditions for optimality to an excellent accuracy. However, in concrete scenarios, precise orbit injection must be guaranteed also in the presence of nonnominal flight conditions. This circumstance implies the need of a closed-loop guidance algorithm for the upper stage. This research outlines and applies the recently introduced, general-purpose variable-time-domain neighboring optimal guidance algorithm (VTD-NOG), which assumes the optimal path as the reference trajectory. Some fundamental, original features of VTD-NOG allow overcoming the main difficulties related to the use of former neighboring optimal guidance schemes, in particular the occurrence of singularities and the lack of an efficient law for the iterative real-time update of the time of flight. This is achieved by adopting a normalized time domain, which leads to defining a novel updating law for the time of flight, a new termination criterion, and a new analytical formulation for the sweep method. Errors on position and velocity at ignition of the upper stage are assumed. A Monte Carlo campaign unequivocally demonstrates that VTD-NOG applied to the upper stage powered arc guarantees orbit injection with excellent accuracy, in spite of relevant deviations from nominal flight conditions.

References

1. Afshari, H.H., Novinzadeh, A.B., Roshanian, J.: Determination of nonlinear optimal feedback law for satellite injection problem using neighboring optimal control. Am. J. Appl. Sci. **6**(3), 430–438 (2009)
2. Bayley, D.J., Hartfield, Jr R.J., Burkhalter, J.E., Jenkins, R.M.: Design optimization of a space launch vehicle using a genetic algorithm. J. Spacecr. Rocket. **45**(4), 733–740 (2008)
3. Calise, A.J., Melamed, N., Lee, S.: Design and evaluation of a three-dimensional optimal ascent guidance algorithm. J. Guid. Control Dynam. **21**(6), 867–875 (1998)
4. Calise, A.J., Tandon, S., Young, D.H., Kim, S.: Further Improvements to a Hybrid Method for launch Vehicle Ascent Trajectory Optimization. AIAA Guidance, Navigation and Control Conference and Exhibit, Denver (2000)
5. Charalambous, C.B., Naidu, S.N., Hibey, J.L.: Neighboring optimal trajectories for aeroassisted orbital transfer under uncertainties. J. Guid. Control Dynam. **18**(3), 478–485 (1995)
6. Di Sotto, E., Teofilatto, P.: Semi-analytical formulas for launcher performance evaluation. J. Guid. Control Dynam. **25**(3), 538–545 (2002)
7. Gath, P.F., Calise, A.J.: Optimization of launch vehicle ascent trajectories with path constraints and coast arcs. J. Guid. Control Dynam. **24**(2), 296–304 (2001)
8. Hull, D.G.: Robust Neighboring Optimal Guidance for the Advanced Launch System. NASA-CR-192087, Austin (1993)
9. Hull, D.G.: Optimal Control Theory for Applications. Springer, New York, pp. 199–254 (2003)
10. Jamilnia, R., Naghash, A.: Simultaneous optimization of staging and trajectory of launch vehicles using two different approaches. Aerosol Sci. Technol. **23**, 65–92 (2012)
11. Lu, P.: Optimal feedback control laws using nonlinear programming. J. Optim. Theory Appl. **71**(3), 599–611 (1991)
12. Lu, P., Griffin, B.J., Dukeman. G.A., Chavez, F.R.: Rapid optimal multiburn ascent planning and guidance. J. Guid. Control Dynam. **31**(6), 45–52 (2008)
13. Lu, P., Pan, B.: Trajectory optimization and guidance for an advanced launch system. 30th Aerospace Sciences Meeting and Exhibit, Reno (1992)

14. Mangiacasale, L.: Meccanica del volo atmosferico. Terne di riferimento, equazioni di moto, linearizzazione, stabilità. Ingegneria 2000, Rome (2008)
15. Martinon, P., Bonnans, F., Laurent-Varin, J., Trélat, E.: Numerical study of optimal trajectories with singular arcs for an Ariane 5 launcher. J. Guid. Control Dynam. **32**(1), 51–55 (2009)
16. Miele, A.: Multiple-subarc gradient-restoration algorithm, part 1: algorithm structure. J. Optim. Theory Appl. **116**(1), 1–17 (2003)
17. Miele, A., Multiple-subarc gradient-restoration algorithm, part 2: application to a multistage launch vehicle design. J. Optim. Theory Appl. **116**(1), 19–19 (2003)
18. Pontani, M.: Particle swarm optimization of ascent trajectories of multistage launch vehicles. Acta Astronaut. **94**(2), 852–864 (2014)
19. Pontani, M., Conway, B.A.: Particle swarm optimization applied to space trajectories. J. Guid. Control Dynam. **33**(5), 1429–1441 (2010)
20. Pontani, M., Conway, B.A.: Particle swarm optimization applied to impulsive orbital transfers. Acta Astronaut. **74**, 141–155 (2012)
21. Pontani, M., Conway, B.A.: Optimal finite-thrust rendezvous trajectories found via particle swarm algorithm. J. Spacecr. Rocket. **50**(6), 1222–1234 (2013)
22. Pontani, M., Cecchetti, G., Teofilatto, P.: Variable-time-domain neighboring optimal guidance, part 1: algorithm structure. J. Optim. Theory Appl. **166**(1), 76–92 (2015)
23. Pontani, M., Cecchetti, G., Teofilatto, P.: Variable-time-domain neighboring optimal guidance, part 2 application to lunar descent and soft landing. J. Optim. Theory Appl. **166**(1), 93–114 (2015)
24. Pontani, M., Cecchetti, G., Teofilatto, P.: Variable-time-domain neighboring optimal guidance applied to space trajectories. Acta Astronaut. **115**, 102–120 (2015)
25. Pontani, M., Conway, B.A.: Optimal low-thrust orbital maneuvers via indirect swarming method. J. Optim. Theory Appl. **162**(1), 272–292 (2014)
26. Pontani, M., Ghosh, P., Conway, B.A.: Particle swarm optimization of multiple-burn rendezvous trajectories. J. Guid. Control Dynam. **35**(4), 1192–1207 (2012)
27. Pontani, M., Teofilatto, P.: Simple method for performance evaluation of multistage rockets. Acta Astronaut. **94**(1), 434–445 (2014)
28. Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., Mishchenko, E.F.: The Mathematical Theory of Optimal Processes. Princeton University Press, Princeton (1962)
29. Qazi, M.D., Linshu, H., Elhabian, T.: Rapid Trajectory Optimization Using Computational Intelligence for Guidance and Conceptual Design of Multistage Launch Vehicles. AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco (2005)
30. Roh, W., Kim, Y.: Trajectory optimization for a multi-stage launch vehicle using time finite element and direct collocation methods. Eng. Optim. **34**(1), 15–32 (2002)
31. Seywald, H., Cliff, E.M.: Neighboring optimal control based feedback law for the advanced launch system. J. Guid. Control Dynam. **17**(3), 1154–1162 (1994)
32. Teofilatto, P., De Pasquale, E.: A non-linear adaptive guidance algorithm for last-stage launcher control. J. Aerosp. Eng. 213, 45–55 (1999)
33. Townsend, G.E., Abbott, A.S., Palmer, R.R.: Guidance, Flight Mechanics and Trajectory Optimization, Volume VIII, Boost Guidance Equations. NASA Contractor Report, Washington (1968)
34. Vought Corporation: Scout User's Manual. National Technical Information Service, Springfield (1980)
35. Weigel, N., Well, K.H.: Dual payload ascent trajectory optimization with a splash-down constraint. J. Guid. Control Dynam. **23**(1), 45–52 (2000)
36. Yan, H., Fahroo, F., Ross, I.: Real-Time Computation of Neighboring Optimal Control Laws. AIAA Guidance, Navigation and Control Conference, Monterey (2002)

Optimization Issues in the Problem of Small Satellite Attitude Determination and Control

Zaure Rakisheva, Anna Sukhenko, and Nazgul Kaliyeva

Abbreviations

λ	Positive constant
$\overrightarrow{\omega_{bi}^{b}}$	Angular velocity of the small satellite in the body coordinate system
$\overrightarrow{\omega_{oi}^{o}}$	Angular velocity of the orbital coordinate system relative to the inertial coordinate system
$\Delta\omega_1, \Delta\omega_2, \Delta\omega_3$	Components of the vector of deviation of the small satellite angular velocity w.r.t. the required angular velocity
$\overrightarrow{\omega_{\mathrm{r}}}$	Required angular velocity
$\otimes$	Multiplication operator for quaternions
A	System matrix
B	Control matrix
$\overrightarrow{B}$	Geomagnetic induction vector
$\overrightarrow{B^{o}} = \left[B_x^o, B_y^o, B_z^o\right]$	Geomagnetic induction vector in the orbital coordinate system
$e(t)$	Misalignment of the angular position of the small satellite
$\overrightarrow{G_b}$	Angular momentum of satellite
$\overrightarrow{h}$	Vector of conjugate variables
$\overrightarrow{h_a^b}$	Angular momentum of the reaction wheels
J	Inertia moments of the small satellite
$\tilde{J}_x, \tilde{J}_{\mathrm{y}}, \tilde{J}_z$	Nominal values of the inertia moments
$\Delta J_x, \Delta J_y, \Delta J_z$	Value of deviations of the satellite moment of inertia

Z. Rakisheva (✉) · A. Sukhenko · N. Kaliyeva
Department of Mechanics, Al-Farabi Kazakh National University, Almaty, Kazakhstan
e-mail: Zaure.Rakisheva@kaznu.kz; suhenko.a@istt.kz; Nazgul.Kaliyeva@kaznu.kz

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_14

K_p, K_d	Proportional gain and derivative gain
LMI	Linear matrix inequalities
$\overrightarrow{M_e^b}$	Moment of the external forces in the body coordinate system
$\overrightarrow{M_c^b}$	Control moment of the actuators in the body coordinate system
$\overrightarrow{m}$	Magnetic moment of the coils
N	Matrix of system noise
P	Initial covariance matrix
PD-controller	Proportional-derivative controller
$\Delta\overrightarrow{\mathrm{Q}}$	Difference between the current angular position and the required angular position
$\overrightarrow{\mathrm{Q}_{bi}}$	Quaternion that sets the current angular position of the satellite in the inertial coordinate system
$\overrightarrow{Q}_{bo}$	Quaternion that sets the current angular position of the satellite in the orbital coordinate system
$\overrightarrow{Q_{bo}}^*$	Quaternion that is the inverse of $\overrightarrow{Q}_{bo}$
$\overrightarrow{\mathrm{Q}_\mathrm{r}}$	Desired angular position
Δq_1, Δq_2, Δq_3	Components of the vector part of the quaternion which describe the deviation of the small satellite current orientation regarding the desired orientation
R_b^o	Direction cosine matrix representing the rotation of the orbital coordinate system axes regarding the axes of the body coordinate system
$\overrightarrow{u}$	Control vector
$\mathrm{W} \geq 0$, $\mathrm{P} > 0$	Constant matrices
$\overrightarrow{X}$	State vector of the dynamical system
$\Delta\overrightarrow{X}$	Deviation of the current state vector of the dynamic system w.r.t. the desired state vector
δ_x, δ_y, δ_z	Normalized parametric uncertainties of the satellite inertia moments

1 Introduction

Currently, small satellites (up to 100 kg) are widely used in the space field giving the opportunity of solving various scientific and technological tasks with the lowest expenses. A small satellite has a relatively low cost and short term of development. Experience of many countries having the program of small satellite development shows that small satellites can fully replace large satellites when solving the task of detailed cartographic surveys of the Earth, can solve the task of fire detection, carry out surveys of disaster zones, and conduct ecological monitoring and weather observations. In addition, this class of satellites is designed to process new technologies and conduct experiments in space.

The main problem of synthesis of attitude determination and control system for small satellites consists in providing the required accuracy of their orientation since a small satellite is more influenced by the external disturbances due to its small mass. In addition, saving energy resources for the use of actuators is a major concern.

Required accuracy of small satellite orientation can be achieved with the help of high-accuracy attitude determination and control algorithms using the minimal set of sensors and actuators. This statement defines one area of study considered in this chapter.

The requirement of saving energy resources in the process of attitude determination and control system development for small satellites can be fulfilled through the use of magnetic actuators generating controlling mechanical torque by interaction with the Earth magnetic field. This statement defines another area of study considered in this chapter.

As a small satellite prototype, this work considers a nanosatellite on the basis of CubeSat3U platform with a mass of 4.2 kg and inertia moments $J = [0.04088; 0.04088; 0.1116]\ kg \cdot m^2$. The relevant orbit is sun-synchronous with a height of 560 km.

Three-axis gyro sensor, three-axis magnetic sensor, two-axis sun sensors, three reaction wheels, and three magnetorquers are the main components of attitude determination and control system of the small satellite considered.

2 Problem Statement

There are many different theories and techniques of attitude determination and control for small satellites that are developed by many authors using various types of sensors and actuators.

An analysis of the topical works shows that the main problem of satellite attitude determination lies in the fact that all sensors on the satellite operate in their specific manner and in some cases generate partial or full inaccessibility to their measurements [1]. Methods for satellite attitude determination such as TRIAD, QUEST [2, 3], and Kalman filter [4] are based on vector measurements, and Kalman filter assumes their tuning to obtain the optimal estimation of the satellite attitude parameters.

A small satellite is more influenced by the external disturbances due to its small mass [1]. The main sources of the disturbances for the attitude of small satellites are the torques of external forces [5], however, the disturbances caused by the failures of sensors and actuators and disturbances caused by the various uncertainties are not less impacting [6]. The classical control theory [7], linear optimal control theory [8], and adaptive control theories [9, 10] are applied to compensate perturbations and provide the necessary attitude control. Recently, the sliding mode control theory [11] and suboptimal H_∞-control theory [12] have become popular for synthesis of robust control dealing with uncertainties.

Hence, the first purpose of this work is the development of algorithms for the precise attitude determination of a small satellite in conditions of inaccessibility of

the full vector of measurement and the development of algorithms of attitude control of a small satellite, taking into account the external disturbances and disturbances caused by uncertainties on the inertia moments.

In this work, the mathematical model of Kalman filter and suboptimal H_∞-control theory are used as the basis for the development of precise small satellite attitude determination algorithms and the relevant robust attitude control.

Nowadays, several schools from many countries carry out their research in the field of the development of satellite attitude control systems, on the basis of magnetorquers that are usually called magnetic attitude control systems and use their results on already launched satellites [13–15]. Magnetorquers are more reliable and have less energy consumption in comparison with mechanical actuators due to the absence of mechanical elements. These can be widely used for the achievement of the required orientation of the satellite.

Classical and modern methods of control system synthesis are used for the development of control laws of change of magnetic moment for magnetorquers [16–18].

From the results of applicability analysis of various control synthesis methods for the magnetic attitude systems, it was determined that when developing linear control laws for magnetorquers the following peculiarities should be considered:

- Standard methods of linear control systems synthesis can be used for control synthesis mainly in the short term, due to the nonlinearity of the mathematical model of the satellite motion controlled by magnetorquers;
- Developing control laws should compensate all impacting external disturbances;
- Developing control laws should provide robustness in the presence of uncertainties provided by change of moments of inertia and the accuracy of the mathematical model of the Earth magnetic field;
- Great difficulty is in the realization of control laws with variable coefficients of feedback on-board the satellite;
- For the production of magnetic torque of magnetorquers, electrical currents only in a definite range can be used; as a consequence, the problems of magnetorquers saturation can arise, especially when working with large deviations from the required orientation.

Synthesis methods of nonlinear control laws in comparison with the linear ones allow to consider a number of the above peculiarities and provide more flexible control systems.

Hence, the second purpose of this work is the development of a magnetic attitude control system of small satellites. This takes into account the complexity of the satellite three-axis orientation, in case of a near-zero angle between the directions of the magnetic induction vector and the magnetic torque vector of the satellite. In addition, the variability of the magnetic field and a residual magnetic torque, significantly impacting the orientation of small satellites, are considered.

In this work, as a result of research of various linear and nonlinear methods for the control system synthesis, two techniques that can be adopted for the development of magnetic attitude systems of small satellites were chosen: synthesis technique of linear control law, on the basis of PD-controller, and synthesis

techniques of nonlinear control law, on the basis of usage of control theory with sliding mode. Modification of these control laws using optimization methods and taking into account peculiarities of the Earth magnetic field allow to achieve a full controllability of the small satellite.

3 Mathematical Model of Rotational Motion of a Satellite

For the description of satellite motion, several coordinate systems are used: fixed inertial coordinate system $O\mathrm{x_i}y_iz_i$ with the origin at the Earth center of mass; body coordinate system $\mathrm{Cx_by_bz_b}$ with the origin at the satellite center of mass and axes coincident with the principal central axes of inertia of satellite; orbital coordinate system $\mathrm{Cx_oy_oz_o}$ with the origin at the satellite center of mass and the direction of $\mathrm{Cx_o}$ axis coincident with the direction of satellite motion, $\mathrm{Cz_o}$ axis directed to the Earth center from the satellite center of mass and $\mathrm{Cy_o}$ axis complementing the system as a right-handed system. The description of orientation of the axes of the body coordinate system related to the axes of other coordinate systems is performed using quaternions.

Gravity-gradient torque and residual magnetic torque are considered as the main external disturbances acting on the satellite. Sun pressure and aerodynamic disturbances are not considered in virtue of the small middle cross-section of the satellite.

Dynamic Euler equations are used as the equations of dynamics of the satellite, and kinematic equations in quaternions are used as the equations of kinematics:

$$\dot{\overrightarrow{\omega}}^{b}_{\mathrm{bi}} = J^{-1}\left[-\overrightarrow{\omega^{b}_{bi}} \times \overrightarrow{G_b} + \overrightarrow{M^{b}_{c}} + \overrightarrow{M^{b}_{e}}\right], \tag{1}$$

where $J = \{J_x, J_y, J_z\}$—diagonal (3x3) matrix of the small satellite inertia tensor; $\overrightarrow{\omega^{b}_{bi}}$—angular velocity of the small satellite in the body coordinate system; $\overrightarrow{G_b}$—angular momentum of the satellite; $\overrightarrow{M^{b}_{e}}$—moment of external forces in the body coordinate system; and $\overrightarrow{M^{b}_{c}}$—control moment of the actuators.

$$\overrightarrow{\omega}^{\mathrm{b}}_{\mathrm{bo}} = 2\overrightarrow{Q_{bo}}^{*} \otimes \dot{\overrightarrow{Q}}_{bo}, \tag{2}$$

where $\overrightarrow{Q}_{bo}$—quaternion that sets the current angular position of the small satellite in the orbital coordinate system; $\overrightarrow{Q_{\mathrm{bo}}}^{*}$—quaternion that is the inverse of $\overrightarrow{Q}_{bo}$, $\overrightarrow{Q_{bo}}^{*} = q_0^{\mathrm{bo}} - \overrightarrow{q^{bo}}$; and $\otimes$—the operation of multiplication of quaternions.

$$\overrightarrow{\omega^{b}_{bi}} = \overrightarrow{\omega^{b}_{bo}} + \overrightarrow{\omega^{b}_{oi}} = \overrightarrow{\omega^{b}_{bo}} + R^{o}_{b}\overrightarrow{\omega^{o}_{oi}}, \tag{3}$$

where R_b^o is the direction cosine matrix representing the rotation of the orbital coordinate system axes with respect to the body coordinate system axes; and $\overrightarrow{\omega_{oi}^o} = [0 - \omega_0\ 0]^T$ is the angular velocity of the orbital coordinate system relative to an inertial coordinate system.

4 Precise Attitude Determination and Perturbations Tolerant Control for Small Satellite

4.1 Optimal Estimation of Satellite Attitude Parameters in Case of Incomplete Vector of Measurements

In this section, we consider the problem of the development of the algorithm for the estimation of the angular position and angular velocity of the small satellite, on the basis of the measurements of the sun and magnetic sensors. The satellite is actuated by the reaction wheels, i.e., the angular momentum has the form:

$$\overrightarrow{G_b} = J\overrightarrow{\omega_{bi}^b} + \overrightarrow{h_a^b}, \tag{4}$$

where $\overrightarrow{h_a^b}$—angular moment of the reaction wheels.

Let us assume that the rotational motion of the small satellite is described by the system of Eqs. (1) and (2), and the control torque of the reaction wheels has the form:

$$\overrightarrow{M_c^b} = -K_\omega \overrightarrow{\omega} - K_Q \overrightarrow{\mathrm{q}} - \omega_o \overrightarrow{a_2} \times \overrightarrow{h}, \tag{5}$$

where:

$$\overrightarrow{\omega} = \overrightarrow{\omega_{\mathrm{bo}}^b},\ \overrightarrow{h} = \overrightarrow{h_a^b},\ \overrightarrow{Q_{bo}} = \overrightarrow{Q},$$

$$\overrightarrow{a_2} = \left[2\,(q_1q_2 + q_0q_3) \quad \mathrm{q}_0^2 - q_1^2 + q_2^2 - q_3^2 \quad 2\,(q_2q_3 - q_0q_1)\right]^{\mathrm{T}}$$

and the value of the coefficients K_q, K_ω is determined using the theory of the optimal synthesis of linear-quadratic regulator.

For an accurate estimation of the angular velocity and the angular position of the small satellite, in the absence of measurements of the angular velocity sensor, an algorithm on the basis of Kalman filter is developed. It operates on the basis of the prediction–correction principle and allows to obtain the optimal estimation of the angular velocity and the angular position of the small satellite.

Predicted values of angular velocity and angular position of the small satellite are determined at the first stage of the algorithm by solving the linearized equations of motion with the initial conditions determined as the output parameters of the Kalman filter, obtained at the previous time step:

$$\dot{\overrightarrow{\mathrm{x}}} = \mathrm{F}\overrightarrow{\delta}, \tag{6}$$

where:

$$\overrightarrow{\mathrm{x}} = \left[\overrightarrow{\widehat{\omega}_{k+1}^{-}}, \overrightarrow{\widehat{Q}_{k+1}^{-}}\right],$$

$$\mathrm{F}=\begin{bmatrix} \left[J\ \overrightarrow{\widehat{\omega}}\times\right]-\left[\overrightarrow{\widehat{\omega}}\times\right]J+\left[\overrightarrow{h}\times\right]-K_{\omega}\ K_q-2K_{\omega}\left[\overrightarrow{\omega_{oi}^{b}}\times\right]+2\left[\overrightarrow{h}\times\right]\left[\overrightarrow{\omega_{oi}^{b}}\times\right] \\ \frac{1}{2}I_{3x3} \qquad\qquad -\left[\overrightarrow{\widehat{\omega}}\times\right] \end{bmatrix}.$$

At the first stage, it is also determined the current deviation of the estimated state vector from its true value which is characterized by a covariance matrix P:

$$\mathrm{P}_{\mathrm{k+1}}^{-} = \Phi_k P_k^{+} \Phi_k^{T} + N, \tag{7}$$

where $\Phi_k = \mathrm{e}^{\mathrm{F}\Delta\mathrm{T}} = 1 + F\Delta T + (F\Delta T)^2/2\,!\,. \ldots\ldots\ldots\ldots, N$—matrix of the system noise.

That is to say that to obtain the predicted values of the attitude parameters of the satellite and covariance matrix, it is required that their corrected values are known at the previous step. This fact is one of the main problems of the implementation of the Kalman filter. The choice of the optimal initial attitude parameters of the small satellite and the covariance matrix can improve the convergence and accuracy of the Kalman filter.

In this work, in order to determine the initial values of the attitude parameters, the algorithm of rough estimate of the small satellite attitude parameters, on the basis of sun sensor measurements, is used. The determination of the optimal initial covariance matrix P and the matrix of the system noise N, that allows to obtain the least time of convergence of the Kalman filter and the higher accuracy in determining the satellite attitude parameters, is conducted as a result of a series of numerical experiments. This allowed to obtain the dependency of Kalman filter convergence and accuracy on the initial values of matrices P, N.

At the second stage of the algorithm, after receiving the measurement vector $\overrightarrow{z_{k+1}}$ from the magnetic and sun sensors, the corrected values of the satellite attitude parameters $\overrightarrow{\widehat{\omega}_{k+1}^{-}}$, $\overrightarrow{\widehat{Q}_{k+1}^{-}}$ and error covariance matrix $\mathrm{P}_{\mathrm{k+1}}^{+}$ are calculated by means of formulas that are known from the Kalman filter algorithm.

By using the proposed method, the initial angular position, angular velocity, and covariance matrix were obtained:

$$\begin{aligned} &\overrightarrow{\mathrm{Q_{bo}\,(t_0)}} = [0.9109;\ 0.1578;\ -0.2678;\ 0.2712]^{\circ}, \\ &\overrightarrow{\omega_{bi}^{b}} = [0.0899;\ 0.01003;\ 0.03005]\ \ \mathrm{rad/\,sec}, \end{aligned} \tag{8}$$

$$P(t_0) = \begin{bmatrix} 0.001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (9)$$

Numerical experiments for deriving the initial satellite attitude parameters are carried out using the initial values (8) and (9). Results of comparison of the estimated attitude parameters (red dotted line) of the satellite with their true values (blue bold line) are shown in Figures 1 and 2.

From Figures 1 and 2, it can be seen that the angular position and the angular velocity of the small satellite, as estimated by means of the proposed algorithm, have a small deviation from their true values.

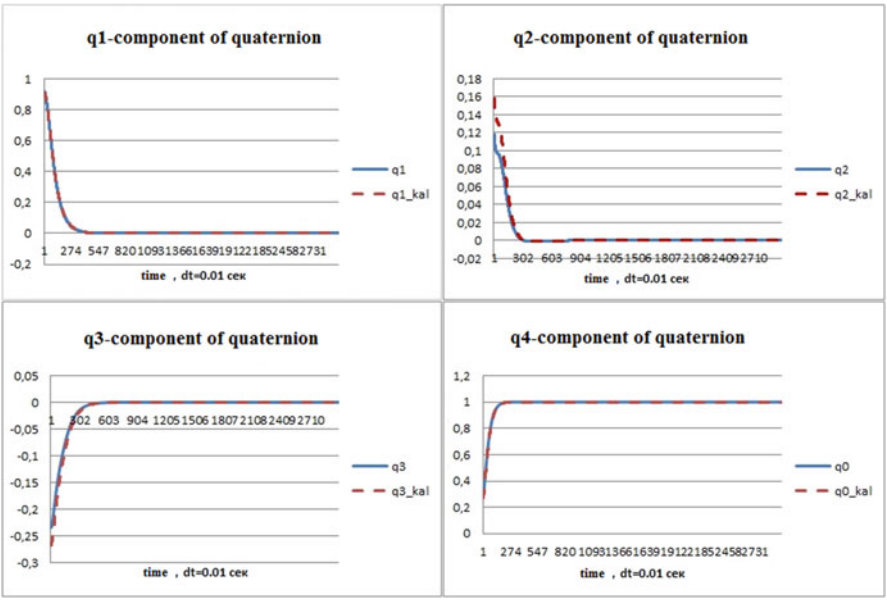


Figure 1 The results of the comparison of the estimated angular position of the satellite with its true value

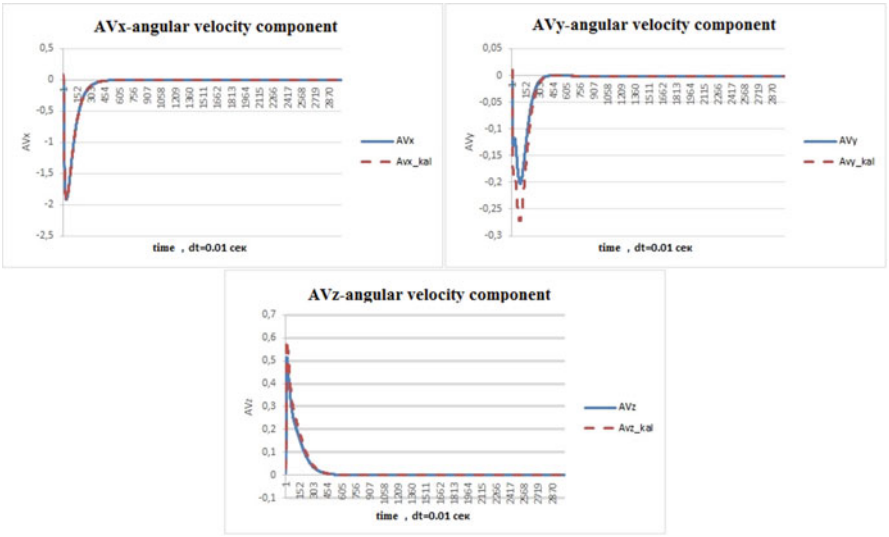


Figure 2 The results of the comparison of the estimated angular velocity of the satellite with its true value

4.2 Robust Satellite Attitude Control Tolerant to External Perturbations and Inertia Moment Uncertainties

This section illustrates the methodology of synthesis of satellite robust attitude control, tolerant to external disturbances and inertia moment uncertainties, using the theory of suboptimal linear H_∞—control.

In this case, the linear system takes the form of equations:

$$\begin{aligned} \dot{\vec{x}} &= A\vec{x} + B_1\vec{w} + B_2\vec{u}, \\ \vec{z} &= C_1\vec{x} + D_{11}\vec{w} + D_{12}\vec{u}, \\ \vec{y} &= C_2\vec{x} + D_{21}\vec{w}. \end{aligned} \tag{10}$$

The suboptimal H_∞-control problem of parameter γ consists in finding a controller $K_\infty(s)$ such that the H_∞-norm of the transfer function $\|T_{wz}\|_\infty$ from $\vec{w}$ к $\vec{z}$ is strictly less then γ [19]:

$$\|T_{wz}\|_\infty < \gamma, \tag{11}$$

where $\gamma > 1$, $\gamma = \text{const}$.

Solutions of this problem will be called γ-suboptimal controllers.

According to the theory of synthesis of H_∞-control for the system (10) with the conditions (11), the H_∞-controller can be determined in the form:

$$\mathrm{K}_\infty(s) = D_k + C_k(sI - A_k)^{-1}B_k, \tag{12}$$

if the following matrix inequality is satisfied for some $X_\infty > 0$:

$$\begin{bmatrix} A_{cl}^T X_\infty + X_\infty A_{cl} & X_\infty B_{cl} & C_{cl}^T \\ B_{cl}^T X_\infty & -\gamma I & D_{cl}^T \\ C_{cl} & D_{cl} & -\gamma I \end{bmatrix} < 0, \tag{13}$$

where:

$$\begin{aligned} A_{cl} &= \begin{pmatrix} A + B_2 D_k C_2 & B_2 C_k \\ B_k C_2 & A_k \end{pmatrix}, B_{cl} = \begin{pmatrix} B_1 + B_2 D_k D_{21} \\ B_k D_{21} \end{pmatrix}, \\ C_{cl} &= (C_1 + D_{12} D_k C_2, D_{12} C_k), D_{cl} = D_{11} + D_{12} D_k D_{21}. \end{aligned} \tag{14}$$

Thus, the control synthesis problem is reduced to the solution of inequality (13) that can be solved using an efficient convex optimization algorithm such as [20].

Taking into account the gravitational and residual magnetic torque as disturbances and the reaction wheel torque as control action, the linearized Eqs. (1) and (2) can be written as:

$$F\ddot{\vec{p}} + H\dot{\vec{p}} + Q\vec{p} = G_d\vec{w} + G_u\vec{u}, \tag{15}$$

where $p = [q_1, q_2, q_3]$, $\vec{u} = \left[-J_{rx}\dot{\omega}_{rx}, -J_{ry}\dot{\omega}_{ry}, -J_{rz}\dot{\omega}_{rz}\right]$—vector of the controlling torques of the reaction wheels; and $\vec{w} = \left[m_x, m_y, m_z\right]$—vector of the external disturbances.

$$F = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix}, \; H = \begin{bmatrix} 0 & 0 & -\omega_0\left(J_x + J_z - J_y\right) \\ 0 & 0 & 0 \\ \omega_0\left(J_x + J_z - J_y\right) & 0 & 0 \end{bmatrix},$$

$$Q = \begin{bmatrix} 4\omega_0^2\left(J_y - J_z\right) & 0 & 0 \\ 0 & 3\omega_0^2\left(J_z - J_x\right) & 0 \\ 0 & 0 & 3\omega_0^2\left(J_y - J_x\right) \end{bmatrix}, G_u = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}, \tag{16}$$

$$G_d = \begin{bmatrix} 0 & \frac{1}{2}B_z^o & -\frac{1}{2}B_y^o \\ -\frac{1}{2}B_z^o & 0 & \frac{1}{2}B_x^o \\ \frac{1}{2}B_y^o & -\frac{1}{2}B_x^o & 0 \end{bmatrix},$$

where $\vec{B^o} = \left[B_x^o, B_y^o, B_z^o\right]$—the magnetic induction vector of the magnetic field of the Earth in the orbital coordinate system.

To simulate the uncertainties of the satellite moments of inertia, we assume that the finite value of the satellite moments of inertia can be considered as the sum of their nominal values $\tilde{J}_x$, $\tilde{J}_y$, $\tilde{J}_z$ and the disturbances:

$$J_x = \tilde{J}_x + \Delta J_x \delta_x,\ J_y = \tilde{J}_y + \Delta J_y \delta_y,\ J_z = \tilde{J}_z + \Delta J_z \delta_z, \tag{17}$$

where ΔJ_x, ΔJ_y, ΔJ_z—the value of the deviations of the satellite moments of inertia; and δ_x, δ_y, δ_z—normalized parametric uncertainties of the satellite moments of inertia, $\delta_x, \delta_y, \delta_z \leq 1$.

Then, the right side of (15) includes the additional term representing the disturbances due to the uncertainty of the moments of inertia of the satellite:

$$F\ddot{\vec{p}} + H\dot{\vec{p}} + Q\vec{p} = G_d\vec{w} + \tilde{G}_d\vec{w} + G_u\vec{u}, \tag{18}$$

where $\mathrm{F} = F_0 + L_M \Delta_M P_M$, $H = H_0 + L_D \Delta_D P_D$, $Q = Q_0 + L_K \Delta_K P_K$, $\tilde{G}_d = [L_M\ L_D\ L_K]$, $\vec{\tilde{z}} = [P_M \ddot{p}, P_D \dot{p}, P_K p]$, and $\vec{\tilde{w}} = -\vec{\tilde{z}}$.

Reducing the system (18) to the form (10) provided the control synthesis by the proposed method.

The results of the simulation of the satellite angular motion under the action of H_∞-control, taking into account the external disturbances and the uncertainties of the moments of inertia, are shown in Figure 3. To evaluate the effectiveness of the developed robust control, a comparison of the results of its work with the results obtained for the linear-quadratic regulator (H_2-control) is provided. For the numerical calculations, it is accepted that the value of the variations of the moments of inertia from their nominal values is $\Delta J = [\text{-}10\ \%\ \mathrm{J_x};\ 10\ \%\ \mathrm{J_y};\ \text{-}10\ \%\ \mathrm{J_z}]$.

Figure 3 shows the graphs of changes of some components of the quaternion, with respect to the time, under the action of H_2-control and robust H_∞-control, taking into account the external gravitational and magnetic disturbances, as well as the disturbances caused by the inertia moment uncertainties of the satellite (red line) and the graphs of the changes of some components of the quaternion, with respect to the time, under the action of H_2-control and robust H_∞-control, neglecting the disturbances caused by the inertia moment uncertainties of the satellite (blue line). As it can be seen from Figure 3, the inertia moment uncertainties almost do not affect the efficiency of the of H_∞-control, in comparison with H_2-control.

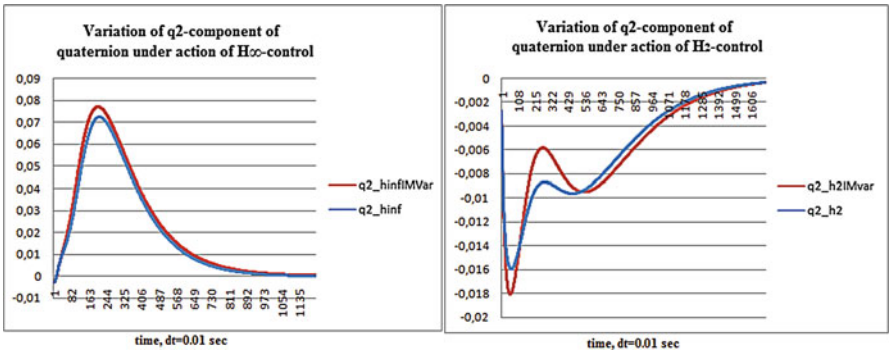


Figure 3 Change of the component q_2 of the quaternion when using the H_2-и H_∞-control

5 Magnetic Attitude Control System for a Small Satellite

5.1 Magnetic Linear Attitude Control

Magnetorquers are widely used for the attitude control of satellites and their use is especially beneficial when there are strict constraints on mass, cost, and energy consumption. However, the accuracy of orientation and maneuverability of satellites with magnetic attitude systems is relatively smaller than control systems with mechanical actuators. Control quality in this case can be improved enhancing the mathematical apparatus of the satellite control.

Magnetic actuators generate control mechanical torque through interaction between its own magnetic moment $\overrightarrow{m} = \left[m_x, m_y, m_z\right]$ and the Earth magnetic field [1]:

$$\overrightarrow{M}^{b}_{c} = \overrightarrow{m} \times \overrightarrow{B}, \tag{19}$$

where $\overrightarrow{B}$ is the geomagnetic induction vector.

As it can be seen from formula (19), the required satellite attitude can be achieved by regulating $\overrightarrow{m}$.

In this work, the PD-controller is selected as the basis for development of the linear magnetic controller in virtue of its simplicity, reliability, and ability to optimization.

In accordance with the principle of constructing a PD-controller, we will consider the function of the magnetic moment as [21]:

$$\overrightarrow{u} = \left[m_x, m_y, m_z\right], \tag{20}$$

$$\overrightarrow{u} = K_p \overrightarrow{e} + K_d \frac{d\overrightarrow{e}}{dt} \tag{21}$$

where K_p, K_d are unknown proportional gain and derivative gain, respectively, and $e(t)$ is a misalignment of the angular position of the small satellite.

The angular position of the small satellite is given in the form of a quaternion $\overrightarrow{Q_{bi}}$, then the misalignment with respect to the angular position of the small satellite is given in the form $\Delta\overrightarrow{Q} = \left[q_0, \Delta\overrightarrow{q}\right]$ as the difference between the current angular position and the required one. The control signal will take the form:

$$\overrightarrow{u} = -\begin{pmatrix} K_p^1 & 0 & 0 & K_d^1 & 0 & 0 \\ 0 & K_p^{21} & 0 & 0 & K_d^2 & 0 \\ 0 & 0 & K_p^3 & 0 & 0 & K_d^3 \end{pmatrix} \begin{bmatrix} \Delta q_1 \\ \Delta q_2 \\ \Delta q_3 \\ \Delta \omega_1 \\ \Delta \omega_2 \\ \Delta \omega_3 \end{bmatrix} = \begin{bmatrix} -K_p^1 \Delta q_1 - K_d^1 \Delta \omega_1 \\ -K_p^2 \Delta q_2 - K_d^2 \Delta \omega_2 \\ -K_p^3 \Delta q_3 - K_d^3 \Delta \omega_3 \end{bmatrix} \quad (22)$$

where Δq_1, Δq_2, Δq_3 are the components of the quaternion vector part which describe the deviation of the small satellite current orientation with respect to the desired orientation; and $\Delta \omega_1$, $\Delta \omega_2$, $\Delta \omega_3$ are the components of the vector of the deviation of the current angular velocity of the small satellite with respect to the required angular velocity.

It can be seen from formula (21) that, to include the PD-controller in the control system, it is necessary to determine the unknown coefficients K_p, K_d [22]. In this section, in order to determine the coefficients, it is proposed to use two approaches: an optimal approach based on a quadratic quality criterion and an approach based on the optimal arrangement of the roots of the characteristic equation of a closed control system.

The optimal approach based on a quadratic quality criterion implies the determination of a control that minimizes the functional:

$$J\left(\overrightarrow{u}\right) = \frac{1}{2}\int_0^\infty \left[\overrightarrow{\Delta X}^T W \overrightarrow{\Delta X} + \overrightarrow{u}^T P \overrightarrow{u}\right] dt \quad (23)$$

where $\overrightarrow{\Delta X}$—deviation of current state vector of the dynamic system with respect to the desired state vector, $\overrightarrow{\Delta X} = \overrightarrow{X} - \overrightarrow{X_m}$; $\overrightarrow{u}$—control vector; and W $\geq$ 0, P > 0—constant matrices.

Optimal problems based on the quadratic quality criterion are solved for linear systems of the form:

$$\frac{d\overrightarrow{X}}{dt} = A\overrightarrow{X} + B\overrightarrow{u}, \quad (24)$$

where $\overrightarrow{X}$—state vector of dynamical system; A—system matrix; and B—control matrix.

