2016

Advanced Space Vehicle Design Taking into Account Multidisciplinary Couplings and Mixed Epistemic/Aleatory Uncertainties

Mathieu Balesdent
The French Aerospace Lab

Loic Brevault
The French Aerospace Lab

Nathaniel B. Price
University of Florida

Sebastien Defoort
The French Aerospace Lab

Rodolphe Le Riche
CNRS LIMOS and Ecole Nationale Supérieure des Mines de Saint-Etienne

See next page for additional authors

Balesdent, Mathieu; Brevault, Loic; Price, Nathaniel B.; Defoort, Sebastien; Le Riche, Rodolphe; Kim, Nam-Ho; Haftka, Raphael T.; and Berend, Nicolas, "Advanced Space Vehicle Design Taking into Account Multidisciplinary Couplings and Mixed Epistemic/Aleatory Uncertainties" (2016). *Papers in Natural Resources*. 884.
https://digitalcommons.unl.edu/natrespapers/884

Authors
Mathieu Balesdent, Loic Brevault, Nathaniel B. Price, Sebastien Defoort, Rodolphe Le Riche, Nam-Ho Kim, Raphael T. Haftka, and Nicolas Berend

Springer Optimization and Its Applications 114

Giorgio Fasano
János D. Pintér *Editors*

Space Engineering

Modeling and Optimization with Case Studies

Springer

Springer Optimization and Its Applications

VOLUME 114

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 work 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

Space Engineering

Modeling and Optimization with Case Studies

Springer

Editors

Giorgio Fasano
Exploration and Science
Thales Alenia Space
Turin, Italy

János D. Pintér
Lehigh University, Bethlehem, PA, USA

PCS Inc., Halifax, NS, Canada

ISSN 1931-6828 ISSN 1931-6836 (electronic)
Springer Optimization and Its Applications
ISBN 978-3-319-41506-2 ISBN 978-3-319-41508-6 (eBook)
DOI 10.1007/978-3-319-41508-6

Library of Congress Control Number: 2016948730

Mathematics Subject Classification (2010): 05B40, 37N05, 37N40, 49-06, 65K05, 70M20, 90Bxx, 90-08, 90C11, 90C26, 90C29, 90C30

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG Switzerland

Preface

Since the beginnings of modern space exploration activities, the simultaneous consideration of key mission objectives and options—including technical feasibility, mission safety, and cost-efficiency aspects—has been essential. Space engineering projects have required, inter alia, the analysis and optimization of trajectories, fuel consumption, cargo handling, and other aspects of mission logistics, with paramount consideration given to crew and environmental safety. The experimental and commercial revenues expected from space activities such as the ones associated with the International Space Station have given rise to further complex cost-benefit analysis and risk analysis issues. The ambitious goals of future interplanetary explorations—including manned missions—will also require advanced analysis, model development, and optimization of the systems and resources involved.

While the necessary depth and quality of the decisions required by space engineering projects has been increasing, we have also witnessed continuing innovation regarding both theoretical advances and the development of ready-to-use decision support tools for such applications. The results of scientific innovation and algorithmic developments are supported and enhanced by today's advanced computational modeling and optimization environments. Since the earliest space engineering applications, the solution of increasingly hard optimization problems has become necessary. Until recent times, the common numerical optimization approaches were essentially limited to handle certain continuous (linear or convex nonlinear), linearly structured combinatorial or mixed integer-continuous optimization problems. Recent advances in the area of optimization support also the handling of (often more realistic) non-convex problem formulations. Additionally, the consideration of integer decision variables in a more flexible nonlinear modeling framework gives rise to often even harder (again, non-convex) combinatorial and mixed integer-continuous nonlinear optimization problems. The solution of such computational challenges is becoming gradually more viable as a direct result of algorithmic advances and software development.

This edited book follows up on a well-received collection of topical studies; consult Fasano and Pintér, Eds., *Modeling and Optimization in Space Engineering*, Springer, 2013. This volume consists of 18 chapters written by domain experts who offer in-depth discussions of mathematical modeling and algorithmic aspects to solve a range of advanced space engineering applications. The topics discussed in the book are briefly summarized below:

- Advanced Space Vehicle Design Taking into Account Multidisciplinary Couplings and Mixed Epistemic/Aleatory Uncertainties, by Balesdent et al.
- Using Direct Transcription to Compute Optimal Low-Thrust Transfers Between Libration Point Orbits, by Betts
- Practical Tentative Solutions for the Indirect Optimization of Spacecraft Trajectories, by Colasurdo and Casalino
- Resource-Constrained Scheduling with Nonconstant Capacity and Non-regular Activities, by Fasano
- Packing Problems in Space Solved by CPLEX: An Experimental Analysis, by Gliozzi et al.
- Designing Complex Interplanetary Trajectories for the Global Trajectory Optimization Competitions, by Izzo et al.
- Satellite Constellation Image Acquisition Problem: A Case Study, by Malladi et al.
- Reentry Test Vehicle Configuration Selection and Analysis, by Mooij
- Rigorous Global Optimization for Collision Risk Assessment on Perturbed Orbits, by Morselli et al.
- Optimal Robust Design of Hybrid Rocket Engines, by Pastrone and Casalino
- Nonlinear Regression Analysis by Global Optimization Applied in Space Engineering, by Pintér et al.
- Regression-Based Sensitivity Analysis and Robust Design, by Ridolfi and Mooij
- Low-Thrust Multi-Revolution Orbit Transfers, by Schäff
- Balance Layout Problems: Mathematical Modeling and Nonlinear Optimization, by Stoyan et al.
- Pilot-Induced Oscillation Alleviation Through Anti-Windup-Based Approach, by Tarbouriech et al.
- Modeling and Optimization of Hybrid Transfers to NEOs, by Topputo and Massari
- Probabilistic Safety Analysis of the Collision Between Space Debris and a Satellite with an Island Particle Algorithm, by Vergé et al.
- Flatness-Based Low-Thrust Trajectory Optimization for Spacecraft Proximity Operations, by Yang et al.

Our book is primarily written for researchers and practitioners in the field of space engineering. Since it offers an in-depth exposition of the mathematical modeling, algorithmic, and numerical solution aspects of the topics covered, it will be useful also for aerospace graduate and postgraduate students who wish to broaden their horizon by studying real-world applications and challenges that they will meet in their professional work. The contributed chapters are more focused on space engineering practice than on theory: for the latter readers are referred, as needed, to further (appropriately cited) literature. With this aspect in mind, researchers and practitioners in mathematical systems modeling, operations research, mathematical optimization, and optimal control will also benefit from the case studies presented.

The approaches discussed here can be extended also to other application areas that are not related to space applications. Hence, the book can be also used as a reference volume to assist researchers and practitioners in developing novel 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
April 2016

Acknowledgments

First and foremost, we wish to thank all contributing authors for offering their high-quality research work and for their efforts to make the timely completion and publication of this volume possible. We also express our thanks to coauthors of joint works cited in our own contributed chapters.

In addition to the review and editorial activities done by ourselves, several colleagues assisted us with valuable comments and suggestions. We wish to express our special thanks to Franco Bernelli-Zazzera, John Betts, Lorenzo Casalino, Pierluigi Di Lizia, Andreas Fischer, Dario Izzo, Mauro Massari, Snezana Mitrovic Minic, Dario Pastrone, and Sven Schäff.

One of the editors (GF) thanks also Piero Messidoro and Annamaria Piras of Thales Alenia Space Italia S.p.A., for their support of the research and development activities related to modeling and optimization for space engineering applications.

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

MSC 2010 Classification: Suggested Categories and Keywords for the Book

05B40	Packing and covering
37N05	Dynamical systems in classical and celestial mechanics
37N40	Dynamical systems in optimization and economics
49-06	Calculus of variations and optimal control; optimization: proceedings, conferences, collections
65K05	Mathematical programming methods
70M20	Orbital mechanics
90Bxx	Operations research and management science
90-08	Computational methods
90C11	Mixed integer programming
90C26	Non-convex programming, global optimization
90C29	Multi-objective and goal programming
90C30	Nonlinear programming

Contents

Editors

Giorgio Fasano is a researcher and practitioner at Thales Alenia Space, with three decades of experience in the field of optimization and space engineering applications. He holds an MSc degree in mathematics, and he is a fellow of the Institute of Mathematics and its Applications (IMA, UK), with the designations of chartered mathematician (IMA, UK) and chartered scientist (Science Council, UK). His interests include mathematical modeling, 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 a number of other publications related to optimization in space.

János D. Pintér is a researcher and practitioner with over four decades of experience in the area of systems modeling, analytics, and optimization, with an emphasis on algorithm and software development for nonlinear optimization. He holds MSc, PhD, and DSc degrees in mathematics. He has authored and edited 10 books (so far) including the award-winning monograph *Global Optimization in Action*. He also wrote nearly 200 peer-reviewed journal articles, book chapters, and other research documents. Dr. Pintér is (or has been) a member and officer of professional organizations including CORS, EUROPT, HORS, and INFORMS; and he serves/served on the editorial board of international professional journals including the *Journal of Global Optimization*. Dr. Pintér is the principal developer or codeveloper of a range of nonlinear optimization software products. These products are used worldwide by researchers and practitioners in academia and at business and research organizations. He also offers optimization courses (as well as a range of other university or professional training courses) on demand. For more information, please visit www.pinterconsulting.com.

In addition to the current volume, Fasano and Pintér are editors of the following volumes:
Modeling and Optimization in Space Engineering, Springer, 2013
Optimized Packings with Applications, Springer, 2015

Advanced Space Vehicle Design Taking into Account Multidisciplinary Couplings and Mixed Epistemic/Aleatory Uncertainties

Mathieu Balesdent, Loïc Brevault, Nathaniel B. Price, Sébastien Defoort, Rodolphe Le Riche, Nam-Ho Kim, Raphael T. Haftka, and Nicolas Bérend

Abstract Space vehicle design is a complex process involving numerous disciplines such as aerodynamics, structure, propulsion and trajectory. These disciplines are tightly coupled and may involve antagonistic objectives that require the use of specific methodologies in order to assess trade-offs between the disciplines and to obtain the global optimal configuration. Generally, there are two ways to handle the system design. On the one hand, the design may be considered from a disciplinary point of view (a.k.a. Disciplinary Design Optimization): the designer of each discipline has to design its subsystem (e.g. engine) taking the interactions between its discipline and the others (interdisciplinary couplings) into account. On the other hand, the design may also be considered as a whole: the design team addresses the global architecture of the space vehicle, taking all the disciplinary design variables and constraints into account at the same time. This methodology is known as Multidisciplinary Design Optimization (MDO) and requires specific mathematical tools to handle the interdisciplinary coupling consistency.

In the first part of this chapter, we present the main classical techniques to efficiently tackle the interdisciplinary coupling satisfaction problem. In particular, an MDO decomposition strategy based on the "Stage-Wise decomposition for Optimal Rocket Design" formulation is described. This method allows the design process to be decentralized according to the different subsystems (e.g. launch vehicle stages) and reduces the computational cost compared to classical MDO methods.

M. Balesdent (✉) • L. Brevault • S. Defoort • N. Bérend
Onera - The French Aerospace Lab, F-91761 Palaiseau, France
e-mail: mathieu.balesdent@onera.fr

N.B. Price
Onera - The French Aerospace Lab, F-91761 Palaiseau, France

CNRS LIMOS and Ecole Nationale Supérieure des Mines de Saint-Etienne, Saint-Etienne, France

University of Florida, Gainesville, FL, USA

R. Le Riche
CNRS LIMOS and Ecole Nationale Supérieure des Mines de Saint-Etienne, Saint-Etienne, France

N.-H. Kim • R.T. Haftka
University of Florida, Gainesville, FL, USA

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization
and Its Applications 114, DOI 10.1007/978-3-319-41508-6_1

Furthermore, when designing an innovative space vehicle including breakthrough technologies (e.g. launch vehicle with new kind of propulsion, new aerodynamics configuration), one has to cope with numerous uncertainties relative to the involved technology models (epistemic uncertainties) and the effects of these on the global design and on the interdisciplinary coupling satisfaction. Moreover, aleatory uncertainties inherent to the physical phenomena occurring during the space vehicle mission (e.g. solar fluxes, wind gusts) must also be considered in order to accurately estimate the performance and reliability of the vehicle. The combination of both epistemic and aleatory uncertainties requires dedicated techniques to manage the computational cost induced by uncertainty handling.

The second part of this chapter is devoted to the handling of design process in the presence of uncertainties. Firstly, we describe a design methodology that enables to define the design rules (e.g. safety factors) taking both aleatory and epistemic uncertainties into account. Secondly, we present new MDO methods that allow to decompose the design process while maintaining the interdisciplinary functional coupling relationships between the disciplines in the presence of uncertainties.

Keywords Multidisciplinary Design Optimization • Launch vehicle design • Aleatory/epistemic uncertainties

Nomenclature

$\mathbf{z}$	Design variable vector
$\mathbf{Y}$	Input coupling variable vector
$\mathbf{U}$	Uncertain variable vector
f	Objective function
$\mathbf{g}$	Inequality constraint vector
$\mathbf{h}$	Equality constraint vector
$\mathbf{c}$	Coupling function vector
Ξ	Objective function uncertainty measure
$\mathbf{K}$	Inequality function uncertainty measure
$\mathbb{E}$	Expected value
σ	Standard deviation
ϕ	Joint Probability Density Function (PDF)
$\boldsymbol{\theta}_Y$	Parameter vector of the uncertain coupling variables $\mathbf{Y}$
$\hat{y}$	Polynomial Chaos Expansion based surrogate model of the coupling y
$\boldsymbol{\alpha}$	Polynomial Chaos Expansion coefficient vector
J	Interdisciplinary coupling constraint
s	Safety margin
$\hat{m}$	Low fidelity model

1 Introduction

Aerospace vehicle designs are long term projects (often around 10 years) involving important budgets and requiring a dedicated design organization. NASA and ESA [63] stress the need to reduce the cost and to increase the effectiveness of space missions and satellite launches. Improving the design process for aerospace vehicles is essential to obtain low cost, high reliability, and effective launch capabilities [12]. This design is a complex multidisciplinary optimization process: the objective is to find the vehicle architecture and characteristics that provide the optimal performance [34] while satisfying design requirements and ensuring a certain level of reliability. The slightest mistake in the design process may induce economical, material and human disastrous consequences (e.g. explosion of the Brazilian VLS launch vehicle in 2002).

As a representative example of aerospace vehicles, the design of launch vehicles involves several disciplines (e.g. propulsion, aerodynamics, trajectory, mass and structure) and is customarily decomposed into interacting submodels (Fig. 1). Each discipline may rely on computing-intensive simulations such as Finite Element Analyses for the structure discipline or Computational Fluid Dynamics analyses for the aerodynamics discipline. The aerospace vehicle performance estimation which results from flight performance, safety, reliability and cost, requires coupled

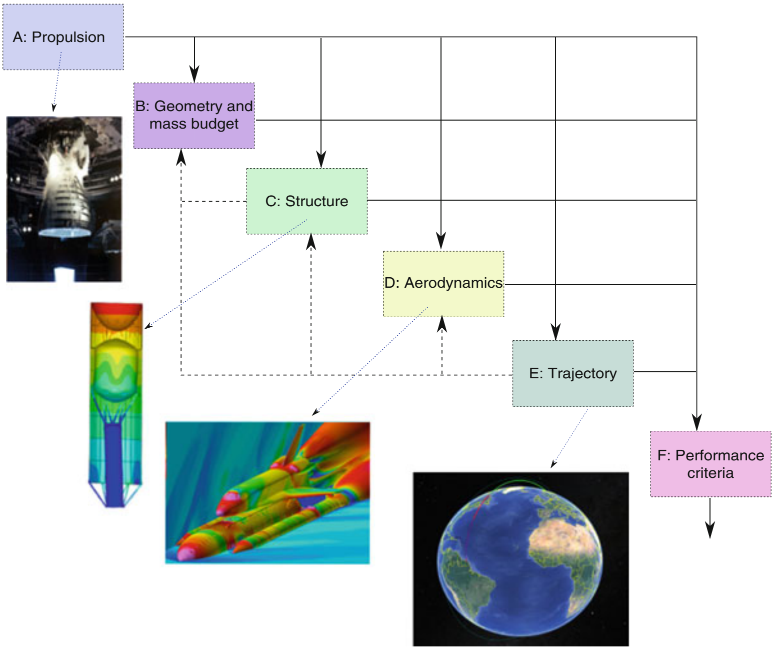


Fig. 1 Example of launch vehicle analysis process of interacting submodels

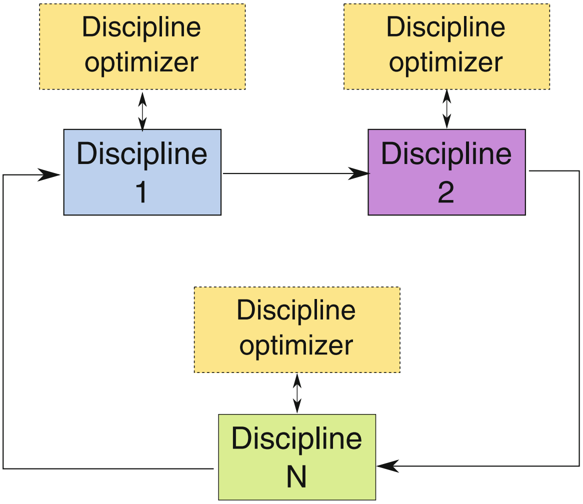


Fig. 2 DDO design process

disciplinary analyses. The different disciplines are a primary source of trade-offs due to the antagonist disciplinary effects on launcher performance.

Two approaches to handle system design may be distinguished:

- **Disciplinary Design Optimization** (DDO). The designer of each discipline has to design its subsystem (e.g. propulsion system) taking the interactions between its discipline and the others (interdisciplinary couplings) into account through specifications that can take the form of simulation parameters or optimization constraints that will be updated at each iteration. The process generally consists of loops between different disciplinary optimizations (Fig. 2). At each iteration of this loop, each discipline is re-processed based on the updated data from the previous discipline optimizations. This approach is particularly suited to the design process of industrial companies which is often broken down according to the different engineering team expertise. However, the difficulty of this approach lies in the handling of other discipline interactions with the designed discipline in the global optimization process.
- **Multidisciplinary Design Optimization** (MDO). MDO deals with the global design problem as a whole by taking advantage of the inherent synergies and couplings between the disciplines involved in the design process (Fig. 3) to decrease the computational cost and/or to improve the quality of the optimal design [53]. Unlike the sequential disciplinary optimizations, the interactions between the disciplines are directly incorporated in the MDO methods [7]. However, the complexity of the problem is significantly increased by the simultaneous handling of all the disciplines.

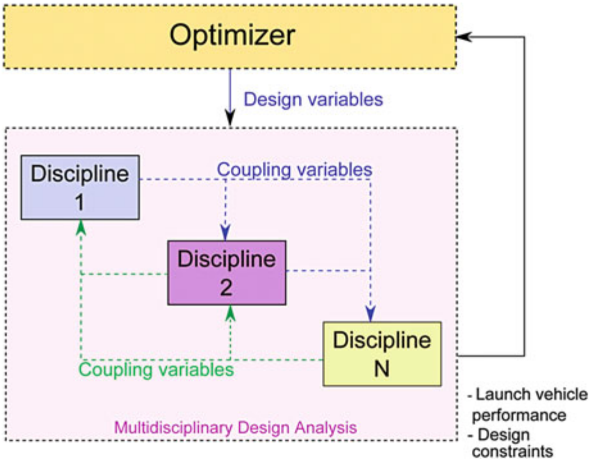


Fig. 3 Example of MDO design process

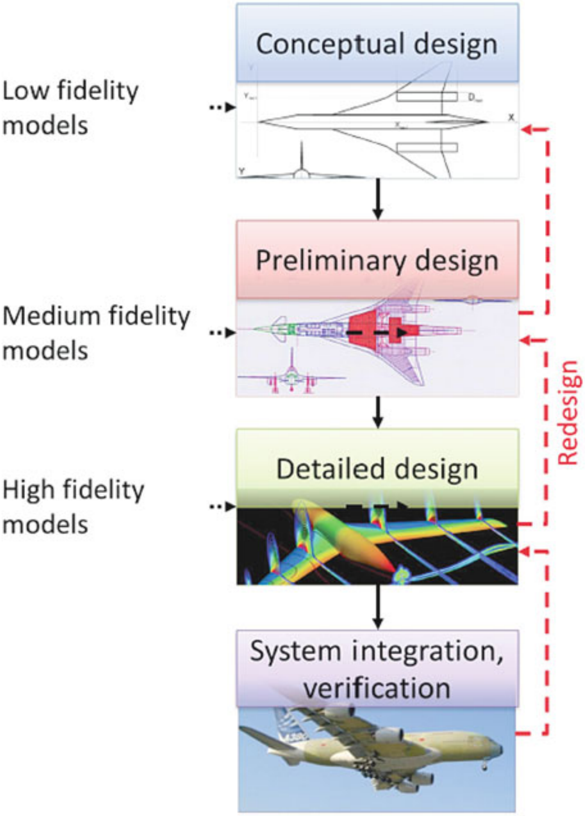
In the next sections, these two approaches are discussed within the context of Launch Vehicle Design (LVD). In Sect. 2, the main classical techniques to efficiently tackle the interdisciplinary coupling satisfaction in MDO problems are introduced. In particular, an MDO decomposition strategy based on the "Stage-Wise decomposition for Optimal Rocket Design" (SWORD) is described. This method allows the design process to be decentralized according to launch vehicle stages and reduces the computational cost compared to classical MDO methods.

Then, in Sect. 3, the handling of uncertainty in the design process is discussed. First, a methodology to define design rules (e.g. safety margins) taking epistemic and aleatory uncertainties into account is described. Then, an approach allowing to ensure multidisciplinary feasibility in the presence of uncertainty for space vehicle design while reducing the computational cost is presented and compared to classical uncertainty-based MDO methods.

2 MDO Decomposition Strategy for Launch Vehicle Design

In the aerospace industry, a new system follows a typical development process involving several specific phases (Conceptual design, Preliminary design, Detailed design, Manufacturing) [12] (Fig. 4). For an aerospace vehicle, the conceptual design phase is decisive for the success of the whole design process. It has been estimated that at least 80 % of the life-cycle cost of a vehicle is locked in by the chosen concept during the conceptual phase [12]. The design space at this early design phase is large since few characteristics of the system are fixed, and traditional design approaches lead to freeze some system characteristics to focus only on

Fig. 4 Phases of design process



alternatives selected by experts [62]. MDO techniques are useful for the conceptual design phase since they are able to handle large design spaces in a multidisciplinary environment. In [41], the authors mention that the global system performance can be enhanced by using MDO at early design phases, and design cycle duration and cost can be decreased.

To overcome the complexity induced by handling all the disciplines at the same time in the system design process, various MDO formulations have been developed. In the 90s, several surveys classed MDO formulations into two general types of architectures: single-level methods [10, 21], and multi-level methods [3, 38]. Multi-level approaches introduce disciplinary level optimizers in addition to the system-level optimizer involved in single-level methods, in order to facilitate the MDO process convergence.

2.1 General MDO Formulation and Review of Main MDO Approaches

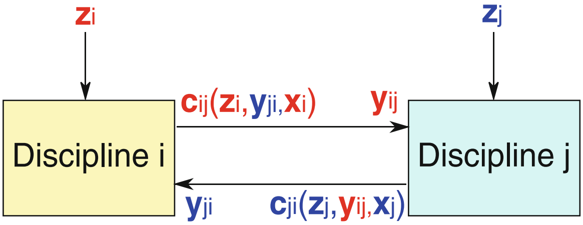
A general single-level MDO problem can be formulated as follows [8]:

$$\begin{aligned}
\min \quad & f(\mathbf{z}, \mathbf{y}, \mathbf{x}) && (1)\\
\text{w.r.t.} \quad & \mathbf{z}, \mathbf{y}, \mathbf{x} && \\
\text{s.t.} \quad & \mathbf{g}(\mathbf{z}, \mathbf{y}, \mathbf{x}) \leq 0 && (2)\\
& \mathbf{h}(\mathbf{z}, \mathbf{y}, \mathbf{x}) = 0 && (3)\\
& \forall (i,j) \in \{1, \ldots, N\}^2\ i \neq j,\ \mathbf{y}_{ij} = \mathbf{c}_{ij}(\mathbf{z}_i, \mathbf{y}_{.i}, \mathbf{x}_i) && (4)\\
& \forall i \in \{1, \ldots, N\},\ \mathbf{r}_i(\mathbf{z}_i, \mathbf{y}_{.i}, \mathbf{x}_i) = 0 && (5)\\
& \mathbf{z_{min}} \leq \mathbf{z} \leq \mathbf{z_{max}} && (6)
\end{aligned}$$

All the variables and functions are described in the following. Three types of variables are involved in a deterministic MDO problem:

- $\mathbf{z}$ is the design variable vector. The design variables evolve all along the optimization process in order to find their optimal values with respect to the optimization problem. Design variables may be shared between several disciplines ($\mathbf{z}_{sh}$) or specific to the discipline i ($\bar{\mathbf{z}}_i$). We note $\mathbf{z}_i = \{\mathbf{z}_{sh}, \bar{\mathbf{z}}_i\}$ the input design variable vector of the discipline $i \in \{1, \ldots, N\}$ with N the number of disciplines and $\mathbf{z} = \bigcup_{i=1}^{N} \mathbf{z}_i$ without duplication. Typical design variables involved in aerospace vehicle design are stage diameters, pressures in the combustion chambers, propellant masses, etc.
- In a multidisciplinary environment, the disciplines exchange coupling variables, $\mathbf{y}$ (Fig. 5). The latter link the different disciplines to model the interactions between them. $\mathbf{c}_{ij}(\mathbf{z}_i, \mathbf{y}_{.i}, \mathbf{x}_i)$ is a coupling function used to compute the *output coupling* variable vector which is calculated by discipline i and input to discipline j. $\mathbf{y}_{.i}$ refers to the vector of all the *input coupling* variables of discipline i and $\mathbf{y}_{ij}$ is the *input coupling* variable vector which is input to discipline j and output from discipline i. We note $\mathbf{y} = \bigcup_{i=1}^{N} \mathbf{y}_{.i} = \bigcup_{i=1}^{N} \mathbf{y}_{i.}$ without duplication. From the design variables and the input coupling variables to the discipline i, the output

Fig. 5 Couplings between the discipline i and the discipline j



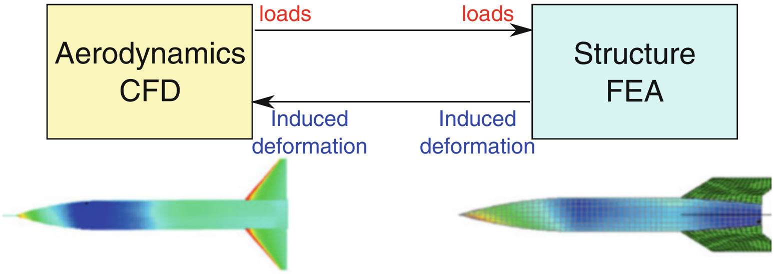


Fig. 6 Couplings between aerodynamics and structure disciplines

coupling variables are computed with the coupling function: $\mathbf{c}_{i.}(\mathbf{z}_i, \mathbf{y}_{.i}, \mathbf{x}_i)$ and $\mathbf{y}_{i.} = (\mathbf{y}_{i1}, \ldots, \mathbf{y}_{iN})$ is the vector of the outputs of discipline i and the input coupling variable vector to all the other disciplines.

For example, the sizing discipline computes the launch vehicle dry mass which is transferred to the trajectory discipline for a simulation of the launch vehicle flight. Another example is the classical aero-structure analysis (Fig. 6) [20, 24, 35]. For a launch vehicle, aero-structure analysis involves coupled analyses between the aerodynamics discipline (which requires the launch vehicle geometry and the deformations) and the structure discipline (which requires the aerodynamics loads on the launch vehicle structure). For coupled systems, it is important to keep in mind that their design involves goals which are often conflicting with each other, for instance reducing weight may lead to higher stresses and the global optimum is a compromise between all the different disciplinary objectives.

- $\mathbf{x}$ is the state variable vector. Unlike $\mathbf{z}$, the state variables are not independent degrees of freedom but depend on the design variables, the coupling variables $\mathbf{y}$ and the state equations characterized by the residuals $\mathbf{r}(\cdot)$. These variables are often defined by implicit relations that require specific numerical methods for solving complex industrial problems. For instance, the guidance law (e.g. modeled by pitch angle interpolation with respect to a set of crossing points) in the launch vehicle trajectory discipline has to be determined in order to ensure payload injection into orbit. The guidance law is often the result of an optimization problem minimizing the discrepancy between the target orbit injection and the real orbit injection. In such a modeling, the pitch angle crossing points are state variables $\mathbf{x}$ and the orbit discrepancy is the residual $\mathbf{r}(\cdot)$ to be canceled. Sometimes, the coupling variables $\mathbf{y}$ can be a subset of state variables $\mathbf{x}$.

In order to solve the MDO problem Eqs. (1)–(6), we are looking for:

- **Inequality and equality constraint feasibility**: the MDO solution has to satisfy the inequality constraints imposed by $\mathbf{g}(\cdot)$ and the equality constraints imposed by $\mathbf{h}(\cdot)$. These constraints are used to represent the requirements for the system in terms of targeted performance, safety, flexibility, etc. For example, a target orbit altitude for a launch vehicle payload is an equality constraint to be satisfied.

- **Individual disciplinary feasibility**: the MDO solution has to ensure the disciplinary state equation satisfaction expressed by the residuals $\mathbf{r}_i(\cdot)$. The latter $\mathbf{r}_i(\cdot)$ quantify the satisfaction of the state equations in discipline i. The state variables $\mathbf{x}_i$ are the roots of the state equations of discipline i. For instance, state equations may be used to represent thermodynamics equilibrium between the chemical components in rocket engine combustion. In the rest of the chapter, it is assumed that the satisfaction of the disciplinary feasibility is directly ensured by the disciplines (disciplinary analysis [8]), therefore, no more references to state variables and residuals will be done, without loss of generality.
- **Multidisciplinary feasibility**: the MDO solution has to satisfy the interdisciplinary equality constraints between the input coupling variable vector $\mathbf{y}$ and the output coupling variable vector $\mathbf{c}(\cdot)$ resulting from the discipline simulations. The couplings between the disciplines i and j are said to be *satisfied* (also called *feasible* or *consistent*) when the following interdisciplinary system of equations is verified:

$$\begin{cases} \mathbf{y}_{ij} = \mathbf{c}_{ij}(\mathbf{z}_i, \mathbf{y}_{.i}) \\ \mathbf{y}_{ji} = \mathbf{c}_{ji}(\mathbf{z}_j, \mathbf{y}_{.j}) \end{cases} \tag{7}$$

When all the couplings are satisfied, i.e. when Eqs. (7) are satisfied $\forall(i,j) \in \{1,\ldots,N\}^2 \quad i \neq j$, the system is said to be *multidisciplinary feasible*. The satisfaction of the interdisciplinary couplings is essential as it is a necessary condition for the modeled system to be physically realistic. Indeed, in the aero-structure example, if the aerodynamics discipline computes a load of 10 kPa, it is necessary that the structure discipline uses as input 10 kPa and not another value otherwise the aero-structure analysis is not consistent.

- **Optimal MDO solution**: $f(\cdot)$ is the objective function (also called performance) to be optimized. The objective function characterizes the system performance and is a measure of its quality expressed with some metrics [e.g. launch vehicle life cycle cost, Gross Lift-Off Weight (GLOW)]. Several performance measures may be considered together by using multi-objective optimization. Multi-objective optimization is not considered in this chapter, so that the interest reader may consult [40].

In MDO, coupled and decoupled approaches may be distinguished to satisfy the interdisciplinary couplings [10].

- *Coupled approaches* (Fig. 7) use a specific process, called MultiDisciplinary Analysis (MDA), in order to satisfy the interdisciplinary couplings at each iteration of the system-level optimization. MDA is an auxiliary process used to find a numerical equilibrium between the disciplines by solving the system of interdisciplinary equations [20]. MDA enables to find the numerical value of the input coupling variables $\mathbf{y}$ in order to solve the system of equations [Eqs. (7)]. MDA can be performed by using classical techniques such as Fixed

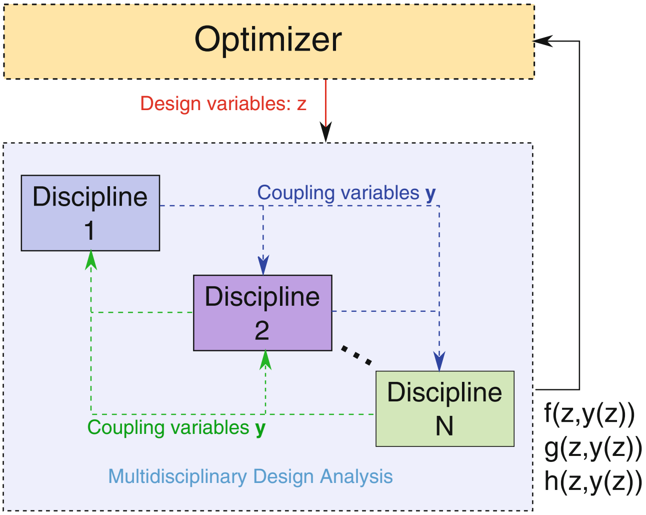
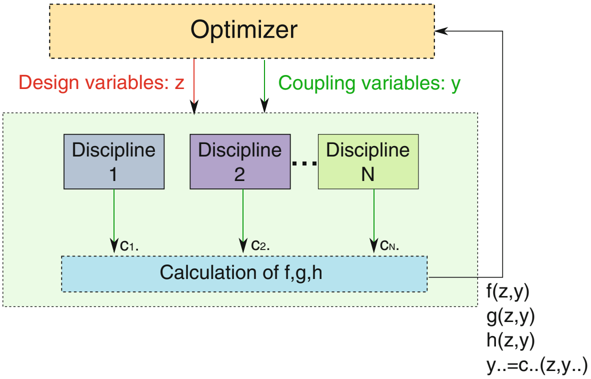


Fig. 7 Multidisciplinary Design Optimization, **coupled** approach

Fig. 8 Multidisciplinary Design Optimization, **decoupled** approach



Point Iteration [8], or by an auxiliary optimization problem allowing to reduce the discrepancy between the input coupling vector and the output coupling vector.

- *Decoupled approaches* (Fig. 8) aim at removing MDA and involve equality constraints on the coupling variables in the MDO formulation at the system-level Eq. (4) to ensure the interdisciplinary coupling satisfaction only for the optimal design, and not at each MDO process iteration in **z** such as coupled approaches do. These additional equality constraints are imposed between the input and the output coupling variables in the MDO formulation at the same level as the system constraints $\mathbf{g}(\cdot)$ and $\mathbf{h}(\cdot)$: $\forall(i,j) \in \{1,\ldots,N\}^2\ i \neq j,\ \mathbf{y}_{ij} = \mathbf{c}_{ij}(\mathbf{z}_i, \mathbf{y}_{.i})$. The basic idea is to define the coupling variables **y** as optimization variables. Consequently, the system-level optimizer has to control both the design variables and the input coupling variables. Hence, the additional degrees of freedom introduced by

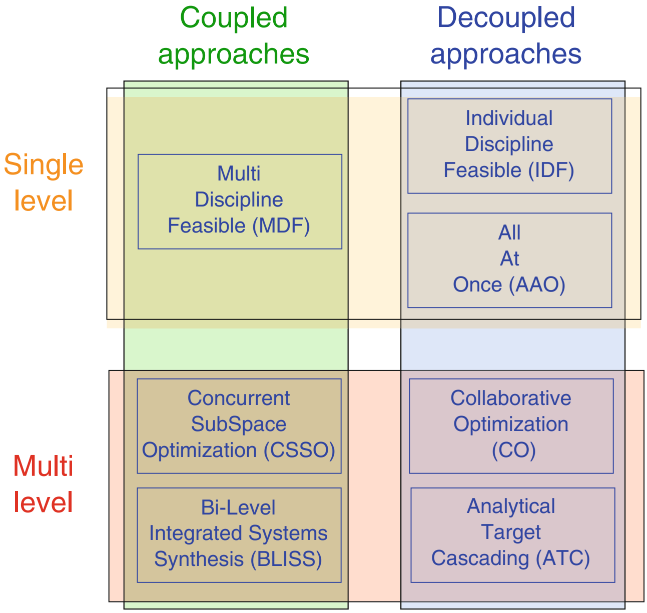

Coupled approaches
Decoupled approaches
Single level
Multi level
Multi Discipline Feasible (MDF)
Individual Discipline Feasible (IDF)
All At Once (AAO)
Concurrent SubSpace Optimization (CSSO)
Bi-Level Integrated Systems Synthesis (BLISS)
Collaborative Optimization (CO)
Analytical Target Cascading (ATC)

Fig. 9 Classification of the main MDO formulations

expanding the optimization variable set handled by the system-level optimizer are controlled by the coupling equality constraints. The equality constraints on coupling variables may not be satisfied at each iteration but allow to guide the search of optimal design.

Several MDO formulations have been proposed in literature to efficiently solve general and specific engineering problems. Some articles [3, 8, 10, 15, 41] provide a review of the different methods and compare them qualitatively and numerically on MDO problem benchmarks [58, 61]. Classical MDO formulations may be classified in four categories (Fig. 9) according to the *coupled* or *decoupled* and to the *single-level* or *multi-level* approaches. The single-level *vs.* multi-level formulations are differentiated by the number of optimizers. Single-level formulations have only one system optimizer to solve the MDO problem whereas in multi-level formulations, in addition to the system optimizer, discipline optimizers are introduced in order to distribute the problem complexity over different dedicated discipline optimizations. The four categories are:

- Single-level approaches with MDA: e.g. *Multi Discipline Feasible* (MDF) [10],
- Multi-level approaches with MDA: e.g. *Concurrent SubSpace Optimization* (CSSO) [52], *Bi-Level Integrated System Synthesis* (BLISS) [54],
- Single-level approaches with equality constraints on the coupling variables: e.g. *Individual Discipline Feasible* (IDF) [10], *All At Once* (AAO) [10],

- Multi-level approaches with equality constraints on the coupling variables: e.g. *Collaborative Optimization* (CO) [13], *Analytical Target Cascading* (ATC) [4], *QuasiSeparable Decomposition* (QSD) [29].

MDF is the most used method in literature [8]. MDF is a single-level optimization formulation involving one system-level optimizer and a MDA to solve the interdisciplinary coupling equations. CSSO and BLISS use MDA to ensure coupling satisfaction but enable parallel discipline optimizations. AAO, ATC, CO, IDF and QSD are fully decoupled formulations with satisfaction of the couplings by incorporating additional variables and corresponding equality constraints. The decoupled MDO formulations offer several advantages compared to MDF [8, 41]:

- Parallel analyses of the disciplines,
- Reduction of the number of calls to the computationally expensive discipline codes (because MDA is removed),
- Improvement of the system optimization process convergence, however, most of the time there is no proof that the convergence is to the same optimum,
- Distribution of the optimization problem complexity: discipline optimizers only control local design variables and system-level optimizer only handles the shared design variables between several disciplines and the coupling variables.

However, in order to be competitive with respect to MDF, decoupled MDO formulations require an appropriate interdisciplinary coupling handling. Moreover, these formulations involve more variables and more constraints. In [8], the authors performed a detailed review of classical MDO formulations applied to LVD. This study points out that LVD present particularities, notably the importance of the trajectory discipline compared to the other disciplines. Exploiting these specificities in an MDO formulation might improve the LVD process. Dedicated formulations for LVD have been proposed such as the Stage-Wise decomposition for Optimal Rocket Design (SWORD) [7]. The next section focuses on these dedicated formulations.

2.2 Stage-Wise decomposition for Optimal Rocket Design

2.2.1 Theoretical Formulations

In literature, the classical way to design a launch vehicle is to decompose the design process according to the involved disciplines (propulsion, aerodynamics, sizing, trajectory, etc.). The decomposition according to the disciplines has been coupled with single-level methods (Individual Discipline Feasible, All At Once [18]) or multi-level methods (Collaborative Optimization [14], Concurrent SubSpace Optimization [52], Bi-Level Integration Systems Synthesis [55], etc.). In these methods, the trajectory is also optimized as a whole and is often considered as a "black box" for the optimization. The SWORD formulations [7] allow to decompose the LVD according to the different stages in order to improve the efficiency of the MDO process. In these formulations, the subsystems are not the disciplines but

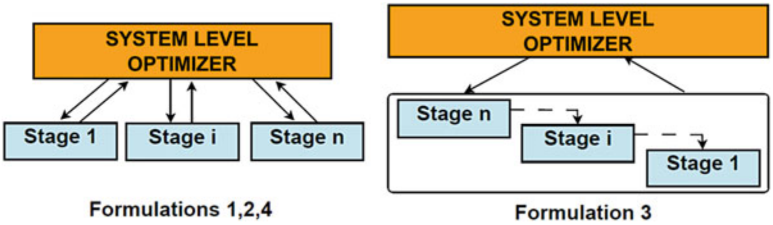


Fig. 10 SWORD formulations (formulations 1,2,4 are parallel and formulation 3 is hierarchical)

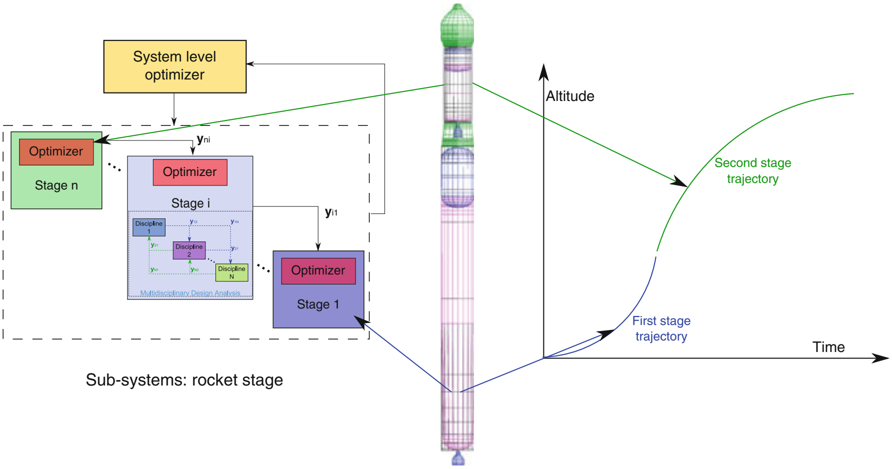


Fig. 11 Third SWORD formulation

the different stage optimizations incorporating all the required disciplines involved in the stage design. SWORD are multi-level decoupled MDO formulations [9]. Four different formulations have been proposed depending on the decomposition process and the interdisciplinary coupling constraint handling (Fig. 10). This type of decomposition is proposed in the context of LVD but is generalizable to systems for which the system-level objective function can be decomposed into a sum of subsystem contributions, as involved in the QSD formulation [29]. According to the comparison of the methods on launch vehicle application cases implemented in [7], the third formulation is the most efficient to solve MDO problems (with respect to the number of discipline evaluations) due to its hierarchical decomposition of the design process and only this formulation is detailed in the following for the sake of conciseness (Fig. 11). For more details about the other SWORD formulations, one may consult [9]. In SWORD, the objective function $f(\cdot)$ is assumed to be decomposed such as $f(\cdot) = \sum_{j=1}^{n} f_j(\cdot)$ with n the number of stages. In practice, the Gross Lift-Off Weight (GLOW) is often minimized in LVD process [8, 19] and it can be decomposed as the sum of the stage masses and upper composite. The MDO formulation of the LVD problem using SWORD is given by:

At the system-level:

$$\begin{aligned}
\min \quad & f\left(\mathbf{z}, \mathbf{y}\right) && (8)\\
\text{w.r.t.} \quad & \mathbf{z}_{sh}, \mathbf{y} \\
\text{s.t.} \quad & \mathbf{g}_0(\mathbf{z}, \mathbf{y}) \leq 0 && (9)\\
& \forall i \in \{1, \ldots, n\}, \mathbf{g}_i \left(\mathbf{z}_{sh}, \bar{\mathbf{z}}_i^*, \mathbf{y}\right) \leq 0 && (10)\\
& \forall i \in \{1, \ldots, n\}, \mathbf{h}_i \left(\mathbf{z}_{sh}, \bar{\mathbf{z}}_i^*, \mathbf{y}\right) = 0 && (11)\\
& \forall i, j \in \{1, \ldots, n\}^2 i \neq j, \mathbf{y}_{ij} = \mathbf{c}_{ij}(\mathbf{z}_{sh}, \bar{\mathbf{z}}_i^*, \mathbf{y}_{.i}) && (12)\\
& \mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max} && (13)
\end{aligned}$$

At the subsystem-level:
$i = n$
While $i > 0$
For the *i*th **stage:**
Given $\mathbf{z}_{sh}, \mathbf{y}_{i+1}, \ldots, \mathbf{y}_n$ (for launch vehicle: the optimal masses of the stages $i+1$ to n):

$$\begin{aligned}
\min \quad & f_i(\mathbf{z}_{sh}, \bar{\mathbf{z}}_i, \mathbf{y}) && (14)\\
\text{w.r.t.} \quad & \bar{\mathbf{z}}_i \\
\text{s.t.} \quad & \mathbf{g}_i \left(\mathbf{z}_{sh}, \bar{\mathbf{z}}_i, \mathbf{y}\right) \leq 0 && (15)\\
& \mathbf{h}_i \left(\mathbf{z}_{sh}, \bar{\mathbf{z}}_i, \mathbf{y}\right) = 0 && (16)\\
& \mathbf{y}_{i.} = \mathbf{c}_{i.}(\mathbf{z}_{sh}, \bar{\mathbf{z}}_i, \mathbf{y}_{.i}) && (17)\\
& \bar{\mathbf{z}}_{i_{\min}} \leq \bar{\mathbf{z}}_i \leq \bar{\mathbf{z}}_{i_{\max}} && (18)
\end{aligned}$$

$i \leftarrow i - 1$
where $\bar{\mathbf{z}}_i^*$ is the optimal variable vector found by the *i*th subsystem optimizer. This formulation allows to separately optimize each stage in a hierarchical process. The last stage is optimized first and the first stage is optimized last. The result of the previous optimization is passed to the next launch vehicle stage optimization (Fig. 10). In order to decouple the different stage optimizations, the added coupling variables $\mathbf{y}$ are the state vectors (position and velocity) at stage separations (to ensure the consistency of the trajectory) and the estimation of the stage masses. Furthermore, in order to ensure the trajectory consistency, additional constraints concerning the reach of each stage separation point (and final orbit for the upper stage) are involved at the subsystem-level. The different stage optimizations cannot be performed in parallel which may be a drawback in terms of computational cost when parallelization is possible. For more details on the SWORD formulations, see [7]. In the following section, this formulation is applied to the design of a three-stage-to-orbit launch vehicle [7] and is compared to MDF.

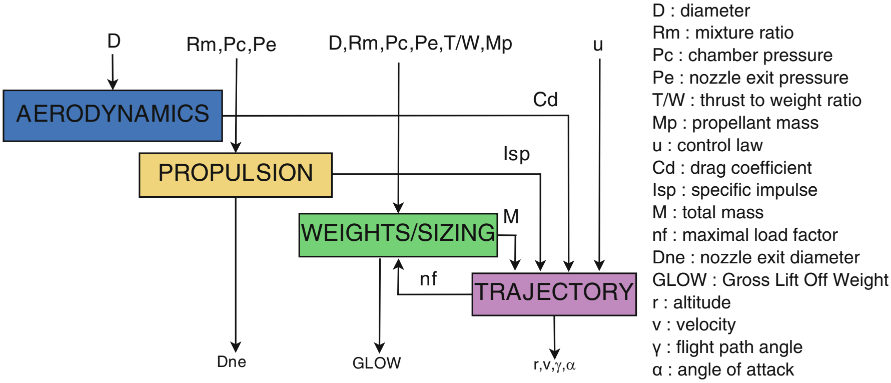


Fig. 12 N2 chart for one stage

Table 1 Design variables for the three stage LVD problem

Design variables	Symbols
Stage diameters	D_1, D_2, D_3
Stage propellant masses	M_{p1}, M_{p2}, M_{p3}
Stage mixture ratio	Rm_1, Rm_2, Rm_3
Stage chamber pressure	Pc_1, Pc_2, Pc_3
Stage nozzle exit pressure	Pe_1, Pe_2, Pe_3
Stage thrust to weight ratio	TW_1, TW_2, TW_3
Stage control law parameter vector	$\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$

2.2.2 Application of SWORD to Launch Vehicle Design

Description of the Test Case

The proposed design problem consists in optimizing a three-stage-to-orbit expendable launch vehicle. The selected criterion is the GLOW minimization. The payload mass is fixed at 4 metric tons. The target orbit is a 250 × 35,786 km Geostationary Transfer Orbit (GTO). The considered disciplines are propulsion, aerodynamics, mass budget, and trajectory (Fig. 12), using low fidelity models [7, 19, 33, 57]. The considered design variables are summarized in Table 1. The constraints taken into account are relative to the reach of the target orbit, the maximal angle of attack during the trajectory, the geometry of the nozzle and the nozzle exit pressure. For more details about the problem description, one can consult [6].

The considered design variables are the chamber pressures Pc, the nozzle exit pressures Pe, the thrust to weight ratios TW, the propellant masses M_p, the mixture ratios Rm, the stage diameters D and the control law $\mathbf{u}$. All the stages are cryogenic propulsion stages (LOX/LH2). The propulsion module consists in computing the specific impulse (Isp) from Pc, Pe, Rm and TW. The aerodynamics module computes the drag coefficient Cd from the geometry characteristics of the launch vehicle. We use a zero-lift hypothesis in this test-case. The weight and sizing module is responsible for determining the dry mass of the different stages by computing the masses of the launch vehicle elements (tanks, combustion chamber, nozzle, pumps, pressurization system, etc.). Finally, the trajectory module consists in defining some crossing points of the pitch angle (**u**) and optimizing them in order to reach the orbit requirements and improving the objective function.

Results

At the system-level (including the MDF optimizer), a genetic algorithm with 100 individuals per generation is used (a penalization technique is used to take the constraints into account). At the subsystem-level (only for SWORD), a SQP algorithm is used. The optimization problem at the system-level is stopped after 10 h of run. Due to the stochastic nature of GA, this study has been performed with ten random initializations and very large variation domains concerning the design variables (global search). Statistics of the obtained results are detailed in the following.

Figure 13 shows the evolution of the objective function (GLOW) with respect to the computation time for only the feasible designs, for one representative initialization. At the stopping time of the optimization process, SWORD allows to obtain a better design than MDF, although it finds a worse first feasible design. Moreover, MDF presents some difficulties in improving the objective function (Fig. 14) while SWORD allows a decrease of the launch vehicle mass of 10 % in mean. The relatively bad results obtained for the MDF can be explained by the important number of optimization variables at the system-level that makes global search very difficult and lead to relatively inconsistent results.

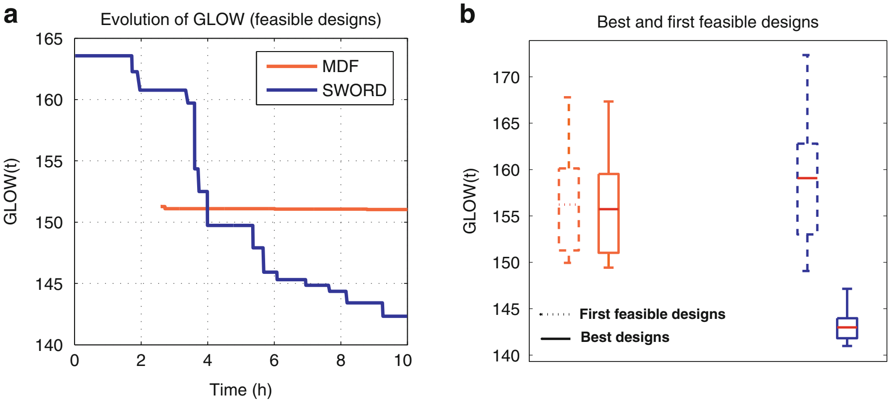


Fig. 13 Comparison of SWORD and MDF. (**a**) Evolution of GLOW for one representative initialization. (**b**) Boxplot of best and first feasible designs

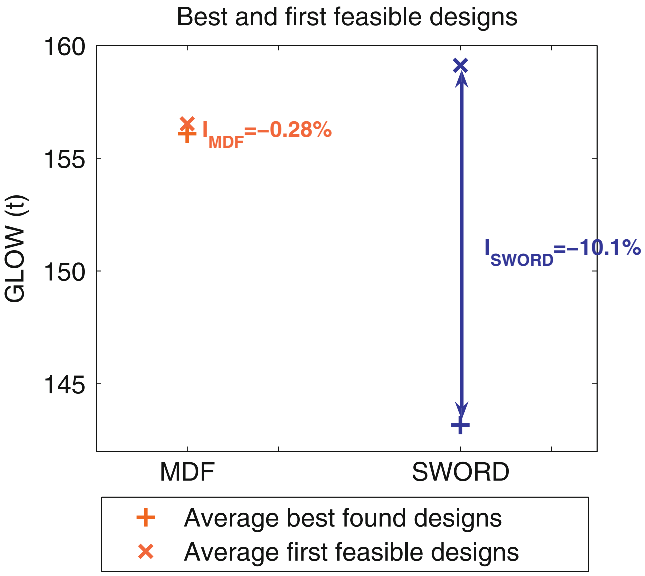


Fig. 14 Improvement (I) of objective function during the optimization

The best advantage of using a dedicated MDO formulation such as SWORD is to benefit from the specificities of the design problem to solve, that allows in this case, to reduce the search domain dimension at the system-level and to move the design complexity from the system-level to the subsystem-level. Indeed, the analysis of the problem dimension shows that the number of variables at the system-level can be reduced threefold with using SWORD. Since the search domain dimension and number of constraints are reduced, SWORD is more adapted to the exploratory search than MDF and the dispersion of the obtained results is lower than MDF (Fig. 13). This test-case illustrates the advantage to particularize classical MDO methods to specific design problem in order to improve the design process, both in terms of found results and robustness to initialization.

3 Introduction of Uncertainty in the Design Process

At the conceptual design stage, a designer often needs to discriminate among innovative technologies that offer high performance but at a high risk, and established technologies that offer lower performance but with less uncertainty. The early design phases are characterized by the use of low fidelity analyses as well as by the lack of knowledge about the future system design and performance. This lack of knowledge in the models is reducible by increasing the model fidelity and is classified as epistemic uncertainty; uncertainty due to variability is irreducible and classified as aleatory uncertainty. The low fidelity analyses are employed due to the necessity to evaluate a high number of system architectures to explore the design space. This global exploration results in repeated discipline evaluations which are impossible to perform at an affordable computational cost with high fidelity models. Moreover, to increase the performance of the aerospace vehicles and to decrease their costs, space agencies and industries introduce new technologies (new propellant mixture, reusable rocket engines) and new architectures (reusable first stage for launch vehicles) which present a high level of uncertainty in the early design phases. If uncertainties are not taken into account at these phases, the detailed design phase might reveal that the optimal design previously found violates specific requirements and constraints. In this case, either the designers go back to the previous design phase to find a set of design alternatives, or they perform design modifications at the detailed design phase that could result in loss of performance. Both options would lead to a loss of time and money due to the re-run of complex simulations. Incorporating uncertainties in design methodologies for aerospace vehicle design has thus become a necessity to offer improvements in terms of [62]:

- reduction of design cycle time, cost and risk,
- robustness of LVD to uncertainty along the development phase,
- increasing system performance while meeting the reliability requirements,
- robustness of the launch vehicle to aleatory events during a flight (e.g. wind gust).