The Hamiltonian for this system is given by:

$$\mathrm{H} = \frac{1}{2}\left(\overrightarrow{\Delta X}^{\mathrm{T}} W \overrightarrow{\Delta X} + \overrightarrow{u}^T P \overrightarrow{u}\right) + \overrightarrow{h}^T \left(A\overrightarrow{X} + B\overrightarrow{u}\right), \quad (25)$$

where $\overrightarrow{h}$—vector of conjugate variables.

The control $\overrightarrow{u}$ is determined through the necessary conditions of optimality:

$$\frac{\partial \mathrm{H}}{\partial \mathrm{u}} = P\overrightarrow{u} + B^T\overrightarrow{h} = 0, \tag{26}$$

$$\overrightarrow{u} = -P^{-1}B^T\overrightarrow{h}. \tag{27}$$

We define $\overrightarrow{h}$ in (26) as a linear combination:

$$\overrightarrow{h} = \mathrm{R}\overrightarrow{X} + \overrightarrow{h_1}. \tag{28}$$

Differentiating (28), we obtain

$$\dot{\overrightarrow{h}} = \dot{R}\overrightarrow{X} + R\dot{\overrightarrow{X}} + \dot{\overrightarrow{h}}_1. \tag{29}$$

Differential equations for conjugate variables are defined as:

$$\dot{\overrightarrow{h}} = -\frac{\partial H}{\partial \overrightarrow{X}} = -W\overrightarrow{\Delta X} - A^T\overrightarrow{h} = W\overrightarrow{X_m} - W\overrightarrow{X} - A^T R\overrightarrow{X} - A^T\overrightarrow{h_1}. \tag{30}$$

Substituting (30) into (29), we obtain

$$\dot{R}\overrightarrow{X} + R\dot{\overrightarrow{X}} + \dot{\overrightarrow{h}}_1 - W\overrightarrow{X_m} + W\overrightarrow{X} + A^T R\overrightarrow{X} + A^T\overrightarrow{h_1} = 0. \tag{31}$$

Substituting (24), (27) into (31), we obtain

$$\left(\dot{\mathrm{R}} + \mathrm{RA} + A^T R - RBP^{-1}B^T R + W\right)\overrightarrow{X} = 0. \tag{32}$$

and

$$\dot{\overrightarrow{h_1}} + \left(A - BP^{-1}B^T R\right)\overrightarrow{h_1} - W\overrightarrow{X_m} = 0. \tag{33}$$

The Eq. (32) has a solution when:

$$\dot{\mathrm{R}} + \mathrm{RA} + A^T R - RBP^{-1}B^T R + W = 0. \tag{34}$$

Thus, to determine the control in the form (27):

$$\begin{aligned} &\overrightarrow{u} = -\mathrm{K}\overrightarrow{\mathrm{X}} - P^{-1}B^T\overrightarrow{h_1}, \\ &K = P^{-1}B^T\mathrm{R}. \end{aligned} \tag{35}$$

It is necessary to solve the differential Eqs. (33) and (34) under the boundary conditions $\overrightarrow{h_1}\,(\mathrm{t_k}) = 0$, $R\,(t_k) = 0$.

An approach based on the optimal arrangement of the roots of the characteristic equation of a closed control system, as the previous approach, is used for linear systems of the form (24). The task of this approach is to determine the parameter matrix K of the control law of form:

$$\overrightarrow{u} = K\overrightarrow{\mathrm{X}}, \tag{36}$$

such that the roots of the characteristic equation of a closed control system are located at given points or regions of the complex plane and provide the required values of such characteristics of the closed system as the time and speed of transient processes.

In this section, we consider the definition of the parameter matrix of the control law that ensures the placement of the characteristic equation roots of a closed system in regions characterized by a system of linear matrix inequalities, called LMI-regions.

The LMI-region is a subset of the complex plane, which is representable by a linear matrix inequality with respect to the variables x = Re (z), y = Im (z) [21]:

$$f(z) < 0. \tag{37}$$

In accordance with the function $f(z)$ in inequality (37), we can put the block matrix [7]:

$$M\,(A, X) = \alpha \mathrm{Y} + \beta \otimes AY + \beta^T \otimes (AY)^T, \tag{38}$$

where ⊗—Kronecker product.

Then, the inequality (38) can be rewritten in the form:

$$M\,(A, Y) < 0, Y > 0. \tag{39}$$

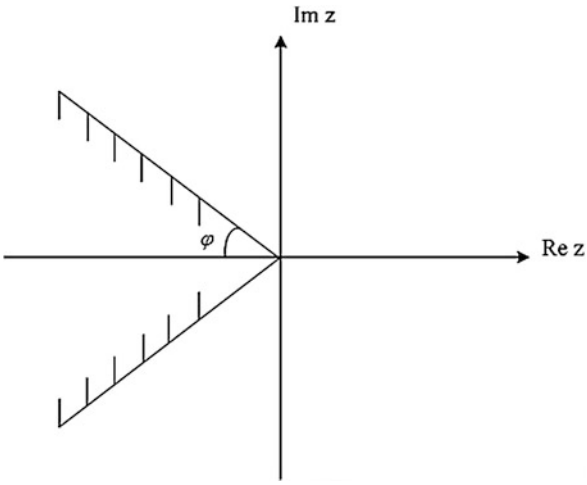
As an example, we give the elementary LMI-region [7], shown in Figure 4, for which inequalities (37) and (39) are written in the form:

$$f_2(z) = \begin{pmatrix} (z+\overline{z})\sin\varphi & (z-\overline{z})\cos\varphi \\ -(z-\overline{z})\cos\varphi & (z+\overline{z})\sin\varphi \end{pmatrix}, \tag{40}$$

$$\begin{pmatrix} \left(AY + YA^T\right)\sin\varphi & \left(AY - YA^T\right)\cos\varphi \\ -\left(AY - YA^T\right)\cos\varphi & \left(AY + YA^T\right)\sin\varphi \end{pmatrix} < 0, \quad Y > 0. \tag{41}$$

Thus, the problem of determining a control at the form (36) in this approach is reduced to finding matrices $Y = Y^T > 0$ and K satisfying the inequality [7]:

Figure 4 LMI-region



$$M\,(A + BK, Y) < 0. \tag{42}$$

Denoting $Z = KY$, the last inequality can be written as a linear matrix inequality with respect to the unknown matrices Y and Z:

$$M\,(A, Y) + \beta \otimes BZ + \beta^T \otimes Z^T B^T < 0. \tag{43}$$

After these matrices are found, the desired parameter matrix of the control law is defined as $K = ZY^{-1}$ [7].

Results of the numerical modeling of the orbital orientation of the small satellite using a PD-controller with tuning coefficients based on the two proposed methods are shown in Figures 5 and 6.

Results of the numerical simulation showed that use of the PD-controller with the adjustment of the coefficients on the basis of the optimal approach and the optimal arrangement of the characteristic equation roots of a closed control system for obtaining the law of the magnetic moment change makes possible to achieve an almost identical accuracy of the satellite orientation (up to 0.15°) in the orbital orientation mode.

5.2 Magnetic Nonlinear Attitude Control

Another important approach in controlling the orientation of a satellite with a magnetic orientation system is based on nonlinear methods. We will consider a strategy for controlling the orientation of a satellite on the basis of the theory of sliding mode control [23].

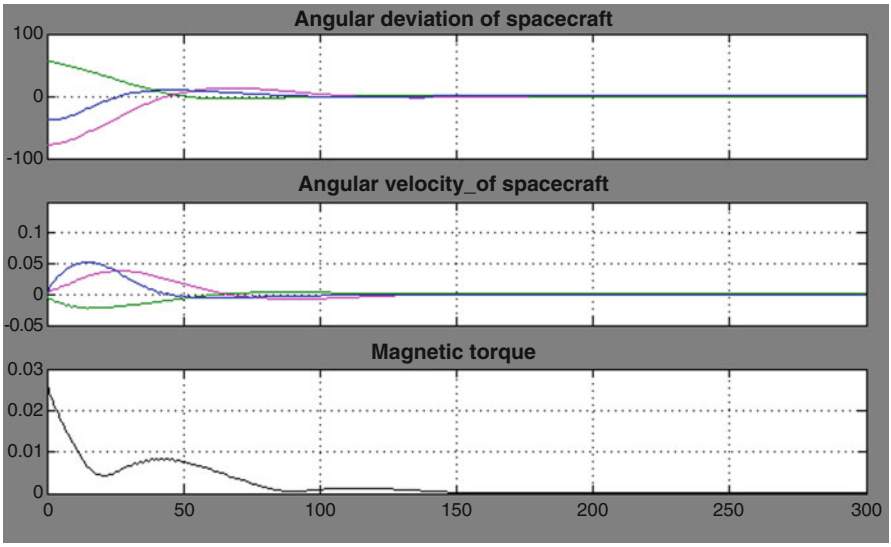


Figure 5 Results of the modeling of the orbital orientation of the small satellite using a PD-controller for determining the magnetic moment when adjusting its coefficients by means of the optimal linear regulator

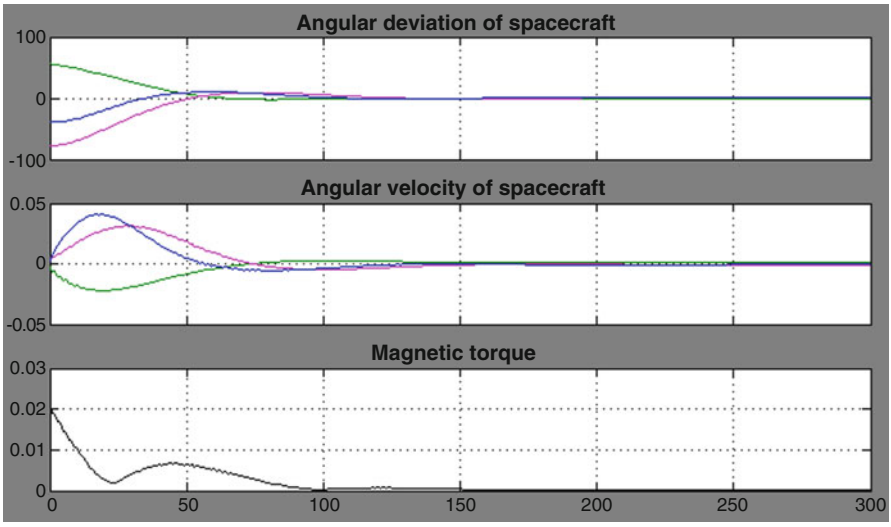


Figure 6 Results of the modeling of the orbital orientation of the small satellite using a PD-controller for determining the magnetic moment when adjusting its coefficients by means of the optimal location of the roots of the characteristic equation

We consider the problem of turning the small satellite from an arbitrary angular position to the required [23]. Let us assume that the desired angular position $\overrightarrow{Q_r}$, and

required angular velocities $\overrightarrow{\omega_r}$ and $\overrightarrow{\dot{\omega}_r}$ are given. Then, the dynamics and kinematics equations of the small satellite with respect to the deviation of the current orientation of the small satellite from the required orientation are given in the transformed form of (1) and (2):

$$J\overrightarrow{\dot{\omega}_{bi}^{b}} + \overrightarrow{\omega_{bi}^{b}} \times J\overrightarrow{\omega_{bi}^{b}} = \overrightarrow{M_c^b} + \overrightarrow{M_e^b}, \tag{44}$$

$$\overrightarrow{\Delta\dot{Q}} = \frac{1}{2}\Omega\left(\overrightarrow{\Delta Q}\right)\overrightarrow{\Delta\omega}, \tag{45}$$

where $\overrightarrow{\Delta Q}$ is a quaternion characterizing the deviation of the current angular position of the small satellite regarding the required angular position, $\overrightarrow{\Delta\omega}$ is a deviation of the angular velocity of the small satellite regarding the required angular velocity, $\overrightarrow{\Delta\omega} = \overrightarrow{\omega_{bi}^{b}} - \overrightarrow{\omega_r}$.

$$\Omega\left(\overrightarrow{\Delta Q}\right) = \begin{bmatrix} \Delta q_0 E + [\Delta q\times] \\ -\Delta q^{\mathrm{T}} \end{bmatrix}, \tag{46}$$

$$[\Delta q\times] = \begin{bmatrix} 0 & -\Delta q_3 & \Delta q_2 \\ \Delta q_3 & 0 & -\Delta q_1 \\ -\Delta q_2 & \Delta q_1 & 0 \end{bmatrix}. \tag{47}$$

The synthesis of the control algorithm based on the control theory with sliding mode consists of two steps [23]:

1. The construction of a surface $\overrightarrow{S}\left(\overrightarrow{\Delta\omega}, \overrightarrow{\Delta Q}, t\right) = 0$ along which a point should move characterizing the state of the control system. If this point is on the surface $\overrightarrow{S}\left(\overrightarrow{\Delta\omega}, \overrightarrow{\Delta Q}, t\right) = 0$ when the small satellite moves, the motion of the small satellite is considered asymptotically stable.
2. The construction of a control providing motion along a given surface.

Usually, the surface is given in the form [23]:

$$\overrightarrow{S} = \overrightarrow{\Delta\omega} + K_q\overrightarrow{\Delta q}, \tag{48}$$

where K_q is a constant, $K_q > 0$.

It must be taken into account that two types of control actions should be obtained when developing a control based on the control system with a sliding mode. The first control action $\overrightarrow{u_{eq}}$ should provide movement on a given surface, as soon as the point characterizing the state of the control system reaches the specified surface. The second control action $\overrightarrow{u_k}$ should provide the point characterizing the state of the control system attains a given surface:

$$\vec{\mathrm{u}} = \overrightarrow{u'_{eq}} + \vec{\mathrm{u}}_{\mathrm{k}}. \tag{49}$$

The motion of the system along the given surface is characterized by the following equations:

$$\vec{\mathrm{S}} = 0, \ \dot{\vec{S}} = 0. \tag{50}$$

and

$$\vec{\mathrm{u}} = \overrightarrow{u'_{eq}}. \tag{51}$$

Taking into account the equations of the small satellite motion, the second equation of (50) takes the form:

$$\begin{aligned} \dot{\vec{\mathrm{S}}} &= \dot{\overrightarrow{\Delta\omega}} + \mathrm{K_q}\dot{\overrightarrow{\Delta q}} = \dot{\overrightarrow{\omega^b_{bi}}} - \dot{\vec{\omega}}_{\mathrm{T}} + \tfrac{1}{2}\mathrm{K_q}\left(\Delta q_0 \mathrm{E} + \left[\overrightarrow{\Delta q}\times\right]\right)\overrightarrow{\Delta\omega} = \\ &= -\mathrm{I}^{-1}\left[\overrightarrow{\omega^b_{bi}}\times\right] I\overrightarrow{\omega^b_{bi}} + I^{-1}\overrightarrow{u'_{eq}} + I^{-1}\overrightarrow{M_{dis}} - \dot{\vec{\omega}}_{\mathrm{T}} \\ &+ \tfrac{1}{2}\mathrm{K_q}\left(\Delta q_0 \mathrm{E} + \left[\overrightarrow{\Delta q}\times\right]\right)\left(\overrightarrow{\omega^b_{bi}} - \vec{\omega}_{\mathrm{T}}\right) = 0. \end{aligned} \tag{52}$$

Hence, $\overrightarrow{u'_{eq}}$ can be obtained directly from (51):

$$\overrightarrow{u'_{eq}} = \left[\overrightarrow{\omega^b_{bi}}\times\right] I\overrightarrow{\omega^b_{bi}} - \overrightarrow{M_{dis}} + \mathrm{I}\dot{\vec{\omega}}_{\mathrm{T}} - \frac{1}{2}\mathrm{K_q} I\left(\Delta q_0 \mathrm{E} + \left[\overrightarrow{\Delta q}\times\right]\right)\left(\overrightarrow{\omega^b_{bi}} - \vec{\omega}_{\mathrm{T}}\right). \tag{53}$$

Until the point characterizing the state of the control system reaches the surface:

$$\vec{\mathrm{S}} \neq 0, \ \dot{\vec{S}} \neq 0 \tag{54}$$

and

$$\vec{\mathrm{u}} = \overrightarrow{u'_{eq}} + \vec{\mathrm{u}}_{\mathrm{k}}.$$

As $\vec{\mathrm{u}}_{\mathrm{k}}$ we use the proportional control law:

$$\vec{\mathrm{u}}_{\mathrm{k}} = -\lambda\,\vec{\mathrm{S}}, \tag{55}$$

where λ is a constant, $\lambda > 0$.

Results of the numerical simulations of the orbital orientation of the small satellite using a sliding control for specifying the law of the change in the magnetic moment of the small satellite are shown in Figure 7.

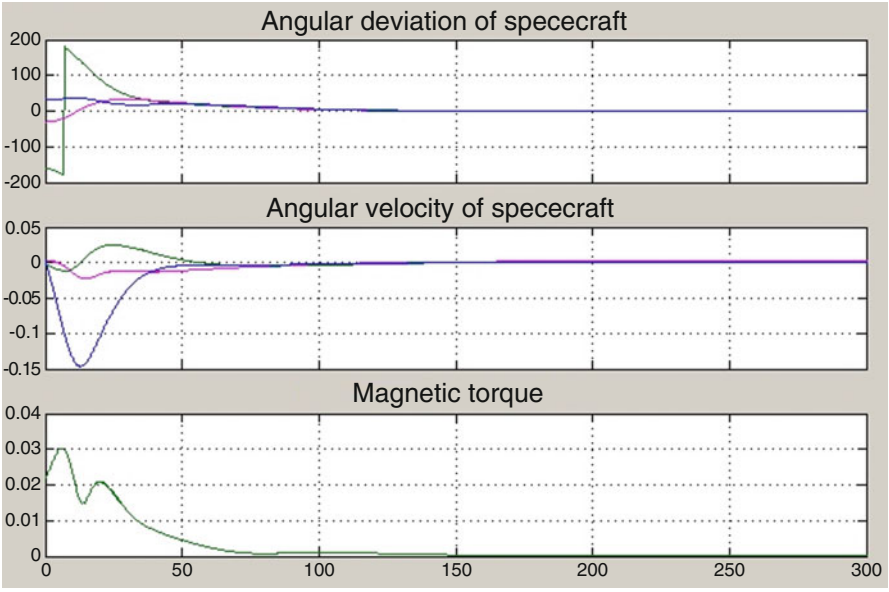


Figure 7 Results of the simulation of the orbital orientation of the small satellite using a sliding control for determining the magnetic moment

Results of the numerical simulation showed that use of the developed sliding mode controller makes it possible to achieve accuracy of the orientation of the satellite up to 0.4° in the orbital orientation mode.

Carrying out optimal tuning of the coefficients suitable for determining the equation of the sliding surface and the proportional control law, a better performance of the developed nonlinear controller can be achieved.

6 Conclusion

In this chapter, the following main results were obtained:

1. An algorithm for optimal estimation of satellite attitude parameters on the basis of Kalman filter that uses the measurements of the angular position sensors (sun sensors and magnetic sensor) is developed.
2. A suboptimal linear H_∞-control for the maintenance of small satellite orbital orientation, under the influence of external gravitational, magnetic disturbances and disturbances caused by the uncertainties of satellite inertia moments and actuator faults, is developed.
3. A magnetic linear satellite attitude control is developed by the optimization of feedback coefficients.
4. A magnetic nonlinear satellite attitude control on the basis of the sliding mode theory is developed.

References

1. Wertz, J.R.: Spacecraft Attitude Determination and Control. Kluwer, Dordrecht (1990)
2. Markley, F.L.: Attitude determination using two vector measurements. http://www.researchgate.net/publication/4706531
3. Shuster, M.D., Oh, S.D.: Three axis attitude determination from vector observations. J. Guid. Contr. Dyn. **4**(1), 70–77 (1981)
4. Marques, S.: Small Satellites Attitude Determination Methods. Thesis, Universidade tecnica de Lisboa (2000)
5. Wertz, J.R., Larson, W.J.: Space Mission Analysis and Design. Microcosm Inc., Torrance, CA (1992)
6. Zhang, A., Wang, Y., Zhang, Z., Reza Karimi, H.: Robust control allocation for spacecraft attitude stabilization under actuator faults and uncertainty. Math. Probl. Eng. **2014**, 789327 (2014). https://doi.org/10.1155/2014/789327
7. Balandin, D.V., Gorodetskiy, S.Y.: Klassicheskie i sovremennyie metodyi postroeniya regulyatorov v primerah. Elektronnoe uchebno-metodicheskoe posobie. Nizhniy Novgorod. Nizhegorodskiy gosuniversitet (2012)
8. Tisa, P., Vergez, P.: Performance analysis of control algorithms for FalconSat-3. http://www.learningace.com/doc/1823864/31/performance-analysis-of-contol-algorithms-for-falconsat-3-vergez
9. Parlos, A.G., Sunkel, J.W.: Adaptive attitude control and momentum management for large—angle spacecraft maneuvers. J. Guid. Control. Dyn. **15**(4), 1018–1028 (1992)
10. Xiao, B., Hu, Q., Ma, G.: Adaptive quaternion-based output feedback control for flexible spacecraft attitude tracking with input constraints. Proc. Inst. Mech, Eng. Part I J. Syst. Cont. Eng. **223**(3), 226–240 (2011)
11. Crassidis, J.L., Markley, F.L.: Sliding mode control using modified Rodriguez parameters. J. Guid. Control. Dyn. **19**(6), 1381–1383 (1996)
12. Doyle, J.C., Glover, K., Khargonekar, P., Francis, B.: State apace solutions to standard H_2 and H infinity control problems. IEEE Trans. Automat. Contr. **34**(8), 831–847 (1989)
13. Wisniewski, R.: Satellite attitude control using only electromagnetic actuation. Ph.D. thesis, Aalborg University, Denmark (1996)
14. Bushenkov, V.A., Ovchinnikov, M.Y., Smirnov, G.V.: Attitude stabilization of a satellite by magnetic coils. Acta Astronaut. **50**(12), 721–728 (2002)
15. Kupriyanova, N.V., Ovchinnikov, M.Y., Pen'kov, V.I., Selivanova, A.S.: Passive magnetic attitude control system of the first Russian nanosatellite TNS-0. KIAM Preprint 46 (2005)
16. Silani, E., Lovera, M.: Magnetic spacecraft attitude control: a survey and some new results. Control. Eng. Pract. **13**(3), 357–371 (2005)
17. Wisniewski, R., Blanke, M.: Fully magnetic attitude control for spacecraft subject to gravity gradient. Automatica. **35**(7), 1201–1214 (1999)
18. Lovera, M.: Optimal magnetic momentum control for inertially pointing spacecraft. Eur. J. Control. **7**(1), 30–39 (2001)
19. Gahinet, P., Apkarian, P.: A linear matrix inequality approach to H∞ control. Int. J. Robust Nonlinear Contr. **4**, 421–448 (1994)
20. Nemirovski, A., Gahinet, P.: The projective method for solving linear matrix inequalities. Proceedings of the American Control Conference. Baltimore, Maryland (1994)
21. Balandin, D.V., Kogan, M.M.: Primenenie lineinyh matrichnyh neravenstv v sinteze zakonov upravleniya: uchebno-metodicheskie materialy po spetsial'nomu kursu "Upravlenie kolebaniyami dinamicheskih sistem". Nizhniy Novgorod (2010)
22. Denisenko, V.V.: PID-regulyatiry: printsipy postroeniya i modificatsii. Sovrem. Tehnologii avtomatizatsii. **1**, 66–73 (2006)
23. Utkin, V.I.: Sliding Modes in Control and Optimization. Springer, Berlin (1992)

Optimized Packings in Space Engineering Applications: Part I

Yuriy Stoyan, Alexandr Pankratov, Tatiana Romanova, Giorgio Fasano, János D. Pintér, Yurij E. Stoian, and Andrey Chugay

Abstract Packing optimization problems have a wide spectrum of real-word applications, including transportation, logistics, chemical/civil/mechanical/power/aerospace engineering, shipbuilding, robotics, additive manufacturing, materials science, mineralogy, molecular geometry, nanotechnology, electronic design automation, very large system integration, pattern recognition, biology, and medicine. In space engineering, ever more challenging packing optimization problems have to be solved, requiring dedicated cutting-edge approaches.

Two chapters in this volume investigate very demanding packing issues that require advanced solutions. The present chapter provides a bird's eye view of challenging packing problems in the space engineering framework, offering some insight on possible approaches. The specific issue of packing a given collection of arbitrary polyhedra, with continuous rotations and minimum item-to-item admissible distance, into a convex container of minimum size, is subsequently analyzed in depth, discussing an ad hoc mathematical model and a dedicated solution algorithm. Computational results show the efficiency of the approach proposed. The following (second) chapter examines a class of packing optimization problems in space with consideration to balancing conditions.

Y. Stoyan · A. Pankratov · T. Romanova · Y. E. Stoian · A. Chugay
Department of Mathematical Modeling and Optimal Design, Institute for Mechanical Engineering Problems of the National Academy of Sciences of Ukraine, Kharkov, Ukraine

G. Fasano (✉)
Thales Alenia Space, Turin, Italy
e-mail: giorgio.fasano@thalesaleniaspace.com

J. D. Pintér
Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA
e-mail: jdp416@lehigh.edu

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_15

1 Introduction

The overall task of packing objects, exploiting the available volume, as much as possible, has represented for centuries a tough challenge. Hilbert's eighteenth problem, consisting in the accommodation of equal spheres, attaining the maximum density, is one of the well-known examples on this subject. In addition to a number of theoretical questions that stimulate relevant mathematical speculation, the compelling need to solve ever more demanding real-world problems has induced specialists in the operations research and computational geometry fields to tackle this topic with a more practical slant, see, e.g., Fasano and Pintér [1].

Packing optimization problems arise in a large number of applications, relevant to transportation, logistics, civil/mechanical/power/aerospace engineering, shipbuilding, robotics, additive manufacturing, materials science, mineralogy, chemical industry, molecular geometry, nanotechnology, electronic design automation, very large system integration, pattern recognition, modern biology, medicine, and other areas.

When dealing with these real-world contexts, an overall numerical approach to obtain high-quality (albeit usually non-proven-optimal) solutions to intricate packing problems is widely adopted. From this standpoint, a vast literature on multi-dimensional packing is available, cf. e.g., Cagan et al. [2], Dyckhoff et al. [3], and Ibaraki et al. [4].

Traditionally, a significant part of the topical research has focused on the orthogonal placement of rectangles/parallelepipeds into rectangles/parallelepipeds, e.g., Faroe et al. [5], Fekete and Schepers [6], Fekete et al. [7], Martello et al. [8, 9], and Pisinger [10]. Several recent works also consider different typologies of packing issues, e.g., Addis et al. [11], Birgin et al. [12], Egeblad et al. [13], Gomes and Oliveira [14], Scheithauer et al. [15], and Teng et al. [16].

Different classes of two- and three-dimensional packing problems can be considered, depending on the specific optimization criterion adopted or, for instance, if the number of containers is fixed and the total load has to be maximized (e.g., in terms of volume or value), then the relevant model is referred to as a knapsack problem [17–20]. It is reduced to the single container one, when only one container is available [21–23]. The issue of loading a set of given objects, while minimizing the number (or, more in general, the total volume/cost) of containers to utilize, is referred to as the bin packing problem (c.f. [24, 25]).

Further issues may concern the minimization of the container size, for instance, in the strip packing problem (c.f. [26–28]), where a single dimension of the (rectangle/parallelepiped-shaped) domain has to be minimized. The problem of minimizing the area/volume of a rectangular domain (c.f. [29, 30]) with a number of applications including manufacturing, logistics, and electronic design (c.f. floor-planning in very large system integration) is well known. A further example, concerning the container volume minimization, consists in sphere packing in optimized spheres [31, 32].

Significant research work has been carried out to solve a broad range of packing issues algorithmically, frequently by resorting to general meta-heuristics or by developing ad hoc heuristics. Consult, e.g., Allen et al. [33], Bennell et al. [34, 35], Bennell and Oliveira [36], Bortfeldt and Gehring [37], Burke et al. [38], Burke et al. [39], Coffman et al. [40], Dowsland et al. [41], Gehring and Bortfeldt [42], Gomes and Oliveira [14], Gonçalves and Resende [43], Hopper and Turton [44, 45], López-Camacho et al. [46], Mack et al. [47], Oliveira et al. [48], Pisinger [10], Ramakrishnan et al. [49], Terashima-Marín et al. [50], Wang et al. [51], and Yeung and Tang [52]. In parallel, modeling-based approaches have been investigated, e.g., Allen et al. [53], Chen et al. [54], Chernov et al. [55], Fasano [56], Fischetti and Luzzi [57], Hadjiconstantinou and Christofides [58], Kallrath [59], Padberg [60], and Pisinger and Sigurd [61]. These works refer to the overall context of mathematical programming, including mixed integer (linear) programming and mixed integer nonlinear programming.

Several types of complex non-standard packing problems arise in space engineering, both at the design and operational levels, c.f. Daughtrey [62], Fasano [63], Fasano et al. [64], Fasano and Pintér [65, 66], Fasano and Vola [67], and Takadama and Shimomura [68]. Exploiting the spacecraft load capacity as much as possible is needed particularly in the perspective of the demanding exploration programs of the near future, including lunar outposts, orbiting platforms, and space stations and forthcoming human missions to Mars.

Generally, concerning logistics and transportation issues, the volume or the mass of the loaded cargo has to be maximized. Other optimization criteria, however, can also be stated, depending on the specific missions in question. This is the case, for instance, of scientific satellites, where the total mass/volume to be loaded is known a priori, consisting of the necessary on-board equipment. In these circumstances, optimization criteria, for instance, relevant to the overall spacecraft balancing can be considered.

Independently from the selected optimization criteria, often very tricky accommodation rules have to be taken into account, in compliance with geometrical and operational conditions, as well as safety or ergonomic requirements, in particular, when manned missions are dealt with. Tight balancing conditions, deriving from attitude control specifications, are usually imposed at an overall system level but can also be stated (e.g., in the case of space modules) for single internal containers, for instance, racks or bags. The internal space-system geometries are normally very intricate. As a consequence, in order to exploit the empty volume, as efficiently as possible, the shape of the available packing domains (or sub-domains) can be very peculiar. This occurred, for instance, in the case of the European Automated Transfer Vehicle (ATV, ESA, cf. <www.esa.int/Our_Activities/Human_Spaceflight/ATV>, see below), conceived in support of the International Space Station Logistics (ISS, see NASA www.nasa.gov/mission_pages/station). There, indeed, some bags had curved surfaces to fit the shape of the racks they had to be accommodated in, determined by the cylindrical form of the carrier.

All in all, packing problems in space possess at least some of the following features: non-trivial shapes of domains and items, possible presence of internal separation planes or structural elements, balancing conditions, and additional accommodation rules.

As is well known, packing problems, in general, even when posed in quite an elementary version (e.g., the placement of simple boxes inside a container box, without any further conditions) are NP-hard. This encourages packing specialists to embrace an overall heuristic perspective, intended to search for satisfactory (i.e., generally sub-optimal, instead of optimal-proven) solutions. Considering the complexity of packing problems in space applications, this area is no exception.

Furthermore, while in several real-world instances, items can often be assumed to be rectangular parallelepipeds, without significant loss of information, in space engineering/transportation/logistics, generally, such an approximation does not hold, especially when dealing with large and complex items. Similar considerations can, moreover, be made, regarding the shapes of the available volumes, since frequently these are not just parallelepipeds and cannot reasonably be approximated as such. The presence of additional geometric and operational conditions, in general, makes the packing problem even more challenging.

When handling non-standard packing problems of the aforementioned typology, characterized in particular by the presence of overall conditions, such as balancing, the simplistic approach (adopted by several packing algorithms) of placing items one at a time is inefficient. A global optimization viewpoint is therefore highly desirable, also in consideration of the outstanding results recently achieved in this discipline (consult, e.g., [31, 69–86]).

Recently, a modeling-based philosophy (as opposed to a pure algorithmic one) emphasizing in particular a global optimization approach has been looked into, in support of packing optimization in space. Dedicated MILP and MINLP formulations have been presented jointly with heuristic procedures [63].

A basic concept, widely utilized in this specific framework, relates to the tetris-like items. Such items consist of a set of rectangular parallelepipeds positioned orthogonally, with respect to an orthogonal reference frame. This frame is called "local" and each parallelepiped represents a "component." A dedicated MILP-model-based heuristic approach has been introduced to solve the problem described next, c.f. Fasano [63, 87].

A set I of N tetris-like items, together with a (bounded) convex polyhedron D, is considered, see Figure 1. This is associated with a given orthogonal reference frame, indicated as the main frame. The general problem is to place items into D, maximizing the loaded volume, while considering the following positioning rules:

- Each local reference frame has to be positioned orthogonally, with respect to the main one (*orthogonality* conditions);
- For each item, each component has to be contained within D (*domain* conditions);
- The components of different items cannot overlap (*non-intersection* conditions).

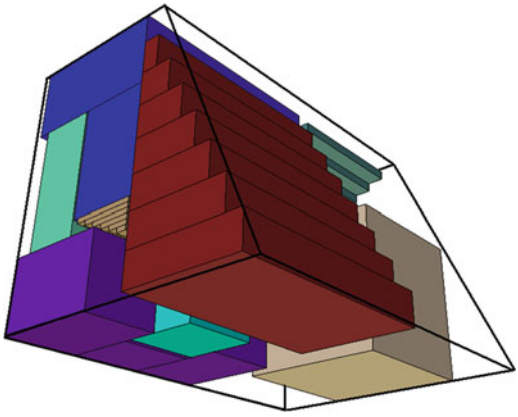


Figure 1 Tetris-like item packing into a convex domain

A number of additional conditions can be taken into account. Primarily, the one regarding the so-called *static* balancing, imposing that the overall center of mass lies within a given convex sub-domain of D. Further packing requirements can be contemplated, for instance:

- Pre-fixed position and orientation for some items;
- Minimum or maximum distance involving pairs of components (belonging to different items);
- Presence in D of "forbidden" zones (e.g., due to clearance requirements, or actual "holes") and/or separating planes;
- Maximum overall mass loaded;
- *Dynamic* balancing conditions on the relevant inertia matrix.

In recent years, an overall heuristic approach based on the aforementioned model (in addition to others concerning further specific packing aspects) found a successful application. This occurred in support of the five ATV missions, carried out by the European Space Agency (ESA), in the period 2008–2014 (see ESA http://www.esa.int). CAST (Cargo Accommodation Support Tool, see [64]) was the resulting dedicated software environment, developed for the purpose by Thales Alenia Space, with ESA funding. A modified version of the tool (see [88]) is currently in use, in support of the ISS Columbus Laboratory on-board stowage activity (see ESA www.esa.int/Our_Activities/Human_Spaceflight/Columbus). A brief overview concerning the ATV cargo accommodation problem and the overall approach adopted (by means of CAST) follows, referring the reader to [64], for a detailed exposition.

The ATV systems were intended to contribute to the resupply of the ISS, in terms of propellant (to re-boost the station to the required orbit altitude), payloads (i.e., equipment devoted to the on-board experimentation), as well as material both for the crew and the station on-board activity, in addition to its maintenance. Each ATV weighed about 20.7 tons at launch and had an overall load capacity of 8 tons. The overall cargo covered the following categories:

- *Pressurized* cargo (frequently referred to as *dry* cargo), consisting of objects of different typologies such as scientific equipment or resupply goods;
- *Unpressurized* cargo, consisting of fluids;
- Propellant for the re-boost maneuvers and station refueling.

The ATV transportation system was essentially aimed at supporting the ISS uploading phases, since it performed a destructive re-entry (the only download activity consisted of trash destruction). Both fluids and cargo items were delivered on-board the ISS, based on the current Cargo Manifest list, provided by NASA for every mission. The amount of each fluid type could generally vary within an admissible range. Some items could, moreover, be rejected, on the basis of a given priority list. In addition to intricate operational rules, tight (*static* and *dynamic*) balancing conditions were imposed at different levels (ranging from the single bag to the whole system), giving rise to both linear and nonlinear constraints (see [64]). As a matter of fact, several three-dimensional packing issues were defined, encompassing different accommodation scenarios (e.g., placement of items into bags, bags into racks, and racks into the module rack locations).

CAST is characterized by an overall architecture, based on a mathematical library that represents the core of the entire optimization framework. Specific subproblems are solved iteratively by adopting, step by step, the corresponding mathematical library module, consisting of specific (MILP) models and heuristic algorithms. Backward iterations are admitted, when the desired solution (at overall/subproblem level) is not attained, performing a recursive process. The relevant subproblems are associated with the following library modules:

- Item accommodation (IA);
- Preprocessing assessment (PPA);
- Item-rack correlation (IRC);
- Rack configuration (RC);
- (Overall) Cargo accommodation (CA).

The IA module is employed to accommodate (tetris-like) items into bags, subject to *static* balancing conditions. The available volume is moreover assumed to consist of a convex domain (in general non-box-shaped). This module is based on the MILP-model-based heuristic approach outlined above.

The PPA module is aimed at providing a fast approximate solution to the whole problem, in order to attain a preliminary feasibility check. This module serves also to identify an upper bound, relevant to the overall loadable mass. An MILP model solves a multiply constrained, multiple continuous knapsack problem with additional conditions. The set of knapsacks is represented by both tanks (for the fluids) and racks with given mass capacities. Additional conditions include:

- Specific loading rules for fluids;
- Compatibility/incompatibility between rack and rack location;
- Compatibility/incompatibility (both for *unpressurized* and *pressurized* cargo) between mass and container type;
- The *static* and *dynamic* balancing requirements stated at system level.

The relevant MILP model considers, as a very preliminary approximation, both the *unpressurized* and *pressurized* cargo in the same way as fluid mass.

The IRC module has the scope of obtaining an initial correlation between integrated bags (by means of the IA module) or pre-integrated bags and the rack locations. It is based on an MILP model. Multiply constrained, multiple (non-continuous) knapsack problems with additional conditions are solved. The model is based on a more sophisticated formulation with respect to the PPA. Items are considered in terms of distinct (flexible) objects, characterized by their mass and volumes (their actual dimensions are however still neglected). All conditions considered by the PPA model, in particular the *static* and *dynamic* balancing, are taken into account. In this case, nonetheless, in order to overcome the difficulties associated with the nonlinearity deriving from the balancing constraints, the overall mass loaded is considered as a constant (a first approximation of its value is derived by the PPA step itself). The error, expressing the difference between the constant mass estimation and the actual amount loaded by the IRC model, is minimized iteratively, as necessary. The total loaded mass is maximized only indirectly. Additionally, the presence of sectors inside the racks is considered, together with the relevant accommodation rules (the knapsacks here consist of the rack sectors and rack fronts). Some items may be rejected, if necessary, on the basis of priority rules.

The RC module has the task of determining the internal/external rack accommodation, in compliance with given accommodation rules. Integrated/pre-integrated bags, drawers, "large" items, and further specific classes of items are accommodated into the racks or on the rack fronts, on the basis of the designations provided by the previous IRC stage. All cargo items and bags involved have their actual shapes and dimensions. The *static* balancing restrictions, imposed at rack level, are taken into account. An MILP model solves the related multiply constrained, multiple knapsack problems with additional conditions. The knapsacks, here, represent the rack (internal) sectors and the external rack fronts. The additional conditions, at this stage, consist of:

- Rules for the internal accommodation (e.g., how bags of different types can be allocated into sectors);
- Mass capacity both at sector and rack-front levels;
- Rules for the external accommodation;
- *Static* balancing of the whole rack.

Since during the accommodation process (involving the aforementioned sub-problems) several approximations are introduced, the CA module has the objective of re-arranging all the partial (so-far obtained) solutions, in order to attain a satisfactory (albeit in general sub-optimal) ultimate result. The assignment of the already accommodated racks to rack locations is reconsidered, looking into a final accommodation, compliant with the given *static* and *dynamic* balancing conditions holding at system level. At this step, errors with respect to the mass loaded in each rack are allowed. They are minimized by the CA model objective function. If the

outcome obtained is acceptable, then (in terms of error tolerance) the final solution is attained. Otherwise, backward/forward iterations are executed throughout the entire accommodation process, until an acceptable result is obtained.

The remainder of this chapter is devoted to the accommodation of arbitrary polyhedra into convex containers of minimum size, as a specific packing problem of interest in space applications. A further chapter of this volume [89] addresses packing problems subject to balancing conditions. Both cases are solved by the established phi-function methodology introduced in [55, 90–93] to handle a wide range of complex non-regular packing problems. The problem of packing a collection of non-identical and non-convex polyhedra into a convex container (in particular, a cuboid, a sphere, a right circular cylinder, and an ellipsoid) of "minimum size" (e.g., in terms of total volume, homothetic coefficient, or one of the container metrical characteristics) is discussed hereinafter. It shall be denoted in the following as the polyhedron layout problem. In the relevant study discussed in this chapter, continuous rotations and translations of polyhedra are allowed. In addition, conditions upon minimum allowable distances between polyhedra are taken into account. An exact mathematical model based on the phi-function technique is formulated in terms of nonlinear programming. An efficient solution algorithm, employing a fast starting point algorithm and an innovative compaction procedure is adopted. Extensive computational results, including a comparative analysis between new and previously published instances, demonstrate the efficiency of the approach proposed.