In classical design processes, both epistemic and aleatory uncertainties are usually controlled by using safety margins and the design problem is deterministically solved [34]. This may lead to over conservative or unreliable solutions depending on the choice of the margins. When breakthrough technologies are used in the design process (and no historic data are available), historically chosen margins may not be appropriate and a specific process to determine them is required to improve performance or restore safety. In the first part of this section, we detail a method to select optimal design rules and margins. This method accounts for the uncertainty reduction that occurs when refining the design models in later design phases.

Another way to take the uncertainty into account is to perform a probabilistic design (i.e. reliability-based design optimization). In the MDO context, Uncertainty-based Multidisciplinary Design Optimization (UMDO) aims at solving MDO problems under uncertainty. UMDO methods are recent and still under development and they have not reached sufficient maturity to efficiently estimate the final system performance and reliability [60, 62]. Incorporating uncertainty in MDO methodologies raises a number of challenges which need to be addressed. Being able, in the early design phases, to design a multidisciplinary system taking the interactions between the disciplines into account and to handle the inherent uncertainties is often computationally prohibitive. For example, a straightforward implementation of UMDO would consist in repeated sampling of the uncertain parameters (Monte-Carlo simulations) and a MultiDisciplinary Analysis (MDA) [solving Eqs. (7)] for each sample, therefore multiplying the already important cost of MDO by the number of Monte-Carlo repetitions. In order to satisfy the designer requirements, it is necessary to find the system architecture which is optimal in terms of system performance while ensuring the robustness and reliability of the optimal system with respect to uncertainty. In the second part of this section, we describe a method to handle interdisciplinary coupling satisfaction in the presence of uncertainty and to decouple the design process.

3.1 Optimization of Design Rules and Safety Margins Taking into Account Mixed Epistemic/Aleatory Uncertainties

At the initial design stage engineers must often rely on low fidelity models that have high epistemic model uncertainty. It is important to make a distinction between *epistemic model uncertainty* and *aleatory parameter uncertainty*. Model uncertainty is defined as the discrepancy between the model and reality when the true model inputs are known [36, 46]. The model uncertainty is classified as epistemic because [22, 26]: (1) There is only a single true model, but it is unknown (2) The model uncertainty is reducible by gaining more knowledge. Parameter uncertainty is defined as uncertainty regarding the model inputs [36, 46]. In general, parameter uncertainty may be either aleatory or epistemic. Here we classify the parameter uncertainty as aleatory because [22, 26]: (1) It arises due to inherent or natural variability (2) It is irreducible. For example, wind gusts and variations of material properties are aleatory. While probability theory is generally accepted as the appropriate method for modeling aleatory uncertainty, several alternative methods have been proposed for modeling epistemic uncertainty [27]. In the proposed method, both aleatory and epistemic uncertainties are modeled using probability theory because this theory is well suited for representing model uncertainty.

When considering both aleatory parameter and epistemic model uncertainties, the objective of the design process is to find a design that is reliable with respect to the natural variability that will be experienced in service (i.e. aleatory parameter uncertainty). The fidelity of the model has no impact on the true reliability of the final design because the true reliability only depends on the true model and the aleatory uncertainty. For example, if the same design is selected using two different models, the designs still have the same true reliability regardless of the fidelity of the models used in the design process. However, the model fidelity determines the accuracy of the design reliability assessment which, in turns, affects the ability to selecting good designs. Therefore, the designer must also compensate for lack of knowledge regarding how well the low fidelity model agrees with reality (i.e. epistemic model uncertainty). The epistemic model uncertainty and aleatory parameter uncertainty are treated separately (see [31, 32, 47]) to distinguish between the quantity of interest, the true probability of failure with respect to aleatory parameter uncertainty, and the lack of knowledge regarding this quantity. The separate treatment of aleatory and epistemic uncertainties results in a distribution of probability of failure that is epistemic in nature (see Fig. 15). That is, the final design will have a single true probability of failure with respect to aleatory parameter

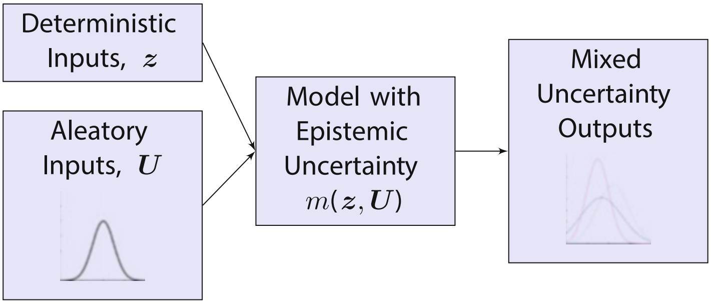

Deterministic Inputs, z
Aleatory Inputs, U
Model with Epistemic Uncertainty $m(z, U)$
Mixed Uncertainty Outputs

Fig. 15 The propagation of aleatory parameter uncertainty through a model with epistemic model uncertainty results in a different distribution for each realization of epistemic model uncertainty (represented here by *colored output curves*)

uncertainty, but it is unknown due to the epistemic model uncertainty introduced by low fidelity modeling. When the epistemic model uncertainty is very high it may force the designer to be overly conservative if, for example, the designer is compensating for worst-case scenario epistemic model uncertainty. High epistemic model uncertainty can also prevent the designer from making any decision if the distribution of possible probability of failure spans the entire zero to one range. The proposed method is an innovative approach to the challenges of design under high epistemic model uncertainty when improved modeling will be available in the future. For the sake of simplicity, this method is described in the context of single discipline design but it can be generalized for MDO.

The proposed method [49] addresses the issue of high epistemic model uncertainty by considering the anticipated uncertainty reduction from future high fidelity modeling. This approach involves first a classical deterministic optimization with uncertainty handled through safety factors, and secondly a probability analysis to assess the reliability of the solution found. Based on these steps, the design rules and safety factors are optimized in order to comply with the reliability specifications. The use of safety margin based optimization is a necessary simplification to reduce computational cost and it agrees well with current safety margin or safety-factor based design regulations [1]. Because of the presence of epistemic uncertainty, this method emulates the possible high fidelity model outcomes, considered as future tests, in order to simulate the occurrence of redesign process. To determine the necessity of redesigning, it is also convenient to formulate a test passing criterion in terms of the safety margin calculated from the possible outcomes of simulated high fidelity model (Fig. 16).The proposed approach is a bi-level optimization method (Fig. 17): at the upper-level, the safety margins are optimized to provide the optimal performance at the specified reliability requirements; at the lower-level, a complete design and redesign process is involved and can be decomposed into the following steps (Fig. 18):

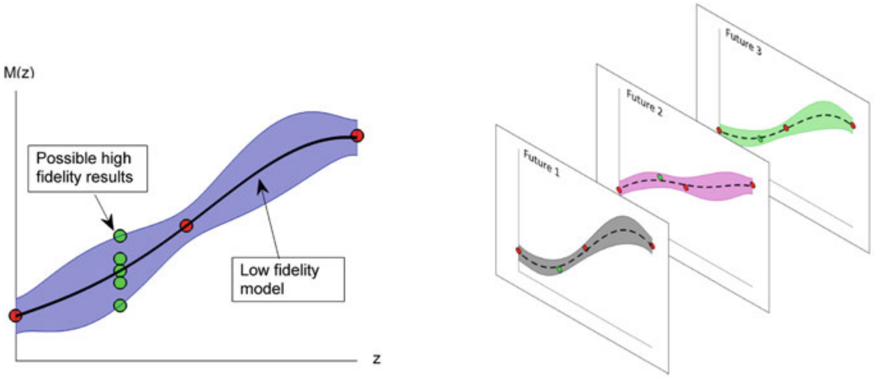
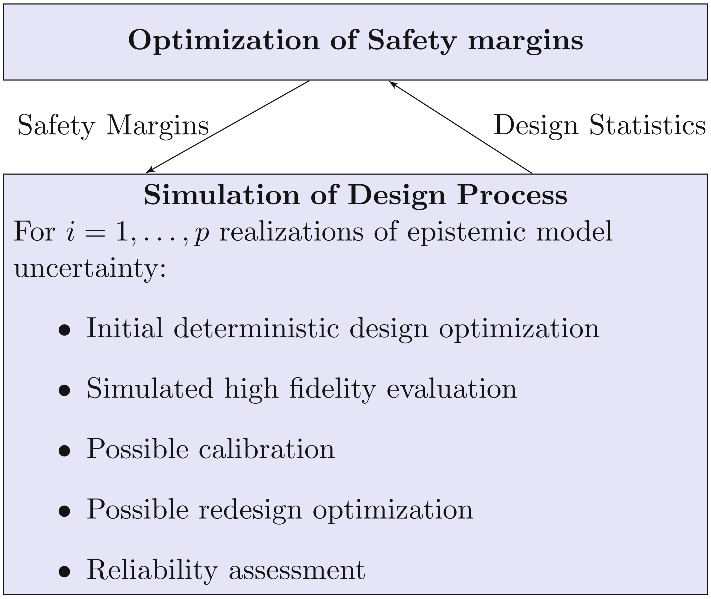


Fig. 16 Possible responses of high-fidelity model (*left*) and simulation of high-fidelity outcomes (*right*)


Optimization of Safety margins

Safety Margins

Design Statistics

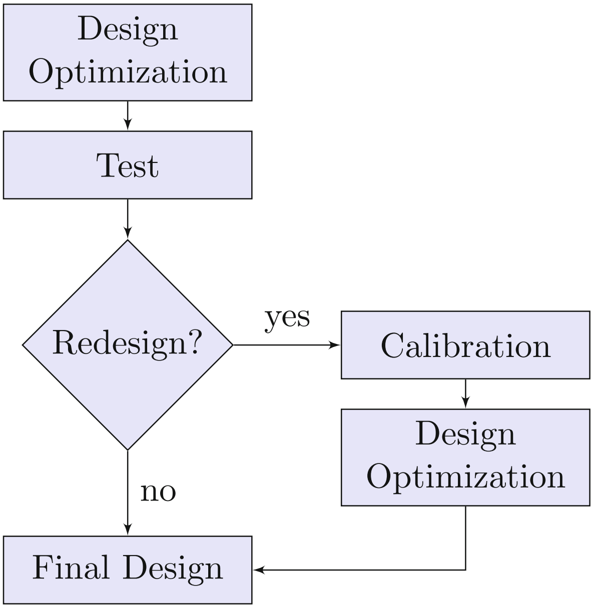
Simulation of Design Process

For $i = 1, \ldots, p$ realizations of epistemic model uncertainty:

- Initial deterministic design optimization
- Simulated high fidelity evaluation
- Possible calibration
- Possible redesign optimization
- Reliability assessment

Fig. 17 The safety margins that govern the deterministic design process are optimized by maximizing the expected performance while satisfying probabilistic constraints on expected reliability and probability of redesign (called the design statistics in the figure)

Fig. 18 The deterministic design process consists of a design optimization, a test (e.g. high fidelity evaluation), and possible calibration and redesign



1. Given safety margins, perform a deterministic design optimization: considering safety-factor vector $\mathbf{s}$, low fidelity model $\hat{m}(\cdot)$, and conservative value of aleatory uncertainty $\mathbf{u}_{det}$, the classical deterministic formulation of design problem is:

$$\min \quad f(\mathbf{z}) \tag{19}$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{z}, \mathbf{u}_{det}, \hat{m}(\mathbf{z}, \mathbf{u}_{det})) - \mathbf{s} \leq 0 \tag{20}$$

$$\mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max} \tag{21}$$

2. Simulate multiple possible outcomes of high fidelity model taking epistemic error into account at the optimal solution given by the previous step, and perform the test of redesign necessity (Fig. 16),
3. If the test calls for redesign,
   (a) Calibrate the low fidelity models taking the possible high fidelity response into account,
   (b) Perform a deterministic redesign optimization with the calibrated low-fidelity model,
4. Perform a probabilistic reliability assessment.

The safety margins that control the initial design, test passing criteria, and possible redesign are optimized to maximize the expected design performance while satisfying constraints on probability of redesign and expected reliability after the test (see Fig. 17). The design process is carried out deterministically for each realization of epistemic model uncertainty. The process of calculating the design statistics is basically a two-stage Monte-Carlo Simulation (MCS). The epistemic model uncertainty is sampled in the outer-loop and the reliability with respect to aleatory parameter uncertainty is calculated in the inner-loop. For each set of safety margins, the two-stage MCS is performed to calculate the probability of redesign, the expected design performance, and the expected probability of failure. The proposed method can help designers find reasonable designs while working under the burden of high epistemic model uncertainty. Furthermore, the method can be used to explore interesting questions such as whether it is better to start with a more conservative initial design and use redesign to improve performance if the initial design is revealed to be too conservative, or to start with a less conservative initial design and use redesign to restore safety if the initial design is revealed to be unsafe [50].

The method does not require any evaluations of the high fidelity model, only that the high fidelity evaluation and possible redesign may be performed in the future. That is, if the test is passed in the future, then the reliability of the initial design is verified to be acceptable. Similarly, alternative designs (i.e. redesigns) can be found that are reliable conditional on the specific epistemic realizations (i.e. specific test results) that will result in failing the future test (see Fig. 19). The test process can be used to not only restore safety if the initial design is revealed to be unsafe, but also to improve performance if the initial design is revealed to be overly conservative. By using the future high fidelity evaluations, the design method can be considered as the selection of multiple candidate designs instead of a single design solution. The decision to keep the initial design or redesign will be made in the future. The method considers the alternative design as a continuous epistemic random variable and relies on only specifying the optimum safety margins for locating alternative designs, rather than the explicit specification of discrete alternative designs. The preference for passing the test and keeping the initial design is controlled by a constraint on the probability of redesign in the upper-level optimization problem.

The core of the proposed method relies on the simulation of possible future test results (i.e. future high fidelity evaluations of initial design). Not only is it necessary to simulate the possible future high fidelity evaluations, but it is also necessary

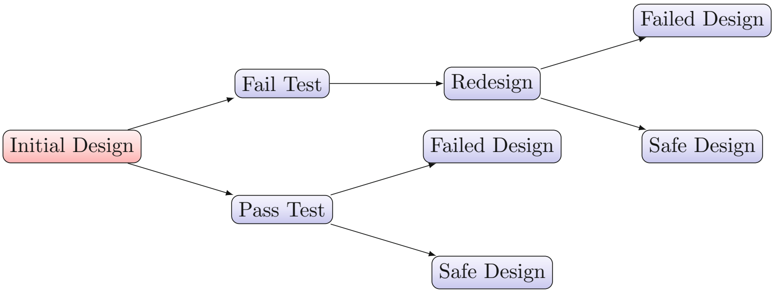


Fig. 19 The final reliability (i.e. probability of a safe design) is conditional on passing or failing a deterministic safety margin based test. Failing the test triggers a redesign process to restore safety or improve design performance

to update the distribution of epistemic model uncertainty conditional on specific realizations. By repeating the updating process for many test results it is possible to find alternative designs whose reliability is conditional on each test outcome. The method used to update the distribution of epistemic model uncertainty must account for the spatial correlations with respect to design variables. For example, it is intuitively clear that the reduction in epistemic model uncertainty from a future high fidelity evaluation is most dramatic in the immediate vicinity of the test location but decreases as the design moves away from this location. In other words, if the alternative design is dramatically different from the initial design that was tested, then the test result might not be very useful in reducing model uncertainty regarding the new design. Early work on simulating a future test and redesign relied on the strong assumption that the model bias was a fixed but unknown constant across the design space [42, 48, 59]. More recently, the method has been extended to consider fluctuations in model uncertainty and spatial correlations through the use of Gaussian Process (GP) models to represent the model uncertainty [49]. The purpose of the GP model representation of model error is twofold: (1) The GP model provides a mathematical formulation of the intuitive idea that the reduction in the variance of the epistemic model uncertainty is greatest at the test location and decreases with distance. (2) The GP model provides a probabilistic representation of epistemic model uncertainty that allows for the propagation of mixed aleatory and epistemic uncertainties. The second point is particularly important in the proposed method because the variation of epistemic model uncertainty across the design space can alter the functional relationship with respect to aleatory parameter inputs.

The proposed method can help designers find reasonable designs while working under the burden of high epistemic model uncertainty. This method has been applied in a structural design problem [48] and an aerospace vehicle design [51] which is not described in this chapter for the sake of conciseness.

3.2 *Uncertainty Multidisciplinary Design Optimization*

Taking uncertainties in MDO into account leads to a Uncertainty-based MDO (UMDO) research field [60]. As for deterministic MDO, several UMDO formulations have been proposed in literature [23, 28, 39, 43] and the generic UMDO problem can be formulated as follows:

$$\begin{aligned}
\text{min} \quad & \Xi\left[f(\mathbf{z}, \boldsymbol{\theta}_{\boldsymbol{Y}}, \mathbf{U})\right] && (22)\\
\text{w.r.t.} \quad & \mathbf{z}, \boldsymbol{\theta}_{\boldsymbol{Y}} && \\
\text{s.t.} \quad & \mathbb{K}\left[\mathbf{g}(\mathbf{z}, \boldsymbol{\theta}_{\boldsymbol{Y}}, \mathbf{U})\right] \leq 0 && (23)\\
& \forall (i,j) \in \{1,\ldots,N\}^2 \; i \neq j, \;\; \boldsymbol{\theta}_{\mathbf{Y}_{ij}} = \mathbb{M}\left[\mathbf{c}_{ij}(\mathbf{z}_i, \boldsymbol{\theta}_{\boldsymbol{Y}.i}, \mathbf{U}_i)\right] \quad \text{for statistical}-\text{based approaches} && (24)\\
& \forall (i,j) \in \{1,\ldots,N\}^2 \; i \neq j \; \boldsymbol{\theta}_{\mathbf{Y}_{ij}} = \mathbf{c}_{ij}(\mathbf{z}_i, \boldsymbol{\theta}_{\boldsymbol{Y}.i}, \mathbf{u}_i^*) \quad \text{for MPP}-\text{based approaches} && (25)\\
& \mathbf{z}_{\min} \leq \mathbf{z} \leq \mathbf{z}_{\max} && (26)
\end{aligned}$$

Several differences exist between the UMDO and the MDO formulations and are summarized in the following:

- $\mathbf{U}$ is the uncertain variable vector. $\mathbf{U}_i$ denotes the input uncertain variable vector of the discipline i and $\mathbf{U} = \bigcup_{i=1}^{N} \mathbf{U}_i$ without duplication. In this chapter, the uncertain variables are modeled with the probability theory, with known input distributions. Aleatory and epistemic uncertainties may be considered in the UMDO problem and the proposed formulation as long as they may be modeled with the probability formalism. For instance, wind gust during a rocket launch or parameter uncertainties in the modeling of the nozzle fluid flow may be sources of uncertainty. The design variables are assumed to be deterministic, and all the uncertainties are represented by $\mathbf{U}$. $(\boldsymbol{\Omega}, \boldsymbol{\sigma}_{\boldsymbol{\Omega}}, P_{\boldsymbol{\Omega}})$ is the probability space with $\boldsymbol{\Omega}$ the sample space for $\mathbf{U}$, $\boldsymbol{\sigma}_{\boldsymbol{\Omega}}$ the sigma-algebra, and $P_{\boldsymbol{\Omega}}$ the probability measure. $\phi(\cdot)$ is the joint Probability Density Function (PDF) of the uncertain variable vector $\mathbf{U}$ and the realizations of $\mathbf{U}$ are noted $\mathbf{u}$.
- Due to the presence of uncertainty, the coupling variable vector $\mathbf{Y}$ is also an uncertain variable vector and therefore a function of $\mathbf{U}$. Coupled formulations derived from MDF have been proposed to handle interdisciplinary coupling variables [34, 37, 45]. For each realization of the uncertain variables, a MDA is solved in order to compute the coupling variables ensuring multidisciplinary feasibility. However, the computational cost introduced by repeated MDA solving is too prohibitive for complex system design. In order to remove the MDA, as in deterministic approaches, decoupled strategies have been developed [23, 28, 39, 43]. Because the input coupling variables are function of the uncertainty, the decoupled optimization problem to solve has an infinite dimension. Several methods focus on these types of problems such as calculus of variations [44], optimal control [64] and shape optimization [56]. To avoid to solve an infinite dimension problem, the classical approaches involve a parameterization of the uncertain coupling variable modeling and the system-level optimizer controls only a finite number of parameters (e.g. the statistical moments, the parameters of the probability density function defining $\mathbf{Y}$, etc.). Two types of decoupled UMDO formulations exist in literature: the statistical-based approaches or the Most Probable Point (MPP)-based approaches. The statistical-based UMDO formulations [28, 39, 43] ensure the multidisciplinary feasibility for the statistical moments $\mathbb{M}$ of the coupling variables (e.g. for the expected value $\mathbb{E}$ of the coupling variables). The MPP-based UMDO formulations [23] ensure the multidisciplinary feasibility only at the Most Probable failure Point $\mathbf{u}^*$ of the uncertain variables.

  The existing UMDO formulations either rely on computationally expensive MDA to rigorously ensure coupling satisfaction, or deal with incomplete coupling conditions (coupling in terms of statistical moments, at the MPP, etc.). The moment matching formulations are interesting since they preserve some disciplinary autonomy via parallel subsystem-level uncertainty propagation and optimizations. However, the interdisciplinary couplings are satisfied only in terms of statistical moments (expected value, standard deviation or covariance matrix) of the coupling variables.

- Ξ is the uncertain objective function measure (e.g. the expected value, a weighted sum of expected value and the standard deviation [11]).

  Regarding the constraint functions, two main measures exist and can be formulated as follows:

  - the robust formulation: $\mathbb{K}\left[\mathbf{g}(\mathbf{z},\boldsymbol{\theta}_Y,\mathbf{U})\right] = \mathbb{E}\left[\mathbf{g}(\mathbf{z},\boldsymbol{\theta}_Y,\mathbf{U})\right] + \eta\sigma\left[\mathbf{g}(\mathbf{z},\boldsymbol{\theta}_Y,\mathbf{U})\right]$ with $\mathbb{E}[\mathbf{g}(\cdot)]$ and $\sigma[\mathbf{g}(\cdot)]$ the vectors of expected values and standard deviation values of the constraint functions $\mathbf{g}$ and $\eta \in \mathbb{R}^+$.
  - the reliability-based formulation: $\mathbb{K}\left[\mathbf{g}(\mathbf{z},\boldsymbol{\theta}_Y,\mathbf{U})\right] = \boldsymbol{\Lambda}\left[\mathbf{g}(\mathbf{z},\boldsymbol{\theta}_Y,\mathbf{U}) > 0\right] - \boldsymbol{\Lambda}_\mathbf{t}$ with $\boldsymbol{\Lambda}[\mathbf{g}(\cdot)]$ the vector of the measures of uncertainty for the inequality constraint function vector. The vector of the uncertainty measures of the constraints have to be at most equal to $\boldsymbol{\Lambda}_\mathbf{t}$ [2]. As the uncertain variables are modeled within the probability theory, we have for the component i of the vector of failure probabilities:

$$\mathbb{K}_i\left[g_i(\mathbf{z},\boldsymbol{\theta}_Y,\mathbf{U})\right] = \mathbb{P}_{[g_i(\mathbf{z},\boldsymbol{\theta}_Y,\mathbf{U})>0]} - \mathbb{P}_{t_i} = \int_{\mathcal{I}_i} \phi(\mathbf{u})\mathrm{d}\mathbf{u} - \mathbb{P}_{t_i} \qquad (27)$$

  with $g_i(\cdot)$ the ith component of the inequality constraint vector and $\mathcal{I}_i = \{\mathbf{U} \in \boldsymbol{\Omega} \,|\, g_i(\mathbf{z},\boldsymbol{\theta}_Y,\mathbf{U}) > 0\}$.

3.2.1 Theoretical Approach for Interdisciplinary Coupling Satisfaction in the Presence of Uncertainty

In order to avoid the repeated MDA used in MDF under uncertainty, decoupled approaches aim at propagating uncertainty on decoupled disciplines allowing one to evaluate them in parallel and to ensure coupling satisfaction by introducing equality constraints in the UMDO formulation. However, two main challenges are faced to decouple the design process:

- The handling of uncertain input coupling variable vector $\mathbf{Y}$ by the system-level optimizer. Moreover, the uncertain variables are function and infinite-dimensional problem are complex to solve.
- The handling of coupling equality constraints between the input coupling variables $\mathbf{Y}$ and the output coupling variables computed by $\mathbf{c}(\cdot)$. Equality between two uncertain variables corresponds to an equality between two functions which is difficult to implement.

In order to understand these two challenges and the approaches described afterwards, a focus on decoupled deterministic MDO formulation is necessary. Consider two disciplines i and j and one scalar feedforward coupling y_{ij} and one scalar feedback coupling y_{ji} as illustrated in Fig. 20. In deterministic decoupled MDO approach, to remove the feedforward coupling, there is only one equality constraint

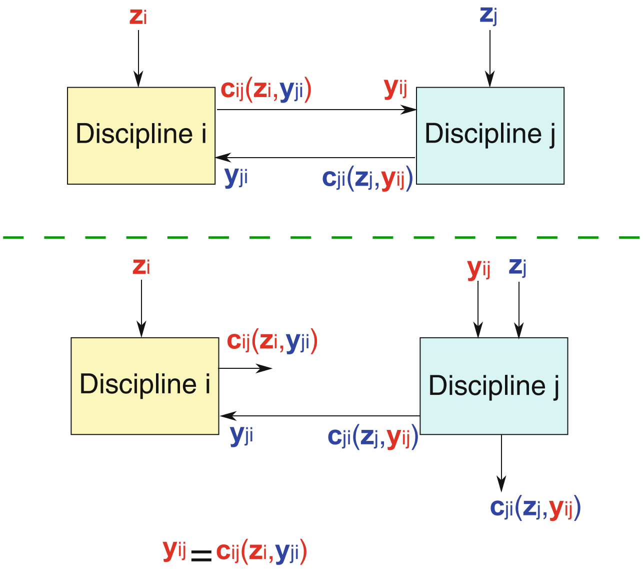


Fig. 20 Two discipline coupling handling approaches

that has to be imposed at the system-level in the optimization formulation, Eq. (28), between the input coupling variable y_{ij} and the output coupling variable $c_{ij}(\mathbf{z}_i, y_{ji})$:

$$y_{ij} = c_{ij}(\mathbf{z}_i, y_{ji}) \tag{28}$$

When introducing uncertainty, coupling satisfaction involves equality constraints between uncertain variables. An uncertain variable is a function. Two uncertain variables are equal, if and only if the two corresponding functions have the same initial and final sets and the same mappings. To ensure coupling satisfaction *in realizations*, an infinite number of equality constraints, Eq. (29), have to be imposed, one for each realization of the uncertain variables:

$$\forall \mathbf{u} \in \boldsymbol{\Omega}, \quad y_{ij} = c_{ij}(\mathbf{z}_i, y_{ji}, \mathbf{u}_i) \tag{29}$$

However, it is important to point out that even if the coupling variables are random variables, for one realization $\mathbf{u}_0$ there is in general only one converged coupling variable realization that satisfies $y_{ij_0} = c_{ij}(\mathbf{z}_i, y_{ji_0}, \mathbf{u}_0)$ ensuring multidisciplinary feasibility. Indeed, the disciplines are modeled with deterministic functions, all the uncertainties arise in the discipline inputs.

Solving an optimization problem with an infinite number of constraints is a challenging task. To overcome this issue, considering an UMDO problem of N

disciplines, we propose to introduce a new integral form for the interdisciplinary coupling constraint:

$$\forall (i,j) \in \{1,\ldots,N\}^2\ i \neq j,\ \mathbf{J}_{ij} = \int_{\mathbf{\Omega}} \left[\mathbf{c}_{ij}(\mathbf{z}_i, \mathbf{y}_{.i}, \mathbf{u}_i) - \mathbf{y}_{ij}\right]^2 \phi(\mathbf{u})\mathrm{d}\mathbf{u} = 0 \tag{30}$$

The integrals in Eq. (30) equal to zero if the input coupling variables are equal to the output coupling variables for each realization of the uncertain variables almost surely. The interdisciplinary coupling constraints $\mathbf{J}_{ij}$ may be seen as the integration of a loss function (the difference between the input and the output coupling variables) over the entire sample space. If the new interdisciplinary coupling constraints Eq. (30) are satisfied, therefore a mathematical equivalence holds with the coupled approach because, as by using MDA, the couplings verify the following system of equations:

$$\forall\ \mathbf{u} \in \mathbf{\Omega},\ \forall (i,j) \in \{1,\ldots,N\}^2\ i \neq j,\ \begin{cases} \mathbf{y}_{ij} = \mathbf{c}_{ij}(\mathbf{z}_i, \mathbf{y}_{.i}, \mathbf{u}_i) \\ \mathbf{y}_{ji} = \mathbf{c}_{ji}(\mathbf{z}_j, \mathbf{y}_{.j}, \mathbf{u}_i) \end{cases} \tag{31}$$

In order to be able to decouple the disciplines, the system-level optimizer has to control the input coupling variables $\mathbf{Y}$. In the proposed formulations, the considered scalar coupling variable y_{ij} is replaced by a surrogate model representing the coupling functional relations:

$$y_{ij} \rightarrow \hat{y}_{ij}\left(\mathbf{u}, \boldsymbol{\alpha}^{(ij)}\right) \tag{32}$$

The surrogate model $\hat{y}_{ij}\left(\mathbf{u}, \boldsymbol{\alpha}^{(ij)}\right)$, allows to model a functional representation of the dependency between the uncertain variables $\mathbf{U}$ and the input coupling variables. $\boldsymbol{\alpha}^{(ij)}$ are the surrogate model parameters. In the proposed formulations, each coupling variable that is removed is replaced by a surrogate model. The metamodels are also functions, represented by parameters that may be used to decouple the UMDO problem by letting the system-level optimizer have the control on the surrogate model coefficients. Therefore, the infinite-dimensional optimization problem is transformed into a q-dimensional optimization problem with q the number of coefficients required to model all the removed coupling variables.

We propose to model the coupling functional relations with Polynomial Chaos Expansion (PCE) [25]. Indeed, this surrogate model has been successfully used to analyze and propagate uncertainty [25]. PCE are particularly adapted to represent the input coupling variables as they are dedicated to model functions that take as input uncertain variables. The scalar coupling y_{ij} is modeled by:

$$\hat{y}_{ij}\left(\mathbf{u}, \boldsymbol{\alpha}^{(ij)}\right) = \sum_{k=1}^{d_{\mathrm{PCE}}} \alpha_{(k)}^{(ij)} \Psi_k(\mathbf{u}) \tag{33}$$

where $q = d_{\mathrm{PCE}}$ is the degree of PCE decomposition and Ψ_k is the basis of orthogonal polynomials chosen in accordance to the input uncertainty distributions.

Note that the dependency between $\hat{y}_{ij}(\cdot)$ and $\mathbf{z}$ is not taken into account in the surrogate model: $\hat{y}_{ij}(\cdot)$ is not a function of $\mathbf{z}$, it is learned only for the specific value of $\mathbf{z}$ which is the optimum of the problem. This interdisciplinary coupling satisfaction for all the realizations of the uncertain variables enables to ensure that the system is *multidisciplinary feasible*. The complex original infinite-dimensional problems are transformed into a finite-dimensional problem and the mathematical equivalence between coupled and decoupled formulations in terms of coupling satisfaction is numerically ensured.

The PCE models of the coupling functional relations is built iteratively during the system-level UMDO optimization. At the optimum, PCE models the coupling functional relations as would MDA under uncertainty do (Fig. 21). A single-level (Individual Discipline Feasible—Polynomial Chaos Expansion) and a multi-level (Multi-level Hierarchical Optimization under Uncertainty) formulations have been developed and are detailed in the following. The proposed approaches do not require any computationally expensive MDA.

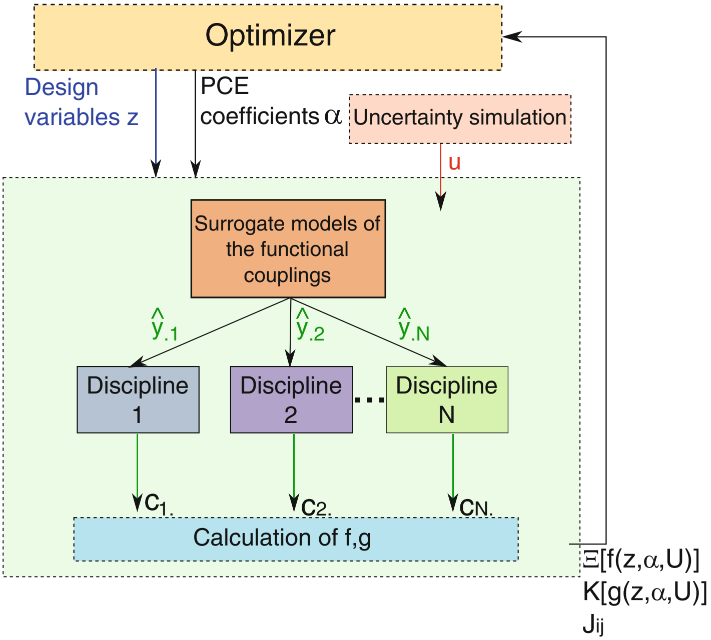


Fig. 21 IDF-PCE [16]

Individual Discipline Feasible—Polynomial Chaos Expansion (IDF-PCE)

IDF-PCE is a single-level decoupled UMDO formulation [16] which can be formulated as follows:

$$\begin{aligned}
\text{min} \quad & \Xi\left[f(\mathbf{z},\boldsymbol{\alpha},\mathbf{U})\right] && (34)\\
\text{w.r.t.} \quad & \mathbf{z},\boldsymbol{\alpha} \\
\text{s.t.} \quad & \mathbb{K}\left[\mathbf{g}(\mathbf{z},\boldsymbol{\alpha},\mathbf{U})\right] \leq 0 && (35)\\
& \forall (i,j) \in \{1,\ldots,N\}^2 \; i \neq j, \\
& \mathbf{J}_{ij} = \int_{\boldsymbol{\Omega}} \left[\mathbf{c}_{ij}\left(\mathbf{z}_i, \hat{\mathbf{y}}_{.i}\left(\mathbf{u},\boldsymbol{\alpha}^{(.i)}\right), \mathbf{u}_i\right) - \hat{\mathbf{y}}_{ij}\left(\mathbf{u},\boldsymbol{\alpha}^{(ij)}\right)\right]^2 \phi(\mathbf{u})\mathrm{d}\mathbf{u} = \mathbf{0} && (36)\\
& \mathbf{z}_{\text{min}} \leq \mathbf{z} \leq \mathbf{z}_{\text{max}} && (37)
\end{aligned}$$

with $\mathbf{J}_{ij}$ the interdisciplinary constraint vector of discipline i and $\hat{\mathbf{y}}_{.i}\left(\mathbf{u},\boldsymbol{\alpha}^{(.i)}\right)$ the PCEs of all the input coupling variables. The system-level optimizer controls the design variables $\mathbf{z}$ and the PCE coefficients of the coupling variables $\boldsymbol{\alpha}$. The dimension of the design space is therefore increased with respect to the coupled approaches, by the number of parameters $\boldsymbol{\alpha}$. To ensure the multidisciplinary feasibility at the optimum, equality constraints involving the generalization error are imposed Eq. (36). The constraints have an integral form to ensure the coupling satisfaction for all the possible realizations of the uncertain variables. If we have: $\forall (i,j) \in \{1,\ldots,N\}^2 \; \forall i \neq j, \; \mathbf{J}_{ij} = 0$, then the couplings are satisfied for all the realizations $\mathbf{u} \in \boldsymbol{\Omega}$ almost surely.

In practice, the multidimensional integrals associated to the statistical moments (expectations, standard deviations), to the coupling constraints $\mathbf{J}$ or to the probability of failure are difficult to compute. We use three techniques to estimate the statistical moments and the coupling constraints (Crude Monte Carlo, quadrature rules and decomposition of the output coupling variables over a PCE) and one to estimate the probability of failure by Subset Sampling using Support Vector Machines. Depending on the technique used to propagate uncertainty, this leads to three variants of IDF-PCE. For more details concerning IDF-PCE, one can consult [16].

Multi-level Hierarchical Optimization Under Uncertainty (MHOU)

The aim of MHOU [17] is to ease the system-level optimization process by introducing a subsystem-level optimization (Fig. 22). The formulation is inspired from SWORD. MHOU is a semi-decoupled hierarchical method that removes all the feedback interdisciplinary couplings in order to avoid the expensive disciplinary

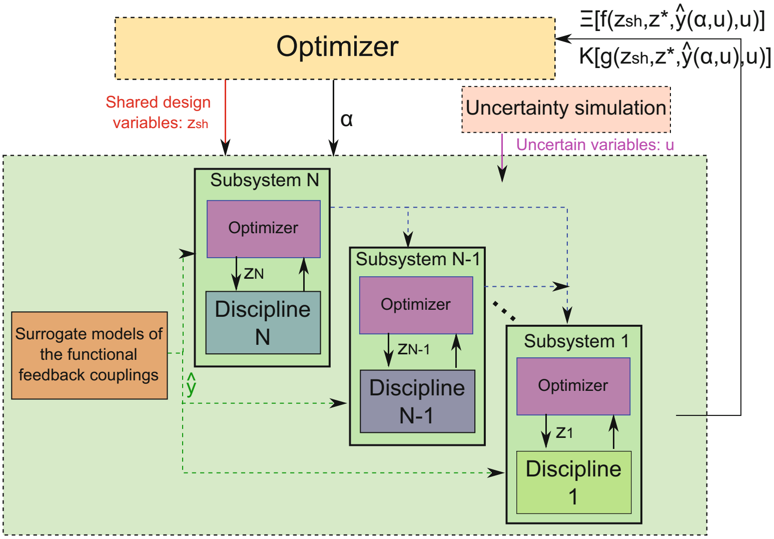


Fig. 22 Multi-level hierarchical optimization under uncertainty (MHOU)

loops through MDA. The proposed approach relies on two levels of optimization and on surrogate models in order to ensure, at the convergence of the system optimization problem, the coupling functional relations between the disciplines. It allows a hierarchical design process without any loops between the subsystems. As for SWORD, this type of decomposition is proposed in the context of LVD, but it may be generalized to other design problems.

The MHOU formulation is given by:

- At the system-level:

$$\min \quad \sum_{k=1}^{N} \Xi\left[f_k(\mathbf{z}_{sh}, \mathbf{z}_k^*, \boldsymbol{\alpha}, \mathbf{U})\right] \tag{38}$$

$$\text{w.r.t.} \quad \mathbf{z}_{sh}, \boldsymbol{\alpha}$$

$$\text{s.t.} \quad \mathbb{K}\left[\mathbf{g}(\mathbf{z}_{sh}, \mathbf{z}_k^*, \boldsymbol{\alpha}, \mathbf{U})\right] \leq 0 \tag{39}$$

$$\forall (k,j) \in \{1, \ldots, N\}^2 \; j \neq k, \;\; \mathbf{J}_{kj}(\mathbf{z}_{sh}, \mathbf{z}_k^*, \boldsymbol{\alpha}) = 0 \tag{40}$$

$$\forall k \in \{1, \ldots, N\}, \;\; \mathbb{K}\left[\mathbf{g}_k(\mathbf{z}_{sh}, \mathbf{z}_k^*, \boldsymbol{\alpha}, \mathbf{U})\right] \leq 0 \tag{41}$$

- At the subsystem-level:
  $k = N$
  While k>0

$$\text{Given} \quad \mathbf{y}_{Nk}, \ldots, \mathbf{y}_{(k+1)k}$$

For the k^{th} subsystem

$$\min \quad \Xi\left[f_k(\mathbf{z}_{sh}, \mathbf{z}_k, \boldsymbol{\alpha}, \mathbf{U})\right] \tag{42}$$

$$\text{w.r.t.} \quad \mathbf{z}_k$$

$$\text{s.t.} \quad \mathbb{K}\left[\mathbf{g}_k(\mathbf{z}_{sh}, \mathbf{z}_k, \boldsymbol{\alpha}, \mathbf{U})\right] \leq 0 \tag{43}$$

$$\forall j \in \{1, \ldots, N\}\ j \neq k,\ \mathbf{J}_{kj} =$$

$$\int_{\boldsymbol{\Omega}} \left[\mathbf{c}_{kj}\left(\mathbf{z}_{sh}, \mathbf{z}_k, \hat{\mathbf{y}}_{.k}\left(\mathbf{u}, \boldsymbol{\alpha}^{(.k)}\right), \mathbf{u}_k\right) - \hat{\mathbf{y}}_{kj}\left(\mathbf{u}, \boldsymbol{\alpha}^{(kj)}\right)\right]^2 \phi(\mathbf{u})\mathrm{d}\mathbf{u} = \mathbf{0} \tag{44}$$

$k \leftarrow k - 1$

$\mathbf{z}_k$ is the local design variable vector of discipline k and it belongs to the set $\mathcal{Z}_k$ and $\mathbf{z}_{sh}$ is the shared design variable vector between several disciplines. $\mathbf{z}_k^*$ is the optimal design variables found by the subsystem-level optimizer. This formulation allows one to optimize each subsystem separately in a hierarchical process. The system-level optimizer handles $\mathbf{z}_{sh}$ and the PCE coefficients $\boldsymbol{\alpha}$ of the feedback coupling variables. The control of PCE coefficients at the system-level allows one to remove the feedback couplings and to optimize the subsystems in sequence. The surrogate models of the functional feedback couplings provide the required input couplings to the different subsystems. The kth subsystem-level optimizer handles $\mathbf{z}_k$ and the corresponding problem aims at minimizing the subsystem contribution to the system objective while satisfying the subsystem-level constraints $\mathbb{K}\left[\mathbf{g}_k(\cdot)\right]$. The interdisciplinary coupling constraint Eq. (44) ensures the couplings whatever the realization of the uncertain variables. In MHOU formulation, Eq. (44) is only considered for $k \neq N$. This formulation is particularly suited for launch vehicle in order to decompose the design process into the different stage optimizations. The decreasing order of the discipline optimization, from N to 1 is more convenient for a launch vehicle (the last stage is optimized first, then the intermediate stages and the first one is optimized last), however, in general case any order may be adopted. In practice, the disciplines are organized to have the minimal number of feedback coupling variables in order to decrease the number of coupling variables controlled at the system-level and therefore the complexity of the optimization problem.

3.2.2 Application for Launch Vehicle Design

Two test cases have been implemented to illustrate IDF-PCE and MHOU formulations.

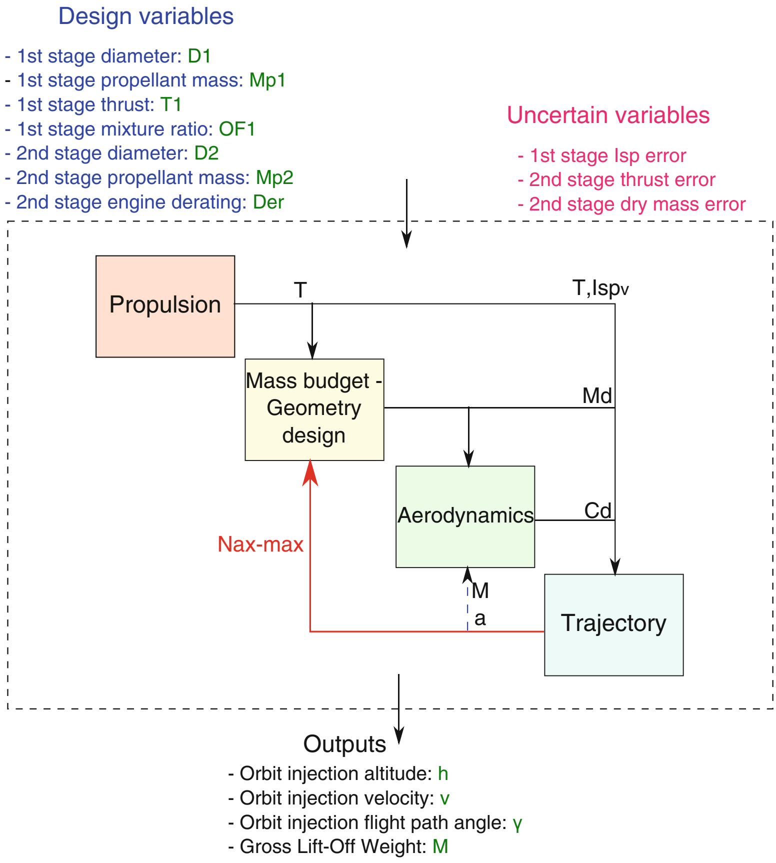

Design variables
- 1st stage diameter: D1
- 1st stage propellant mass: Mp1
- 1st stage thrust: T1
- 1st stage mixture ratio: OF1
- 2nd stage diameter: D2
- 2nd stage propellant mass: Mp2
- 2nd stage engine derating: Der
Uncertain variables
- 1st stage Isp error
- 2nd stage thrust error
- 2nd stage dry mass error
Propulsion
T
T,Ispv
Mass budget - Geometry design
Md
Aerodynamics
Cd
Nax-max
M
a
Trajectory
Outputs
- Orbit injection altitude: h
- Orbit injection velocity: v
- Orbit injection flight path angle: γ
- Gross Lift-Off Weight: M

Fig. 23 Design Structure Matrix for the two stage launch vehicle [16]

First Test Case: Comparison of IDF-PCE and MDF on a Two-Stage-to-Orbit Launch Vehicle Design Problem

The first test case [16] consists in designing a two-stage-to-orbit launch vehicle to inject a payload of 4000kg into a Geostationary Transfer Orbit from Kourou (French Guyana). MDF and IDF-PCE are compared. The LVD process consists of four disciplines: propulsion, mass budget and geometry design, aerodynamics and trajectory, using low-fidelity models [19, 33, 57] (Fig. 23). The expected value of the Gross Lift-Off Weight of the launch vehicle has to be minimized. The problem involves design variables and is initialized at a given baseline (Table 2). Three aleatory uncertain variables are present:

- Second stage dry mass error (mass and sizing discipline),
- First stage specific impulse error (propulsion discipline),
- Second stage thrust error (propulsion discipline).

Table 2 Design variables for the two stage launch vehicle

Design variables	Symbols
Stage diameters	D_1, D_2
Stage propellant masses	M_{p1}, M_{p2}
First stage thrust	T_1
First stage mixture ratio	OF_1
Second stage engine derating coefficient	Der

These errors are additional terms to the nominal value of specific impulse (Isp_{v10}), of dry mass (Me_0) and thrust (T_{20}).

One inequality constraint is considered. It is an output of the trajectory discipline and corresponds to the probability of failure of the mission (taking into account the altitude h, velocity v and flight path angle γ of the injection point). This probability of failure has to be lower than 5×10^{-2}. A failure occurs when the payload is injected outside a closed ball around the target injection point defined in the rotating frame by: $h_t = 250\,\text{km}$, $v_t = 9.713\,\text{km/s}$ and $\gamma_t = 0°$. The radius of the ball corresponds to the injection tolerances and is set to be at 1 % of the target altitude, at 0.5 % of the target velocity and at 0.4° for the target flight path angle. The uncertainty propagation is performed with Crude Monte-Carlo (CMC). A pattern search optimization algorithm [5] is used to solve both MDF and IDF-PCE problems.

Results

Both MDF and IDF-PCE converge to the same optimum (163.7t), and the constraints are satisfied. The mean of the error between the input and the output load factor (coupling variable) is of 0.1 % in IDF-PCE. IDF-PCE converges 11 times faster than MDF to the optimum as it does not require any MDA (Fig. 24). For the optimal launch vehicle, the results of uncertainty propagation for trajectory altitude are represented in Figs. 25 and 26.

Comparison Between MDO and UMDO Solutions

In order to stress the need of taking the uncertainties into account in the early design phase, the deterministic MDO problem has been solved considering the uncertainties fixed to their mean values [16]. The found optimum is 158.21t (5.5t lower than the solution taking the uncertainty into account). The optimal nominal (i.e. without uncertainty) trajectory altitude profile is represented in Fig. 27. For the deterministic optimal launch vehicle, a propagation of uncertainty is performed by CMC and MDA with the same uncertainties as considered in the UMDO problem. In Fig. 28, the trajectory altitude is represented for CMC realizations of the uncertain variables. The deterministic optimal launch vehicle is not robust to the presence of uncertainty as the injection altitude is scattered between 200 and

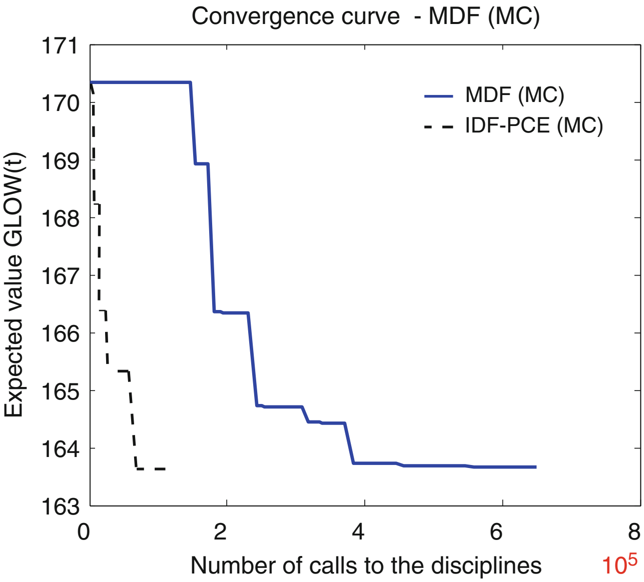


Fig. 24 Convergence curves with the points satisfying the constraints

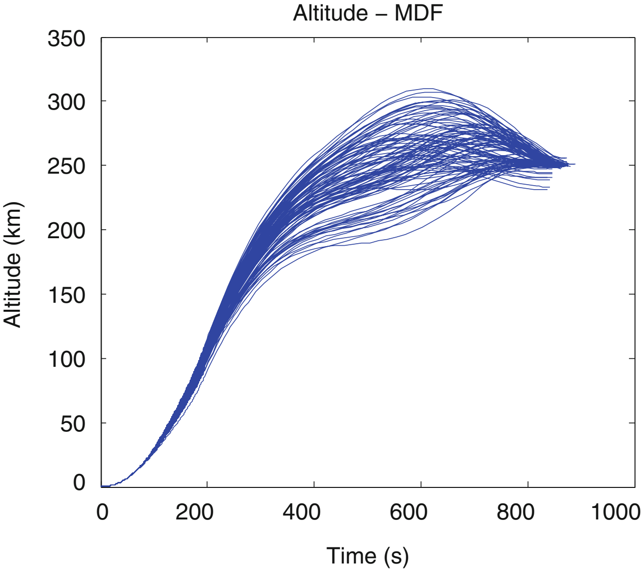


Fig. 25 Optimal trajectory altitude under uncertainty—MDF

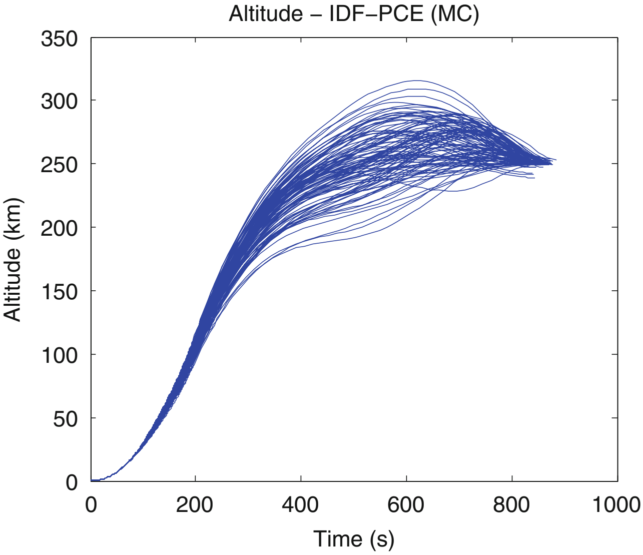


Fig. 26 Optimal trajectory altitude under uncertainty—IDF-PCE

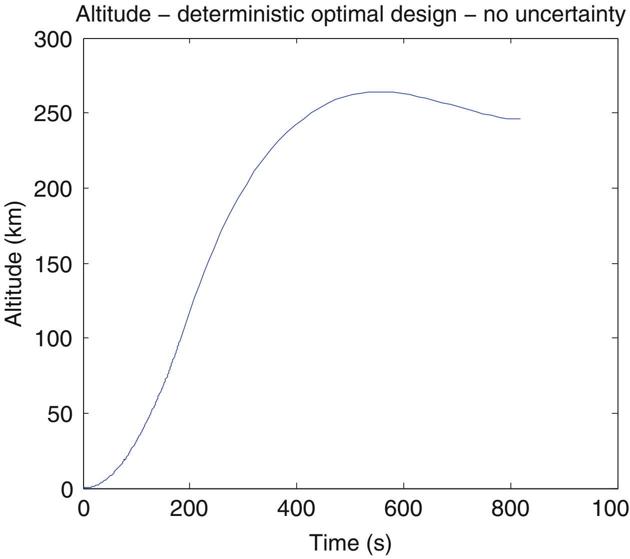


Fig. 27 Trajectory altitude for the deterministic optimal launch vehicle, no uncertainty

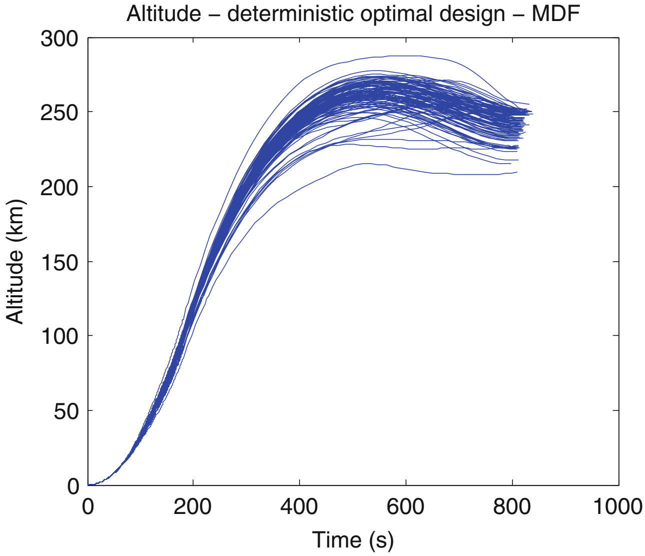


Fig. 28 Uncertainty propagation—trajectory altitude—deterministic optimal launch vehicle

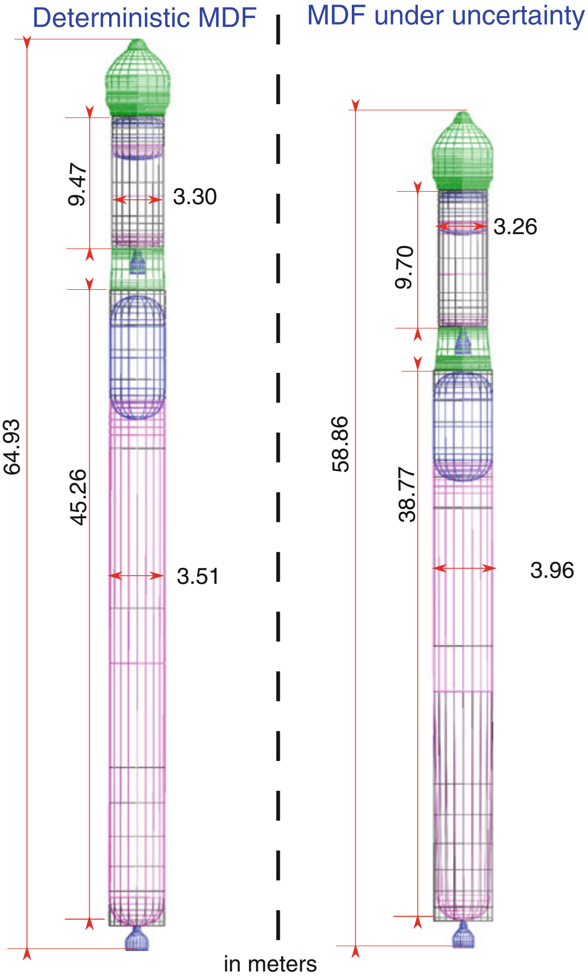
250 km due the lack of propellant to reach the injection point. Figure 25 highlights the robustness of the UMDO found solution compared to the deterministic one. The deterministic MDF and the MDF under uncertainty optimal launch vehicle dimensions are represented in Fig. 29.