Polyhedron packing problems are, in general, NP-hard: consequently, solution methodologies generally employ heuristics. Some researchers adopt approaches based on mathematical modeling and general optimization procedures. In most works free rotations of 3D objects are not allowed. For example, in [94] the translation algorithm is used for the packing of pre-oriented convex polytopes. Liu et al. [95] propose an algorithm, dealing with arbitrarily shaped polyhedra which can be rotated around each coordinate axis of eight discrete angles. Youn-Kyoung and Sang [96] show that, for 3D packing problems, a unit-discretization of the rotations, around each axis, from 0° to 360°, would make the actual computation impossible.

The approach illustrated here is based on the mathematical modeling of geometric relations between objects, leading to continuous NLP formulations [90, 97, 98]. The phi-function technique (e.g. [55, 92]) has, indeed, the scope of providing an analytic description of objects inside a container, taking into account their continuous translations and rotations.

In this chapter the quasi-phi-functions concept, as introduced in Stoyan et al. [91, 93], is applied. It is based on the idea proposed by Kallrath [59] of using separation planes to model non-overlapping constraints for circles and convex polygons. The concept of quasi-phi-functions extends the phi-function domain by introducing auxiliary variables. Their use, as an alternative to phi-functions, allows for simplified descriptions of non-overlapping constraints, at the expense of a larger number of variables. The phi- and quasi-phi-functions have been successfully used to model a variety of packing problems [55, 99, 100], also in space engineering, as

illustrated by Stoyan and Romanova [92], Stoyan et al. [101], and Stoyan et al. [91, 93]. In this chapter, the approach introduced by Romanova et al. [90] is extended, in order to tackle the layout problem of arbitrarily shaped polyhedra in convex containers of various types, by considering different objective functions.

Section 2 states the problem in question. The modeling methodology adopted to formulate non-overlapping, containment, and relative-distance constraints is described in Sect. 3. The mathematical model and the solution algorithm are discussed in Sects. 4 and 5, respectively, while Sect. 6 computational results are presented.

2 Polyhedron Layout Problem

We consider the layout problem in the following setting. Let $\Omega = \Omega(p)$ denote a convex container with *variable sizes* p, $\Omega = \{(x, y, z, p) \in R^3 : \Psi(x, y, z, p) \geq 0\}$, given in the global coordinate system OXYZ, where $\Psi(x, y, z, p) = \min\{\Psi_s(x, y, z, p), s = 1, \ldots, n_\Omega\}$ and $\Psi_s(x, y, z, p)$ is a differentiable function, $s = 1, \ldots, n_\Omega$. In particular, we consider the following types of containers:

A cuboid $\mathbf{B} = \{(x, y, z, l, w, h) \in R^3 | \min\{x + l, -x + l, y + w, -y + w, z + h, -z + h\} \geq 0\}$ of variable dimensions l, w, and h, $p = (l, w, h)$;

A sphere $\mathbf{S} = \{(x, y, z, r) \in \mathbb{R}^3 | r^2 - x^2 - y^2 - z^2 \geq 0\}$ of variable radius r, $p = (r)$;

A cylinder $\mathbf{C} = \{(x, y, z, \lambda) \in \mathbb{R}^3 | \min\{(\lambda r)^2 - x^2 - y^2, -z + \lambda h, z + \lambda h\} \geq 0\}$, of radius λr and height λh, where λ is a variable homothetic coefficient, $p = (\lambda)$, with $\lambda = 1$, for the original container;

An ellipsoid $\mathbf{E} = \left\{ (x, y, z, \lambda) \in \mathbb{R}^3 \middle| \lambda^2 - \frac{x^2}{a^2} - \frac{y^2}{b^2} - \frac{z^2}{c^2} \geq 0 \right\}$ of variable semiaxes λa, λb, λc, $p = (\lambda)$, with $\lambda = 1$, for the original container.

Remark: Each metrical characteristic of Ω can be a variable by itself, i.e., r and h—for cylinder, or a, b, c—for ellipsoid.

Let $\{1, 2, \ldots, N\} = J_N$ and a set of polyhedra $\mathbb{Q}_q$, $q \in J_N$, be given.

Each polyhedron $\mathbb{Q}_q$ can be either non-convex or convex. For the purposes of this paper, we assume that the mass M_q and center of mass c_q of each polyhedron $\mathbb{Q}_q$ are known.

With each polyhedron $\mathbb{Q}_q$ we associate its local coordinate system with the relevant origin denoted by v_q. Assume that each non-convex polyhedron $\mathbb{Q}_q$ is represented as a union of convex polyhedra K_j^q, $j = 1, \ldots, n_q$. With each convex polyhedron K_j^q we associate the local coordinate system of the polyhedron $\mathbb{Q}_q$. Each convex polyhedron K_j^q is defined by its vertices p_s^{qj}, $s = 1, \ldots, m_j^q$, in the local coordinate system of $\mathbb{Q}_q$.

We give here input data that form a non-convex polyhedron $\mathbb{Q}_q$ by two lists: List_1 contains the vertex coordinates of all the convex polyhedra $K_j^q, j = 1, \ldots, n_q$, and List_2 contains the index sets J_j^q, $j = 1, \ldots, n_q$, of the numbers of vertices (with respect to List_1) that define appropriate convex polyhedra $K_j^q, j = 1, \ldots, n_q$. We note that List_1 includes all the original vertices of the non-convex polyhedron and the additional vertices that appear as a result of decomposing the non-convex polyhedron into convex polyhedral (see [90] for details).

Let $\mathbb{Q}_q = \bigcup_{j=1}^{n_q} K_j^q$ be given. Without loss of generality, we assume that the origin v_q of a polyhedron $\mathbb{Q}_q$ coincides with the center point of its circumscribed sphere S_q of radius r_q. In order to circumscribe a sphere around a polyhedron we employ the algorithm described in [102], which computes the smallest enclosing sphere of a collection of points. We use the library function found at https://github.com/hbf/miniball, which is sufficiently fast.

The location and orientation of each polyhedron $\mathbb{Q}$ are defined by a vector $u = (v, \theta)$ of its variable placement parameters. Here $v = (x, y, z)$ is a translation vector, and $\theta = (\theta^1, \theta^2, \theta^3)$ is a vector of rotation parameters, where θ^1, θ^2, and θ^3 are Euler angles (https://en.wikipedia.org/wiki/Euler_angles).

A polyhedron rotated through angles θ^1, θ^2, and θ^3 and translated by vector v is denoted as $\mathbb{Q}(u) = \{t \in R^3 : t = v + M(\theta) \bullet t^0, t^0 \in \mathbb{Q}^0\}$, where $u = (v, \theta)$, $\mathbb{Q}^0$ denotes the non-translated and non-rotated polyhedron $\mathbb{Q}$, and $M(\theta)$ is a rotation matrix of the form:

$$M(\theta) = \begin{pmatrix} \mu_{11} & \mu_{12} & \mu_{13} \\ \mu_{21} & \mu_{22} & \mu_{23} \\ \mu_{31} & \mu_{32} & \mu_{33} \end{pmatrix},$$

$$\mu_{11} = \cos\theta^1 \cos\theta^3 - \sin\theta^1 \cos\theta^2 \sin\theta^3, \quad \mu_{12} = -\cos\theta^1 \sin\theta^3 - \sin\theta^1 \cos\theta^2 \cos\theta^3,$$

$$\mu_{13} = \sin\theta^1 \sin\theta^2, \quad \mu_{21} = \sin\theta^1 \cos\theta^3 + \cos\theta^1 \cos\theta^2 \sin\theta^3,$$

$$\mu_{22} = -\sin\theta^1 \sin\theta^3 + \cos\theta^1 \cos\theta^2 \cos\theta^3, \quad \mu_{23} = -\cos\theta^1 \sin\theta^2,$$

$$\mu_{31} = \sin\theta^2 \sin\theta^3, \quad \mu_{32} = \sin\theta^2 \cos\theta^3, \quad \mu_{33} = \cos\theta^2.$$

It is possible to define minimum allowable distances between each pair of polyhedra $\mathbb{Q}_q$ and $\mathbb{Q}_g$, $q < g \in I_N$, as well as between a polyhedron $\mathbb{Q}_q$, $q \in I_N$, and the boundary of container Ω. It means that each polyhedron $\mathbb{Q}_q$ has to be located no closer to polyhedron $\mathbb{Q}_g$ than the given allowable distance and each polyhedron $\mathbb{Q}_q$ has to be located inside the container and no closer to the boundary of the container than the given allowable distance. We note that the minimum allowable

distance between each pair of convex polyhedra $K_j^q \subset \mathbb{Q}_q$, $j = 1, \ldots, n_q$, and $K_l^g \subset \mathbb{Q}_g$, $l = 1, \ldots, n_g$, $q < g \in I_N$, is equal to the given allowable distance between the original polyhedra $\mathbb{Q}_q$ and $\mathbb{Q}_g$. Moreover, the minimum allowable distance between each polyhedron K_j^q, $q \in I_N$, and the boundary of the container Ω is equal to the given allowable distance between the original polyhedron $\mathbb{Q}_q$, $q \in I_N$, and the boundary of container Ω. Balance constraints can be taken into account as the minimum deviation of the mechanical system from a given point.

We consider different types of objective functions that depend on placement parameters of polyhedra and variable metrical characteristics of container Ω, in particular: minimum volume, minimum of homothetic coefficient, maximum of packing density, and minimum deviation of the mechanical system from a given point. Note that the metrical characteristic of Ω can be variable or fixed.

The polyhedron layout problem can be formulated in the form:

Place the set of polyhedra $\mathbb{Q}_q$, $q \in J_N$, within a convex container Ω, taking into account the geometric and balance constraints, such that the value of a given objective function will reach its appropriate extremum value.

3 Analytical Description of Placement Constraints

In this section we describe our methodology for the mathematical modeling of the non-overlapping, containment and distance constraints. In the following, $d(a, b)$ represents the Euclidean distance between two points $a, b \in R^3$.

Let us consider placement constraints that are met in the polyhedron layout problem:

Let $\rho_{qg} > 0$ denote the minimum allowable distance between two polyhedra $\mathbb{Q}_q$ and $\mathbb{Q}_g$ and $\rho_q > 0$ denote the minimum allowable distance between a polyhedron $\mathbb{Q}_q$ and the object Ω^*.

We formulate the placement constraints in the form:

1. $\text{dist}(\mathbb{Q}_q, \mathbb{Q}_g) \geq \rho_{qg}$ for each $q, g \in J_N$ with $q \neq g$,

   where $\text{dist}\left(\mathbb{Q}_q, \mathbb{Q}_g\right) = \min\limits_{a \in \mathbb{Q}_q, b \in \mathbb{Q}_g} d(a, b)$.

   It means that each polyhedron $\mathbb{Q}_q$ has to be located no closer to polyhedron $\mathbb{Q}_g$ than the given allowed distance ρ_{qg}.

2. $\text{dist}(\mathbb{Q}_q, \Omega^*) \geq \rho_q$ for each $q \in J_N$, $\Omega^* = R^3 \backslash \text{int}\, \Omega$,

   where $\text{dist}\left(\mathbb{Q}_q, \Omega^*\right) = \min\limits_{a \in \mathbb{Q}_q, b \in \Omega^*} d(a, b)$.

This means that each polyhedron $\mathbb{Q}_q$ has to be located inside the container and no closer to the boundary of the container than the given allowed distance ρ_q.

Note that if $\rho_q = 0$ and $\rho_q = 0$, then we deal with regular non-overlapping ($\text{int}\, \mathbb{Q}_q \cap \text{int}\, \mathbb{Q}_g = \varnothing$) and containment, $\mathbb{Q}_q \subset \Omega \iff \text{int}\, \mathbb{Q}_q \cap \Omega^*$, constraints.

In order to describe analytically the relationships between a pair of objects considered in the placement constraints, we employ the phi-function technique. Here we use adjusted phi-functions [55] for containment constraints and quasi-phi-functions [91, 93] for non-overlapping constraints. See Appendix for definitions of phi-functions and quasi-phi-functions.

3.1 Modeling Non-overlapping Constraints

To construct a quasi-phi-function and an adjusted quasi-phi-function of two non-convex polyhedra, we will use a quasi-phi-function and an adjusted quasi-phi-function for each pair of convex polyhedra that together form the original non-convex polyhedra.

First we consider a quasi-phi-function for a pair of convex polyhedra.

Let $A(u_A)$ and $B(u_B)$ be two convex polyhedra given by their vertices p_s^A, $s = 1, \ldots, m_A$, and p_s^B, $s = 1, \ldots, m_B$.

A quasi-phi-function $\Phi'^{AB}(u_A, u_B, u_P)$ for convex polyhedra $A(u_A)$ and $B(u_B)$ can be defined by the following formula:

$$\Phi'^{AB}(u_A, u_B, u_P) = \min\left\{\Phi^{AP}(u_A, u_P), \Phi^{BP^*}(u_B, u_P)\right\}, \tag{1}$$

where $\Phi^{AP}(u_A, u_P)$ is a phi-function of $A(u_A)$ and half plane $P(u_P)$, $\Phi^{BP^*}(u_B, u_P)$ is a phi-function of $B(u_B)$ and half plane $P^*(u_P)$ (the complement to $P(u_P)$), $P(u_P) = \{(x, y, z) : \psi_P = \alpha \cdot x + \beta \cdot y + \gamma \cdot z + \mu_P \le 0\}$ is a half-space, $\alpha = \sin\theta_P^1 \sin\theta_P^2, \beta = -\cos\theta_P^1 \sin\theta_P^2, \gamma = \cos\theta_P^2, \theta_P^1$ and θ_P^2 are appropriate variable Euler angles (under intrinsic rotation $\theta_P^3 = 0$), $u_P = \left(\theta_P^1, \theta_P^2, \mu_P\right)$ is a vector of variable parameters (auxiliary variables) that define a plane $L_{AB} = \{(x, y, z) : \psi_P = 0\}$ in R^3,

$$\Phi^{AP}(u_A, u_P) = \min_{1 \le s \le m_A} \psi_P\left(p_s^A\right), \quad \Phi^{BP^*}(u_B, u_P) = \min_{1 \le s \le m_B}\left(-\psi_P\left(p_s^B\right)\right).$$

We use here the important characteristic of a *quasi-phi-function*: if $\Phi'^{AB}(u_A, u_B, u_P) \ge 0$ for some u_P, then int$A(u_A) \cap$ int $B(u_B) = \varnothing$(see [91, 93] for details).

It is known that if two fixed convex objects A and B do not have common points, then there exists at least one separating plane. Therefore there exists a vector u_P^* of parameters of a plane L_{AB} such that the distance $d_1 = \Phi^{AP}\left(u_A, u_P^*\right)$ from A to L_{AB} equals to the distance $d_2 = \Phi^{BP^*}\left(u_B, u_P^*\right)$ from B to L_{AB}. Thus function $\Phi'^{AB}(u_A, u_B, u_P)$ reaches its maximum when $\Phi'^{AB}\left(u_A, u_B, u_P^*\right) = d^*$, where $d^* = d_1 = d_2$.

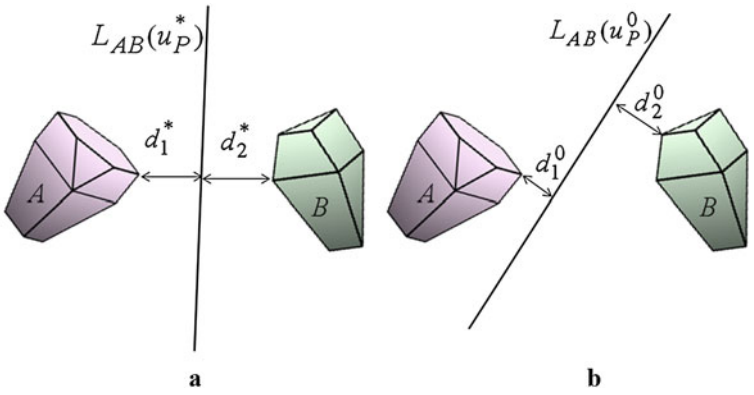


Figure 2 Separating planes for two fixed convex objects A and B: (**a**) $\Phi'^{AB} = d^*$; (**b**) $\Phi'^{AB} = d_1^0$

Figure 2 illustrates two cases when $\Phi'^{AB} > 0$: (a) $\Phi'^{AB}\left(u_1, u_2, u_P^0\right) = \min\left\{d_1^0, d_2^0\right\} = d_1^0$; (b) $\max\limits_{u_P} \Phi'^{AB}(u_1, u_2, u_P) = \Phi'^{AB}\left(u_1, u_2, u_P^*\right) = \min\left\{d_1^*, d_2^*\right\} = d_1^* = d_2^* = d^*$.

Let the *minimum allowable distance* ρ_{AB} between two arbitrary convex polyhedra $A(u_A)$ and $B(u_B)$ be given. To describe a *distance* constraint, $\text{dist}(A(u_A), B(u_B)) \geq \rho_{AB}$, we use an adjusted quasi-phi-function for convex polyhedra $A(u_A)$ and $B(u_B)$ of the form

$$\widehat{\Phi'}^{AB}(u_A, u_B, u_P) = \Phi'^{AB}(u_A, u_B, u_P) - 0.5\rho_{AB}. \tag{2}$$

Note that $\widehat{\Phi'}^{AB}(u_A, u_B, u_P) \geq 0$ implies $\text{dist}(A, B) \geq \rho_{AB}$ (see [91, 93] for details).

A quasi-phi-function of two non-convex polyhedra is composed by quasi-phi-functions for all pairs of convex polyhedra that together form the original non-convex polyhedra. By analogy an adjusted quasi-phi-function of two non-convex polyhedra is constructed.

Before we introduce a quasi-phi-function and an adjusted quasi-phi-function for a pair of non-convex polyhedra, we renumber a given collection of convex polyhedra, K_j^q, $j = 1, \ldots, n_q$, $q \in J_N$, as a set of $n = \sum\limits_{q=1}^{N} n_q$ convex polyhedra K_i, $i \in \{1, 2, \ldots, n\} = I_n$, using the following rule: $K_j^q \to K_i$, $i = \sum\limits_{l=0}^{q-1} n_l + j$, $j = 1, \ldots, n_q$, $q \in J_N$, provided that $n_0 = 0$.

Now we introduce the "gluing" vector $\mathbf{a} = (a_1, , , , a_n)$, $a_i \in J_N$, where $a_i = q$, if K_i takes part in the composition of a polyhedron $\mathbb{Q}_q$, $q \in J_N$. Let $I_n = I^1 \cup I^2 \cup \ldots \cup I^N$ be an ordered partition of I_n, where $I^q = \{i \in I_n, a_i = q\}$,

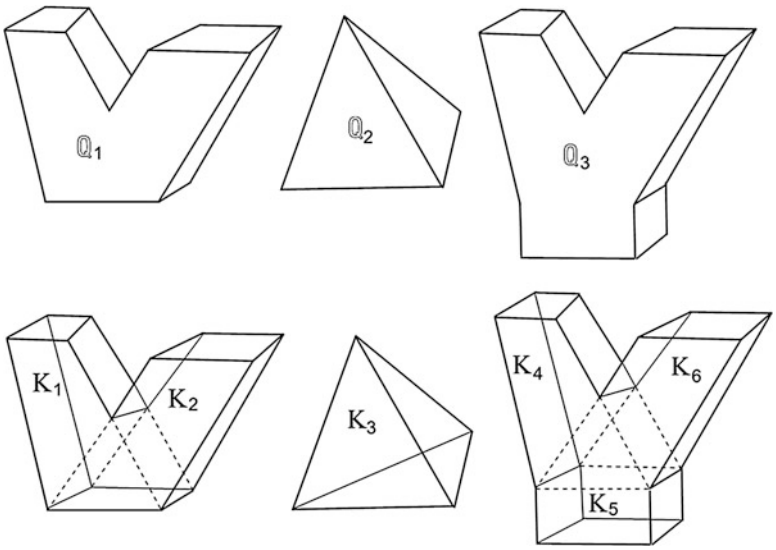


Figure 3 Generation of the "gluing" vector for polyhedra $\mathbb{Q}_1$, $\mathbb{Q}_2$, and $\mathbb{Q}_3$

$|I^q| = n_q$, $q \in J_N$. For example, the "gluing" vector for polyhedra $\mathbb{Q}_1 = K_1 \cup K_2$, $\mathbb{Q}_2 = K_3$, and $\mathbb{Q}_3 = K_4 \cup K_5 \cup K_6$ has the form $\mathbf{a} = (a_1, a_2, a_3, a_4, a_5, a_6)$ $=(1, 1, 2, 3, 3, 3)$ (Figure 3). In the example $N = 3$ and $n = \sum_{q=1}^{3} n_q = 2+1+3 = 6$.

Let $\mathbb{Q}_q = \underset{i \in I^q}{\cup} K_i$ and $\mathbb{Q}_g = \underset{j \in I^g}{\cup} K_j$ be non-convex polyhedra and $q \neq g$.

An adjusted quasi-phi-function for non-convex polyhedra $\mathbb{Q}_q(u_q)$ and $\mathbb{Q}_g(u_g)$ has the form (see [90]):

$$\widehat{\boldsymbol{\Phi}}'_{qg}\left(u_q, u_g, u_{qg}\right) = \min\left\{\widehat{\Phi}'_{ij}\left(u_q, u_g, u'_{ij}\right), i \in I^q, j \in I^g\right\}, \tag{3}$$

where $\widehat{\Phi}'_{ij}\left(u_q, u_g, u'_{ij}\right)$ is the adjusted quasi-phi-function and u'_{ij} is a vector of auxiliary variables for a pair of convex polyhedra $K_i(u_q)$ and $K_j(u_g)$, $i \in I^q, j \in I^g$, $u_{qg} = \left(u'_{ij}, i \in I^q, j \in I^g\right)$.

If ρ_{qg}=0, then function (3) is a quasi-phi-function $\Phi'_{ij}\left(u_q, u_g, u'_{ij}\right)$ for $\mathbb{Q}_q$ and $\mathbb{Q}_g$.

Let us consider an example of a quasi-phi-function for two polyhedra: $\mathbb{Q}_1(u_1) = K_1(u_1)$ and $\mathbb{Q}_2(u_2) = K_2(u_2) \cup K_3(u_2)$ (Figure 4a). A quasi-phi-function for $\mathbb{Q}_1(u_1)$ and $\mathbb{Q}_2(u_2)$ can be defined in the following form: $\boldsymbol{\Phi}'_{12}(u_1, u_2, u_{12}) = \min\left\{\Phi'_{12}\left(u_1, u_2, u'_{12}\right), \Phi'_{13}\left(u_1, u_2, u'_{13}\right)\right\}$,where $u_{12} =$

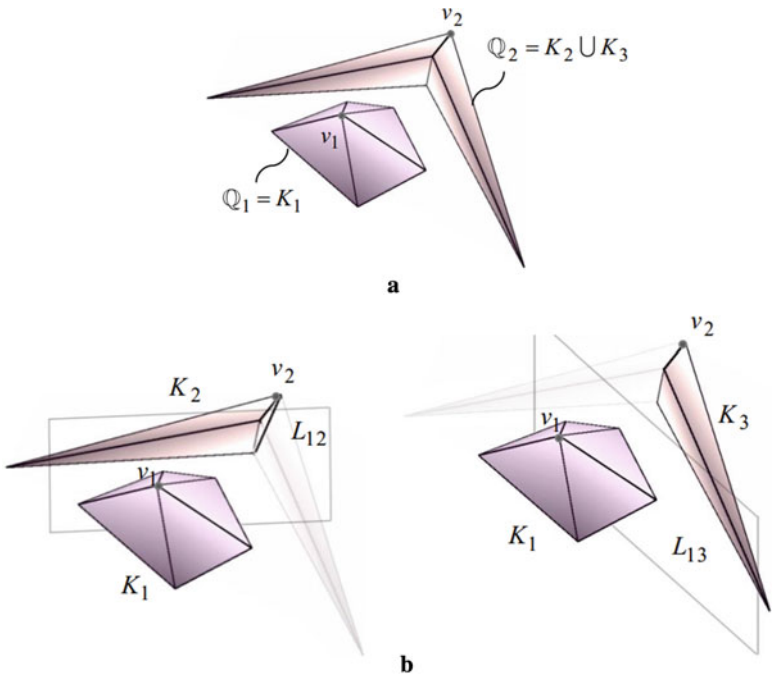


Figure 4 Illustration to a quasi-phi-function for two polyhedra: (**a**) polyhedra $\mathbb{Q}_1$ and $\mathbb{Q}_2$; (**b**) separating planes L_{12} and L_{13} for two pairs of appropriate convex polyhedra K_1 and K_2; K_1 and K_3

$\left(u'_{12}, u'_{13}\right)$, $\Phi'_{12}\left(u_1, u_2, u'_{12}\right)$ is a quasi-phi-function, u'_{12} is a vector of auxiliary variables for a pair of convex polyhedra $K_1(u_1)$ and $K_2(u_2)$, $\Phi'_{13}\left(u_1, u_2, u'_{13}\right)$ is a quasi-phi-function, and u'_{13} is a vector of auxiliary variables for a pair of convex polyhedra $K_1(u_1)$ and $K_3(u_2)$.

Figure 4b illustrates two separating planes L_{12} and L_{13} that provide $\Phi'_{12}\left(u_1, u_2, u'_{12}\right) > 0$ and $\Phi'_{13}\left(u_1, u_2, u'_{13}\right) > 0$ that implies $\mathbf{\Phi}'_{12}\left(u_1, u_2, u_{12}\right) > 0$. Here $L_{ij} = \{(x, y, z) : \psi_{ij} = 0\}$ is a separating plane for $K_i(u_q)$ and $K_j(u_g)$, where $\psi_{ij} = \alpha_{ij} \cdot x + \beta_{ij} \cdot y + \gamma_{ij} \cdot z + \mu_{ij}$, $\alpha_{ij} = \sin\theta^1_{ij} \sin\theta^2_{ij}$, $\beta_{ij} = -\cos\theta^1_{ij} \sin\theta^2_{ij}$, $\gamma_{ij} = \cos\theta^2_{ij}$, and $u'_{ij} = \left(\theta^1_{ij}, \theta^2_{ij}, \mu_{ij}\right)$, $i = 1, j = 2, 3, q = 1, g = 2$.

3.2 Modeling Containment Constraints

An adjusted phi-function for a non-convex polyhedron $\mathbb{Q}_q(u_q)$ and the object Ω^* can be defined in the form [90]:

$$\widehat{\mathbf{\Phi}}_q\left(u_q,\, p\right) = \min\left\{\widehat{\Phi}_i\left(u_q,\, p\right), i \in I^q\right\}, \tag{4}$$

where $\widehat{\Phi}_i\left(u_q,\, p\right)$ is an adjusted phi-function for a convex polyhedron $K_i(u_q)$ and Ω^*, $i \in I^q$. Replacing each adjusted phi-function $\widehat{\Phi}_i\left(u_q,\, p\right)$ in (4) by a phi-function $\Phi_i(u_q, p)$ for $i \in I^q$, we can get a phi-function $\mathbf{\Phi}_q(u_q, p)$ for a polyhedron $\mathbb{Q}_q(u_q)$ and the object Ω^*.

To describe a containment constraint, $K_i(u_q) \subset \Omega \iff \text{int } K_i(u_q) \cap \Omega^* = \varnothing$, we use a phi-function for a convex polyhedron $K_i(u_q)$ and the object Ω^*.

Let $K_i(u_q)$ be a convex polyhedron, given in its local coordinate system by their vertices p_k^i, $k = 1, \ldots, m_i$, where $p_k^i = \left(p_{xk}^i, p_{yk}^i, p_{zk}^i\right)$.

A phi-function for $K_i(u_q)$ and the object Ω^* has the form

$$\Phi_i\left(u_q,\, p\right) = \min\left\{\Psi_s\left(u_q,\, p\right), s = 1, \ldots, n_\Omega\right\}, \tag{5}$$

$$\Psi_s\left(u_q,\, p\right) = \min\left\{\varphi_s\left(p_k^i\left(u_q\right),\, p\right), k = 1, \ldots, m_i\right\}.$$

In particular, we use the following phi-functions for containment constraints:
A phi-function for a convex polyhedron $K_i(u_q)$ and the object $\mathbf{S}^* = \mathbb{R}^3/\text{ int } \mathbf{S}$

$$\Phi^{K_i\mathbf{S}^*}\left(u_q, r\right) = \Psi\left(u_q, r\right) = \min\left\{\varphi\left(p_k^i\left(u_q\right), r\right), k = 1, \ldots, m_i\right\},$$
$$\varphi\left(p_k^i\left(u_q\right), r\right) = (r)^2 - \left(p_{kx}^i\left(u_q\right)\right)^2 - \left(p_{ky}^i\left(u_q\right)\right)^2 - \left(p_{kz}^i\left(u_q\right)\right)^2.$$

A phi-function for a convex polyhedron $K_i(u_q)$ and the object $\mathbf{C}^* = \mathbb{R}^3/\text{ int } \mathbf{C}$

$$\Phi^{K_i\mathbf{C}^*}\left(u_q, \lambda\right) = \min\left\{\Psi_s\left(u_q, \lambda\right), s = 1, 2, 3\right\},$$

$$\Psi_s\left(u_q, \lambda\right) = \min\left\{\varphi_s\left(p_k^i\left(u_q\right), \lambda\right), k = 1, \ldots, m_i\right\},$$
$$\varphi_1\left(p_k^i\left(u_q\right), \lambda\right) = (\lambda r)^2 - \left(p_{xk}^i\left(u_q\right)\right)^2 - \left(p_{yk}^i\left(u_q\right)\right)^2,$$
$$\varphi_2\left(p_k^i\left(u_q\right), \lambda\right) = -p_{zk}^i\left(u_q\right) + \lambda h,$$
$$\varphi_3\left(p_k^i\left(u_q\right), \lambda\right) = p_{zk}^i\left(u_q\right) + \lambda h.$$

A phi-function for a convex polyhedron $K_i(u_q)$ and the object $\mathbf{B}^* = \mathbb{R}^3/\text{ int } \mathbf{B}$

$$\Phi^{K_i\mathbf{B}^*}\left(u_q, l, h, w\right) = \min\left\{\Psi_s\left(u_q, l, h, w\right), s = 1, \ldots, 6\right\},$$

$$\Psi_s\left(u_q, l, h, w\right) = \min\left\{\varphi^i_{ks}\left(u_q, l, h, w\right), k = 1, \ldots, m_i\right\},$$

$$\varphi^i_{k1}\left(u_q, l, h, w\right) = p^i_{xk}\left(u_q\right) + l, \;\; \varphi^i_{k2}\left(u_q\right) = -p^i_{xk}\left(u_q\right) + l,$$

$$\varphi^i_{k3}\left(u_q, l, h, w\right) = p^i_{yk}\left(u_q\right) + w, \;\; \varphi^i_{k4}\left(u_q, l, h, w\right) = -p^i_{yk}\left(u_q\right) + w,$$

$$\varphi^i_{k5}\left(u_q, l, h, w\right) = p^i_{zk}\left(u_q\right) + h, \;\; \varphi^i_{k6}\left(u_q, l, h, w\right) = -p^i_{zk}\left(u_q\right) + h.$$

A phi-function for a convex polyhedron $K_i(u_q)$ and the object $\mathbf{E}^* = \mathbb{R}^3 / \operatorname{int}\mathbf{E}$

$$\Phi^{K_i\mathbf{E}^*}\left(u_q, \lambda\right) = \Psi\left(u_q, \lambda\right) = \min\left\{\varphi\left(p^i_k\left(u_q\right), \lambda\right), k = 1, \ldots, m_i\right\},$$
$$\varphi\left(p^i_k\left(u_q\right), \lambda\right) = (\lambda)^2 - \frac{\left(p^i_{kx}(u_q)\right)^2}{a^2} - \frac{\left(p^i_{ky}(u_q)\right)^2}{b^2} - \frac{\left(p^i_{kz}(u_q)\right)^2}{c^2}.$$

Let the minimum allowable distance ρ_q between a convex polyhedron $K_i(u_q)$ and the object Ω^* be given. To describe distance constraint, dist(K_i,Ω^*)$\geq\rho_q$, we use an adjusted phi-function for a convex polyhedron $K_i(u_q)$ and the object Ω^* defined by

$$\widehat{\Phi}^{K_i\Omega^*}\left(u_q, p\right) = \Phi_i\left(u_q, p\right) - \rho_q. \tag{6}$$

4 Mathematical Model

The vector $u \in R^\sigma$ of all variables can be described as follows: $u = (\varsigma, \tau) \in R^\sigma$, where $\varsigma = (p, u_1, u_2, \ldots, u_N)$, p is a vector of variable dimensions of the given container Ω, and $u_{a_i} = \left(v_{a_i}, \theta_{a_i}\right) = \left(x_{a_i}, y_{a_i}, z_{a_i}, \theta^1_{a_i}, \theta^2_{a_i}, \theta^3_{a_i}\right)$ is the vector of placement parameters of K_i, $i \in I_n$, an index $a_i \in \{1, 2, \ldots, N\}$ is a component of the "gluing" vector $\mathbf{a}$, defined in Sect. 3 Here $\tau = (u_{P1}, \ldots, u_{Pm})$ denotes the vector of all auxiliary variables, where $u^s_P = \left(\theta^{1s}_P, \theta^{2s}_P, \mu^s_P\right)$ is a vector of auxiliary variables for the sth pair of convex polyhedra defined in (1), $s = 1, \ldots, m$, $m = \mathit{card}\,(\Xi)$,

$$\Xi = \left\{(i, j), a_i \neq a_j, i < j, i, j \in I_n\right\}. \tag{7}$$

The number of the problem variables is derived as $\sigma = |p| + 6N + 3m$.

Now a mathematical model of the polyhedron layout problem can be stated in the form

$$\underset{u\in W\subset R^{\sigma}}{extr} F(u), \tag{8}$$

$$W=\left\{u\in R^{\sigma}:\widehat{\Phi'}_{ij}\left(u_{a_i},u_{a_j},u'_{a_ia_j}\right)\geq 0,(i,j)\in\Xi,\widehat{\Phi}_i\left(p,u_{a_i}\right)\geq 0,i\in I_n\right\}, \tag{9}$$

where $\widehat{\Phi'}_{ij}\left(u_{a_i},u_{a_j},u'_{a_ia_j}\right)$ is an adjusted quasi-phi-function defined by (2), a_i, $a_j\in J_N$, under $(i,j)\in\Xi$, $u'_{a_ia_j}=u^s_P$, $s=1,\ldots,m$, Ξ is given by (7), for the pair of polyhedra K_i and K_j, taking into account the minimum allowable distance $\rho_{qg}>0$, and $\widehat{\Phi}_i\left(u_{a_i},p\right)$ is an adjusted phi-function defined by (6) for a polyhedron K_i and the object Ω^*, taking into account the minimum allowable distance $\rho_q>0$. If $\rho_{qg}=0$ and $\rho_q=0$, then we replace the adjusted quasi-phi-function $\widehat{\Phi'}_{ij}\left(u_{a_i},u_{a_j},u'_{a_ia_j}\right)$ by the quasi-phi-function $\Phi'_{ij}\left(u_{a_i},u_{a_j},u'_{a_ia_j}\right)$, defined by (1), to enforce the non-overlapping constraint and the adjusted phi-function $\widehat{\Phi}_i\left(u_{a_i},p\right)$ by the phi-function $\Phi_i\left(u_{a_i},p\right)$, defined in (5), to enforce the containment constraint.

It should be noted that in order to avoid redundant inequalities in containment constraints one can use a collection of adjusted phi-functions $\widehat{\mathbf{\Phi}'}^{h}_{q}\left(u_q,p\right)$, $q\in J_N$, for the convex hull of non-convex polyhedra $\mathbb{Q}_q$, $q\in J_N$, instead of the collection of adjusted phi-functions $\widehat{\Phi}_i\left(u_{a_i},p\right)\geq 0$, $i\in I_n$, for convex polyhedra K_i, $i\in I_n$.

Each quasi-phi-function inequality in (9) is presented by a system of inequalities with analytical functions. Our model (8) and (9) is a non-convex and continuous nonlinear programming problem and an exact formulation for the polyhedron layout problem. It contains all globally optimal solutions. It is possible, at least in theory, to use a global solver for the nonlinear programming problem and to obtain a solution, which is an optimal layout.

However in practice, the model contains a large number of variables and a huge number of inequalities. Specifically, the model (8) and (9) involves $O(n^2)$ nonlinear inequalities and $O(n^2)$ variables due to the auxiliary variables in quasi-phi-functions, where n is the number of convex polyhedra. As a result, finding a globally optimal solution becomes an unrealistic task for the available state of the art NLP-solvers applied directly to model (8) and (9): for $N>15$ starting from a random point and for $N>30$ starting from a feasible point.

In order to search for a "good" locally optimal polyhedron layout within a reasonable computational time, we propose here an efficient solution algorithm, which employs a fast starting point algorithm and a compaction procedure. In most cases the procedure reduces our problem to a sequence of nonlinear programming subproblems of considerably smaller dimension ($O(n)$) and a smaller number of nonlinear inequalities ($O(n)$).

5 Solution Algorithm

Our multi-start solution strategy involves the following steps:

1. Generate a set $\{\varsigma^0\}$ of vectors $\varsigma^0 = \left(p^0, u_1^0, \ldots, u_N^0\right)$ of feasible placement parameters $\left(u_1^0, \ldots, u_N^0\right)$ of polyhedra placed into the container Ω^0 of sizes (p^0) in problem (8) and (9). Various algorithms exist for obtaining a feasible solution (for example. [99]). We employ here the fast algorithm described in Sect. 5.1.
2. Search for a local extremum of the objective function $F(u)$ in problem (8) and (9), starting from each point from the set $\{\varsigma^0\}$ obtained at Step 1. To get a local extremum of problem (8) and (9) we develop a compaction algorithm for rotated polyhedra described in Sect. 5.2.
3. Choose the best local minimum from those found at Step 2 as the solution of problem (8) and (9).

The actual search for a local extremum in all optimization procedures (to realize Steps 1–2) is performed by IPOPT [103], which is available at an open access noncommercial software depository (https://projects.coin-or.org/Ipopt).

5.1 Feasible Placement Parameters Algorithm (FPPA)

In order to find a vector of starting feasible placement parameters of polyhedra we apply an algorithm, which is based on the homothetic (scaling) transformation of objects. The algorithm consists of the following steps:

Step 1 If the container does not have variable metrical characteristics, then for the purpose of the algorithm we assume that $\Omega = \Omega(p)$, where $p = (\lambda)$ is a variable homothetic coefficient.

Step 2 Choose a sufficiently large starting size for a container $\Omega^0 = \Omega(p^0)$ to allow for a placement of all embedding spheres S_q^ρ, $q \in J_N$, within the container Ω^0, where $S_q^\rho = S_q \oplus S^\rho$ is the Minkowski sum of a sphere S_q of radius r_q (Figure 5) and a sphere S^ρ of radius $\rho = 0.5 \max\left\{\max_{q,g \in J_N} \rho_{qg}, \max_{q \in J_N} \rho_q\right\}$, provided that S_q and S^ρ have the same center point.

Step 3 Generate within the container Ω^0 a set of N randomly chosen center points $\left(x_q^0, y_q^0, z_q^0\right)$ of S_q^ρ, $i \in I^q$.