Second Test-Case: Comparison of IDF-PCE, MDF and MHOU on a Multi-stage Sounding Rocket Design Problem

This LVD test case consists in designing a sounding rocket with two solid stages to launch a payload of 800 kg from Kourou that has to reach at least an altitude of 300 km. Sounding rockets carry scientific experiments into space along a parabolic trajectory. Their overall time in space is brief and the cost factor makes sounding rockets an interesting alternative to heavier launch vehicles as they are sometimes more appropriate to successfully carry out a scientific mission and are less complex to design. Four disciplines are involved in the considered test case, the propulsion, the mass budget and geometry design, the aerodynamics and the trajectory (Fig. 30) [19, 33, 57]. The sounding rocket design is decomposed into two subsystems, one for each stage (Table 3). MHOU enables a hierarchical design process decomposed into two teams, one for each sounding rocket stage. On this test case, MDF, IDF-PCE and MHOU are compared (Fig. 31).

The uncertain variables taken into account are the first stage combustion regression rate coefficient $\mathcal{N}(3.99, 0.05)$ in cm/s/MPa$^{0.3}$ and the second stage dry mass error $\mathcal{N}(0, 50)$ in kg. The uncertainty on the combustion model through

Fig. 29 Comparison of optimal deterministic MDF and MDF under uncertainty launch vehicles



the combustion regression rate results in uncertainty on the first stage thrust. The mission has to ensure that the payload reaches at least an altitude of 300 km (with a probability of failure of 3×10^{-2}). CMA-ES optimization algorithm [30] is used at the subsystem-level for MHOU. The same feasible baseline is considered as initialization for the three methods. The baseline corresponds to the deterministic optimal solution of the two stage sounding rocket problem (Figs. 32, 33, and 34) found by a deterministic MDF approach. However, this solution is not robust to the presence of uncertainty. Indeed, the deterministic optimal solution does not succeed to reach with a probability of failure lower than 3×10^{-2} an altitude of 300 km, the failure rate is around 70 %.

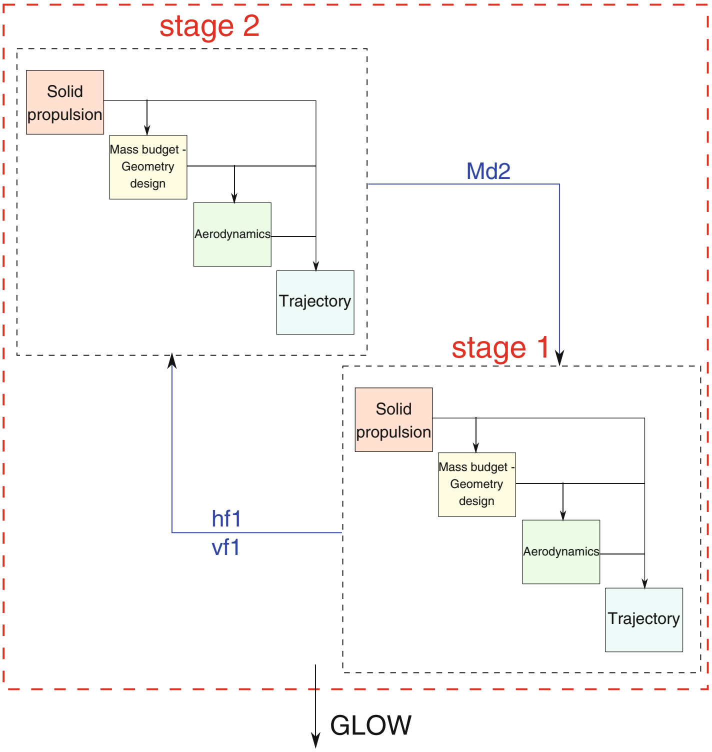


Fig. 30 Design Structure Matrix for the two stage sounding rocket

Table 3 Design variables for the two stage sounding rocket

Design variables	Symbols
Stage diameters	D_1, D_2
Stage propellant masses	M_{p1}, M_{p2}
Stage nozzle expansion ratio	ϵ_1, ϵ_2
Stage grain relative length	RL_1, RL_2
Stage combustion depth	W_1, W_2

Results

MHOU (6.68t) and IDF-PCE (6.88t) presents better characteristics in terms of quality of objective function than MDF (7.07t) for a fixed discipline evaluation budget (Fig. 31). MDF, IDF-PCE and MHOU solutions satisfy the constraints especially the apogee altitude of 300 km as illustrated in Fig. 33 for MDF and MHOU. Only 2.9 % of the trajectories do not reach the required apogee altitude. MHOU ensures interdisciplinary coupling satisfaction for the feedback couplings as illustrated by the comparison of the couplings found respectively by the coupled approach and the decoupled approach for the optimal solution found by MHOU. The same coupling satisfaction is found for IDF-PCE. The separation altitude and

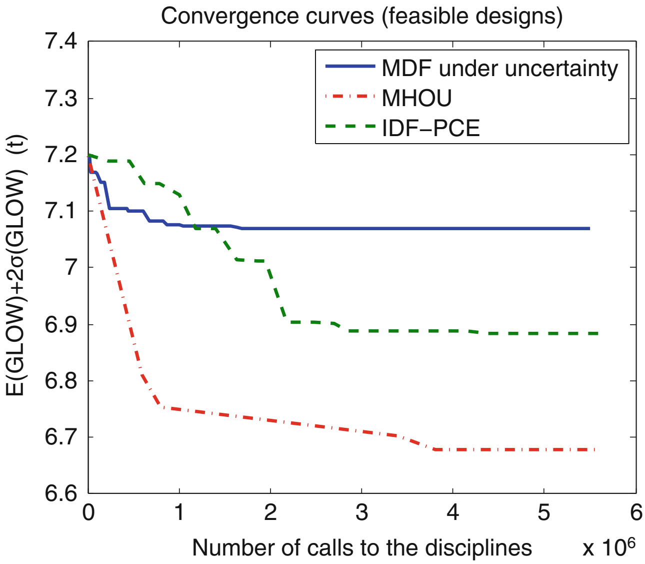


Fig. 31 Convergence curves with the points satisfying the constraints

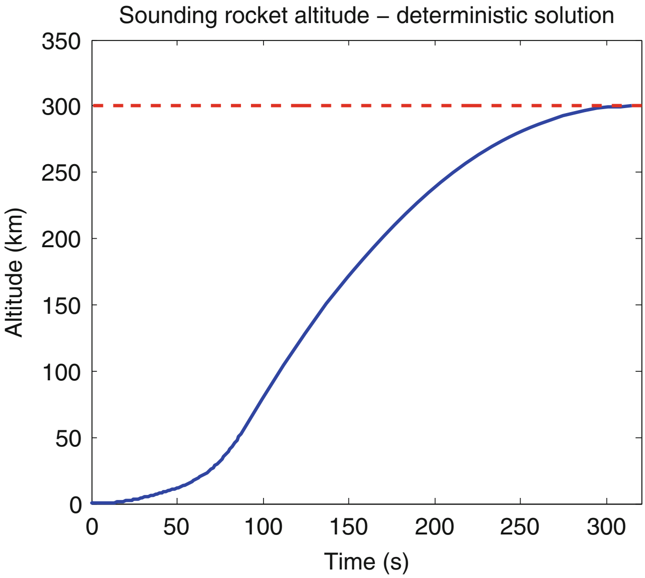


Fig. 32 Optimal sounding rocket altitude without uncertainty

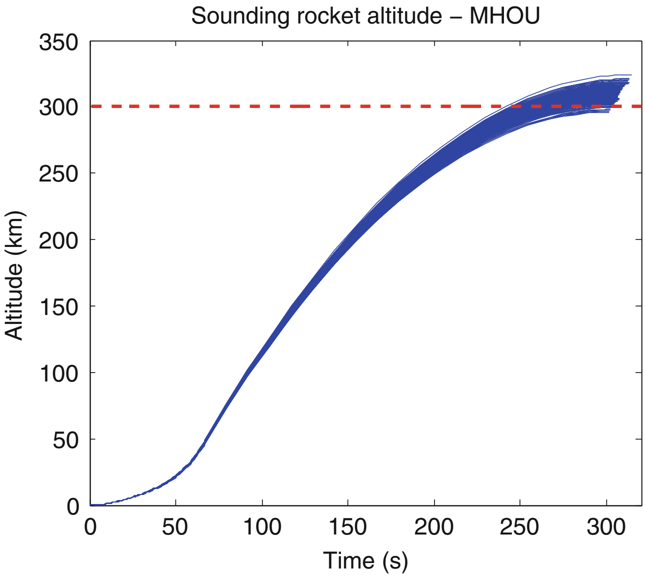


Fig. 33 Optimal sounding rocket altitude

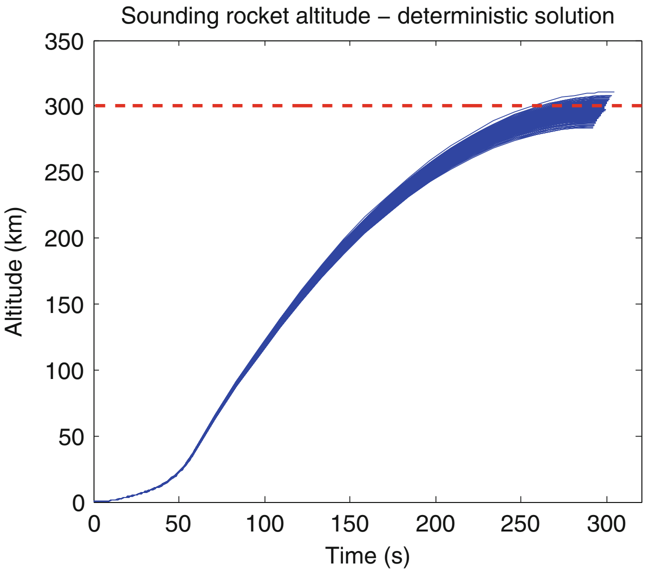


Fig. 34 Deterministic optimal sounding rocket altitude in the presence of uncertainty

Fig. 35 Distribution of the separation altitude for the optimal MHOU solution—by MDA

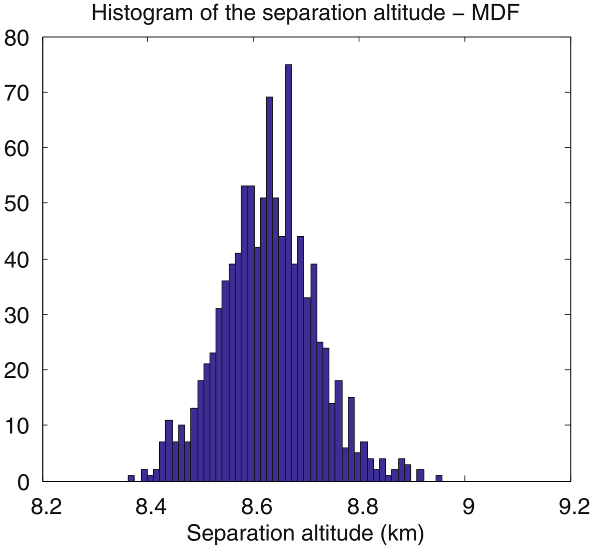
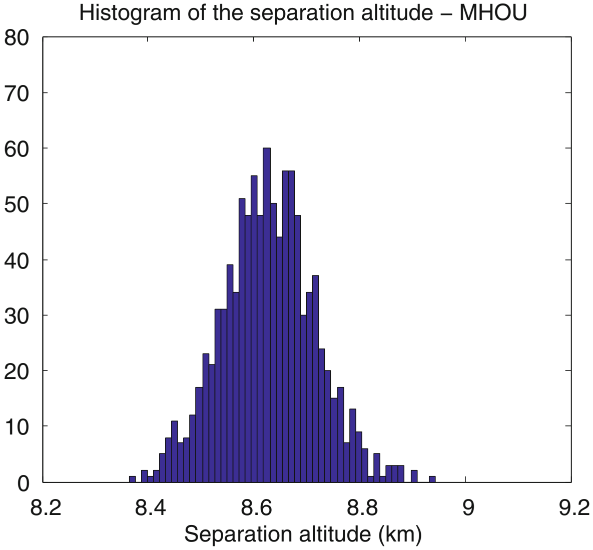


Fig. 36 Distribution of the separation altitude for the optimal MHOU solution



velocity distributions for the optimal MHOU found solution are similar by using MDA or MHOU (Figs. 35, 36, 37, and 38). Moreover, the interdisciplinary coupling error for the separation altitude and velocity are represented in Figs. 39 and 40. The coupling error is always lower than 2 % and concentrated around 0–0.5 %. The design space dimension for the system-level is increased from 10 for MDF to 13 for MHOU, however it enables multi-level optimization where each stage

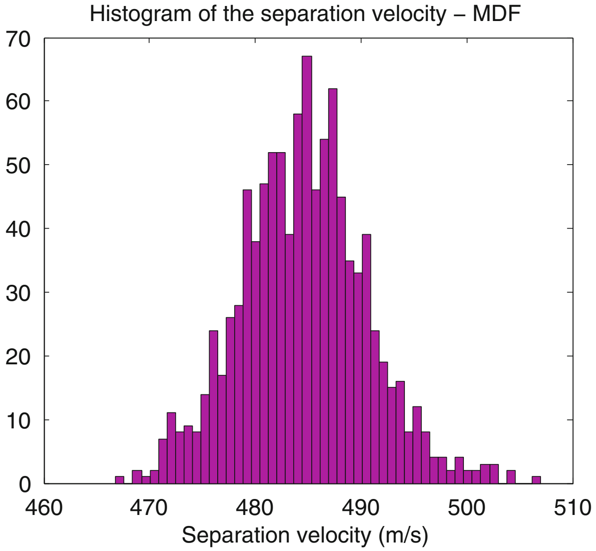


Fig. 37 Distribution of the separation velocity for the optimal MHOU solution—by MDA

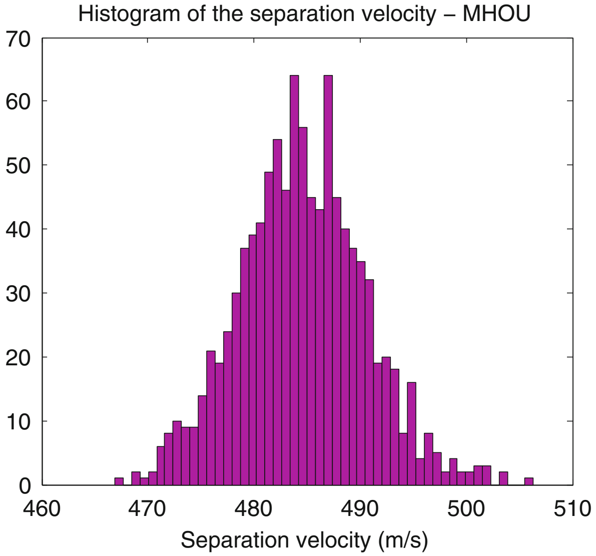


Fig. 38 Distribution of the separation velocity for the optimal MHOU solution

subsystem handles its local design variables. For IDF-PCE, the dimension of the system-level design space is 22. Thanks to the two levels of optimization, MHOU allows to converge to a better optimum than IDF-PCE in this test case while enabling decoupled design strategy and autonomy to each engineering team working on each stage.

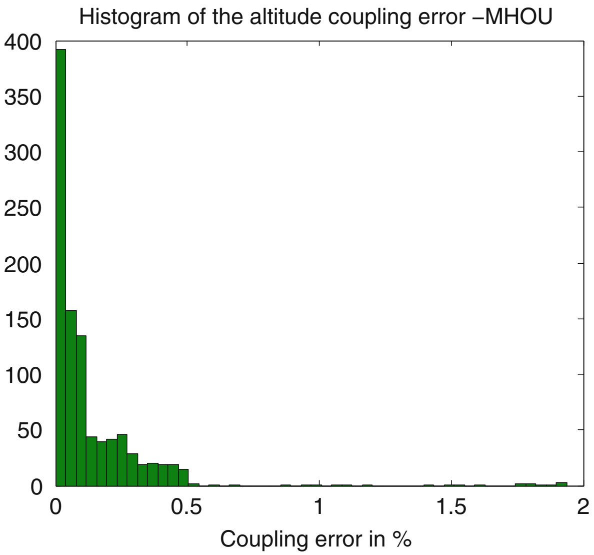


Fig. 39 Distribution of the altitude coupling error MHOU

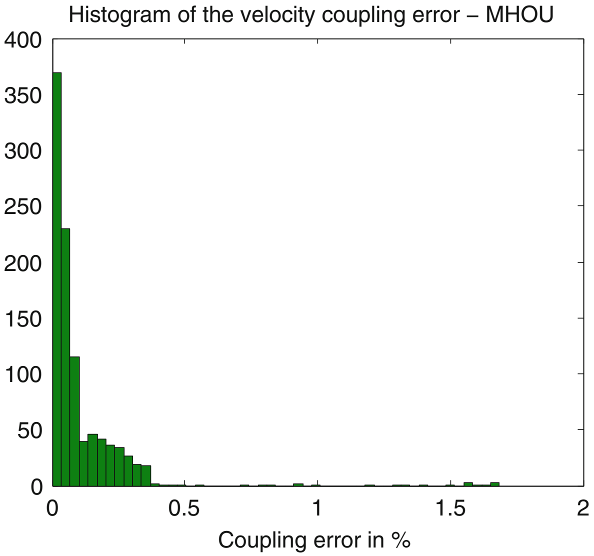


Fig. 40 Distribution of the velocity coupling error MHOU

4 Conclusion

This chapter describes several methods to handle multidisciplinary and uncertainty aspects in the context of aerospace vehicle design process. Such processes are complex and present some specificities (e.g. predominance of trajectory for LVD) that stress the need to adapt existing MDO methods to exploit these latter and improve the problem solving efficiency. A dedicated hierarchical MDO method

(SWORD) has been described in this chapter and compared to classical MDF on a three-stage-to-orbit launch vehicle design problem. The results of this comparison shown that such a dedicated MDO method allows to obtain a better optimum with respect MDF with less computation time. In addition to the multidisciplinary aspects, considering both epistemic and aleatory uncertainties in the design process is primordial in order to assess the designed vehicle performance and to ensure its reliability. For that purpose, a method allowing to bridge the gap between classical deterministic optimization and full probabilistic optimization has been described. The proposed bi-level optimization approach optimizes the safety factors at the upper-level and perform a full design/redesign process at the lower-level, providing for the designer with a set of optimal design rules satisfying the reliability requirements without having overly conservative designs. In the third part of this chapter, one single-level (IDF-PCE) and one multi-level (MHOU) formulations have been proposed in order to solve MDO problems in the presence of uncertainty. These methods allow to ensure the interdisciplinary functional coupling satisfaction for all the realizations of the uncertain variables. These approaches have been compared to MDF on two LVD problems and allow to obtain a better optimum with a less computational cost than classical MDF.

References

1. §25.303: Factor of safety. In: Federal Aviation Regulations. Federal Aviation Administration, Washington (2015)
2. 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)
3. Alexandrov, N.M.: Multilevel methods for MDO. In: Multidisciplinary Design Optimization: State of the Art, pp. 79–89. SIAM, Philadelphia (1997)
4. Allison, J., Kokkolaras, M., Zawislak, M., Papalambros, P.Y.: On the use of analytical target cascading and collaborative optimization for complex system design. In: 6th World Congress on Structural and Multidisciplinary Optimization, Rio de Janeiro, pp. 3091–3100 (2005)
5. Audet, C., Dennis Jr., J.E.: Analysis of generalized pattern searches. SIAM J. Optim. **13**(3), 889–903 (2002)
6. Balesdent, M.: Multidisciplinary design optimization of launch vehicles. PhD thesis, Ecole Centrale de Nantes (2011)
7. Balesdent, M., Bérend, N., Dépincé, P.: Stagewise multidisciplinary design optimization formulation for optimal design of expendable launch vehicles. J. Spacecr. Rocket. **49**, 720–730 (2012)
8. Balesdent, M., Bérend, N., Dépincé, P., Chriette, A.: A survey of multidisciplinary design optimization methods in launch vehicle design. Struct. Multidiscip. Optim. **45**(5), 619–642 (2012)
9. Balesdent, M., Bérend, N., Dépincé, P.: New multidisciplinary design optimization approaches for launch vehicle design. Proc. Inst. Mech. Eng. G J. Aerosp. Eng. **227**(10), 1545–1555 (2013)
10. Balling, R.J., Sobieszczanski-Sobieski, J.: Optimization of coupled systems-a critical overview of approaches. AIAA J. **34**(1), 6–17 (1996)
11. Baudoui, V.: Optimisation robuste multiobjectifs par modèles de substitution. PhD thesis, ISAE-Institut Supérieur de l'Aéronautique et de l'Espace (2012)

12. Blair, J., Ryan, R., Schutzenhofer, L.: Launch Vehicle Design Process: Characterization, Technical Integration, and Lessons Learned. NASA, Langley Research Center, Isakowitz (2001)
13. Braun, R., Moore, A., Kroo, I.: Use of the collaborative optimization architecture for launch vehicle design. In: 6th Symposium on Multidisciplinary Analysis and Optimization, Bellevue, pp. 306–318 (1996)
14. Braun, R., Moore, A., Kroo, I.: Use of the collaborative optimization architecture for launch vehicle design. In: 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue (1996)
15. Breitkopf, P., Coelho, R.F.: Multidisciplinary Design Optimization in Computational Mechanics. Wiley, Hoboken (2013)
16. Brevault, L., Balesdent, M., Bérend, N., Le Riche, R.: Decoupled MDO formulation for interdisciplinary coupling satisfaction under uncertainty. AIAA J. **54**(1), 186–205 (2016)
17. Brevault, L., Balesdent, M., Bérend, N., Le Riche, R.: Multi-level hierarchical MDO formulation with functional coupling satisfaction under uncertainty, application to sounding rocket design. In: 11th World Congress on Structural and Multidisciplinary Optimization, Sydney (2015)
18. Brown, N., Olds, R.: Evaluation of multidisciplinary optimization techniques applied to a reusable launch vehicle. J. Spacecr. Rocket. **43**, 1289–1300 (2006)
19. Castellini, F.: Multidisciplinary design optimization for expendable launch vehicles. PhD thesis, Politecnico de Milano (2012)
20. Coelho, R.F., Breitkopf, P., Knopf-Lenoir, C., Villon, P.: Bi-level model reduction for coupled problems. Struct. Multidiscip. Optim. **39**(4), 401–418 (2009)
21. Cramer, E.J., Dennis Jr., J., Frank, P.D., Lewis, R.M., Shubin, G.R.: Problem formulation for multidisciplinary optimization. SIAM J. Optim. **4**(4), 754–776 (1994)
22. Der Kiureghian, A., Ditlevsen, O.: Aleatory or epistemic? Does it matter? Struct. Saf. **31**(2), 105–112 (2009)
23. Du, X., Guo, J., Beeram, H.: Sequential optimization and reliability assessment for multidisciplinary systems design. Struct. Multidiscip. Optim. **35**(2), 117–130 (2008)
24. El Majd, B.A., Desideri, J.-A., Habbal, A.: Optimisation de forme fluide-structure par un jeu de Nash. Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées **13**, 3–15 (2010)
25. Eldred, M.: Recent advances in non-intrusive polynomial chaos and stochastic collocation methods for uncertainty analysis and design. In: AIAA Structures, Structural Dynamics, and Materials Conference, Palm Springs (2009)
26. Ferson, S., Ginzburg, L.R.: Different methods are needed to propagate ignorance and variability. Reliab. Eng. Syst. Saf. **54**(2–3), 133–144 (1996)
27. Ferson, S., Joslyn, C.A., Helton, J.C., Oberkampf, W.L., Sentz, K.: Summary from the epistemic uncertainty workshop: consensus amid diversity. Reliab. Eng. Syst. Saf. **85**(1–3), 355–369 (2004)
28. Ghosh, S., Lee, C.H., Mavris, D.N.: Covariance matching collaborative optimization for uncertainty-based multidisciplinary aircraft design. In: 15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Atlanta (2014)
29. Haftka, R.T., Watson, L.T.: Multidisciplinary design optimization with quasiseparable subsystems. Optim. Eng. **6**(1), 9–20 (2005)
30. Hansen, N., Müller, S., Koumoutsakos, P.: Reducing the time complexity of the derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). Evol. Comput. **11**(1), 1–18 (2003)
31. Helton, J.C.: Treatment of uncertainty in performance assessments for complex systems. Risk Anal. **14**(4), 483–511 (1994)
32. Hoffman, F.O., Hammonds, J.S.: Propagation of uncertainty in risk assessments: the need to distinguish between uncertainty due to lack of knowledge and uncertainty due to variability. Risk Anal. **14**(5), 707–712 (1994)

33. Humble, R.W., Henry, G.N., Larson, W.J., et al.: Space Propulsion Analysis and Design, vol. 1. McGraw-Hill, New York (1995)
34. Jaeger, L., Gogu, C., Segonds, S., Bes, C.: Aircraft multidisciplinary design optimization under both model and design variables uncertainty. J. Aircraft **50**, 528–538 (2013)
35. Kennedy, G., Martins, J.: A parallel aerostructural optimization framework for aircraft design studies. Struct. Multidiscip. Optim. **50**(6), 1079–1101 (2014)
36. Kennedy, M.C., O'Hagan, A.: Bayesian calibration of computer models. J. R. Stat. Soc. Ser. B Stat. Methodol. **63**(3), 425–464 (2001)
37. Koch, P.N., Wujek, B., Golovidov, O., Simpson, T.W.: Facilitating probabilistic multidisciplinary design optimization using Kriging approximation models. In: 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis & Optimization (September 2002), vol. 5415. AIAA paper (2002)
38. Kroo, I.: MDO for large-scale design. In: Multidisciplinary Design Optimization: State-of-the-Art, pp. 22–44. SIAM, Philadelphia (1997)
39. Liu, H., Chen, W., Kokkolaras, M., Papalambros, P.Y., Kim, H.M.: Probabilistic analytical target cascading: a moment matching formulation for multilevel optimization under uncertainty. J. Mech. Des. **128**(2), 991–1000 (2006)
40. Marler, R.T., Arora, J.S.: Survey of multi-objective optimization methods for engineering. Struct. Multidiscip. Optim. **26**(6), 369–395 (2004)
41. Martins, J.R., Lambe, A.B.: Multidisciplinary design optimization: a survey of architectures. AIAA J. **51**(9), 2049–2075 (2013)
42. Matsumura, T., Haftka, R.T.: Reliability based design optimization modeling future redesign with different epistemic uncertainty treatments. J. Mech. Des. **135**(9), 091006–091006 (2013)
43. McAllister, C.D., Simpson, T.W.: Multidisciplinary robust design optimization of an internal combustion engine. J. Mech. Des. **125**(1), 124–130 (2003)
44. Noton, A.R.M.: Introduction to Variational Methods in Control Engineering. Elsevier, Amsterdam (2013)
45. Oakley, D.R., Sues, R.H., Rhodes, G.S.: Performance optimization of multidisciplinary mechanical systems subject to uncertainties. Probab. Eng. Mech. **13**(1), 15–26 (1998)
46. O'Hagan, A., Oakley, J.E.: Probability is perfect, but we can't elicit it perfectly. Reliab. Eng. Syst. Saf. 85(1–3), 239–248 (2004)
47. Paté-Cornell, M.E.: Uncertainties in risk analysis: six levels of treatment. Reliab. Eng. Syst. Saf. 54(2–3), 95–111 (1996)
48. Price, N.B., Matsumura, T., Haftka, R.T., Kim, N.H.: Deciding how conservative a designer should be: simulating future tests and redesign. In: 16th AIAA Non-Deterministic Approaches Conference, National Harbor, MD (2014). American Institute of Aeronautics and Astronautics
49. Price, N.B., Balesdent, M., Defoort, S., Le Riche, R., Kim, N.H., Haftka, R.: Simulating future test and redesign considering epistemic model uncertainty. In: 18th AIAA Non-Deterministic Approaches Conference. AIAA Science and Technology Forum and Exposition, San Diego (2016)
50. Price, N.B., Kim, N.-H., Haftka, R.-T., Balesdent, M., Defoort, S., Le Riche, R.: Deciding degree of conservativeness in initial design considering risk of future redesign. J. Mech. Design (2016). ASME, Published online
51. Price N.B.: Optimizing the safety margins governing a deterministic process while considering the effects of a future test and redesign on epistemic model uncertainty. PhD thesis, University of Florida (2016)
52. Sobieszczanski-Sobieski, J.: Optimization by Decomposition: Step from Hierarchic to Non-Hierarchic Systems. NASA Technical Report CP-3031 (1988)
53. Sobieszczanski-Sobieski, J., Haftka, R.: Multidisciplinary aerospace design optimization: survey of recent developments. Struct. Multidiscip. Optim. **14**(1), 1–23 (1997)
54. Sobieszczanski-Sobieski, J., Agte, J., Sandusky, R.: Bi-Level Integrated System Synthesis (BLISS). Langley Research Center, Hampton, Virginia. NASA Technical Report TM-1998-208715 (1998)

55. Sobieszczanski-Sobieski, J., Agte, J., Sandusky, R.: Bi-Level Integrated System Synthesis (BLISS). NASA/TM-1998-208715 (1998)
56. Sokolowski, J., Zolesio, J.-P.: Introduction to Shape Optimization. Springer, Heidelberg (1992)
57. Sutton, G.P., Biblarz, O.: Rocket Propulsion Elements. Wiley, New York (2010)
58. Tedford, N.P., Martins, J.R.: Benchmarking multidisciplinary design optimization algorithms. Optim. Eng. **11**(1), 159–183 (2010)
59. Villanueva, D., Haftka, R.T., Sankar, B.V.: Accounting for future redesign to balance performance and development costs. Reliab. Eng. Syst. Saf. **124**, 56–67 (2014)
60. Yao, W., Chen, X., Luo, W., van Tooren, M., Guo, J.: Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles. Prog. Aerosp. Sci. **47**(6), 450–479 (2011)
61. Yi, S.-I., Shin, J.-K., Park, G.: Comparison of MDO methods with mathematical examples. Struct. Multidiscip. Optim. **35**(5), 391–402 (2008)
62. Zang, T.A., Hemsch, M.J., Hilburger, M.W., Kenny, S.P., Luckring, J.M., Maghami, P., Padula, S.L., Stroud, W.J.: Needs and Opportunities for Uncertainty-Based Multidisciplinary Design Methods for Aerospace Vehicles. NASA, Langley Research Center, Hampton (2002)
63. Zeitlin, N.P., Schaefer, S., Brown, B., Clements, G., Fawcett, M.: NASA ground and launch systems processing technology area roadmap. In: 2012 IEEE Aerospace Conference, pp. 1–19. IEEE, Big Sky (2012)
64. Zhou, K., Doyle, J.C., Glover, K., et al.: Robust and Optimal Control, vol. 40. Prentice Hall, Upper Saddle River (1996)

Using Direct Transcription to Compute Optimal Low Thrust Transfers Between Libration Point Orbits

John T. Betts

Abstract The direct transcription method has been used to solve many challenging optimal control problems. One such example involves the calculation of a low thrust orbit transfer between libration point orbits. The recent implementation of high order discretization techniques is first described and then illustrated by computing optimal low thrust trajectories between orbits about the L_1 and L_2 Earth-Moon libration points.

Keywords Direct Transcription • Optimal Control • Low-thrust Transfer • Lobatto Methods • Libration Point Orbits

1 The Optimal Control Problem

The primary focus of this paper is the presentation of efficient numerical methods to solve the optimal control problem. The goal is to choose the control functions $\mathbf{u}(t)$ to minimize the objective

$$F = \int_{t_I}^{t_F} w\left[\mathbf{y}(t), \mathbf{u}(t), \mathbf{p}, t\right] dt \tag{1}$$

subject to the state equations

$$\dot{\mathbf{y}} = \mathbf{f}[\mathbf{y}(t), \mathbf{u}(t), \mathbf{p}, t] \tag{2}$$

and the boundary conditions

$$\mathbf{0} = \boldsymbol{\psi}_I[\mathbf{y}(t_I), \mathbf{u}(t_I), \mathbf{p}, t_I] \tag{3}$$

$$\mathbf{0} = \boldsymbol{\psi}_F[\mathbf{y}(t_F), \mathbf{u}(t_F), \mathbf{p}, t_F]. \tag{4}$$

J.T. Betts (✉)
Applied Mathematical Analysis, LLC, P.O. Box 1135, Issaquah, WA 98027, USA
http://www.appliedmathematicalanalysis.com/
e-mail: john.betts@ama-consulting.net

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization and Its Applications 114, DOI 10.1007/978-3-319-41508-6_2

Denote the initial and final states by $\mathbf{y}(t_I) = \mathbf{y}_I$ and $\mathbf{y}(t_F) = \mathbf{y}_F$ respectively, with corresponding notation for other quantities. Although the problem can be stated in many equivalent formats, this *Lagrange* formulation is sufficiently general for our purpose.

2 Transcription Method

All numerical results presented here were obtained using the $\mathbb{SOS}$ (Sparse Optimization Suite) software. This implements a *direct transcription method* as described in [3]. The fundamental idea of a *transcription method* is to introduce a discrete approximation to the differential equations (2) in terms of the dynamic variables $\mathbf{y}(t)$, and $\mathbf{u}(t)$ evaluated at the discrete times

$$t_I = t_1 < t_2 < \cdots < t_M = t_F, \tag{5}$$

referred to as node, mesh, or grid points. In so doing the differential equation (2) is transcribed into a set algebraic constraints, defined in terms of a finite set of variables. Thus the optimal control problem is converted into a large nonlinear programming (NLP) problem. In principle any nonlinear programming method can be used to solve the discretized problem, but to do so the NLP must evaluate both first and second derivatives of the relevant discretization equations. These Jacobian and Hessian matrices are both large and sparse. To efficiently solve the NLP it is critical to exploit a computational benefit that accrues from the matrix sparsity itself. It is well known that the computational complexity for solving a system of n *dense* linear equations is $\mathcal{O}(n^3)$. In contrast, for a sparse linear system the cost is $\mathcal{O}(\kappa n)$, where κ is a factor related to sparsity. The $\mathbb{SOS}$ software has two nonlinear programming algorithms, a sparse Schur-complement sequential quadratic programming (SQP) algorithm, and a primal-dual interior point (barrier) algorithm. All numerical results presented use the SQP algorithm, and although we will not discuss details of the underlying sparse nonlinear programming algorithm the reader is referred to [3, Chaps. 1 and 2]. In fact since the computational expense of the entire algorithm is dominated by the quantity $\mathcal{O}(\kappa n)$ the remaining discussion will be focused on how to keep both κ and n as small as possible.

To summarize, the transcription method has three fundamental steps:

Direct Transcription: *Transcribe* the optimal control problem into a nonlinear programming (NLP) problem by discretization;

Sparse Nonlinear Program: Solve the sparse (SQP or Barrier) NLP

Mesh Refinement: Assess the accuracy of the approximation (i.e. the finite dimensional problem), and if necessary refine the discretization, and then repeat the optimization steps.

3 Runge-Kutta Methods

Let us begin by introducing methods for discretization of the differential equations (2). A popular family of one-step methods called *S-stage Runge-Kutta* [9] can be defined as follows:

$$\mathbf{y}_{k+1} = \mathbf{y}_k + h_k \sum_{j=1}^{S} \beta_j \mathbf{f}_{kj}, \tag{6}$$

where for $1 \leq j \leq S$

$$\mathbf{y}_{kj} = \mathbf{y}_k + h_k \sum_{\ell=1}^{S} \alpha_{j\ell} \mathbf{f}_{k\ell}, \tag{7}$$

$$\mathbf{f}_{kj} = \mathbf{f}\left[\mathbf{y}_{kj}, \mathbf{u}_{kj}, t_{kj}\right], \tag{8}$$

$$\mathbf{u}_{kj} = \mathbf{u}\left(t_{kj}\right), \tag{9}$$

$$t_{kj} = t_k + h_k \rho_j \tag{10}$$

$$h_k = t_{k+1} - t_k. \tag{11}$$

S is referred to as the "stage," and the intermediate values $\mathbf{y}_{kj}$ called *internal stages* and can be considered approximations to the solution at t_{kj}. In these expressions, $\{\rho_j, \beta_j, \alpha_{j\ell}\}$ are known constants with $0 \leq \rho_1 \leq \rho_2 \leq \cdots \leq \rho_S \leq 1$. A convenient way to define the coefficients is to use the Butcher array

$$\begin{array}{c|ccc} \rho_1 & \alpha_{11} & \ldots & \alpha_{1S} \\ \vdots & \vdots & & \vdots \\ \rho_S & \alpha_{S1} & \ldots & \alpha_{SS} \\ \hline & \beta_1 & \ldots & \beta_S \end{array} .$$

The schemes are called *explicit* if $\alpha_{j\ell} = 0$ for $l \geq j$ and *implicit* otherwise. The focus here is on a particular family of implicit Runge-Kutta (IRK) schemes called Lobatto IIIA methods. The Lobatto IIIA family has the following properties:

- The methods are symmetric with $\rho_1 = 0$ and $\rho_S = 1$.
- The coefficients $\alpha_{1j} = 0$ and $\alpha_{Sj} = \beta_j$ for $j = 1, \ldots, S$.
- As such the internal variables $\mathbf{y}_{j1}$ and $\mathbf{y}_{jS}$ for $j = 1, \ldots, S$ as well as the implicit constraints (7) can be analytically eliminated.
- The methods are collocation methods as described below. The variable and constraint definitions introduced are consistent with the collocation conditions given as Eqs. (5.71a) and (5.71b), in [1, p. 218].
- The method with S stages has (nonstiff) order $\eta = 2S - 2$.

The numerical values of the Lobatto IIIA coefficients for $S = 2, 3, 4, 5$ are given in the appendix.

Variable Phase Length

For many optimal control problems it is convenient to break the problem into *phases* either for numerical purposes or to describe different physical processes. In general the length of a phase is defined by t_I and t_F either of which can be an optimization variable. Therefore let us define a second time transformation

$$t = t_I + \tau(t_F - t_I) = t_I + \tau\sigma \tag{12}$$

where the phase length $\sigma \doteq t_F - t_I$ and $0 \leq \tau \leq 1$. Thus for $\Delta\tau_k = (\tau_{k+1} - \tau_k)$ we have

$$h_k = (\tau_{k+1} - \tau_k)(t_F - t_I) = \Delta\tau_k\sigma \tag{13}$$

In light of the transformation (12)

$$\mathbf{y}' = \frac{d\mathbf{y}}{d\tau} = \frac{d\mathbf{y}}{dt}\frac{dt}{d\tau} = \sigma\dot{\mathbf{y}} \tag{14}$$

and so the original ODE (2) becomes

$$\mathbf{y}' = \sigma\mathbf{f}[\mathbf{y}(\tau), \mathbf{u}(\tau), \tau] \tag{15}$$

Collocation Methods

The Runge-Kutta scheme (6)–(10) is often motivated in another way. Suppose we consider approximating the solution of the ODE (2) by a function $\mathbf{z}(t)$. In what follows it will be convenient denote component-wise operations using $z(t)$. As an approximation, let us use a polynomial of degree S (order $S + 1$) over each step $t_k \leq t \leq t_{k+1}$:

$$z(t) = a_0 + a_1(t - t_k) + \cdots + a_S(t - t_k)^S. \tag{16}$$

The coefficients $(a_0, a_1, \ldots, a_S)$ are chosen such that the approximation matches at the beginning of the step t_k, that is,

$$z(t_k) = y_k, \tag{17}$$

and has derivatives that match at the internal stage points (10)

$$\frac{dz(t_{kj})}{dt} = f\left[\mathbf{y}_{kj}, \mathbf{u}_{kj}, t_{kj}\right] = f_{kj}. \tag{18}$$

Observe that within a particular step $t_k \leq t \leq t_{k+1}$ the parameter $0 \leq \rho \leq 1$ defines the *local* time parameterization $t = t_k + h_k\rho$ and so from (16) it follows that

$$z(t) = a_0 + a_1 h_k \rho_j + \cdots + a_S h_k^S \rho_j^S \tag{19}$$

Similarly from (16)

$$\frac{dz(t)}{dt} = a_1 + \cdots + a_{S-1}(S-1)(t-t_k)^{S-2} + a_S S(t-t_k)^{S-1} \tag{20}$$

and so substituting (10), (18) gives

$$f_{kj} = a_1 + \cdots + a_{S-1}(S-1)h_k^{S-2}\rho_j^{S-2} + a_S S h_k^{S-1}\rho_j^{S-1} \tag{21}$$

Moreover, it is demonstrated in Ref. [1, p. 219] that when the derivatives match (cf. (18)) the function values $y_{kj} = z_{kj}$ also match for $1 \leq j \leq S$. Thus it follows from (19) and (10) that

$$y_{kj} = a_0 + a_1 h_k \rho_j + \cdots + a_S h_k^S \rho_j^S \tag{22}$$

The conditions (18) are called *collocation* conditions and the resulting method is referred to as a *collocation method*. The Runge-Kutta scheme (6)–(10) is a collocation method [1], and the solution produced by the method is a piecewise polynomial.

The focus of a collocation method is on a polynomial representation for the differential *state* variables. When the state is a polynomial of degree S over each step $t_k \leq t \leq t_{k+1}$ it is natural to use a polynomial approximation of degree $S-1$ for the algebraic variables $\mathbf{u}(t)$ similar to (16)

$$v(t) = b_0 + b_1(t-t_k) + \cdots + b_{S-1}(t-t_k)^{S-1} \tag{23}$$

for $j = 0, \ldots, S-1$ and the coefficients $(b_0, b_1, \ldots, b_{S-1})$ are determined such that the approximation matches at the intermediate points (10) for $j = 1, \ldots, S$

$$v(t_{kj}) = u_{kj}. \tag{24}$$

3.1 Lobatto IIIA, S = 2

The simplest Lobatto IIIA method has two stages and is of order $\eta = 2$. It is commonly referred to as the *trapezoidal* method (abbreviated TRP or LA2). The nonlinear programming constraints, called defects, and the corresponding NLP variables are as follows:

Defect Constraints

$$\mathbf{0} \equiv \boldsymbol{\zeta}_k = \mathbf{y}_{k+1} - \mathbf{y}_k - \frac{\Delta\tau_k}{2}\left[\sigma\mathbf{f}_k + \sigma\mathbf{f}_{k+1}\right] \tag{25a}$$

Variables

$$\mathbf{x}^{\mathsf{T}} = (\ldots, \mathbf{y}_k, \mathbf{u}_k, \mathbf{y}_{k+1}, \mathbf{u}_{k+1}, \ldots, \mathbf{p}, t_I, t_F, \ldots) \tag{25b}$$

3.2 Lobatto IIIA, S = 3

There are three common forms when there are three stages all having order $\eta = 4$. We abbreviate the primary form LA3.

Primary Form

Defect Constraints

$$\mathbf{0} = \mathbf{y}_{k+1} - \mathbf{y}_k - \Delta\tau_k\left[\beta_1\sigma\mathbf{f}_k + \beta_2\sigma\mathbf{f}_{k2} + \beta_3\sigma\mathbf{f}_{k+1}\right] \tag{26a}$$

$$\mathbf{0} = \mathbf{y}_{k2} - \mathbf{y}_k - \Delta\tau_k\left[\alpha_{21}\sigma\mathbf{f}_k + \alpha_{22}\sigma\mathbf{f}_{k2} + \alpha_{23}\sigma\mathbf{f}_{k+1}\right] \tag{26b}$$

where

$$\mathbf{f}_{k2} = \mathbf{f}\left[\mathbf{y}_{k2}, \mathbf{u}_{k2}, t_{k2}\right] \tag{26c}$$

$$t_{k2} = t_k + h_k\rho_2 = t_k + \frac{1}{2}h_k \tag{26d}$$

$$\mathbf{u}_{k2} = \mathbf{u}(t_{k2}) \tag{26e}$$

Variables

$$\mathbf{x}^{\mathsf{T}} = (\ldots, \mathbf{y}_k, \mathbf{u}_k, \mathbf{y}_{k2}, \mathbf{u}_{k2}, \mathbf{y}_{k+1}, \mathbf{u}_{k+1}, \ldots, \mathbf{p}, t_I, t_F, \ldots) \tag{26f}$$

Hermite-Simpson (Separated)

By solving (26a) for the quantity $\mathbf{f}_{k2}$ and then substituting the result into (26b) one obtains the second form. The method is referred to as *Hermite-Simpson (Separated)* or simply *Separated Simpson* and abbreviated HSS [3, Sect. 4.6.6].

Defect Constraints

$$\mathbf{0} = \mathbf{y}_{k+1} - \mathbf{y}_k - \Delta\tau_k \left[\beta_1 \sigma \mathbf{f}_k + \beta_2 \sigma \mathbf{f}_{k2} + \beta_3 \sigma \mathbf{f}_{k+1}\right] \tag{27a}$$

$$\mathbf{0} = \mathbf{y}_{k2} - \frac{1}{2}(\mathbf{y}_k + \mathbf{y}_{k+1}) - \frac{\Delta\tau_k}{8}(\sigma \mathbf{f}_k - \sigma \mathbf{f}_{k+1}) \tag{27b}$$

where the internal stage values and NLP variables are given by (26c)–(26f).

Hermite-Simpson (Compressed)

The third form is obtained by solving (27b) for the internal state $\mathbf{y}_{k2}$ and simply using this to evaluate $\mathbf{f}_{k2}$. This eliminates the explicit internal stage constraints (26b) and also the internal stage variables $\mathbf{y}_{k2}$. Referred to as *Hermite-Simpson (Compressed)* or simply *Compressed Simpson* it is abbreviated HSC [3, Sect. 4.6.5]. This form benefits from a smaller number of NLP variables and constraints, however at the expense of matrix sparsity.

Defect Constraints

$$\mathbf{0} = \mathbf{y}_{k+1} - \mathbf{y}_k - \Delta\tau_k \left[\beta_1 \sigma \mathbf{f}_k + \beta_2 \sigma \mathbf{f}_{k2} + \beta_3 \sigma \mathbf{f}_{k+1}\right] \tag{28a}$$

where

$$\mathbf{y}_{k2} = \frac{1}{2}(\mathbf{y}_k + \mathbf{y}_{k+1}) + \frac{h_k}{8}(\mathbf{f}_k - \mathbf{f}_{k+1}) \tag{28b}$$

$$\mathbf{f}_{k2} = \mathbf{f}\left[\mathbf{y}_{k2}, \mathbf{u}_{k2}, t_{k2}\right] \tag{28c}$$

$$t_{k2} = t_k + h_k \rho_2 = t_k + \frac{1}{2} h_k \tag{28d}$$

$$\mathbf{u}_{k2} = \mathbf{u}(t_{k2}) \tag{28e}$$

Variables

$$\mathbf{x}^{\mathsf{T}} = (\ldots, \mathbf{y}_k, \mathbf{u}_k, \mathbf{u}_{k2}, \mathbf{y}_{k+1}, \mathbf{u}_{k+1}, \ldots, \mathbf{p}, t_I, t_F, \ldots) \tag{28f}$$

3.3 Lobatto IIIA, S = 4

This sixth order scheme is abbreviated LA4.

Defect Constraints

$$\mathbf{0} = \mathbf{y}_{k+1} - \mathbf{y}_k - \Delta\tau_k \left[\beta_1 \sigma \mathbf{f}_k + \beta_2 \sigma \mathbf{f}_{k2} + \beta_3 \sigma \mathbf{f}_{k3} + \beta_4 \sigma \mathbf{f}_{k+1}\right] \tag{29a}$$

$$\mathbf{0} = \mathbf{y}_{k2} - \mathbf{y}_k - \Delta\tau_k \left[\alpha_{21} \sigma \mathbf{f}_k + \alpha_{22} \sigma \mathbf{f}_{k2} + \alpha_{23} \sigma \mathbf{f}_{k3} + \alpha_{24} \sigma \mathbf{f}_{k+1}\right] \tag{29b}$$

$$\mathbf{0} = \mathbf{y}_{k3} - \mathbf{y}_k - \Delta\tau_k \left[\alpha_{31} \sigma \mathbf{f}_k + \alpha_{32} \sigma \mathbf{f}_{k2} + \alpha_{33} \sigma \mathbf{f}_{k3} + \alpha_{34} \sigma \mathbf{f}_{k+1}\right] \tag{29c}$$

where

$$\mathbf{f}_{k2} = \mathbf{f}\left[\mathbf{y}_{k2}, \mathbf{u}_{k2}, t_{k2}\right] \tag{29d}$$

$$t_{k2} = t_k + h_k \rho_2 \tag{29e}$$

$$\mathbf{u}_{k2} = \mathbf{u}(t_{k2}) \tag{29f}$$

$$\mathbf{f}_{k3} = \mathbf{f}\left[\mathbf{y}_{k3}, \mathbf{u}_{k3}, t_{k3}\right] \tag{29g}$$

$$t_{k3} = t_k + h_k \rho_3 \tag{29h}$$

$$\mathbf{u}_{k3} = \mathbf{u}(t_{k3}) \tag{29i}$$

Variables

$$\mathbf{x}^{\mathsf{T}} = (\ldots, \mathbf{y}_k, \mathbf{u}_k, \mathbf{y}_{k2}, \mathbf{u}_{k2}, \mathbf{y}_{k3}, \mathbf{u}_{k3}, \mathbf{y}_{k+1}, \mathbf{u}_{k+1}, \ldots, \mathbf{p}, t_I, t_F, \ldots) \tag{29j}$$

3.4 Lobatto IIIA, S = 5

This eighth order scheme is abbreviated LA5.

Defect Constraints

$$\mathbf{0} = \mathbf{y}_{k+1} - \mathbf{y}_k - \Delta\tau_k \left[\beta_1 \sigma \mathbf{f}_k + \beta_2 \sigma \mathbf{f}_{k2} + \beta_3 \sigma \mathbf{f}_{k3} + \beta_4 \sigma \mathbf{f}_{k4} + \beta_5 \sigma \mathbf{f}_{k+1}\right] \tag{30a}$$

$$\mathbf{0} = \mathbf{y}_{k2} - \mathbf{y}_k - \Delta\tau_k \left[\alpha_{21} \sigma \mathbf{f}_k + \alpha_{22} \sigma \mathbf{f}_{k2} + \alpha_{23} \sigma \mathbf{f}_{k3} + \alpha_{24} \sigma \mathbf{f}_{k4} + \alpha_{25} \sigma \mathbf{f}_{k+1}\right] \tag{30b}$$

$$\mathbf{0} = \mathbf{y}_{k3} - \mathbf{y}_k - \Delta\tau_k \left[\alpha_{31} \sigma \mathbf{f}_k + \alpha_{32} \sigma \mathbf{f}_{k2} + \alpha_{33} \sigma \mathbf{f}_{k3} + \alpha_{34} \sigma \mathbf{f}_{k4} + \alpha_{35} \sigma \mathbf{f}_{k+1}\right] \tag{30c}$$

$$\mathbf{0} = \mathbf{y}_{k4} - \mathbf{y}_k - \Delta\tau_k \left[\alpha_{41} \sigma \mathbf{f}_k + \alpha_{42} \sigma \mathbf{f}_{k2} + \alpha_{43} \sigma \mathbf{f}_{k3} + \alpha_{44} \sigma \mathbf{f}_{k4} + \alpha_{45} \sigma \mathbf{f}_{k+1}\right] \tag{30d}$$

where

$$\mathbf{f}_{k2} = \mathbf{f}\left[\mathbf{y}_{k2}, \mathbf{u}_{k2}, t_{k2}\right] \tag{30e}$$

$$t_{k2} = t_k + h_k \rho_2 \tag{30f}$$

$$\mathbf{u}_{k2} = \mathbf{u}(t_{k2}) \tag{30g}$$

$$\mathbf{f}_{k3} = \mathbf{f}\left[\mathbf{y}_{k3}, \mathbf{u}_{k3}, t_{k3}\right] \tag{30h}$$

$$t_{k3} = t_k + h_k \rho_3 \tag{30i}$$

$$\mathbf{u}_{k3} = \mathbf{u}(t_{k3}) \tag{30j}$$

$$\mathbf{f}_{k4} = \mathbf{f}\left[\mathbf{y}_{k4}, \mathbf{u}_{k4}, t_{k4}\right] \tag{30k}$$

$$t_{k4} = t_k + h_k \rho_4 \tag{30l}$$

$$\mathbf{u}_{k4} = \mathbf{u}(t_{k4}) \tag{30m}$$

Variables

$$\mathbf{x}^{\mathsf{T}} = (\ldots, \mathbf{y}_k, \mathbf{u}_k, \mathbf{y}_{k2}, \mathbf{u}_{k2}, \mathbf{y}_{k3}, \mathbf{u}_{k3}, \mathbf{y}_{k4}, \mathbf{u}_{k4}, \mathbf{y}_{k+1}, \mathbf{u}_{k+1}, \ldots, \mathbf{p}, t_I, t_F, \ldots) \tag{30n}$$

Quadrature Equations
The IRK methods have been introduced as a way to solve ODE's. When treating problems involving integral expressions such as

$$\mathcal{I} = \int_{t_I}^{t_F} \mathbf{w}[\mathbf{y}(t), \mathbf{u}(t), t]dt \tag{31}$$

it is common to introduce new dynamic variables $\mathbf{r}(t)$ and then solve the following augmented system:

$$\dot{\mathbf{y}} = \mathbf{f}[\mathbf{y}(t), \mathbf{u}(t), t] \tag{32}$$

$$\dot{\mathbf{r}} = \mathbf{w}[\mathbf{y}(t), \mathbf{u}(t), t] \tag{33}$$

in conjunction with the initial conditions

$$\mathbf{r}(t_I) = 0 \tag{34}$$

and it then follows that

$$\mathbf{r}(t_F) = \mathcal{I}. \tag{35}$$

If we apply a recursive scheme to the augmented system we can write

$$\mathbf{r}(t_F) = \mathbf{r}_M = \sum_{k=1}^{M-1} \mathbf{r}_{k+1} - \mathbf{r}_k \tag{36}$$

for the subset of dynamic variables in (33). It then follows from (25a), (26a), (29a), and (30a) that

$$\mathbf{r}_{k+1} - \mathbf{r}_k = \begin{cases} \Delta\tau_k \left[\beta_1\sigma\mathbf{w}_k + \beta_2\sigma\mathbf{w}_{k+1}\right] & S = 2 \\ \Delta\tau_k \left[\beta_1\sigma\mathbf{w}_k + \beta_2\sigma\mathbf{w}_{k2} + \beta_3\sigma\mathbf{w}_{k+1}\right] & S = 3 \\ \Delta\tau_k \left[\beta_1\sigma\mathbf{w}_k + \beta_2\sigma\mathbf{w}_{k2} + \beta_3\sigma\mathbf{w}_{k3} + \beta_4\sigma\mathbf{w}_{k+1}\right] & S = 4 \\ \Delta\tau_k \left[\beta_1\sigma\mathbf{w}_k + \beta_2\sigma\mathbf{w}_{k2} + \beta_3\sigma\mathbf{w}_{k3} + \beta_4\sigma\mathbf{w}_{k4} + \beta_5\sigma\mathbf{w}_{k+1}\right] & S = 5 \end{cases} \tag{37}$$

Now it is important to note that the dynamic variables $\mathbf{r}(t)$ *do not appear* in the functions $\mathbf{w}$ and so there is no need to introduce the values at the grid points $\mathbf{r}_k$, and the internal stage points $\mathbf{r}_{k2}, \mathbf{r}_{k3}, \ldots$ when evaluating the integrands, i.e. $\mathbf{w}_k, \mathbf{w}_{k2}, \mathbf{w}_{k3}, , \mathbf{w}_{k4}$, etc.