Step 4 Grow the spheres S_q^ρ of radius $\beta(r_q + \rho)$, $q \in J_N$, starting from $\beta = 0$ to the full size ($\beta = 1$); the decision variables are: the centers of S_q^ρ and a homothetic coefficient β, where $0 \le \beta \le 1$ (Figure 6). In order to realize this step we fix $p = p^0$ and, starting from the point $v^0 = \left(x_1^0, y_1^0, z_1^0, \ldots, x_N^0, y_N^0, z_N^0, \beta^0 = 0\right)$, solve the following NLP subproblem:

Figure 5 Non-convex polyhedra $\mathbb{Q}_q$ and appropriate circumscribed spheres S_q

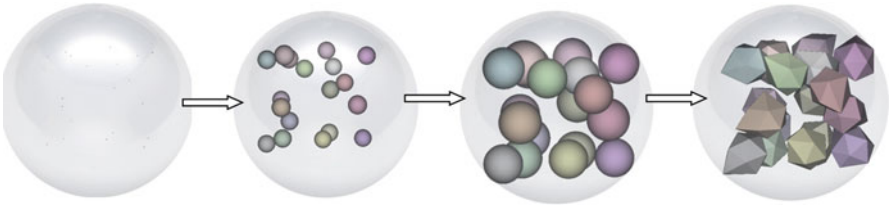


Figure 6 Illustration of the optimization procedure FPPA to search for feasible placement parameters of convex polyhedra within a sphere, using homothetic transformations

$$\max_{v \in W_\beta} \beta, \tag{10}$$

$$W_\beta = \left\{ v \in \mathrm{R}^{3N+1} : \widehat{\Phi}^{S_q S_g}(v) \geq 0, \widehat{\Phi}^{*}_{q}(v) \geq 0, q < g, q, g \in J_N, 1 - \beta \geq 0, \beta \geq 0 \right\}, \tag{11}$$

where $v = (x_1, y_1, z_1, \ \ldots \ , x_N, y_N, z_N, \beta)$,

$$\widehat{\Phi}^{S_q S_g}(v) = \left(x_q - x_g\right)^2 + \left(y_q - y_g\right)^2 + \left(z_q - z_g\right)^2 - \beta^2 \left(r_q + 2\rho + r_g\right)^2,$$

$\widehat{\Phi}^{*}_{q}(v)$ is an adjusted phi-function for sphere S_q of radius βr_q and the object Ω^* if Ω is a cuboid, a cylinder, a sphere, or a container that is formed by means of the shapes mentioned, otherwise we apply an adjusted phi-function for the oriented convex hull H_q of $\beta\mathbb{Q}_q$ and the object Ω^*.

We denote a point of the global maximum of problem (10) and (11) by $v^* = \left(x_1^*, y_1^*, z_1^*, \ldots, x_N^*, y_N^*, z_N^*, \beta^* = 1\right)$.

Step 5 Form a vector of feasible parameters $\varsigma^0 = \left(p^0, u_1^0, \ldots, u_q^0, \ldots, u_N^0\right)$, assuming that $u_q^0 = \left(x_q^0, y_q^0, z_q^0, \theta_q^0\right)$, $\left(x_q^0, y_q^0, z_q^0\right) = \left(x_q^*, y_q^*, z_q^*\right)$, and θ_q^0 is a vector of randomly generated rotation parameters of polyhedra $\mathbb{Q}_q$,$q \in J_N$.

We note that the global solution of problem (10) and (11) always can be found (since the chosen starting sizes p^0 at the first step are sufficiently large).

Step 6 If the container in problem (10) and (11) has variable metrical characteristics, then go to Step 7, otherwise (see Step 1) we solve the following auxiliary subproblem, using the COMPOLY algorithm described in Sect. 5.2 and starting from point $\varsigma^0 = \left(p^0, u_1^0, \ldots, u_q^0, \ldots, u_N^0\right)$:

$$u^* = \left(\varsigma^*, \tau^*\right) = \arg\min \lambda \;\; \text{s.t.} \;\; u \in W \subset R^\sigma, \tag{12}$$

$$W = \left\{u \in R^\sigma : \widehat{\Phi}'_{ij}\left(u_{a_i}, u_{a_j}, u'_{a_i a_j}\right) \geq 0, (i, j) \in \Xi, \widehat{\Phi}_i\left(u_{a_i}\right) \geq 0, \lambda \geq 1, i \in I_n\right\}. \tag{13}$$

If the global solution of problem (12) and (13) has been found, i.e., $\lambda = 1$, then assume $\varsigma^0 = \varsigma^*$ and go to Step 7, otherwise ς^0 has not been found for problem (8) and (9) and we stop the algorithm.

Step 7 Return the vector ς^0 to generate a starting point $u^0 = (\varsigma^0, \tau^0)$ for a subsequent search for a local minimum of problem (8) and (9). To search for vector τ^0 we apply a special optimization procedure, called feasible auxiliary parameters algorithm (FAPA), described below.

5.2 Compaction Algorithm (COMPOLY)

Since our problem (8) and (9) cannot be solved for $N > 30$ by direct use of state of the art NLP-solvers (starting from a feasible point), we propose an iterative compaction algorithm to search for local minima of the problem.

Our algorithm reduces problem (8) and (9) that has a large number of inequalities and dimension $O(n^2)$ of the feasible set W, described by (9), to a sequence of nonlinear programming subproblems that have a smaller number of nonlinear inequalities ($O(n)$) and dimension $O(n)$.

Let us consider the algorithm in details.

We assume here that spheres $S_q^0 \equiv S_q(0)$ of radius r_q and the center point $v_q = (x_q, y_q, z_q)$, circumscribed around each non-translated and non-rotated non-convex polyhedron $\mathbb{Q}_q$, $q \in J_N$, as well as, spheres $S_i^0 \equiv S_i(0)$ of radius r_i and

the center point $v_{ci} = (x_{ci}, y_{ci}, z_{ci})$ circumscribed around each non-translated and non-rotated convex polyhedron K_i^0, $i \in I_n$, are constructed.

The COMPOLY algorithm is an iterative procedure and involves the following steps:

Step 1 Let $k = 1$. Take the vector $\varsigma^{k-1} = \left(p^{k-1}, u_1^{k-1}, \ldots, u_N^{k-1}\right)$ of feasible placement parameters of polyhedra $\mathbb{Q}_q$, $q \in J_N$, within the container Ω^{k-1}.

Step 2 Derive the appropriate vector $\left(v_{c1}^{(k-1)}, .., v_{cn}^{(k-1)}\right)$ of center points of spheres $S_i\left(u_{a_i}^{(k-1)}\right)$, $i \in I_n$. With respect to the gluing vector **a**, the center point v_{ci} of $S_i \supset K_i$ after translation and rotation of initial convex polygon K_i^0 takes the form $v_{ci}^{(k-1)} = v_{ci}\left(u_{a_i}^{(k-1)}\right) = v_{a_i}^{(k-1)} + M\left(\theta_{a_i}^{(k-1)}\right) \cdot v_{ci}$.

Step 3 For each sphere $S_i\left(u_{a_i}^{(k-1)}\right)$ we construct a fixed individual container $\Omega_i^k \supset S_i \supset K_i$ with equal half-sides of length $r_i + \varepsilon$, $i \in I_n$, and the center of symmetry point $v_{ci}^{(k-1)}$, assuming $\varepsilon = \sum_{i=1}^{n} r_i/n$.

Step 4 Move each sphere S_i, associated with the convex polyhedron K_i, within the appropriate fixed individual container Ω_i^k (found at Step 3). Hence, for each sphere S_i we construct a phi-function $\Phi^{S_i\Omega_i^*}$ for sphere S_i and $\Omega_i^{k*} = R^3 \setminus \operatorname{int}\Omega_i^k$ in the form:

$$\Phi^{S_i\Omega_i^{k*}}\left(v_{a_i}, v_{a_i}^{(k-1)}\right) = \min\Big\{ - x_{ci}\left(u_{a_i}^{(k-1)}\right) + x_{a_i} + \varepsilon, -y_{ci}\left(u_{a_i}^{(k-1)}\right) + y_{a_i} + \varepsilon,$$

$$-z_{ci}\left(u_{a_i}^{(k-1)}\right) + z_{a_i} + \varepsilon, x_{ci}\left(u_{a_i}^{(k-1)}\right) - x_{a_i} + \varepsilon,$$

$$y_{ci}\left(u_{a_i}^{(k-1)}\right) - y_{a_i} + \varepsilon, z_{ci}\left(u_{a_i}^{(k-1)}\right) - z_{a_i} + \varepsilon\Big\}.$$

The inequality $\Phi^{S_i\Omega_i^{k*}}\left(v_{a_i}, v_{a_i}^{(k-1)}\right) \geq 0$ provides $S_i \subset \Omega_i^k$ and can be described by the following inequality system of six linear "ε-constraints":

$$- x_{ci}\left(u_{a_i}^{(k-1)}\right) + x_{a_i} + \varepsilon \geq 0, -y_{ci}\left(u_{a_i}^{(k-1)}\right) + y_{a_i} + \varepsilon \geq 0,$$

$$- z_{ci}\left(u_{a_i}^{(k-1)}\right) + z_{a_i} + \varepsilon \geq 0, \quad x_{ci}\left(u_{a_i}^{(k-1)}\right) - x_{a_i} + \varepsilon \geq 0,$$

$$y_{ci}\left(u_{a_i}^{(k-1)}\right) - y_{a_i} + \varepsilon \geq 0, z_{ci}\left(u_{a_i}^{(k-1)}\right) - z_{a_i} + \varepsilon \geq 0.$$

Now we introduce an auxiliary subset Λ_k^{ε} of additional "ε-constraints" on the translation vectors $v_{a_i} = (x_{a_i}, y_{a_i}, z_{a_i})$, $i \in I_n$, of convex polyhedra K_i, $i \in I_n$:

$$\Lambda_k^{\varepsilon} = \Big\{ u \in R^{\sigma} : -x_{ci}\left(u_{a_i}^{(k-1)}\right) + x_{a_i} + \varepsilon \geq 0, -y_{ci}\left(u_{a_i}^{(k-1)}\right) + y_{a_i} + \varepsilon \geq 0,$$

$$- z_{ci}\left(u_{a_i}^{(k-1)}\right) + z_{a_i} + \varepsilon \geq 0, \;\; x_{ci}\left(u_{a_i}^{(k-1)}\right) - x_{a_i} + \varepsilon \geq 0, \;\; y_{ci}\left(u_{a_i}^{(k-1)}\right)$$

$$- y_{a_i} + \varepsilon \geq 0, z_{ci}\left(u_{a_i}^{(k-1)}\right) - z_{a_i} + \varepsilon \geq 0, \;\; i \in I_n \Big\}.$$

Then we add the inequality system of $6n$ additional linear "ε-constraints" that describe the subset Λ_k^{ε} to the inequality system that defines the feasible region $W$and obtain the k-th subregion $W_k = W \cap \Lambda_k^{\varepsilon}$.

It should be noted that the inequality system that describes the feasible subregion $W_k$in most cases involves $O(n^2)$ redundant phi-inequalities.

Step 5 To avoid the redundant phi-inequalities that describe $W_k$we form special index sets Ξ_1^k and Ξ_2^k that involve indexes of all pairs of objects that are associated with non-redundant non-overlapping and containment constraints, respectively.

To form index set Ξ_1^k we exclude from Ξ (7) indexes of all pairs of convex polyhedra where individual containers do not intersect each other:

$$\Xi_1^k = \left\{ (i, j) \in \Xi_1^{kS} : \varphi^{\Omega_i^k \Omega_j^k}\left(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}\right) < 0 \right\},$$

where

$$\Xi_1^{kS} = \left\{ (i, j) \in \Xi : \widehat{\Phi}^{S_{a_i} S_{a_j}}\left(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}\right) < 0 \right\},$$

$$\varphi^{\Omega_i^k \Omega_j^k}\left(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}\right) = \max\left\{ \varphi_{ij}^{s}\left(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}\right), s = 1, \ldots, 6 \right\},$$

$$\varphi_{ij}^{1}\left(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}\right) = \left(x_i^{(k-1)} - x_j^{(k-1)}\right) - R_{ij},$$

$$\varphi_{ij}^{2}\left(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}\right) = \left(y_i^{(k-1)} - y_j^{(k-1)}\right) - R_{ij},$$

$$\varphi_{ij}^{3}\left(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}\right) = \left(z_i^{(k-1)} - z_j^{(k-1)}\right) - R_{ij},$$

$$\varphi_{ij}^{4}\left(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}\right) = -\left(x_i^{(k-1)} - x_j^{(k-1)}\right) - R_{ij},$$

$$\varphi_{ij}^{5}\left(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}\right) = -\left(y_i^{(k-1)} - y_j^{(k-1)}\right) - R_{ij},$$

$$\varphi_{ij}^{6}\left(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}\right) = -\left(z_i^{(k-1)} - z_j^{(k-1)}\right) - R_{ij}, \quad R_{ij} = \left(r_i + r_j\right) + \rho_{ij} + 2\varepsilon,$$

$\widehat{\Phi}^{S_{a_i}S_{a_j}}\left(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}\right)$ is an adjusted phi-function for a pair of spheres S_q and S_g $(a_i = q, a_j = g)$, circumscribed around non-convex polyhedra $\mathbb{Q}_q\left(u_q^{(k-1)}\right) \supset K_i\left(u_q^{(k-1)}\right)$ and $\mathbb{Q}_g\left(u_g^{(k-1)}\right) \supset K_j\left(u_g^{(k-1)}\right)$.

We note that if $(i, j) \notin \Xi_1^k$, then we do not need to check the distance (or non-overlapping) constraint for the corresponding pair of polyhedra $K_i\left(u_{a_i}^{(k-1)}\right)$ and $K_j\left(u_{a_j}^{(k-1)}\right)$. If $\rho_{ij} = 0$, then function $\varphi^{\Omega_i^k\Omega_j^k}\left(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}\right)$ becomes a phi-function for two oriented parallelepipeds Ω_i and Ω_j.

As an example let us consider three polyhedra placed inside the container Ω^k at kth iteration of the COMPOLY algorithm (Figure 7).

For the example the index set Ξ defined by (7) has the form: $\Xi = \{(1,3),(1,4),(1,5),(1,6),(2,3),(2,4),(2,5),(2,6),(3,5),(3,6),(4,5),(4,6)\}$. First we define the index set Ξ_1^{kS} (Figure 7a): $\Xi_1^{kS} = \left\{(i,j) \in \Xi : \widehat{\Phi}^{S_{a_i}S_{a_j}}\left(v_1^{(k-1)}, v_2^{(k-1)}\right) < 0\right\} = \{(1,3),(1,4),(2,3),(2,4)\}$. It means that only spheres S_1^ρ and S_2^ρ for concave polyhedra $\mathbb{Q}_1$ and $\mathbb{Q}_2$ have nonempty intersection, i.e., $\widehat{\Phi}^{S_1S_2}\left(v_1^{(k-1)}, v_2^{(k-1)}\right) < 0$, and therefore it is sufficient to consider only possible intersection of convex polyhedra: K_1 and K_3, K_1 and K_4, K_2 and K_3, and K_2 and K_4.

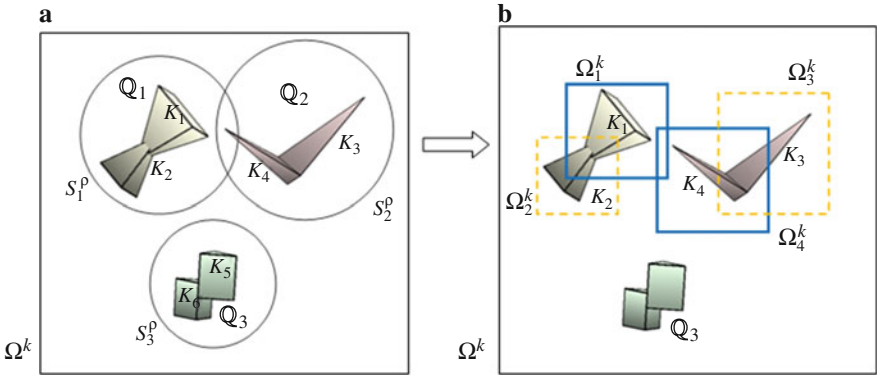


Figure 7 Construction of the index set Ξ_1^k at the k-th iteration of the COMPOLY algorithm: (**a**) step 1; (**b**) step 2

Then we form the index set Ξ_1^k (Figure 7b):

$$\Xi_1^k = \left\{(i, j) \in \Xi_1^{kS} : \varphi^{\Omega_i^k \Omega_j^k}\left(v_{a_i}^{(k-1)}, v_{a_j}^{(k-1)}\right) < 0\right\} = \{(1, 4)\}.$$

It means that only individual containers Ω_1^k and Ω_4^k for convex polyhedra K_1 and K_4 have nonempty intersection, i.e., $\varphi^{\Omega_1^k \Omega_4^k}\left(v_1^{(k-1)}, v_2^{(k-1)}\right) < 0$ and therefore we include in the subproblem only quasi-phi-function for K_1 and K_4.

To form index set Ξ_2^k we exclude from (9) all phi-inequalities for containment constraints of convex polyhedra where individual containers do not intersect the set $\Omega_\varepsilon^{k*} = R^3 \setminus \operatorname{int} \Omega_\varepsilon^k$, such that

$$\Omega_\varepsilon^k = \left\{(x, y, z, p) : \Psi(x, y, z, p) \geq 0, f^{(k-1)}(p, \varepsilon) \geq 0\right\}.$$

Thus, $\Xi_2^k = \left\{i \in \Xi_2^{kS} : \widehat{\Phi}^{\Omega_i^k \Omega_\varepsilon^*}\left(v_{a_i}^{(k-1)}\right) < 0\right\}$, where $\widehat{\Phi}^{\Omega_i^k \Omega_\varepsilon^*}\left(v_{a_i}^{(k-1)}\right)$ is an adjusted phi-function for a polyhedron $K_i\left(u_{a_i}^{(k-1)}\right)$ and the object Ω_ε^{k*}, $\Xi_2^{kS} = \left\{i \in I_n : \widehat{\Phi}^{S_{a_i} \Omega_\varepsilon^{k*}}\left(v_{a_i}^{(k-1)}\right) < 0\right\}$, $\widehat{\Phi}^{S_{a_i} \Omega_\varepsilon^{k*}}$ is an adjusted phi-function for a sphere S_q, associated with non-convex polyhedron $\mathbb{Q}_q\left(u_q^{(k-1)}\right) \supset K_i\left(u_q^{(k-1)}\right)$, and the object Ω_ε^*, $a_i = q$.

We note that if $i \notin \Xi_2^k$, then we do not need to check the distance or containment constraint for the polyhedron $K_i\left(u_{a_i}^{(k-1)}\right)$ and the object Ω_ε^k.

Step 6 Generate the k-th subproblem on solution subset $W_k = W \cap \Lambda_k^\varepsilon$ with deleted redundant phi-function inequalities and reduced dimension (O(n)):

$$\min F\left(u_{w_k}\right) \quad \text{s.t. } u_{w_k} \in W_k \subset R^{\sigma - \sigma_k}, \tag{14}$$

$$W_k = \left\{u_{w_k} = \left(\varsigma, \tau_{w_k}\right) \in R^{\sigma - \sigma_k} : \widehat{\Phi}'_{ij}\left(u_{a_i}, u_{a_j}\right) \geq 0, (i, j) \in \Xi_1^k,\right.$$
$$\left.\widehat{\Phi}_i\left(u_{a_i}, p\right) \geq 0, i \in \Xi_2^k, \Phi^{S_i \Omega_i^{k*}}\left(u_{a_i}\right) \geq 0, i \in I_n, f^{(k-1)}(p, \varepsilon) \geq 0\right\},$$

where Ξ_1^k and Ξ_2^k are defined in Step 5, and $\sigma_k = 3\left(m - \operatorname{card}\left(\Xi_1^k\right)\right)$ is the number of all deleted auxiliary variables meeting in the appropriate redundant phi-function inequalities, $\sigma - \sigma_k = |p| + 6N + \operatorname{card}\left(\Xi_1^k\right)$, $\operatorname{card}\left(\Xi_1^k\right)$ is (O(n)).

Step 7 Generate a feasible starting point $u^{(k-1)} = \left(\varsigma^{(k-1)}, \tau_{w_k}^{(k-1)}\right)$ for problem (14). Since a vector $\varsigma^{(k-1)}$ is already defined, we need to find values of the

vector of auxiliary variables $\tau_{w_k}^{(k-1)} = \left(u_P^{(k-1)1}, \ldots, u_P^{(k-1)s}, \ldots, u_P^{(k-1)m}\right)$ for such $s \in \{1, \ldots, m\}$ that $(i, j) \in \Xi_1^k$.

To derive a vector $u_{P(k-1)s}$ we employ the FAPA algorithm. The key idea of the FAPA algorithm lies in the following: we derive a vector $u_{P(k-1)s}$ as a vector of feasible parameters of a separating plane for two spheres $S_i\left(u_{a_i}^{(k-1)}\right)$ and $S_j\left(u_{a_j}^{(k-1)}\right)$ if $\widehat{\Phi}^{S_i S_j} \geq 0$, using simple geometrical calculations; otherwise, we find a vector $u_P^{(k-1)s}$, solving the following auxiliary subproblem:

$$\max \alpha \quad \text{s.t.} \quad \left(u_P^s, \alpha\right) \in W'_\alpha, \tag{15}$$

where

$$W'_\alpha = \left\{\left(u_P^s, \alpha\right) \in R^4 : \widehat{\Phi}'_{ij}\left(u_{a_i}^{(k-1)}, u_{a_j}^{(k-1)}, u_P^s\right) - \alpha \geq 0\right\},$$

$\alpha \in R^1$, $u_P^s = \left(\theta_P^{1s}, \theta_P^{2s}, \alpha_P^s\right)$, with fixed parameters $\left(u_{a_i}^{(k-1)}, u_{a_j}^{(k-1)}\right)$ involving in the appropriate adjusted quasi-phi-function $\widehat{\Phi}'_{ij}\left(u_{a_i}^{(k-1)}, u_{a_j}^{(k-1)}, u_P^s\right)$, $\forall (i,j) \in \Xi$. It should be noted that $\alpha \geq 0$ in (15) provides feasible values of u_P^s.

Figure 8 illustrates two cases to derive a vector of feasible parameters u_P of a separating plane: (a) for two spheres S_i and S_j if int$S_i \cap$ int $S_j = \varnothing$; for two convex polyhedra K_i and K_j if int$S_i \cap$ int $S_j \neq \varnothing$.

For case (a) we use simple geometrical calculations to find u_P (Figure 8a); for case (b) we solve the NLP subproblem (15) to find a nonnegative value of α that corresponds to the problem of searching for a nonnegative value of a quasi-phi-function of two convex polyhedra K_i and K_j (Figure 8b).

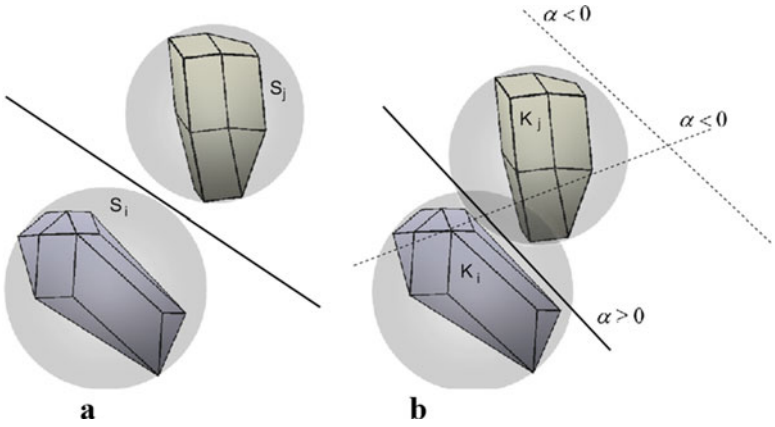


Figure 8 Deriving a vector of feasible parameters u_P: (**a**) $S_i \cap S_j = \varnothing$; (**b**) int$S_i \cap$ int $S_j \neq \varnothing$

Thus, all adjusted quasi-phi-functions and phi-functions in (14) at the point $u^{(k-1)}$ take nonnegative values.

Step 8 Solve subproblem (14), starting from the feasible point $u^{(k-1)}$

$$\min F\left(u_{w_k}\right) \quad \text{s.t. } u_{w_k} \in W_k \subset R^{\sigma-\sigma_k} \tag{16}$$

and get a local minimum point $u^*_{w_k} = \left(\varsigma^{*k}, \tau^{*k}_{w_k}\right)$.

If the point $u^*_{w_k}$ of local minimum of subproblem (16) belongs to the frontier of an auxiliary subset Λ^{ε}_k, i.e., $u^*_{w_k} \in fr\,\Lambda^{\varepsilon}_k$, then we take ς^{*k} as a starting vector ς^k for the next iteration of the procedure (set $k = k + 1$ and go to Step 2), otherwise we stop the optimization procedure.

We claim that the point $u^* = u^{*k} = (\varsigma^{*k}, \tau^{*k}) \in R^{\sigma}$ is a point of local minimum of problem (8) and (9), where τ^{*k} involves $\tau^{*k}_{w_k}$ and auxiliary variables that are deleted at the k-th iteration. Note that the σ_k previously deleted auxiliary variables can be redefined by FAPA algorithm. However we do not need to redefine the deleted auxiliary variables at the last step of the algorithm, since the values of auxiliary variables have no effect on the value of the objective function, i.e., $F\left(u^*_{w_k}\right) = F\left(u^{*k}\right)$.

Figure 9 shows the diagram of the COMPOLY procedure to solve problem (8) and (9). We illustrate the procedure of solving a sequence of subproblems, given by (16), for $k = 2,3,4$. Note, that feasible starting point $u^{(0)}$ is found by algorithm FPPA. Each auxiliary (artificial) set Λ^{ε}_k, described at Step 4 of the COMPOLY procedure, is shown as a square with the center point $u^{(k-1)}$, $k = 1,2,3,4$.

We take the feasible point $u^{(0)}$, form set Λ^{ε}_1 with the center point $u^{(0)}$, solve subproblem (16) in the subregion $W_1 = \Lambda^{\varepsilon}_1 \cap W$, and get a local minimum point $u^*_{w_1}$. The point $u^*_{w_1}$ belongs to the frontier of set Λ^{ε}_1; therefore, we form the next set Λ^{ε}_2 with the center point $u^{(1)} = u^*_{w_1}$ and search for a local minimum point $u^*_{w_2}$ of subproblem (16) on subregion $W_2 = \Lambda^{\varepsilon}_2 \cap W$. The point $u^*_{w_2}$ belongs to the frontier of set Λ^{ε}_2; therefore, we form the next set Λ^{ε}_3 with the center point $u^{(2)} = u^*_{w_2}$ and search for a local minimum point $u^*_{w_3}$ of subproblem (16) on subregion $W_3 = \Lambda^{\varepsilon}_3 \cap W$. The point $u^*_{w_3}$ belongs to the interior of set Λ^{ε}_3, i.e., $u^*_{w_3} \in \text{int}\,\Lambda^{\varepsilon}_3$; therefore, we stop our procedure. The point $u^*_{w_3} = u^*$ is a point of local minimum of problem (8) and (9).

Figure 10 illustrates the iterative procedure of packing concave polyhedra related to the diagram shown in Figure 9.

We note that $\text{dist}(u^*_{w_k}, u^*_{w_{k+1}}) \geq \varepsilon$, if $u^*_{w_{k+1}} \in fr\,\Lambda^{\varepsilon}_k$, and we take the value of ε that is considerably greater than the accuracy of IPOPT (10^{-8}). Thus, we can conclude that the stopping condition of the COMPOLY procedure is always reached in a finite number of iterations.

If the IPOPT program fails to find a local minimum of subproblem (14), then we halve the value of ε and start up the COMPOLY procedure. If a local minimum is found under the half value of ε, then we recover the initial value of epsilon and

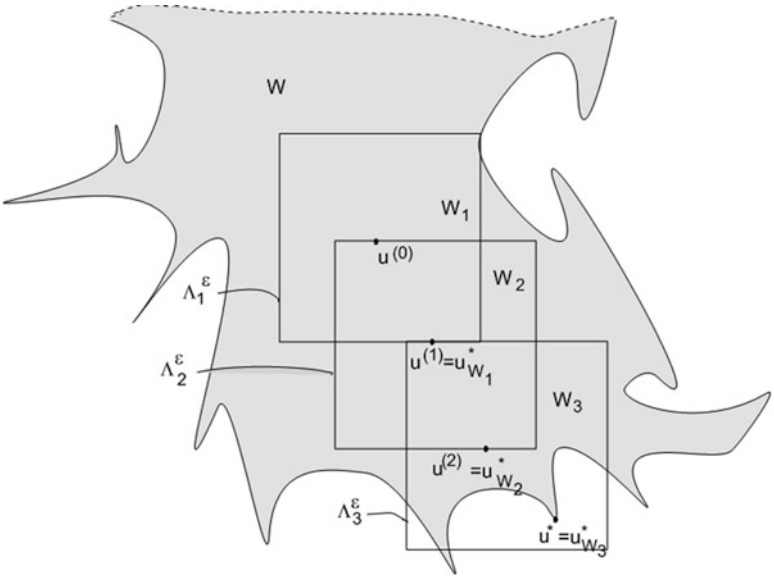


Figure 9 Diagram of the COMPOLY procedure

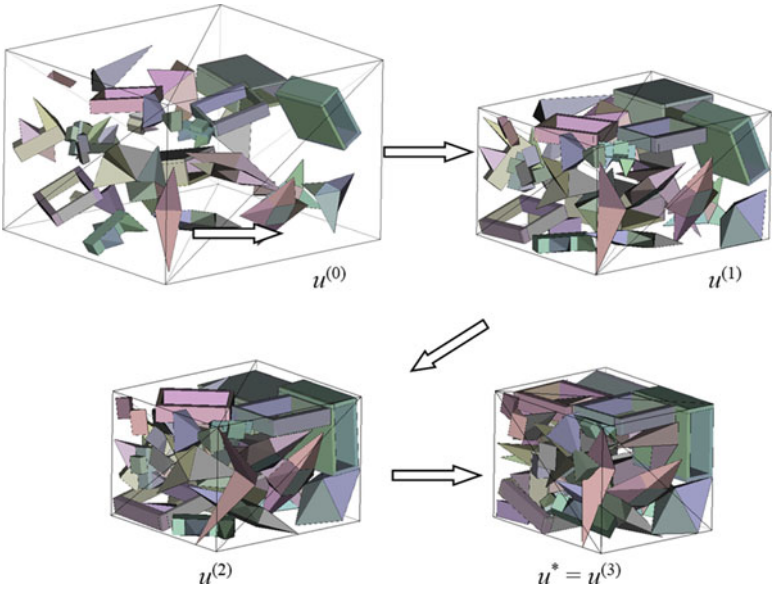


Figure 10 Arrangements of non-convex polyhedra, corresponding to the sequence of feasible points $u^{(0)}$, $u^{(1)}$, $u^{(2)}$, $u^* = u^{(3)}$

continue the COMPOLY procedure for a new feasible starting point, otherwise we terminate the procedure.

Our algorithm, in most cases, takes consideration of significantly fewer pairs of polyhedra than m (here m is the number of all pairs of convex polyhedra considered in problem (8) and (9)), because for each non-convex polyhedron only its "ε-neighbors" have to be monitored. It should be noted that the algorithm is not efficient for special cases when all objects are "ε-neighbors."

Thus the COMPOLY algorithm allows us to reduce problem (8) and (9) with a large number of inequalities and dimension $O(n^2)$ of the feasible set W, described by (9), to a sequence of subproblems (14) with a smaller number of nonlinear inequalities and dimension $O(n)$ of solution subset W_k.

Remark: To reduce the dimension of subproblem (16) for large problem instances ($N > 100$) we fix the values of auxiliary variables $\tau_{w_k}^{(k-1)}$ at Step 8.

6 Computational Experiments

We present a number of examples to demonstrate the efficiency of our methodology. We have run all experiments on an AMD Athlon 64 X2 5200+ computer, programming language C++, Windows 7 OS. For local optimization we use the IPOPT code (https://projects.coin-or.org/Ipopt) by means of program interface using the default options. In the following examples $\varepsilon = 5$ for the COMPOLY procedure.

First we compare our results to those given in Stoyan et al. [99, 100]. We search for locally optimal solutions using the compaction algorithm: (a) starting from a feasible point generated by FPPA described in Sect. 5.1 and (b) starting from a feasible point found by the algorithm developed in [99, 100]. We introduce the comparison in [90].

Example 1 Figure 11 shows the locally optimal placement of the collection of $N = 80$ convex polyhedra from [90]. The optimized container has dimensions and volume: (a) (l^*, w^*, h^*)=(43.4338, 41.8435, 45.0059) and $F(u^*)$=81795.2169, starting from the feasible point found by FPPA (Figure 11a). Computational time is 46035.78 s; (b) (l^*, w^*, h^*)=(36.3569, 40.8764, 56.2557) and $F(u^*)$=83604.0544, starting from the feasible point found by the algorithm given in [100] (Figure 11b). The computational time is 42950.4 s. Improvement of the value of objective function in comparison to the result given in [100]: (a) 27.88%; (b) 26.29%.

Example 2 Figure 12 shows the locally optimal placement of the collection of $N = 40$ non-convex polyhedra from [90]. The optimized container has dimensions and volume: (a) (l^*, w^*, h^*)=(34.9974, 36.9655, 43.2777) and $F(u^*)$=55988.4619, starting from the feasible point found by FPPA (Figure 12a); (b) (l^*, w^*, h^*)=(31.1419, 30.8086, 55.4061) and $F(u^*)$=53158.8838, starting from the feasible point found by the algorithm given in [99] (Figure 12b). We use three

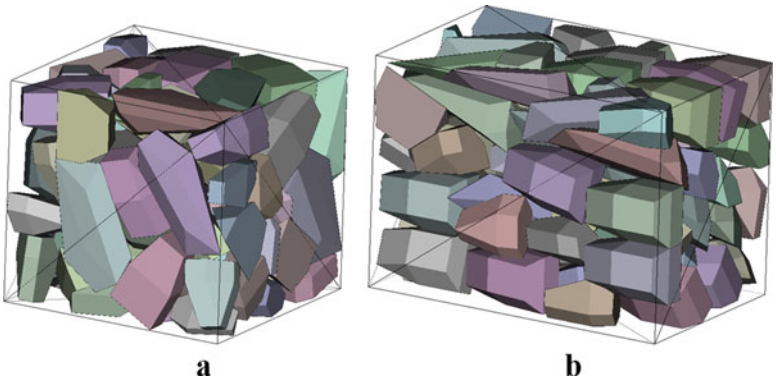


Figure 11 Locally optimal placement of polyhedra in Example 1: (**a**) starting from the feasible point found by FPPA; (**b**) starting from the feasible point found by the algorithm given in [100]

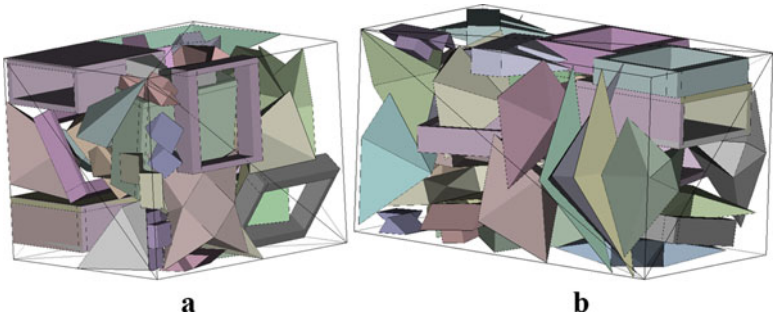


Figure 12 Locally optimal placement of polyhedra in Example 2: (**a**) starting from the feasible point found by FPPA; (**b**) starting from the feasible point found by the algorithm given in [99]

starting points, and set the time limit of 10 h. Improvement of the value of objective function in comparison to the result given in [100]: (a) 15.64%; (b) 19.91%.

Table 1 lists some examples presented in [95]. For each example the minimum volume of the container found by our method is smaller than the best solution reported in [95].

To show the effectiveness of the COMPOLY procedure, further tests were performed. In the example for $N = 10$ non-convex polyhedra from [90], the average computational time per one local extremum is: (a) 1380 s without the use of the COMPOLY procedure; (b) 283 s using the COMPOLY procedure. The number of variables and inequalities are: (a) 1791 and 7934 without the use of the COMPOLY procedure; 626 and 3086 using the COMPOLY procedure at the last iteration. In the example for $N = 20$ non-convex polyhedra from [90], the average computational time per local extremum is: (a) 75026.31 s without the use of the COMPOLY procedure; (b) 4980.74 s using the COMPOLY procedure. The number of variables and inequalities are: (a) 7471 and 30916 without the use

Table 1 Comparison of our results to those in [95]

Problem	The best volume in [95]	The best time (s) in [95]	Found by FPPA[a] + Compoly volume	Found by FPPA[a] + Compoly time (s)	Found by [99][b] + compoly volume	Found by [99][b] + compoly time (s)
20 in [99]	32550	26202.1	27432.64	34313.34	28106.64	5360.67
30 in [99]	48300	53741.5	41646.17	35289.34	40277.19	33008.89
40 in [99]	61950	99952.0	53158.88	201501.5	55988.46	195051.5
50 in [99]	77280	125210.6	68214.56	215144.5	61630.68	270654.8
36 in [95]	12480	9637.5	10461.67	23023.12	–	–

[a]A starting feasible point found by FPPA
[b]A starting feasible point found by algorithm found in [99]

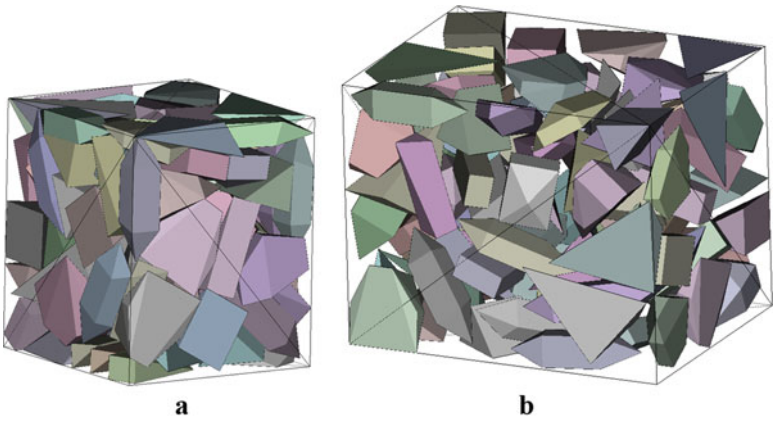


Figure 13 Locally optimal placement of polyhedra in Example 3: (**a**) $\rho = 0$; (**b**) $\rho = 1.5$

of the COMPOLY procedure; 1334 and 8028 using the COMPOLY procedure at the last iteration. In the example for $N = 30$ non-convex polyhedra from [90], a local minimum has not been found within the time limit of 72 h without using of the COMPOLY procedure. The average computational time per local extremum is 35289.34 s using the COMPOLY procedure. Further results reported in [90] are provided.

Example 3 We generate a collection of $n = 98$ convex polyhedra, consisting of the seven types of polyhedra from Example 1 given in [99]. We include 14 of each type of polyhedra. Figure 13 shows the locally optimal placement of the collection of convex polyhedra. The container has dimensions and volume: (a) $(l^*, w^*, h^*) =$ (30.9324, 28.1897, 26.5064) and $F(u^*) = 23113.06$ with $\rho = 0$ (Figure 13a). One starting point is used. Computational time is 147967.3 s; (b) (l^*, w^*, h^*)=(41.3510, 33.0721, 31.7988) and $F(u^*)$=43487.0040 with $\rho = 1.5$ (Figure 13b). One starting point is used. Computational time is 48152.79 s.

Example 4 We generate a collection of $N = 20$ non-convex polyhedra, consisting of the two types of polyhedra given in [99]. We include ten of each type of polyhedra. Figure 14 shows the locally optimal placement of the collection of non-convex polyhedra. The container has dimensions and volume: (a) (l^*, w^*, h^*) =(26.3522, 23.7514, 24.4055) and $F(u^*) = 15275.4815$ with $\rho = 0$ (Figure 14a). Two starting points are used. Computational time is 8729.45 s; (b) (l^*, w^*, h^*)=(26.5890, 26.5239, 36.1706) and $F(u^*)$=25509.2576 with $\rho = 1.5$. Ten starting points are used. Computational time is 24696.46 s (Figure 14b).

Now we present our new results.