4 Nonlinear Programming

The general nonlinear programming (NLP) problem can be stated as follows: Find the n-vector $\mathbf{x}^\mathsf{T} = (x_1, \ldots, x_n)$ to minimize the scalar objective function

$$F(\mathbf{x}) \tag{38}$$

subject to the m constraints

$$\mathbf{c}_L \le \mathbf{c}(\mathbf{x}) \le \mathbf{c}_U \tag{39}$$

and the simple bounds

$$\mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U. \tag{40}$$

Equality constraints can be imposed by setting $\mathbf{c}_L = \mathbf{c}_U$.

In the preceding section we outlined the quantities that must be treated as constraints and variables when an ODE is approximated by discretization. So for example, when using the trapezoidal method with M grid points, we must impose the defect constraints (25a) $k = 1, \ldots, M - 1$. In the trapezoidal case the goal is to choose the NLP variables (25b) to minimize the objective function and satisfy the defect constraints (25a) as well as any boundary conditions. To solve the nonlinear programming problem it is also necessary to compute the derivatives of the objective and constraint functions. When a finite difference method is used to construct the Jacobian, it is natural to identify the constraint functions as the quantities being differentiated. In other words, if we define

$$\mathbf{q} = \begin{bmatrix} \mathbf{c} \\ F \end{bmatrix} \tag{41}$$

then we can use finite differences to compute

$$\mathbf{D} = \frac{\partial \mathbf{q}}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{G} \\ \mathbf{g}^{\mathsf{T}} \end{bmatrix} \tag{42}$$

where $\mathbf{G}$ is the constraint Jacobian and $\mathbf{g}$ is the objective gradient. The Hessian of the Lagrangian $\mathbf{H}_L$ can also be constructed using differencing techniques as described in [3, Sect. 2.2].

However to exploit separability we write

$$\begin{bmatrix} \mathbf{c}(\mathbf{x}) \\ F(\mathbf{x}) \end{bmatrix} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{q}(\mathbf{x}). \tag{43}$$

By isolating the linear terms $\mathbf{A}\mathbf{x}$ from the nonlinear terms $\mathbf{B}\mathbf{q}(\mathbf{x})$, it is then easy to demonstrate that for all of the Lobatto methods the elements of the vector $\mathbf{q}(\mathbf{x})$ are of the form

$$\mathbf{q}(\mathbf{x}) = \begin{bmatrix} \vdots \\ \sigma\mathbf{f}_{k1} \\ \sigma\mathbf{w}_{k1} \\ \sigma\mathbf{f}_{k2} \\ \sigma\mathbf{w}_{k2} \\ \vdots \\ \sigma\mathbf{f}_{kS} \\ \sigma\mathbf{w}_{kS} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \sigma\mathbf{f}_{k} \\ \sigma\mathbf{w}_{k} \\ \sigma\mathbf{f}_{k2} \\ \sigma\mathbf{w}_{k2} \\ \vdots \\ \sigma\mathbf{f}_{k+1} \\ \sigma\mathbf{w}_{k+1} \\ \vdots \end{bmatrix} \tag{44}$$

The constant matrices **A** and **B** are defined by the method coefficients $\alpha_{j\ell}$, and β_j. It then follows that

$$\begin{bmatrix} \mathbf{G} \\ \mathbf{g}^{\mathsf{T}} \end{bmatrix} = \mathbf{A} + \mathbf{B}\mathbf{D} \tag{45}$$

where the finite difference matrix

$$\mathbf{D} = \frac{\partial \mathbf{q}}{\partial \mathbf{x}} \tag{46}$$

involves the right hand side quantities at the grid points and internal stage points. In particular these quantities are often sparse with a structure defined by the *sparsity template*

$$\mathcal{T} = \text{struct} \left[\begin{array}{c|c|c} \frac{\partial \mathbf{f}}{\partial \mathbf{y}} & \frac{\partial \mathbf{f}}{\partial \mathbf{u}} & \frac{\partial \mathbf{f}}{\partial \mathbf{p}} \\ \frac{\partial \mathbf{g}}{\partial \mathbf{y}} & \frac{\partial \mathbf{g}}{\partial \mathbf{u}} & \frac{\partial \mathbf{g}}{\partial \mathbf{p}} \\ \frac{\partial \mathbf{w}}{\partial \mathbf{y}} & \frac{\partial \mathbf{w}}{\partial \mathbf{u}} & \frac{\partial \mathbf{w}}{\partial \mathbf{p}} \end{array} \right] \tag{47}$$

and consequently **D** can be computed very efficiently using sparse finite difference techniques. In particular, when using sparse differencing the number of perturbations is dictated by the number of *index sets* γ, and for optimal control problems $\gamma \ll n$. For the low thrust problem below $\gamma = 6$ and this value does not change as the mesh size increases, even though the number of NLP variables n can become very large. In short we use the same number of perturbations whether the grid is coarse or fine!

5 Mesh Refinement

There are two primary goals for the final step in a direct transcription method, namely to

- decide whether the discrete solution is "accurate" enough, and
- if not do something about it.

To define "accuracy" in this context consider a single interval $t_k \le t \le t_k + h_k$. Suppose the NLP problem has produced a solution to the ODEs (2) so that,

$$\mathbf{y}(t_k + h_k) = \mathbf{y}(t_k) + \int_{t_k}^{t_k+h_k} \dot{\mathbf{y}} dt = \mathbf{y}(t_k) + \int_{t_k}^{t_k+h_k} \mathbf{f}\left[\mathbf{y}, \mathbf{u}, t\right] dt. \tag{48}$$

Unfortunately the expression for $\mathbf{y}(t_k + h_k)$ involves the true value for both $\mathbf{y}$ and $\mathbf{u}$ which are unknown. Consequently, we may consider the approximation

$$\widehat{\mathbf{y}}(t_k + h_k) \equiv \mathbf{y}(t_k) + \int_{t_k}^{t_k+h_k} \mathbf{f}\left[\mathbf{z}(t), \mathbf{v}(t), t\right] dt, \tag{49}$$

or the alternative expression

$$\widetilde{\mathbf{y}}(t_k + h_k) \equiv \mathbf{y}(t_k) + \int_{t_k}^{t_k+h_k} \mathbf{f}\left[\mathbf{y}(t), \mathbf{v}(t), t\right] dt, \tag{50}$$

Observe that the collocation solution $\mathbf{z}(t)$ and $\mathbf{v}(t)$ appears in the integrand of (49) whereas (50) involves the real solution state $\mathbf{y}(t)$ and the collocation control approximation $\mathbf{v}(t)$. In either case the "error" over the step is measured by the difference $\mathbf{y}(t_k + h_k) - \widehat{\mathbf{y}}(t_k + h_k)$ or $\mathbf{y}(t_k + h_k) - \widetilde{\mathbf{y}}(t_k + h_k)$. Motivated by this we define the *absolute local error* on a particular step by

$$\boldsymbol{\eta}_{i,k} = \left| \int_{t_k}^{t_{k+1}} \dot{\mathbf{z}}_i(s) - \mathbf{f}_i\left[\mathbf{z}(s), \mathbf{v}(s), s\right] ds \right| \tag{51}$$

Notice that the arguments of the integrand use the collocation approximations (16) and (23) for the state and control evaluated at intermediate points in the interval. From this expression for the absolute error, we can define the *relative local error* by

$$\epsilon_k \approx \max_i \frac{\boldsymbol{\eta}_{i,k}}{(w_i + 1)}, \tag{52}$$

where the scale weight $w_i = \max_{k=1}^{M} \left[|\tilde{y}_{i,k}|, |\dot{\tilde{y}}_{i,k}| \right]$ defines the maximum value for the ith state variable or its derivative over the M grid points in the phase. In the $\mathbb{SOS}$ software implementation ϵ_k is computed in every interval $t_k \le t \le t_k + h_k$ by evaluating $\boldsymbol{\eta}_{i,k}$ using a high precision Romberg quadrature method with Richardson extrapolation.

There are many complicated and often conflicting factors that determine the overall computational efficiency of the direct transcription algorithm. The cost of solving a sparse linear system $\mathbf{Ax} = \mathbf{b}$ is $\mathcal{O}(\kappa n)$, and typically the solution of a nonlinear program will require many iterations each requiring the solution of a sparse system. This suggests that the number of NLP variables n should be kept small. However, the cost also depends on the sparsity κ and consequently a small dense formulation may actually be more costly than a large sparse formulation. Furthermore, the number of NLP iterations is a function of the nonlinearity of the differential equations, as well as the quality of the initial guess. It is often more expensive to solve a small system of very nonlinear ODE's, when compared to a large system of linear ODE's. Similarly a "good" initial guess for the NLP variables may converge in one or two iterations regardless of the system nonlinearity. The situation is also unclear when comparing different discretization methods. For example, is it better to use two steps of a Lobatto two stage method, or a single step of a Lobatto three stage method? The number of NLP variables n is the same for both approaches. Since the two stage method is of order two and the three stage method is of order four, a naive analysis would suggest that the fourth order method is preferable. However, this conclusion ignores the impact of high order derivatives in a nonlinear differential equation. In fact, it is often better to utilize a low order scheme in regions with rapidly changing dynamics. Furthermore, when a "bad" initial guess is used to begin the NLP a low order method can be more robust, that is, converging to a solution in fewer iterations. The $\mathbb{SOS}$ software incorporates a number of options that permit the user to tailor the algorithm to the physical application. The basic mesh refinement procedure can be summarized as follows:

Mesh Refinement Algorithm

Estimate Discretization Error

Compute error ϵ_k for all intervals

Terminate if $\max_k \epsilon_k \leq \delta$

Change Discretization Method or Stepsize

Change the discretization method (if possible), otherwise

Reduce the stepsize(s) h_k by subdividing one or more intervals

The goal of the mesh refinement algorithm is to reduce the local error (52) below a user specified relative tolerance δ in all intervals. There are two mechanisms for achieving this goal, namely changing the discretization method, or reducing the stepsize h_k. In $\mathbb{SOS}$ the user can specify a sequence of methods for each phase in the problem description. For example suppose it is desirable to use a trapezoidal method for the first two refinement iterations, followed by the Compressed Simpson Method for the remaining iterations. This sequence is abbreviated as follows:

$$(\mathsf{TRP}), 2; (\mathsf{HSC}), 20.$$

As a second option if the sequence is

$$\mathsf{(LA2), -2; (LA3), -4; (LA5), -20}$$

the Lobatto IIIA (two-stage) discretization will be used if until $\epsilon_k > 10^{-2}$, followed by the Lobatto IIIA (three-stage) method if $\epsilon_k > 10^{-4}$, followed by the Lobatto IIIA (five-stage) method if $\epsilon_k > 10^{-20}$. It is worth recalling that the mesh refinement procedure occurs only after solving an NLP problem, and is terminated when the solution is sufficiently accurate. Thus for the second example, suppose the first refinement iteration using the LA2 discretization terminates with a discretization error of $\epsilon_k = 0.5 \times 10^{-4}$. The second and all subsequent refinement iterations will utilize an LA5 method until converging, provided of course the requested accuracy $\delta > 10^{-20}$.

If the discretization method is not changed on a particular refinement iteration, the error can be reduced by subdividing one or more intervals in the current mesh. In $\mathbb{SOS}$ there are three approaches. The default scheme described in [3, Sect. 4.7.4], solves an integer programming problem in order to minimize the maximum error over all intervals. A second approach attempts to distribute the error equally in all intervals using inverse interpolation of the existing error distribution. When solving delay-differential equations it is occasionally useful to use a simple bisection scheme which is also available.

6 Optimal Low Thrust Transfers Between Libration Point Orbits

6.1 Introduction

To illustrate the techniques discussed above let us consider an example which is noteworthy in two respects. First we address a problem of ongoing practical interest. Low thrust propulsion systems have been studied for many practical missions [4], and in addition there are a number of missions that are either currently operating in libration point orbits or proposed for the future. Secondly, this example represents the real manifestation of a class of *hypersensitive* optimal control problems. A series of investigations by Rao and Mease [10] demonstrate the solution is characterized by an initial and terminal boundary layer region, separated by a long duration equilibrium segment. The techniques they use are related to singular perturbation methods and Example 4.4 [3, pp. 170–171] illustrates the solution of a "toy problem" with this structure. The use of a high order Lobatto discretization was investigated by Herman and Conway [8], however, their approach did not exploit sparsity. Finally, because of periodic behavior and dynamic sensitivity there are many local solutions. In short this example is both numerically challenging and of great practical interest.

6.2 Dynamic Model

A formulation of an optimal low thrust transfer between libration point orbits is presented by Epenoy [7]. The dynamic model is based on the Planar Circular Restricted Three Body Problem (PCR3BP) with Earth as one primary and the Moon as the second. The equations of motion are constructed in a rotating reference frame, in which the x-axis extends from the barycenter of the Earth-Moon system to the Moon, and the y-axis completes the right hand coordinate frame. A set of non-dimensional units is chosen such that the unit of distance is the distance between the two primaries, the unit of mass is the sum of the primaries' masses, and the unit of time is such that the angular velocity of the primaries around their barycenter is one. Thus the Moon has mass μ and is fixed at the coordinates $(1-\mu, 0)$ while the Earth has mass $(1-\mu)$ and is fixed at the coordinates $(-\mu, 0)$. The mass parameter is defined as

$$\mu = \frac{M_m}{M_e + M_m} = 0.0121506683 \tag{53}$$

where M_e and M_m are the masses of the Earth and Moon respectively.

The equations of motion in the rotating frame are

$$\dot{x} = v_x \tag{54}$$

$$\dot{y} = v_y \tag{55}$$

$$\dot{v}_x = x + 2v_y - \frac{(1-\mu)(x+\mu)}{r_1^3} - \frac{\mu(x+\mu-1)}{r_2^3} + u_1 \tag{56}$$

$$\dot{v}_y = y - 2v_x - \frac{(1-\mu)y}{r_1^3} - \frac{\mu y}{r_2^3} + u_2 \tag{57}$$

where dot denotes the non-dimensional time derivative relative to an observer in the rotating frame. The position in the rotating frame is denoted by (x, y) with corresponding relative velocity (v_x, v_y). The distances from the Earth and Moon respectively are given by

$$r_1 = \sqrt{(x+\mu)^2 + y^2} \tag{58}$$

$$r_2 = \sqrt{(x+\mu-1)^2 + y^2} \tag{59}$$

The dynamics are defined by the state vector $\mathbf{z}^\mathsf{T} = (x, y, v_x, v_y)$ in the domain $t_0 \le t \le t_f$ where both the initial and final times are fixed. The control variables $\mathbf{u}^\mathsf{T} = (u_1, u_2)$ denote the spacecraft acceleration in the rotating frame. Thus the dynamics (54)–(57) are given by

$$\dot{\mathbf{z}} = \mathbf{f}[\mathbf{z}, \mathbf{u}]. \tag{60}$$

The initial and terminal states are fixed by the following boundary conditions:

$$\mathbf{z}(t_0) = \boldsymbol{\xi}_1(\tau_0) \tag{61}$$

$$\mathbf{z}(t_f) = \boldsymbol{\xi}_2(\tau_f) \tag{62}$$

where the states on the Lyapunov orbits are denoted by $\boldsymbol{\xi}_1(\tau_0)$ and $\boldsymbol{\xi}_2(\tau_f)$ respectively. The Lyapunov states are computed by means of Lindstedt-Poincare approximation as functions of the parameters τ_0 and τ_f. These non-dimensional times determine the departure and arrival location. The goal is to minimize the energy consumed during the transfer, i.e.

$$F = \frac{1}{2} \int_{t_0}^{t_f} \mathbf{u}^{\mathsf{T}} \mathbf{u} \, dt. \tag{63}$$

For the PCR3BP it is also convenient to introduce a time scaling. In particular we fix $t_0 = 0$ and define

$$t_F = 2\pi \frac{T_f}{P_M} \tag{64}$$

where T_f is the duration of the transfer and $P_M = 27.321577$ days is the orbital period of the moon. Numerical results will be constructed for two cases, namely a *short transfer* where $T_f = 12$ days, and a *long transfer* where $T_f = 44$ days. After time scaling $t_F = 2.759659$ for the short transfer and $t_F = 10.11874803$ for the long transfer.

6.3 Lyapunov Orbits

The libration points are defined in the rotating frame at $L_1 = (x_1, 0)$ and $L_2 = (x_2, 0)$ where x_1 and x_2 are roots of the nonlinear equations

$$0 = x_1 - \frac{1-\mu}{\mu + x_1} + \frac{\mu}{x_1 - 1 + \mu} \tag{65}$$

and

$$0 = x_2 - \frac{1-\mu}{\mu + x_2} - \frac{\mu}{x_2 - 1 + \mu} \tag{66}$$

respectively. For the particular case of interest with μ given by (53) we have $x_1 = 0.83691471889320190$ and $x_2 = 1.1556824834786137$.

Table 1 Lyapunov orbit around L_1

Location	τ_0	x	y
max x	0.000000000000000	0.8604642913942989	0.000000000000000
min y	0.6799402049577115	0.8460673759976699	−0.07126150424351556
min x	1.388012472385421	0.8203198878278488	0.000000000000000
max y	2.096084695912298	0.8460673739125611	0.07126150424351496
max x	2.776024944790721	0.8604642913942989	0.000000000000000

Table 2 Lyapunov orbit around L_2

Location	τ_0	x	y
max x	0.000000000000000	1.170863515501900	0.000000000000000
min y	0.8632085886843588	1.150862903956706	−0.04864318483502234
min x	1.692646170500000	1.137392572452755	0.000000000000000
max y	2.522083753145239	1.150862903981730	0.04864318483502232
max x	3.385292341000000	1.170863515501900	0.000000000000000

The specific Lyapunov orbits given in Epenoy [7] correspond to orbits around the Earth-Moon libration points L_1 and L_2 with the same value for the Jacobi constant, namely 3.178. FORTRAN code implementing the calculation of the functions $\boldsymbol{\xi}_1(\tau_0)$ and $\boldsymbol{\xi}_2(\tau_f)$ was graciously supplied by R. Epenoy. Tables 1 and 2 present a summary of the extreme points in these orbits. Note also that the functions $\boldsymbol{\xi}_1(\tau_0)$ and $\boldsymbol{\xi}_2(\tau_f)$ are periodic functions of the parameters τ_0 and τ_f respectively. Thus $\boldsymbol{\xi}_1(\tau_0) = \boldsymbol{\xi}_1(\tau_0 + kT_1)$ where $T_1 = 2.776024944790721$ for $k = 0, \pm 1, \pm 2, \ldots$. Similarly for the orbit around L_2 we have $\boldsymbol{\xi}_2(\tau_f) = \boldsymbol{\xi}_2(\tau_f + kT_2)$ where $T_2 = 3.385292341000000$.

6.4 Adjoint Equations

For the sake of reference in what follows it is useful to define the adjoint equations. The Hamiltonian is given by

$$H = F + \boldsymbol{\lambda}^{\mathsf{T}}\mathbf{f} = \frac{1}{2}\left[u_1^2 + u_2^2\right] + \lambda_1 f_1 + \lambda_2 f_2 + \lambda_3 f_3 + \lambda_4 f_4 \tag{67}$$

and the adjoint equations are:

$$\dot{\lambda}_1 = -\frac{\partial H}{\partial x} = -\lambda_3 \frac{\partial f_3}{\partial x} - \lambda_4 \frac{\partial f_4}{\partial x} = -\lambda_3 \frac{\partial f_3}{\partial x} - \lambda_4 \frac{\partial f_4}{\partial x} \tag{68}$$

$$\dot{\lambda}_2 = -\frac{\partial H}{\partial y} = -\lambda_3 \frac{\partial f_3}{\partial y} - \lambda_4 \frac{\partial f_4}{\partial y} = -\lambda_3 \frac{\partial f_3}{\partial y} - \lambda_4 \frac{\partial f_4}{\partial y} \tag{69}$$

$$\dot{\lambda}_3 = -\frac{\partial H}{\partial v_x} = -\lambda_1 \frac{\partial f_1}{\partial v_x} - \lambda_4 \frac{\partial f_4}{\partial v_x} = -\lambda_1 \{1\} - \lambda_4 \{-2\} = -\lambda_1 + 2\lambda_4 \tag{70}$$

$$\dot{\lambda}_4 = -\frac{\partial H}{\partial v_y} = -\lambda_2 \frac{\partial f_2}{\partial v_y} - \lambda_3 \frac{\partial f_3}{\partial v_y} = -\lambda_2 \{1\} - \lambda_3 \{2\} = -\lambda_2 - 2\lambda_3 \tag{71}$$

To evaluate these expressions first define:

$$\frac{\partial}{\partial x}\left\{r_1^{-3}\right\} = -3r_1^{-4}\frac{\partial r_1}{\partial x} = -\frac{3(x+\mu)}{r_1^5} \tag{72}$$

$$\frac{\partial}{\partial y}\left\{r_1^{-3}\right\} = -3r_1^{-4}\frac{\partial r_1}{\partial y} = -\frac{3y}{r_1^5} \tag{73}$$

$$\frac{\partial}{\partial x}\left\{r_2^{-3}\right\} = -3r_2^{-4}\frac{\partial r_2}{\partial x} = -\frac{3(x+\mu-1)}{r_2^5} \tag{74}$$

$$\frac{\partial}{\partial y}\left\{r_2^{-3}\right\} = -3r_2^{-4}\frac{\partial r_2}{\partial y} = -\frac{3y}{r_2^5} \tag{75}$$

where

$$\frac{\partial r_1}{\partial x} = \frac{1}{2r_1}\left[2(x+\mu)\right] = \frac{(x+\mu)}{r_1} \tag{76}$$

$$\frac{\partial r_2}{\partial x} = \frac{1}{2r_2}\left[2(x+\mu-1)\right] = \frac{(x+\mu-1)}{r_2} \tag{77}$$

$$\frac{\partial r_1}{\partial y} = \frac{1}{2r_1}\left[2y\right] = \frac{y}{r_1} \tag{78}$$

$$\frac{\partial r_2}{\partial y} = \frac{1}{2r_2}\left[2y\right] = \frac{y}{r_2} \tag{79}$$

The derivatives needed to define the right hand sides of the adjoint equations are then given by

$$\frac{\partial f_3}{\partial x} = 1 - \frac{\partial}{\partial x}\left\{\frac{(1-\mu)(x+\mu)}{r_1^3}\right\} - \frac{\partial}{\partial x}\left\{\frac{\mu(x+\mu-1)}{r_2^3}\right\} = 1 - d_1 - d_2 \tag{80}$$

$$\frac{\partial f_4}{\partial x} = -\frac{\partial}{\partial x}\left\{\frac{(1-\mu)y}{r_1^3}\right\} - \frac{\partial}{\partial x}\left\{\frac{\mu y}{r_2^3}\right\} = -d_3 - d_4 \tag{81}$$

$$\frac{\partial f_3}{\partial y} = -\frac{\partial}{\partial y}\left\{\frac{(1-\mu)(x+\mu)}{r_1^3}\right\} - \frac{\partial}{\partial y}\left\{\frac{\mu(x+\mu-1)}{r_2^3}\right\} = -d_5 - d_6 \tag{82}$$

$$\frac{\partial f_4}{\partial y} = 1 - \frac{\partial}{\partial y}\left\{\frac{(1-\mu)y}{r_1^3}\right\} - \frac{\partial}{\partial y}\left\{\frac{\mu y}{r_2^3}\right\} = 1 - d_7 - d_8 \tag{83}$$

where the intermediate terms are defined as follows:

$$d_1 \doteq \frac{\partial}{\partial x}\left\{\frac{(1-\mu)(x+\mu)}{r_1^3}\right\} = (1-\mu)(x+\mu)\frac{\partial}{\partial x}\left\{r_1^{-3}\right\} + (1-\mu)r_1^{-3} \tag{84}$$

$$d_2 \doteq \frac{\partial}{\partial x}\left\{\frac{\mu(x+\mu-1)}{r_2^3}\right\} = \mu(x+\mu-1)\frac{\partial}{\partial x}\left\{r_2^{-3}\right\} + \mu r_2^{-3} \tag{85}$$

$$d_3 \doteq \frac{\partial}{\partial x}\left\{\frac{(1-\mu)y}{r_1^3}\right\} = (1-\mu)y\frac{\partial}{\partial x}\left\{r_1^{-3}\right\} \tag{86}$$

$$d_4 \doteq \frac{\partial}{\partial x}\left\{\frac{\mu y}{r_2^3}\right\} = \mu y\frac{\partial}{\partial x}\left\{r_2^{-3}\right\} \tag{87}$$

$$d_5 \doteq \frac{\partial}{\partial y}\left\{\frac{(1-\mu)(x+\mu)}{r_1^3}\right\} = (1-\mu)(x+\mu)\frac{\partial}{\partial y}\left\{r_1^{-3}\right\} \tag{88}$$

$$d_6 \doteq \frac{\partial}{\partial y}\left\{\frac{\mu(x+\mu-1)}{r_2^3}\right\} = \mu(x+\mu-1)\frac{\partial}{\partial y}\left\{r_2^{-3}\right\} \tag{89}$$

$$d_7 \doteq \frac{\partial}{\partial y}\left\{\frac{(1-\mu)y}{r_1^3}\right\} = (1-\mu)y\frac{\partial}{\partial y}\left\{r_1^{-3}\right\} + (1-\mu)r_1^{-3} \tag{90}$$

$$d_8 \doteq \frac{\partial}{\partial y}\left\{\frac{\mu y}{r_2^3}\right\} = \mu y\frac{\partial}{\partial y}\left\{r_2^{-3}\right\} + \mu r_2^{-3} \tag{91}$$

The optimal controls are defined by the optimality conditions

$$\frac{\partial H}{\partial u_1} = 0 = u_1 + \lambda_3 \tag{92}$$

$$\frac{\partial H}{\partial u_2} = 0 = u_2 + \lambda_4 \tag{93}$$

7 Numerical Results

7.1 Short Transfer

When modeling the dynamic behavior of the short transfer it is convenient to break the problem into separate regions. The short transfer was modeled using two distinct *phases* as defined in Table 3.

Table 3 Short transfer phase structure

Phase	Description	Domain	Free parameters
1	L_1 departure	$0 \leq t \leq t_1$	$t_1^{(-)}$, τ_0
2	L_2 arrival	$t_1 \leq t \leq t_f$	$t_1^{(+)}$, t_f, τ_f

The phase structure permits modeling two distinct portions of the overall trajectory, namely departure from the L_1 Lyapunov orbit, and arrival at the L_2 Lyapunov orbit. The initial state given by the boundary condition (61) fixes the beginning of phase 1. It is convenient to terminate phase 1 when the trajectory passes below the moon in a posigrade direction, and so we require

$$x(t_1) = 1 - \mu \tag{94}$$

$$y(t_1) \leq y_{min} \tag{95}$$

$$v_x(t_1) \geq 0 \tag{96}$$

where y_{min} is a bound on the closest approach to the moon. For our results we choose $y_{min} = -0.04$. To ensure continuity from phase to phase we impose the continuity constraints

$$t_1^{(-)} = t_1^{(+)} \tag{97}$$

$$\mathbf{z}[t_1^{(-)}] = \mathbf{z}[t_1^{(+)}] \tag{98}$$

where conditions at the end of phase one are denoted by $t_1^{(-)}$ and the corresponding conditions at the beginning of phase two are denoted by $t_1^{(+)}$. At the end of phase two we must also satisfy

$$t_f \leq t_F \tag{99}$$

$$\mathbf{z}(t_f) = \boldsymbol{\xi}_2(\tau_f) \tag{100}$$

where $t_F = 2.759659$. Note that in (99) the final time t_f is treated as a free parameter limited by the fixed upper bound t_F. Treating the final time as free, yields more stable intermediate iterates and the final optimal trajectory is achieved much more readily. In particular it is expected that the inequality constraint (99) will be *active* at the solution. It is also worth noting that formulating the problem using two phases is done strictly for numerical purposes. In particular the constraints (94)–(96) prevent intermediate trajectories that are unreasonable, and thus improve the robustness of the algorithm.

The direct transcription method requires an initial guess for all free parameters as well as the dynamic history for the state and control. For the short transfer we guess

$$\mathbf{p}^{\mathsf{T}} = (t_1, t_f, \tau_0, \tau_f) = (t_F/2, t_F, 2.096084695912298, 2.522083753145239)$$

where the choice of τ_0 and τ_f correspond to the maximum values of y as given in Tables 1 and 2. For quantities that change dynamically during phase one a guess that linearly interpolates between the boundary values is given by

$$[t_k, x(t_k), y(t_k), v_x(t_k), v_y(t_k), u_1(t_k), u_2(t_k)] = \alpha_k \mathbf{a}^{\mathsf{T}} + \beta_k \mathbf{b}^{\mathsf{T}} \quad k = 1, \ldots, M \tag{101}$$

Table 4 Mesh refinement summary–short transfer

k	M	Disc.	m	n	n_d	NRHS	Iter	ϵ	Time (s)
1	(10,10)	(LA2,LA2)	87	125	38	9119	28	6.4×10^{-3}	0.6179
2	(10,10)	(LA3,LA3)	159	233	74	58,474	69	3.1×10^{-3}	1.289
3	(19,19)	(LA3,LA3)	303	449	146	30,332	16	8.6×10^{-5}	0.5749
4	(19,19)	(LA4,LA4)	447	665	218	36,954	10	2.7×10^{-6}	0.7068
5	(36,36)	(LA4,LA4)	855	1277	422	32,430	3	4.9×10^{-8}	0.4019
Total	72					167,309			3.591

for M grid points on the phase where

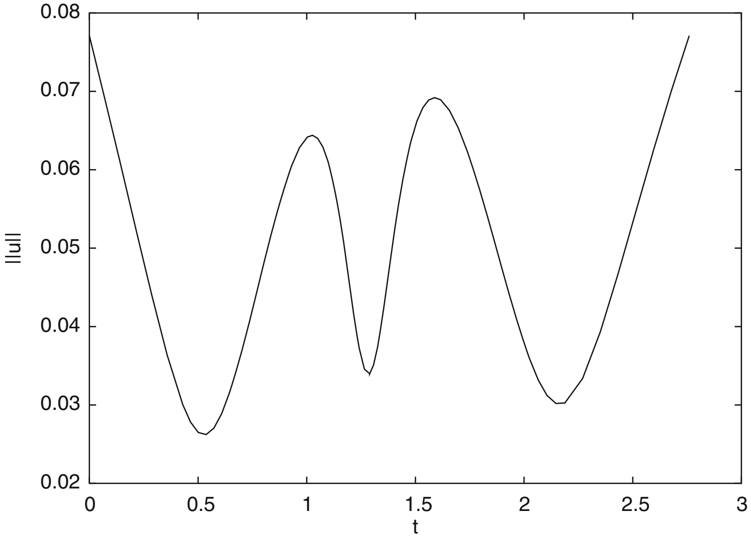
$$\beta_k = \frac{k-1}{M-1} \qquad \alpha_k = 1 - \beta_k \tag{102}$$

$$\mathbf{a}^{\mathsf{T}} = [0, \boldsymbol{\xi}^{\mathsf{T}}{}_1(\tau_o), 0, 0] \qquad \mathbf{b}^{\mathsf{T}} = [t_F/2, 1-\mu, -R, v_x(0), 0, 0, 0]. \tag{103}$$

We also make a guess for the radial distance from the moon as $R = 0.1$ in (103). On the second phase we again use (101) but replace the boundary quantities (103) with

$$\mathbf{a}^{\mathsf{T}} = [t_F/2, 1-\mu, -R, v_x(t_f), 0, 0, 0] \qquad \mathbf{b}^{\mathsf{T}} = [t_F, \boldsymbol{\xi}_2^{\mathsf{T}}(\tau_f), 0, 0]. \tag{104}$$

Using this information the solution was computed using $\mathbb{SOS}$ and Table 4 presents a summary of the algorithm behavior when using "(LA2),-2;(LA3),-3;(LA4),-20" as a mesh refinement strategy. The first refinement iteration of the algorithm began with $M = 10$ equi-distributed grid points in each phase and the linear initial guess given by (101). Using an LA2 (trapezoidal) discretization produced an NLP with $m = 87$ constraints, $n = 125$ variables, and $n_d = 38$ degrees of freedom. The solution required 28 NLP iterations, as well as (NRHS = 9119) evaluations of the right hand side of the ODE. This NLP was solved in 0.6179 CPU seconds, and resulted in a discretization error of $\epsilon = 6.4 \times 10^{-3}$. A second mesh refinement iteration using 10 grid points in each phase with the LA3 discretization reduced the relative error to $\epsilon = 3.1 \times 10^{-3}$. The LA3 method was used again on the third refinement iteration, but with 19 grid points in each phase, distributed by the default minimax scheme. Subsequently, two additional refinement iterations were executed using the higher order LA4 (four stage Lobatto IIIA) discretization in order to reduce the error below the requested tolerance of $\epsilon = 1 \times 10^{-7}$ which corresponds to approximately eight significant figures in the solution variables. The total CPU time on a desktop computer using an Intel I7 processor (3.06 Ghz), with Linux operating system, and GNU Fortran compiler, was 3.591 s. A summary of the optimal solution values for this case is given in Fig. 1, and it is worth noting that $t_1^* \neq t_F/2$ however, $t_f^* = t_F$. Figures 2, 3, 4, 5, 6, 7 and 8 illustrate the optimal solution history.



$$\boxed{\begin{array}{ll} \multicolumn{2}{l}{F^* = \dfrac{1}{2}\displaystyle\int_{t_0}^{t_f} \|\mathbf{u}\|^2 dt = 3.6513908 \times 10^{-3}} \\ t_1^* = 1.2892611 & t_f^* = 2.7596586 \\ \tau_0^* = 1.6548720 & \tau_f^* = 3.0315362 \end{array}}$$

Fig. 1 Optimal solution–short transfer $\|\mathbf{u}(t)\|$

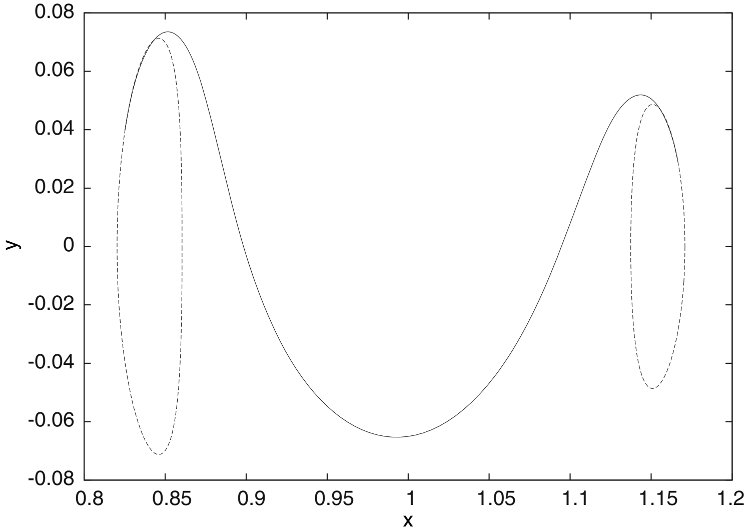


Fig. 2 Short transfer trajectory

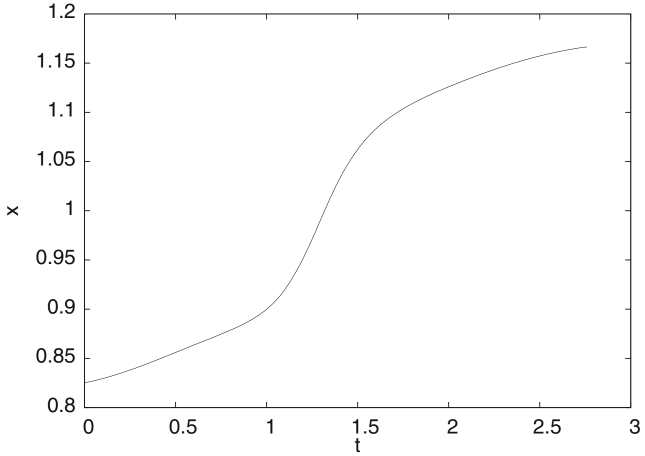


Fig. 3 Short transfer $x(t)$

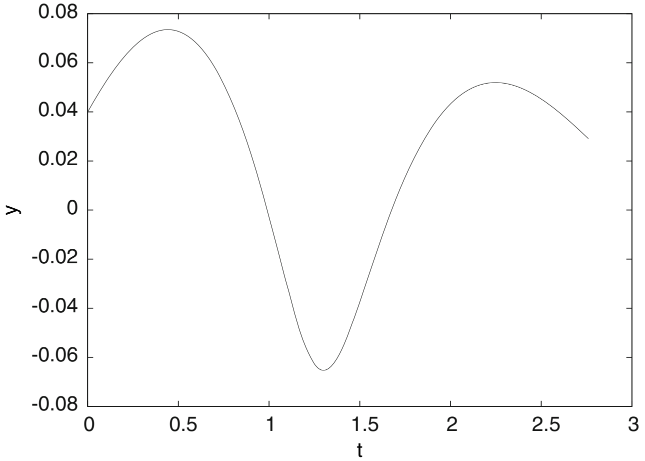


Fig. 4 Short transfer $y(t)$

7.2 Long Transfer

In contrast to the short transfer, the long transfer is modeled using four distinct *phases* as defined in Table 5. As before, the first phase models the departure from the L_1 orbit, and the last phase models the arrival at the L_2 orbit. However, two additional intermediate phases are introduced between the first and last phase to accommodate two revolutions about the moon.

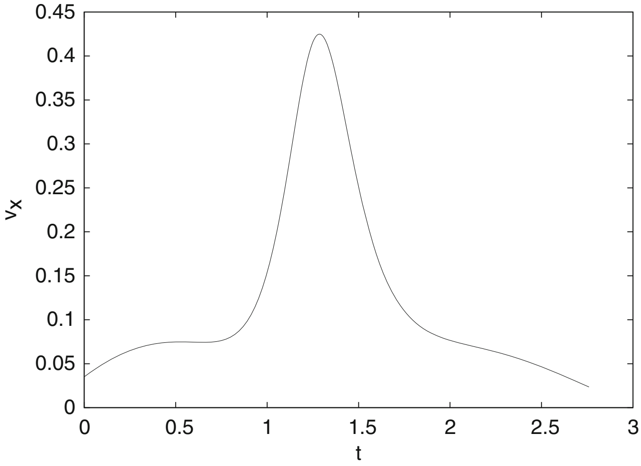


Fig. 5 Short transfer $v_x(t)$

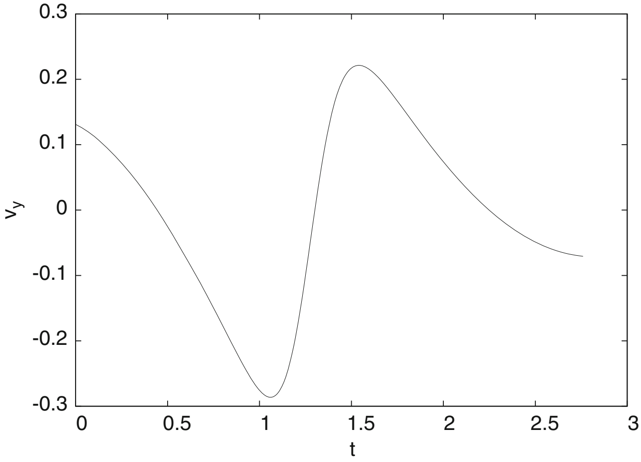


Fig. 6 Short transfer $v_y(t)$

As before, the initial state given by the boundary condition (61) fixes the beginning of phase 1. However, since we are modeling two revolutions about the moon, we terminate phases one, two, and three when the trajectory passes below the moon in a posigrade direction, and so for $j = 1, 2, 3$ we require

$$x(t_j) = 1 - \mu \tag{105}$$

$$y(t_j) \leq y_{min} \tag{106}$$

$$v_x(t_j) \geq 0. \tag{107}$$

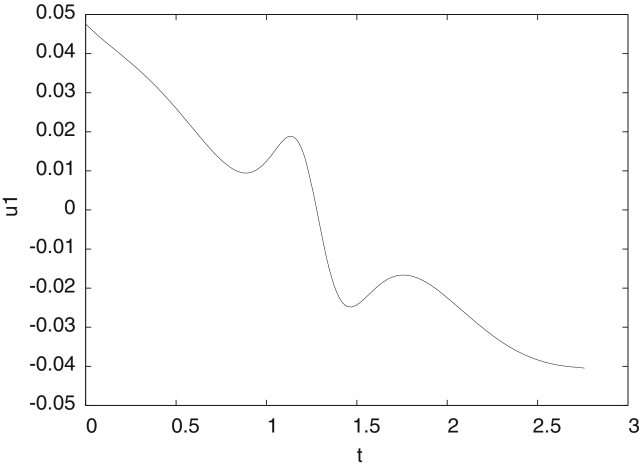


Fig. 7 Short transfer $u_1(t)$

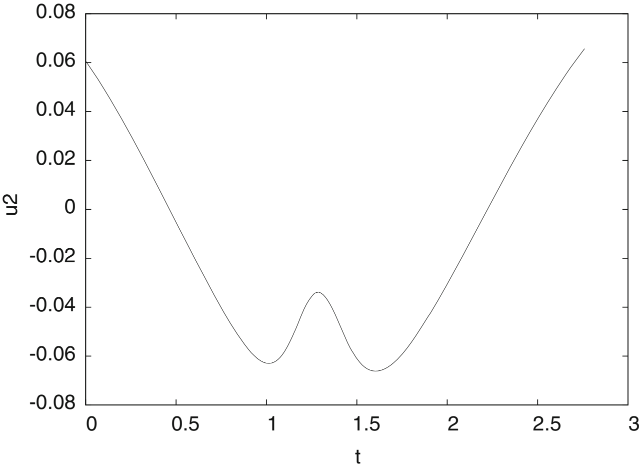


Fig. 8 Short transfer $u_2(t)$

Table 5 Long transfer phase structure

Phase	Description	Domain	Free parameters
1	L_1 departure	$0 \leq t \leq t_1$	$t_1^{(-)}$, τ_0
2	Lunar revolution 1	$t_1 \leq t \leq t_2$	$t_1^{(+)}$, $t_2^{(-)}$
3	Lunar revolution 2	$t_2 \leq t \leq t_3$	$t_2^{(+)}$, $t_3^{(-)}$
4	L_2 arrival	$t_3 \leq t \leq t_f$	$t_3^{(+)}$, t_f, τ_f

Continuity from phase to phase is insured by imposing the continuity constraints

$$t_j^{(-)} = t_j^{(+)} \tag{108}$$

$$\mathbf{z}[t_j^{(-)}] = \mathbf{z}[t_j^{(+)}] \tag{109}$$

for all three phases. At the end of phase four we must also satisfy

$$t_f \leq t_F \tag{110}$$

$$\mathbf{z}(t_f) = \boldsymbol{\xi}_2(\tau_f) \tag{111}$$

where $t_F = 10.11874803$. As before, formulating the problem using four phases is done strictly for numerical purposes. In particular the constraints (105)–(107) prevent intermediate trajectories that are unreasonable, and thus improve the robustness of the algorithm.

Since the long transfer formulation has more phases, we must supply an initial guess for all of the free parameters. Thus we guess

$$\begin{aligned}\mathbf{p}^{\mathsf{T}} &= (t_1, t_2, t_3, t_f, \tau_0, \tau_f) \\ &= (0.1t_F, 0.5t_F, 0.9t_F, t_F, 2.096084695912298, 2.522083753145239)\end{aligned} \tag{112}$$

where the choice of τ_0 and τ_f correspond to the maximum values of y as given in Tables 1 and 2. For quantities that change dynamically during phase one a guess that linearly interpolates between the boundary values is given by (101) as defined by the boundary values

$$\mathbf{a}^{\mathsf{T}} = [0, \boldsymbol{\xi}_1^{\mathsf{T}}(\tau_o), 0, 0] \quad \mathbf{b}^{\mathsf{T}} = [0.1t_F, 1-\mu, -R, v_x(0), 0, 0, 0]. \tag{113}$$

On the last phase we again use (101) but replace the boundary quantities (113) with

$$\mathbf{a}^{\mathsf{T}} = [0.9t_F, 1-\mu, -R, v_x(t_f), 0, 0, 0] \quad \mathbf{b}^{\mathsf{T}} = [t_F, \boldsymbol{\xi}_2^{\mathsf{T}}(\tau_f), 0, 0]. \tag{114}$$

While a simple linear guess of the dynamic history is acceptable on the first and last phase, for the intermediate phases it is more reasonable to supply a trajectory that circles the moon. Thus we supply a simple circle of radius R as a guess for the second and third phase dynamic history. To be more specific on phase j the value of a dynamic variable at grid point k is given by

$$\Delta t = p_j - p_{j-1} \tag{115}$$

$$V = \frac{2\pi R}{\Delta t} \tag{116}$$

$$t_k = p_{j-1} + \Delta t \left(\frac{k-1}{M-1} \right) \tag{117}$$

$$\theta_k = -\frac{\pi}{2} + 2\pi\left(\frac{k-1}{M-1}\right) \tag{118}$$

$$x(t_k) = (1-\mu) + R\cos\theta_k \tag{119}$$

$$y(t_k) = R\sin\theta_k \tag{120}$$

$$v_x(t_k) = -V\sin\theta_k \tag{121}$$

$$v_y(t_k) = +V\cos\theta_k \tag{122}$$

$$u_1(t_k) = 0 \tag{123}$$

$$u_2(t_k) = 0 \tag{124}$$

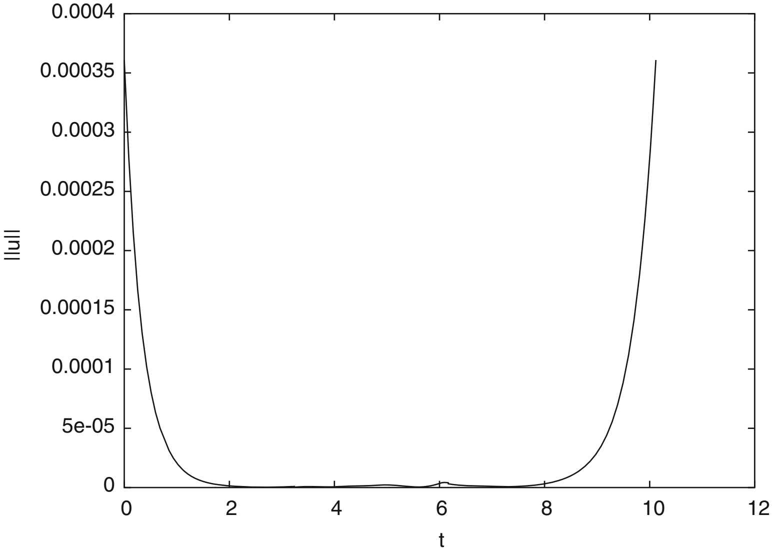
where $j = 2, 3$ and the number of grid points $k = 1, \ldots, M$. Note that for both phase 2 and phase 3, the value of Δt is the same, since $\Delta t = p_2 - p_1 = 0.5t_F - 0.1t_F = 0.4t_F = p_3 - p_2$. As before the radius guess is $R = 0.1$, and the number of grid points on all phases for the first refinement iteration is $M = 20$. Using this information the solution was computed using $\mathbb{SOS}$ and Table 6 presents a summary of the algorithm behavior. For the first and last phase we use " (LA2),-1;(LA3),-3;(LA4),-20", and for phases two and three we use "(LA3),-3;(LA4),-4;(LA5),-20". Using an LA2 discretization in the first and last phase, and an LA3 discretization in the second and third phase, produced an NLP with $m = 482$ constraints, $n = 717$ variables, and $n_d = 235$ degrees of freedom. This NLP was solved in 2.609 CPU seconds, and resulted in a discretization error of $\epsilon = 7.9 \times 10^{-3}$. A second mesh refinement iteration with an LA4 discretization in all phases and a grid point distribution of $(20, 20, 20, 20)$ for phases 1 through 4 respectively reduced the relative error to $\epsilon = 5.7 \times 10^{-4}$. Subsequently, three additional refinement iterations were executed using the sixth order LA4 method in the first and last phase, and the eighth order LA5 method for phases two and three to reduce the error below the requested tolerance of $\epsilon = 1 \times 10^{-7}$ which corresponds to approximately eight significant figures in the solution variables. The total CPU time, was 18.112 s. A summary of the optimal solution values for this case is given in Fig. 9. Figures 10, 11, 12, 13, 14, 15 and 16 illustrate the optimal solution history.

8 Computational Comparisons

When solving a complicated problem such as the low-thrust transfer between libration points, there are often alternatives that can be utilized in an attempt to improve robustness and/or efficiency. This section presents a few possible alternatives for this example.

Table 6 Mesh refinement summary–long transfer

k	M	Disc.	m	n	n_d	NRHS	Iter	ϵ	Time (s)
1	(20,20,20,20)	(LA2,LA3,LA3,LA2)	482	717	235	246,033	144	$7.9{\times}10^{-3}$	2.609
2	(20,20,20,20)	(LA4,LA4,LA4,LA4)	938	1401	463	491,828	76	$5.7{\times}10^{-4}$	5.941
3	(29,20,20,25)	(LA4,LA5,LA5,LA4)	1258	1881	623	127,756	12	$2.0{\times}10^{-5}$	2.107
4	(49,37,32,35)	(LA4,LA5,LA5,LA4)	2082	3117	1034	193,452	13	$1.4{\times}10^{-6}$	5.972
5	(54,52,42,62)	(LA4,LA5,LA5,LA4)	2867	4293	1426	88,870	2	$9.1{\times}10^{-8}$	1.481
Total	210					1,147,939			18.112



$F^* = \frac{1}{2}\int_{t_0}^{t_f} \|\mathbf{u}\|^2 dt = 2.54378004 \times 10^{-8}$	
$t_1^* = 3.2370798$	$t_2^* = 4.6847498$
$t_3^* = 6.1674429$	$t_f^* = 10.118748$
$\tau_0^* = -4.4748330 \times 10^{-3}$	$\tau_f^* = 5.2013294$

Fig. 9 Optimal solution–long transfer $\|\mathbf{u}(t)\|$

8.1 Spline Approximation

In order to evaluate the boundary condition (61) and (62) the results in the previous section utilized a FORTRAN software implementation of a Lindstedt-Poincare approximation. Since these quantities are each functions of the a single parameter it is reasonable to consider the following B-spline approximations:

$$\boldsymbol{\xi}_1(\tau_0) \approx \sum_k \boldsymbol{\alpha}_k B_k(\tau_0) \tag{125}$$

$$\boldsymbol{\xi}_2(\tau_f) \approx \sum_k \boldsymbol{\alpha}_k B_k(\tau_f) \tag{126}$$

The coefficients that define the natural cubic B-spline can be computed by evaluating the Lindstedt-Poincare approximation over the parameter range and then interpolating these values. Evaluation of this approximation can then be utilized during the optimization process. Table 7 summarizes the reduction in overall CPU time when B-spline approximations are used to evaluate the boundary conditions.

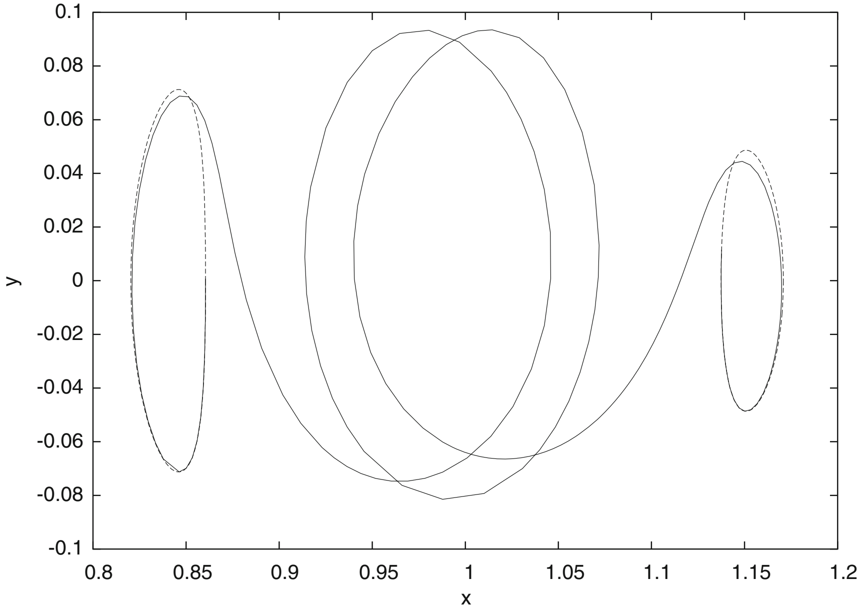


Fig. 10 Long transfer trajectory

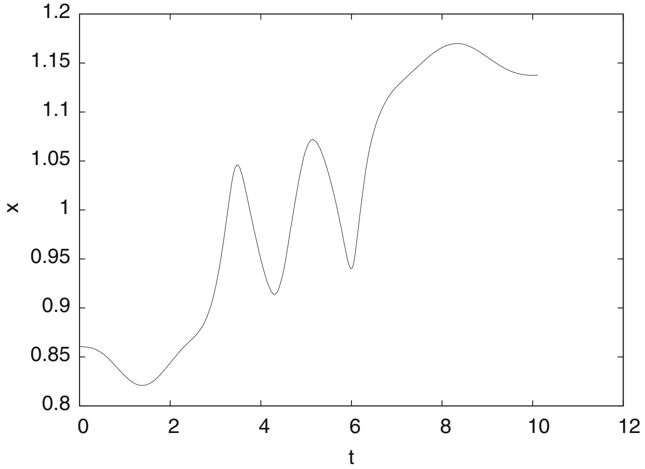


Fig. 11 Long transfer $x(t)$

The time spent by the $\mathbb{SOS}$ algorithm is given in the first column, whereas the time spent computing the problem functions, i.e. the ODE right hand sides $\mathbf{f}[\mathbf{z}, \mathbf{u}]$ and the boundary conditions $\boldsymbol{\xi}_1(\tau_0)$ and $\boldsymbol{\xi}_2(\tau_f)$, is summarized in the second column. Clearly there is a significant reduction in the overall solution time for both the short and long transfers when using a spline approximation.

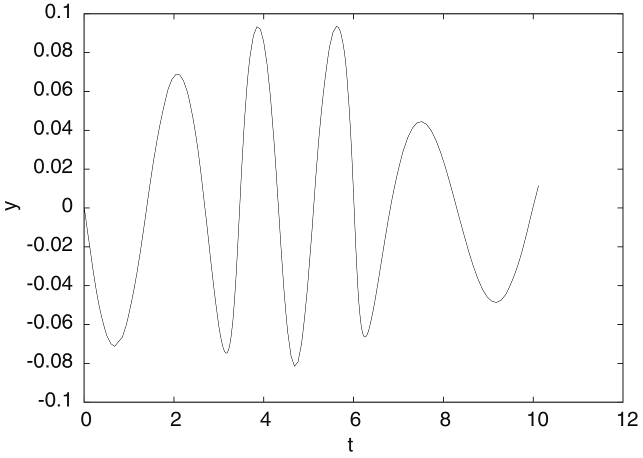


Fig. 12 Long transfer $y(t)$

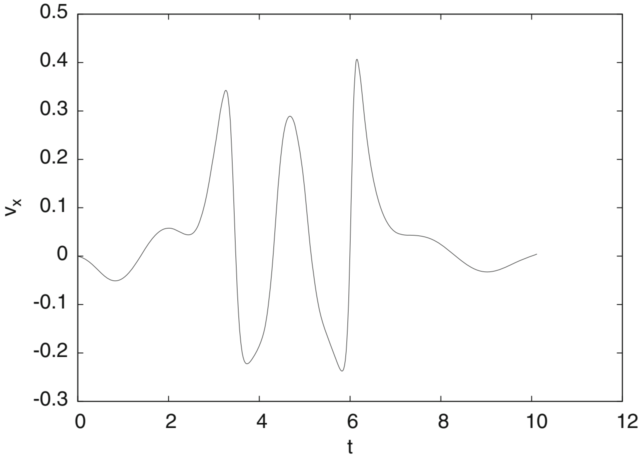


Fig. 13 Long transfer $v_x(t)$

8.2 Mesh Refinement Strategy

The results presented for the short transfer in Sect. 7.1 and the long transfer in Sect. 7.2 were obtained using a mesh refinement strategy that exploits the benefits of the higher order Lobatto discretization. The standard default strategy used by $\mathbb{SOS}$ is "(TRP),2;(HSC),20", that is, the trapezoidal method on the first two refinement iterations, followed by the compressed Simpson method on succeeding iterations. Table 8 summarizes the difference between the "optimal" and "standard" strategy for both cases. Observe that the optimal strategy requires fewer grid points, and

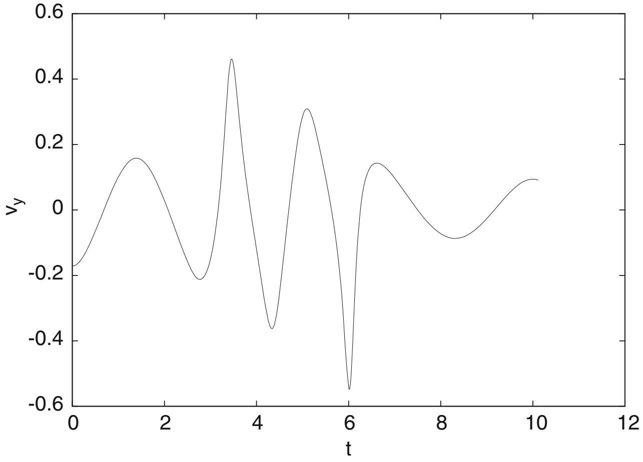


Fig. 14 Long transfer $v_y(t)$

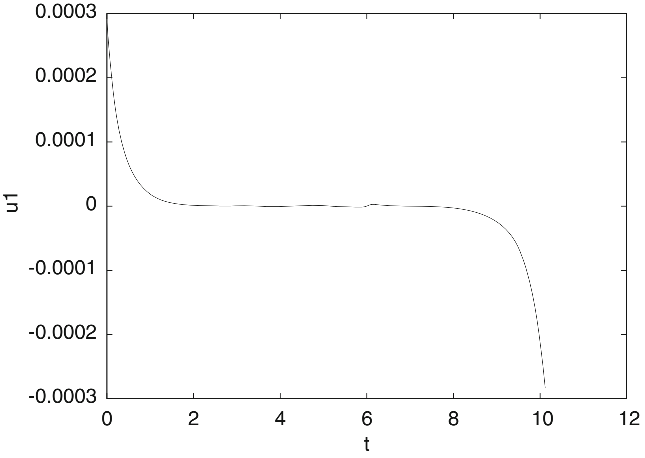


Fig. 15 Long transfer $u_1(t)$

less solution time when compared to the standard strategy, for both the short and long transfers. Furthermore an additional refinement iteration was required for the short transfer. It is also apparent that for the long transfer using a high order method during phases 2 and 3 is particularly effective.

8.3 Indirect Collocation

It is well known that the optimal control problem (1)–(4) can be formulated as a two-point boundary value problem. This approach is referred to as *indirect* because

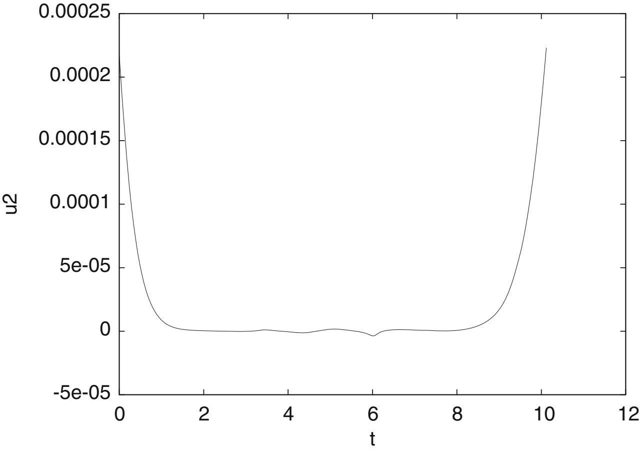


Fig. 16 Long transfer $u_2(t)$

Table 7 Impact of spline boundary condition on CPU time

Problem	𝕊𝕆𝕊 Algorithm (s)	Problem functions (s)	Total time (s)
Short transfer	1.54077	2.05368	3.59445
Short, spline BC	2.58261	0.326948	2.90956
Long transfer	14.5898	3.52747	18.1172
Long, spline BC	10.7704	0.409929	11.1803

Table 8 Impact of refinement strategy on efficiency

Problem, strategy	Ref. Iter.	M	NRHS	Total time (s)
Short, optimal	5	72	167,309	3.591
Short, standard	6	161	441,931	3.619
Long, optimal	5	210	1147,939	18.112
Long, standard	5	536	4,388,050	30.063

it entails both the problem dynamics as well as the optimality conditions. The adjoint equations for this example are given in Sect. 6.4, and when combined with the *transversality conditions*, a complete boundary value problem can be stated. As an alternative, let us consider an approach that does not require computation of the transversality conditions as suggested in reference [5], and illustrated in [2, 11]. In this formulation we consider a problem with the augmented set of *differential* variables $[\mathbf{z}(t), \boldsymbol{\lambda}(t)]$ and minimize

$$F = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{u}^\mathsf{T}(\boldsymbol{\lambda})\mathbf{u}(\boldsymbol{\lambda})\, dt. \tag{127}$$

subject to the differential equations

$$\dot{\mathbf{z}} = \mathbf{f}[\mathbf{z}, \mathbf{u}(\boldsymbol{\lambda})] \tag{128}$$

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \mathbf{z}} \tag{129}$$

where we define

$$\mathbf{u}(t) = \mathbf{u}\left[\boldsymbol{\lambda}(t)\right] = \begin{bmatrix} -\lambda_3(t) \\ -\lambda_4(t) \end{bmatrix} \tag{130}$$

which follows from (92) and (93). As before the initial and terminal states are fixed by the boundary conditions (61) and (62). The collocation method can be used to solve this problem having only differential variables and no algebraic (control) variables. In so doing the adjoint Eq. (129) can be viewed as simply a different way to parameterize the control variables. Since the objective function (127) appears directly there is no need to compute the transversality boundary conditions.

Unfortunately an indirect formulation suffers from a number of well known drawbacks. First, one must derive the adjoint equations i.e. (67)–(93), and in general this can be a rather daunting task! Secondly, an initial guess for the adjoint variables must be provided in order to begin an iterative solution, and this can be very challenging because the adjoint variables are not physical quantities. Even with a reasonable guess for the adjoint variables, the numerical solution of the adjoint equations can be very ill-conditioned! The sensitivity of the indirect method has been recognized for some time. Computational experience with the technique in the late 1960s is summarized by Bryson and Ho [6, p. 214]:

> The main difficulty with these methods is *getting started*; i.e., finding a first estimate of the unspecified conditions at one end that produces a solution reasonably close to the specified conditions at the other end. The reason for this peculiar difficulty is the extremal solutions are often *very sensitive* to small changes in the unspecified boundary conditions.... Since the system equations and the Euler–Lagrange equations are coupled together, it is not unusual for the numerical integration, with poorly guessed initial conditions, to produce "wild" trajectories in the state space. These trajectories may be so wild that values of $x(t)$ and/or $\lambda(t)$ exceed the numerical range of the computer!

These observations by Bryson and Ho reflect experience with the most common way to treat an indirect formulation such as (127)–(129) referred to as *indirect shooting*. For a shooting method, the differential equations are propagated using a standard numerical integration algorithm. In so doing the number of free variables is small, since the dynamic history is not discretized. Indeed, Epenoy [7] describes the steps required to solve this example using a shooting method, and deal with the solution sensitivity.

The reason for this particular difficulty serves to illustrate a fundamental difference between the collocation and shooting methods. With a shooting method the ordinary differential equations are numerically integrated, step by step from the initial to final time. The steps are of the form

$$\mathbf{y}(t_k + h_k) = \mathbf{y}(t_k) + \boldsymbol{\Delta}\mathbf{y}_k + \mathbf{e}_k$$

Table 9 Mesh refinement summary–indirect collocation, short transfer

k	M	Disc.	m	n	n_d	NRHS	Iter	ϵ	Time (s)
1	71	LA3	1128	1130	2	11,533	5	3.4×10^{-6}	0.4859
2	141	LA3	2248	2250	2	46,923	9	1.0×10^{-7}	0.4459
3	146	LA3	2328	2330	2	22,985	2	7.9×10^{-8}	0.1309
Total	146					81,441			1.0628

Table 10 Mesh refinement summary–indirect collocation, long transfer

k	M	Disc.	m	n	n_d	NRHS	Iter	ϵ	Time (s)
1	168	HSC	1344	1346	2	90,033	5	3.0×10^{-6}	0.6319
2	335	HSC	2680	2682	2	922,547	2	5.7×10^{-8}	1.7567
Total	335					1,012,580			2.3886

where $\boldsymbol{\Delta}\mathbf{y}_k$ is an estimate of the change over a step of length h_k, and $\mathbf{e}_k$ is the error in this approximation. Although the numerical integration algorithm will attempt to keep $\|\mathbf{e}_k\|$ "small" rarely is the error exactly zero. Thus, after a series of steps, the accumulated error in the solution can become large, i.e. $\sum_k \|\mathbf{e}_k\| \to \infty$. When this happens the system of ODE's is considered *unstable*. In contrast, a collocation method is not a serial process, because the entire dynamic history is altered by the NLP iterations. Since the defect constraints over the entire domain are addressed simultaneously, the collocation method can be viewed as global corrector iteration. Finally, it is worth noting that since the Lobatto IIIA schemes are symmetric, there is no particular advantage to integrating "forward" or "backwards." In short, a collocation method can deal with stability much more effectively than a shooting algorithm. Furthermore, if a direct collocation solution is available, then estimates for the adjoint variables can be computed from the NLP Lagrange multipliers [3, Sect. 4.11].

Using the direct solution as an initial guess, the indirect collocation method was used for both the short and long transfer examples. Tables 9 and 10 summarize the mesh refinement history for these cases. The results for these two cases demonstrate a number of points. Clearly the single phase indirect collocation formulation converges quickly for both cases. Unfortunately, when a good initial guess is not available, many of the issues of an indirect approach are not resolved by using collocation. In particular, from a "bad" guess, the method may either converge to a local solution or fail to converge at all.

9 Summary

This paper describes the implementation of a direct transcription method that incorporates high order Lobatto discretization of the problem dynamics. The technique is illustrated on a challenging low thrust orbit transfer example originally studied by Epenoy [7]. In addition a number of computational alternatives are discussed.

Appendix: Lobatto IIIA Method Coefficients

$$S = 2$$

$$\begin{array}{c|cc} \rho_1 & \alpha_{11} & \alpha_{12} \\ \rho_2 & \alpha_{21} & \alpha_{22} \\ \hline & \beta_1 & \beta_2 \end{array} \quad = \quad \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

$$S = 3$$

$$\begin{array}{c|ccc} \rho_1 & \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \rho_2 & \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \rho_3 & \alpha_{31} & \alpha_{32} & \alpha_{33} \\ \hline & \beta_1 & \beta_2 & \beta_3 \end{array} \quad = \quad \begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

$$S = 4$$

$$\begin{array}{c|cccc} \rho_1 & \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \rho_2 & \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \rho_3 & \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ \rho_4 & \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} \\ \hline & \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{array} \quad = \quad \begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} - \frac{\sqrt{5}}{10} & \frac{11+\sqrt{5}}{120} & \frac{25-\sqrt{5}}{120} & \frac{25-13\sqrt{5}}{120} & \frac{-1+\sqrt{5}}{120} \\ \frac{1}{2} + \frac{\sqrt{5}}{10} & \frac{11-\sqrt{5}}{120} & \frac{25+13\sqrt{5}}{120} & \frac{25+\sqrt{5}}{120} & \frac{-1-\sqrt{5}}{120} \\ 1 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \\ \hline & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \end{array}$$

$$S = 5$$

$$\begin{array}{c|ccccc}
\rho_1 & \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} & \alpha_{15} \\
\rho_2 & \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} & \alpha_{25} \\
\rho_3 & \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} & \alpha_{35} \\
\rho_4 & \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} & \alpha_{45} \\
\rho_5 & \alpha_{51} & \alpha_{52} & \alpha_{53} & \alpha_{54} & \alpha_{55} \\
\hline
 & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5
\end{array}$$

$$= \begin{array}{c|ccccc}
0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} - \frac{\sqrt{21}}{14} & \frac{119+3\sqrt{21}}{1960} & \frac{343-9\sqrt{21}}{2520} & \frac{392-96\sqrt{21}}{2205} & \frac{343-69\sqrt{21}}{2520} & \frac{-21+3\sqrt{21}}{1960} \\
\frac{1}{2} & \frac{13}{320} & \frac{392+105\sqrt{21}}{2880} & \frac{8}{45} & \frac{392-105\sqrt{21}}{2880} & \frac{3}{320} \\
\frac{1}{2} + \frac{\sqrt{21}}{14} & \frac{119-3\sqrt{21}}{1960} & \frac{343+69\sqrt{21}}{2520} & \frac{392+96\sqrt{21}}{2205} & \frac{343+9\sqrt{21}}{2520} & \frac{-21-3\sqrt{21}}{1960} \\
1 & \frac{1}{20} & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20} \\
\hline
 & \frac{1}{20} & \frac{49}{180} & \frac{16}{45} & \frac{49}{180} & \frac{1}{20}
\end{array}$$

References

1. Ascher, U.M., Mattheij, R.M.M., Russell, R.D.: Numerical Solution of Boundary Value Problems for Ordinary Differential Equations. Prentice-Hall, Englewood Cliffs (1988)
2. Bauer, T.P., Betts, J.T., Hallman, W.P., Huffman, W.P., Zondervan, K.P.: Solving the optimal control problem using a nonlinear programming technique part 2: optimal shuttle ascent trajectories. In: Proceedings of the AIAA/AAS Astrodynamics Conference, AIAA-84-2038, Seattle, WA (1984)
3. Betts, J.T.: Practical Methods for Optimal Control and Estimation using Nonlinear Programming, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2010)
4. Betts, J.T.: Optimal low-thrust orbit transfers with eclipsing. Optim. Control Appl. Methods **36**, 218–240 (2015)
5. Betts, J.T., Bauer, T.P., Huffman, W.P., Zondervan, K.P.: Solving the optimal control problem using a nonlinear programming technique part 1: general formulation. In: Proceedings of the AIAA/AAS Astrodynamics Conference, AIAA-84-2037, Seattle, WA (1984)
6. Bryson, A.E. Jr., Ho, Y.-C.: Applied Optimal Control. Wiley, New York (1975)
7. Epenoy, R.: Optimal long-duration low-thrust transfers between libration point orbits. In: 63rd International Astronautical Congress, no. 4 in IAC-12-C1.5.9, Naples (2012)
8. Herman, A.L., Conway, B.A.: Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules. AIAA J. Guid. Control Dyn. **19**, 592–599 (1996)
9. Jay, L.O.: Lobatto methods. In: Engquist, B. (ed.) Encyclopedia of Applied and Computational Mathematics, Numerical Analysis of Ordinary Differential Equations. Springer - The Language of Science, Berlin (2013)
10. Rao, A.V., Mease, K.D.: Eigenvector approximate dichotomic basis method for solving hypersensitive optimal control problems. Optim. Control Appl. Methods **20**, 59–77 (1999)
11. Zondervan, K.P., Bauer, T.P., Betts, J.T., Huffman, W.P.: Solving the optimal control problem using a nonlinear programming technique part 3: optimal shuttle reentry trajectories. In: Proceedings of the AIAA/AAS Astrodynamics Conference, AIAA-84-2039, Seattle, WA (1984)

Tentative Solutions for Indirect Optimization of Spacecraft Trajectories

Guido Colasurdo and Lorenzo Casalino

Abstract In this chapter, the problem of improving convergence and finding suitable tentative solutions for the indirect optimization of spacecraft trajectories is discussed. The application of theory of optimal control to spacecraft trajectories transforms the optimal control problem into a multi-point boundary value problem, which is usually solved by means of an iterative procedure. The convergence radius of the problem may be small and convergence to the optimal solution is only obtained if the tentative solution, which is used to start the procedure, is sufficiently close to the optimum. The definition of a suitable solution is often the hardest part of the solution procedure for the optimization problem. Several cases and examples are presented in this chapter to illustrate the measures that could be adopted for the most common difficulties, which may be found during the optimization of space trajectories.

Keywords Trajectory optimization • Optimal control theory • Indirect methods

1 Introduction

Trajectory analysis and optimization is a fundamental task in the design of a space mission. The trajectory directly influences the propellant consumption and consequently the mass budget, which is in turn directly related to the mission feasibility and costs. Trip time is an additional important factor, also related to the flown trajectory. Final mass or payload maximization, and flight time minimization are the problems that must be typically dealt with.

Most of the methods for the optimization of spacecraft trajectories can be grouped into three main classes [2]. Direct methods transform the problem into a parameter optimization (nonlinear programming) and solve it by means of gradient-

G. Colasurdo (✉)
Università di Roma "Sapienza", Via Eudossiana, 18, Roma, Italy
e-mail: guido.colasurdo@uniroma1.it

L. Casalino
Politecnico di Torino, Corso Duca degli Abruzzi, 24, Torino, Italy
e-mail: lorenzo.casalino@polito.it

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization and Its Applications 114, DOI 10.1007/978-3-319-41508-6_3

based procedures. Indirect methods use the theory of optimal control to transform the optimization problem into a boundary value problem (BVP), solved by means of shooting procedures. Evolutionary algorithms, instead, exploit large populations of solutions which evolve towards the global optimum according to specific rules that mimic natural phenomena.

The indirect approach offers many advantages and has been widely applied in the past to different problems concerning space trajectory optimization, mainly, but not exclusively, when low-thrust maneuvers are analyzed. A non-exhaustive list of examples comprises generic analysis of optimal trajectories [14, 18, 22, 23, 26] and global low-thrust trade studies [28], interplanetary transfers [24, 25, 27], geocentric transfers and Moon's missions [17, 20], and ascent trajectories [15]. The most important advantage of indirect methods is that they allow for an exact optimization (in the limits of the dynamical model and the accuracy of numerical integration). In addition, it is common opinion, even though not unanimous, that, at least for low-thrust missions, the computational cost of indirect methods is typically lower compared to direct methods, which require a very large number of parameters for an accurate description of the trajectory (severe approximations are instead necessary to genetic algorithms). Finally, the indirect approach provides useful theoretical information on the problem which is dealt with.

Indirect optimization methods are the subject of this chapter; they are based on the optimal control theory (OCT). The optimization problem is turned into a multi-point boundary value problem by the introduction of *adjoint variables*. OCT provides differential equations for the adjoint variables, algebraic equations for the determination of the control variables during the trajectory (as a function of state and adjoint variables at the same point), and a set of boundary conditions for optimality. The arising problem is therefore characterized by boundary conditions that must be fulfilled at initial, final and intermediate points (e.g., points where state variables are constrained or exhibit discontinuities). Some of the initial values of the state and adjoint variables are unknown and the values that allow satisfying the boundary conditions are sought. A shooting procedure is commonly employed to solve this kind of problems. Betts [2] highlights that the region of convergence for a shooting algorithm may be quite small, as it is necessary to guess at values for adjoint variables that may not have a clear physical meaning.

The BVP solution is made easier if the problem is formulated in a way that mitigates the drawbacks of the indirect methods. The authors in the past developed an approach [11] that makes the position of the problem and the derivation of the optimal conditions general and easy, thus allowing for the application of the indirect approach to very complex problems of spaceflight mechanics. Some enhancements have been introduced also in the BVP formulation, to improve convergence of the shooting procedure. However, the capability of achieving the numerical solution is still dependent on the tentative solution, which is assumed in order to start the procedure; methods to define suitable tentative solutions are presented in this chapter. The application of OCT to a generic spacecraft trajectory is considered and the derivation of the optimal control law and conditions for optimality are first summarized. The numerical procedure to obtain converged solutions is described; examples are presented with the main aim of providing indications on strategies capable of improving the numerical accuracy and finding the tentative solutions that guarantee convergence to the optimal solution.

2 Optimal Control Problem

The indirect approach to optimization uses OCT, which is based on calculus of variations; detailed presentation of OCT can be found in [3, 4, 18, 21]. The position of the optimization problem, which is here described, has the most suitable form to deal with the optimization of space trajectories and to exploit the capabilities of the numerical procedure that has been selected to solve the BVP.

The system is described by a set of state variables $\mathbf{x}$; differential equations rule the evolution from the initial to the final state

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \tag{1}$$

They are functions of $\mathbf{x}$, of the control variables $\mathbf{u}$, and of the independent variable t (usually but not necessarily, the time).

The trajectory between the initial and final point (external boundaries) is usefully split into n arcs at the points (internal boundaries) where the state or control variables are discontinuous or constraints are imposed. The jth arc starts at $t_{(j-1)_+}$ and ends at t_{j_-}, where the state variables are $\mathbf{x}_{(j-1)_+}$ and $\mathbf{x}_{j_-}$, respectively ($j-$ and $j+$ denote values just before and after point j).

Nonlinear constraints are imposed at both internal and external boundaries. These boundary conditions are grouped into a vector $\boldsymbol{\psi}$ and written in the form

$$\boldsymbol{\psi}(\mathbf{x}_{(j-1)_+}, \mathbf{x}_{j_-}, t_{(j-1)_+}, t_{j_-}) = 0 \qquad j = 1, \ldots, n \tag{2}$$

Additional path constraints may hold along an entire arc; constraints may also concern the control variables $\mathbf{u}$.

Meyer formulation is preferred to define the optimization problem, which searches for extremal values (maxima or minima) of a functional

$$J = \varphi(\mathbf{x}_{(j-1)_+}, \mathbf{x}_{j_-}, t_{(j-1)_+}, t_{j_-}) \qquad j = 1, \ldots, n \tag{3}$$

A necessary condition for optimality requires that the first variation of J is null for any admissible variation along the path ($\delta\mathbf{x}$ and $\delta\mathbf{u}$) and at boundary points ($\delta\mathbf{x}_{(j-1)_+}$, $\delta\mathbf{x}_{j_-}$, $\delta t_{(j-1)_+}$ and δt_{j_-}).

Lagrange multipliers (constants $\boldsymbol{\mu}$ associated with boundary conditions and adjoint variables $\boldsymbol{\lambda}$ associated with the differential equations) are introduced and a modified functional is defined

$$J^* = \varphi + \boldsymbol{\mu}^T \boldsymbol{\psi} + \sum_j \int_{t_{(j-1)+}}^{t_{j-}} \boldsymbol{\lambda}^T (\mathbf{f} - \dot{\mathbf{x}}) \mathrm{d}t \tag{4}$$

where the dot ($\dot{}$) denotes the time derivative.

The functionals J and J^* coincide if all boundary conditions and differential equations are satisfied. One can differentiate J^* and obtain

$$\begin{aligned}
\delta J^* = & \left(-H_{(j-1)+} + \frac{\partial \varphi}{\partial t_{(j-1)+}} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\psi}}{\partial t_{(j-1)+}} \right) \delta t_{(j-1)+} + \\
& + \left(H_{j-} + \frac{\partial \varphi}{\partial t_{j-}} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\psi}}{\partial t_{j-}} \right) \delta t_{j-} + \\
& + \left(\boldsymbol{\lambda}^T_{(j-1)+} + \frac{\partial \varphi}{\partial \mathbf{x}_{(j-1)+}} + \boldsymbol{\mu}^T \left[\frac{\partial \boldsymbol{\psi}}{\partial \mathbf{x}_{(j-1)+}} \right] \right) \delta \mathbf{x}_{(j-1)+} + \\
& + \left(-\boldsymbol{\lambda}^T_{j-} + \frac{\partial \varphi}{\partial \mathbf{x}_{j-}} + \boldsymbol{\mu}^T \left[\frac{\partial \boldsymbol{\psi}}{\partial \mathbf{x}_{j-}} \right] \right) \delta \mathbf{x}_{j-} + \\
& + \int_{t_{(j-1)+}}^{t_j} \left(\left(\frac{\partial H}{\partial \mathbf{x}} + \dot{\boldsymbol{\lambda}}^T \right) \delta \mathbf{x} + \frac{\partial H}{\partial \mathbf{u}} \delta \mathbf{u} \right) \mathrm{d}t \qquad j = 1, \ldots, n
\end{aligned} \tag{5}$$

where the Hamiltonian has been introduced

$$H = \boldsymbol{\lambda}^T \mathbf{f} \tag{6}$$

Optimality requires $\delta J^* = 0$ for any admissible variation. By nullifying the coefficients of $\delta \mathbf{x}$ and $\delta \mathbf{u}$ one has the Euler-Lagrange equations for the adjoint variables

$$\frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}t} = -\left(\frac{\partial H}{\partial \mathbf{x}} \right)^T \tag{7}$$

and algebraic equations for the control variables

$$\left(\frac{\partial H}{\partial \mathbf{u}} \right)^T = 0 \tag{8}$$

A control variable may be subject to constraints (e.g., the thrust magnitude varies between a minimum value, typically 0, and a maximum value T_{max}). In these cases, Eq. (8) might not provide the optimal control. However, in agreement with

Pontryagin's Maximum Principle (PMP) [3], the optimal control must maximize H, given by Eq. (6). In particular, when the Hamiltonian is linear with respect to a control variable (e.g., thrust magnitude), a bang-bang control arises, and the control assumes either its maximum or minimum value, except for singular arcs [3, 6].

Finally, the boundary conditions for optimality are obtained by nullifying the coefficients of $\delta\mathbf{x}_{(j-1)_+}$, $\delta\mathbf{x}_{j_-}$, $\delta t_{(j-1)_+}$, δt_{j_-}. One has

$$-\boldsymbol{\lambda}_{j_-}^T + \frac{\partial\varphi}{\partial\mathbf{x}_{j_-}} + \boldsymbol{\mu}^T\left[\frac{\partial\boldsymbol{\psi}}{\partial\mathbf{x}_{j_-}}\right] = 0 \qquad j = 1, \dots, n \tag{9}$$

$$\boldsymbol{\lambda}_{j_+}^T + \frac{\partial\varphi}{\partial\mathbf{x}_{j_+}} + \boldsymbol{\mu}^T\left[\frac{\partial\boldsymbol{\psi}}{\partial\mathbf{x}_{j_+}}\right] = 0 \qquad j = 0, \dots, n-1 \tag{10}$$

$$H_{j_-} + \frac{\partial\varphi}{\partial t_{j_-}} + \boldsymbol{\mu}^T\frac{\partial\boldsymbol{\psi}}{\partial t_{j_-}} = 0 \qquad j = 1, \dots, n \tag{11}$$

$$-H_{j_+} + \frac{\partial\varphi}{\partial t_{j_+}} + \boldsymbol{\mu}^T\frac{\partial\boldsymbol{\psi}}{\partial t_{j_+}} = 0 \qquad j = 0, \dots, n-1 \tag{12}$$

The constant Lagrange multipliers are eliminated form Eqs. (9)–(12); the resulting boundary conditions for optimality and the boundary conditions on the state variables, given by Eq. (2), are collected in a single vector in the form

$$\boldsymbol{\sigma}(\mathbf{x}_{(j-1)_+}, \mathbf{x}_{j_-}, \boldsymbol{\lambda}_{(j-1)_+}, \boldsymbol{\lambda}_{j_-}, t_{(j-1)_+}, t_{j_-}) = 0 \tag{13}$$

which, together with state and adjoint differential equations, defines a multi-point boundary value problem (MPBVP).

Noticeable difficulties in the MPBVP solution arise when the relevant times are unknown and the lengths of the integration intervals are free. A change of independent variable is introduced [12] to overcome the indetermination of the relevant times and fix the extremes of the integration intervals; in the jth phase, t is replaced by

$$\varepsilon = j - 1 + \frac{t - t_{j-1}}{t_j - t_{j-1}} \tag{14}$$

which assumes consecutive integer values at the boundaries. The differential equation for the variable vector $\mathbf{y}$ (which collects state, $\mathbf{x}$, and adjoint, $\boldsymbol{\lambda}$, variables, and also unknown constant parameters) during phase j becomes

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\varepsilon} = (t_j - t_{j-1})\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{f}'(\mathbf{y}) \tag{15}$$

This peculiar position of the problem is also useful to handle problems involving multiple satellites, as described in Sect. 5.

3 Application to Space Trajectories Optimization

Preliminary analysis of spacecraft trajectories is typically carried out assuming a point mass spacecraft under the influence of a single body. The two-body model can also be used to deal with interplanetary trajectories, as the patched-conic approximation is usually employed in preliminary analyses. An efficient approach only analyzes the heliocentric legs; at the patch points with the planetocentric legs, suitable boundary conditions take the maneuvers inside the planets' spheres of influence into account. Indirect methods are however capable of dealing with more complex dynamical models, which consider, for instance, main body oblateness, third-body perturbations, solar radiation pressure, aerodynamic forces. This indirect approach has also been applied to trajectory optimization in the restricted three-body problem [10]. The formulation of the trajectory optimization in the two-body problem with the addition of a generic perturbation term $\mathbf{a}_p$, which is a function of time and state variables, is given here.

The state of the spacecraft is described by position $\mathbf{r}$, velocity $\mathbf{v}$ and mass m and the state equations are

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \mathbf{v} \tag{16}$$

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{g} + \frac{\mathbf{T}}{m} + \mathbf{a}_p \tag{17}$$

$$\frac{\mathrm{d}m}{\mathrm{d}t} = -\frac{T}{c} \tag{18}$$

where $\mathbf{T}$ is the engine thrust and $\mathbf{g}$ is the gravitational acceleration (an inverse-square gravity field $\mathbf{g} = -\mu\mathbf{r}/r^3$ is typically assumed); the propellant mass-flow rate is expressed by the ratio of the thrust magnitude to the constant effective exhaust velocity c. This indirect method is also capable of treating variable c, i.e., propulsion systems with adjustable specific impulse [7].

The Hamiltonian, defined by Eq. (6), is

$$H = \boldsymbol{\lambda}_r^T \mathbf{v} + \boldsymbol{\lambda}_v^T (\mathbf{g} + \mathbf{T}/m + \mathbf{a}_p) - \lambda_m T/c \tag{19}$$

The thrust direction and its magnitude are typically the control variables, which must maximize H in agreement with PMP [3]. The optimal thrust direction is therefore parallel to the velocity adjoint vector $\boldsymbol{\lambda}_v$, which is named primer vector [22]. The *switching function*

$$S_F = \frac{\lambda_v}{m} - \frac{\lambda_m}{c} \tag{20}$$

is introduced, and Eq. (19) is rewritten as

$$H = \boldsymbol{\lambda}_r^T \mathbf{v} + \boldsymbol{\lambda}_v^T \mathbf{g} + T S_F \tag{21}$$

The thrust magnitude assumes its maximum value when the switching function S_F is positive, whereas it is set to zero when S_F is negative, again to maximize the Hamiltonian. Singular arcs occur when S_F remains zero during a finite time; Eq. (21) is not sufficient to decide the optimal thrust magnitude (singular arcs are here excluded; they may be required during atmospheric flight [6]).

The Euler-Lagrange equations for the adjoint variables, Eq. (7), provide [8]

$$\left[\frac{\mathrm{d}\boldsymbol{\lambda}_r}{\mathrm{d}t}\right]^T = -\boldsymbol{\lambda}_v^T\left[\frac{\partial(\mathbf{g}+\mathbf{a}_p)}{\partial\mathbf{r}}\right] \tag{22}$$

$$\left[\frac{\mathrm{d}\boldsymbol{\lambda}_v}{\mathrm{d}t}\right]^T = -\boldsymbol{\lambda}_r^T - \boldsymbol{\lambda}_v^T\left[\frac{\partial\mathbf{a}_p}{\partial\mathbf{v}}\right] \tag{23}$$

$$\frac{\mathrm{d}\lambda_m}{\mathrm{d}t} = \frac{\lambda_v T}{m^2} - \boldsymbol{\lambda}_v^T\left[\frac{\partial\mathbf{a}_p}{\partial m}\right] \tag{24}$$

Equations (16)–(18) and (22)–(24) constitute the system of differential equations, which is numerically integrated. The MPBVP is completed by Eq. (13), which collects the constraints on the state variables, enforced by Eq. (2), and the boundary conditions for optimality, derived from Eqs. (9)–(12). Some examples can be found in [11].

A certain number of initial values of the state and adjoint variables are unknown; several constant parameters, such as the relevant times t_j, may also be unknown. They are collected in vector $\mathbf{p}$. An iterative procedure is used to determine the unknowns that permit the fulfillment of Eq. (13). Tentative values are first assumed; it is very important that these initial values are sufficiently close to the optimal solution to guarantee convergence. The assumption of a suitable solution is a fundamental step in the optimization procedure; details will be given in Sect. 5.

4 BVP Solution and Improvements of Numerical Accuracy

Variable normalization should be adopted. Convergence difficulties may arise when the orders of magnitude of variables exhibit large differences. A proper scaling, for instance to make the magnitude of all variables close to unit, allows for a much easier convergence.

Once tentative values have been assigned to the unknowns $\mathbf{p}$, the differential equations are integrated and the errors $\boldsymbol{\sigma}$ on the boundary conditions are found. Newton's method is used to bring the errors to zero. The unknowns are in turn varied by a small amount (e.g., 10^{-6}) to evaluate, according to a forward-finite-difference scheme, the derivatives of the errors with respect to the unknowns. The correction of the tentative values is thus obtained under a linear approximation

$$\Delta\mathbf{p} = -K\left[\frac{\partial\boldsymbol{\sigma}}{\partial\mathbf{p}}\right]^{-1}\mathbf{p} \tag{25}$$

where the relaxation factor K (between 0 and 1) reduces the theoretical correction. The linearized approach is accurate only for small variations. Relaxation may be required in the presence of large theoretical corrections (often typical of the first iterations), and implemented by adopting $K < 1$. Either constant or variable K (increasing towards 1 as the number of iterations grows) can be used.

4.1 Gradient Evaluation

An accurate gradient evaluation is usually required to obtain convergence. Under this point of view, some techniques are very useful. If the problem is very sensitive to the initial values (e.g., three-body problem, atmospheric flight), a more precise analytical approach [12] may replace the one-sided finite-difference scheme described above. The procedure is usually rather heavy in terms of both analytical effort to derive the necessary equations, and time to program and debug the numerical code; computational time also increases and this approach should be used only when other techniques fail to provide convergence. The error gradient matrix can be evaluated as

$$\left[\frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{p}}\right] = \left[\frac{\partial \boldsymbol{\sigma}}{\partial \mathbf{s}}\right]\left[\frac{\partial \mathbf{s}}{\partial \mathbf{p}}\right] \tag{26}$$

where $\mathbf{s} = (\mathbf{y}_0, \mathbf{y}_{1-}, \mathbf{y}_{1+}, \ldots, \mathbf{y}_f)$ collects the values that state and adjoint variables assume at relevant boundaries. The error gradient with respect to these values $[\partial\boldsymbol{\sigma}/\partial\mathbf{s}]$ is obtained by analytical derivation. The derivative of $\mathbf{s}$ with respect to the unknowns $\mathbf{p}$ contains the values assumed by the transition matrix $[\partial\mathbf{y}/\partial\mathbf{p}]$ at the same boundaries ($\varepsilon = 0, 1-, 1+, \ldots, f$). The transition matrix is in turn obtained by integrating the homogeneous system

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\left[\frac{\partial \mathbf{y}}{\partial \mathbf{p}}\right] = \left[\frac{\partial}{\partial \mathbf{p}}\left(\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\varepsilon}\right)\right] = \left[\frac{\partial \mathbf{f}'}{\partial \mathbf{p}}\right] = \left[\frac{\partial \mathbf{f}'}{\partial \mathbf{y}}\right]\left[\frac{\partial \mathbf{y}}{\partial \mathbf{p}}\right] \tag{27}$$

The Jacobian matrix $[\partial\mathbf{f}'/\partial\mathbf{y}]$ is obtained by analytical derivation.

4.2 Thrust Discontinuities

The integration strategy may greatly affect gradient-evaluation accuracy and therefore convergence, in particular when the control variables exhibit jumps; this is the case of the thrust magnitude in space trajectory optimization, as it usually exhibits a bang-bang control. During integration, the current value of the switching function is often used to choose the thrust level. In such an instance, variable step integration schemes (such as the Adams-Moulton variable-step, variable-order scheme used by the authors) are preferable as the integration step is adjusted in correspondence of the discontinuities to guarantee the required accuracy. A fixed-step integration may introduce errors that affect the convergence process.

Additional techniques may however be required to improve the numerical accuracy, even though an intrinsically precise integration scheme is employed; to this purpose, the trajectory is split into maximum-thrust arcs and coast arcs. The number and order of the arcs, that is, the trajectory switching structure, is assigned a priori, and the arc time-lengths are additional unknowns. The boundary conditions for optimality state that the switching function S_F must be null at the extremities of each thrust arc. According to Eq. (20), one enforces

$$\frac{\lambda_{vj}}{m_j} - \frac{\lambda_{mj}}{c} = 0 \tag{28}$$

at each relevant boundary. These constraints are added to Eq. (13).

The numerical procedure provides the optimal solution that corresponds to the assigned switching structure. This solution is eventually checked in the light of PMP, by analyzing the switching function; if PMP is violated, coast or propelled arcs are inserted or removed, according to the behavior of S_F, to obtain an improved solution (e.g., a coast arc is introduced when S_F becomes negative during a propelled arc). Smoothing techniques [1] are alternatively employed. However, they usually require a very large computational effort, offsetting the advantages of indirect methods.

4.3 Multiple Shooting

Multiple shooting is an additional technique that can be used to improve convergence. Trajectory problems are typically highly nonlinear and small variations of a parameter may have very large effects on the boundary conditions. In this case, Newton's linearized method may not work. A multiple shooting approach splits the trajectory into subarcs at specified points (e.g., planetary encounters). The variables, which are necessary to start the integration in each new subarc, are treated as additional problem unknowns. Proper boundary conditions are introduced to guarantee the trajectory continuity at the arc junctions. In this way the influence of the unknowns on the errors is reduced, at the expense of a larger number of unknowns. Convergence is typically easier even though computational times increase.

5 Techniques to Improve Convergence

Due to the intrinsic difficulty of trajectory optimization, convergence problems may still persist after suitable techniques have been adopted to allow for a correct gradient evaluation. In these cases, it is the user's experience that guides and handles the convergence process, as each problem may require its peculiar approach.

A traditional method to find suitable tentative solutions is the homotopy continuation approach, which has been profitably applied to many problems of space trajectory optimization [1, 5, 9, 13, 16, 19, 30]. Homotopy is based on the definition of an auxiliary problem (with some sort of connection to the original one) whose solution is known or can be easily found. The solution of the auxiliary problem is used as starting guess and is gradually moved towards the solution of the original optimal problem by constructing intermediate problems. For instance, a linear combination of the performance indexes and boundary conditions of the original and auxiliary problems can be used. Each intermediate problem is solved starting from the solution of the previous step. This technique is usually very effective but may be demanding from the computational point of view, as the solution of many boundary value problems is required. A vast literature exists about homotopy; the most common applications of this approach are for the transition from known or easy-to-find solutions to the optimal one. For instance, from minimum-energy to minimum-fuel problem [5, 19], from a simpler to a more detailed dynamical model [9, 13] or from a known highly-constrained solution to the optimal one [16, 30]. Homotopy in conjunction with smoothing techniques has been proposed [1, 19] to assess the switching times in bang-bang control problems. Discussion is here limited to aspects related to search for the optimal switching structure.

One of the most critical issues for indirect optimization of a space trajectory is often the definition of the switching structure. Techniques specifically tailored for the problem at hand can greatly facilitate this task. A multi-revolution perigee raising transfer to a high eccentricity orbit (HEO) with an accurate dynamical model, which takes relevant perturbations (Earth oblateness and lunisolar gravity perturbation) into account, can be used as an example to illustrate these techniques [31–33]. A perigee raising maneuver is conveniently split between multiple apogee passages, when perturbations are not considered. However, the relatively large effect of Moon's gravity may cause some apogee burn to increase and others to diminish or even vanish, so that the switching structure cannot be assessed a priori. The duration of the apogee burns is tailored to modify the orbital period with the purpose of reaching favorable configurations (where the Moon contributes to the required orbit changes) and avoid unfavorable ones (where the Moon acts against them). The initial burns of the unperturbed solution vanish when it is convenient to shorten the mission length, whereas they are enlarged (and the final burns vanish) when the mission time-length must be increased.

The switching structure of the unperturbed problem is modified by the introduction of perturbations; it is advisable to consider a fraction P_f of the perturbation that, according to a continuation approach, gradually increases from 0 to 1. Failure to converge using the switching structure of the previous solution signals a request for a modified structure, which can be easily determined by inspecting the switching function of the last converged solution. This task is automatically done by comparing the maximum value of the switching function during each burn arc and removing the arc that presents the lowest values. An example of switching function behavior for different perturbation fractions during a HEO transfer is shown in Fig. 1, where the third apogee burn (A3) must be removed to obtain convergence

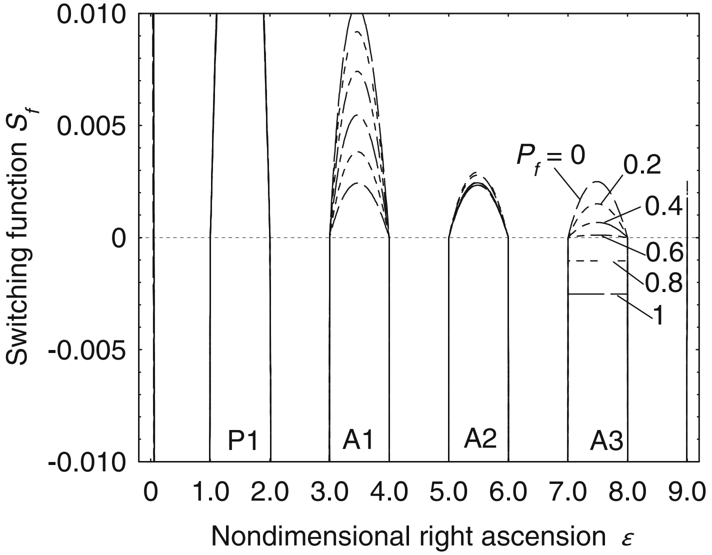


Fig. 1 Spacecraft deployment in HEO: switching function history for different values of the perturbation fraction P_f

for $P_f > 0.6$. It is interesting to note that the solution of the unperturbed problem can also be easily obtained by means of a continuation technique, starting from a minimum-time trajectory with a lower thrust level. As the thrust magnitude is increased, coast arcs are introduced where required by PMP.

Besides the initial assumption of the switching structure, a guess at the location of the arc extremities is necessary. The use of time as the independent variable may prevent convergence when multiple revolutions are performed, because it is often difficult to estimate the apogee passage times (which depend on the period of previous revolutions and therefore on the amount of energy increase obtained during the previous burns). Using longitude as the independent variable facilitates the definition of a starting guess, since the burns are always in the proximity of passages at the apsides and can therefore be easily estimated. As an example, Table 1 compares times and longitudes at the extremities of each burn and orbit insertion for deployments to a given HEO, departing on different dates (P and A indicate perigee and apogee burns, respectively). It is evident that times exhibit relevant changes, whereas it is quite easy to guess at the corresponding longitudes which assume similar values. In Table 1, the same values of time and longitude at both extremes of a burn arc indicate that this arc (A3 or A1, for the early or late departure, respectively) has vanished and a ballistic apogee passage is instead imposed.

The HEO deployment of a two-spacecraft formation is also useful to illustrate how a smart position of the constraints can be effective to improve convergence. In the case considered in [32], two satellites had to be inserted into the same orbit with a 10-km distance in the apogee proximity. This constraint may be difficult to

Table 1 Switching times and longitudes for HEO deployment starting on different dates

	Jan. 1, 2015		Jan. 15, 2015	
Event	Time (h)	Long. (rad)	Time (h)	Long. (rad)
P1 start	84.4	10.73	84.3	10.75
P1 end	84.5	11.27	84.4	11.25
A1 start	125.7	14.13	126.5	14.14
A1 end	129.2	14.16	126.5	14.14
A2 start	212.4	20.41	211.7	20.42
A2 end	219.2	20.46	214.9	20.44
A3 start	311.8	26.70	299.9	26.69
A3 end	311.8	26.70	304.9	26.74
Arrival	408.2	33.00	397.8	33.00

handle, as relative distance depends on three state variables that define the spacecraft position. The observation that the satellites share the same orbit suggests turning the position constraint into a time constraint. The same point (i.e., the apogee) is chosen as the final point for deployment of each satellite, but a time-delay (the ratio of the required distance to the velocity at the final point) is imposed for the arrival of the follower spacecraft. The same simple constraints enforce the required position and velocity at apogee for both satellites.

In the case of a cooperative deployment, each satellite adopts a specific strategy with the purpose of better splitting the phasing duty and reducing the overall propellant consumption. The motion of the satellites has to be evaluated and optimized simultaneously; handling the engine switching times of both spacecraft can be a major difficulty. The unique time-like variable ε is able to handle differences in time of the switching points, as shown in Fig. 2. The variable transformation allows defining the switching structure of each spacecraft independently from the other, since each of them has its own time scale. The number of thrust and coast arcs is, in general, different for the two spacecraft. It is convenient to split the trajectory of both satellites into the same number of arcs, by adding a suitable number of zero-length arcs (e.g., the fifth and sixth arc of SAT2 in Fig. 2). This also permits the alignment of perigee and apogee burns, simplifying the problem description and solution.

When the homotopic approach is not feasible or too demanding, specific strategies should be devised for the particular problem that must be solved. In a former work [11], the authors discussed and highlighted the benefits of trajectory patching and of the use of optimal phasing solutions as starting guesses. Optimal phasing trajectories, which assume the most favorable position of a relevant body along its orbit, are useful when several bodies are concerned (e.g., planets that can provide gravity assist) and can be extremely effective in finding suitable launch windows for high-performance trajectories. The trajectory is split into simple legs, that are first separately optimized and then patched together to provide a tentative solution to optimize the complete trajectory, possibly using a multiple shooting approach.

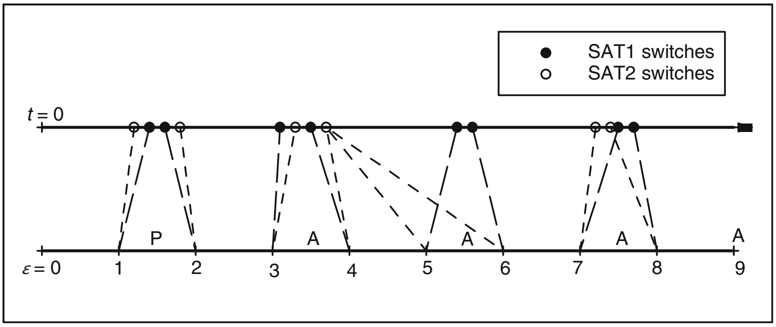


Fig. 2 Two-spacecraft cooperative maneuver: alignment of thrust and coast arcs provided by nondimensional time-like variable ε

Evolutionary algorithms (EAs) may be an useful tool when used in conjunction with direct and indirect methods. The authors employed an EA [29] to define tentative solutions for their indirect method. A suitable choice of the performance index (e.g., overall error on the boundary conditions, time of flight, propellant consumption or a combination of them) can facilitate finding suitable sets of tentative values, exploring different launch windows, and defining the switching structure for the indirect optimization. In particular, when interplanetary transfers are analyzed, one usually needs the mission opportunities (that is, all the local optima that characterize this kind of transfers) and the Pareto front, in terms of payload and trip-time. Finding each local optimum requires a suitable tentative solution, and this task is often the hardest part of the optimization process. When properly employed, an EA can provide a set of tentative solutions that allow convergence to the maximum-payload missions that are present in the considered time window. Figure 3 shows an example for an Earth-Mars transfer using solar electric propulsion. A genetic algorithm was used to provide tentative solutions to the indirect method, which in turn found six locally optimal trajectories (D1–D6). Starting from these solutions, the Pareto front was determined by constraining the trip time at different values. The final mass remains constant between D2 and D2'. D3–D3' and D4–D4': these solutions only differ because of the addition of a final coasting arc to attain the required time-length.

6 Final Remarks

This chapter presented an indirect procedure that is based on a multi-arc structure of the trajectory. Derivation of the necessary conditions for optimality is quite simple and can be applied to a very large number of different problems. The multi-arc structure is useful to handle constraints and variable discontinuities, permitting the optimization of complex trajectories, which present features such as staging and

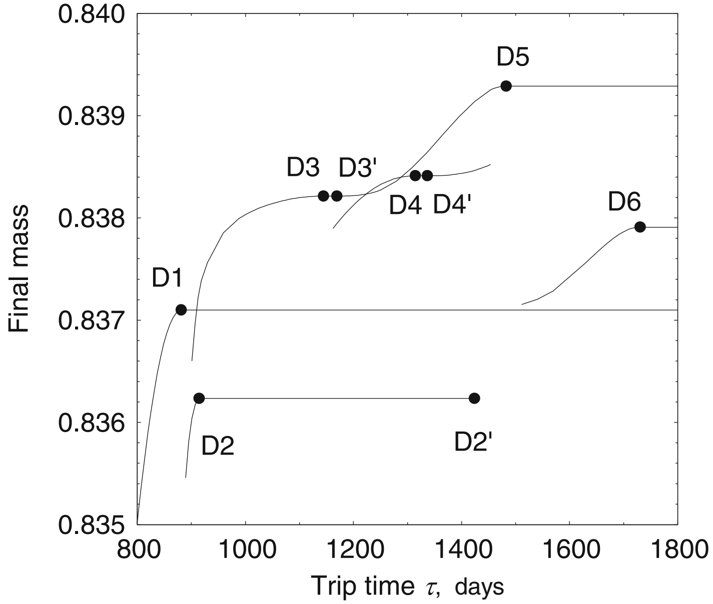


Fig. 3 Example of Pareto front for Earth-Mars transfer in an assigned time window

multiple gravity assists. Integration accuracy and precise determination of the error gradients have a strong influence on the capability of attaining the optimal solution. To this purpose, it is convenient a further split of the trajectory, permitting only a constant thrust-level in any arc; thrust discontinuities occur at arc extremities.

Indirect optimization methods require a suitable tentative solution to start the process and permit convergence to the optimal solution. According to the proposed approach, also the switching structure, that is the sequence of coasting and burn arcs together with the constraint positions, has to be a priori guessed. The tentative solutions, which comprise arc time-lengths, are typically found by means of a homotopic approach. Every problem has however its specific features and appropriate strategies to get the optimal solution. Several examples have been presented in this chapter and could provide useful suggestions to solve similar problems.