Example 5 We consider homothetic convex polyhedra of two types: (a) four vertices; (b) sixteen vertices. Figure 15 shows the locally optimal placement of collection of: (a) 300 convex polyhedra; (b) 400 convex polyhedra. The con-

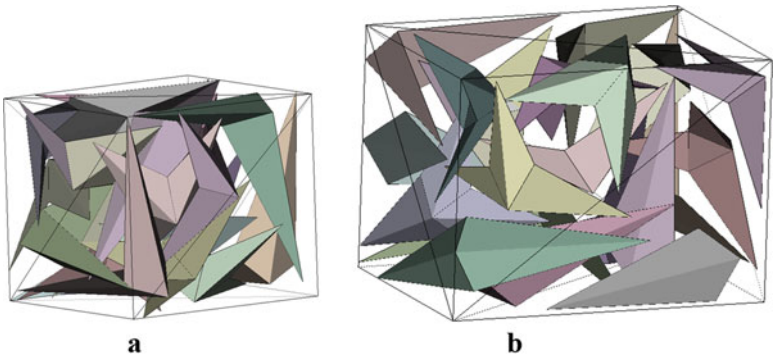


Figure 14 Locally optimal placement of polyhedra in Example 4: (**a**) $\rho = 0$; (**b**) $\rho = 1.5$

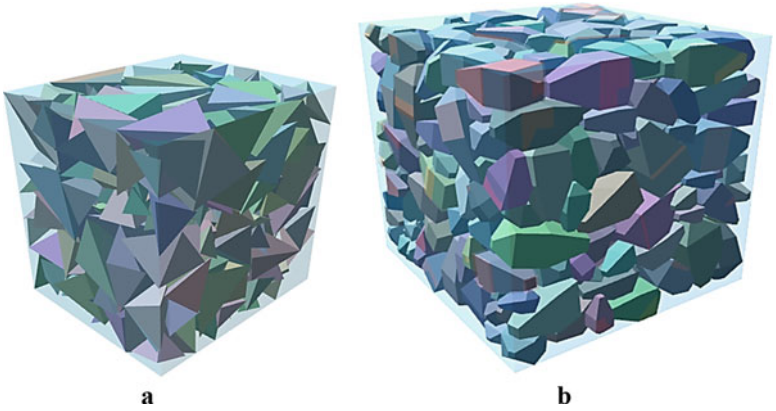


Figure 15 Locally optimal placement of polyhedra in Example 5: (**a**) $n = 300$; (**b**) $n = 400$

tainer has dimensions and volume: (a) (l^*, w^*, h^*)=(8.3164, 7.9712, 3.5752) and $F(u^*)$=122.60 (Figure 15a). One starting point is used. Computational time is 14 h; (b) (l^*, w^*, h^*)=(132.2662, 139.0069, 127.7707) and $F(u^*)$=2349181.1577 (Figure 15b). One starting point is used. Computational time is 30 h.

Example 6 We consider $n = 80$ convex polyhedra from [90]. Figure 16a shows the locally optimal placement of the collection of convex polyhedra. The sphere has radius: $r^* = 27.327$. Ten starting points are used. Computational time is 43613.42 s.

Example 7 We consider $N = 30$ non-convex polyhedra from [90]. Figure 16b shows the locally optimal placement of the collection of non-convex polyhedra. The sphere has radius: $r^* = 23.0709$. Ten starting points are used. Computational time is 29491.35 s.

Example 8 We consider $n = 80$ convex polyhedra from [90]. Figure 17a shows the locally optimal placement of the collection of convex polyhedra. The cylinder has

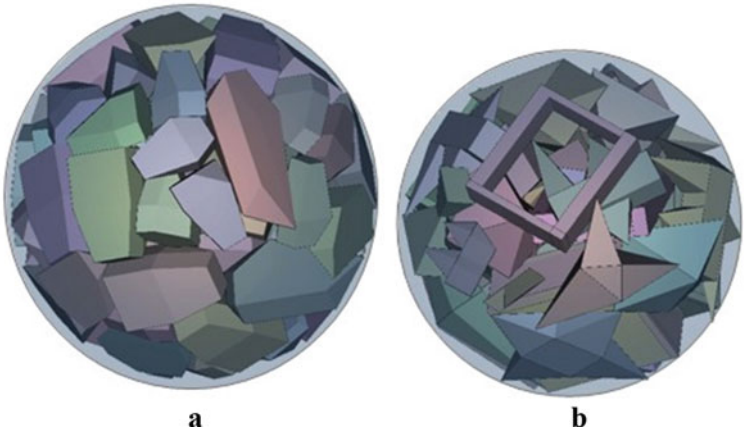


Figure 16 Locally optimal placement of polyhedra: (**a**) in Example 6; (**b**) in Example 7

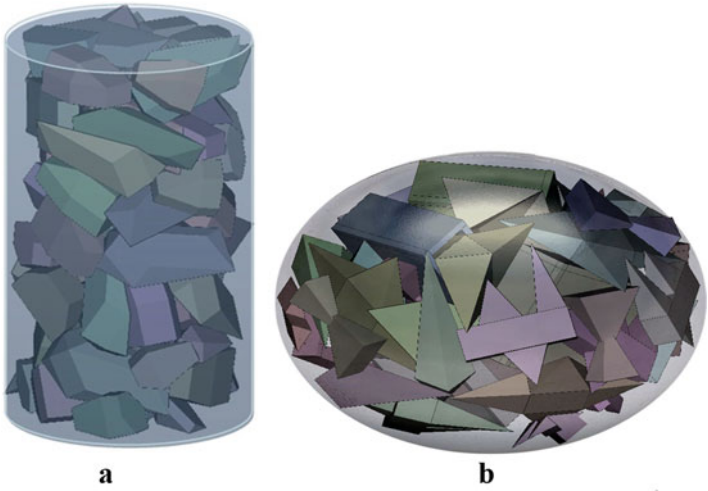


Figure 17 Locally optimal placement of polyhedra: (**a**) in Example 8; (**b**) in Example 9

dimensions: $r^* = 0.25$, $h^* = 17.8744$. Ten starting points are used. Computational time is 43613.42 s.

Example 9 We consider $N = 40$ non-convex polyhedra from [90]. Figure 17b shows the locally optimal placement of the collection of non-convex polyhedra. The ellipsoid has dimensions: $a^* = 28.3608$, $b^* = 24.8157$, and $c^* = 21.270$. Ten starting points are used. Computational time is 29491.35 s.

Example 10 We consider $n = 40$ convex polyhedra from [90]. The container is an intersection of two spheres with centers at points $(-10,0,0)$ and $(10,0,0)$ of identical variable radius r. Figure 18a shows the locally optimal placement of the collection of

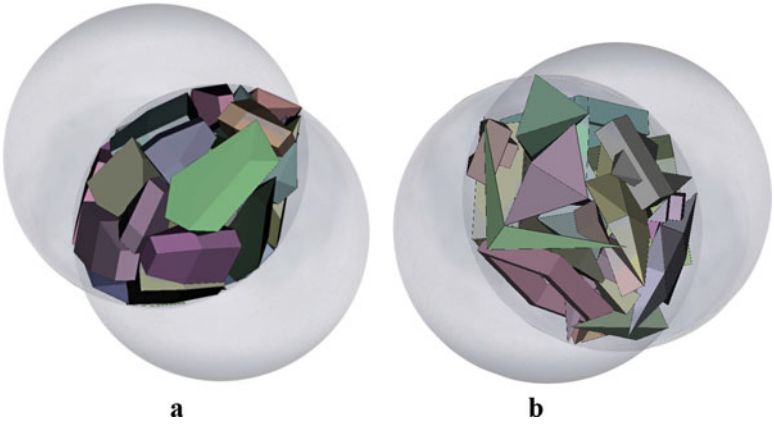


Figure 18 Locally optimal placement of polyhedra: (**a**) in Example 10; (**b**) in Example 11

Figure 19 Locally optimal placement of convex polyhedra in Example 12

convex polyhedra. The value of the objective function is $r^* = 23.4802$. Two starting points are used. Computational time is 3730.34 s.

Example 11 We consider $N = 20$ non-convex polyhedra from [90]. The container is defined in Example 10. Figure 18b shows the locally optimal placement of the collection of non-convex polyhedra. The value of the objective function is $r^* = 26.376$. One starting point is used. Computational time is 8658.87 s.

Example 12 We consider $n = 98$ convex polyhedra from [90] with given weights. Figure 19 shows the locally optimal placement of the collection of convex polyhedra. The objective is to minimize both radius r of a sphere and deviation d of the collection of polyhedra from the center point of the sphere: $F(u^*) = \kappa_1 r^* + \kappa_2 d^*$, $r^* = 18.0409$, $d^* = 0$. One starting point is used. Computational time is 26150.23 s.

7 Conclusions

Two chapters of this volume (written by us, with co-authors) are devoted to packing problems in space engineering and logistics. The present chapter, representing the first part of the discussion, introduces the subject from quite a general point of view, underlining the special features of the context. Packing problems in space are generally characterized by the presence of complex geometries, relevant both to the items and the container to deal with, quite often subject to additional accommodation rules and overall conditions, such as *static* and *dynamic* balancing.

As examples of challenging exercises, a recent study, concerning the packing of tetris-like items into a convex domain, with additional conditions is outlined, as well as a real-world cargo accommodation issue in space. The underlying contention of these two chapters is that a modeling-based heuristic approach (as opposed to pure algorithmic ones) is particularly efficient in this application framework. The specific problem of packing polyhedra into convex containers of minimum size has been discussed here in depth, as an issue of interest in the space context (albeit not specific to it).

Different shapes of containers, such as spheres, cylinders, polytopes, spheroids, and ellipsoids, with (minimum-size container) optimization objectives and additional conditions, such as minimum allowable distances, fixed orientations, and forbidden zones, are investigated.

The concept of quasi-phi-functions has been used to represent non-overlapping constraints, while phi-functions have been used to deal with containment constraints. Adjusted phi-functions and adjusted quasi-phi-functions have further been used for the analytical description of relative-distance conditions. An exact model for the specific problem in question has been formulated in terms of continuous nonlinear programming. An ad hoc solution algorithm involves a fast starting point strategy and a variant of the COMPOLY procedure to search for "good" locally optimal extrema reducing the computational costs dramatically. The proposed approach offers significant improvements to the already existing technology, in terms of computational effort and solution quality. A similar overall methodology is adopted to solve packing problems with balancing conditions. This is the topic of the second part of the present discussion, as presented in the related chapter of this volume.

Appendix: Phi-Functions and Quasi-Phi-Functions

Phi-Functions

Phi-functions allow us to distinguish among the following three cases: sets A and B are intersecting so that A and B have common interior points; A and B do not have

common points; A and B are in contact, i.e., A and B have only common boundary points.

Let $A \subset R^3$ and $B \subset R^3$ be two objects. The sizes of objects can change according to homothetic coefficients (scaling parameters) λ_A, $\lambda_B > 0$. The position of object A is defined by a vector of *placement parameters* (v_A, θ_A), where $v_A = (x_A, y_A, z_A)$ is a translation vector and $\theta_A = \left(\theta_A^1, \theta_A^2, \theta_A^3\right)$ is a vector of rotation angles. We denote the vector of variables for the object A by $u_A = (v_A, \theta_A, \lambda_A)$ and the vector of variables for the object B by $u_B = (v_B, \theta_B, \lambda_B)$. The object A, rotated by angles $\theta_A^1, \theta_A^2, \theta_A^3$, translated by vector v_A, and rescaled by homothetic coefficient λ_A,will be denoted by $A(u_A)$.

Definition A1 A continuous function $\Phi^{AB}(u_A, u_B)$ is called a phi-function for objects $A(u_A)$ and $B(u_B)$ if

$\Phi^{AB} > 0$, if $A(u_A) \cap B(u_B) = \varnothing$;
$\Phi^{AB} = 0$, if $\text{int}A(u_A) \cap \text{ int } B(u_B) = \varnothing$ and $frA(u_A) \cap frB(u_B) \neq \varnothing$;
$\Phi^{AB} < 0$, if $\text{int}A(u_A) \cap \text{ int } B(u_B) \neq \varnothing$;

provided that λ_A, $\lambda_B$are fixed.

Here *frA* means the boundary (frontier) and *intA* means the interior of object A.

Thus, inequality $\Phi^{AB} \geq 0$ represents the *non-overlapping relationship* $\text{int}A(u_A) \cap \text{ int } B(u_B) = \varnothing$, i.e., $\Phi^{AB} \geq 0 \iff \text{ int } A(u_A) \cap \text{ int } B(u_B) = \varnothing$.

We use phi-functions for the description of the *containment relation* $A \subseteq B$ as follows: $\Phi^{AB^*} \geq 0$, where $B^* = R^3 \setminus \text{int } B$.

We emphasize that according to Definition 1, the phi-function Φ^{AB} for a pair of objects A and B can be constructed by many different formulas [55], and we can choose the most convenient ones for our optimization algorithms.

We can take into account *minimum allowable distance constraints* by replacing the phi-functions in the *non-overlapping* and *containment constraints* with adjusted phi-functions.

Let $\rho > 0$ be a given minimum allowable distance between objects $A(u_A)$ and $B(u_B)$.

Definition A2 A continuous and everywhere defined function $\widehat{\Phi}^{AB}(u_A, u_B)$ is called an adjusted phi-function for objects $A(u_A)$ and $B(u_B)$, if

$\widehat{\Phi}^{AB} > 0$, if $\text{dist}(A, B) > \rho$; $\widehat{\Phi}^{AB} = 0$, if $\text{dist}(A, B) = \rho$;
$\widehat{\Phi}^{AB} < 0$, if $\text{dist}(A, B) < \rho$.

Now we can describe the distance constraint for objects $A(u_A)$ and $B(u_B)$ in the form: $\widehat{\Phi}^{AB} \geq 0 \iff \text{dist}(A,B) \geq \rho$.

Quasi-Phi-Functions

We introduce a function $\Phi'^{AB}(u_A, u_B, u')$ that must be defined for all values of u_A and u_B. In addition to the placement parameters of objects used with phi-functions, quasi-phi-functions depend on auxiliary variables u'. These extra variables u' take values in some domain $U \subset R^{\eta}$. The number and the nature of variables u' depend on the shapes of objects $A(u_A)$ and $B(u_B)$, as well as on the restrictions of a layout problem.

Definition A3 A continuous function $\Phi'^{AB}(u_A, u_B, u')$ is called *a quasi-phi-function* for two objects $A(u_A)$ and $B(u_B)$ if $\max_{u' \in U} \Phi'^{AB}\left(u_A, u_B, u'\right)$ is a phi-function for the objects.

The main property of a quasi-phi-function is the following:

- if $\Phi'^{AB}(u_A, u_B, u') \geq 0$ for some u', then int$A(u_A) \cap$ int $B(u_B) = \varnothing$,

where $\Phi'^{AB}(u_A, u_B, u')$ is a quasi-phi-function for two objects $A(u_A)$ and $B(u_B)$.

Let $\rho > 0$ be a given minimum allowable distance between objects $A(u_A)$ and $B(u_B)$.

Definition A4 Function $\widehat{\Phi}'^{AB}\left(u_A, u_B, u'\right)$ is called an adjusted quasi-phi-function for objects $A(u_A)$ and $B(u_B)$, if function $\max_{u'} \widehat{\Phi}'^{AB}\left(u_A, u_B, u'\right)$ is an adjusted phi-function for the objects.

We can define the distance constraint for objects $A(u_A)$ and $B(u_B)$ in the form: $\widehat{\Phi}'^{AB} \geq 0$. The inequality implies dist(A,B)$\geq\rho$.

References

1. Fasano, G., Pintér, J.D.: Optimized Packings and their Applications. Springer Optimization and its Applications. Springer, New York (2015)
2. Cagan, J., Shimada, K., Yin, S.: A survey of computational approaches to three-dimensional layout problems. Comput. Aided Des. **34**, 597–611 (2002)
3. Dyckhoff, H., Scheithauer, G., Terno, J.: Cutting and Packing. In: Dell'Amico, M., Maffioli, F., Martello, S. (eds.) Annotated Bibliographies in Combinatorial Optimization, pp. 393–412. Wiley, Chichester (1997)
4. Ibaraki, T., Imahori, S., Yagiura, M.: Hybrid Metaheuristics for Packing Problems. In: Blum, C., Aguilera, M.J., Roli, A., Sampels, M. (eds.) Hybrid Metaheuristics: an Emerging Approach to Optimization. Studies in Computational Intelligence (SCI), vol. 114, pp. 185–219. Springer, Berlin (2008)
5. Faroe, O., Pisinger, D., Zachariasen, M.: Guided local search for the three-dimensional bin packing problem. INFORMS J. Comput. **15**(3), 267–283 (2003)
6. Fekete, S., Schepers, J.: A combinatorial characterization of higher-dimensional orthogonal packing. Math. Oper. Res. **29**, 353–368 (2004)

7. Fekete, S., Schepers, J., van der Veen, J.C.: An exact algorithm for higher-dimensional orthogonal packing. Oper. Res. **55**(3), 569–587 (2007)
8. Martello, S., Pisinger, D., Vigo, D.: The three-dimensional bin packing problem. Oper. Res. **48**(2), 256–267 (2000)
9. Martello, S., Pisinger, D., Vigo, D., Den Boef, E., Korst, J.: Algorithms for general and robot-packable variants of the three-dimensional bin packing problem. ACM Trans. Math. Softw. **33**(1), 7 (2007)
10. Pisinger, D.: Heuristics for the container loading problem. Eur. J. Oper. Res. **141**(2), 382–392 (2002)
11. Addis, B., Locatelli, M., Schoen, F.: Efficiently packing unequal disks in a circle: a computational approach which exploits the continuous and combinatorial structure of the problem. Oper. Res. Lett. **36**(1), 37–42 (2008a)
12. Birgin, E., Martinez, J., Nishihara, F.H., Ronconi, D.P.: Orthogonal packing of rectangular items within arbitrary convex regions by nonlinear optimization. Comput. Oper. Res. **33**(12), 3535–3548 (2006)
13. Egeblad, J., Nielsen, B.K., Odgaard, A.: Fast neighborhood search for two-and three-dimensional nesting problems. Eur. J. Oper. Res. **183**(3), 1249–1266 (2007)
14. Gomes, A.M., Olivera, J.F.: A 2-exchange heuristics for nesting problems. Eur. J. Oper. Res. **141**, 359–570 (2002)
15. Scheithauer, G., Stoyan, Y.G., Romanova, T.Y.: Mathematical modeling of interactions of primary geometric 3D objects. Cybern. Syst. Anal. **41**, 332–342 (2005)
16. Teng, H., Sun, S., Liu, D., Li, Y.: Layout optimization for the objects located within a rotating vessel a three-dimensional packing problem with behavioural constraints. Comput. Oper. Res. **28**(6), 521–535 (2001)
17. Caprara, A., Monaci, M.: On the 2-dimensional knapsack problem. Oper. Res. Lett. **32**(1), 5–14 (2004)
18. Egeblad, J., Pisinger, D.: Heuristic approaches for the two- and three-dimensional knapsack packing problems. DIKU Technical-Report 2006–13, SSN: 0107-8283, Department of Computer Science, University of Copenhagen, Denmark (2006)
19. Egeblad, J., Pisinger, D.: Heuristic approaches for the two- and three-dimensional knapsack packing problem. Comput. Oper. Res. **36**, 1026–1049 (2009)
20. Fekete, S.P., Schepers, J.: A New Exact Algorithm for General Orthogonal D-dimensional Knapsack Problems. In: Algorithms ESA'97, Springer Lecture Notes in Computer Science, vol. 1284, pp. 144–156 (1997)
21. Bortfeldt, A., Wäscher, G.: Container loading problems - A state-of-the-art review. FEMM Working Papers 120007, Otto-von-Guericke University Magdeburg, Faculty of Economics and Management (2012)
22. Kang, M.K., Jang, C.S., Yoon, K.S.: Heuristics with a new block strategy for the single and multiple container loading problems. J. Oper. Res. Soc. **61**, 95–107 (2010)
23. Parreño, F., Alvarez-Valdes, R., Oliveira, J.F., Tamarit, J.M.: A maximal-space algorithm for the container loading problem. INFORMS J. Comput. **20**(3), 412–422 (2008)
24. Lodi, A., Martello, S., Monaci, M., Vigo, D.: Two-dimensional Bin Packing Problems. In: Paschos, V.T. (ed.) Paradigms of Combinatorial Optimization, pp. 107–129. Wiley/ISTE, Hoboken (2010)
25. Pisinger, D., Sigurd, M.: Using decomposition techniques and constraint programming for solving the two-dimensional bin packing problem. INFORMS J. Comput. **19**(1), 36–51 (2007)
26. Iori, M., Martello, S., Monaci, M.: Metaheuristic Algorithms for the Strip Packing Problem. In: Pardalos, P.M., Korotkikh, V. (eds.) Optimization and Industry: New Frontiers, pp. 159–179. Kluwer, Hardbound (2003)
27. Kenmochi, M., Imamichi, T., Nonobe, K., Yagiura, M., Nagamochi, H.: Exact algorithms for the two-dimensional strip packing problem with and without rotations. Eur. J. Oper. Res. **198**, 73–83 (2009)

28. Zhang, D., Kang, Y., Deng, A.: A new heuristic recursive algorithm for the strip rectangular packing problem. Comput. Oper. Res. **33**, 2209–2217 (2006)
29. Li, H.L., Chang, C.T., Tsai, J.F.: Approximately global optimization for assortment problems using piecewise linearization techniques. Eur. J. Oper. Res. **140**, 584–589 (2002)
30. Pan, P., Liu, C.L.: Area minimization for floorplans. IEEE Trans. Comput. Aided Design Integr. Circuits Syst. **14**(1), 123–132 (2006)
31. Pintér, J.D., Kampas, F.J.: MathOptimizer Professional: Key Features and Illustrative Applications. In: Liberti, L., Maculan, N. (eds.) Global Optimization: From Theory to Implementation, pp. 263–279. Springer, Berlin (2006)
32. Pintér, J.D., Kampas, F.J., Castillo, I.: Globally optimized packings of non-uniform size spheres in $\mathbb{R}^d$: a computational study. Optim. Lett. **12**(3), 585–613 (2017)
33. Allen, S.D., Burke, E.K., Kendall, G.: A hybrid placement strategy for the three-dimensional strip packing problem. Eur. J. Oper. Res. **209**(3), 219–227 (2011)
34. Bennel, J.A., Lee, L.S., Potts, C.N.: A genetic algorithm for two-dimensional bin packing with due dates. Int. J. Prod. Econ. **145**(2), 547–560 (2013)
35. Bennell, J.A., Han, W., Zhao, X., Song, X.: Construction heuristics for two-dimensional irregular shape bin packing with guillotine constraints. Eur. J. Oper. Res. **230**(3), 495–504 (2013)
36. Bennell, J.A., Oliveira, J.F.: A tutorial in irregular shape packing problems. J. Oper. Res. Soc. **60**(S1), S93–S105 (2009)
37. Bortfeldt, A., Gehring, H.: A hybrid genetic algorithm for the container loading problem. Eur. J. Oper. Res. **131**(1), 143–161 (2001)
38. Burke, E.K., Hellier, R., Kendall, G., Whitwell, G.: A new bottom-left-fill heuristic algorithm for the 2D irregular packing problem. Oper. Res. **54**(3), 587–601 (2006)
39. Burke, E.K., Guo, Q., Hellier, R., Kendall, G.: A hyper-heuristic approach to strip packing problems. PPSN. **1**, 465–474 (2010)
40. Coffman, E., Garey, J.M., Johnson, D.: Approximation Algorithms for Bin Packing: A Survey. PWS Publishing, Boston (1997)
41. Dowsland, K.A., Herbert, E.A., Kendall, G., Burke, E.: Using tree search bounds to enhance a genetic algorithm approach to two rectangle packing problems. Eur. J. Oper. Res. **168**(2), 390–402 (2006)
42. Gehring, H., Bortfeldt, A.: A parallel genetic algorithm for solving the container loading problem. Int. Trans. Oper. Res. **9**(4), 497–511 (2002)
43. Gonçalves, J.F., Resende, M.G.: A parallel multi-population biased random-key genetic algorithm for a container loading problem. Comput. Oper. Res. **39**(2), 179–190 (2012)
44. Hopper, E., Turton, B.C.: A review of the application of meta-heuristic algorithms to 2D strip packing problems. Artif. Intell. Rev. **16**(4), 257–300 (2001)
45. Hopper, E., Turton, B.C.: An empirical study of meta-heuristics applied to 2D rectangular bin packing - part II. Studia Inform. Universalis. **2**(1), 93–106 (2002)
46. López-Camacho, E., Ochoa, G., Terashima-Marín, H., Burke, E.: An effective heuristic for the two-dimensional irregular bin packing problem. Annals OR. **206**(1), 241–264 (2013)
47. Mack, D., Bortfeldt, A., Gehring, H.: A parallel hybrid local search algorithm for the container loading problem. Int. Trans. Oper. Res. **11**(5), 511–533 (2004)
48. Oliveira, J.F., Gomes, A.M., Ferreira, J.S.: TOPOS - a new constructive algorithm for nesting problems. OR Spectr. **22**(2), 263–284 (2000)
49. Ramakrishnan, K., Bennel, J.A., Omar, M.K.: Solving Two Dimensional Layout Optimization Problems with Irregular Shapes by Using Meta-Heuristic. In: 2008 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 178–182 (2008)
50. Terashima-Marín, H., Ross, P., Farías-Zárate, C.J., López-Camacho, E., Valenzuela-Rendón, M.: Generalized hyper-heuristics for solving 2D regular and irregular packing problems. Ann. Oper. Res. **179**, 369–392 (2010)
51. Wang, Z., Li, K.W., Levy, J.K.: A heuristic for the container loading problem: a tertiary-tree-based dynamic space decomposition approach. Eur. J. Oper. Res. **191**(1), 86–99 (2008)

52. Yeung, L.H., Tang, W.K.: A hybrid genetic approach for container loading in logistics industry. IEEE Trans. Ind. Electron. **52**(2), 617–627 (2005)
53. Allen, S.D., Burke, E.K., Mareček, J.: A space-indexed formulation of packing boxes into a larger box. Oper. Res. Lett. **40**(1), 20–24 (2012)
54. Chen, C.S., Lee, S.M., Shen, Q.S.: An analytical model for the container loading problem. Eur. J. Oper. Res. **80**, 68–76 (1995)
55. Chernov, N., Stoyan, Y.G., Romanova, T.: Mathematical model and efficient algorithms for object packing problem. Comput. Geom. Theory Appl. **43**(5), 535–553 (2010)
56. Fasano, G.: Satellite Optimal Layout, Application of Mathematical and Optimization Techniques, pp. 22–26. IBM Europe Institute, Garmisch (1989)
57. Fischetti, M., Luzzi, I.: Mixed-integer programming models for nesting problems. J. Heuristics. **15**(3), 201–226 (2009)
58. Hadjiconstantinou, E., Christofides, N.: An exact algorithm for general, orthogonal, two-dimensional knapsack problems. Eur. J. Oper. Res. **83**(1), 39–56 (1995)
59. Kallrath, J.: Cutting circles and polygons from area-minimizing rectangles. J. Glob. Optim. **43**(2–3), 299–328 (2009)
60. Padberg, M.W.: Packing Small Boxes into a Big Box. Office of Naval Research, N00014-327, New York University, New York (1999)
61. Pisinger, D., Sigurd, M.: The two-dimensional bin packing problem with variable bin sizes and costs. Discret. Optim. **2**(2), 154–167 (2005)
62. Daughtrey, R.S.: A Simulated Annealing Approach to 3-D Packing with Multiple Constraints. Cosmic Program MFS28700. Boeing Huntsville AI Center, Huntsville (1991)
63. Fasano, G.: Solving Non-Standard Packing Problems by Global Optimization and Heuristics. In: Springer Briefs in Optimization. Springer, New York (2014)
64. Fasano, G., Lavopa, C., Negri, D., Vola, M.C.: CAST: A Successful Project in Support of the International Space Station Logistics. In: Fasano, G., Pintér, J.D. (eds.) Optimized Packings and their Applications. Springer Optimization and its Applications. Springer, New York (2015)
65. Fasano, G., Pintér, J.D.: Modeling and Optimization in Space Engineering. Springer, New York (2013)
66. Fasano, G., Pintér, J.D.: Space Engineering. Modeling and Optimization with Case Studies. Springer Optimization and its Applications. Springer, New York (2016)
67. Fasano, G., Vola, M.C.: Space Module On-Board Stowage Optimization Exploiting Containers' Empty Volumes. In: Fasano, G., Pintér, J.D. (eds.) Modeling and Optimization in Space Engineering, pp. 249–269. Springer, New York (2013)
68. Takadama, A.K., Shimomura, K.: Cargo Layout Optimization in Spacecraft: Exploring Heuristics for Branch-and- Bound Method. In: The 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space (2005)
69. Addis, B., Locatelli, M., Schoen, F.: Disk packing in a square: a new global optimization approach. INFORMS J. Comput. **20**(4), 516–524 (2008b)
70. Castillo, I., Kampas, F.J., Pintér, J.D.: Solving circle packing problems by global optimization: numerical results and industrial applications. Eur. J. Oper. Res. **191**(3), 786–802 (2008)
71. Floudas, C.A., Akrotirianakis, I.G., Caratzoulas, S., Meyer, C.A., Kallrath, J.: Global optimization in the 21st century: advances and challenges for problems with nonlinear dynamics. Comput. Chem. Eng. **29**, 1185–1202 (2005)
72. Floudas, C.A., Pardalos, P.M.: A Collection of Test Problems for Constrained Global Optimization Algorithms. Springer, New York (1990)
73. Floudas, C.A., Pardalos, P.M., et al.: Handbook of Test Problems in Local and Global Optimization, Nonconvex Optimization and its Applications Series, vol. 33. Kluwer, Dordrecht (1999)
74. Floudas, C.A., Pardalos, P.M. (eds.): Encyclopedia of Optimization. Kluwer, Dordrecht (2001)
75. Horst, R., Pardalos, P.M. (eds.): Handbook of Global Optimization, vol. 1. Kluwer, Dordrecht (1995)

76. Horst, R., Pardalos, P.M. (eds.): Developments in Global Optimization. Kluwer, Dordrecht (1997)
77. Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches, 3rd edn. Springer, Berlin (1996)
78. Kallrath, J.: Mixed-Integer Nonlinear Applications. In: Ciriani, T., Ghiozzi, S., Johnson, E.L. (eds.) Operations Research in Industry, pp. 42–76. Macmillan, London (1999)
79. Kallrath, J.: Modeling Difficult Optimization Problems. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, 2nd edn, pp. 2284–2297. Springer, New York (2008)
80. Liberti, L., Maculan, N. (eds.): Global Optimization: from Theory to Implementation. Springer, New York (2005)
81. Locatelli, M., Raber, U.: Packing equal circles into a square: a deterministic global optimization approach. Discret. Appl. Math. **122**, 139–166 (2002)
82. Pardalos, P.M., Resende, M.G. (eds.): Handbook of Applied Optimization. Oxford University Press, Oxford (2002)
83. Pardalos, P.M., Romeijn, H.E. (eds.): Handbook of Global Optimization, vol. 2. Kluwer, Dordrecht (2002)
84. Pintér, J.D.: Global Optimization in Action. Kluwer, Dordrecht (1996)
85. Rebennack, S., Kallrath, J., Pardalos, P.M.: Column enumeration based decomposition techniques for a class of non-convex MINLP problems. J. Glob. Optim. **43**(2–3), 277–297 (2009)
86. Pintér, J.D.: Global optimization in practice: state of the art and perspectives. In: Gao, D., Sherali, H. (eds.) Advances in Applied Mathematics and Global Optimization. Advances in Mechanics and Mathematics, vol. 17. Springer, Boston (2009)
87. Fasano, G.: A Modeling-Based Approach for Non-standard Packing Problems. In: Fasano, G., Pintér, J.D. (eds.) Optimized Packings and their Applications. Springer Optimization and its Applications. Springer, New York (2015)
88. Fasano, G., Saia, D., Piras, A.: Columbus stowage optimization by CAST (Cargo Accommodation Support Tool). Acta Astronaut. **67**(3–4), 489–495 (2010)
89. Stoyan, Y., Grebennik, I., Romanova, T., Kovalenko, A.: Optimized Packings in Space Engineering Applications - Part II. In: Fasano, G., Pintér, J.D. (eds.) Modeling and Optimization in Space Engineering - 2019. Springer, New York (2019)
90. Romanova, T., Bennell, J., Stoyan, Y., Pankratov, A.: Packing of concave polyhedra with continuous rotations using nonlinear optimization. Eur. J. Oper. Res. **268**(1), 37–53 (2018)
91. Stoyan, Y., Pankratov, A., Romanova, T.: Quasi phi-functions and optimal packing of ellipses. J. Glob. Optim. **65**(2), 283–307 (2016)
92. Stoyan, Y., Romanova, T.: Mathematical Models of Placement Optimisation: Two- and Three-Dimensional Problems and Applications. In: Fasano, G., Pintér, J. (eds.) Modeling and Optimization in Space Engineering, Springer Optimization and its Applications, vol. 73, pp. 363–388. Springer, New York (2012)
93. Stoyan, Y., Romanova, T., Pankratov, A., Kovalenko, A., Stetsyuk, P.: Modeling and Optimization of Balance Layout Problems. In: Fasano, G., Pintér, J. (eds.) Space Engineering. Modeling and Optimization with Case Studies. Springer Optimization and its Applications, vol. 114, pp. 369–400. Springer, New York (2016)
94. Egeblad, J., Nielsen, B.K., Brazil, M.: Translational packing of arbitrary polytopes. Comput. Geom. Theory Appl. **42**(4), 269–288 (2009)
95. Liu, X., Liu, J., Cao, A., Yao, Z.: HAPE3D - a new constructive algorithm for the 3D irregular packing problem. Front. Inform. Tech. Elect. Eng. **16**(5), 380–390 (2015)
96. Youn-Kyoung, J., Sang, D.N.: Intelligent 3D packing using a grouping algorithm for automotive container engineering. J. Comput. Des. Eng. **1**(2), 140–151 (2014)
97. Pankratov, O., Romanova, T., Stoyan, Y., Chuhai, A.: Optimization of packing polyhedra in spherical and cylindrical containers. East. Eur. J. Enterp. Tech. **4**(79), 39–47 (2016)
98. Stetsyuk, P., Romanova, T., Scheithauer, G.: On the global minimum in a balanced circular packing problem. Optim. Lett. **10**, 1347–1360 (2016)

99. Stoyan, Y.G., Gil, N.I., Pankratov, A.V.: Packing Non-convex Polyhedra into a Parallelepiped. Technische Universitat Dresden, Dresden (2004)
100. Stoyan, Y., Gil, N., Scheithauer, G., Pankratov, A., Magdalina, I.: Packing of convex polyhedra into a parallelepiped. Optimization. **54**(2), 215–235 (2005)
101. Stoyan, Y., Pankratov, A., Romanova, T., Chugay, A.: Optimized Object Packings Using Quasi-Phi-Functions. In: Fasano, G., Pintér, J. (eds.) Optimized Packings and their Applications, Springer Optimization and its Applications, vol. 105, pp. 265–291. Springer, New York (2015)
102. Fischer, K., Gärtner, B., Kutz, M.: Fast Smallest-Enclosing-Ball Computation in High Dimensions. In: Algorithms - ESA 2003, vol. 2832, pp. 630–641 (2003)
103. Wachter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large- scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)

Optimized Packings in Space Engineering Applications: Part II

Yu. Stoyan, I. Grebennik, T. Romanova, and A. Kovalenko

Abstract This chapter, dedicated to a specific packing optimization scenario of considerable interest in space engineering and logistics, follows a previous one appearing in this volume [1]. Although it is presented as the second part of the whole topical discussion proposed, it can be read independently.

The layout optimization, with balancing conditions, of a given set of 3D-objects, in a container partitioned by horizontal planes into subcontainers, is considered.

We define special combinatorial configurations describing the specific structure of the problem. A mathematical model, based on the combination of the phi-function technique and the introduced configurations, is provided. The model takes into account not only the placement constraints (i.e., nonoverlapping, containment) and the mechanical characteristics of the system but also the combinatorial features relevant to the partitions of the set of objects placed inside the subcontainers. The solution strategy is proposed and the results of numerical experiments are presented.

1 Introduction

Layout instances with balancing conditions belong to the class of NP-hard problems and are a subject of study in computational geometry and operations research [2, 3], where the methods adopted for their solution represent quite a recent branch.

The essence of the problem lies in the search for the optimal placement of a given set of 3D-objects in a container, ensuring the balancing of the system under consideration.

Y. Stoyan · T. Romanova (✉)
Department of Mathematical Modeling and Optimal Design, Institute for Mechanical Engineering Problems of the National Academy of Sciences of Ukraine, Kharkov, Ukraine
e-mail: tarom27@yahoo.com

I. Grebennik · A. Kovalenko
Department of System Engineering, Kharkov National University of Radio Electronics, Kharkov, Ukraine

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_16

The necessity of taking into account the assigned balancing constraints in optimization layout problems arises in various applied science fields and technologies, such as, for example, in engineering concerning the design of aircraft, ships, submarines, devices, and components, in logistics problems (when packing goods for transportation or storage) [4, 5]. Rocket design and space technology represent an area of particular interest in this class of problems.

At the initial stages of the layout definition of a spacecraft, it is necessary to take into account special constraints on static and dynamic characteristics (center of mass, axial and centrifugal moments of inertia), as described in [6].

In [7–10], the problems of the layout of cylinders in a cylindrical container with balancing constraints are considered. These publications provide mathematical models with different objective functions. To solve these problems, heuristic algorithms, based on the specific features of each mathematical model, are proposed. Papers [11–13] consider mathematical models and methods for solving the layout problem of a given set of objects, with balancing conditions.

In [3], the authors study (NP-hard) placement optimization problems, which cover a wide spectrum of industrial applications, including space engineering. The present chapter considers mathematical modelling tools and a solution strategy for placement problems. A class of 2D/3D geometric objects, called *phi*-objects, is introduced and considered as mathematical models of real items. The concept of *phi*-functions is used to describe nonoverlapping and containment constraints. A mathematical model of a basic placement issue is constructed as a constrained optimization problem that takes into account allowable distances between objects. A solution strategy is proposed. As an example, a placement optimization problem with balancing conditions arising in space engineering is considered. This consists in the placement of cylinders and cuboids of given weights and sizes in a parabolic container, divided by parallel axial circles minimizing the deviation of the system center of gravity from a given point. The chapter also provides a number of computational results for 2D and 3D applications.

Paper [13] studies the layout optimization problem, called BLP, of 3D-objects (solid spheres, straight circular cylinders, spherocylinders, straight regular prisms, cuboids, and tori) in a container (with cylindrical, parabolic, or truncated conical shapes) partitioned into sectors by parallel axial circles. The problem takes into account given minimum and maximum allowable distances between objects, as well as balancing conditions in terms of equilibrium, moments of inertia, and stability constraints. A continuous nonlinear programming (NLP) model of the problem is developed using the phi-function technique. The abovementioned paper also considers several BLP variants, providing appropriate mathematical models and solution algorithms, based on nonlinear programming and nonsmooth optimization methods, as well as the relevant computational analysis. In this work, however, the assignment of objects to the container sectors is assumed to be established a priori.

The innovative contribution of this chapter relates to the following points:

1. We extend the formulation of the layout optimization problem discussed in [13]. Our new formulation, called CBLP, takes into account not only placement

and balancing constraints of the system but also combinatorial features of the problem. These consist, in particular, in the assignment (no longer established a priori) of the given items to the system sectors.
2. We investigate the concept of combinatorial configurations to handle the discrete structure of the CBLP problem.
3. We define mathematical modelling tools for placement constraints, with both continuous and discrete variables, called D-phi-function and quasi-D-phi-function.
4. We provide a mathematical model formulation of the CBLP problem that involves both continuous and discrete variables.
5. We propose a solution strategy that uses the novel algorithm for the combinatorial configuration generation.

2 Problem Formulation

Let Ω be a container of height H that can take the form of a cuboid, cylinder, paraboloid of rotation, or truncated cone. The container Ω is defined in the fixed coordinate system $Oxyz$, where Oz is the longitudinal axis of symmetry. We assume that container Ω is divided by horizontal planes into subcontainers Ω^j, $j \in J_m = \{1, \ \dots \ , m\}$. We denote distances between circles S_j and S_{j+1} by t_j, $j \in J_m$, $\sum_{j=1}^{m} t_j = H$. The center of the base of container Ω is located in the origin of the coordinate system $Oxyz$.

Let $A = \{\mathbb{T}_i, i \in J_n\}$ be a set of homogeneous 3D-objects given by their metrical characteristics. Each object $\mathbb{T}_i$ of height h_i and mass m_i is defined in its local coordinate system $O_i x_i y_i z_i$, $i \in J_n$. The location of object $\mathbb{T}_i$ inside container Ω is defined by vector $u_i = (v_i, z_i, \theta_i)$, where (v_i, z_i) is a translation vector in the coordinate system $Oxyz$, θ_i is a rotation angle of object $\mathbb{T}_i$ in the plane $O_i x_i y_i$, where $v_i = (x_i, y_i)$, and the value of z_i, $i \in J_n$, is uniquely defined by subcontainer Ω^j, $j \in J_m$, in which object $\mathbb{T}_i$ will be placed.

In the BLP problem, the requirement for placing objects in specific subcontainers $\Omega^j, j \in J_m$, is known a priori. In this study, the issue of the balanced layout of objects is formulated, considering the generation and selection of a partition of the set A into nonempty subsets $A^j, j \in J_m$. Here, A^j is a subset of objects which have to be placed on circle S_j inside subcontainer Ω^j.

Regarding the placement of object $\mathbb{T}_i$, $i \in J_n$, inside subcontainer Ω^j the following constraints are imposed:

$$z_i = \sum_{l=1}^{j} t_{l-1} + h_i, \tag{1}$$

where $j \in J_m$.We consider that $t_0 = 0$ and $\forall i \in J_n$ there exists $j^* \in J_m$: $h_i \leq t_{j^*}$.

Let $J_n^j \subseteq J_n$ be a set of indexes of objects which are placed in subcontainer Ω^j, $j \in J_m$:

$$\bigcup_{j=1}^{m} J_n^j = J_n, \quad J_n^i \cap J_n^j = \varnothing, i \neq j \in J_m; \tag{2}$$

$k_j = |A^j|$ is the number of objects which are placed in subcontainer Ω^j, $k_j > 0$, $j \in J_m$:

$$\sum_{j=1}^{m} k_j = n. \tag{3}$$

In addition, the following placement constraints have to be taken into account:

$$int\mathbb{T}_{i_1} \cap int\mathbb{T}_{i_2} = \emptyset, i_1 < i_2 \in J_n^j, j \in J_m, \tag{4}$$

$$\mathbb{T}_i \subseteq \Omega^j, i \in J_n^j, j \in J_m, \tag{5}$$

$$h^j \leq t_j, h^j = \max\left\{h_i^j, i \in J_n^j\right\}, j \in J_m. \tag{6}$$

We designate a system, formed as a result of the placement of objects $\mathbb{T}_i$ of the set A in container Ω by Ω_A, and a reference frame of Ω_A by O_sXYZ, where $O_s = (x_s(v), y_s(v), z_s(v))$ is the mass center of Ω_A:

$$x_s(v) = \frac{\sum_{i=1}^{n} m_i x_i}{M}, y_s(v) = \frac{\sum_{i=1}^{n} m_i y_i}{M}, z_s = \frac{\sum_{i=1}^{n} m_i z_i}{M}, \tag{7}$$

$M = \sum_{i=1}^{n} m_i$ is the mass of system Ω_A and $O_sX \| Ox$, $O_sY \| Oy$, $O_sZ \| Oz$.

We consider the deviation of the center of mass O_s of system Ω_A from the given point (x_0, y_0, z_0) as the objective function.

Combinatorial Balanced Layout Problem (CBLP) Define a partition of set A into nonempty subsets A^j, $j \in J_m$, and the corresponding placement parameters $u_i = (v_i, z_i, \theta_i)$ of objects $\mathbb{T}_i$, $i \in J_n$, such that the objective function is minimized, taking into account relations (2)–(6).

We assume that the problem has at least one feasible solution.

N.B Restrictions on the axial and centrifugal moments of the system and allowable distances between objects may also be given.

3 Mathematical Model

Now, we define special combinatorial configurations describing the discrete structure of the CBLP problem. Some basic approaches for mathematical modelling of optimization problems on combinatorial configurations are described, for instance, in [14–16].

The possibilities of partitioning set A into nonempty subsets A^j, $j \in J_m$, are determined by both the number of elements in each subset and the order of the subsets.

Let us consider the subcontainers and the assumed corresponding sets of objects $A^j, j \in J_m$. Then, the tuple of natural numbers $(k_1, k_2, \ldots, k_m)$, such that $\sum_{j=1}^{m} k_j = n$, denotes the number k_j of objects associated with each subcontainer Ω^j.

The number of all such tuples is equal to the number of compositions of the number n of length m [17], which is $\frac{(n-1)!}{(m-1)!(n-m)!}$.

We shall derive how it is possible to partition a set A of n objects into m subcontainers Ω^j, $j \in J_m$, containing $k_1, k_2, \ldots, k_m$ items, respectively, with no ordering condition within each Ω^j. We denote subsets of objects that are placed inside corresponding subcontainers Ω^j by $A^j, j \in J_m$.

Without loss of generality, we will distinguish the objects with the same values of metrical characteristics, height h_i and mass m_i (for example, providing them with different identification numbers).

We order the elements of set A. We assign to each object the number of the subcontainer into which it is expected to be placed. We get a tuple consisting of n elements that form a permutation with repetitions from m numbers $1, 2, \ldots, m$, in which the first element is repeated k_1 times, the second element is repeated k_2 times, ..., the last element is repeated k_m times.

The total number of permutations is equal to:

$$P(n, k_1, k_2, \ldots, k_m) = \frac{n!}{k_1! \cdot k_2! \cdot \cdots \cdot k_m!}. \tag{8}$$

Therefore, the total number of partitions of n objects into m subcontainers Ω^j, provided that each Ω^j contains at least one object and the order in which objects are placed inside Ω^j is not considered, is equal to:

$$\sum_{k_1+k_2+\cdots+k_m=n} P(n, k_1, k_2, \ldots, k_m) = \sum_{k_1+k_2+\cdots+k_m=n} \frac{n!}{k_1! \cdot k_2! \cdot \cdots \cdot k_m!} \tag{9}$$

Note that the number of summands in (9) is equal to $N = \left| C_{n-1}^{m-1} \right| = \frac{(n-1)!}{(m-1)!(n-m)!}$.

To generate subsets A^j, $j \in J_m$, we introduce a special combinatorial configuration [18].

Rather complex combinatorial configurations can formally be described and generated using compositional κ-images of combinatorial sets (κ-sets) introduced in [19]. A combinatorial set is a set of tuples that are constructed from a finite set of arbitrary elements (so-called generating elements), according to certain rules. Permutations, combinations, and binary sequences are examples of classical combinatorial sets. We describe the concept of κ-sets.

The basic idea of κ-sets generation is introduced in [19]. However, the problem of generating κ-sets of more complicated combinatorial structure remains an open problem. One of these special cases is studied in [20].

The problem of generating κ-sets is based on special techniques of generating basic combinatorial sets. The basic sets can be defined as combinatorial sets with the given features, i.e., both classical combinatorial sets (e.g., permutations, combinations, compositions, partitions, and n-tuples) and nonclassical combinatorial sets (e.g., permutations of tuples, compositions of permutations, and permutations with a prescribed number of cycles). Generation algorithms for basic combinatorial sets are described, e.g., in [21–25].

A generation strategy for the compositional κ-images of combinatorial sets (κ-sets).

We denote as $\mathbb{C}_{\mathbb{P}}(n,m)$ the set of compositions of the number n of length m (which corresponds to the partition of different objects from set A into m subcontainers Ω^j, $j \in J_m$), provided that each subcontainer contains at least one object and the order of objects inside the subcontainer is not considered, where, $|\mathbb{C}_{\mathbb{P}}(n,m)| = N = \left|C_{n-1}^{m-1}\right|$.

Let $\mathbb{k} = (k_1, k_2, \ldots, k_m) \in \mathbb{C}_{\mathbb{P}}(n,m)$, $\sum_{j=1}^{m} k_j = n$, $k_j \geq 1, j \in J_m$.

We introduce a combinatorial set $\mathbb{Q}(\mathbb{k})$ that is a composition image of combinatorial sets (κ-set) $\mathbb{C}_{\mathbb{P}}(n,m)$; $C_n^{k_1}$, $C_{n_1}^{k_2}$, $C_{n_2}^{k_3}$, …, $C_{n_{m-1}}^{k_m}$, generated by sets I_{n_0}, I_{n_1}, I_{n_2}, …, $I_{n_{m-1}}$, where $n_i = n - k_1 - \ldots - k_i$, $i \in J_{m-1}$,

$$I_{n_0} = J_n,$$

$$I_{n_1} = I_{n_0} \setminus \left\{ j_1^{n_0}, j_2^{n_0}, \ldots, j_{k_1}^{n_0} \right\}, \left(j_1^{n_0}, j_2^{n_0}, \ldots, j_{k_1}^{n_0} \right) \in C_n^{k_1},$$

$$I_{n_2} = I_{n_1} \setminus \left\{ j_1^{n_1}, j_2^{n_1}, \ldots, j_{k_2}^{n_1} \right\}, \left(j_1^{n_1}, j_2^{n_1}, \ldots, j_{k_2}^{n_1} \right) \in C_{n_1}^{k_2},$$

…

$$I_{n_{m-1}} = I_{n_{m-2}} \setminus \left\{ j_1^{n_{m-2}}, j_2^{n_{m-2}}, \ldots, j_{k_{m-1}}^{n_{m-2}} \right\}, \left(j_1^{n_{m-2}}, j_2^{n_{m-2}}, \ldots, j_{k_{m-1}}^{n_{m-2}} \right) \in C_{n_{m-2}}^{k_{m-1}},$$

$$I_{n_{m-1}} = \left\{ j_1^{n_{m-1}}, j_2^{n_{m-1}}, \ldots, j_{k_m}^{n_{m-1}} \right\}, \left(j_1^{n_{m-1}}, j_2^{n_{m-1}}, \ldots, j_{k_m}^{n_{m-1}} \right) \in C_{n_{m-1}}^{k_m}.$$

Note that

$$I_{n_0} \cup I_{n_1} \cup \cdots \cup I_{n_{m-1}} = J_n = \{1, 2, \ldots, n\},$$

$$I_{n_s} \cap I_{n_t} = \varnothing, s \neq t \in J_{m-1}^0 = \{0, 1, \ldots, m-1\}.$$

Each element $q\,(\Bbbk) \in \mathbb{Q}\,(\Bbbk)$ can be described in the form:

$$q\,(\Bbbk) = \left(q_1, \ldots, q_{k_1} \left| q_{k_1+1}, \ldots, q_{k_1+k_2} \right|, \ldots, \left| q_{k_1+\cdots+k_{m-1}}, \ldots, q_{k_{m-1}+k_m} \right.\right),$$

where $\left(q_1, \ldots, q_{k_1}\right) = \left(j_1^{n_0}, j_2^{n_0}, \ldots, j_{k_1}^{n_0} \right) \in C_n^{k_1}$,

$$\left(q_{k_1+1}, \ldots, q_{k_1+k_2}\right) = \left(j_1^{n_1}, j_2^{n_1}, \ldots, j_{k_2}^{n_1} \right) \in C_{n_1}^{k_2},$$

$$\ldots$$

$$\left(q_{k_1+\cdots+k_{m-1}}, \ldots, q_{k_{m-1}+k_m}\right) = \left(j_1^{n_{m-1}}, j_2^{n_{m-1}}, \ldots, j_{k_m}^{n_{m-1}} \right) \in C_{n_{m-1}}^{k_m}.$$

The cardinality of set $\mathbb{Q}\,(\Bbbk)$ is derived by (9).

An element $q\,(\Bbbk)$ of the set $\mathbb{Q}\,(\Bbbk)$ is said to be *a tuple of partition* of the set A into subsets $A^j, j \in J_m$.

Now, we define the vector of the basic variables of the problem CBLP: $u = (v, z, \theta)$, where $v = (v_1, \ldots, v_n) \in \mathbf{R}^{2n}$, $\theta = (\theta_1, \ldots, \theta_n) \in \mathbf{R}^n$, $v_i = (x_i, y_i) \in \mathbf{R}^2$, x_i, y_i, θ_i are continuous variables, and $z = (z_1, \ldots, z_n) \in \mathbf{R}^n$, z_i, $i \in J_n$, are discrete variables defined by (1).

The values of variables z_i, $i \in J_n$, are determined in the order given by elements $q\,(\Bbbk)$ of combinatorial set $\mathbb{Q}\,(\Bbbk)$:

$$z_{q_i} = \sum_{l=1}^{s} t_{l-1} + h_{q_i}, \tag{10}$$

where:

$$s = \begin{cases} 1, \text{if} \quad i \le k_1, \\ 2, \text{if} \quad k_1 < i \le k_1 + k_2, \\ \dots \\ m, \text{if} \quad k_1 + k_2 + \dots + k_{m-1} < i \le k_1 + k_2 + \dots + k_m, \end{cases}$$

$i = 1, 2, .., n$, $q_i \in \{1, 2, .., n\}$, $q\ (\mathbb{k}) \in \mathbb{Q}\ (\mathbb{k})$.

Let us formalize placement constraints (4)–(6), using the phi-function technique.

We consider two objects $\mathbb{T}_1$ and $\mathbb{T}_2$ with variable parameters $u_1 = (v_1, z_1, \theta_1) \in \mathbf{R}^3$, $u_2 = (v_2, z_2, \theta_2) \in \mathbf{R}^3$, where $v_1 = (x_1, y_1)$, $v_2 = (x_2, y_2)$, x_1, y_1, θ_1 x_2, y_2, θ_2 are continuous variables and z_1, z_2 are discrete variables.

By definition [2, 3], a phi-function is a continuous function, therefore we extend the concept to discrete variables z_1, z_2.

Definition 1 Function $\Upsilon_{12}(u_1, u_2)$ is called a *D-phi*-function of 3D-objects $\mathbb{T}_1$ and $\mathbb{T}_2$ if function $\Upsilon_{12}\left(v_1, z_1^0, \theta_1, v_2, z_2^0, \theta_2\right)$ is a *phi*-function $\Phi_{12}\left(v_1, z_1^0, \theta_1, v_2, z_2^0, \theta_2\right)$ of objects $\mathbb{T}_1$ and $\mathbb{T}_2$ for fixed values $z_1 = z_1^0$ and $z_2 = z_2^0$.

Definition 2 Function $\Upsilon'_{12}\left(u_1, u_2, u_{12}\right)$ is called a quasi-D-*phi*-function of 3D-objects, $\mathbb{T}_1$ and $\mathbb{T}_2$ if function $\Upsilon'_{12}\left(v_1, z_1^0, \theta_1, v_2, z_2^0, \theta_2, u_{12}\right)$ is a quasi-*phi*-function $\Phi'_{12}\left(v_1, z_1^0, \theta_1, v_2, z_2^0, \theta_2, u_{12}\right)$ of objects $\mathbb{T}_1$ and $\mathbb{T}_2$ for fixed values $z_1 = z_1^0$ and $z_2 = z_2^0$.

Here, u_{12} is the vector of auxiliary continuous variables that is used to construct a quasi-*phi*-function of objects $\mathbb{T}_1$ and $\mathbb{T}_2$.

The placement constraints (4)–(6) are described by the system of inequalities $\Upsilon_1(u, \tau) \ge 0$, $\Upsilon_2^*(u) \ge 0$, where the inequality $\Upsilon_1(u, \tau) \ge 0$ describes the nonoverlapping constraints and the inequality $\Upsilon_2^*(u) \ge 0$ describes the containment constraints:

$$\Upsilon_1\left(u, \tau\right) = \min\left\{\Upsilon_1^j\left(u, \tau\right), j \in J_m\right\},$$

$$\Upsilon_1^j\left(u, \tau\right) = \min\left\{\Upsilon_{q_1 q_2}^j\left(u_{q_1}, u_{q_2}, u_{q_1 q_2}\right), q_1 < q_2 \in J_n^j\right\}, \tag{11}$$

$$\tau = \left(u_{q_1 q_2}, q_1 < q_2 \in J_n^j\right),$$

$$\Upsilon_2^*(u) = \min\left\{\Upsilon_2^{*j}(u), j \in J_m\right\}, \Upsilon_2^{*j}(u) = \min\left\{\Upsilon_{q_i}^*\left(u_{q_i}\right), q_i \in J_n^j\right\}, \tag{12}$$

$\Upsilon_{q_1 q_2}^j\left(u_{q_1}, u_{q_2}, u_{q_1 q_2}\right)$ is the function that describes the nonoverlapping condition between objects $\mathbb{T}_{q_1}$ and $\mathbb{T}_{q_2}$, and $u_{q_1} = \left(x_{q_1}, y_{q_1}, z_{q_1}, \theta_{q_1}\right)$, $u_{q_2} = \left(x_{q_2}, y_{q_2}, z_{q_2}, \theta_{q_2}\right)$, $\Upsilon_{q_i}^*\left(u_{q_i}\right)$ is the function that describes the nonoverlapping condition between objects $\mathbb{T}_{q_i}$ and $\Omega^{*j} = \mathbf{R}^3 / \operatorname{int} \Omega^j$.

Thus, in expressions (11) and (12) for fixed values z_{q_1} and z_{q_2}, we have: $\Upsilon^{j}_{q_1q_2}\left(u_{q_1}, u_{q_2}\right)$ is a *phi*-function [26] $\Phi^{\mathbb{T}\mathbb{T}}_{q_1q_2}\left(u_{q_1}, u_{q_2}\right)$ for objects $\mathbb{T}_{q_1}$ and $\mathbb{T}_{q_2}$ or a quasi-*phi*-function [27] $\Phi'^{\mathbb{T}\mathbb{T}}_{q_1q_2}\left(u_{q_1}, u_{q_2}, u_{q_1q_2}\right)$ for objects $\mathbb{T}_{q_1}$ and $\mathbb{T}_{q_2}$; $\Upsilon^{*}_{q_i}\left(u_{q_i}\right)$ is a *phi*-function $\Phi^{\mathbb{T}\Omega^{*j}}_{q_i}\left(u_{q_i}\right)$ for objects $\mathbb{T}_{q_i}$ and Ω^{*j}.

If a minimum allowable distance condition between objects is given, adjusted *phi*-functions (quasi-*phi*-functions) are used for the corresponding pairs of objects [26, 27].

The mathematical model of the CBLP problem can be defined as follows:

$$F\left(u^*, \tau^*\right) = \min F\left(u, \tau\right) \ \text{ s.t. } (u, \tau) \in W, \tag{13}$$

$$W = \left\{(u, \tau) \in \mathbf{R}^{\sigma} : \Upsilon_1\left(u, \tau\right) \geq 0, \Upsilon^{*}_2(u) \geq 0, \mu(u) \geq 0\right\}, \tag{14}$$

where:

$$F(u) = d = \left(x_s\left(v, z\right) - x_0\right)^2 + \left(y_s\left(v, z\right) - y_0\right)^2 + \left(z_s - z_0\right)^2$$

$u = (v, z, \theta)$, $v = (v_1, \ \ldots \ , v_n)$, $\theta = (\theta_1, \ \ldots \ , \theta_n)$, $v_i = (x_i, y_i)$, $i \in J_n$, $v = (v_1, \ \ldots \ , v_n)$, $\theta = (\theta_1, \ \ldots \ , \theta_n)$, $v_i = (x_i, y_i)$, $i \in J_n$, $z = (z_1, \ \ldots \ , z_n)$, function $\Upsilon_1(u, \tau)$ is described by (11) with $\Xi = \bigcup\limits_{j=1}^{m} \Xi^j$, $\Xi^j = \left\{(q_1, q_2) : q_1 < q_2 \in J^j_n\right\}$, $\tau = (\tau_1, \ldots, \tau_s) = \left(u_{q_1q_2}, q_1 < q_2 \in J^j_n\right)$ is a vector of auxiliary variables for quasi-phi-functions, $s = |\Xi|$, function $\Upsilon^{*}_2(u)$ is defined by (12), elements of vector z are given by (10), and $\mu(u) \geq 0$ describes the given balancing constraints.

For example, problem (13) and (14) for the layout of cylinders in a cylindrical container takes the form:

$$\min d, \text{ s.t.} u = (v, z) \in W, \tag{15}$$

where:

$$v = \left(x_1, y_1, \ldots, x_n, y_n\right), \mathrm{z} = \left(z_1, \ldots, z_n\right),$$

$$d = \left[\sum_{i=1}^{n} m'_i x_i\right]^2 + \left[\sum_{i=1}^{n} m'_i y_i\right]^2 + \left[\sum_{i=1}^{n} m'_i z_i - z_0\right]^2,$$

and the feasible region W is described by the inequality system:

$$\begin{cases} \left(x_{q_2} - x_{q_1}\right)^2 + \left(y_{q_2} - y_{q_1}\right)^2 - \left(r_{q_2} + r_{q_1}\right)^2 \geq 0, \\ q_1, q_2 \in \Xi^j,\ j \in J_m, \\ -{x_{q_i}}^2 - {y_{q_i}}^2 + \left(R^z_{q_i} - r_{q_i}\right)^2 \geq 0, \\ q_i \in \Xi^j,\ j \in J_m. \end{cases} \tag{16}$$

Note, that $m'_i = \frac{m_i}{M} = const$, $M = \sum_{i=1}^{n} m_i = const$.

The problem of packing cylinders into cylindrical containers, with balancing conditions, is considered, for instance, in [11, 13].

The CBLP problem can be represented as a mixed integer programming (MIP) problem, using Boolean variables. However, unlike (13) and (14), this approach increases the number of discrete variables and therefore increases the dimension of the CBLP problem.

4 Solution Strategy

The following strategy is used to solve CBLP problems:

1. Generate a subset $\{q\,(\Bbbk)\}_\chi \subset \mathbb{Q}\,(\Bbbk)$ using the concept of the Nested Combinatorial κ-sets.
2. Construct a subset $\left\{q'\,(\Bbbk)\right\}_{\chi'} \subseteq \{q\,(\Bbbk)\}_\chi$ of tuples that satisfy (6). If $\left\{q'\,(\Bbbk)\right\}_{\chi'} = \varnothing$, then go to Step 1.
3. Construct a set of feasible starting points $\left\{u'_0\right\}$ for each tuple from the set $\left\{q'\,(\Bbbk)\right\}_{\chi'}$, using the algorithm presented in [13].
4. Search for a local extremum of problem (13) and (14) for each starting point $u'_0 \in W$ with respect to $q'\,(\Bbbk) \in \left\{q'\,(\Bbbk)\right\}_{\chi'}$.
5. Choose the best of the local extrema found for all tuples of the set $\left\{q'\,(\Bbbk)\right\}_{\chi'}$ and feasible starting points $\left\{u'_0\right\}$ as a local optimal solution of problem (13) and (14).

To solve nonlinear programming problems, IPOPT is used, being available as an open noncommercial resource (https://projects.coin-or.org/Ipopt). IPOPT is based on the internal point method described in [28].

In order to generate a subset $\{q\,(\Bbbk)\}_\chi \subset \mathbb{Q}\,(\Bbbk)$, we use the concept of the Nested Combinatorial κ-sets.

Concept of the Nested Combinatorial κ-Sets To define a structure of the nested combinatorial κ-set, we use the κ-level tree.

Let $i \in J_{\kappa}^{0} = \{0, 1, \ldots, \kappa\}$, where κ is the number of levels of the tree. At each level of the tree, we have η_i nodes.

And, let

$$\begin{gathered} Y_{01} \\ Y_{11}, Y_{12}, \ldots, Y_{1\eta_1} \\ Y_{21}, Y_{22}, \ldots, Y_{2\eta_2} \\ \ldots \\ Y_{\kappa 1}, Y_{\kappa 2}, \ldots, Y_{\kappa\eta_\kappa} \end{gathered} \tag{17}$$

be combinatorial sets that correspond to the nodes of the i-th level of the κ-level tree, $i = 0, 1, \ldots, \kappa$.

Each combinatorial set Y_{ij} of the tree is defined by a finite set β_{ij} of generative elements of Y_{ij}, $i = 0, 1, \ldots, \kappa, j = 1, \ldots, \eta_i$. We denote the number of elements of β_{ij} by n_{ij}, therefore:

$$\eta_0 = 1, \eta_1 = n, \eta_2 = \sum_{j=1}^{\eta_1} n_{2j}, \eta_3 = \sum_{j=1}^{\eta_2} n_{3j}, \cdots, \eta_i = \sum_{j=1}^{\eta_{i-1}} n_{ij}, \eta_\kappa = \sum_{j=1}^{\eta_{\kappa-1}} n_{\kappa j}.$$

The core idea of the *Nested Combinatorial κ-set* is based on relationships between generative elements β_{ij} of each combinatorial set Y_{ij} of *i-th* level and elements of combinatorial sets of *(i + 1)-th* level, using the following correspondence:

$$\beta \in \beta_{ij} \leftrightarrow \left(y_1, y_2, \ldots, y_{n_l}\right) \in Y_{i+1,l}, l \in \{1, \ldots, \eta_{i+1}\}, i \in \{0, 1, \ldots, \kappa - 1\}. \tag{18}$$

A nested combinatorial κ-set is a composed combinatorial set produced by means of the tree structure (17) and correspondence (18) (see [19] for formal definition).

Let us consider an example to make clear the concept. We denote a nested combinatorial κ-set by T_κ.

Example 1 Let T_κ have a two-level structure (17), where Y_{01} is a permutation set $P(a,b)$ generated by $\beta_{01} = \{a, b\}$; Y_{11} is a combination set $C_3^2(c, d, e)$ generated by $\beta_{11} = \{c, d, e\}$; and Y_{12} is a permutation set $P(g,h)$ generated by $\beta_{12} = \{g, h\}$ (see Figure1). Therefore, $n_{01} = 2, n_{11} = 3, n_{12} = 2, \eta_0 = 1, \eta_1 = 2$.

In order to produce the nested combinatorial set T_κ, we replace the generative elements a, b in each element of $P(a,b)$ by each element of the combinatorial sets

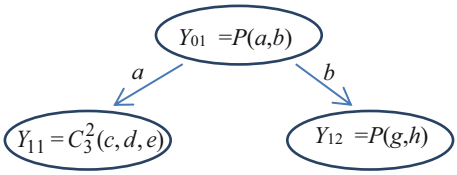


Figure 1 Two-level structure (17) of the nested combinatorial κ-set T_κ

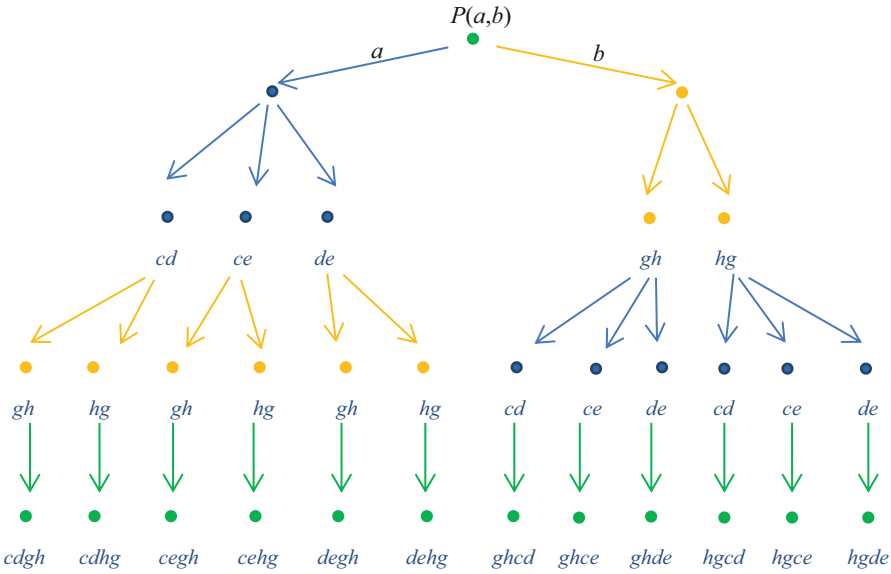


Figure 2 Elements of the combinatorial set T_κ

C_3^2 (c, d, e) and $P(g,h)$ consequently, using correspondence (18) (see Figure 2). Terminal nodes of the tree correspond to elements of the combinatorial set T_κ:*(cdgh), (cdhg), (cegh), (cehg), (degh), (dehg), (ghcd), (ghce), (ghde), (hgcd), (hgce), (hgde).*

Details of algorithm for generating a nested combinatorial κ-set are represented in [25].

Now, we consider an example of the algorithm generating q $(\mathbb{k})$.

Example 2 Let the basic sets $\mathbb{C}_{\mathbb{P}}(5,2)$, $C_5^{k_1}$, $C_{5-k_1}^{k_2}$ generated by elements $\{q_1, q_2, q_3, q_4, q_5\}$ be given, $k_1 + k_2 = n = 5$.

Since $\mathbb{C}_{\mathbb{P}}(5,2) = \left\{\left(k_1^i, k_2^i\right)\right\} = \{(1,4),(2,3),(3,2),(4,1)\}$, then we have:

1. $k_1 = 1,\ \ k_2 = 4$
   $C_5^{k_1} = C_5^1 = \{(q_1),(q_2),(q_3),(q_4),(q_5)\}$
   $C_{5-k_1}^{k_2} = C_4^4 : \{(q_2,q_3,q_4,q_5)\},\{(q_1,q_3,q_4,q_5)\},\{(q_1,q_2,q_4,q_5)\},$
   $\{(q_1,q_2,q_3,q_5)\},\{(q_1,q_2,q_3,q_4)\}$
2. $k_1 = 4,\ \ k_2 = 1$
   $C_5^{k_1} = C_5^4 = \{(q_1,q_2,q_3,q_4),(q_1,q_2,q_3,q_5),(q_1,q_2,q_4,q_5),$
   $(q_1,q_3,q_4,q_5),(q_2,q_3,q_4,q_5)\}$
   $C_{5-k_1}^{k_2} = C_1^1 : \{(q_5)\},\{(q_4)\},\{(q_3)\},\{(q_2)\},\{(q_1)\}$
3. $k_1 = 2,\ \ k_2 = 3$
   $C_5^{k_1} = C_5^2 = \{(q_1,q_2),(q_1,q_3),(q_1,q_4),(q_1,q_5),(q_2,q_3),$
   $(q_2,q_4),(q_2,q_5),(q_3,q_4),(q_3,q_5),(q_4,q_5)\}$
   $C_{5-k_1}^{k_2} = C_3^3 : \{(q_3,q_4,q_5)\},\{(q_2,q_4,q_5)\},\{(q_2,q_3,q_5)\},$
   $\{(q_2,q_3,q_4)\},\{(q_1,q_4,q_5)\},\{(q_1,q_3,q_5)\},\{(q_1,q_3,q_4)\},$
   $\{(q_1,q_2,q_5)\},\{(q_1,q_2,q_4)\},\{(q_1,q_2,q_3)\}$
4. $k_1 = 3,\ \ k_2 = 2$
   $C_5^{k_1} = C_5^3 = \{(q_1,q_2,q_3),(q_1,q_2,q_4),(q_1,q_2,q_5),(q_1,q_3,q_4),(q_1,q_3,q_5),$
   $(q_1,q_4,q_5),(q_2,q_3,q_4),(q_2,q_3,q_5),(q_2,q_4,q_5),(q_3,q_4,q_5)\}$
   $C_{5-k_1}^{k_2} = C_2^2 : \{(q_4,q_5)\},\{(q_3,q_5)\},\{(q_3,q_4)\},\{(q_2,q_5)\},\{(q_2,q_4)\},$
   $\{(q_2,q_3)\},\{(q_1,q_5)\},\{(q_1,q_4)\},\{(q_1,q_3)\},\{(q_1,q_2)\}$

Example 3 We show here the κ-set of compositions of two combinations $C_5^{k_1}, C_{5-k_1}^{k_2}$ generated by elements $\{q_1,q_2,q_3,q_4,q_5\}$, $k_1 + k_2 = n = 5$, using the *Gen_κ-set* algorithm and the results of Example 1. Then, κ=1 and Y_0 is the set of compositions $\mathbb{C}_{\mathbb{P}}(5,2)$, $Y_{11} = C_5^{k_1}$, $Y_{12} = C_{5-k_1}^{k_2}$. Let us present the structure of the κ-set constructed in Example 2.

In the set Y_{01}, the first generating element k_1 will be replaced with the unique element of the set Y_{11}, i.e., with the tuple $\left(q_{i_1}, q_{i_2}, \ldots, q_{i_{k_1}}\right) \in C_5^{k_1}$, and element k_2 with the tuple $\left(q_{j_1}, q_{j_2}, \ldots, q_{j_{k_2}}\right) \in C_5^{k_2}$.

We apply the algorithm presented in [25] to generate all elements of the set $\mathbb{Q}(\mathbb{k})$ (see Table 1). According to (9), the number of elements in the set $\mathbb{Q}(\mathbb{k})$ is equal to 30.

5 Computational Results

Example 4 We consider the problem (15) and (16) for cylinders $\mathbb{C}_i$, $i \in J_n$ that have to be placed into the cylindrical container Ω with one separation plane (circle) in order to minimize the deviation of the center of mass of Ω_A from the given point (x_0, y_0, z_0). Characteristics of cylinders are given in Table 2.

Table 1 Data for Example 3 and appropriate elements of set $\mathbb{Q}(\mathbb{k})$

№	$q \in \mathbb{Q}$	$\mathbb{k}$	$q(\mathbb{k}) \in \mathbb{Q}(\mathbb{k})$
1	$(q_1, q_2, q_3, q_4, q_5)$	(1,4)	$(q_1, \|q_2, q_3, q_4, q_5)$
2	$(q_2, q_1, q_3, q_4, q_5)$	(1,4)	$(q_2, \|q_1, q_3, q_4, q_5)$
3	$(q_3, q_1, q_2, q_4, q_5)$	(1,4)	$(q_3, \|q_1, q_2, q_4, q_5)$
4	$(q_4, q_1, q_2, q_3, q_5)$	(1,4)	$(q_4, \|q_1, q_2, q_3, q_5)$
5	$(q_5, q_1, q_2, q_3, q_4)$	(1,4)	$(q_5, \|q_1, q_2, q_3, q_4)$
6	$(q_1, q_2, q_3, q_4, q_5)$	(4,1)	$(q_1, q_2, q_3, q_4, \|q_5)$
7	$(q_1, q_2, q_3, q_5, q_4)$	(4,1)	$(q_1, q_2, q_3, q_5, \|q_4)$
8	$(q_1, q_2, q_4, q_5, q_3)$	(4,1)	$(q_1, q_2, q_4, q_5, \|q_3)$
9	$(q_1, q_3, q_4, q_5, q_2)$	(4,1)	$(q_1, q_3, q_4, q_5, \|q_2)$
10	$(q_2, q_3, q_4, q_5, q_1)$	(4,1)	$(q_2, q_3, q_4, q_5, \|q_1)$
11	$(q_1, q_2, q_3, q_4, q_5)$	(2,3)	$(q_1, q_2, \|q_3, q_4, q_5)$
12	$(q_1, q_3, q_2, q_4, q_5)$	(2,3)	$(q_1, q_3, \|q_2, q_4, q_5)$
13	$(q_1, q_4, q_2, q_3, q_5)$	(2,3)	$(q_1, q_4, \|q_2, q_3, q_5)$
14	$(q_1, q_5, q_2, q_3, q_4)$	(2,3)	$(q_1, q_5, \|q_2, q_3, q_4)$
15	$(q_2, q_3, q_1, q_4, q_5)$	(2,3)	$(q_2, q_3, \|q_1, q_4, q_5)$
16	$(q_2, q_4, q_1, q_3, q_5)$	(2,3)	$(q_2, q_4, \|q_1, q_3, q_5)$
17	$(q_2, q_5, q_1, q_3, q_4)$	(2,3)	$(q_2, q_5, \|q_1, q_3, q_4)$
18	$(q_3, q_4, q_1, q_2, q_5)$	(2,3)	$(q_3, q_4, \|q_1, q_2, q_5)$
19	$(q_3, q_5, q_1, q_2, q_4)$	(2,3)	$(q_3, q_5, \|q_1, q_2, q_4)$
20	$(q_4, q_5, q_1, q_2, q_3)$	(2,3)	$(q_4, q_5, \|q_1, q_2, q_3)$
21	$(q_3, q_4, q_5, q_1, q_2)$	(3,2)	$(q_3, q_4, q_5, \|q_1, q_2)$
22	$(q_2, q_4, q_5, q_1, q_3)$	(3,2)	$(q_2, q_4, q_5, \|q_1, q_3)$
23	$(q_2, q_3, q_5, q_1, q_4)$	(3,2)	$(q_2, q_3, q_5, \|q_1, q_4)$
24	$(q_2, q_3, q_4, q_1, q_5)$	(3,2)	$(q_2, q_3, q_4, \|q_1, q_5)$
25	$(q_1, q_4, q_5, q_2, q_3)$	(3,2)	$(q_1, q_4, q_5, \|q_2, q_3)$
26	$(q_1, q_3, q_5, q_2, q_4)$	(3,2)	$(q_1, q_3, q_5, \|q_2, q_4)$
27	$(q_1, q_3, q_4, q_2, q_5)$	(3,2)	$(q_1, q_3, q_4, \|q_2, q_5)$
28	$(q_1, q_2, q_5, q_3, q_4)$	(3,2)	$(q_1, q_2, q_5, \|q_3, q_4)$
29	$(q_1, q_2, q_4, q_3, q_5)$	(3,2)	$(q_1, q_2, q_4, \|q_3, q_5)$
30	$(q_1, q_2, q_3, q_4, q_5)$	(3,2)	$(q_1, q_2, q_3, \|q_4, q_5)$

Table 2 Characteristics of cylinders in Example 4

	$\mathbb{C}_1$	$\mathbb{C}_2$	$\mathbb{C}_3$	$\mathbb{C}_4$	$\mathbb{C}_5$
m_i	4	2	1	3	5
r_i	1	0.7	0.45	0.8	0. 9
h_i	1.27	1.3	1.57	1.49	1.96

Let $m = 2, H = 6, R = 2.5, t_1 = 3$ be the parameters characterizing our cylindrical container and $(x_0, y_0, z_0) = (0, 0, 3)$.

The values of the objective function for all $n = 30$ tuples of the partition $q(\mathbb{k}) \in \mathbb{Q}(\mathbb{k})$ and appropriate compositions $\mathbb{k}$ are presented in Table 3.

Figure 3 shows the local optimal placements of cylinders in the two subcontainers found by our algorithm that correspond to the tuples: (a) $q^1(\mathbb{k})$, (b) $q^{18}(\mathbb{k})$ in Example 4.

The best value of the objective function in Example 4 is 0.0003 that corresponds to two tuples $q^1(\mathbb{k}) = (1|\ 2\ 3\ 4\ 5)$ and $q^{18}(\mathbb{k}) = (3\ 4|\ 1\ 2\ 5)$.

Table 3 Output data in Example 4

i	$q\,(\Bbbk) \in \mathbb{Q}\,(\Bbbk)$	$\Bbbk$	Objective function, d
1	(1 |2 3 4 5)	(1,4)	0.0003
2	(2 |1 3 4 5)	(1,4)	0.1475
3	(3 |1 2 4 5)	(1,4)	0.3411
4	(4 |1 2 3 5)	(1,4)	0.0339
5	(5 |1 2 3 4)	(1,4)	0.0467
6	(1 2 3 4| 5)	(4,1)	1.4787
7	(1 2 3 5| 4)	(4,1)	2.6115
8	(1 2 4 5| 3)	(4,1)	4.0643
9	(1 3 4 5| 2)	(4,1)	3.2979
10	(2 3 4 5| 1)	(4,1)	2.0051
11	(1 2| 3 4 5)	(2,3)	0.1731
12	(1 3| 2 4 5)	(2,3)	0.0467
13	(1 4| 2 3 5)	(2,3)	0.3795
14	(1 5| 2 3 4)	(2,3)	1.0323
15	(2 3| 1 4 5)	(2,3)	0.0339
16	(2 4| 1 3 5)	(2,3)	0.0467
17	(2 5| 1 3 4)	(2,3)	0.3795
18	(3 4| 1 2 5)	(2,3)	0.0003
19	(3 5| 1 2 4)	(2,3)	0.1731
20	(4 5| 1 2 3)	(2,3)	0.6659
21	(3 4 5| 1 2)	(3,2)	1.0323
22	(2 4 5| 1 3)	(3,2)	1.4787
23	(2 3 5| 1 4)	(3,2)	0.6659
24	(2 3 4| 1 5)	(3,2)	0.1731
25	(1 4 5| 2 3)	(3,2)	2.6115
26	(1 3 5| 2 4)	(3,2)	1.4787
27	(1 3 4| 2 5)	(3,2)	0.6659
28	(1 2 5| 3 4)	(3,2)	2.0051
29	(1 2 4| 3 5)	(3,2)	1.0323
30	(1 2 3| 4 5)	(3,2)	0.3795

Example 5 Let us consider the problem (13) and (14). Let Ω be a cylindrical container of height $H = 1$ and the basis radius $R = 0.55$. The container has two separation circles. We assume that $t_1 = t_2 = 0.35$. Let $m_0 = 500$ be the mass and $(x_0, y_0, z_0) = (0, 0, 0.5)$ be the center of mass of the cylindrical container Ω.