References

1. Bertrand, R., Epenoy, R.: New smoothing techniques for solving bang-bang optimal control problems-numerical results and statistical interpretation. Optim. Control Appl. Methods **23**, 171–197 (2002)
2. Betts, T.: Survey of numerical methods for trajectory optimization. J. Guid. Control Dyn. **21**, 193–207 (1998)

3. Bryson, E.A., Ho, Y.-C.: Applied Optimal Control. Hemisphere, New York (1975)
4. Burghes, D.N., Graham, A.: Introduction to Control Theory, Including Optimal Control. Wiley, New York (1980)
5. Caillau, J.B., Daoud, B., Gergaud, G.: Minimum fuel control of the planar circular restricted three-body problem. Celest. Mech. Dyn. Astron. **114**, 137–150 (2012)
6. Casalino, L.: Singular arc during aerocruise. J. Guid. Control Dyn. **23**, 118–123 (2000)
7. Casalino, L., Colasurdo, G.: Optimization of variable-specific-impulse interplanetary trajectories. J. Guid. Control Dyn. **27**, 678–684 (2004)
8. Casalino, L., Colasurdo, G., Pastrone, D.: Optimal low-thrust escape trajectories using gravity assist. J. Guid. Control Dyn. **22**, 637–642 (1999)
9. Cerf, M., Haberkorn, T., Trelat, E.: Continuation from a flat to a round Earth model in the coplanar orbit transfer problem. Optim. Control Appl. Methods **33**, 654–675 (2012)
10. Colasurdo, G., Casalino, L.: Optimal ΔV-Earth-gravity-assist trajectories in the restricted three-body problem. Paper AAS 99-409 (1999)
11. Colasurdo, G., Casalino, L.: Indirect methods for the optimization of spacecraft trajectories. In: Fasano, G., Pinter, J.D. (eds.) Modeling and Optimization in Space Engineering, pp. 141–158. Springer, New York (2012)
12. Colasurdo, G., Pastrone, D.: Indirect optimization method for impulsive transfer. Paper AIAA 94-3762, AIAA, Reston (1994)
13. Dadebo, S.A., McAuley, K.B., McLellan, P.J.: On the computation of optimal singular and bang-bang controls. Optim. Control Appl. Methods **19**, 287–297 (1998)
14. Edelbaum, T.N.: Optimal Space Trajectories. Analytical Mechanics Associates, Gericho (1969)
15. Gath, P.F., Well, K.H., Mehlem, K.: Initial guess generation for rocket ascent trajectory optimization using indirect methods. J. Spacecr. Rockets **39**, 515–521 (2002)
16. Graichen, K., Petit, N.: A continuation approach to state and adjoint calculation in optimal control applied to the reentry problem. In: 17th IFAC World Congress, Seoul (2008)
17. Guelman, M.: Earth-to-moon transfer with a limited power engine. J. Guid. Control Dyn. **18**, 1133–1138 (1995)
18. Hull, D.G.: Optimal Control Theory for Applications. Springer, New York (2003)
19. Jiang, F., Baoyin, H., Li, J.: Practical techniques for low-thrust trajectory optimization with homotopic approach. J. Guid. Control Dyn. **35**, 245–258 (2012)
20. Kechichian, J.A.: Reformulation of Edelbaum's low-thrust transfer problem using optimal control theory. J. Guid. Control Dyn. **20**, 988–994 (1997)
21. Kirk, D.E.: Optimal Control Theory: An introduction. Prentice-Hall, Englewood Cliffs (1970)
22. Lawden, D.F.: Optimal Trajectories for Space Navigation. Butterworths, London (1963)
23. Marec, J.P.: Optimal Space Trajectories. Elsevier, Amsterdam (1979)
24. Melbourne, W.G., Sauer, C.G. Jr.: Optimum Earth-to-Mars roundtrip trajectories utilizing a low-thrust power-limited propulsion system. J. Astronaut. Sci. **13**, 547–570 (1963)
25. Nah, R.S., Vadali, S.R., Braden, E.: Fuel-optimal, low-thrust, three-dimensional Earth-Mars trajectories. J. Guid. Control Dyn. **24**, 1100–1107 (2001)
26. Prussing, J.E.: Equations for optimal power-limited spacecraft trajectories. J. Guid. Control Dyn. **16**, 391–393 (1993)
27. Ranieri, C.L., Ocampo, C.A.: Optimization of roundtrip, time-constrained, finite burn trajectories via an indirect method. J. Guid. Control Dyn. **28**, 306–314 (2005)
28. Russell, R.P.: Primer vector theory applied to global low-thrust trade studies. J. Guid. Control Dyn. **30**, 460–472 (2007)
29. Sentinella, M.R., Casalino, L.: Genetic algorithm and indirect method coupling for low-thrust trajectory optimization. Paper AIAA 2006-4468 (2006)
30. Shen, H.X., Casalino, L.: Indirect optimization of three-dimensional multiple-impulse Moon-to-Earth transfers. J Astronaut. Sci. **61**, 255–274 (2014)

31. Simeoni, F., Casalino, L., Zavoli, A., Colasurdo, G.: Indirect optimization of satellite deployment into a highly elliptic orbit. Int. J. Aerosp. Eng. **2012**, Article ID 152683, 14 pp. (2012)
32. Simeoni, F., Casalino, L., Zavoli, A., Colasurdo, G.: Deployment of a two-spacecraft formation into a highly elliptic orbit with collision avoidance. Paper AIAA-2012-4740 (2012)
33. Zavoli, A., Simeoni, F., Casalino, L., Colasurdo, G.: Optimal cooperative deployment of a two-satellite formation into a highly elliptic orbit. Paper AAS 11-641 (2011)

Resource-Constrained Scheduling with Non-constant Capacity and Non-regular Activities

Giorgio Fasano

Abstract This work is inspired by very challenging issues arising in space logistics. The problem of scheduling a number of activities, in a given time elapse, optimizing the resource exploitation is discussed. The available resources are not constant, as well as the request, relative to each job. The mathematical aspects are illustrated, providing a time-indexed MILP model. The case of a single resource is analysed first. Extensions, including the multi-resource case and the presence of additional conditions are considered. Possible applications are suggested and an in-depth experimental analysis is reported.

Keywords Resource constrained project scheduling problem • Non-constant resource capacity • Non-constant resource request • Irregular job/activity/cycle profile • Multi-resource • Time-indexed scheduling • Mixed integer linear programming • Global optimization

1 Introduction

This work is inspired by the logistic context in space activities. It is notorious that, in this framework, the exploitation of the resources available (e.g. on orbit or on the exploration surface) is usually an extremely challenging issue. Complex scheduling problems arise, presenting the experts with the necessity of optimizing the sequencing of what is usually a significant number of jobs, requiring contemporarily the utilization of different resources, such as, electrical power, data handling capacity and crew time. As a further non-trivial difficulty, the operational cycles (jobs) are frequently associated with an irregular activity, i.e. they are characterized by a variable resource request profile. Similarly, the overall capacities of the relevant resources vary. Figure 1 provides, as an illustrative example, the case of a single (non-constant) resource and three different (non-constant) request cycle types.

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

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization and Its Applications 114, DOI 10.1007/978-3-319-41508-6_4

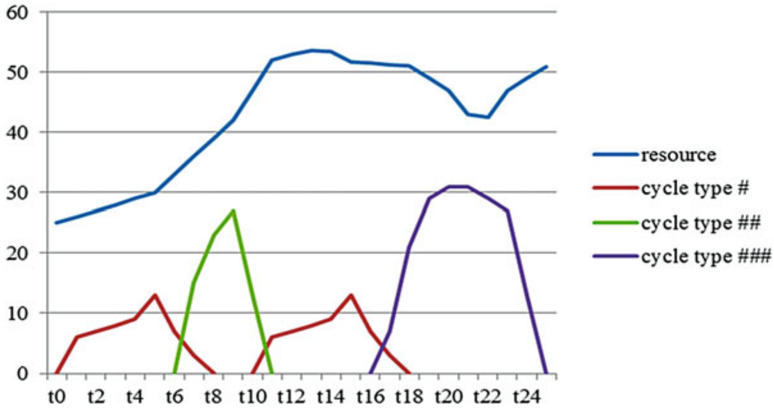


Fig. 1 Irregular cycles and non-constant resource

The specialist literature on scheduling is vast [1–4, 8, 15–21], covering several specific problems and methodologies. This chapter focuses on a non-standard resource constrained project scheduling problem (RCPSP). For the classical RCPSP see [5–7, 10–12, 14, 22–27]. In the author's previous work [9], an approximate MILP (Mixed Integer Linear Programming) time-continuous approach, was proposed. A novel approximate MILP formulation of the problem, based on a time-indexed approach, is discussed here. This latter option was motivated by the efficiency of discretized models for scheduling problems [28]. As in the previous work, the approach proposed in this chapter provides a global optimization (GO) perspective on the problem in question. The discussed formulation is suitable for tackling a number of different variants of the RCPSP, involving either single or multiple resources and characterized by the specific objective functions adopted.

The remainder of the chapter is organized as follows. The first part of Sect. 2 provides an MILP model for the case of a single resource, namely electrical power [8, 29, 30]. Afterwards, the presence of possible additional conditions is outlined and an extended formulation, addressing the multi-resource scenario is introduced. Sect. 3 is devoted to the computational study.

2 Time-Indexed Formulation

The problem considered in this section, concerns the electrical power consumption, by a number of devices (e.g. payloads, in the case of the space framework), in a pre-specified time period. Each device may be requested to execute a sequence of cycles, of type τ, between a given minimum and maximum limit, i.e. $\underline{N}_\tau$ and $\overline{N}_\tau$, respectively. Assuming, for the sake of simplicity, that the value associated with each device cycle is the same (this assumption could be generalized by introducing appropriate weights), the optimization objective consists of maximizing the exploitation of the energy available during the entire time period $[0, T_f]$, where T_f denotes the final time.

Some additional conditions (e.g. of time-precedence), might be imposed on the execution of the cycles (relevant examples shall be provided). The (electrical) power available at each instant $t \in \left[0, T_f\right]$ is represented by a given function of time $w(t)$ (e.g. step-wise or continuous, see Fig. 1). Similarly, each cycle type $\tau \in T$ (T indicates the set of all cycle types) is associated with a given function of time (cycle type profile) $w_\tau(t)$, defined (conventionally) over $[0, D_\tau]$, where D_τ corresponds to cycle type τ duration. All activated cycles must obviously be entirely executed within the given overall time period. This means that, denoting the initial instant of cycle i of type τ with $t_{0\tau i}$, $\forall \tau$, $\forall i$ $t_{0\tau i} \in \left[0, T_f - D_\tau\right]$ (in the following I shall indicate the generic set of cycle indices).

For each cycle of each type τ, the binary variables $\eta_{\tau i} \in \{0, 1\}$ are introduced with the following meaning:

$$\begin{array}{ll} \eta_{\tau i} = 1 & \text{if cycle } i \text{ of type } \tau \text{ is activated;} \\ \eta_{\tau i} = 0 & \text{otherwise.} \end{array}$$

For each cycle i of type τ, the function of time $w_{\tau i}(t)$ is defined as follows:

$$\forall t \in [t_{0\tau i}, t_{0\tau i} + D_\tau] \quad w_{\tau i}(t) = w_\tau\left(t - t_{0\tau i}\right) \tag{1a}$$

$$\forall t \notin [t_{0\tau i}, t_{0\tau i} + D_\tau] \quad w_{\tau i}(t) = 0 \tag{1b}$$

More precisely, this means that each $t_{0\tau i} \in \left[0, T_f - D_\tau\right]$ generates a specific $w_{\tau i}(t)$, belonging to the set of functions with compact support (such that $\forall t \in \left[0, T_f\right]$ $w_{\tau i}(t) \geq 0$ and $\forall t \notin \left[0, T_f\right]$ $w_{\tau i}(t) = 0$). In the following, only their restrictions to the intervals $[0, T_f]$ will be considered.

The optimization task, in a normalized form, can be expressed as follows:

$$\max_{\eta_{\tau i}, t_{0\tau i} \in \left[0, T_f - D_\tau\right]} \frac{\sum\limits_{\substack{\tau \in T \\ i \in I}} \int\limits_0^{T_f} \eta_{\tau i} w_{\tau i}(t) dt}{\int\limits_0^{T_f} w(t) dt} \tag{2}$$

Here, $\int\limits_0^{T_f} w(t) dt = E$ represents the total energy available, while $\int\limits_0^{D_f} w_\tau(t) dt = E_\tau$ the energy requested by each cycle of type τ (therefore each integral $\int\limits_0^{T_f} \eta_{\tau i} w_{\tau i}(t) dt$ appearing in (2) may simply be substituted with $\eta_{\tau i} E_\tau$).

It is assumed, without any loss of generality, that

$$\forall \tau \in T, \forall i,j \in I/i < j \quad [t_{0\tau i}, t_{0\tau i} + D_\tau] \cap \left[t_{0\tau j}, t_{0\tau j} + D_\tau\right] = \emptyset \tag{3}$$

Were parallel processes (for the same cycle types) indeed to be considered, it would be sufficient to extend the set T appropriately.

For any selection $\left(\underline{t}_{011}, \ldots, \underline{t}_{0\tau i}, \ldots, \underline{t}_{0|T||I|}\right)$, where $|T|$ and $|I|$ represent the cardinalities of T and I respectively, the following conditions are imposed upon the corresponding functions of time $w_{\tau i}(t)$:

$$\forall t \in \left[0, T_f\right] \quad \sum_{\substack{\tau \in T \\ i \in I}} \eta_{\tau i} w_{\tau i}(t) \le w(t) \tag{4}$$

If a minimum and a maximum limit are specified on the number of cycles, the following constraints are added: $\forall \tau \in T\ \underline{N}_\tau \le \sum_{i \in I} \eta_{\tau i} \le \overline{N}_\tau$. (If a proper set of indices I_τ were defined for each cycle type, all variables $\eta_{\tau i}$ outside the corresponding index ranges could be eliminated from the model, together with the previous upper limit conditions).

The continuous-time model outlined above, although quite simple to formulate, is extremely difficult to solve by an exact approach. A very simple time-indexed reformulation is therefore put forward hereinafter, in order to provide approximate solutions, useful in practice. To this purpose, a discretization of the entire period $[0, T_f]$ (from now on, it is assumed $T_f \in \mathbb{N}$) is carried out, by utilizing an appropriate time unit, i.e.: $\left[0, T_f\right] = [0, 1] \cup \cdots \cup [\nu, \nu + 1] \cup \cdots \cup \left[T_f - 1, T_f\right]$. The power function associated with each corresponding sub-interval $[\nu, \nu + 1]$ is now assumed to be constant. This gives rise to an approximating step-function, whose values W_ν are defined as follows:

$$\forall \nu / 0 \le \nu \le T_f - 1 \quad W_\nu = \mathop{min}\limits_{t \in [\nu, \nu+1]} w(t) \tag{5}$$

Analogously, the activity period associated with each cycle type is discretized. To this purpose, each duration D_τ is substituted with a new one, consisting of the shortest integer interval $\overline{D}_\tau$, in terms the above mentioned time unit, containing D_τ (i.e. $\overline{D}_\tau = \lceil D_\tau \rceil$). The sub-intervals $[0, 1], \ldots, [\gamma, \gamma + 1], \ldots, \left[\overline{D}_\tau - 1, \overline{D}_\tau\right]$ are subsequently associated to each $\overline{D}_\tau$. Also in this case, for each cycle type, the power consumption, corresponding to each sub-interval $[\gamma, \gamma + 1]$, is assumed to be constant and the function $w_\tau(t)$ is therefore approximated by a step-function, whose values are now expressed as:

$$\forall \tau, \forall \gamma / 0 \le \gamma \le \overline{D}_\tau - 1 \quad W_{\tau\gamma} = \mathop{max}\limits_{t \in [\gamma, \gamma+1]} w_\tau(t) \tag{6}$$

Remark 1 The adopted approximations for the functions $w(t)$ and $w_\tau(t)$ guarantee that every solution of the discretized model is a feasible solution of the time-continuous one.

For each cycle type τ, the time limit $T_{f\tau} = T_f - \overline{D}_\tau$ is stated. It represents the maximum time breakpoint at which such a cycle type can be activated, in order to be entirely executed within the interval $[0, T_f]$. The binary variables $\chi_{\tau\nu} \in \{0, 1\}$ are then defined, with the following meaning:

$\chi_{\tau\nu} = 1$ if a cycle of type τ is activated at instant ν, such that $0 \leq \nu \leq T_{f\tau}$;
$\chi_{\tau\nu} = 0$ otherwise.

A basic formulation of the approximated MILP model reads as follows. Firstly, oobjective function (2) is transformed into:

$$max \sum_{\substack{\tau \in T \\ \nu \leq T_{f\tau}}} \frac{E_\tau}{E} \chi_{\tau\nu} \tag{7}$$

The constraints below are introduced:

$$\forall \tau, \forall \nu / 0 \leq \nu \leq T_{f\tau}, \forall \gamma / \nu \leq \gamma \leq \nu + \overline{D}_\tau - 1$$

$$u_{\tau\gamma\nu} = W_{\tau\gamma} \chi_{\tau\nu} \tag{8}$$

$$\forall \gamma / 0 \leq \gamma \leq T_f - 1 \quad \sum_{\substack{\tau \in T \\ \gamma - D_\tau + 1 \leq \nu \leq \gamma}} u_{\tau\gamma\nu} \leq W_\gamma \tag{9}$$

$$\forall \tau \in T, \forall \nu / 0 \leq \nu \leq T_{f\tau} \quad \chi_{\tau\nu} + \sum_{\substack{\nu' \geq \nu \\ \nu' \leq \nu + D_\tau - 1}} \chi_{\tau\nu'} \leq 1 \tag{10}$$

$$\forall \tau \in T \quad \underline{N}_\tau \leq \sum_{\nu \leq T_{f\tau}} \chi_{\tau\nu} \leq \overline{N}_\tau \tag{11}$$

The variables $u_{\tau\gamma\nu}$ (defined a priori as continuous) express, through Eq. (8), the power consumption associated with a cycle of type τ, during the sub-interval $[\nu + \gamma, \nu + \gamma + 1]$, if activated at instant ν (in such a case the power consumption equals $W_{\tau\gamma}$). If no cycle of type τ is activated at instant ν, the relative variables $u_{\tau\gamma\nu}$ are zero.

Inequalities (8) and (9) state that during each time sub-interval $[\nu, \nu + 1]$ the power request cannot exceed what is available. Conditions (10) prevent the (total or partial) simultaneity of two (or more) cycles of the same type. The minimum and maximum limits for each cycle type are respected in virtue of inequalities (11).

As a first consideration, conditions (8) and (9) could be rewritten in a single one, getting rid of the variables $u_{\tau\gamma\nu}$, i.e.:

$$\forall\gamma/0 \leq \gamma \leq T_f - 1 \quad \sum_{\substack{\tau\in T \\ \gamma - D_\tau + 1 \leq \nu \leq \gamma}} W_{\tau\gamma}\chi_{\tau\nu} \leq W_\gamma \tag{12}$$

Remark 2 Once (8) and (9) are substituted with (12), all the variables are of the binary type only (binary integer programming, BIP, model).

Hereinafter, extensions of the basic discretized model shall be outlined, considering firstly the possibility of including additional conditions. Two relevant examples are illustrated. The first refers to the case where the total number of cycles of type $\tau^{'}$ is a multiple of the one of τ''. This is expressed by the following equations:

$$\sum_{\nu\leq T_{f\tau'}} \chi_{\tau'\nu} = R_{\tau'\tau''} \sum_{\nu\leq T_{f\tau''}} \chi_{\tau''\nu} \tag{13}$$

($R_{\tau'\tau''}\in\mathbb{N}$). A second example contemplates the case where the execution of each cycle of type τ'' must be preceded by (at least) $P_{\tau'\tau''}$ cycles of type τ '. It is understood, in particular, that each activated τ''-cycle can always be associated (through an injective function) with a set of $P_{\tau'\tau''}$ preceding $\tau^{'}$-cycles and all these sets are disjoint. The conditions below serve the scope:

$$\forall\nu/\nu \leq T_{f\tau''} \quad P_{\tau'\tau''}\sum_{\nu''\leq\nu} \chi_{\tau''\nu''} \leq \sum_{\nu'\leq\nu - D_{\tau'}} \chi_{\tau'\nu'} \tag{14}$$

As mentioned previously, the approach proposed in this work is extendible to the cases where a number of different resources have to be allowed for. The relevant formulation is briefly reported in the following. The symbolism adopted hitherto is adapted to the extended context, in order to stress the analogies with the basic model. To this purpose, the functions associated with the resources available, whose set is denoted by R, are now simply indicated with $w_r(t)$, where $r \in R$ is the corresponding index. With an obvious meaning of the symbols, the extended version of the basic model is reformulated as follows, keeping inequalities (10) and (11) unaltered (and corresponding to (17) and (18) below), i.e.:

$$max \sum_{\substack{r\in R \\ \tau\in T \\ \nu\leq T_{f\nu}}} \frac{E_{r\tau}}{|R|\, E_r}\chi_{\tau\nu} \tag{15}$$

$$\forall r \in R, \forall\tau\in T, \forall\gamma/0 \leq \gamma \leq T_f - 1 \quad \sum_{\substack{\tau\in T \\ \nu - D_\tau + 1 \leq \gamma \leq \nu}} W_{r\tau\gamma}\chi_{\tau\nu} \leq W_{r\gamma} \tag{16}$$

$$\forall \tau \in T, \forall \nu / 0 \leq \nu \leq T_f - 1 \quad \chi_{\tau\nu} + \sum_{\substack{\nu' \geq \nu \\ \nu' \leq \nu + D_\tau - 1}} \chi_{\tau\nu'} \leq 1 \tag{17}$$

$$\forall \tau \in T \quad \underline{N}_\tau \leq \sum_{\nu \leq T_{f\tau}} \chi_{\tau\nu} \leq \overline{N}_\tau \tag{18}$$

Different versions of objective function (15) could also be conceived, if necessary, introducing proper weights, depending on the relevance of each single resource. Additional conditions such as those represented by constraints (13) and (14) could moreover be introduced.

The basic MILP model, when expressed by (7), (10), (11) and (12), contains:

$O\left(|T|\, T_f\right)$ binary variables $\chi_{\tau\nu}$;
$O\left(|T|\, T_f\right)$ cycle non-simultaneity constraints;
$O\left(2\,|T|\right)$ cycle minimum and maximum number constraints;
$O(T_f)$ power capacity constraints.

In the multiple-resource case, the relative number of capacity constrains becomes:
$O\left(|R|\, T_f\right)$.

Remark 3 Differently from the usual indexed-packing-like formulations for scheduling, in the models presented here, the generation of binary variables depends solely on the time discretization adopted and the total number of cycle types involved.

3 Applications and Computational Results

Time-indexed methods for scheduling problems are well known for their efficiency both in terms of solution quality and computational time. This is essentially due to the fact that their LP-relaxations provide, in general, strong bounds. The corresponding matrix size/density, nonetheless, usually represents a major difficulty and, as a consequence, the computer's memory capacity often becomes the actual stumbling block.

The approach proposed in this chapter is addressed to the previously discussed non-standard scheduling problems, bearing in mind a 'reasonable' limitation for the sizes of the instances to cope with. As a rule of thumb, problems with fewer than 250 sub-intervals (time units) and 35 cycle types, involving three different resources, are expected to be solved quite easily, as well as equivalent instances, in terms of matrix size.

Obviously, from a practical point of view, a large-scale problem could be subdivided into a number of sub-problems, by partitioning the total time period appropriately. Moreover, the author's time-continuous model [9] may be utilized within a heuristic procedure, to refine the approximated solutions obtained with the time-indexed approach. To this purpose, when a single resource is involved, it is opportune to interpret each discretized cycle as if it were composed of a number of components corresponding to successive sub-periods (whose duration is not necessarily integer) having constant consumption (extensions to the multiple resource case can be considered). Precedence constraints, deriving from the solution found through the time-indexed model, are hence imposed. They assume the form $t_{h\tau} - t_{k\zeta} \geq D_{h\tau} + D_{k\zeta}$, where $t_{h\tau}$ and $t_{k\zeta}$ are the (time) coordinates (with respect to the given time-resource reference frame) of the centers of components h and k of cycles τ and ζ respectively, while $D_{h\tau}$ and $D_{k\zeta}$ are the corresponding durations. This way, further cycles can tentatively be added by following an overall *hole-filling* logic [9].

Analogies between some classes of scheduling and packing problems (e.g. [11]) are well known. Applications of the approach proposed in this chapter to two-dimensional rectangular packing, when an MILP-based formulation is adopted (e.g. [31]) are quite straightforward. Similarly to the above mentioned time-precedence constraints, relative positions, derived from the discretized model, indeed, can be imposed with respect to one of the axes, in order to solve the overall packing model. This (heuristic) approach is expected to prove quite advantageous as a support strategy to solve hybrid packing models (e.g. [32]).

Hereinafter, a significant number of tests concerning the class of non-standard scheduling problems discussed in the previous section are reported. They are grouped in the following sets: Basic, A, B, C, D, E, F and G. Additionally, considering the analogies between scheduling and packing problems, a set of two-dimensional rectangular packing instances from literature (Fekete and Shepers, see www.or.deis.unibo.it/research_pages/ORinstances/ORinstances; www.fe.up.pt/esicup) have been taken into account. These are denoted as FS. The Basic test set is considered firstly. All other test sets (except for FS) have been constructed as extensions of the Basic set. These test sets are introduced, in this section, step by step.

All the case studies considered have been solved by utilizing IBM ILOG CPLEX 12.3 [13], supported by a personal computer, equipped with: Core 2 Duo P8600, 2.40 GHz processor; 1.93 GB RAM; MS Windows XP Professional, Service Pack 2.

3.1 Basic Test Set

In all tests of the Basic set, the (electrical) power is chosen (with reference to the formulation of Sect. 2) as the only resource considered, with a constant capacity of 25 (power) units. Fifty types of cycles have been defined. They are reported in Table 1. For each cycle τ, the term $K_{h\tau} \times L_{h\tau}$ is associated with its component h (see above). $K_{h\tau}$ indicates the (constant) consumption and $L_{h\tau}$ the duration of the corresponding sub-period, i.e. the number of sub-intervals covered by component h, expressed in time units. It is understood that the duration of each sub-interval is one

Table 1 Basic cycle characterization by their duration/power consumption

Cycle type	Power consumption (units)	Max. No. of cycles	Cycle type	Power consumption (units)	Max. No. of cycles
1	1,2,1	700	26	2 × 23	100
2	2,1,1	700	27	2 × 10,3 × 13	50
3	1,5,3	300	28	1,2 × 9,7 × 3,1 × 12	50
4	1 × 2,5,7,1	500	29	2 × 27	100
5	1,2,5 × 2,2	200	30	1 × 25,5 × 3,1 × 2	70
6	1,3,4,7,3	150	31	2 × 30	50
7	2,3 × 4,4 × 2	150	32	1 × 30	100
8	2,3 × 2,4 × 2,6,5	100	33	2 × 10,1 × 21	70
9	1,3,5 × 2,7 × 2,5	100	34	2 × 11,0,1 × 19	70
10	1 × 2,2,3,7,9 × 2,1 × 3	100	35	1 × 30,2 × 2	100
11	2 × 3,3 × 8	100	36	3 × 3,1 × 30	70
12	3 × 3,4 × 8	100	37	1 × 30,3 × 3	70
13	5 × 5,11 × 6	30	38	1 × 25,2 × 5,3 × 3	70
14	3 × 5,11 × 6,2 × 4	30	39	1 × 30,7,1 × 3	70
15	4 × 5,13 × 6,2 × 4	30	40	1 × 31,5 × 4	50
16	1 × 5,15 × 3,2 × 9	50	41	2,1 × 34	70
17	1 × 5,15 × 5,2 × 7	30	42	5,1 × 34	70
18	1 × 5,17 × 12	30	43	1 × 33,5 × 2	70
19	1 × 10,5 × 5,1 × 4	70	44	2 × 35,3	50
20	1 × 7,5 × 7,1 × 5	70	45	3 × 15,2 × 15,1 × 6	30
21	1 × 9,5 × 7,1 × 3	70	46	2 × 30,1 × 7	50
22	1 × 9,13 × 7,1 × 3	30	47	2 × 30,3 × 7	30
23	1 × 3,21 × 3,1 × 15	50	48	2 × 30,15,3 × 6	30
24	1 × 7,21 × 3,1 × 11	50	49	1 × 20,3 × 18	50
25	1 × 13,25 × 3,1 × 5	30	50	3 × 9,1 × 30	50

time unit and $L_{h\tau}$ is integer (K_iL_i is the energy requested by component h). In the table, all the terms $K_{h\tau} \times L_{h\tau}$ of the same cycle are separated by a comma. If, for a component h of a cycle τ, the relative duration is one single time unit (i.e. $L_{h\tau} = 1$), the term $K_{h\tau} \times L_{h\tau}$ is substituted with $K_{h\tau}$. If a component has zero consumption, the corresponding term is denoted explicitly by $0 \times L_{h\tau}$ (or 0, if the duration of the relative sub-period is one single time unit). An example of the notation adopted is reported here below.

Cycle type τ: $K_{1\tau}, K_{2\tau} \times L_{2\tau}, 0, K_{4\tau} \times L_{4\tau}, K_{5\tau} \times L_{5\tau}$.
This reads, for cycle type τ, as follows:

- the cycle starts with one sub-interval (i.e. one time unit) with a (constant) power consumption of $K_{1\tau}$ units (component/sub-period 1);
- $L_{2\tau}$ sub-intervals (i.e. $L_{2\tau}>1$ time units) follow, with a (constant) power consumption of $K_{2\tau}$ units (component/sub-period 2);

- one sub-interval (i.e. one time unit) follows with a (constant) zero power consumption (component/sub-period 3);
- $L_{4\tau}$ sub-intervals (i.e. $L_{4\tau} > 1$ time units) follow with a (constant) power consumption of $K_{4\tau}$ units (component/sub-period 4);
- $L_{5\tau}$ sub-intervals (i.e. $L_{5\tau} > 1$ time units) follow with a (constant) power consumption of $K_{5\tau}$ units (component/sub-period 5);
- the cycle total duration is $1 + L_{2\tau} + 1 + L_{4\tau} + L_{5\tau}$ time units.

The 50 types of cycles reported in Table 1 are utilized in test sets A, B, C, D, E and G. In test set G, where three generic resources are considered, the electrical power is replaced by the first (generic) resource. In this case, the consumptions (per cycle type) appearing in Table 1 are interpreted in terms of the first generic resource units. This table also reports the maximum number of cycles admissible and these limits shall hold for all test sets from A to G (including F).

Figures 2, 3, 4, 5, and 6 provide a graphical representation of the cycle types considered. Each figure includes (in sequence) ten cycle types (some have been shifted to the right, in order to make the picture clearer).

The Basic test set consists of 25 instances, corresponding to a total time elapse of 100 units (i.e. 100 sub-intervals). Table 2 reports their sequential number in the first column. The second column indicates, for each test, the cycle types (from Table 1) that are available. The third column of the table shows the minimum number of cycles requested for each type. The last two columns report the results obtained, in terms of solution quality and computational effort.

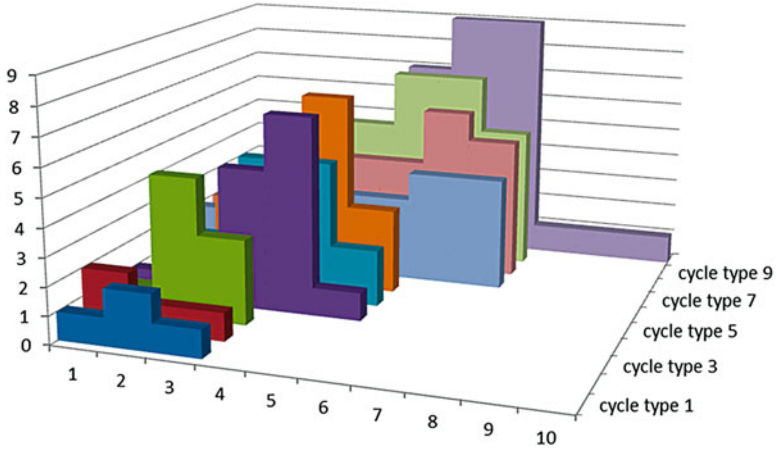


Fig. 2 Graphical representation of cycle types 1–10

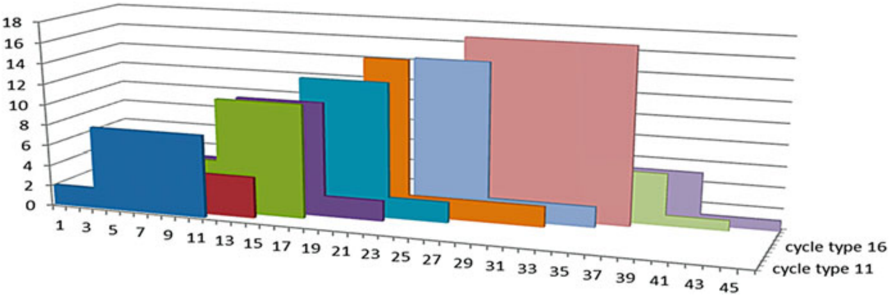


Fig. 3 Graphical representation of cycle types 11–20

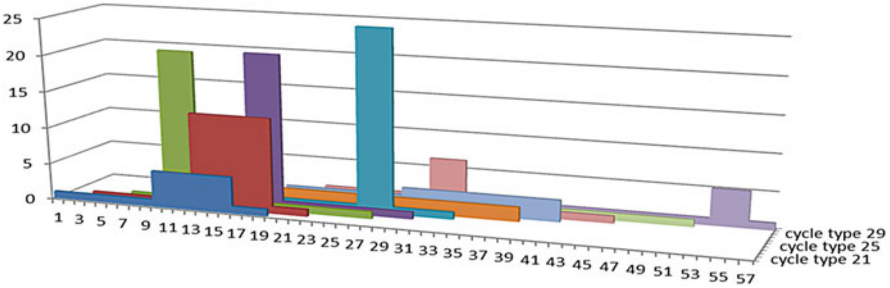


Fig. 4 Graphical representation cycle types 21–30

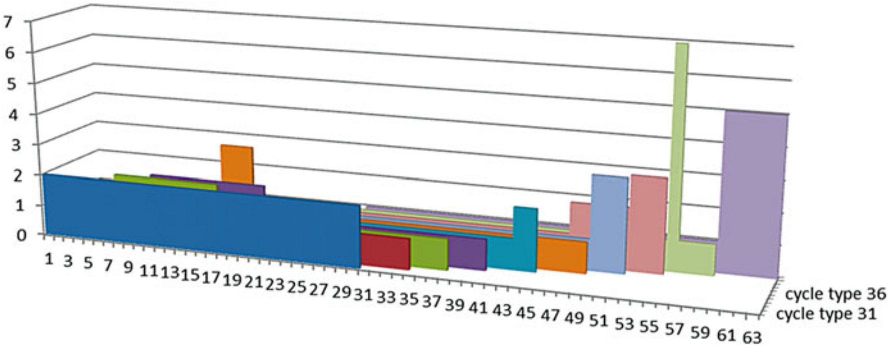


Fig. 5 Graphical representation of cycle types 31–40

A stopping criterion is a time limit of 300 s and it was kept for all the tests (A to G and FS) considered in this work. The last column shows the CPU time (seconds) that was required, for each test, to reach the best solution found, in terms of overall energy exploitation percentage. The same principle concerning the CPU time consumption estimation was adopted for all tests (A to G and FS). The symbol '*' indicates that in some tests, a memory capacity saturation had occurred before the threshold of 300 s was reached. In the Basic test set, some additional conditions (see Sect. 2) were introduced. They are detailed here below.

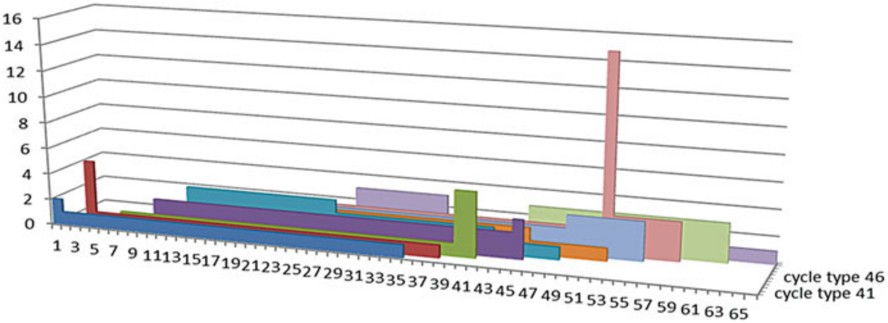


Fig. 6 Graphical representation of cycle types 41–50

Table 2 Basic test set instances and performance results

Test	Cycle type	Min. no. of cycles	Energy exploitation (%)	CPU time (s)
1	1–10	1–10: >0	95.6	298
2	11–20	1–20: >0	84.27	8
3	1–10; 21–30	1–10: > 0; 21–30: >0	95.12	299
4	1–10; 31–40	1–10: > 0; 31–40: > 0	96.03	13*
5	1–10; 41–50	1–10: > 0; 41–50: >0	96.39	4*
6	1–30	1–30: >0	95.48	269
7	1–10; 21–40	1–10: >0; 21–40: >0	95.08	283
8	1–10; 31–50	1–10: >0; 31–50: >0	96.23	44*
9	11–40	11–40: >0	92.0	190
10	11–20; 31–50	11–20: >0; 31–50: >0	96.39	296
11	1–20; 31–40	1–20: >0; 31–40: >0	96.95	117
12	11–30; 41–50	11–30: >0; 41–50: >0	93.42	299
13	11–30; 41–50	41–50: >0	94.09	153
14	1–40	11–20: >0	97.26	114*
15	1–40	11–30: >0	95.72	180*
16	1–40	11–40: >0	96.11	292
17	11–50	11–20: >0	96.31	276
18	11–50	11–30: >0	91.2	143
19	11–50	11–40: >0	89.34	141
20	11–50	21–50: >0	91.59	299
21	1–50	11–20: >0	95.28	190
22	1–50	11–30: >0	94.81	144
23	1–50	11–40: >0	95.56	225
24	1–50	21–50: >0	95.4	287
25	1–50	1–10: >6; 11–20: >0	95.72	299

Test 2:

- total No. of cycles of type 11 = 9 times total No. of cycles of type 12
- total No. of cycles of type 13 = 3 times total No. of cycles of type 14
- total No. of cycles of type 15 = 2 times total No. of cycles of type 16

Test 7:

- each cycle of type 21 must be preceded by 19 cycles of type 1
- each cycle of type 22 must be preceded by 23 cycles of type 2
- each cycle of type 23 must be preceded by 11 cycles of type 3

Test 13:

- total No. of cycles of type 11 = 8 times total No. of cycles of type 48
- total No. of cycles of type 12 = 5 times total No. of cycles of type 49
- total No. of cycles of type 13 = 3 times total No. of cycles of type 50

Test 21:

- each cycle of type 17 must be preceded by seven cycles of type 4
- each cycle of type 18 must be preceded by five cycles of type 5
- each cycle of type 19 must be preceded by five cycles of type 6
- each cycle of type 20 must be preceded by three cycles of type 7

Test 24:

- total No. of cycles of type 1 = 17 times total No. of cycles of type 21
- total No. of cycles of type 2 = 7 times total No. of cycles of type 26
- total No. of cycles of type 3 = 3 times total No. of cycles of type 29

3.2 Test Sets A, B, C, D, E and F

Test set A, similarly to the Basic one, considers a constant power capacity of 25 units. It consists of subsets A1, A2, A3 and A4. Subset A1 coincides with the Basic test set. Subsets A2, A3 and A4 differ from the Basic set only for the total time availability. The following time periods have been considered:

- A1: [0,100] time units
- A2: [0,150] time units
- A3: [0,200] time units
- A4: [0,250] time units

Similarly, subsets B1, B2, B3, B4, ..., F1, F2, F3 and F4 are defined over the same time periods (i.e. [0,100], [0,150], [0,200] and [0,250] time units). Tests B, C, D and E are derived from test set A by changing the power capacity only. Four different power functions, not constant any longer, were hence introduced. They are represented in Figs. 7, 8, 9 and 10 respectively (and reported in detail in the Appendix).

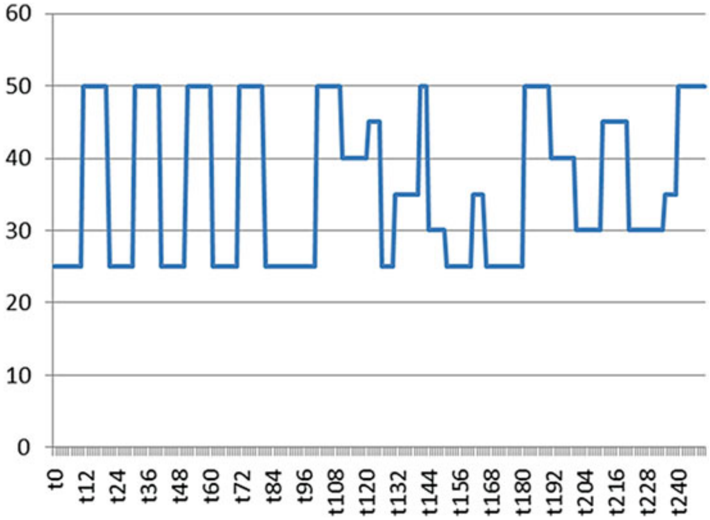


Fig. 7 Test set B power function

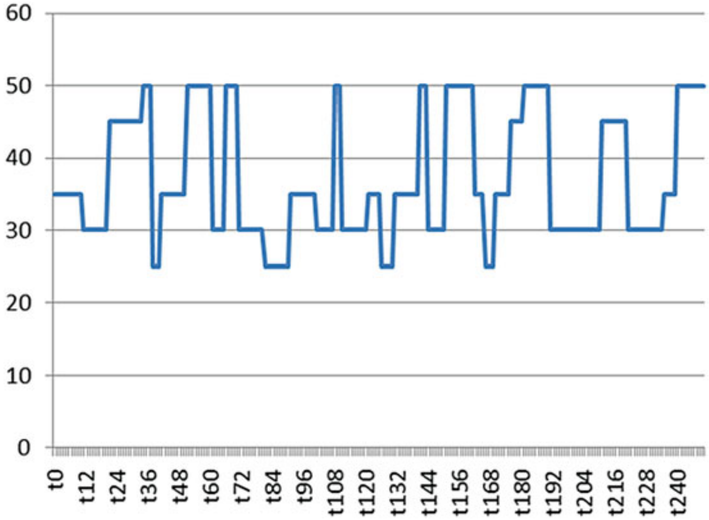


Fig. 8 Test set C power function

For test sets A, B, C, D and E, the constants W_ν (associated with the power step-functions, see (5)) are integer, as well as $K_{h\tau}$ (representing, for each component h of each cycle τ, the power consumption, see Sect. 3.1). Test set F was purposely introduced to consider the case where both W_ν and $K_{h\tau}$ may take, instead, any (non-negative) real values. Test set F was obtained from test set B by adding/subtracting fractional quantities, between 0 and 1, to/from the values corresponding both to the

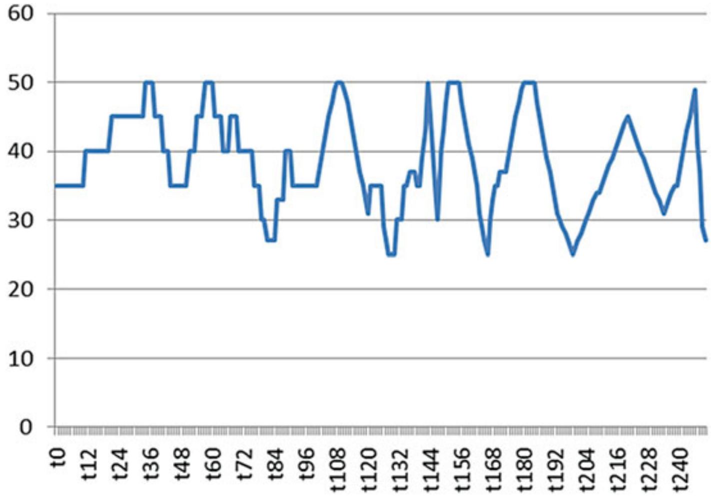


Fig. 9 Test set D power function

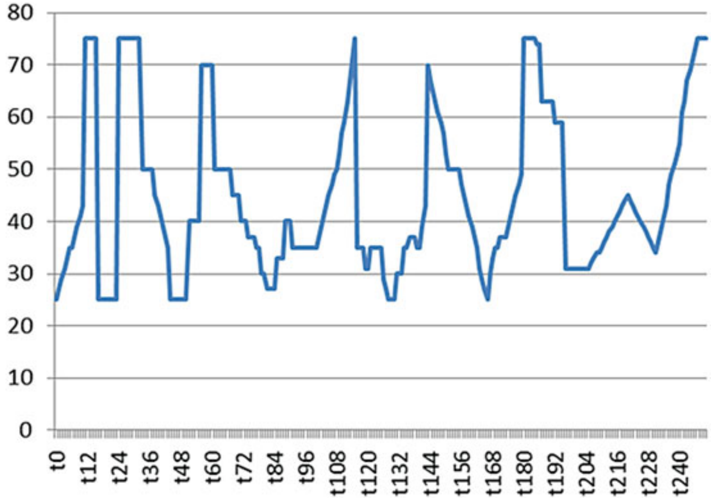


Fig. 10 Test set E power function

power function and the power consumption (see the Appendix for more details). It is understood that for all test sets from A to F, all data relevant to the cycles (as reported in Table 1) are considered, as well as the additional conditions introduced in Sect. 3.1.

The relative computational results, in terms of energy exploitation percentage and CPU time (seconds) are reported in Table 3. There, each test set (i.e. A, B, C, D, E and F) is partitioned into the corresponding subsets of 25 tests each (i.e. A1, A2, A3, A4, . . . , F1, F2, F3 and F4). For each subset, the average of the energy exploitation percentage and CPU time (seconds) is reported.

Table 3 Performance results of test sets A, B, C, D, E and F

Test subset	Energy exploitation (%) average	CPU time (s)	Test subset	Energy exploitation (%) average	CPU time (s)
A1	94.45	195	D1	93.98	204
A2	94.18	245	D2	93.87	259
A3	94.17	233	D3	93.07	196
A4	92.40	221	D4	91.70	219
B1	94.81	225	E1	93.33	245
B2	94.28	253	E2	91.99	254
B3	93.05	242	E3	91.51	194
B4	90.19	159	E4	89.63	217
C1	94.30	210	F1	92.07	201
C2	94.20	254	F2	91.31	220
C3	93.04	216	F3	88.94	186
C4	90.71	190	F4	87.73	168

Table 4 shows, for test subsets A4, B4, C4, D4, E4 and F4, the MILP model matrix dimension, in terms of number of rows, non-zero elements and 0-1 variables after the (MIP) pre-processing carried out by the solver.

3.3 Test Set G

This group of tests contains instances with three different resources each. It was derived from the previous, by substituting the power function with a generic one and adding resources 2 and 3. The power consumption values corresponding to each cycle type (as reported in Table 1) were kept unaltered and associated with resource 1 (no longer necessarily representing the power), 2 and 3 (the same cycle profiles were in fact assumed for the three resources). Three new functions were defined for resource 1, 2 and 3, respectively. These are represented in Fig. 11 (relevant details are reported in the Appendix).

As is gathered, when more than one resource is involved, their total exploitation is generally expected to decrease markedly. The MILP model matrix, on the other hand, increases significantly. The overall performance results obtained for subsets G1, G2, G3 and G4 (corresponding, as in the previous cases, to the time periods [0,100], [0,150], [0,200] and [0,250] time units) are summarized in Table 5.

For test G-16 (of subset G1), no solution was found within the time limit of 300 s. This test has been excluded by the overall results reported in Table 5 (a solution with 75.94 % of total resource exploitation was found in 399 s). Table 6 shows, for test subsets G4, the MILP model matrix dimension, in terms of number of rows, non-zero elements and 0-1 variables after the (MIP) pre-processing carried out by the solver.

Table 4 Test subsets A4–F4 matrix size/density

Test A4–F4	No. of rows	No. of (0-1) variables	No. of non-zero elements
1	2664	2450	29, 010
2	2471	2342	72, 581
3	4741	4733	130, 388
4	4556	4643	161, 840
5	4470	4600	174, 796
6	6968	7094	202, 078
7	7004	6765	379, 560
8	6356	6788	307, 563
9	6433	6816	305, 470
10	6162	6691	350, 098
11	6782	7001	233, 448
12	6328	6784	314, 225
13	6331	6784	315, 448
14	8824	9282	327, 921
15	8834	9282	330, 202
16	8844	9282	332, 390
17	8204	8972	444, 881
18	8214	8972	447, 162
19	8224	8972	449, 350
20	8225	8972	449, 369
21	11,334	11,321	656, 327
22	10,624	11,427	473, 780
23	10,634	11,427	475, 968
24	10,637	11,427	477, 183
25	10,624	11,427	473, 954

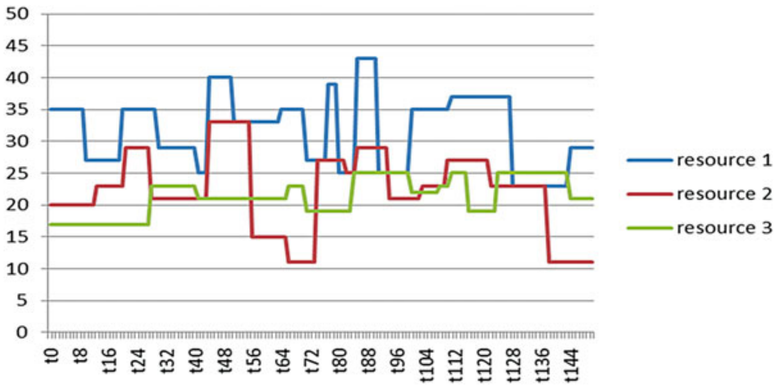


Fig. 11 Test G set resource 1-2-3 functions

Table 5 Test set G performance results

Test subset	Total resource exploitation (%) average	CPU time (s)	Test subset	Total resource exploitation (%) average	CPU time (s)
G1	77.33	255	G3	78.31	246
G2	78.74	257	G4	77.76	172

Table 6 Test subset G4 matrix size/density

Test G4	No. of rows	No. of (0-1) variables	No. of non-zero elements
1	3161	2453	48,080
2	2791	2269	122,564
3	4810	4377	206,294
4	5054	4643	301,253
5	4929	4579	325,577
6	6867	6647	327,873
7	7056	6432	560,047
8	6815	6767	578,457
9	6332	6380	532,318
10	6448	6581	651,557
11	7108	6912	422,764
12	6187	6316	552,270
13	6190	6316	553,809
14	8723	8835	574,177
15	8733	8835	576,100
16	8743	8835	578,288
17	8063	8504	803,308
18	8073	8504	805,231
19	8083	8504	807,419
20	8084	8504	807,506
21	11,129	10,853	1,012,594
22	10,483	10,959	851,183
23	10,493	10,959	853,371
24	10,496	10,959	854,654
25	10,483	10,959	851,715

In addition to test set G, a large-scale instance was considered, in order to probe the applicability of the approach proposed, in terms of problem dimensions (a further in-depth research effort could be devoted to investigate the practical limits that are to be expected). This instance was derived from test G-100 (of subset G4).

The overall time period was quadrupled, giving rise to an overall elapse of 1000 sub-intervals (time units). Each resource profile was extended, by replicating (three times) the corresponding function (defined originally over the period [0,250] time units), as adopted for subset G4. A set of 75 cycle types was taken into account. The first 50 coincided exactly with the ones reported in Table 1, while the remaining 25 were derived from the first 25 (of Table 1). The corresponding resource consumption profiles were replicated, giving rise to cycles of double duration and the original function duplicated (as if the corresponding cycle had been executed twice, sequentially). No minimum-number conditions were imposed on the added cycles. The relative maximum-number limits were the same adopted for cycle types 1–25 (of Table 1).

After the solver (MIP) pre-processing, the resulting instance included 65,798 rows, 68,265 binary variables and 6,211,136 non-zero elements. A solution utilizing 85.24 % of the overall resources available was found in 597 s.

3.4 Tests Extracted from Fekete's and Shepers' Set

The case studies reported in this Section refer to the sets '2D constrained non-guillotine NGCUTFS, file ngcutfs1, provided by Fekete and Shepers (see www.or.deis.unibo.it/research_pages/ORinstances/ORinstances; www.fe.up.pt/esicup) that are expressed in terms of classical two-dimensional knapsack problems, without rotations. These instances were hence interpreted in terms of scheduling problems by considering one of the two axes as the temporal one. It has to be noticed that in these cases, the solutions of the MILP model proposed in this chapter return only the activation times of each job, i.e. only one of the two coordinates of the corresponding rectangle in the packing problem. This means that the solution for the scheduling problem is only a partial solution for the corresponding packing instance (even if its feasibility is guaranteed). It is understood that, also in this case, the activation times can be imposed, in order to provide a partial solution to the corresponding (MILP) packing model.

In this set of tests, the target consisted of maximizing the total value of the items loaded, as proposed by Fekete and Shepers. Table 7 shows the results obtained, pointing out the area exploitation percentage and the CPU time (seconds). Also in this case a limit of 300 CPU seconds was imposed (the symbol '*' indicates, as previously, that a memory capacity saturation occurred before the timeout).

4 Conclusions

This work is inspired by very challenging issues arising in space logistics, where, quite often, the activity requested has to be carried out in extremely limited conditions, both in terms of time and resource capacity. The necessity of optimizing the scheduling of activities, subject to a number of tight constraints, is nonetheless becoming, day after day, ever more demanding in several contexts apart from space.

Table 7 Fekete and Shepers (ngcutfs1) test results

Test	Overall area exploitation (%)	CPU time (s)	Test	Overall area exploitation (%)	CPU time (s)
ngcutfs1_61	98.56	9 (optimal solution proven)	ngcutfs1_91	98.89	29
ngcutfs1_62	96.71	267	ngcutfs1_92	98.32	136
ngcutfs1_63	98.22	105	ngcutfs1_93	98.43	294
ngcutfs1_64	96.63	39*	ngcutfs1_94	97.47	37
ngcutfs1_65	97.43	83	ngcutfs1_95	98.03	19
ngcutfs1_66	96.83	42	ngcutfs1_96	97.96	139
ngcutfs1_67	97.89	15	ngcutfs1_97	98.00	112
ngcutfs1_68	97.97	3*	ngcutfs1_98	98.08	21*
ngcutfs1_69	98.14	83	ngcutfs1_99	97.68	91
ngcutfs1_70	97.57	111	ngcutfs1_100	97.81	80

The problem tackled in this chapter, addresses the cases where the resource capacities, in a given time elapse, are not constant. The activities themselves are characterized by non-constant resource request profiles. The case of a single resource, identified with electrical power, is discussed firstly, pointing out the relevant modelling aspects. An MILP formulation, based on a time-indexed approximation approach is provided. Extensions of the basic model to multiple-resource scenarios are discussed, as well as the introduction of additional conditions. Hints on possible applications of the methodology adopted are put forward and an in-depth experimental analysis is provided. The investigation of ad-hoc computational strategies to solve the relevant models ever more efficiently might represent the objective of future research.

Acknowledgements The author is very grateful to the two referees whose suggestions contributed to the improvement of the original version of this chapter, significantly. Thanks are also due to Jane Evans for her very valuable support in revising the whole manuscript.