We consider the collection of 3D-objects of six shapes (Figure 4): $A = \{\mathbb{S}_i, i = 1, \ldots, 6, \mathbb{C}_i, i = 7, \ldots, 13, \mathbb{Q}_i, i = 14...17, \mathbb{S}_{\mathbb{C}i}, i = 18, ...21, \mathbb{P}_i, i = 22, 23, 24c, \mathbb{K}_{25}\}$ with the following characteristics:

$\{m_i, i = 1, .., 25\}$= {20.944, 15.2681, 27.8764, 20.944, 20.944, 34.5575, 63.7115, 41.8146, 30.4106, 6.28319, 20.1062, 31.4159, 28.4245, 49.9649, 24.8714, 38.6888, 26.2637, 20.7764, 17.2159, 16.8756, 52.8, 52.8, 52.8, 23.1489} are the given masses of the 3D-objects;

$r_1 = 0.1$, $r_2 = 0.09$, $r_3 = 0.11$, $r_4 = 0.11$, $r_5 = 0.1$, $r_6 = 0.1$ are the radii of spheres $\mathbb{S}_i, i = 1, \ldots, 6$;

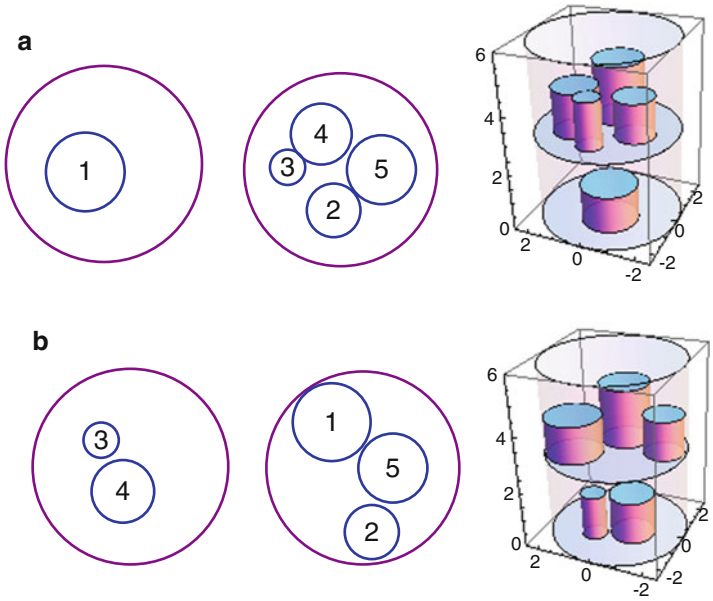


Figure 3 Balanced layout of cylinders and relevant item projections on the base of the container and separation circle corresponding to tuples: (**a**) $q^1(\mathbb{k})$, (**b**) $q^{18}(\mathbb{k})$ in Example 4

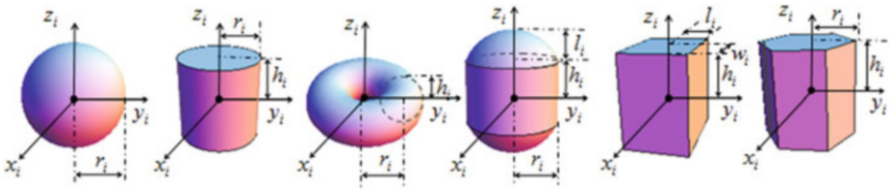


Figure 4 Shapes of objects in Example 5: $\mathbb{S}_i, \mathbb{C}_i, \mathbb{Q}_i, \mathbb{S}_{\mathbb{C}i}, \mathbb{P}_i, \mathbb{K}_i$

$r_7 = 0.1$, $h_7 = 0.11$, $r_8 = 0.13$, $h_8 = 0.12$, $r_9 = 0.11$, $h_9 = 0.11$, $r_{10} = 0.11$, $h_{10} = 0.08$, $r_{11} = 0.05$, $h_{11} = 0.08$, $r_{12} = 0.08$, $h_{12} = 0.1$, $r_{13} = 0.1$, $h_{13} = 0.1$ are the radii and half-heights of cylinders $\mathbb{C}_i$, $i = 7, \ldots, 13$;

$r_{14} = 0.08$, $h_{14} = 0.07$, $r_{15} = 0.09$, $h_{15} = 0.075$, $r_{16} = 0.07$, $h_{16} = 0.06$, $r_{17} = 0.08$, $h_{17} = 0.07$ are the radii of the generating circles and half-heights of tori $\mathbb{Q}_i$, i= 14...17;

$r_{18} = 0.1$, $h_{18} = 0.05$, $l_{18} = 0.07$, $r_{19} = 0.05$, $h_{19} = 0.05$, $l_{19} = 0.08$, $r_{20} = 0.08$, $h_{20} = 0.05$, $l_{20} = 0.06$, $r_{21} = 0.08$, $h_{21} = 0.04$, $l_{21} = 0.07$ are the radii, half-heights of cylinders, and heights of spherical segments for spherocylinders $\mathbb{S}_{\mathbb{C}i}$, i = 18, ...21;

$w_i = 0.11$, $l_i = 0.1$, $h_i = 0.12$, $i = 22, 23, 24$ are the half-widths, half-lengths, and half-heights of cuboids $\mathbb{P}_i$, $i = 22, 23, 24$;

$r_{25} = 0.09$, $h_{25} = 0.11$ are the length of the basis side and half-height of the right hexagonal prism $\mathbb{K}_{25}$.

Table 4 Output data in Example 5

i	$q\,(\mathbb{k}) \in \mathbb{Q}\,(\mathbb{k})$	$\mathbb{k}$	Objective function, d
1	(2,3,6,8,10,14,17,21,24|4,7,11,13,18,20,22,23|1,5,9,12,15,16,19,25)	(9,8,8)	0.002387
2	(3,5,6,7,10,13,17,21|2,4,8,12,14,18,19,22,25|1,9,11,15,16,20,23,24)	(8,9,8)	0.000269
3	(1,5,10,17,19,21,22,25|2,4,6,9,12,13,14,18|3,7,8,11,15,16,20,23,24)	(8,8,9)	0.943362×10^{-7}

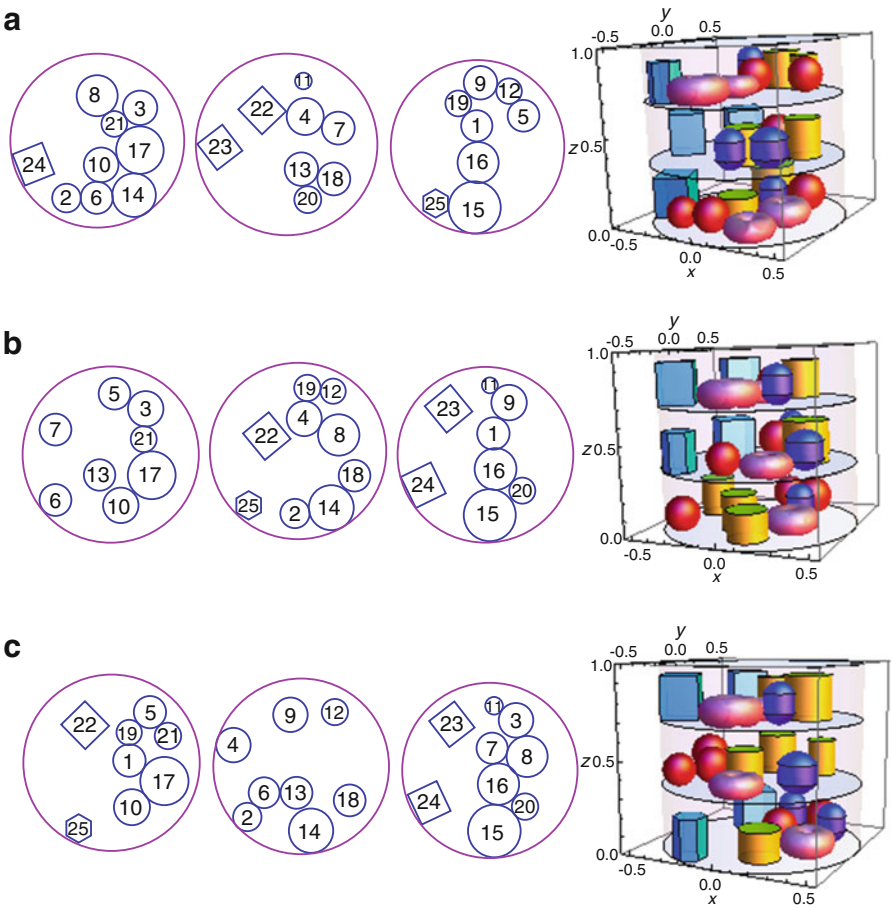


Figure 5 Local optimal placement of 3D-objects and the relative object projections onto the separation circles corresponding to tuples: (**a**) $q^1(\mathbb{k})$, (**b**) $q^2(\mathbb{k})$, (**c**) $q^3(\mathbb{k})$ in Example 5

The values of the objective function for three chosen tuples of the partition $q\,(\mathbb{k}) \in \mathbb{Q}\,(\mathbb{k})$ and appropriate compositions $\mathbb{k}$ are presented in Table 4.

Figure 5 shows the local optimal placement of 3D-objects in subcontainers and the relative object projections onto the separation circles corresponding to tuples: (a) $q^1(\mathbb{k})$, (b) $q^2(\mathbb{k})$, (c) $q^3(\mathbb{k})$ given in Table 4.

The best values found for the objective function in Example 5 is 0.943362×10^{-7} that corresponds to tuple $q^3(\mathbb{k}) = (1,5,10,17,19,21,22,25|2,4,6,9,12,13,14,18|3,7,8,11,15,16,20,23,24)$.

6 Concluding Remarks

This chapter discusses the problem of placing 3D-objects into a container, partitioned into sectors by parallel separation planes, minimizing the distance of the overall center of mass from an assigned position.

The mathematical model formulated for the purpose, on the basis of the phi-function methodology, is illustrated in detail. It takes into account not only the geometrical and balancing constraints, but also the combinatorial features relevant to the assignment of items to sectors.

A solution strategy is provided, which includes the following procedures: generation of partition tuples, based on combinatorial configurations, construction of feasible starting points, and local optimization. This approach implements the multi-start strategy to search for "good" feasible solutions. The results of the numerical experiments show the efficiency of the proposed approach for the class of layout problems considered.

References

1. Stoyan, Y., Pankratov, A., Romanova, T., Fasano, G., Pintér, J., Stoian, Y., Chugay, A., Kovalenko, A.: Optimized packings in space engineering applications—part I. In: Fasano, G., Pintér, J.D. (eds.) Modeling and Optimization in Space Engineering. Springer, New York (2019)
2. Chernov, N., Stoyan, Y., Romanova, T.: Mathematical model and efficient algorithms for object packing problem. Comput. Geom. **43**(5), 535–553 (2010)
3. Stoyan, Y., Romanova, T.: Mathematical models of placement optimisation: Two- and three-dimensional problems and applications. In: Fasano, G., Pinter, J. (eds.) Modeling and Optimization in Space Engineering, Springer Optimization and Its Applications, vol. 73, pp. 363–388. Springer, New York (2012)
4. Fasano, G.: Solving Non-standard Packing Problems by Global Optimization and Heuristics. Springer Briefs in Optimization. Springer, New York (2014)
5. Fasano, G., Pintér, J.D.: Optimized Packings and Their Applications, Springer Optimization and its Applications. Springer, New York (2015)
6. Fasano, G., Pinter, J.: Modeling and Optimization in Space Engineering. Springer, New York (2013)
7. Che, C., Wang, Y., Teng, H.: Test problems for quasi-satellite packing: Cylinders packing with behavior constraints and all the optimal solutions known. Optimization Online (2008.) http://www.optimizationonline.org/DB_HTML/2008/09/2093.html
8. Fasano, G., Pinter, J.: Space engineering. In: Modeling and Optimization with Case Studies, Springer Optimization and its Applications. Springer, New York (2016)

9. Sun, Z., Teng, H.: Optimal layout design of a satellite module. Eng. Opt. **35**(5), 513–530 (2003)
10. Lei, K.: Constrained layout optimization based on adaptive particle swarm optimizer. In: Zhihua, C., Zhenhua, L., Zhuo, K., Yong, L. (eds.) Advances in Computation and Intelligence, Series 1, pp. 434–442. Springer, Heidelberg (2009)
11. Kovalenko, A., Romanova, T., Stetsyuk, P.: Balance layout problem for 3D-objects: mathematical model and solution methods. Cybern. Syst. Anal. **51**(4), 556–565 (2015)
12. Stetsyuk, P., Romanova, T., Scheithauer, G.: On the global minimum in a balanced circular packing problem. Opt. Lett. **10**, 1347–1360 (2016)
13. Stoyan, Y., Romanova, T., Pankratov, A., Kovalenko, A., Stetsyuk, P.: Modeling and optimization of balance layout problems. In: Fasano, G., Pinter, J. (eds.) Space Engineering. Modeling and Optimization with Case Studies, Springer Optimization and its Applications, vol. 114, pp. 369–400. Springer, New York (2016)
14. Hulianytskyi, L., Riasna, I.: Formalization and classification of combinatorial optimization problems. In: Butenko, S., Pardalos, P., Shylo, V. (eds.) Optimization Methods and Applications, pp. 239–250. Springer, New York (2017)
15. Papadimitriou, C., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Courier Corporation (1998)
16. Yakovlev, S., Pichugina, O.: Properties of combinatorial optimization problems over polyhedral-spherical sets. Cybern. Syst. Anal. **54**(1), 111–123 (2018)
17. Reingold, E., Nievergelt, J., Deo, N.: Combinatorial Algorithms: Theory and Practice. Pearson Education, North York, ON (1977)
18. Sachkov, V.: Combinatorial Methods in Discrete Mathematics, 1st edn. Cambridge University Press, Cambridge (1996)
19. Stoyan, Y., Grebennik, I.: Description and generation of combinatorial sets having special characteristics. Int. J. Biomed. Soft Comput. Hum. Sci. **18**(1), 83–88 (2013)
20. Grebennik, I.: Description and generation of permutations containing cycles. Cybern. Syst. Anal. **46**(6), 945–952 (2010)
21. Knuth, D.: The Art of Computer Programming, **4**(2): Generating All Tuples and Permutations. Addison-Wesley, Boston (2005)
22. Kreher, D., Stinson, D.: Combinatorial Algorithms: Generation, Enumeration and Search. CRC Press, Boca Raton, FL (1999)
23. Ruskey, F.: Combinatorial Generation, Department of Computer Science, University of Victoria, Canada, 1j-CSC 425/20 (2003)
24. Grebennik, I., Kovalenko, A., Romanova, T., Urniaieva, I., Shekhovtsov, S.: Combinatorial configurations in balance layout optimization problems. Cybern. Syst. Anal. **54**(2), 55–67 (2018)
25. Grebennik, I., Lytvynenko, O.: Generating combinatorial sets with given properties. Cybern. Syst. Anal. **48**(6), 890–898 (2012)
26. Stoyan, Y., Pankratov, A., Romanova, T., Chugay, A.: Optimized object packings using quasi-phi-functions. In: Fasano, G., Pinter, J. (eds.) Optimized Packings and Their Applications, Springer Optimization and its Applications, vol. 105, pp. 265–291. Springer, New York (2015)
27. Stoyan, Y., Pankratov, A., Romanova, T.: Quasi phi-functions and optimal packing of ellipses. J. Glob. Optim. **65**(2), 283–307 (2016)
28. Wachter, A., Biegler, L.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**(1), 25–57 (2006)

A Catalogue of Parametric Time-Optimal Transfers for All-Electric GEO Satellites

Francesco Topputo and Simone Ceccherini

Abstract In this chapter a catalogue of time-optimal low-thrust transfers from arbitrary departure orbits to the geostationary orbit is constructed. This catalogue is obtained by solving a multitude of optimal control problems with a combination of simple and multiple shooting techniques, augmented by a multi-dimensional homotopy. Modified equinoctial elements are used to describe the satellite dynamics, and state-of-the-art values for thrust and specific impulse are considered. The ultimate outcome consists of a synthetic law for transfer time, and thus cost, as function of the orbit injection parameters and engine figures. This law can be consulted in the early stages of mission design.

1 Introduction

Geostationary Earth orbit (GEO) satellites for communication have the lion's share in the space market. This sector being the most profitable and competitive, brand-new, efficient technologies are required. Low-thrust propulsion, which is usually obtained by exploiting solar electric propulsion (SEP) systems, is an appealing solution because of its high specific impulses. Nowadays, SEP is a mature technology, and thus novel mission concepts can be envisaged.

All-electric GEO satellites are more efficient because they maximize the payload mass, or equivalently minimize the mass at launch, by involving a lower fuel consumption. The first all-electric GEO platform, Boeing 702SP, has been launched in 2015. However, all-electric satellites suffer from very high transfer times because of low control accelerations. The growth in signal throughput of GEO satellites entails also heavier payloads and satellites, which further lower the thrust-to-mass ratios.

F. Topputo (✉) · S. Ceccherini
Department of Aerospace Science and Technology, Politecnico di Milano, Milano, Italy
e-mail: francesco.topputo@polimi.it; simone.ceccherini@polimi.it

G. Fasano, J. D. Pintér (eds.), *Modeling and Optimization in Space Engineering*, Springer Optimization and Its Applications 144,
https://doi.org/10.1007/978-3-030-10501-3_17

In principle, modifying the approach to reach the GEO may involve searching for new transfer orbits. The traditional geostationary transfer orbit (GTO) and low Earth orbit (LEO) are not the only possible choices anymore. New injection orbits have been used by Boeing 702SP, i.e., super-synchronous transfer orbit (SSTO) with *apogee* radius higher than the standard one, and the GTO+ proposed by ArianeSpace [2], with a *perigee* altitude greater than the usual one (250–350 km). The concept of hybrid transfer (transfers that combine chemical and electric propulsion) involves subdividing the transfer orbit into a high-thrust and a low-thrust segment, with variable switching orbits [8, 9, 20].

The literature on low-thrust transfer to GEO is vast [7, 11, 12, 17–19]. Mission analysis studies to preliminary assess GEO transfers should be based on fast and reliable tools to evaluate the main trajectory performances, one of which is the time of flight. Furthermore, evaluating SEP transfers for various injection orbits, which can also be noncircular, provides the customer with a wide range of solutions. Consequently, trade-off analyses to design GEO missions can be performed by comparing launchers capability, i.e., injected mass vs achieved orbit, and transfer duration.

In this chapter, a number of minimum-time solutions to the GEO are obtained. These are characterized by having different departure orbits. The optimal control problem is solved by exploiting the indirect optimization method, whereas the satellite dynamics is modeled by using modified equinoctial elements (MEE) [10]. The two-point boundary value problem (TPBVP) is analyzed with SEP performances for Hall effect thruster class [13], which is characterized by thrust-to-mass ratio of 10^{-4}–10^{-3} m/s^2 and specific impulse of 1500–2500 s. Single and multiple shooting methods are properly combined with numerical continuation methods on thrust and orbital parameters. A database of solutions has been built and a compact law has been spun off. This law can be used in the early stages of mission design for fast evaluation of transfer time and, thus, cost.

2 Background

2.1 Dynamics Model

The orbital dynamics is formulated using MEE [10]. If compared to Cartesian coordinates, MEE guarantee good numerical stability when dealing with low-thrust transfers [16]. Moreover, when MEE are used, the boundary conditions of orbit-to-orbit transfers can be expressed as fixed values for some elements of the initial and final states, rather than nonlinear functions of them.

The relations between MEE (left-hand side) and the classical orbital elements (right-hand side) are

$$\begin{aligned} p &= a\left(1-e^2\right) \\ e_x &= e\cos\left(\omega+\Omega\right) \\ e_y &= e\sin\left(\omega+\Omega\right) \\ h_x &= \tan\left(i/2\right)\cos\left(\Omega\right) \\ h_y &= \tan\left(i/2\right)\sin\left(\Omega\right) \\ L &= \omega+\Omega+\theta \end{aligned} \tag{1}$$

where a is the semi-major axis, e the eccentricity, i the orbital inclination, Ω the right ascension of the ascending node, ω the argument of perigee, θ the true anomaly, p is the semilatus rectum, and L the true longitude.

The complete dynamics shall consider the variation of MEE as well as the variation of satellite mass and control law parameters. Let the augmented state vector be $\boldsymbol{x} = \left[p, e_x, e_y, h_x, h_y, L, m\right]^\top$, where m is the satellite mass. The control, $\boldsymbol{u}$, is made up of the thrust throttling factor, $f \in [0, 1]$, and the thrust pointing angle $\boldsymbol{\alpha}$; $\boldsymbol{u} := [\boldsymbol{\alpha}, f]^\top$.

Accordingly, the equations of motions take the form

$$\dot{\boldsymbol{x}} = \boldsymbol{F}\left(\boldsymbol{x},\ \boldsymbol{u}\right) := \begin{bmatrix} f\dfrac{T}{m}\boldsymbol{B}\boldsymbol{\alpha} + \boldsymbol{A} \\ -\dfrac{T}{I_{sp}\ g_0}f \end{bmatrix} \tag{2}$$

where $\boldsymbol{A} = [0, 0, 0, 0, 0, \kappa]^\top$ and

$$\boldsymbol{B} = \begin{bmatrix} 0 & \frac{2p}{\upsilon}\sqrt{\frac{p}{\mu}} & 0 \\ \sqrt{\frac{p}{\mu}}\sin L & \sqrt{\frac{p}{\mu}}[(\upsilon+1)\cos L + e_x]\frac{1}{\upsilon} & -\sqrt{\frac{p}{\mu}}[h_x\sin L - h_y\cos L]\frac{e_y}{\upsilon} \\ -\sqrt{\frac{p}{\mu}}\cos L & \sqrt{\frac{p}{\mu}}[(\upsilon+1)\sin L + e_y]\frac{1}{\upsilon} & \sqrt{\frac{p}{\mu}}[h_x\sin L - h_y\cos L]\frac{e_x}{\upsilon} \\ 0 & 0 & \sqrt{\frac{p}{\mu}}\frac{s^2}{2\upsilon}\cos L \\ 0 & 0 & \sqrt{\frac{p}{\mu}}\frac{s^2}{2\upsilon}\sin L \\ 0 & 0 & \frac{1}{\upsilon}\sqrt{\frac{p}{\mu}}(h_x\sin L - h_y\cos L) \end{bmatrix} \tag{3}$$

with

$$\upsilon = 1 + e_x\cos L + e_y\sin L, \quad s^2 = 1 + h_x^2 + h_y^2, \quad \kappa = \sqrt{\mu p}\left(\frac{\upsilon}{p}\right)^2 \tag{4}$$

The thrust pointing unit vector is

$$\boldsymbol{\alpha} = [\alpha_r, \alpha_\theta, \alpha_h]^\top \tag{5}$$

Table 1 Boundary conditions for the state vector (G: Given, F: Free)

	p	e_x	e_y	h_x	h_y	L	m
t_0	G	G	G	G	G	G	G
t_f	G	G	G	G	G	F	F

where α_r, α_θ, and α_h are the components of the control vector along the radius vector outwards, in-plane perpendicular to the radius vector, and normal to the orbital plane in the direction of the angular momentum, respectively. These three directions define the local vertical-local horizontal (LVLH) frame. In Equation (2), T is the maximum thrust, I_{sp} the specific impulse, g_0 the gravity acceleration at sea level, and μ the Earth gravitational parameter.

In order to construct a catalogue of time-optimal, all-electric transfers to the Geostationary orbit, the problem is to find the functions $f(t)$ and $\boldsymbol{\alpha}(t)$, $t \in [t_0, t_f]$ with t_f unknown, that plugged into Equation (2) produce solutions satisfying the boundary conditions. These are qualitatively specified in Table 1: the state is fully given at t_0, while it is partially specified at t_f. It is worth observing that specifying L, and thus the departure point, narrows the search space since, in principle, there might exist trajectory that slightly outperform those reported hereinafter. However, in pursuance to standardize the presentation of all-electric solutions to GEO, the initial true anomaly has been fixed, and in turn the longitude L through Equation (1).

From Table 1, 12 boundary conditions are known. Since the problem will be solved as a pure TPBVP (i.e., the control will be expressed in terms of seven states and seven costates), two additional conditions as well as one equation for t_f are needed. These are found by virtue of the optimal control theory. In this work, the optimal trajectories are found by using low-thrust trajectory optimizer (LT2O), a tool developed at Politecnico di Milano. This implements a root-finding algorithm with mixed shooting-/multiple-shooting scheme and a variable step explicit predictor-corrector seventh to eighth order Runge–Kutta integrator [14, 25]. Notice that eclipses and non-spherical gravity perturbations are not modeled in this context.

2.2 *Trajectory Optimization*

Indirect optimization methods are the key feature of the problem under investigation. In minimum-time problems, a solution of Equation (2) minimizes

$$J = \int_{t_0}^{t_f} L \, \mathrm{d}t := \int_{t_0}^{t_f} 1 \, \mathrm{d}t \tag{6}$$

Without any loss of generality, the initial time t_0 is set to 0. The Hamiltonian of the problem is [3]

$$H = \boldsymbol{\lambda}^\top \boldsymbol{F} + L := \lambda_L \kappa + f \frac{T}{m} \boldsymbol{\lambda}_{\mathrm{mee}}^\top \boldsymbol{B} \boldsymbol{\alpha} - \lambda_m f \frac{T}{I_{sp} g_0} + 1, \tag{7}$$

where $\boldsymbol{\lambda} = \left[\boldsymbol{\lambda}_{\text{mee}}^{\top}, \lambda_m\right]^{\top}$ is the vector of costates, with $\boldsymbol{\lambda}_{\text{mee}} = \left[\lambda_p, \lambda_{e_x}, \lambda_{e_y}, \lambda_{h_x}, \lambda_{h_y}, \lambda_L\right]^{\top}$. The evolution of $\boldsymbol{\lambda}$ is governed by

$$\dot{\boldsymbol{\lambda}} = \boldsymbol{G}\left(\boldsymbol{x},\ \boldsymbol{\lambda},\ \boldsymbol{u}\right) := -\frac{\partial H}{\partial \boldsymbol{x}} \tag{8}$$

The boundary conditions for the costate depend on those imposed to the state. At t_0 the state is fully given (Table 1), therefore $\boldsymbol{\lambda}(t_0)$ is unknown. At t_f, L and m are free (Table 1), thus

$$\lambda_L\left(t_f\right) = 0, \quad \lambda_m\left(t_f\right) = 0 \tag{9}$$

whereas the other five components of $\boldsymbol{\lambda}(t_f)$ are unknown.

It is now needed to express $f(t)$ and $\boldsymbol{\alpha}(t)$ as functions of $\boldsymbol{x}$ and $\boldsymbol{\lambda}$. This is done by applying the Pontryagin maximum principle [24], which states that $f(t)$ and $\boldsymbol{\alpha}(t)$ are such that H in Equation (7) is extremized (minimized in this case). Because fT/m is always semi-positive, minimizing H means choosing a thrust pointing direction as

$$\boldsymbol{\alpha}^* = -\frac{\boldsymbol{B}^{\top}\boldsymbol{\lambda}_{\text{mee}}}{||\boldsymbol{B}^{\top}\boldsymbol{\lambda}_{\text{mee}}||} \tag{10}$$

Substituting (10) into Equation (7) yields

$$H = 1 + \lambda_L \kappa - f\frac{T}{I_{sp}g_0}\left[\lambda_m + \frac{I_{sp}g_0}{m}\frac{\boldsymbol{\lambda}_{\text{mee}}^{\top}\boldsymbol{B}\boldsymbol{B}^{\top}\boldsymbol{\lambda}_{\text{mee}}}{||\boldsymbol{B}^{\top}\boldsymbol{\lambda}_{\text{mee}}||}\right] \tag{11}$$

and therefore the value of f depends upon the sign of

$$S = \lambda_m + \frac{I_{sp}g_0}{m}\frac{\boldsymbol{\lambda}_{\text{mee}}^{\top}\boldsymbol{B}\boldsymbol{B}^{\top}\boldsymbol{\lambda}_{\text{mee}}}{||\boldsymbol{B}^{\top}\boldsymbol{\lambda}_{\text{mee}}||} \tag{12}$$

That is, $f = 1$ if $S > 0$ and $f = 0$ if $S < 0$ (f is unspecified in case $S = 0$). The second term in Equation (12) is always positive. Moreover, it is easy to show that λ_m is positive as well: $\dot{\lambda}_m = -\partial H/\partial m < 0$ and $\lambda_m(t_f) = 0$ from (9). Therefore, $f^* = 1$ for all times. Eventually, t_f^* is found through the transversality condition [3], i.e., $H(t_f^*) = 0$.

2.2.1 Statement of the Problem

Since $\boldsymbol{x}_0$ is fully given, the problem consists in finding $\{\boldsymbol{\lambda}_0^*, t_f^*\}$ such that

$$\begin{bmatrix}\boldsymbol{x}(t_f^*)\\ \boldsymbol{\lambda}(t_f^*)\end{bmatrix} = \boldsymbol{\varphi}\left(\begin{bmatrix}\boldsymbol{x}_0\\ \boldsymbol{\lambda}_0^*\end{bmatrix}, t_0; t_f^*\right) \tag{13}$$

satisfies the five conditions on $\boldsymbol{x}(t_f^*)$ (Table 1), the two conditions on $\boldsymbol{\lambda}(t_f^*)$ in Equation (9) as well as $H(t_f^*) = 0$. In Equation (13), $\boldsymbol{\varphi}$ is the flow of

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}(\boldsymbol{x},\ \boldsymbol{\lambda}) \\ \boldsymbol{G}(\boldsymbol{x},\ \boldsymbol{\lambda}) \end{bmatrix} \tag{14}$$

The right-hand side in Equation (14) is the one in Equations (2) and (8) with $f^* = 1$ and $\boldsymbol{\alpha}^*$ in Equation (10).

2.3 Numerical Implementation

The TPBVP defined in Sect. 2.2.1 is solved numerically using shooting methods. A multi-dimensional Newton-based gradient method has been implemented in LT20. The strategy implemented to solve the problem at hand exploits a combination of single and multiple shooting methods. Moreover, as finding an initial guess for $\boldsymbol{\lambda}_0^*$ that assures convergence is difficult, a multi-dimensional homotopy (i.e., numerical continuation) has been used as well.

The objective of the continuation is to approximate, in an efficient way, the trend of a curve that maps a well-defined quantity, say ϵ, from a simple problem (ϵ_0) to the original problem (ϵ_f), called zero path of that mapping. It is highlighted that the existence of this zero path cannot be guaranteed as it depends on the family of problems under study. In accordance to the zero path, discrete continuation is the preferred option as it has been proven effective in optimal trajectory problems [6, 16, 22]. The advantages of this continuation are in its simplicity and in the fact that the smoothness of the zero path is not required. In fact, it can be seen as a predictor-corrector method.

The considered paths for guessing the initial costate vector and the final time are two: zero-order (ZO) and linear paths. The zero-path predictor method is used for approximating the guess at the following step (ϵ_{i+1}) in the continuation processes. In a ZO method, the solution of the ith step (ϵ_i^*) is the prediction for ϵ_{i+1}. In the present case

$$\boldsymbol{\lambda}_0(\epsilon_{i+1}) = \boldsymbol{\lambda}_0^*(\epsilon_i) \tag{15}$$

The same approach is used when t_f is guessed. The continuation parameter, ϵ, can be either the maximum thrust value (T) or a pair of orbital elements (R_p and R_a, see Sect. 3), so yielding to an overall three-dimensional continuation.

In [23], it has been shown that thrust continuation in minimum-time problems with MEE fails because initial costates impose a too-low time of flight. The solved minimum-time transfer with an improved method led always to an extra orbital revolution that permitted the homotopic procedure to continue. In [5], it was found in a heuristic way that $t_f \times T \approx$ const for low thrust-to-mass authority.

To this purpose, a linear continuation path is used to guess t_f when thrust continuation is exploited:

$$T_{i+1} \times t_f(T_{i+1}) = T_i \times t_f^*(T_i) \tag{16}$$

where $t_f^*(T_i)$ is the optimal transfer time when using a thrust magnitude T_i. Moreover, for low thrust-to-mass ratios and high eccentricities of the starting orbit, the convergence of TPBVP was found out to be very hard to succeed [22]. Consequently, a linear predictor path has been defined in [22] for guessing the initial costate vector. For this analysis, the continuation method implemented in LT20 is

$$\boldsymbol{\lambda}_0(T_{i+1}) = \boldsymbol{\lambda}_0^*(T_i) + \frac{T_{i+1} - T_i}{T_i - T_{i-1}} \left(\boldsymbol{\lambda}_0^*(T_i) - \boldsymbol{\lambda}_0^*(T_{i-1})\right), \tag{17}$$

where two previous (solved) solutions are used ($\boldsymbol{\lambda}_0^*(T_i)$ and $\boldsymbol{\lambda}_0^*(T_{i-1})$). Quadratic or cubic predictions could also be used but numerical experiments [22] have showed that they do not bring significant improvements with respect to the results obtained with the linear approach.

2.4 Search Space for All-Electric GEO Transfers

In Sect. 1, it has been outlined how the low-thrust propulsion paved the way to a paradigm shift on how to reach the GEO, especially for all-electric satellites. New injection orbits were used, so widening the trade space in GEO mission analysis. In this context, a multitude of parametric TPBVP from various starting orbits to GEO are solved for different initial thrust-to-mass ratios and specific impulses.

The selected starting orbits are defined by the grid in Figure 1, where each point corresponds to a combination of perigee and apogee radii, R_p and R_a, respectively. Choosing R_p and R_a sets the semi-major axis and the eccentricity of the starting orbits through $a = (R_p + R_a)/2$ and $e = (R_a - R_p)/(R_a + R_P)$. The inclination is set to 5ř, which is compatible with launches from Kourou. Since the perturbations are not considered, and so the natural motion is axial-symmetric, both Ω and ω are set to zero without any loss of generality. All transfers start from the apogee, i.e., $\theta = 180$ř. This stems from both numerical experiment and preliminary considerations on operations. The numerical values for R_p and R_a are given in the Appendix. It is convenient to keep in mind the triangular shape of the search space in Figure 1 because the results will be superimposed over this region.

3 Methodology

Each transfer derived from the grid in Figure 1 is analyzed for various accelerations and specific impulses. The idea lies in combining simple and multiple shooting with numerical continuation. This is performed by varying the maximum thrust T as well as R_p and R_a, so resulting in a three-dimensional continuation. I_{sp} is kept constant during continuations. The regions of interest for T/m_0 and I_{sp} are 10^{-4}–10^{-3} m/s^2 and 1500–2500 s, respectively. These correspond to values achievable with current technology for primary low-thrust propulsion. The procedure implemented to solve for all combinations of thrust values and departure orbits is described hereinafter.

1. The first step is fixing the specific impulse, say $I_{sp} = 2000$ s, and selecting an initial circular orbit close to the GEO region, e.g., the black dot in Figure 2a ($R_p = R_a = 38{,}492.31$ km). Then, the TPBVP is solved for $T/m_0 = 1 \times 10^{-3}$ m/s^2 (upper bound). To this purpose, multiple shooting and numerical continuation on T with ZO predictor for $\boldsymbol{\lambda}_0(T)$ and $t_f(T)$ is performed starting from a higher T/m_0, e.g., 5×10^{-3} m/s^2. Referring to this solution, the initial guesses for λ_0 and t_f are random values generated from the standard normal distribution, with absolute value applied to λ_m. Figure 3 shows the components of the thrust direction $\boldsymbol{\alpha}$ in the LVLH frame for $T/m_0 = 1 \times 10^{-3}$ m/s^2.

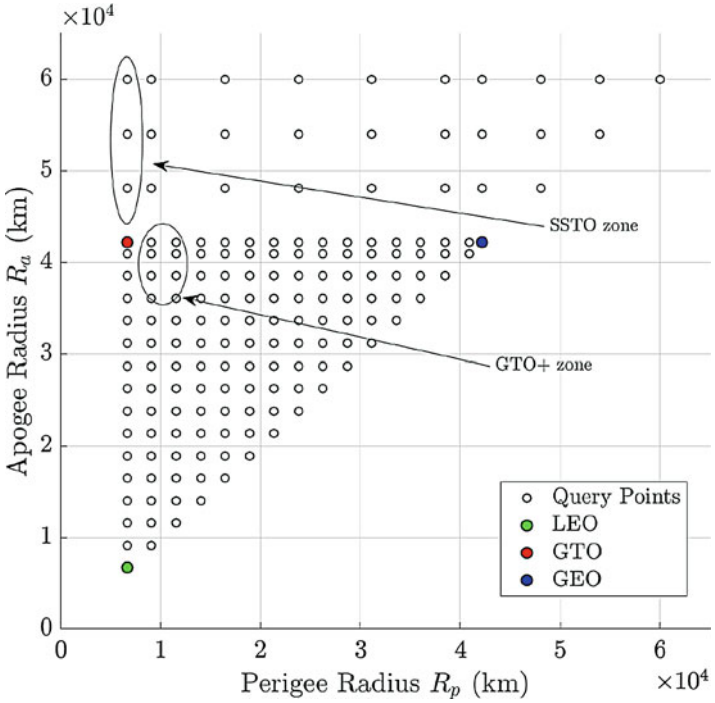


Figure 1 Starting orbit search grid for GEO transfers

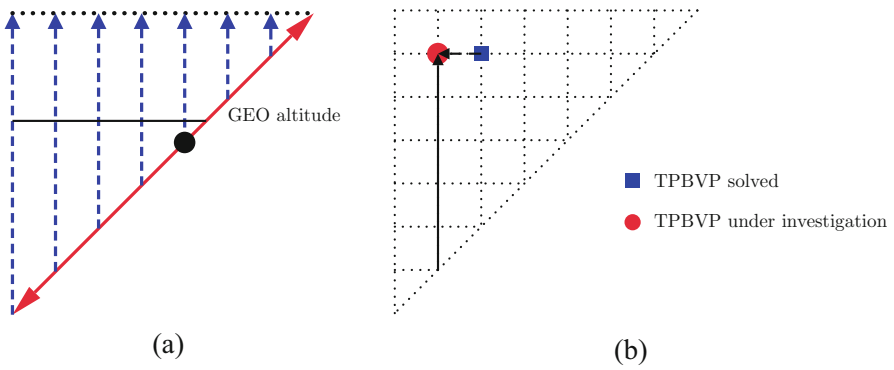


Figure 2 Numerical continuation on orbital parameters. (**a**) Strategy for numerical continuation on orbital parameters. (**b**) Strategy to overcome issues of orbital parameters continuation along the perigee radius

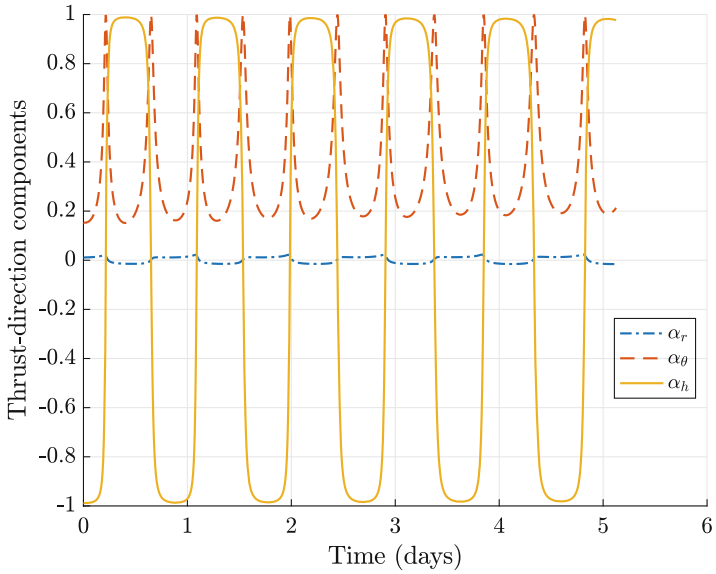


Figure 3 Thrust-direction components in the LVLH frame for $T/m_0 = 1 \times 10^{-3}\,\mathrm{m/s^2}$

2. The next step consists in solving TPBVP for orbits that lie on the null-eccentricity line, red line in Figure 2a. This approach has been selected because solving problems where the initial orbits are circular has been found easier than those using elliptic orbits, in agreement with [11, 21]. Solving the null-eccentricity line is done with homotopy fed by ZO predictor and multiple shooting, with $T/m_0 = 1 \times 10^{-3}\,\mathrm{m/s^2}$. The proposed continuation technique uses the initial costate and transfer time from the previously solved problem as first guess for the next problem: $\boldsymbol{\lambda}_0(a_{i+1}) = \boldsymbol{\lambda}_0^*(a_i)$ and $t_f(a_{i+1}) = t_f^*(a_i)$, with $a_i = (R_{p,i} + R_{a,i})/2$

$a_{i+1} = (R_{p,i+1} + R_{a,i+1})/2$. This approach is repeated for circular orbits with radii both greater and lower than the GEO one.

3. Once the solutions on the zero-eccentricity line are obtained, in order to fill the grid, a continuation along the apogee radius is performed. The perigee radius is maintained constant, while the apogee radius is increased from bottom to top, as represented by the blue-dashed arrows in Figure 2a. One important remark is that the bottom-up approach should be followed arrow by arrow, that is, from the innermost part (near GEO) to the outer parts of the grid. This path is necessary to overcome possible convergence problems in solving the TPBVP when continuing orbits featuring changes in both semi-major axis and eccentricity. If this condition occurs, based on the idea of grid shooting [22], the first guessed costate is initialized with the solution of the already-solved optimal trajectory with the previous perigee, as depicted in Figure 2b.
4. The last step consists in using data of the converged problems at $T/m_0 = 1 \times 10^{-3}\,\mathrm{m/s^2}$ to initiate a number of TPBVP until $T/m_0 = 1 \times 10^{-4}\,\mathrm{m/s^2}$ is reached. To this end, linear order predictors in Equations (16)–(17) are used to guess final time and initial costate vector. Moreover, both multiple and simple shooting are used. Specifically, in [4] it is highlighted that when multiple shooting and continuation are coupled, the convergence degenerates as the problems become more difficult (lower T/m_0). Properly, a strategy able to automatically change from multiple to single shooting has been developed and implemented in LT20.