Appendix

Test Set F Power Consumption

Cycle type	Power consumption per sub-interval (units)	Max. No. of cycles	Cycle type	Power consumption per sub-interval (units)	Max. No. of cycles
1	0.5,1.4,0.9	700	26	1.9 × 23	100
2	1.3,0.4,0.3	700	27	1.7 × 10, 2.3 × 13	50
3	0.7, 4.6, 2.9	300	28	0.3, 1.6 × 9, 6.1 × 3, 0.5 × 12	50
4	0.3, 0.2, 4.4, 6.9, 0.8	500	29	0.3 × 9, 0.7 × 18	100
5	0.7, 1.7, 4.7, 4.2, 1.3	200	30	0.3 × 9, 0.9 × 16, 4.7 × 3, 0.3 × 2	70
6	0.5, 2.5, 3.7, 6.9, 2.2	150	31	1.1 × 30	50
7	1.7, 2.6, 2.8, 2.9, 2.2, 3.3, 3.1	150	32	0.4 × 9, 0.9 × 21	100
8	1.2, 2.2 × 2, 3.2 × 2, 5.2, 5, 4.2	100	33	1.9 × 10, 0.2 × 21	70
9	0.3, 2.7, 4.5, 4.4, 6.3, 6.6, 4.9	100	34	1.9 × 11, 0.9 × 19	70
10	0.9, 0.8, 1.0, 2.1, 6.2, 8.4, 8.6, 0.5, 0.6, 0.5	100	35	0.7 × 9, 0.9 × 21, 1.8 × 2	100
11	1.3 × 3, 2.6 × 8	100	36	2.8 × 3, 0.9 × 27, 0.2 × 3	70
12	2.7, 2.5, 2.7, 3.9, 3.8, 3.7, 3.0, 3.1, 3.3, 3.5, 3.7	100	37	0.7 × 9, 0.9 × 21, 2.8 × 3	70
13	4.1 × 5, 10.7 × 6	30	38	0.9 × 25, 1.9 × 5, 2.8 × 3	70
14	2.7 × 5, 10.2 × 6, 1.9 × 4	30	39	0.8 × 9, 0.7 × 3, 0.9 × 18, 6.9, 0.2 × 3	70
15	4.9 × 5, 13.7 × 6, 2.1 × 4	30	40	0.3 × 31, 4.1 × 4	50
16	1.3 × 5, 15.5 × 3, 2.7, 2.9 × 8	50	41	1.9, 0.8 × 8, 0.7 × 3, 0.9 × 23	70
17	1.9 × 2, 0.4 × 3, 14.7 × 5, 1.1 × 7	30	42	4.5, 0.9, 0.8 × 7, 0.9 × 26	70
18	0.3 × 5, 16.7 × 12	30	43	0.3 × 3, 0.2 × 5, 0.9 × 19, 0.6 × 6, 4.9 × 2	70
19	0.5 × 10, 4.9 × 5, 07 × 4	70	44	1.7 × 9, 1.8 × 3, 1.9 × 7, 1.7 × 11, 1.3 × 5, 2.9	50

(continued)

Cycle type	Power consumption per sub-interval (units)	Max. No. of cycles	Cycle type	Power consumption per sub-interval (units)	Max. No. of cycles
20	0.1 × 7, 4.5 × 7, 0.7 × 5	70	45	2.8 × 9, 2.9 × 6, 1.9 × 8, 1.7 × 7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1	30
21	0.4 × 9, 4.3 × 7, 0.9 × 3	70	46	1.3, 1.7, 1.9, 1.1, 1.3, 1.2, 1.5, 1.7, 1.9, 1.3 × 8, 1.9 × 9, 1.7 × 4, 0.8 × 7	50
22	0.7 × 9, 12.9 × 7, 0.7 × 3	30	47	1.7, 1.8, 1.7, 1.8, 1.7, 1.8, 1.7, 1.8, 1.7, 1.3 × 5, 1.9 × 5, 1.7 × 2, 1.8, 1.9, 1.8, 1.9, 1.8, 1.9, 1.8, 1.9, 1.7, 2.1, 2.3, 2.5, 2.7, 2.9 × 3	30
23	0.3 × 3, 20.5 × 3, 0.3 × 15	50	48	1.5 × 4, 1.7 × 3, 1.2 × 2, 1.4, 1.5, 1.4, 1.5, 1.4, 1.5, 1.4, 1.5 × 2, 1.7 × 5, 1.9 × 7, 14.9, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6	30
24	0.7 × 7, 20.3 × 3, 0.6 × 11	50	49	0.9 × 5, 0.5 × 4, 0.8 × 10, 0.7, 2.7 × 8, 2.3 × 6, 2.9 × 4	50
25	0.3 × 13, 24.7 × 3, 0.9 × 5	30	50	2.5 × 6, 2.9 × 3, 0.9 × 5, 0.5 × 3, 0.7 × 9, 0.9 × 5, 0.4 × 5, 0.9 × 3	50

Test set B Power Function

25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 45, 45, 45, 45, 45, 25, 25, 25, 25, 25, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 50, 50, 50, 30, 30, 30, 30, 30, 30, 30, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 35, 35, 35, 35, 35, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 35, 35, 35, 35, 35, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50

Test Set C Power Function

35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 50, 50, 50, 50, 25, 25, 25, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 50, 50, 50, 50, 50, 50, 50, 50, 50, 30, 30, 30, 30, 30, 50, 50, 50, 50, 50, 30, 30, 30, 30, 30, 30, 30, 30, 30, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 30, 30, 30, 30, 30, 30, 30, 50, 50, 50, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 35, 35, 35, 35, 35, 25, 25, 25, 25, 25, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 50, 50, 50, 30, 30, 30, 30, 30, 30, 30, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 35, 35, 35, 35, 25, 25, 25, 25, 35, 35, 35, 35, 35, 35, 45, 45, 45, 45, 45, 50, 50, 50, 50, 50, 50, 50, 50, 50, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 35, 35, 35, 35, 35, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50

Test Set D Power Function

35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 45, 50, 50, 50, 50, 45, 45, 45, 40, 40, 40, 35, 35, 35, 35, 35, 35, 35, 40, 40, 40, 45, 45, 45, 50, 50, 50, 50, 45, 45, 45, 40, 40, 40, 45, 45, 45, 40, 40, 40, 40, 40, 40, 35, 35, 35, 30, 30, 27, 27, 27, 27, 33, 33, 33, 40, 40, 40, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 37, 39, 41, 43, 45, 47, 49, 50, 50, 50, 49, 47, 45, 43, 41, 39, 37, 35, 33, 31, 35, 35, 35, 35, 35, 29, 27, 25, 25, 25, 30, 30, 30, 35, 35, 37, 37, 37, 35, 35, 39, 43, 50, 45, 40, 35, 30, 40, 43, 47, 50, 50, 50, 50, 50, 47, 45, 43, 41, 39, 37, 35, 31, 29, 27, 25, 30, 33, 35, 35, 37, 37, 37, 39, 41, 43, 45, 47, 49, 50, 50, 50, 50, 50, 47, 45, 43, 41, 39, 37, 35, 33, 31, 30, 29, 28, 27, 26, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 32, 33, 34, 35, 35, 37, 39, 41, 43, 45, 47, 49, 41, 37, 29, 27

Test Set E Power Function

25, 27, 29, 31, 33, 35, 35, 37, 39, 41, 43, 75, 75, 75, 75, 75, 25, 25, 25, 25, 25, 25, 25, 25, 75, 75, 75, 75, 75, 75, 75, 75, 75, 50, 50, 50, 50, 50, 45, 43, 41, 39, 37, 35, 25, 25, 25, 25, 25, 25, 25, 40, 40, 40, 40, 40, 70, 70, 70, 70, 70, 50, 50, 50, 50, 50, 50, 50, 45, 45, 45, 40, 40, 40, 37, 37, 37, 35, 35, 30, 30, 27, 27, 27, 27, 33, 33, 33, 40, 40, 40, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 37, 39, 41, 43, 45, 47, 49, 50, 53, 57, 59, 63, 67, 71, 75, 35, 35, 35, 31, 31, 35, 35, 35, 35, 35, 29, 27, 25, 25, 25, 30, 30, 30, 35, 35, 37, 37, 37, 35, 35, 39, 43, 70, 67, 65, 63, 61, 59, 57, 53, 50, 50, 50, 50, 50, 47, 45, 43, 41, 39, 37, 35, 31, 29, 27, 25, 30, 33, 35, 35, 37, 37, 37, 39, 41, 43, 45, 47, 49, 75, 75, 75, 75, 75, 74, 74, 63, 63, 63, 63, 63, 59, 59, 59, 59, 31, 31, 31, 31, 31, 31, 31, 31, 31, 31, 32, 33, 34, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 37, 39, 41, 43, 47, 49, 51, 53, 55, 61, 63, 67, 69, 71, 73, 75, 75, 75, 75

Test set F Power Function

25.2, 25.3, 25.5, 25.9, 25.8, 25.7, 25.4, 25.3, 25.5, 25.6, 25.8, 50.9, 50.1, 50.1, 50.2, 50.5, 50.7, 50.8, 50.9, 50.6, 50.5, 25.7, 25.8, 25.6, 25.5, 25.4, 25.7, 25.0, 25.1, 25.4, 25.7, 50.8, 50.9, 50.2, 50.3, 50.4, 50.6, 50.9, 50.6, 50.7, 50.3, 25.5, 25.2, 25.1, 25.4, 25.5, 25.9, 25.6, 25.7, 25.4, 25.7, 50.5, 50.6, 50.7, 50.6, 50.8, 50.9, 50.0, 50.1, 50.2, 50.4, 25.2, 25.5, 25.6, 25.8, 25.4, 25.6, 25.9, 25.0, 25.9, 25.9, 50.4, 50.5, 50.6, 50.7, 50.8, 50.2, 50.4, 50.9, 50.7, 50.8, 25.4, 25.2, 25.4, 25.5, 25.1, 25.3, 25.2, 25.3, 25.5, 25.7, 25.8, 25.9, 25.0, 25.2, 25.5, 25.7, 25.6, 25.3, 25.3, 25.4, 50.7, 50.9, 50.8, 50.0, 50.4, 50.2, 50.1, 50.3, 50.2, 50.5, 40.3, 40.2, 40.7, 40.8, 40.9, 40.2, 40.3, 40.7, 40.6, 40.9, 45.1, 45.3, 45.2, 45.6, 45.5, 25.8, 25.9, 25.9, 25.6, 25.7, 35.5, 35.6, 35.4, 35.3, 35.3, 35.1, 35.9, 35.1, 35.5, 35.3, 50.5, 50.6, 50.7, 30.6, 30.8, 30.9, 30.0, 30.3, 30.2, 30.5, 25.4, 25.6, 25.5, 25.8, 25.7, 25.1, 25.1, 25.2, 25.3, 25.4, 35.8, 35.7, 35.9, 35.4, 35.4, 25.5, 25.6, 25.9, 25.5, 25.9, 25.0, 25.7, 25.8, 25.9, 25.5, 25.6, 25.1, 25.3, 25.4, 25.8, 50.9, 50.4, 50.7, 50.2, 50.7, 50.9, 50.1, 50.3, 50.5, 50.8, 40.6, 40.9, 40.2, 40.6, 40.8, 40.4, 40.6, 40.1, 40.3, 40.2, 30.6, 30.7, 30.4, 30.5, 30.9, 30.0, 30.7, 30.8, 30.1, 30.3, 45.2, 45.4, 45.3, 45.7, 45.6, 45.9, 45.8, 45.1, 45.6, 45.3, 30.9, 30.8, 30.4, 30.1, 30.2, 30.5, 30.4, 30.5, 30.5, 30.7, 30.7, 30.7, 30.9, 30.0, 35.2, 35.4, 35.3, 35.5, 35.1, 50.6, 50.8, 50.5, 50.9, 50.0, 50.2, 50.4, 50.5, 50.6, 50.1, 50.8

Test Set G: Resource 1 Function

35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 29, 25, 25, 25, 40, 40, 40, 40, 40, 40, 40, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 35, 35, 35, 35, 35, 35, 35, 27, 27, 27, 27, 27, 27, 39, 39, 39, 25, 25, 25, 25, 25, 43, 43, 43, 43, 43, 43, 25, 25, 25, 25, 25, 25, 25, 25, 25, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35, 37, 37, 37, 37, 37, 37, 37, 37, 37, 37, 37, 37, 37, 37, 37, 37, 37, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 29, 29, 29, 29, 29, 29, 29

Test Set G: Resource 2 Function

20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 23, 23, 23, 23, 23, 23, 23, 23, 29, 29, 29, 29, 29, 29, 29, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 11, 11, 11, 11, 11, 11, 11, 11, 27, 27, 27, 27, 27, 27, 27, 27, 25, 25, 25, 29, 29, 29, 29, 29, 29, 29, 29, 29, 21, 21, 21, 21, 21, 21, 21, 21, 21, 23, 23, 23, 23, 23, 23, 23, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11

Test Set G: Resource 3 Function

17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 23, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 21, 23, 23, 23, 23, 23, 19, 19, 19, 19, 19, 19, 19, 19, 19, 19, 19, 19, 19, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 22, 22, 22, 22, 22, 22, 22, 22, 23, 23, 23, 25, 25, 25, 25, 25, 19, 19, 19, 19, 19, 19, 19, 19, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 21, 21, 21, 21, 21, 21, 21

References

1. Agnetis, A., Billaut, J.C., Gawiejnowicz, S., Pacciarelli, D., Soukhal, A.: Multiagent Scheduling. Springer, Berlin (2014)
2. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener. Comput. Syst. **28**(5), 755–768 (2012)
3. Błażewicz, J., Ecker, K.H., Pesch, E., Schmidt, G., Weglarz, J.: Handbook on Scheduling, International Handbooks on Information Systems. Springer, Berlin (2007)
4. Brucker, P., Knust, S.: Complex Scheduling. Springer, Berlin (2012)
5. Chen, Z., Chyu, C.: An evolutionary algorithm with multi–local search for the resource-constrained project scheduling problem. Intell. Inf. Manag. **2**, 220–226 (2010)
6. Coelho, J., Vanhoucke, M.: Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers. Eur. J. Oper. Res. **213**(1), 73–82 (2011)
7. Damak, N., Jarboui, B., Siarry, P., Loukil, T.: Differential evolution for solving multi-mode resource-constrained project scheduling problems. Comput. Oper. Res. **36**(9), 2653–2659 (2009)
8. Ha, D.L., Ploix, S., Zamai, E., Jacomino, M.: Control of energy consumption in home automation by resource constraint scheduling. The 15th International Conference on Control System and Computer Science, Bucharest, Romania, May 25–27 (2005)
9. Fasano, G.: Solving Non-standard Packing Problems by Global Optimization and Heuristics. SpringerBriefs in Optimization. Springer Science+Business Media, New York (2014)
10. Gonzalez, F., Ramies, R.D.: Multi-objective optimization of the resource constrained project scheduling problem (RCPSP) a heuristic approach based on the mathematical model. Int. J. Comput. Sci. Appl. **2**(2), 1–13 (2013)
11. Hartmann, S.: Packing problems and project scheduling models: an integrating perspective. J. Oper. Res. Soc. **51**, 1083–1092 (2000)
12. Hartmann, S.: Project Scheduling Under Limited Resources: Models, Methods, and Applications. Lecture Notes in Economics and Mathematical Systems, vol. 478. Springer, Berlin (2013)

13. IBM Corporation: ILOG CPLEX Optimizer: High Performance Mathematical Optimization Engines. IBM Corporation Software Group, New York (2010). WSD14044-USEN-01
14. Jaberi, M.: A multi-objective resource-constrained project-scheduling problem using mean field annealing neural networks. J. Math. Comput. Sci. **9**, 228–239 (2014)
15. Kabra, S., Shaik, M.A., Rathore, A.S.: Multi-period scheduling of a multistage multiproduct bio-pharmaceutical process. Comput. Chem. Eng. **52**, 95–103 (2013)
16. Lee, Y.C., Zomaya, A.Y.: Rescheduling for reliable job completion with the support of clouds. Futur. Gener. Comput. Syst. **26**, 1192–1199 (2010)
17. Li, J., Misener, R., Floudas, C.A.: Continuous-time modeling and global optimization approach for scheduling of crude oil operations. AIChE J. **58**(1), 205–226 (2012)
18. Malakooti, B.: Operations and Production Systems with Multiple Objectives. Wiley, Chichester (2013). ISBN ISBN 978-1-118-58537-5
19. Pinedo, M.L.: Planning and Scheduling in Manufacturing and Services. Springer, New York (2005)
20. Shaik, M.A., Floudas, C.A.: Novel unified modeling approach for short-term scheduling. Ind. Eng. Chem. Res. **48**, 2947–2964 (2009)
21. Zhang, Z., Chen, J.: Solving the spatial scheduling problem: a two-stage approach. Int. J. Prod. Res. **50**(10), 2732–2743 (2012)
22. Bidot, J., Vidal, T., Laborie, P., Beck, J.C.: A theoretic and practical framework for scheduling in a stochastic environment. J. Sched. **12**(3), 315–344 (2009)
23. Mendes, J.J.M., Gonvalces, J.F., Resende, M.G.C.: A random key based genetic algorithm for the resource constrained project scheduling problem. Comput. Oper. Res. **36**, 92–109 (2009)
24. Peteghem, V.V., Vanhoucke, M.: A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. Eur. J. Oper. Res. **201**(2), 409–418 (2010)
25. Xu, J.P., Zeng, Z.Q., Han, B., Lei, X.: A dynamic programming-based particle swarm optimization algorithm for an inventory management problem under uncertainty. Eng. Optim. **45**(7), 851–880 (2013)
26. Yannibelli, V., Amandi, A.: Hybridizing a multi-objective simulated annealing algorithm with a multi-objective evolutionary algorithm to solve a multi-objective project scheduling problem. Expert Syst. Appl. **40**, 2421–2434 (2013)
27. Ziarati, K., Akbari, R., Zeighami, V.: On the performance of bee algorithms for resource-constrained project scheduling problem. Appl. Soft Comput. J. **11**(4), 3720–3733 (2011)
28. van den Akker, J.M., Hurkens, C.A.J., Savelsbergh, M.W.P.: Time-indexed formulations for machine scheduling problems: column generation. Informs J. Comput. **12**(2), 111–124 (2000)
29. Liu, Y., Dong, H., Lohse, N., Petrovic, S., Gindy, N.: An investigation into minimising total energy consumption and total weighted tardiness in job shops. J. Clean. Prod. **65**, 87–96 (2014)
30. Miyamoto, T., Mori, K., Izui, Y., Kitamura, S.: A study of resource constraint project scheduling problem for energy saving. International Conference on System Science and Engineering, ICSSE (2014). doi:10.1109/ICSSE.2014.6887897
31. Pisinger, D., Sigurd, M.: The two-dimensional bin packing problem with variable bin sizes and costs. Discret. Optim. **2**(2), 154–167 (2005)
32. Castro, P.M., Oliveira, J.F.: Scheduling inspired models for two-dimensional packing problems. Eur. J. Oper. Res. **215**, 45–56 (2011)

Packing Problems in Space Solved by CPLEX: An Experimental Analysis

Stefano Gliozzi, Alessandro Castellazzo, and Giorgio Fasano

Abstract Cargo loading of module and vehicles, as well as satellite/spacecraft layout design, notoriously represent very challenging space engineering tasks, deemed to become, day after day, ever more demanding in the perspective of the upcoming exploration adventures. Extremely thought provoking packing optimization problems have to be coped with, in the presence of intricate geometries, operational conditions and, usually, very tight balancing requirements.

A modeling-based (as opposed to a pure algorithmic) approach has been the object of a dedicated long lasting research, carried out by Thales Alenia Space. In this chapter, an extension of the classical container loading problem is considered, allowing for tetris-like items, (convex) non-rectangular domains, (non-prefixed) separation planes and *static* balancing.

The relevant space engineering framework is illustrated firstly, contextualizing its relationship with the more general subject of packing optimization and the topical literature. The problem in question is stated, outlining the underlying mathematical model in use (formulated in terms of Mixed Integer Linear Programming, MILP) and the overall heuristic approach adopted to obtain efficient solutions in practice. An extensive experimental analysis, based on the utilization of CPLEX, as the MILP optimizer, represents the core of this work. Both the MILP model and the related heuristic have been tested on a number of quite demanding case studies, investigating effective MILP strategies up to obtaining satisfactory solutions from a global-optimization point of view. The results shown well pave the way for a promising further dedicated research.

S. Gliozzi (✉)
IBM Italia S.p.A., Rome, Italy
e-mail: Stefano_gliozzi@it.ibm.com

A. Castellazzo
Altran Italia S.p.A.Consultant c/o Thales Alenia Space Italia S.p.A., Turin, Italy
e-mail: alessandro.castellazzo@altran.com

G. Fasano
Exploration and Science, Thales Alenia Space, Turin, Italy
e-mail: giorgio.fasano@thalesaleniaspace.com

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization
and Its Applications 114, DOI 10.1007/978-3-319-41508-6_5

Keywords Orthogonal packing with rotations • Tetris-like items • Balancing • Container loading • Analytical cargo accommodation • Satellite layout • MILP model • Heuristics • CPLEX • Computational results

1 Introduction

This chapter relates to an extended research activity carried out, for more than two decades, in the space engineering context and focused on the loading optimization task, regarding, in particular, the so-called cargo accommodation of vehicles and modules.

This undertaking is well known, in the specialist context, for being very challenging, both at the design phase and at the further stages concerning the spacecraft utilization. As is understood, the exploitation of the overall available volume and/or mass capacity, as effectively as possible, in compliance with the given mission profile, constitutes the ultimate objective.

Tight conditions usually have to be taken account of, in order to satisfy the frequently very demanding attitude control specifications, relevant to the various mission phases (e.g. launch, flight, on-orbit stay and re-entry). Well known examples are represented by the (*static* and *dynamic*) balancing requirements imposed to the whole spacecraft [20]. Mandatory requirements deriving from operational, human factor and safety features, give rise to tricky accommodation rules, including, for instance, the presence of forbidden zones, due to clearance and accessibility necessities, inside the spacecraft, or part of it.

It is furthermore gathered that, although cargo items can often be realistically approximated by single cuboids (i.e. rectangular) parallelepipeds of homogeneous density, this is not always the case, especially when significant dimensions and quite intricate geometrical forms are involved. In addition, the available bags and racks may be characterized by curved surfaces (purposely shaped to exploit the spacecraft structure, as, for instance, when it is cylindrical). Items can be assigned prefixed positions and/or orientations, with respect to the predisposed container (e.g. a bag) or facility (e.g. a rack). Different sectors are often present within the same rack, as well as separation planes inside bags, conceived to make the contained load easier to handle. Demanding balancing conditions are moreover often imposed also at bag/rack level.

A notable example, in terms of cargo accommodation, was represented by the Automated Transfer Vehicle (ATV, see [16]), utilized (between 2008 and 2014) for five successful missions to support the International Space Station (ISS) logistics (see [36]). Due to the very high complexity to cope with, a dedicated methodology was conceived in a previous dedicated research [20]. The heuristic approach thought up for this specific purpose consisted essentially of partitioning the whole cargo accommodation issue into a number of more simple loading sub-problems (at different levels, i.e.: system, rack and bag).

Even more challenging cargo accommodation scenarios are expected for the interplanetary missions of the near future. From a different perspective, a further and by no means less demanding task, in terms of load optimization, concerns the satellite(/spacecraft) layout design (e.g. [49, 50]). In this case, usually, a prefixed number of devices (payloads/equipment) have to be located inside the available volume, often taking advantage of appropriate support planes. Requirements of relative minimum/maximum distance between items frequently have to be respected, in addition to predefined positioning/orientation rules. Specific balancing criteria usually represent the optimization objectives.

From the methodological point of view, in any case, either cargo accommodation or satellite layout jobs entail particular applications of the very general issue, arising in a huge number of technical and scientific applications, of placing geometrical 'objects' inside 'domains' (interpreting these terms even with very abstract meanings). This overall subject, usually referred to as 'packing', is the object of study both by Operations Research (OR, e.g. [51]) and Computational Geometry (e.g. [42]). Packing problems are notoriously well known for being *NP-hard* (e.g. [2, 26]). The topical literature is certainly vast (comprehensive overviews are provided, for instance, by: Cagan et al. [5], Dyckhoff et al. [13], Ibaraki et al. [27]) and covers a number of different classes, depending on the geometrical space considered (e.g. two/three-dimensional), the typology of the objects involved (e.g. cuboids), the domain shape (e.g. a sphere), the presence of additional conditions (e.g. balancing), the availability either of a single or multiple domains, the optimization criteria (if any).

In this overall context, a remarkable effort has traditionally been devoted to studying the problem of loading (orthogonally) 'small boxes' into 'big boxes' (e.g.: [6, 22, 29, 35, 40]). Extended packing scenarios, allowing for more complex (regular and irregular) items and domains, with possible additional conditions (such as balancing), are nonetheless attracting the interest of an increasing number of researchers (e.g.: [1, 3, 4, 8, 15, 17, 23, 32, 38, 43–46, 48, 52]). In order to cope with overall conditions such as balancing (when expressed in terms of actual constraints), a global optimization perspective seems to be preferable to different approaches, essentially consisting of sequential algorithms (e.g. [34]). This leads, in particular, to the adoption of a modeling philosophy, as opposed to a pure algorithmic one (e.g.: [7, 9–11, 14, 24, 31, 33, 39, 41, 46–48]).

Cargo accommodation and satellite layout, due to the quite peculiar geometries involved, as well as specific positioning rules and overall requirements (such as balancing), typically give rise to a number of non-standard packing issues with additional conditions. Approximations are often adopted to make the task tractable. From this perspective, a general problem, quite useful in practice, consists of the orthogonal placement of tetris-like items into a convex domain, with the objective of maximizing the loaded volume (see below). This issue (with further additional conditions, including balancing) occurred, for instance, in the above mentioned ATV project, when treating the packing of items into bags. A modeling approach, based on Mixed Integer Linear Programming (MILP, e.g. [30, 37]) has been studied in previous works [18, 19], including variants and extensions [21]. A brief description of the general problem itself, as well its mathematical formulation, is nonetheless recalled hereinafter for the reader's convenience.

To this purpose, the preliminary definition below is given:

a tetris-like item is a set of rectangular parallelepipeds positioned orthogonally, with respect to an (orthogonal) reference frame. This frame is called 'local' and each parallelepiped is a 'component'

(in this chapter, 'tetris-like item' will be simply referred to as 'item', when no ambiguity occurs, and similarly, 'rectangular parallelepipeds' as 'parallelepipeds'). The general problem is stated as follows.

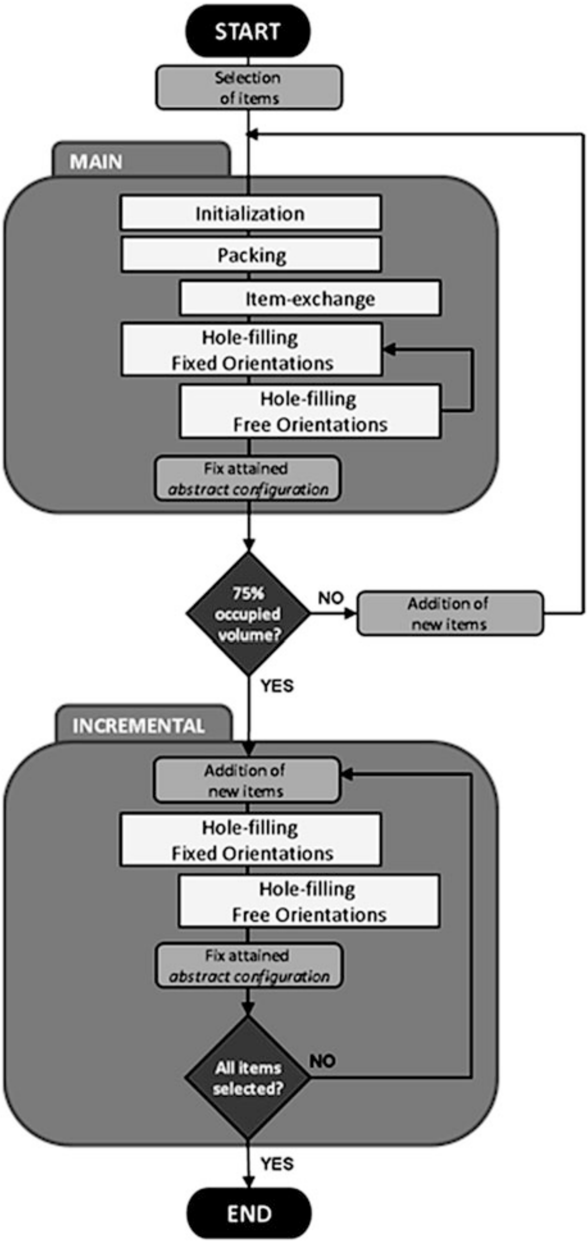
A set I of N items, together with a domain D, consisting of a (bounded) convex polyhedron, is considered. This is associated with a given orthogonal reference frame, indicated in the following as main. Items are picked (from I), in order to maximize the loaded volume, considering the positioning rules here below:

- *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).

This problem, as a matter of fact, is an extension of the classical container loading (e.g. [12]), allowing for tetris-like items (instead of solely box-shaped objects) and a convex (in general non-box shaped) domain. As gathered, the so-called *static* balancing condition is often imposed, requiring that the overall center of mass has to stay within a subdomain $D*$ (assumed to be convex) of D. It is moreover understood, that $D*$ may be given any location (even 'asymmetric') inside the container. This situation occurred, for instance, with the above mentioned ATV cargo accommodation problem. The *static* balancing restriction, at bag level, had, actually, two different statements. In the case of (box-shaped) bags to be loaded into racks, their center of mass was requested to be within a centered (box-shaped) subdomain. The center of mass of (box-shaped) bags that had to be loaded externally, on the rack front panel, had, instead, a different specification. This was requested to stay within a box-shaped subdomain, adjacent to the side of the bag in contact with the rack front panel. This rule was stated in order to reduce the bag unbalancing towards the rack-front outside, as much as possible.

This chapter is an extension of a previous experimental study [25] focused on a MILP-based heuristic procedure, aimed at solving the general packing problem (with possible additional conditions). Section 2 recalls the relevant (MILP) mathematical model, as well as this heuristics briefly, referring the reader to the quoted works for a detailed discussion. Section 3 reports recent advances in the experimental analysis.

Fig. 1 Heuristics overall logic


START
Selection of items
MAIN
Initialization
Packing
Item-exchange
Hole-filling Fixed Orientations
Hole-filling Free Orientations
Fix attained *abstract configuration*
75% occupied volume?
NO
Addition of new items
YES
INCREMENTAL
Addition of new items
Hole-filling Fixed Orientations
Hole-filling Free Orientations
Fix attained *abstract configuration*
All items selected?
NO
YES
END

2 MILP Model and Heuristic Approach

The MILP model relevant to the general packing problem stated in Sect. 1 is outlined hereinafter (referring the reader to [18, 19] for a detailed discussion, also including additional conditions, such as static balancing). It is assumed that the main orthogonal reference frame has origin O with axes w_β, $\beta \in \{1, 2, 3\} = B$ and that the whole domain D is entirely contained inside its first octant. Similarly, each local reference frame, associated with every item, is chosen so that all item components lie within its first octant. Its origin coordinates, with respect to the main reference frame, are denoted by $o_{\beta i}$. The set Ω represents all the orthogonal rotations, admissible for any local reference frame.

The set of components of an item i is denoted by C_i. For each item i, the set E_{hi} of all vertices associated with each of its components h is defined. For each item i and each possible orthogonal orientation $\omega \in \Omega$, the following binary variables are introduced:

$\chi_i \in \{0, 1\}$, with $\chi_i = 1$ if item i is chosen; $\chi_i = 0$ otherwise;
$\vartheta_{\omega i} \in \{0, 1\}$, with $\vartheta_{\omega i} = 1$ if item i is chosen and it has the orthogonal orientation $\omega \in \Omega$; $\vartheta_{\omega i} = 0$ otherwise.

The *orthogonality* conditions can be expressed as follows:

$$\forall i \in I \quad \sum_{\omega \in \Omega} \vartheta_{\omega i} = \chi_i \tag{1}$$

$$\begin{aligned} &\forall \beta \in B, \forall i \in I, \forall h \in C_i, \forall \eta \in E_{hi} \\ &w_{\beta\eta hi} = o_{\beta i} + \sum_{\omega \in \Omega} W_{\omega\beta\eta hi} \vartheta_{\omega i} \end{aligned} \tag{2}$$

Here $w_{\beta\eta hi}$ are the vertex coordinates of component h, with respect to the main reference frame, relative to item i; $W_{\omega\beta\eta hi}$ are the projections on the axes w_β of the coordinate differences between points $\eta \in E_{hi}$ and the origin of the local reference frame, corresponding to orientation ω of item i.

The *domain* conditions are expressed as follows.

$$\begin{aligned} &\forall \beta \in B, \forall i \in I, \forall h \in C_i, \forall \eta \in E_{hi} \\ &w_{\beta\eta hi} = \sum_{\gamma \in V} V_{\beta\gamma} \lambda_{\gamma\eta hi} \end{aligned} \tag{3}$$

$$\begin{aligned} &\forall i \in I, \forall h \in C_i, \forall \eta \in E_{hi} \\ &\sum_{\gamma \in V} \lambda_{\gamma\eta hi} = \chi_i \end{aligned} \tag{4}$$

Here V is the set of vertices delimiting D, $V_{\beta\gamma}$ are their coordinates (with respect to the main reference frame) and $\lambda_{\gamma\eta hi}$ are non-negative variables. These conditions correspond to the well-known necessary and sufficient conditions for a point to belong to a convex domain.

The *non-intersection* conditions are represented by the constraints shown below:

$$\begin{aligned}&\forall\beta \in B, \forall i,j \in I/i<j, \forall h \in C_i, \forall k \in C_j\\&w_{\beta 0hi} - w_{\beta 0kj} \geq \tfrac{1}{2}\sum_{\omega\in\Omega}\left(L_{\omega\beta hi}\vartheta_{\omega i} + L_{\omega\beta kj}\vartheta_{\omega j}\right) - D_\beta\left(1-\sigma^{+}_{\beta hkij}\right)\end{aligned} \tag{5a}$$

$$\begin{aligned}&\forall\beta \in B, \forall i,j \in I/i<j, \forall h \in C_i, \forall k \in C_j\\&w_{\beta 0kj} - w_{\beta 0hi} \geq \tfrac{1}{2}\sum_{\omega\in\Omega}\left(L_{\omega\beta hi}\vartheta_{\omega i} + L_{\omega\beta kj}\vartheta_{\omega j}\right) - D_\beta\left(1-\sigma^{-}_{\beta hkij}\right)\end{aligned} \tag{5b}$$

$$\begin{aligned}&\forall i,j \in I/i<j, \forall h \in C_i, \forall k \in C_j\\&\sum_{\beta\in B}\left(\sigma^{+}_{\beta hkij} + \sigma^{-}_{\beta hkij}\right) \geq \chi_i + \chi_j - 1\end{aligned} \tag{6}$$

Here the constants D_β are the sides of the parallelepiped, of minimum dimensions, containing D; $w_{\beta 0hi}$ and $w_{\beta 0kj}$ are the center coordinates, with respect to the main reference frame, of components h and k of items i and j respectively; $L_{\omega\beta hi}$ and $L_{\omega\beta kj}$ are their side projections on the w_β axes, corresponding to the orientation ω; $\sigma^{+}_{\beta hkij}$ and $\sigma^{-}_{\beta hkij} \in \{0, 1\}$.

The maximization of the loaded volume is expressed as:

$$max \sum_{i\in I, h\in C_i} \frac{V_{hi}}{\sum_{\alpha\in A} L_{\alpha hi}} \left(\sum_{\substack{\beta\,\in\,B,\\ \omega\,\in\,\Omega}} L_{\omega\beta hi}\vartheta_{\omega i}\right) \tag{7}$$

where V_{hi} represents the volume of component h of item i, $L_{\alpha hi}$, $\alpha \in \{1,2,3\} = A$, are the sides of this component (objective function (7) is an efficient reformulation, from the computational point of view, of $max\sum_{i\in I} V_i\chi_i$).

Although, usually, the mathematical model expressed by (1), (2), (3), (4), (5a), (5b), (6), and (7) results in being more efficient than others available in the specialist literature (e.g. [9]), large-scale instances can hardly ever be solved, tout court, by general purpose (MILP) solvers. To this aim, an overall heuristic methodology [19], based on the recursive use of the general MILP model, has been investigated, in addition to previous non-rigorous procedures (also involving further mathematical models, see [18]). Different versions of this overall approach can be defined by introducing specific recursive logic or even MILP solution strategies. One, in particular, is deemed to be the most promising, in accordance with the experimental results available to date [25]. As discussed at a detailed level in the previous works quoted, the overall heuristic approach is based on the following modules: *Initialization*, *Packing*, *Item-exchange*, *Hole-filling*.

The *Initialization* is aimed at providing a starting *abstract configuration*, i.e. a set of relative positions (one for each pair of items) giving rise to a feasible solution in any unbounded domain (see [18, 19]). An ad hoc *LP-relaxation* of the general MILP model is employed for the purpose, (tentatively) including all the N items available and a first approximate solution, 'minimizing' their total overlapping, obtained. The corresponding *abstract configuration* is imposed to the *Packing* module that, through the general MILP model, yields a non-approximate solution maximizing the loaded volume (and rejecting some items if necessary). Both *Item-exchange* and *Hole-Filling* modules (based, in this specific version, on the general MILP model, with subsets of variables time after time fixed) are devoted to the improvement, if possible, of the current *Packing* solution, providing upgraded *abstract configurations*. The specific heuristic procedure here considered consists of two macro-phases, i.e. *main* and *incremental*. The whole process follows the overall logic illustrated in Fig. 1 (see [25] for more details).

3 Experimental Framework and Results

IBM ILOG CPLEX (see [28]) is the MILP optimizer. CPLEX carries out the optimization process by a branch & cut (B&C) algorithm, including several general purpose heuristics. It is also able to perform parallel optimization. Like most of the optimizers available to date, CPLEX has a default strategy for the MILP solution, which is flexible and adaptable to the model characteristics. Its level of sophistication is so advanced that a number of ad hoc optimizer parameters, able to outperform the default mode, can hardly be found. Moreover, the risk of 'over engineering' the setting of the parameters, tuning them to a particular class of instances, rather than to the model intrinsic characteristics, cannot be neglected. Sometimes however, it can be useful to define a specific CPLEX optimization strategy. This holds, in particular, when the solution search is somehow time-boxed, and the proof of optimality is not necessary.

Quite a detailed study on a tailored use of CPLEX, in terms of MILP strategies, aimed at solving both the general MILP model directly and the heuristic procedure of Sect. 2, is reported in the quoted previous work [25]. Hereinafter, a novel and extended experimental analysis is discussed.

Since Version 12.6.1, CPLEX is capable of solving MIP problems using a new Distributed Parallel feature to split the B&C job among different cores on different HW servers. This opens up to different usable parallelization strategies. Moreover, the parallelization works:

(a) even over a TCP/IP network;
(b) without the need of any other Software installed;
(c) over heterogeneous cores;
(d) in a core-fault tolerant mode.

This, in perspective, makes the parallel optimization a ‘low cost’ alternative for difficult problems, allowing the consumption of idle space from several cores on a network (see [28]).

This section considers first 100 non-trivial test cases for the classical container loading problem, with additional balancing conditions. They are extracted from: ‘Three Dimensional Cutting and Packing Data Sets—THPACK 1–7 BR’ (Bischoff and Ratcliff: http://www.euro-online.org/web/ewg/25/esicup-euro-special-interest-group-on-cutting-and-packing). As is known, this test-bed consists of seven sets of 100 test cases each. Among these, a subset of those with an available number of items between 100 and 200 were selected (a list of the selected test cases is reported in Appendix).

Different test case configurations, in term of position of the domain center of mass, have been analyzed: in 80 tests it was coincident with respect to the geometric center of the domain and in 20 tests it was shifted close to one side (25 % of the corresponding domain edge). The items mass has been varied according to two probability distributions: the Gaussian and the Gamma.

The test campaign was performed using IBM CPLEX 12.6.2 as the optimizing engine, and IBM EasyModeler as the model generator. More precisely, the MILP solver available within CPLEX, statically linked to the C++ code generated by EasyModeler, was adopted, using the open source Coin-OR OSI 0.105.3 library as the interface between EasyModeler and the optimizer. The following computational supports were, moreover, utilized:

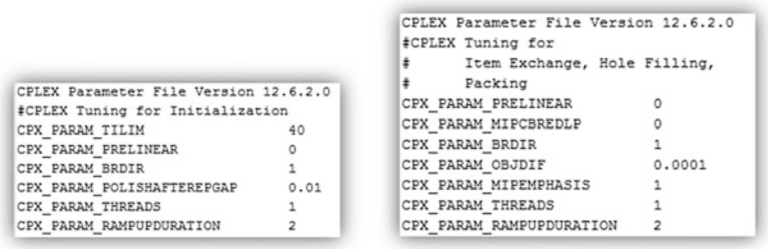
- platform: Lenovo Thinkpad W520 Laptop. with an Intel(R) Core (TM) i7-2620 M at 2.7 GHz clock frequency (two real core seen as four with Intel Hyperthreading) and 8 GB Ram available;
- operating system: Windows(R) 7 Professional OS.

Preliminary experiments have shown that the usage of the distributed parallel feature, using the cited heuristic, was advantageous, with respect to the default CPLEX single CPU parallel behavior, which utilizes a multi-thread parallel B&C implementation, even when using the cores of a single CPU. Therefore, all the test cases described in this section have been solved using the CPLEX distributed parallel capabilities. We have also carried out a minimal tuning of the CPLEX run time parameters. The parameter tuning (Fig. 2) has been kept to a minimum, in order to:

i. minimize the risk of over engineering strategy;
ii. to make it easier to capture the added value of the Parallel feature.

The distributed parallel algorithm is usually performed in two phases. During a first ramp-up phase, the B&C is executed by each core, using a slightly different strategy. After a certain (parametric) time of execution, the most promising strategy among those applied, is selected, and executed in parallel by all the cores.

The CPLEX parameters are the same as those used in the previous work [25], and only two parameters, related to the distributed parallel feature have been added:



```
CPLEX Parameter File Version 12.6.2.0
#CPLEX Tuning for Initialization
CPX_PARAM_TILIM                 40
CPX_PARAM_PRELINEAR             0
CPX_PARAM_BRDIR                 1
CPX_PARAM_POLISHAFTEREPGAP      0.01
CPX_PARAM_THREADS               1
CPX_PARAM_RAMPUPDURATION        2
```

```
CPLEX Parameter File Version 12.6.2.0
#CPLEX Tuning for
#       Item Exchange, Hole Filling,
#       Packing
CPX_PARAM_PRELINEAR           0
CPX_PARAM_MIPCBREDLP          0
CPX_PARAM_BRDIR               1
CPX_PARAM_OBJDIF              0.0001
CPX_PARAM_MIPEMPHASIS         1
CPX_PARAM_THREADS             1
CPX_PARAM_RAMPUPDURATION      2
```

Fig. 2 CPLEX parameters selection for the heuristic solution

- CPX_PARAM_THREADS set to 1 allocates a single thread optimization for each available core;
- CPX_PARAM_RAMPUPDURATION set to 2, implements a peculiar way of using the parallelism, performing only the so called ‘ramp-up’ phase of the Parallel B&C;
- the number of parallel processes was set to 3, leaving one of the four cores to the master process.

The whole selected test cases were solved using the heuristics of Sect. 2, with a maximum time limit of 1 h. The relevant results are summarized in Tables 1, 2, 3, and 4. Figures 3, 4, 5, and 6 refer to case studies GC_18, NC_8, GS_7 and NS_1.

The 37 instances with Gamma-distributed items mass and center of mass in the geometric center of the domain, had an average load factor of 83.98 % showing also a very consistent behavior (the Load Factor Standard Deviation is 2.3, and the minimum and maximum Load Factor are 77.35 and 88.14 % respectively). It also has to be noted that none of these instances was stopped by the time limit. As a comparison with the other tests described below, these look easier to solve.

The 43 instances with Gaussian-distributed item mass and center of mass in the geometric center of the domain, had an average load factor of 81.40 % showing somehow a less consistent behavior (the Load Factor Standard Deviation is slightly higher: 4.17, and the minimum and maximum Load Factor, with values of 68.51 and 89.79 % respectively, represent a broader range). 22 out of 43 instances hit the time limit.

The ten instances with Gamma-distributed item mass and center of mass shifted, had an average load factor of 55.74 % (the Load Factor Standard Deviation is 5.0, and the minimum and maximum Load Factor are 46.79 and 61.92 % respectively). All the instances were stopped by the time limit.

Table 1 Gamma-distributed item mass and center of mass in the geometric center of the domain

Test name	Instance items	Loaded items	Loading factor (%)	Execution time
GC_1	102	84	81.44	00:39:36
GC_2	141	112	86.78	00:44:51
GC_3	116	82	81.40	00:10:01
GC_4	154	106	83.50	00:08:15
GC_5	120	72	81.66	00:04:18
GC_6	196	118	83.12	00:09:35
GC_7	187	114	77.35	00:41:18
GC_8	188	128	82.80	00:27:21
GC_9	174	117	83.01	00:12:39
GC_10	115	81	85.85	00:05:38
GC_11	135	90	83.21	00:08:08
GC_12	194	124	81.61	00:30:50
GC_13	100	72	88.14	00:06:04
GC_14	140	100	82.28	00:31:00
GC_15	169	114	80.24	00:29:33
GC_16	119	83	85.07	00:15:38
GC_17	151	95	83.50	00:14:33
GC_18	156	104	87.55	00:10:50
GC_19	123	97	87.83	00:14:03
GC_20	108	79	87.12	00:06:53
GC_21	138	103	84.74	00:14:11
GC_22	142	109	85.17	00:20:34
GC_23	122	83	84.35	00:11:38
GC_24	118	75	83.60	00:06:57
GC_25	171	126	82.72	00:27:14
GC_26	113	84	85.53	00:06:58
GC_27	155	105	84.23	00:24:53
GC_28	138	97	82.73	00:15:03
GC_29	108	82	87.29	00:07:00
GC_30	153	108	84.11	00:11:44
GC_31	116	76	82.15	00:06:18
GC_32	137	102	86.67	00:09:08
GC_33	100	74	85.48	00:05:53
GC_34	140	102	83.56	00:21:17
GC_35	127	81	85.02	00:08:02
GC_36	154	111	82.09	00:20:20
GC_37	145	99	84.30	00:10:39

Table 2 Gaussian-distributed item mass and center of mass in the geometric center of the domain

Test name	Instance items	Loaded items	Loading factor (%)	Execution time
NC_1	112	72	76.73	00:47:40
NC_2	197	121	86.16	01:04:10
NC_3	126	81	80.43	00:51:57
NC_4	155	105	80.46	00:34:22
NC_5	109	76	86.09	00:21:23
NC_6	166	103	82.25	00:46:22
NC_7	174	103	77.76	00:43:29
NC_8	112	92	89.79	00:38:07
NC_9	155	97	81.44	00:36:35
NC_10	141	84	77.90	00:18:36
NC_11	181	113	82.71	00:48:53
NC_12	114	81	87.03	00:47:08
NC_13	188	110	77.58	01:02:15
NC_14	140	86	68.51	01:00:12
NC_15	199	125	82.00	01:00:11
NC_16	161	101	84.69	01:03:41
NC_17	119	78	84.30	00:57:49
NC_18	110	77	88.20	00:43:51
NC_19	158	104	81.11	01:00:52
NC_20	117	71	77.57	01:01:51
NC_21	153	98	80.54	01:00:46
NC_22	178	112	73.25	01:01:00
NC_23	104	72	81.33	00:37:08
NC_24	115	81	86.36	00:55:00
NC_25	160	98	85.18	01:00:42
NC_26	108	74	81.86	01:04:04
NC_27	126	87	81.89	00:54:43
NC_28	129	94	82.55	01:00:05
NC_29	144	97	80.41	01:01:26
NC_30	122	78	80.53	01:03:55
NC_31	113	82	84.34	01:02:41
NC_32	152	89	75.98	01:03:36
NC_33	131	87	84.46	01:04:23
NC_34	156	89	81.65	01:01:51
NC_35	103	75	84.03	00:41:29
NC_36	129	99	85.03	01:00:09
NC_37	143	89	80.93	00:59:08
NC_38	172	90	72.45	01:03:30
NC_39	133	85	79.87	01:04:30
NC_40	149	77	76.89	01:02:40
NC_41	100	71	82.57	00:38:23
NC_42	129	76	81.66	01:00:39
NC_43	142	102	83.86	00:52:24

Table 3 Gamma-distributed item mass and center of mass shifted

Test name	Instance items	Loaded items	Loading factor (%)	Execution time
GS_1	132	34	52.01	01:00:14
GS_2	122	60	57.61	01:00:35
GS_3	143	32	46.79	01:02:41
GS_4	126	64	60.08	01:00:14
GS_5	124	37	49.09	01:02:27
GS_6	141	45	53.44	01:00:15
GS_7	125	58	61.92	01:01:15
GS_8	128	41	59.62	01:02:50
GS_9	144	58	61.21	01:01:54
GS_10	143	52	54.66	01:04:10

Table 4 Gaussian-distributed item mass and center of mass shifted

Test name	Instance items	Loaded items	Loading factor (%)	Execution time
NS_1	138	52	55.79	01:03:14
NS_2	147	56	52.08	01:04:00
NS_3	128	42	52.74	01:00:45
NS_4	138	30	45.98	01:02:02
NS_5	138	51	53.12	01:00:37
NS_6	127	28	48.59	01:00:36
NS_7	149	39	53.78	01:01:07
NS_8	129	58	52.09	01:02:37
NS_9	135	57	46.88	01:01:05
NS_10	126	40	52.51	01:02:25

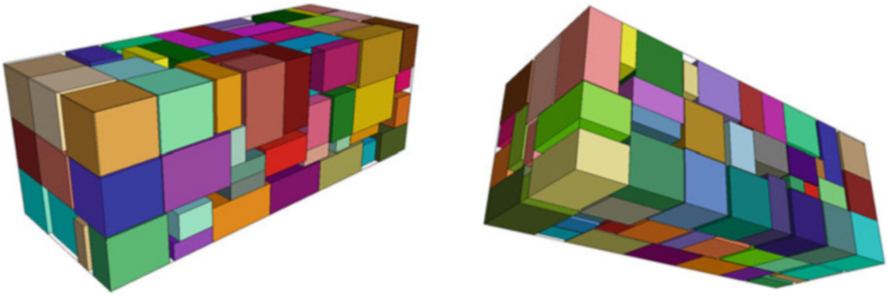


Fig. 3 Solution of test GC_18

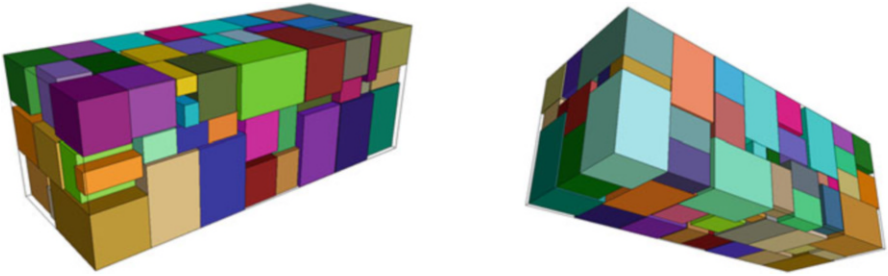


Fig. 4 Solution of test NC_8

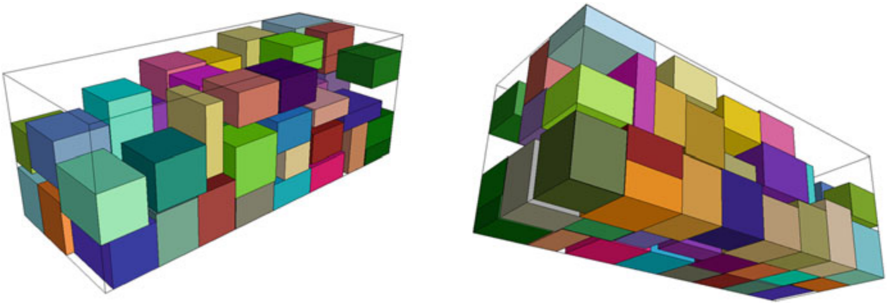


Fig. 5 Solution of test GS_7

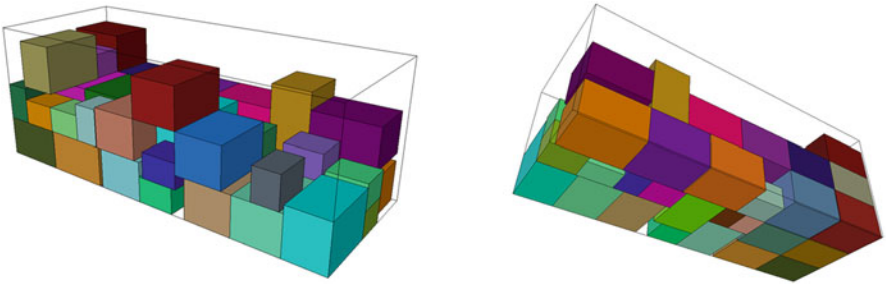
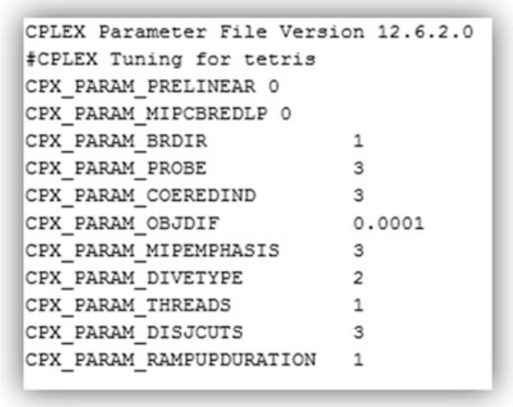


Fig. 6 Solution of test NS_1

The ten instances with Gaussian-distributed item mass and center of mass shifted, had an average load factor of 51.35 % showing also a very consistent behavior (the Load Factor Standard Deviation is 3.0, and the minimum and maximum Load Factor are 45.98 and 55.79 % respectively). All the instances were stopped by the time limit (Table 4).



```
CPLEX Parameter File Version 12.6.2.0
#CPLEX Tuning for tetris
CPX_PARAM_PRELINEAR 0
CPX_PARAM_MIPCBREDLP 0
CPX_PARAM_BRDIR           1
CPX_PARAM_PROBE           3
CPX_PARAM_COEREDIND       3
CPX_PARAM_OBJDIF          0.0001
CPX_PARAM_MIPEMPHASIS     3
CPX_PARAM_DIVETYPE        2
CPX_PARAM_THREADS         1
CPX_PARAM_DISJCUTS        3
CPX_PARAM_RAMPUPDURATION  1
```

Fig. 7 Tetris CPLEX parameters

A further set of ten ‘fabricated’ case studies involving tetris-like items has been considered. The overall problem still consists of maximizing the occupied volume of a convex domain in the presence of balancing conditions. Table 5 reports the number and the typology of objects considered for each test case. Table 6 summarizes the relevant results, in terms of solutions obtained and computational performances. Figures 8 and 9 refer to case studies T_3 and T_4 respectively.

These cases were solved using a direct solution process over the model formulation, without splitting the instance with a heuristic process. For these tests, the same Software configuration of EasyModeler and IBM CPLEX described above was used. The HW platform was however different and slightly faster: a Lenovo Thinkpad T440 Laptop. with an Intel(R) Core (TM) i5-4300U at 1.9 GHz clock frequency (4 real core) and 8 GB Ram available.

A different strategy has been specified for the Tetris case (see Fig. 7). In particular the usage of disjunctive cuts and probing has been enforced and a Branch & Cut general strategy aimed at increasing the lower bound was chosen. The distributed parallel was run without any ramp up, allowing all the parallel processes to operate with the same strategy on the B&C Tree.

As with the previous tests, three cores were used for the parallel search while one core was used for the master, and a 1-h time limit was set.

All test were solved within the 1-h time limit. A further run was done for test T_3, relaxing the time limit with the aim of understanding how harder the instance was. In fact it was solved after just over 90 min while the sum of the solution times of the other nine instances was merely 35 min.