   The time-histories of thrust-direction angles and orbital parameters for a relevant case are given in Figure 4. This optimal solution has a duration of 241.96 days and performs nearly 354 orbital revolutions. From Figure 5a, it can be seen

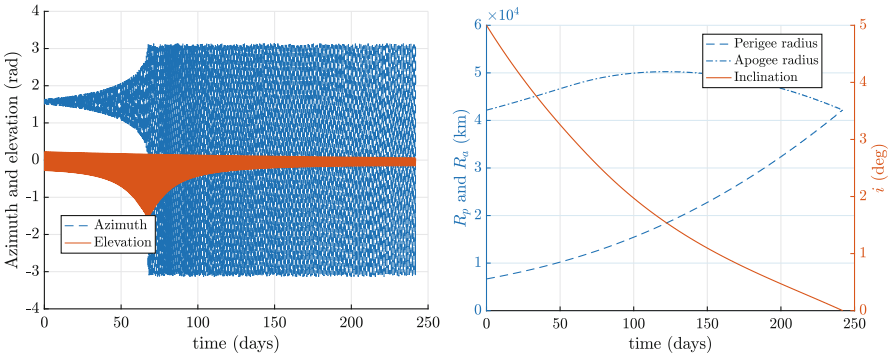


Figure 4 Solution of the TPBVP for a relevant case: $T/m_0 = 1 \times 10^{-4}\,\mathrm{m/s^2}$, $I_{sp} = 2000\,\mathrm{s}$, $R_p = 6671.00\,\mathrm{km}$, $R_a = 42{,}164.00\,\mathrm{km}$. Azimuth and elevation are the in-plane and out-of-plane angle between the thrust and α_r and α_h, respectively. (**a**) Time-history of thrust azimuth and elevation in the LVLH reference frame. (**b**) Time-history of perigee radius, apogee radius, and inclination

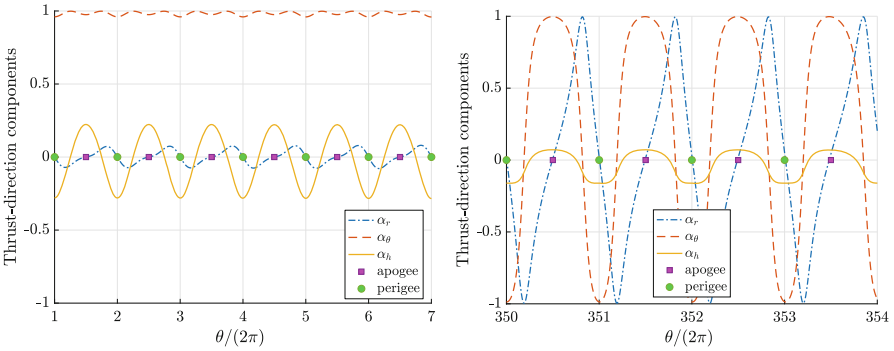


Figure 5 Thrust-direction components during first and last orbital revolutions of the solution in Figure 4. (**a**) First few revolutions. (**b**) Last few revolutions

that during the first revolutions α_θ is always positive, i.e., the semi-major axis increases, α_r is close to zero at the apogee and perigee radii ($\theta = k\pi$), whereas α_h is zero when the true anomaly is in the vicinity of $k\pi/2$. During the time of last few revolutions (Figure 5b), Figure 5b α_θ is negative near the perigee and positive near the apogee, whereas α_r is positive near $\theta = 3/2\pi$ and negative at $\theta = \pi/2$. According to these directions, R_a decreases and R_p increases, which is confirmed by the rightmost trend in Figure 4b.

4 Results

The results obtained with the aforementioned procedure involve solving more than 150 TPBVP for each combination of the parameters, whose ranges are given below

$$\begin{aligned} I_{sp} &= \{1500, 2000, 2500\}\ \text{s} \\ T/m_0 &\in \left[1 \times 10^{-4}, 1 \times 10^{-3}\right]\ \text{m/s}^2 \\ R_p &\in [6671, 60, 000]\ \text{km} \\ R_a &\in [6671, 60, 000]\ \text{km} \end{aligned} \tag{18}$$

Figure 6 outlines time of flight and the initial costate vector components for an initial thrust-to-mass ratio of 10^{-4} m/s^2 and a specific impulse of 2000 s.

The parametric study of a variety of minimum-time problems in which the initial orbit, the control acceleration, and the specific impulse vary enables synthesizing a concise law for a rapid estimation of transfer time. This may be convenient in the first stages of the mission design. Based on the expression in [15], the curve fitting function

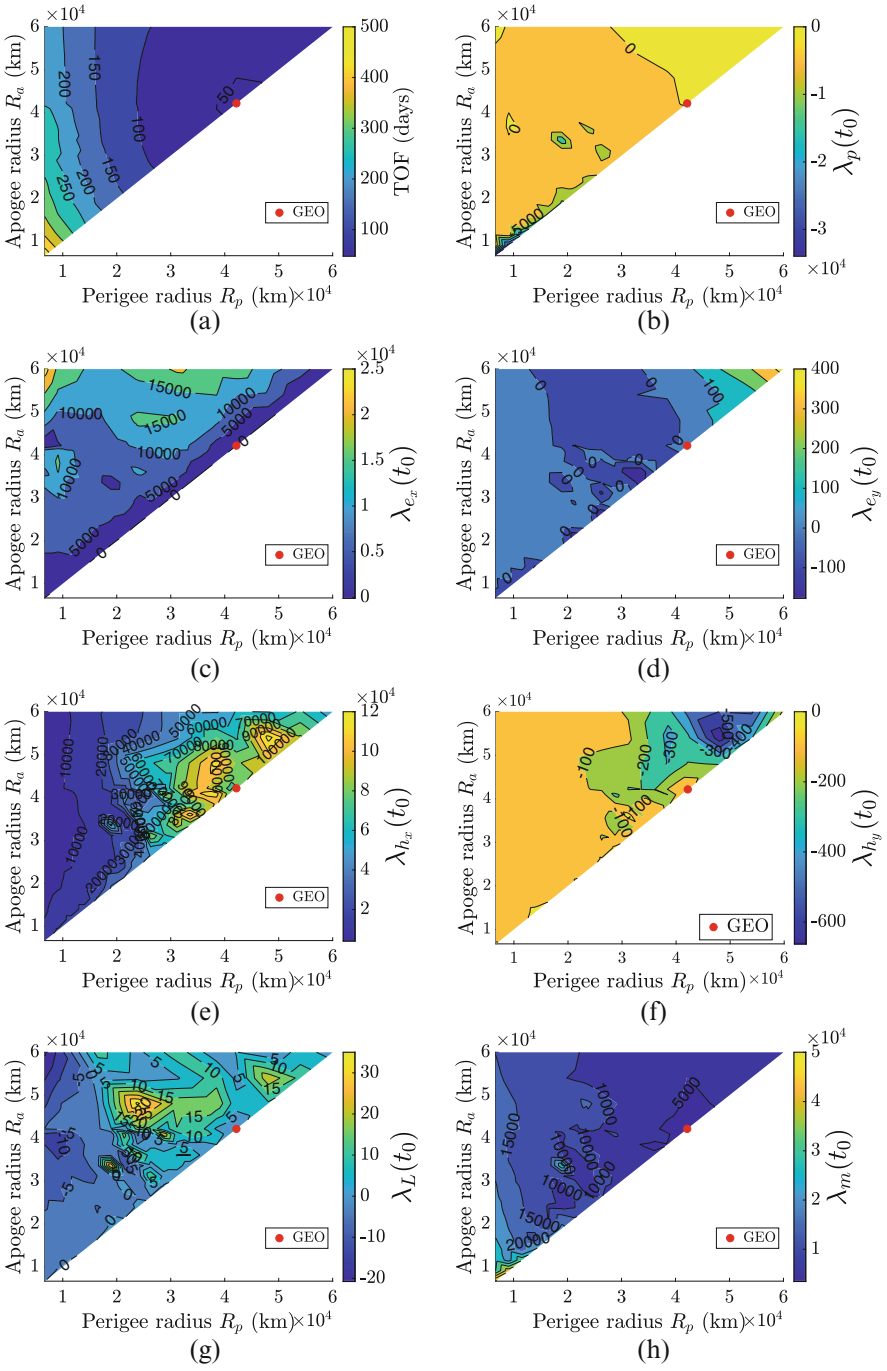


Figure 6 Transfer time and initial costate components for $T/m_0 = 1 \times 10^{-4}\,\mathrm{m/s^2}$ and $I_{sp} = 2000\,\mathrm{s}$, superimposed to the search grid in Figure 1. (**a**) Time of flight on the search grid. (**b**) $\lambda_p(t_0)$ on the search grid. (**c**) $\lambda_{e_x}(t_0)$ on the search grid. (**d**) $\lambda_{e_y}(t_0)$ on the search grid. (**e**) $\lambda_{h_x}(t_0)$ on the search grid. (**f**) $\lambda_{h_y}(t_0)$ on the search grid. (**g**) $\lambda_L(t_0)$ on the search grid. (**h**) $\lambda_m(t_0)$ on the search grid

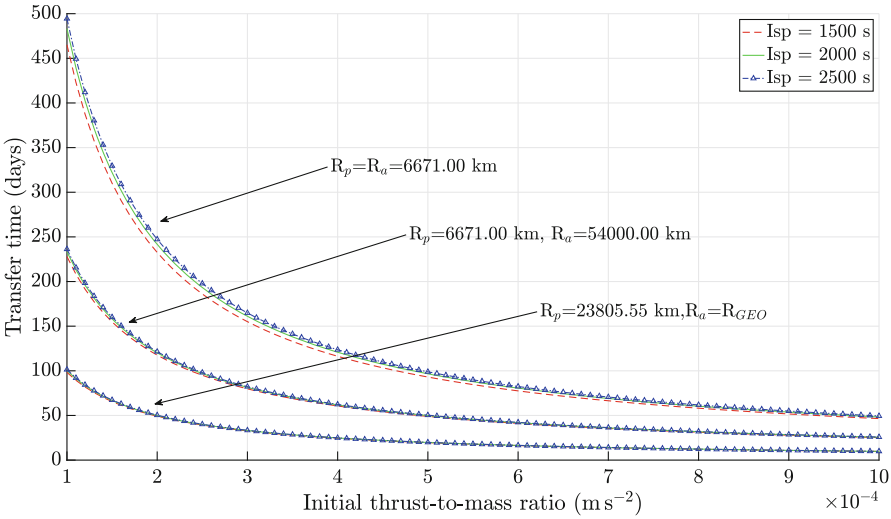


Figure 7 Estimated transfer duration τ_f vs T/m_0 for three representative initial orbits evaluated with Equation (19)

$$\tau_f = F\left(\frac{T}{m_0},\ I_{sp}\right) := a\left(\frac{T}{m_0}\right)^b (I_{sp})^c + d\left(\frac{T}{m_0}\right)^e \tag{19}$$

is proposed for a fast estimation of time-to-GEO. In Equation (19), the duration τ_f is in days, whereas T/m_0 and I_{sp} are in m/s^2 and s, respectively. It is worth specifying that Equation (19) holds for T/m_0 and I_{sp} within the bounds in Equation (18).

The curve fitting coefficients (a, b, c, d, e) are listed in the Appendix (Tables 2, 3, 4 and 5) for every combination of $\{R_p, R_a\}$ considered. The quality with which the model in Equation (19) fits the computed data, or the coefficient of determination R^2, is also given in [15]. This is a value between 0 and 1: the closer R^2 is to 1, the better the data fitting process is, which is the case in the present fitting. Figure 7 shows the trend of τ_f vs T/m_0 obtained with Equation (19) for three representative initial orbits and three values of I_{sp} each.

4.1 Launcher Performances

Concerning new injection orbits for all-electric GEO satellites, the knowledge produced with the catalogue can be coupled with launcher performances. A simplified model of Ariane 64 payload performances for 5ř inclination injection orbits has been constructed by combining (1) the data of Ariane 5 ECA payload for varying apogee altitudes [1], and (2) the Ariane 64 payload performances in GTO [2]. The mass performances are computed considering the following:

1. The mass capability of the launcher is computed for a fixed perigee altitude of 250 km and the desired apogee altitude h_a through

$$\tilde{m} = G(h_a), \tag{20}$$

where G is the best curve to payload capacity vs apogee injection data in [1], which have been adapted for Ariane 64 payload capability in [2].
2. Then, it has been assumed that the payload mass is delivered with a single impulsive apogee maneuver, if needed, to achieve the planned perigee altitude, h_p. That maneuver takes into account the structural mass ($m_{s,LV}$) of the upper stage of Ariane 6 and the specific impulse of Vinci engine ($I_{sp,V}$) [2]. No gravity losses have been considered. That procedure is represented by

$$m_{pl} = F\left(h_p, \tilde{m}(h_a), m_s, LV, I_{sp,V}\right). \tag{21}$$

In Equation (21) the function F represents the Rocket equation used to study the maneuver to achieve the desired injection orbit (h_p, h_a) from the orbit characterized by $h_p = 250$ km and variable h_a.

The map resulting from the combination of the low-thrust database and the modeled Ariane 6 payload performances is shown in Figure 8 for $T/m_0 =$

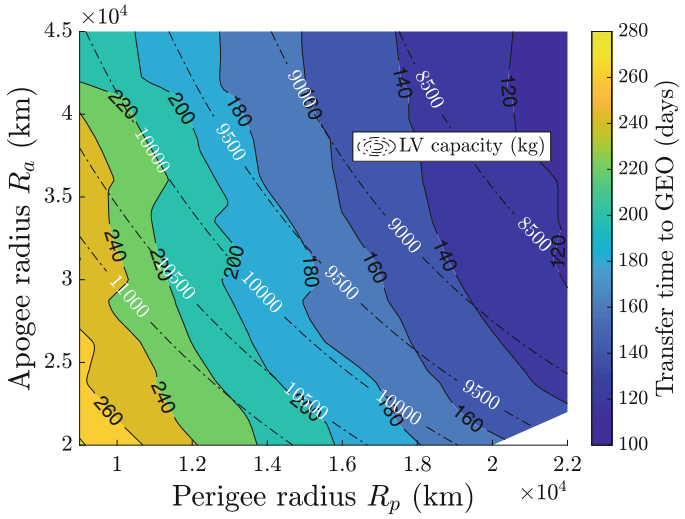


Figure 8 Ariane 64 payload performances referred to Equations (20)–(21) (dash-dot lines) and all-electric transfers times to reach GEO

$10^{-4}\,\mathrm{m/s^2}$ and $I_{sp} = 2000\,\mathrm{s}$. Superimposing m_{pl} on τ_f is useful for trade-off analyses in designing all-electric GEO satellites. For instance, two platforms with $m_0 = 5000\,\mathrm{kg}$ each and equipped with $T = 0.5\,\mathrm{N}$ would reach the GEO in ~240 days, if they had injected onto an orbit identified with $R_p = 1\times 10^3\,\mathrm{km}$ and $R_a = 30 \times 10^3\,\mathrm{km}$ through a shared launch with Ariane 64, according to Equations (20)–(21).

5 Conclusions

A catalogue of low-thrust transfers to the GEO has been constructed for variable initial orbit. From a computational point of view, this has been obtained by combining simple and multiple shooting techniques with a multi-dimensional homotopy in maximum thrust value and orbital parameters. This analysis is useful in preliminary mission design, when a rapid estimation of the transfer time for given engine parameters is needed. The multitude of solutions have been synthesized in a compact expression, and the coefficients of the curve fittings have been given. This allows an independent evaluation and use of the catalogue.

Acknowledgements The authors would like to acknowledge Natividad Ramos Morón whose MSc Thesis inspired the work described in this chapter and Juan Luis Gonzalo for reviewing LT2O and making it more efficient.

Appendix

Tables 2, 3, 4 and 5 display the curve fitting parameters referred to Equation (19) and the relative coefficients of determination R^2.

Table 2 Table 1 of 4, referring to Equation (19)

R_p (km)	R_a (km)	a	b	c	d	e	R^2
6671.00	6671.00	−3.991046	−1.000010	−0.843221	0.054904	−0.999969	9.99999999E-01
6671.00	9118.79	−8.181741	−1.034037	−1.022528	0.046174	−1.00494	9.99993407E-01
9118.79	9118.79	−2.916757	−0.999729	−0.872420	0.041737	−0.99994	9.99999999E-01
6671.00	11,566.59	−5.854503	−1.063703	−1.056649	0.040853	−1.00543	9.99995080E-01
9118.79	11,566.59	−0.286098	−1.020437	−0.504813	0.040299	−1.00513	9.99989810E-01
11,566.59	11,566.59	−2.234149	−0.999761	−0.898462	0.033076	−0.99995	9.99999998E-01
6671.00	14,014.38	−0.463467	−1.081282	−0.724548	0.037445	−1.00915	9.99985326E-01
9118.79	14,014.38	−0.758313	−0.944652	−0.639640	0.037357	−0.99438	9.99990847E-01
11,566.59	14,014.38	−14.985666	−0.988839	−1.172638	0.030640	−0.99853	9.99981486E-01
14,014.38	14,014.38	−1.629516	−1.000988	−0.912325	0.026850	−1.00003	9.99999997E-01
6671.00	16,462.17	−0.260781	−1.066490	−0.609559	0.036488	−1.00789	9.99967907E-01
9118.79	16,462.17	−0.328186	−1.012066	−0.604713	0.032999	−1.00274	9.99975314E-01
11,566.59	16,462.17	−8.912786	−1.066689	−1.192966	0.027126	−1.00565	9.99994247E-01
14,014.38	16,462.17	−19.999929	−0.779836	−1.097810	0.026610	−0.98980	9.99849469E-01
16,462.17	16,462.17	−1.306779	−1.000348	−0.934799	0.022118	−1.00000	9.99999993E-01
6671.00	18,909.97	−0.229448	−1.013929	−0.515459	0.039075	−0.99633	9.99971433E-01
9118.79	18,909.97	−19.999997	−0.971001	−1.243398	0.031761	−0.99209	9.99967415E-01
11,566.59	18,909.97	−19.999961	−1.027149	−1.364913	0.026367	−0.99745	9.99948082E-01
14,014.38	18,909.97	−13.807303	−0.972575	−1.290448	0.022350	−1.00132	9.99987921E-01
16,462.17	18,909.97	−19.999877	−1.031859	−1.371524	0.020084	−1.00166	9.99988707E-01
18,909.97	18,909.97	−0.843471	−0.997845	−0.918818	0.018428	−0.99986	9.99999990E-01
6671.00	21,357.76	−19.999995	−1.019445	−1.279931	0.034361	−0.99415	9.99938661E-01
9118.79	21,357.76	−0.111109	−1.040626	−0.405862	0.031672	−1.01088	9.99957078E-01
11,566.59	21,357.76	−19.999997	−1.110769	−1.413072	0.024952	−1.00017	9.99942931E-01
14,014.38	21,357.76	−9.834767	−1.180966	−1.499999	0.022112	−0.99685	9.99799765E-01
16,462.17	21,357.76	−19.999872	−0.954116	−1.347515	0.021912	−0.98333	9.99655834E-01
18,909.97	21,357.76	−0.010915	−1.385625	−0.840951	0.015385	−1.01088	9.99992979E-01
21,357.76	21,357.76	−0.841522	−0.999438	−0.974641	0.015345	−0.99996	9.99999976E-01
6671.00	23,805.55	−19.999992	−1.084119	−1.334473	0.033322	−0.99474	9.99818289E-01
9118.79	23,805.55	−17.169553	−0.981408	−1.266108	0.028825	−0.99237	9.99935195E-01
11,566.59	23,805.55	−19.999981	−0.948089	−1.245729	0.026339	−0.98881	9.99859530E-01
14,014.38	23,805.55	−7.314150	−1.007785	−1.207490	0.021063	−0.99954	9.99993998E-01
16,462.17	23,805.55	−3.616529	−1.350117	−1.500000	0.015988	−1.01918	9.99353989E-01
18,909.97	23,805.55	−13.002576	−1.138722	−1.445269	0.015182	−1.00901	9.99965849E-01
21,357.76	23,805.55	−0.263571	−1.113462	−0.960356	0.013599	−1.00540	9.99996463E-01
23,805.55	23,805.55	−0.529181	−1.003440	−0.962038	0.012832	−1.00013	9.99999948E-01
6671.00	26,253.35	−19.999928	−1.082261	−1.386742	0.032068	−0.99249	9.99873153E-01
9118.79	26,253.35	−19.999999	−1.193243	−1.422892	0.023959	−1.01871	9.99885388E-01
11,566.59	26,253.35	−2.120151	−1.221961	−1.142554	0.019847	−1.02715	9.99961343E-01
14,014.38	26,253.35	−0.072406	−0.996893	−0.520879	0.019962	−1.00480	9.99977076E-01
16,462.17	26,253.35	−10.237115	−1.150737	−1.500000	0.016064	−1.00952	9.99892045E-01
18,909.97	26,253.35	−10.693368	−1.188191	−1.499999	0.013285	−1.01906	9.99310326E-01
21,357.76	26,253.35	−0.881420	−1.315672	−1.359048	0.011627	−1.01785	9.99975526E-01
23,805.55	26,253.35	−4.402509	−1.114956	−1.436880	0.011491	−1.00311	9.99997608E-01
26,253.35	26,253.35	−0.392741	−1.001073	−0.967640	0.010754	−1.00002	9.99999868E-01

Table 3 Table 2 of 4, referring to Equation (19)

R_p (km)	R_a (km)	a	b	c	d	e	R^2
6671.00	28,701.14	−19.252728	−1.013391	−1.217822	0.032973	−0.98930	9.99751421E-01
9118.79	28,701.14	−0.106716	−0.970614	−0.586964	0.026826	−0.99553	9.99902948E-01
11,566.59	28,701.14	−0.757259	−1.244339	−1.013497	0.018625	−1.03368	9.99925272E-01
14,014.38	28,701.14	−0.189119	−1.001908	−0.787181	0.019803	−0.99669	9.99985573E-01
16,462.17	28,701.14	−17.469488	−1.042063	−1.499997	0.018810	−0.98617	9.99864492E-01
18,909.97	28,701.14	−10.029476	−1.029983	−1.321809	0.014888	−1.00047	9.99979326E-01
21,357.76	28,701.14	−0.105422	−0.749197	−0.168272	0.024730	−0.95022	9.99859194E-01
23,805.55	28,701.14	−0.439287	−1.005902	−0.978352	0.010727	−1.00579	9.99984121E-01
26,253.35	28,701.14	−0.211340	−0.981339	−0.872754	0.009994	−0.99987	9.99998347E-01
28,701.14	28,701.14	−0.552132	−0.987460	−1.054395	0.009010	−0.99992	9.99999608E-01
6671.00	31,148.93	−19.999689	−0.898166	−1.084664	0.038725	−0.96846	9.99374031E-01
9118.79	31,148.93	−19.999962	−1.050101	−1.213991	0.028510	−0.99640	9.99748765E-01
11,566.59	31,148.93	−19.999739	−1.021687	−1.398421	0.023110	−0.99188	9.99878473E-01
14,014.38	31,148.93	−14.945587	−1.089936	−1.499999	0.019257	−0.99640	9.99952072E-01
16,462.17	31,148.93	−5.476535	−1.171533	−1.499994	0.017385	−0.99174	9.99844911E-01
18,909.97	31,148.93	−0.036488	−1.470189	−0.838586	0.006352	−1.12302	9.98134523E-01
21,357.76	31,148.93	−0.106096	−0.891143	−0.343145	0.017727	−0.98351	9.99779042E-01
23,805.55	31,148.93	−20.000000	−0.970824	−1.499001	0.010955	−0.99771	9.99956314E-01
26,253.35	31,148.93	−0.226040	−0.834384	−0.684042	0.009985	−0.99490	9.99992354E-01
28,701.14	31,148.93	−0.088447	−1.058277	−0.895293	0.008266	−1.00155	9.99997828E-01
31,148.93	31,148.93	−0.139364	−1.026937	−0.946550	0.007553	−1.00068	9.99999202E-01
6671.00	33,596.72	−19.996343	−0.965228	−1.211926	0.035499	−0.97271	9.99258600E-01
9118.79	33,596.72	−17.965538	−1.228183	−1.500000	0.023743	−1.00850	9.99722201E-01
11,566.59	33,596.72	−0.141316	−1.181712	−0.603589	0.017381	−1.05613	9.99761018E-01
14,014.38	33,596.72	−14.384997	−1.120162	−1.500000	0.019824	−0.99141	9.99729836E-01
16,462.17	33,596.72	−0.416156	−1.011585	−0.835429	0.016675	−0.99796	9.99983416E-01
18,909.97	33,596.72	−0.033959	−1.134293	−0.579102	0.013011	−1.01627	9.99962495E-01
21,357.76	33,596.72	−0.035516	−1.092315	−0.702629	0.011969	−1.00212	9.99987126E-01
23,805.55	33,596.72	−3.063466	−1.156435	−1.499968	0.010173	−1.00287	9.99988343E-01
26,253.35	33,596.72	−3.199121	−0.698918	−0.935494	0.009697	−0.99302	9.99983348E-01
28,701.14	33,596.72	−0.018023	−1.150781	−0.787328	0.007766	−1.00329	9.99991502E-01
31,148.93	33,596.72	−0.115833	−0.939050	−0.816579	0.007174	−0.99894	9.99997534E-01
33,596.72	33,596.72	−0.136354	−1.023852	−0.984502	0.006433	−1.00070	9.99997219E-01
6671.00	36,044.52	−0.690101	−1.090952	−0.800237	0.029813	−1.00094	9.99565004E-01
9118.79	36,044.52	−19.999951	−1.150750	−1.373473	0.021814	−1.02356	9.99869366E-01
11,566.59	36,044.52	−11.832291	−1.208439	−1.500000	0.023450	−0.98755	9.99117930E-01
14,014.38	36,044.52	−0.026870	−0.993289	−0.503163	0.018408	−0.99972	9.99975308E-01
16,462.17	36,044.52	−19.998362	−0.695534	−1.380771	0.015506	−0.99929	9.99983271E-01
18,909.97	36,044.52	−0.084653	−1.034717	−0.751890	0.013443	−1.00311	9.99974375E-01
21,357.76	36,044.52	−0.040146	−1.120819	−0.700543	0.011501	−1.00666	9.99984798E-01
23,805.55	36,044.52	−0.019509	−1.257495	−0.876000	0.009735	−1.00687	9.99988740E-01
26,253.35	36,044.52	−0.219817	−0.797128	−0.649992	0.009325	−0.99617	9.99983763E-01
28,701.14	36,044.52	−0.135267	−0.809397	−0.665236	0.008011	−0.99638	9.99990460E-01
31,148.93	36,044.52	−5.445805	−1.000406	−1.499672	0.006685	−1.00104	9.99992702E-01
33,596.72	36,044.52	−0.065212	−0.981615	−0.837635	0.006109	−1.00006	9.99995334E-01
36,044.52	36,044.52	−0.024994	−1.042492	−0.783320	0.005598	−1.00171	9.99995162E-01

Table 4 Table 3 of 4, referring to Equation (19)

R_p (km)	R_a (km)	a	b	c	d	e	R^2
6671.00	38,492.31	−0.348795	−1.013454	−0.651181	0.030916	−0.99414	9.99554845E-01
9118.79	38,492.31	−16.584450	−1.233690	−1.500000	0.020711	−1.02476	9.99767267E-01
11,566.59	38,492.31	−19.999992	−1.023206	−1.390062	0.023209	−0.98685	9.99721183E-01
14,014.38	38,492.31	−16.679462	−0.919975	−1.499999	0.018136	−0.99584	9.99936159E-01
16,462.17	38,492.31	−19.999938	−1.011070	−1.473319	0.015596	−0.99904	9.99960640E-01
18,909.97	38,492.31	−0.035706	−1.167463	−0.674906	0.011692	−1.02320	9.99939747E-01
21,357.76	38,492.31	−13.290839	−0.786575	−1.200232	0.012082	−0.99616	9.99988709E-01
23,805.55	38,492.31	−2.736925	−1.213461	−1.499922	0.009782	−1.00478	9.99985935E-01
26,253.35	38,492.31	−0.391198	−0.804362	−0.754504	0.009214	−0.99565	9.99984999E-01
28,701.14	38,492.31	−0.703057	−1.267555	−1.499923	0.007409	−1.00255	9.99983187E-01
31,148.93	38,492.31	−1.054701	−0.847843	−1.083561	0.006683	−0.99900	9.99992787E-01
33,596.72	38,492.31	−0.990928	−0.807715	−1.041271	0.006001	−0.99822	9.99994488E-01
36,044.52	38,492.31	−0.066634	−1.152356	−1.092012	0.005321	−1.00229	9.99993246E-01
38,492.31	38,492.31	−3.699864	−0.904451	−1.397947	0.005168	−0.99936	9.99991076E-01
6671.00	40,940.10	−0.035542	−1.083244	−0.402956	0.029418	−1.00349	9.98793426E-01
9118.79	40,940.10	−0.142287	−1.135132	−0.591276	0.021730	−1.03375	9.99666998E-01
11,566.59	40,940.10	−10.567847	−1.219865	−1.499999	0.022208	−0.99167	9.99221130E-01
14,014.38	40,940.10	−0.087181	−0.959339	−0.521021	0.019469	−0.99199	9.99909129E-01
16,462.17	40,940.10	−0.633631	−1.630517	−1.499992	0.007064	−1.10329	9.98420843E-01
18,909.97	40,940.10	−1.815872	−0.674847	−0.466860	0.030907	−0.92755	9.99735803E-01
21,357.76	40,940.10	−4.388515	−0.607448	−0.803780	0.013188	−0.98646	9.99940909E-01
23,805.55	40,940.10	−1.248293	−1.261255	−1.500000	0.009794	−1.00310	9.99908239E-01
26,253.35	40,940.10	−2.940299	−1.132725	−1.499854	0.008520	−1.00263	9.99984399E-01
28,701.14	40,940.10	−19.999788	−0.665353	−1.282104	0.007604	−0.99915	9.99986468E-01
31,148.93	40,940.10	−19.999926	−0.585670	−1.189512	0.006804	−0.99663	9.99990770E-01
33,596.72	40,940.10	−1.956137	−1.081781	−1.499830	0.005790	−1.00128	9.99992553E-01
36,044.52	40,940.10	−0.018358	−1.213477	−0.977251	0.005126	−1.00549	9.99988654E-01
38,492.31	40,940.10	−0.079342	−1.065331	−1.032129	0.004982	−1.00171	9.99992818E-01
40,940.10	40,940.10	−0.152776	−1.076519	−1.164397	0.004837	−1.00180	9.99986525E-01
6671.00	42,164.00	−0.039125	−0.984013	−0.128465	0.040937	−0.99558	9.98529803E-01
9118.79	42,164.00	−0.052329	−1.091262	−0.323109	0.023345	−1.04235	9.98558278E-01
11,566.59	42,164.00	−0.044839	−1.011233	−0.409351	0.023970	−0.98976	9.99544708E-01
14,014.38	42,164.00	0.000000	−0.500000	−1.500000	0.017536	−0.99932	9.98671285E-01
16,462.17	42,164.00	−0.353670	−1.282916	−1.130619	0.014095	−1.01298	9.99933550E-01
18,909.97	42,164.00	−0.025890	−1.227461	−0.617243	0.011893	−1.02595	9.99805536E-01
21,357.76	42,164.00	−0.327134	−0.866326	−0.733690	0.012124	−0.99649	9.99981596E-01
23,805.55	42,164.00	−0.015756	−1.379087	−0.876003	0.008410	−1.02595	9.99940998E-01
26,253.35	42,164.00	−0.252191	−0.753729	−0.638009	0.009341	−0.99383	9.99976682E-01
28,701.14	42,164.00	−0.246522	−0.790721	−0.711171	0.007962	−0.99585	9.99981660E-01
31,148.93	42,164.00	−4.025674	−0.919167	−1.366240	0.006572	−1.00101	9.99987400E-01
33,596.72	42,164.00	−0.126108	−0.904337	−0.851004	0.005952	−0.99936	9.99991369E-01
36,044.52	42,164.00	−0.324140	−1.033721	−1.191696	0.005285	−1.00178	9.99985868E-01
38,492.31	42,164.00	−0.135008	−0.757545	−0.725060	0.005208	−0.99679	9.99989375E-01
40,940.10	42,164.00	−0.006110	−1.452674	−1.149747	0.004708	−1.00465	9.99988154E-01

Table 5 Table 4 of 4, referring to Equation (19)

R_p (km)	R_a (km)	a	b	c	d	e	R^2
6671.00	48,082.00	−6.697292	−1.331006	−1.500000	0.026655	−0.99559	9.99054091E-01
9118.79	48,082.00	−19.997043	−1.066198	−1.441095	0.027124	−0.97503	9.99615530E-01
16,462.17	48,082.00	−0.804244	−1.263392	−1.353306	0.013943	−1.00748	9.99968990E-01
23,805.55	48,082.00	−0.086398	−0.997675	−0.670929	0.010393	−1.00184	9.99976974E-01
31,148.93	48,082.00	−0.022853	−1.072941	−0.740816	0.006834	−1.00407	9.99995965E-01
38,492.31	48,082.00	−0.169001	−1.053232	−1.108350	0.005327	−1.00092	9.99993934E-01
42,164.00	48,082.00	−0.027551	−1.072560	−0.856241	0.005003	−1.00176	9.99992933E-01
48,082.00	48,082.00	−0.109343	−0.914173	−0.872699	0.005129	−0.99930	9.99996728E-01
6671.00	54,000.00	−19.999998	−1.120246	−1.441381	0.033669	−0.96501	9.97939149E-01
9118.79	54,000.00	−6.112870	−1.202850	−1.500000	0.024936	−0.98210	9.99314462E-01
16,462.17	54,000.00	−0.144031	−0.912894	−0.568368	0.015927	−0.99646	9.99996747E-01
23,805.55	54,000.00	−0.220075	−1.054398	−0.906094	0.010437	−1.00318	9.99992938E-01
31,148.93	54,000.00	−0.093665	−1.072959	−0.939195	0.007522	−1.00262	9.99997865E-01
38,492.31	54,000.00	−0.112091	−0.986687	−0.923650	0.006123	−0.99983	9.99994383E-01
42,164.00	54,000.00	−3.360923	−0.863195	−1.296165	0.005780	−0.99889	9.99992975E-01
48,082.00	54,000.00	−0.510286	−0.966344	−1.145309	0.005625	−0.99990	9.99997289E-01
54,000.00	54,000.00	−0.118844	−1.033063	−1.008212	0.005968	−0.99987	9.99999268E-01
6671.00	60,000.00	−12.061653	−1.130091	−1.499998	0.032474	−0.96662	9.97296588E-01
9118.79	60,000.00	−16.189659	−1.162350	−1.499999	0.024816	−0.98372	9.99308501E-01
16,462.17	60,000.00	−0.026715	−0.920935	−0.286679	0.017089	−0.99351	9.99991984E-01
23,805.55	60,000.00	−10.673870	−0.896819	−1.316881	0.011159	−0.99889	9.99995773E-01
31,148.93	60,000.00	−0.332560	−1.097507	−1.144686	0.008344	−1.00224	9.99995560E-01
38,492.31	60,000.00	−0.108121	−1.011227	−0.909389	0.007087	−1.00083	9.99987085E-01
42,164.00	60,000.00	−0.008138	−1.258684	−0.855102	0.006491	−1.00500	9.99987349E-01
48,082.00	60,000.00	−2.389354	−1.087228	−1.499799	0.006438	−1.00091	9.99992688E-01
54,000.00	60,000.00	−0.112651	−1.062657	−0.997274	0.006613	−1.00063	9.99997459E-01
60,000.00	60,000.00	−0.416432	−1.008036	−1.108419	0.007092	−0.99957	9.99999416E-01

References

1. Arianespace: Ariane 5 User's Manual (Issue 3 - Revision 0) (2000)
2. Arianespace: Ariane 6 User's Manual (Issue 0 - Revision 0) (2016)
3. Bryson, A., Ho, Y.: Applied Optimal Control. Wiley, New York (1975)
4. Caillau, J.B.: Contribution à l'étude du contrôle en temps minimal des transferts orbitaux. Ph.D. dissertation, Institut National Polytechnique de Toulouse (2000)
5. Caillau, J.B., Daoud, B.: Minimum time control of the restricted three-body problem. SIAM J. Control Optim. **50**, 3178–3202 (2012)
6. Caillau, J.B., Noailles, J.: Coplanar control of a satellite around the Earth. ESAIM Control Optim. Calc. Var. **6**, 239–258 (2001)

7. Casalino, L., Colasurdo, G.: Improved Edelbaum's approach to optimize low earth/geostationary orbits low-thrust transfers. J. Guid. Control Dyn. **30**, 1504–1511 (2007)
8. Ceccherini, S., Topputo, F.: System-trajectory optimization of hybrid transfers to the geostationary orbit. In: Proceedings of 2018 Space Flight Mechanics Meeting, AIAA SciTech Forum, Kissimmee, FL, 8–12 January 2018
9. Ceccherini, S., Ferella, L., Topputo, F.: Assessment of hybrid propulsion for geostationary transfer orbit: a mission design approach. In: Proceedings of the 67th International Astronautical Congress (IAC 2015), Guadalajara, 26–30 September 2016
10. Cefola, P.J.: Equinoctial orbit elements–application to artificial satellite orbits. In: Proceedings of AIAA/AAS Astrodynamics Conference, AIAA 72-937, Palo Alto, CA, 11–12 September 1972
11. Edelbaum, T.N.: Propulsion requirements for controllable satellites. ARS J. **31**, 1079–1089 (1961)
12. Edelbaum, T., Sackett, L., Malchow, H.: Optimal low thrust geocentric transfer. In: Proceedings of AIAA 10th Electric Propulsion Conference, AIAA 73-1074, Lake Tahoe, NV, October 31–November 2 1973
13. Goebel, D.M., Katz, I.: Fundamentals of Electric Propulsion: Ion and Hall thrusters. Wiley, New York (2008)
14. Gonzalo, J.L., Topputo, F., Armellin, R.: Indirect optimization of end-of-life disposal for Galileo constellation using low thrust propulsion. In: Proceedings of the 26th International Symposium on Space Flight Dynamics, Matsuyama, 3–9 June 2017
15. Graham, K.F., Rao, A.V.: Minimum-time trajectory optimization of multiple revolution low-thrust Earth-orbit transfers. J. Spacecr. Rockets **52**, 711–727 (2015)
16. Haberkorn, T., Martinon, P., Gergaud, J.: Low thrust minimum-fuel orbital transfer: a homotopic approach. J. Guid. Control Dyn. **27**, 1046–1060 (2004)
17. Kechichian, J.A.: Optimal low-earth-orbit-geostationary-earth-orbit intermediate acceleration orbit transfer. J. Guid. Control Dyn. **20**, 803–811 (1997)
18. Kechichian, J.A.: Reformulation of Edelbaum's low-thrust transfer problem using optimal control theory. J. Guid. Control. Dyn. **20**, 988–994 (1997)
19. Kluever, C.A.: Using Edelbaum's method to compute low-thrust transfers with earth-shadow eclipses. J. Guid. Control Dyn. **34**, 300–303 (2001)
20. Kluever, C.A.: Designing transfers to geostationary orbit using combined chemical-electric propulsion. J. Spacecr. Rockets **52**, 1144–1151 (2015)
21. Kluever, C.A., Oleson, S.R.: Direct approach for computing near-optimal low-thrust Earth-orbit transfers. J. Spacecr. Rockets **35**, 509–515 (1998)
22. Martinon, P., Gergaud, J.: SHOOT2.0: an indirect grid shooting package for optimal control problems, with switching handling and embedded continuation. Research Report 7380, INRIA (2010)
23. Pan, B., Lu, P., Pan, X., Ma, Y.: Double-homotopy method for solving optimal control problems. J. Guid. Control Dyn. **39**, 1706–1720 (2016)
24. Pontryagin, L.: Mathematical Theory of Optimal Processes, pp. 1–114. Interscience, New York (1962)
25. Zhang, C., Topputo, F., Bernelli-Zazzera, F., Zhao, Y.S.: Low-thrust minimum-fuel optimization in the circular restricted three-body problem. J. Guid. Control Dyn. **38**, 1501–1510 (2015)