Table 5 Instance configuration according to the item typologies

Test number	Number of objects	Number of object per type									Total number of components
		"L" shape	"H" shape	"T" shape	"C" shape	Triangular prism	Cylinder	Sphere	Tetrahedron	Cuboid	
T_1	7	2	–	–	1	–	–	1	–	3	19
T_2	10	2	1	4	1	–	1	–	–	1	27
T_3	7	–	1	1	2	2	1	–	–	–	34
T_4	8	2	1	1	2	–	–	–	–	2	15
T_5	7	1	1	2	2	–	–	–	–	1	15
T_6	6	2	–	1	–	2	–	–	–	1	21
T_7	10	4	1	–	–	–	–	–	1	4	22
T_8	22	2	1	2	1	–	–	–	–	16	30
T_9	7	–	–	–	–	–	7	–	–	–	49
T_10	8	4	1	–	–	–	–	1	1	1	27

Table 6 Summary of results for tetris-like item test set

Test number	Objects	Parts	Load factor (%)	Constraints	Variables	Integers	Time (s)	Nodes
T_1	7	9	79.29	1166	1021	894	1	0
T_2	10	9	76.13	2815	2464	2267	77	1112
T_3*	7	9	71.90	3732	3232	3006	5429	54,850
T_4	8	3	92.86	1090	1021	900	71	3593
T_5	7	3	95.25	958	913	798	411	43,027
T_6	6	8	66.76	1790	1876	1200	59	3091
T_7	10	7	75.23	1741	1537	1386	205	68,329
T_8	22	3	78.81	3967	2989	2790	691	36,473
T_9	7	7	53.90	8392	7834	6342	551	8810
T_10	8	9	75.05	2545	2248	2058	9	0

*Solved using a 3-h time limit

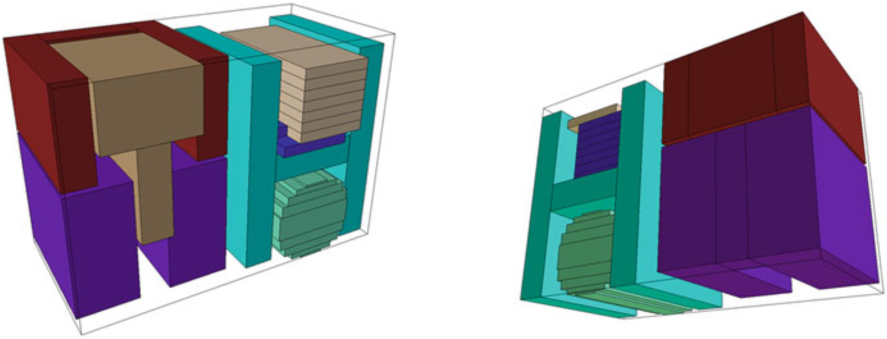


Fig. 8 Solution of test T_3

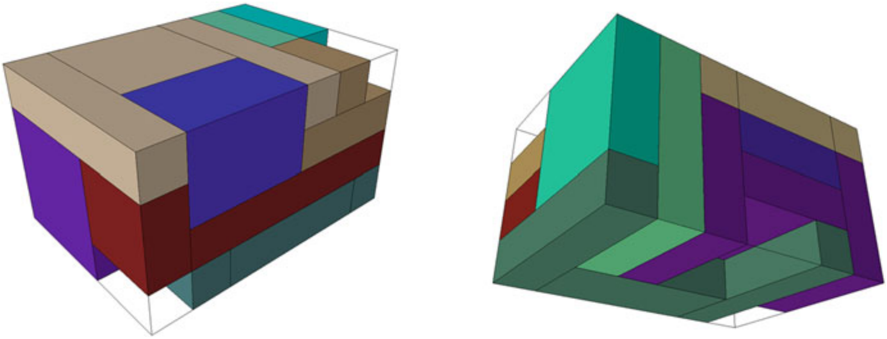


Fig. 9 Solution of test T_4

It is interesting also to note that two non-trivial instances (T_1 and T_10) were solved to optimality during the pre-processing phase.

4 Conclusive Remarks

This work focuses on experimental aspects relevant to an extension of the container loading problem, solved by a modeling-based heuristic approach. This is aimed at coping with complex non-standard packing problems, involving tetris-like items, non-box-shaped domains and additional conditions, such as balancing.

In this context we solved a set of 100 non-trivial test cases for the classical container loading problem, with additional balancing conditions, using a heuristic approach. We adopted a strategy based on the distributed parallel variant of B&C algorithm, as it is implemented in IBM CPLEX V. 12.6.2 on a single 4 cores machine. The quality of the results looks interesting, with an average load factor higher than 80 % in the instances with the center of mass in the geometric center of the domain.

We also tested the solution of a set of ten instances involving tetris-like items. These instances were directly solved according to the distributed parallel approach, without the usage of a heuristic. Also in this case, the results were encouraging, and a possible follow up is the study of the behavior of the optimizer with a higher degree of parallelism, on several distinct servers.

Acknowledgements We are very grateful to Jane Evans, for her accurate revision of the whole text and the number of suggestions provided.

Appendix

A list of selected test cases and their correspondence within the ‘Three Dimensional Cutting and Packing Data Sets - THPACK 1–7 BR’ is reported below.

	THPACK tests			THPACK tests			THPACK tests	
Article test name	Set number	Test case	Article test name	Set number	Test case	Article test name	Set number	Test case
GC_1	1	11	NC_1	1	1	GS_1	1	74
GC_2	1	43	NC_2	1	4	GS_2	1	98
GC_3	1	46	NC_3	1	7	GS_3	3	52
GC_4	1	99	NC_4	1	32	GS_4	3	87
GC_5	2	20	NC_5	1	41	GS_5	6	9
GC_6	2	30	NC_6	2	7	GS_6	6	33
GC_7	2	65	NC_7	2	10	GS_7	6	42
GC_8	2	76	NC_8	2	17	GS_8	6	85
GC_9	2	99	NC_9	2	50	GS_9	7	28
GC_10	3	2	NC_10	2	54	GS_10	7	54
GC_11	3	20	NC_11	2	56			
GC_12	3	29	NC_12	2	58			
GC_13	3	53	NC_13	2	76			
GC_14	3	83	NC_14	2	88			
GC_15	4	4	NC_15	3	13			
GC_16	4	27	NC_16	3	30	NS_1	1	2
GC_17	4	30	NC_17	3	68	NS_2	1	6
GC_18	4	65	NC_18	3	99	NS_3	3	16
GC_19	4	85	NC_19	4	45	NS_4	3	72
GC_20	4	87	NC_20	4	73	NS_5	4	19
GC_21	5	2	NC_21	4	82	NS_6	4	29
GC_22	5	19	NC_22	4	88	NS_7	4	96
GC_23	5	37	NC_23	5	15	NS_8	5	5
GC_24	5	38	NC_24	5	58	NS_9	5	99
GC_25	5	45	NC_25	5	65	NS_10	6	59
GC_26	5	87	NC_26	5	66			
GC_27	5	96	NC_27	5	86			
GC_28	6	29	NC_28	6	1			
GC_29	6	49	NC_29	6	10			
GC_30	6	56	NC_30	6	12			
GC_31	6	66	NC_31	6	34			
GC_32	6	84	NC_32	6	35			
GC_33	6	89	NC_33	6	38			
GC_34	7	45	NC_34	6	65			
GC_35	7	57	NC_35	6	68			
GC_36	7	84	NC_36	7	2			
GC_37	7	99	NC_37	7	11			
			NC_38	7	13			
			NC_39	7	21			
			NC_40	7	50			
			NC_41	7	61			
			NC_42	7	75			
			NC_43	7	78			

References

1. 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 (2008)
2. Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., Protasi, M.: Complexity and Approximation (Corrected Ed.). Springer, Berlin (2003). ISBN ISBN 978–3540654315
3. 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)
4. Bennell, J.A., Oliveira, J.F.: A tutorial in irregular shape packing problems. J. Oper. Res. Soc. **60**(S1), S93–S105 (2009)
5. Cagan, J., Shimada, K., Yin, S.: A survey of computational approaches to three-dimensional layout problems. Comput. Aided Des. **34**, 597–611 (2002)
6. Caprara, A., Monaci, M.: On the 2-dimensional knapsack problem. Oper. Res. Lett. **1**(32), 5–14 (2004)
7. Cassioli, A., Locatelli, M.: A heuristic approach for packing identical rectangles in convex regions. Comput. Oper. Res. **38**(9), 1342–1350 (2011)
8. 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)
9. 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)
10. 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)
11. 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)
12. 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)
13. 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)
14. 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)
15. Egeblad, J.: Placement of two- and three-dimensional irregular shapes for inertia moment and balance. In: Morabito, R., Arenales, M.N., Yanasse, H.H. (eds.) Int. Trans. Oper. Res. Special Issue on Cutting, Packing and Related Problems **16**(6), 789–807 (2009)
16. European Space Agency (ESA): Automated transfer vehicle (ATV). www.esa.int/Our_Activities/Human_Spaceflight/ATV. Accessed 3 Jan 2016
17. Fadel, G.M., Wiecek, M.M.: Packing optimization of free-form objects in engineering design. In: Fasano, G., Pintér, J.D. (eds.) Optimized Packings and Their Applications. Springer Optimization and Its Applications, pp. 37–66. Springer Science+Business Media, New York (2015)
18. Fasano, G.: Solving Non-standard Packing Problems by Global Optimization and Heuristics. SpringerBriefs in Optimization. Springer Science+Business Media, New York (2014)
19. 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, pp. 67–85. Springer Science+Business Media, New York (2015)
20. 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, pp. 87–117. Springer Science+Business Media, New York (2015)

21. 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 Science+Business Media, New York (2013)
22. Fekete, S., Schepers, J., van der Veen, J.C.: An exact algorithm for higher-dimensional orthogonal packing. Oper. Res. **55**(3), 569–587 (2007)
23. Fischer, A., Scheithauer, G.: Cutting and packing problems with placement constraints. In: Fasano, G., Pintér, J.D. (eds.) Optimized Packings and Their Applications. Springer Optimization and Its Applications, pp. 119–156. Springer Science+Business Media, New York (2015)
24. Fischetti, M., Luzzi, I.: Mixed-integer programming models for nesting problems. J. Heuristics **15**(3), 201–226 (2009)
25. Gliozzi, S., Castellazzo, A., Fasano, G.: Container loading problem MIP-based heuristics solved by CPLEX: an experimental analysis. In: Fasano, G., Pintér, J.D. (eds.) Optimized Packings and Their Applications. Springer Optimization and Its Applications, pp. 157–173. Springer Science+Business Media, New York (2015)
26. Goldreich, O.: Computational Complexity: A Conceptual Perspective. Cambridge University Press, Cambridge (2008)
27. 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)
28. IBM: CPLEX 12.6.0 User Manual. http://www-01.ibm.com/support/knowledgecenter/SSSA5P_12.6.2/ilog.odms.studio.help/Optimization_Studio/topics/COS_home.html?lang=en (2015)
29. 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 Academic, Hardbound (2003)
30. Jünger, M., Liebling, T.M., Naddef, D., Nemhauser, G.L., 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, Berlin (2010)
31. Junqueira, L., Morabito, R., Yamashita, D.S., Yanasse, H.H.: Optimization models for the three-dimensional container loading problem with practical constraints. In: Fasano, G., Pintér, J.D. (eds.) Modeling and Optimization in Space Engineering, pp. 271–294. Springer Science+Business Media, New York (2013)
32. Kallrath, J.: Cutting circles and polygons from area minimizing rectangles. J. Global Optim. **43**, 299–328 (2009)
33. Litvinchev, I., Infante, L., Ozuna, L.: Approximate packing: integer programming models, valid inequalities and nesting. In: Fasano, G., Pintér, J.D. (eds.) Optimized Packings and Their Applications. Springer Optimization and Its Applications, pp. 187–205. Springer Science+Business Media, New York (2015)
34. Martello, S., Pisinger, D., Vigo, D.: The Three-Dimensional Bin Packing Problem. Oper. Res. **48**(2), 256–267 (2000)
35. 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)
36. NASA (National Aeronautics and Space Administration): International Space Station (ISS). www.nasa.gov/mission_pages/station. Accessed 3 Jan 2016
37. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley, NewYork (1988)
38. Oliveira, J.F., Gomes, A.M., Ferreira, J.S.: TOPOS—a new constructive algorithm for nesting problems. OR Spectr **22**(2), 263–284 (2000)
39. Padberg, M.W.: Packing small boxes into a big box. Office of Naval Research, N00014-327, New York University (1999)

40. Pisinger, D.: Heuristics for the container loading problem. Eur. J. Oper. Res. **141**(2), 382–392 (2002)
41. Pisinger, D., Sigurd, M.: The two-dimensional bin packing problem with variable bin sizes and costs. Discret. Optim. **2**(2), 154–167 (2005)
42. Preparata, F.P., Shamos, M.I.: Computational Geometry. Monographs in Computer Science. Springer, Berlin (1990)
43. 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)
44. Silva, J.L.C., Soma, N.Y., Maculan, N.: A greedy search for the three- dimensional bin packing problem: the packing static stability case. Int. Trans. Oper. Res. **10**(2), 141–153 (2003)
45. Stetsyuk, P., Romanova, T., Scheithauer, G.: On the global minimum in a balanced circular packing problem. Opt. Lett. (2015). doi:10.1007/s11590-015-0937-9
46. Stoyan, Y., Pankratov, A., Romanova, T.: Quasi-phi-functions and optimal packing of ellipses. J. Glob. Optim. (2015). doi:10.1007/s10898-015-0331-2
47. Stoyan, Y., Romanova, T.: Mathematical models of placement optimization: two- and three-dimensional problems and applications. In: Fasano, G., Pintér, J.D. (eds.) Modeling and Optimization in Space Engineering, pp. 249–269. Science+Business Media, New York (2013)
48. Stoyan, Y., Romanova, T., Pankratov, A., Chugay, A.: Optimized object packings using quasi-phi-functions. In: Fasano, G., Pintér, J.D. (eds.) Optimized Packings and Their Applications. Springer Optimization and Its Applications, pp. 265–293. Springer Science+Business Media, New York (2015)
49. Stoyan, Y., Romanova, T., Pankratov, A., Kovalenko, A., Stetsyuk, P.: Balance layout problems: mathematical modeling and nonlinear optimization. In: Fasano, G., Pintér, J.D. (eds.) Space Engineering: Modeling and Optimization with Case Studies. Springer Science+Business Media, New York (2016)
50. Sun, Z., Teng, H.: Optimal layout design of a satellite module. Eng. Opt. **35**(5), 513–530 (2003)
51. Taha, H.A.: Operations Research, 7th edn. Macmillan, New York (2003)
52. 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)

Designing Complex Interplanetary Trajectories for the Global Trajectory Optimization Competitions

Dario Izzo, Daniel Hennes, Luís F. Simões, and Marcus Märtens

Abstract The design of interplanetary trajectories often involves a preliminary search for options later refined/assembled into one final trajectory. It is this broad search that, often being intractable, inspires the international event called Global Trajectory Optimization Competition. In the first part of this chapter, we introduce some fundamental problems of space flight mechanics, building blocks of any attempt to participate successfully in these competitions, and we describe the use of the open source software PyKEP to solve them. In the second part, we formulate an instance of a multiple asteroid rendezvous problem, related to the 7th edition of the competition, and we show step by step how to build a possible solution strategy. In doing so, we introduce two new techniques useful in the design of this particular mission type: the use of an asteroid phasing value and its surrogates and the efficient computation of asteroid clusters. We show how the basic building blocks, sided to these innovative ideas, allow designing an effective global search for possible trajectories.

Keywords Interplanetary trajectory optimization • Phasing indicators • Orbit clustering • Multi-objective tree search • Multiple-asteroid rendezvous • GTOC

D. Izzo (✉)
European Space Agency, ESTEC, 2201AZ Noordwijk, The Netherlands
e-mail: dario.izzo@esa.int

D. Hennes
DFKI GmbH, 28359 Bremen, Germany
e-mail: daniel.hennes@dfki.de

L.F. Simões
Vrije Universiteit Amsterdam, 1081 HV Amsterdam, The Netherlands
e-mail: luis.simoes@vu.nl

M. Märtens
TU Delft, 2628 CD Delft, The Netherlands
e-mail: m.maertens@tudelft.nl

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization
and Its Applications 114, DOI 10.1007/978-3-319-41508-6_6

1 Introduction

The design of interplanetary trajectories is a fundamental part of any future endeavour for the exploration of our solar system and beyond. Be it a sample return mission to Mars, the exploration of one of our gas giants, the first-time probing of objects in the Kuiper Belt, an asteroid deflection mission or the removal of dangerous orbiting debris, the complex interplay between the trajectory details and the mission objectives is what ultimately defines the overall mission value. The complexity and the diversity of interplanetary trajectories can be most immediately appreciated by looking at some remarkable examples such as the SMART-1 [23] transfer to Moon orbit, the Cassini tour of the Saturn system [27], or the Messenger [21] interplanetary transfer to Mercury. Many interplanetary trajectories were successfully flown by past spacecraft and even more were designed in the process of learning how to best navigate around our solar system. An outstanding example is that of the international event known as the Global Trajectory Optimization Competition (GTOC). Initiated in 2006, and currently heading towards its 9th edition, the GTOC is an event where, as the official web portal[1] reports: *the best aerospace engineers and mathematicians world wide challenge themselves to solve a "nearly-impossible" problem of interplanetary trajectory design.* Providing a valid solution to these problems is a rather complex endeavour requiring a solid understanding of space-flight mechanics and a good dose of innovative thinking as all of the problems have unique characteristics and thus require the development of new methods and solution approaches built on top of available common knowledge. In this paper, we summarize part of the necessary (but often not sufficient) basic knowledge required to participate to these competitions and we report and discuss, as an example, part of the solution strategy we employed to design our submission to the 7th edition. We base the selection of techniques reported on past GTOC editions, mainly focused on the preliminary design of low-thrust missions neglecting effects of a third body gravitational attraction.

The paper is divided in two main sections as follows: in Sect. 2 we describe fundamental problems that are often encountered during GTOCs. These include space flight mechanics problems (Sect. 2.1), specific types of optimal control problems (Sect. 2.2) and the efficient search of a computational tree (Sect. 2.3). In the second part of the chapter (Sect. 3) we build a search strategy for multiple asteroid rendezvous in the main belt. We formally define the problem making large use of the 7th GTOC problem data in Sect. 3.1. We then describe a set of new theoretical developments and their integration with the building blocks in a final algorithm described in Sect. 3.4 able to search for multiple asteroid rendezvous mission opportunities.

[1] http://sophia.estec.esa.int/gtoc_portal/.

2 Building Blocks

The design of a complex interplanetary mission is often made by (optimally) assembling solutions to a number of smaller problems, rather than tackling the problem as a whole. In this section, we introduce some of these basic "building blocks," and we show the reader how to solve them on a computer using the open source code PyKEP [15].[2] PyKEP is an open source software library developed and maintained at the European Space Agency, which allows non-experts to perform research on interplanetary trajectory optimization. We report rough estimates on the CPU time employed to find solutions to these basic problems, assuming one single thread of an Intel(R) Core(TM) i7-4600U CPU having the clock at 3.3 GHz and with a cache of 4096 KB. Note that we use non dimensional units throughout the PyKEP examples, but any set of consistent units would also be compatible with PyKEP.

2.1 Basic Space Flight Mechanics Problems

During the preliminary design of an interplanetary trajectory, and thus also in most GTOC problems, the spacecraft motion is approximated by that of a variable mass point, subject to the gravity attraction of one primary massive body with known gravitational parameter μ, and to the spacecraft thrust $\mathbf{T}$. Denoting with $\mathbf{r}$, $\mathbf{v}$ and m the position vector, velocity vector and mass of the spacecraft, the initial value (IV) problem describing its free motion in some inertial reference frame goes under the name of Kepler's problem (KP), mathematically defined as:

$$KP : \begin{cases} \ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} \\ \mathbf{r}(t_s) = \mathbf{r}_s \\ \mathbf{v}(t_s) = \mathbf{v}_s \end{cases} \tag{1}$$

where t_s is the starting time and $\mathbf{r}_s$, $\mathbf{v}_s$ the initial conditions. The position and velocity of a spacecraft at any time t is then obtained by propagating the above equations. Numerical integration can be avoided in this well studied case by the use of the Lagrange coefficients technique (see [2, 25] for implementation details). In PyKEP the KP is solved as follows:

```
from PyKEP import *
rs = [1,0,0]; vs = [0,1,0]; t = pi / 2; mu = 1
rf, vf = propagate_lagrangian(rs, vs, t, mu)
```

[2] https://github.com/esa/pykep/.

The CPU time requested by this operation is roughly constant across the whole spectrum of possible inputs. Using the above mentioned hardware, we measured a mean time of roughly 40 µs per KP, corresponding to 250,000 KPs solved in 1 s.

The boundary value problem (BVP) associated with the free motion of our spacecraft is, instead, known as Lambert's Problem (LP) and is mathematically described as follows:

$$LP : \begin{cases} \ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} \\ \mathbf{r}(t_s) = \mathbf{r}_s \\ \mathbf{r}(t_f) = \mathbf{r}_f \end{cases} \tag{2}$$

where t_s is the starting time, t_f the final time and $\mathbf{r}_s, \mathbf{r}_f$ the boundary conditions. The search for techniques to efficiently solve this problem has an interesting history [16]. The LP always results in at least one solution (the zero-revolutions solution) but, according to the value of $t_f - t_s$, may also result in several multi-revolutions solutions (mostly appearing in couples). In PyKEP (Python version) the LP is solved as follows:

```
from PyKEP import *
rs = [1,0,0]; rf = [0,1,0]; t = 20 * pi / 2; mu = 1; mr = 5
l = lambert_problem(rs, rf, t, mu, False, mr)
v1 = l.get_v1()[0]
v2 = l.get_v2()[0]
```

The CPU time requested by this operation depends on the number of existing multiple revolution solutions. If we limit ourselves to $mr = 0$, that is to the zero-revolutions case (which indeed is often the most important), the algorithm implemented in PyKEP (Python version) can be considered to have constant CPU time across all possible inputs [16]. Using the above mentioned hardware, we measured a mean time of roughly 40 µs/LP, corresponding to 250,000 zero revolutions LPs solved in 1 s.

We then consider the spacecraft motion subject to a thrust force $\mathbf{T}$ constant in the inertial frame. This problem is also called the constant thrust problem (CTP). Since the spacecraft operates its propulsion system, some mass needs to be expelled in order to obtain an effect on the spacecraft acceleration. The efficiency of such a reaction process is described by the constant I_{sp}, i.e. the propulsion specific impulse. The corresponding initial value problem is:

$$CTP : \begin{cases} \ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \frac{\mathbf{T}}{m} \\ \dot{m} = -\frac{T}{I_{sp} g_0} \\ \mathbf{r}(t_s) = \mathbf{r}_s, \mathbf{v}(t_s) = \mathbf{v}_s, m(t_s) = m_s \end{cases} \tag{3}$$

where g_0 is the Earth gravitational acceleration at sea level, typically set to 9.80665 [m/s^2]. The technique we employ to efficiently solve this problem is a Taylor series numerical propagator [19]. Other, more common, numerical propagators such as

Runge-Kutta-Fehlberg would be significantly slower. In PyKEP (Python version) the CTP is solved, to a relative and absolute precision of 10^{-12} (see [19] for the definition of such errors in the context of Taylor propagation), as follows:

```
from PyKEP import *
rs = [1,0,0]; vs = [0,1,0]; ms = 10; t = 2 * pi
T = [0.01, 0.01, 0.01]; mu = 1; veff = 1
rf, vf, mf = propagate_taylor(rs, vs, ms, T, t, mu, veff, −12,
   −12)
```

The CPU time requested by this operation depends linearly on the integration time t. Assuming the data in the example above (corresponding to one revolution along a circular orbit perturbed by a small thrust) and our reference thread performance, the algorithm implemented in PyKEP is able to solve the problem in roughly 230 ms, corresponding to 45,000 CTP solved in 1 s.

2.2 Optimal Control Problems

A fundamental aspect of interplanetary trajectory optimization problems where the spacecraft is equipped with a low-thrust propulsion system is the capability to solve Optimal Control Problems (OCPs) having the following mathematical description [7]:

$$
OCP : \left\{ \begin{array}{rl}
\text{find:} & \mathbf{T}(t) \in \mathscr{F}, \mathbf{x}_s, t_s, \mathbf{x}_f, t_f \\
\text{to minimize:} & J = \Phi(\mathbf{x}_s, t_s, \mathbf{x}_f, t_f) + \int_{t_s}^{t_f} \mathscr{L}(T(t), x(t), t)dt \\
\text{subject to:} & \\
 & \ddot{\mathbf{r}} = \frac{\mu}{r^3}\mathbf{r} + \frac{\mathbf{T}(t)}{m} \\
 & \dot{m} = -\frac{T(t)}{I_{sp}g_0} \\
 & \mathbf{g}(\mathbf{x}_s, t_s, \mathbf{x}_f, t_f) = \mathbf{0} \\
 & \boldsymbol{\varphi}(\mathbf{x}_s, t_s, \mathbf{x}_f, t_f) \leq \mathbf{0}
\end{array} \right. \tag{4}
$$

where $\mathscr{F}$ is the functional space containing all piece-wise continuous functions, $\mathbf{g}$ are equality constraints and $\boldsymbol{\varphi}$ are inequality constraints. We also introduced the spacecraft state $\mathbf{x} = [\mathbf{r}, \mathbf{v}, m]$ to shorten our notation. Note that the above problem, and particularly some of its more complex variants, are still the subject of active research. In most cases the objective J is one of (1) $J = t_f$ (time optimal control), (2) $J = m_f$ (mass optimal control), (3) $J = \int_{t_s}^{t_f} T^2(t)dt$ (quadratic control) or some combination of the above.

We will here shortly describe our approach (i.e. a direct approach based on the Sims-Flanagan transcription [24]) to solving the OCP as implemented in PyKEP, with the understanding that different approaches may perform better in some specific problems. Essentially, we divide the trajectory in $2n$ segments of constant duration $(t_f - t_s)/2n$ and we consider the thrust $\mathbf{T}(t)$ as fixed along these segments in an inertial reference frame. The value of $\mathbf{T}$ fixed for each segment is denoted with $\mathbf{T}_i$. This allows the OCP to be transformed into an equivalent Non Linear Programming problem (NLP) [20] having the following mathematical description:

$$NLP: \begin{cases} \text{find:} & \mathbf{T}_i \in F, i = 1..2n, \mathbf{x}_s, t_s, \mathbf{x}_f, t_f \\ \text{to minimize:} & J = \Phi(\mathbf{x}_s, t_s, \mathbf{x}_f, t_f) + \sum \int_{t_i}^{t_i+1} \mathscr{L}(\mathbf{T}_i, x(t), t)dt \\ \text{subject to:} & \mathbf{x}^- = \mathbf{x}^+ \\ & \mathbf{g}(\mathbf{x}_s, t_s, \mathbf{x}_f, t_f) = \mathbf{0} \\ & \boldsymbol{\varphi}(\mathbf{x}_s, t_s, \mathbf{x}_f, t_f) \leq \mathbf{0} \end{cases} \tag{5}$$

where F is a closed subset of $\mathbb{R}^3$, $\mathbf{x}^-$ is the spacecraft state as found propagating forward from $\mathbf{x}_s$ along the first n segments, while $\mathbf{x}^+$ is the spacecraft state as found propagating backward from $\mathbf{x}_f$ along the last n segments. The equality constraint $\mathbf{x}^- = \mathbf{x}^+$ is called mismatch constraint, while the requirement $\mathbf{T}_i \in F$ is generally transformed into an inequality constraint called throttle constraint and representing a limit to the maximum thrust allowed by the spacecraft propulsion system.

As a fictitious example, we consider the transfer from Earth conditions to Mars conditions of a spacecraft having $m_s = 4500$ [kg] and $I_{sp} = 2500$ [s]. The spacecraft is equipped with a low-thrust propulsion system capable of thrusting at $T_{max} = 0.05$ [N]. We consider the minimum time problem, thus the following formal description:

$$EM: \begin{cases} \text{find:} & t_s, t_f, m_f, T_{xi}, T_{yi}, T_{zi}, i = 1..2n \\ \text{to minimize:} & J = t_f \\ \text{subject to:} & \mathbf{x}^- = \mathbf{x}^+ \\ & (T_{xi}^2 + T_{yi}^2 + T_{zi}^2)^2 \leq T_{max}^2 \end{cases} \tag{6}$$

where $\mathbf{x}_s, \mathbf{x}_f$ are no longer in the decision vector as they are determined from t_s and t_f computing the Earth and Mars ephemerides. In PyKEP, the two constraints of the above EM problem are computed as follows, assuming the decision vector is known:

```
from PyKEP import *
# Example Decision Vector for 10 segments
n_seg = 10
ts = epoch(0, 'mjd2000'); tf = epoch(350, 'mjd2000'); mf = 2400
throttles = [0, 0, 1] * n_seg
# Computing the planets positions and velocity (ephemerides)
earth = planet.jpl_lp('earth')
mars = planet.jpl_lp('mars')
# Computing the equality and inequality constraints
sc = sims_flanagan.spacecraft(4500, 0.05, 2500)
rs, vs = earth.eph(ts)
rf, vf = mars.eph(tf)
xs = sims_flanagan.sc_state(rs, vs, sc.mass)
xf = sims_flanagan.sc_state(rf, vf, mf)
leg = sims_flanagan.leg(ts, xs, throttles, tf, xf, sc, MU_SUN)
ceq = leg.mismatch_constraints()
cineq = leg.throttles_constraints()
```

The CPU time requested for computing these constraints is $2n$ times that requested by the underlying CTP. Once equality, inequality constraints and the objective function computations are available, they can be used by an NLP solver to find the optimal solution. Widely spread solvers like IPOPT [26], SNOPT [12] and WORHP [4] are an obvious choice and have indeed been successfully used in connection to this type of NLPs and more in general in GTOCs and interplanetary trajectory design.

2.3 Tree Searches

The various problems described in Sects. 2.1 and 2.2 can be used in the design of interplanetary trajectories, such as those of the GTOC problems, as building blocks of a more complex search strategy. Such a search strategy is often some form of tree search, where each node represents a trajectory that can be incrementally built towards the mission goal expanding one of its branches (e.g. adding a fly-by, a rendezvous or, more generically, a trajectory leg). The exact detail of the nodes definition and their possible branching must be carefully designed according to the problem under consideration. A concrete example is given in a later section to clarify such a process in one particular, selected case. For the time being, one may picture each node as representing a partial trajectory and the branching as the process to add one or more phases to such a trajectory. Branching involves the solution of one or more sub-problems such as an LP, or an OCP, etc. and its complexity may vary greatly. Due to the complexity of the search, it is often impossible to exhaustively search the entire tree of possibilities. Simple text book implementations of breadth first search (BFS) or depth first search (DFS) are exhaustive search-strategies that will eventually branch out every possible node, which is most of the time never an option due to the enormous tree size.

Consequently, one has to develop a strategy that explores only areas of the tree that give best results while staying within a reasonable computational budget, i.e. the number of sub-problems to be solved. The key aspect in the design of a tree search strategy is then, given a set of active leaf nodes (i.e. partial trajectories), to choose which one is worth branching and what branches are worth computing. Since each node represents only a partial interplanetary trajectory, its value with respect to the achievement of the final mission goal is not necessarily available. In a typical example, at each node the remaining propellant mass r_m and the remaining mission time r_t are known and, only in some cases, a partial objective value J measuring the mission value achieved so far is available. Using this knowledge to decide what node to branch next is, as mentioned, of paramount importance.

The text book implementation of DFS [8] can be improved by introducing a pruning criterion preventing nodes to be further expanded. Such a criteria can make use of r_m, r_t and, when available, J as well as of the information on the best full trajectory found so far which will be available rather soon during the search since the tree is searched in depth. The main problem with this approach is that its running

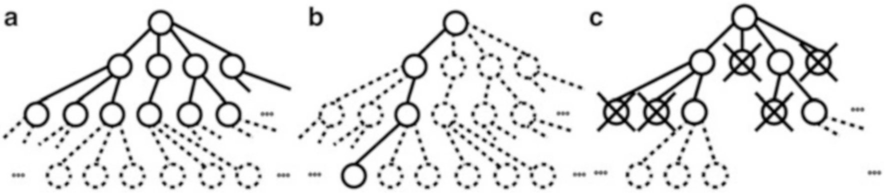


Fig. 1 Different tree search strategies in comparison. *Dotted nodes* are yet to be explored. *Crossed out nodes* are pruned and will not be branched. (**a**) Breadth-first-search (BFS). (**b**) Depth-first-search (DFS). (**c**) Beam-search (BS)

time is very sensitive to the pruning criteria, but it cannot be estimated upfront. As a consequence when a tree search starts one needs to wait for it to finish in order to decide whether the pruning criteria was too strict or loose. During the 5th GTOC this strategy was used [18] to explore the tree of Lambert's solutions that would then be converted into a low-thrust trajectory.

The text book implementation of BFS [8] can be improved by considering only a fixed number of nodes for branching at each depth. The nodes are prioritized using r_m, r_t and, when available, J. The resulting tree search is a standard tree search called beam search (BS). A version of this tree search strategy was employed by Jet Propulsion Laboratory in the design of their winning trajectory in the 5th GTOC edition [22]. Figure 1 gives a schematic comparison of beam search with BFS and DFS. A multi-objective version of beam search was also used during the 7th GTOC by our team, resulting in a search strategy we called MOBS and that is described in detail in a later section of this paper. An advantage of the BS/MOBS approach is that the complexity to explore an additional tree depth is easily computed as the complexity of the sub-problem to solve times the beam size. It is thus possible to estimate rather accurately the running time of the search before starting it.

In some problems, it is not possible to make a fair selection among nodes having equal depth in the tree. In fact, in most problems, the tree depth information is not directly related to relevant physical phenomena and it is just an artifact of how the problem under consideration is mapped into a tree search problem. A fairer comparison can be made among nodes representing trajectories that have a similar remaining mission time r_t at disposal to achieve their objectives. A tree search based on this simple idea, called Lazy Race Tree Search (LRTS) was used during the 6th GTOC and the resulting search strategy, employing self-adaptive differential evolution in the trajectory branching, received the gold "Humies" award for human-competitive results produced by genetic and evolutionary computation [17].

A different approach to tree searches, and one that is most popular between AI practitioners as it proved to be able to deal with the vast combinatorial complexity of board games such as the game Go [6], is the Monte Carlo Tree Search (MCTS). An implementation of the MCTS paradigm in the design of complex interplanetary trajectories was recently studied in the context of purely ballistic trajectories and fly-by sequences generation [14] suggesting that its use may be competitive with beam search.

3 Example: Multiple-Asteroid Rendezvous Mission

In this section we define a multiple-asteroid rendezvous mission. We reuse large part of the problem description released for the 7th edition of the GTOC and we describe a possible solution strategy. Our focus is on showing how new innovative ideas and design methods have to be developed and used aside the basic building blocks to allow for an efficient search of design options in this particular case, highlighting how the problem of "interplanetary trajectory design" is still far from being automated in its most general case.

3.1 Problem Definition

Consider a spacecraft S having an initial mass $m_0 = m_s + m_p$, where m_s is the dry mass and m_p the propellant mass. The spacecraft has a propulsion system defined by a maximum thrust τ_M, and a specific impulse I_{sp}. The maximum acceleration allowed by the propulsion system is denoted by $\alpha_M = \tau_M / m_s$. The set $\mathscr{A}$ contains N possible target asteroids of which the ephemerides (e.g. position and velocity) at each epoch $t_0 \in [\bar{t}_0, \underline{t}_0]$ are known or computable. We want to perform a preliminary search for possible multiple rendezvous missions allowing for the spacecraft to visit the largest possible number of asteroids within a maximum mission duration tof and allowing for a minimum stay time t_w on each of its visited asteroids. A visit is defined, mathematically, as a perfect match between the asteroid and the spacecraft positions and velocities. We focus on the case where the cardinality N of the set $\mathscr{A}$ (i.e. the number of possible target asteroids) is in the order of thousands and we assume the spacecraft can be delivered on a chosen starting asteroid at a chosen starting date.

This problem is relevant to the design of advanced asteroid belt exploration missions, such as the one considered in the 7th edition of the Global Trajectory Optimization Competition [5], advanced In Situ Resource Utilization (ISRU) missions or future concepts such as the APIES concept [9], as well as to the design of multiple active debris removal missions (in which case the set $\mathscr{A}$ contains orbiting debris rather than asteroids). The asteroid belt scenario types are studied in depth here, but the novel methods proposed are of more general significance. As data set, we use the 16,256 asteroids from the main belt that were used during the GTOC7 competition (visualized in Fig. 2). The ephemerides of such asteroids are computed

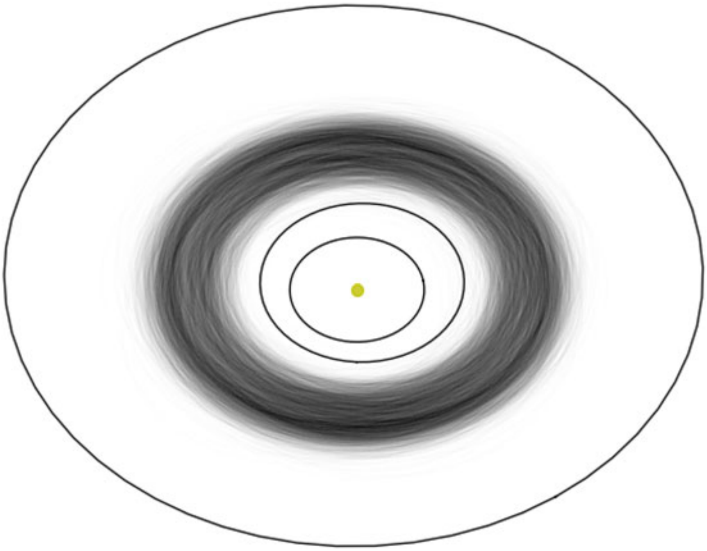


Fig. 2 Visualization of the orbits of 16,256 main belt asteroids considered for the GTOC7 problem. Going outward, the orbits of the Earth, Mars and Jupiter are also shown as reference

from the orbital parameters assuming perfectly Keplerian orbits. The actual orbital parameters can be downloaded from the GTOC7 problem data at http://sophia.estec.esa.int/gtoc_portal/. Each asteroid of such a data set is assigned a consecutive index, so that one can write A_j, with $j \in [1, 16{,}256]$ to identify uniquely the asteroid.

3.2 *Asteroid Phasing*

A fundamental problem in a multiple rendezvous mission is that of assessing which good transfer opportunities (target body, arrival epoch, arrival mass, etc.) are presented to a spacecraft S. Since it is computationally demanding to define and solve an optimal control problem (OCP) for each possible target body and launch/arrival window, we introduce a new quantity, easily computed and related to the cost of performing a given transfer: the phasing value, or asteroid phasing value in our chosen context. When good transfer opportunities from A_s to A_f exist at some epoch we say that those two asteroids are well "phased" and their phasing value will be small. Before introducing the formal definition of the phasing value, it is worth noting immediately how such a notion depends also on the spacecraft S and its propulsion system and not only on the starting and final body.

Assume the spacecraft S to be on the asteroid A_s at t_0 and consider possible Lambert problems (LP) to target the asteroid A_f. Let the starting (t_s) and final (t_f) epochs vary freely in $[t_0, t_0 + T_M]$ and consider the minimization of two final objectives ΔV and t_f only for trajectories for which the thruster can actually deliver the requested velocity increment, that is if $\Delta V \leq \alpha_M \Delta T$, where α_M is the maximum value for the thruster acceleration and $\Delta T = t_f - t_s$. The ΔV is computed from the solution of the LP as $\Delta V = \Delta V_1 + \Delta V_2$, where the two velocity increments represent the departure and arrival relative velocities along the Lambert solution. This results in the following two dimensional, two objectives, constrained optimization problem:

$$\begin{aligned} \text{find:} \quad & t_s, t_f \in [t_0, t_0 + T_M] \\ \text{to minimize:} \quad & f_1 = \Delta V, f_2 = t_f \\ \text{subject to:} \quad & \Delta V \leq \alpha_M \Delta T \\ & t_s < t_f \end{aligned} \tag{7}$$

The quality of its Pareto front is proposed as a quantitative measure for the notion of phasing, henceforth referred to as *phasing value* and indicated with $\varphi(A_s, A_f)$, shortened from $\varphi(A_s, A_f, t_0, T_M, \alpha_M)$. Note that one of the two objectives is the arrival epoch t_f and not the total time of flight which is, in this case, not relevant, also note that t_s indicates the starting epoch of the Lambert transfer and is not necessarily equal to t_0. As an example, take $A_s = A_{13155}$, $t_0 = 11{,}000$ [MJD2000] and $\Delta T = 365.25$ [days] and solve the above problem for $A_f = A_{12538}$ and $A_f = A_{3418}$ and a value of $\alpha_M = 0.375\ 10^{-4}$ [m/s^2]. The resulting Pareto fronts (computed using MOEA/D [28] and accounting for the constraints using a death penalty method [1]) are shown in Fig. 3 together with all the transfer orbits in the front. In this case, one may state that A_{3418} is better phased, at t_0, than A_{12538} with respect to A_{13155} (w.r.t. T_M and α_M). This definition has a straight forward application to the planning of multiple asteroid rendezvous missions as it allows to introduce a strict ordering over the set of possible target asteroids. In other words, given A_s, t_0 and ΔT and α_M, one can rank all the possible target asteroids with respect to φ and thus select the best for a further more detailed analysis.

The formal definition of Pareto front quality remains to be introduced. For the purpose of this work, the hypervolume [29] is used. In our simple two-dimensional case the hypervolume can be quickly visualized as the area between the front and the vertical and horizontal line passing through a reference point. In Fig. 3 (graphs on the right) this is easily done as the reference point also corresponds to the maximum values of the axes. When using the hypervolume as a quality indicator for Pareto fronts, one must take care to select a reference point p^*. We may use the following reference point: $p^* = [\Delta T \tau_M / m_s, t_0 + \Delta T]$. This definition ensures that whenever a feasible trajectory exists, its objectives are below the reference point. Note that the hypervolume has, in our case, the dimensions of a length and that larger values indicate better phasing values. In the example introduced, the computation of such a metric returns $\varphi = 1.1$ [AU] for the case $A_f = A_{12538}$ and $\varphi = 1.37$ [AU] for $A_f = A_{3418}$ indicating quantitatively that $A_f = A_{3418}$ is better phased.

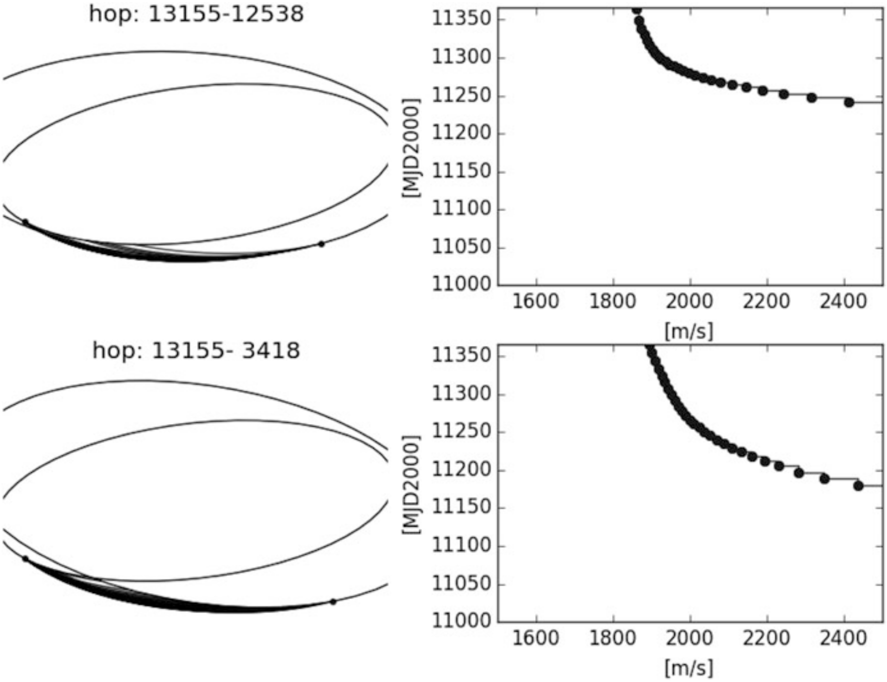


Fig. 3 Visual representation of the phasing value. Trajectories (*left*), Pareto front (*right*)

3.2.1 Phasing Indicators

The computation of the phasing value φ is done referring to its definition given by Eq. (7). A multi-objective optimization problem is solved and the hypervolume of the resulting Pareto front computed. While this procedure is fast in a single case, whenever a large number of phasing values are to be computed it does require significant computational resources. Since φ is ultimately used to rank transfer opportunities, one may consider to compute different quantities and study their correlation to the ground truth defined by φ. Such an approach will only be valuable if the new quantities, which we refer to as phasing indicators, are computed with less computational cost with respect to φ and the derived ranks have a high degree of correlation with those computed from φ. Two different phasing indicators are proposed and studied: the Euclidean indicator d_e and the orbital indicator d_o.

The *Euclidean indicator* is defined as $d_e = |\mathbf{x}_2 - \mathbf{x}_1|$, where $\mathbf{x} = [\mathbf{r}, \mathbf{v}]$, and contains information on both the asteroids relative positions and their relative velocities. The basic idea is that asteroids physically near to each other (and having a small relative velocity) are likely to be good candidates for an orbital transfer. The euclidean distance indicator can also be written as $d_e = \sqrt{|\Delta\mathbf{r}|^2 + |\Delta\mathbf{v}|^2}$ where $\Delta\mathbf{r}$ and $\Delta\mathbf{v}$ are the differences between the asteroid ephemerides. The main drawback of this indicator is that it is unable to distinguish between a case where the relative velocity eventually brings the asteroids closer and a case (e.g. having an identical $|\mathbf{x}_2 - \mathbf{x}_1|$) where the relative velocity tends to separate the asteroids.

A different indicator, which we call the *orbital indicator*, derives from the following simple linear model of an orbital transfer. Consider three points P_0, P_1 and P undergoing a uniform rectilinear motion. These represent the two asteroids and the spacecraft. Assume that the motion of the three points is determined by the equations:

$$\begin{aligned} \mathbf{r}_0 &= \mathbf{r}_{00} + \mathbf{v}_0 t \\ \mathbf{r}_1 &= \mathbf{r}_{10} + \mathbf{v}_1 t \\ \mathbf{r} &= \mathbf{r}_{00} + \mathbf{v} t \end{aligned}$$

At $t = \Delta T$ we let $\mathbf{r} = \mathbf{r}_1$ and we compute $\mathbf{v}$ as $\frac{1}{\Delta T}(\mathbf{r}_{10} - \mathbf{r}_{00}) + \mathbf{v}_1$. The point P (i.e. the spacecraft) is then moving, in ΔT, from P_1 to P_2. The necessary velocity increments to match the asteroid velocities are then:

$$\begin{aligned} \Delta V_0 &= \tfrac{1}{\Delta T}\Delta\mathbf{r} + \Delta\mathbf{v} \\ \Delta V_1 &= \tfrac{1}{\Delta T}\Delta\mathbf{r} \end{aligned} \tag{8}$$

The quantity $d_o = \sqrt{|\Delta\mathbf{V}_0|^2 + |\Delta\mathbf{V}_1|^2}$ is here proposed as a phasing indicator. In order to highlight that this indicator accounts for a linearized orbital geometry, we refer to it as the orbital indicator. The great thing about the orbital indicator is that if we associate in t_0 each asteroid A_i to a vector defined as $\mathbf{x}_i = [\frac{1}{\Delta T}\mathbf{r}_i + \mathbf{v}_i, \frac{1}{\Delta T}\mathbf{r}_i]$, then the orbital phasing indicator for the A_i, A_j transfer, is simply the euclidean distance between the corresponding vectors $\mathbf{x}_i$, $\mathbf{x}_j$.

3.2.2 Phasing Indicators as Phasing Value Surrogates

Both phasing indicators introduced result in a fast ranking of transfer opportunities. Assume to have t_0, ΔT and A_s (i.e. S is sitting on an asteroid at t_0) and to have to rank all asteroids in $\mathscr{A}$ as to consider only the first k for a detailed computation of the orbital transfer. If one is to use the euclidean indicator, this task is efficiently solved by computing the k-nearest neighbours (k-NN) in $\mathscr{A}$ to A_s using $\mathbf{x} = [\mathbf{r}, \mathbf{v}]$ (i.e. the asteroids position and velocities at t_0) to define the points in a six dimensional space. In a similar way, if one is to use the orbital indicator, the same task is as efficiently solved by computing the k-nearest neighbours using $\mathbf{x} = [\frac{1}{\Delta T}\mathbf{r} + \mathbf{v}, \frac{1}{\Delta T}\mathbf{r}]$ to define the points in a six dimensional space. In both cases, given the low dimensionality of the k-NN problem, a k-d tree data structure [3] is an efficient choice to perform the computation. The complexity to build a static k-d tree is $O(N \log N)$, while the k-NN query has complexity $O(k \log N)$. One single k-NN computation including the construction of the k-d tree, on our test case, takes on average 0.25 s, while the

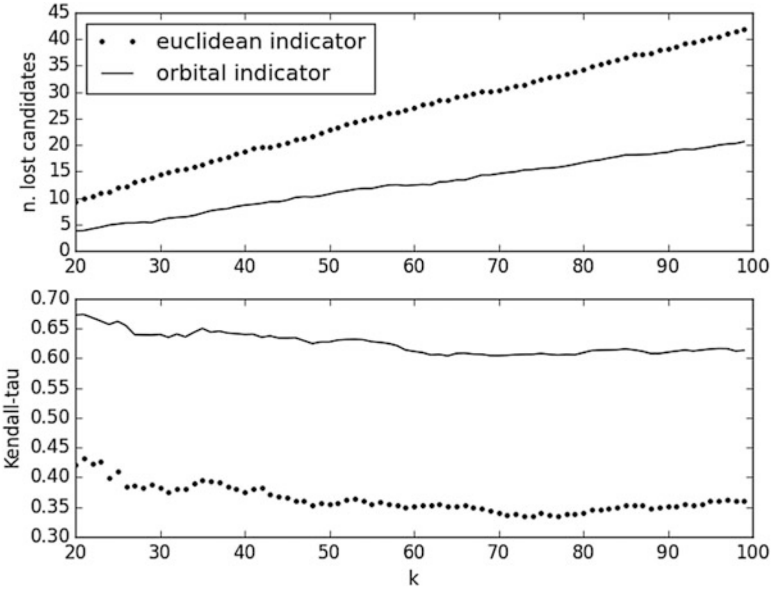


Fig. 4 Rank correlations of the proposed phasing indicators (average over 100 random cases)

computation of all the phasing values φ to then extract the best k targets, takes, on average, 5 min (tests made on an Intel(R) Core(TM) i7-4600U CPU having the clock at 3.3 GHz and with a cache of 4096 KB, exact implementation details available online as part of PyKEP.phasing module).

Such a speed increase (three orders of magnitude) is only useful if the resulting rankings are correlated. In Fig. 4 we show the rank correlations between the ground truth rank, computed using φ and those resulting from the newly introduced phasing indicators. The plot shows the average over 100 randomly selected t_0 and A_s. The value of the Kendall-tau coefficient is reported together with the number of false negatives, that is the asteroids that are within the best k according to the φ value, but are not within the k-NN computed using d_e or d_o.

The Kendall-tau coefficient is defined as $\tau = \frac{n_c - n_d}{(1/2k(k-1))}$, where n_c is the number of concordant pairs, whereas n_d is the number of discordant pairs. A value of $\tau = 1$ corresponds two identical rankings, similarly a $\tau = -1$ corresponds to two perfectly discordant rankings. In general, if the two rankings are uncorrelated, a value $\tau = 0$ is expected. The results show how both the new introduced quantities d_e and d_o are directly correlated with the phasing value φ and thus can be used as surrogates for the phasing value φ. The orbital indicator outperforms the euclidean indicator resulting in ranks better correlated with respect to the ground truth.

3.3 Clustering Asteroids

Besides ranking possible transfer opportunities, the phasing value indicators d_e and d_o can be useful to define a metric over the set of all asteroids $\mathscr{A}$ at each t_0. Such a metric can then be used to compute, for a given t_0, asteroid clusters. Using the orbital metric d_o as explained above, we define the points $\mathbf{x}_i = [\frac{1}{\Delta T}\mathbf{r}_i + \mathbf{v}_i, \frac{1}{\Delta T}\mathbf{r}_i]$ and apply clustering algorithms [13] directly on them. Large clusters of well-phased asteroids are likely to result in good opportunities for a multiple asteroid rendezvous mission. The clustering algorithm DBSCAN [11] is particularly suitable to find clusters in this domain. The algorithm has two fundamental parameters: ϵ, indicating the radius of the ball that defines each point neighbourhood and m_{pts}, defining the minimum number of neighbours necessary to be part of a cluster core. According to DBSCAN, an asteroid A belongs to a cluster $\mathscr{C}$ (i.e. a subset of $\mathscr{A}$) if it either has at least m_{pts} asteroids inside its ϵ neighbourhood or at least one of its neighbours does. In the first case A is said to be in the cluster core, otherwise it is labelled as a border point. If the orbital metric is used, the ϵ neighbourhood has an interesting interpretation. Asteroids within the ϵ neighbourhood of A will be reachable from A, according to the simple linear trajectory transfer model, with a transfer requiring a $\Delta V \leq \epsilon$ and a transfer time of T. In Fig. 5 a simple example visualizing a cluster, as defined by DBSCAN, is shown. Asteroids that are not associated to any cluster are labelled as outliers.

Consider now our data-set and a starting epoch in a 3 days resolution grid defined in [7500, 12,000] [mjd2000]. At each epoch, one can run DBSCAN and compute all

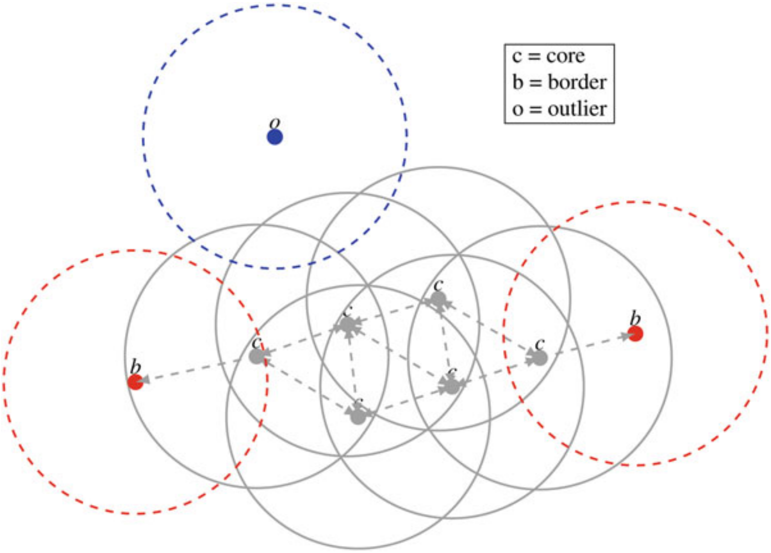


Fig. 5 DBSCAN clustering illustration for a 2-D case and a naive metric. $m_{pts} = 3$

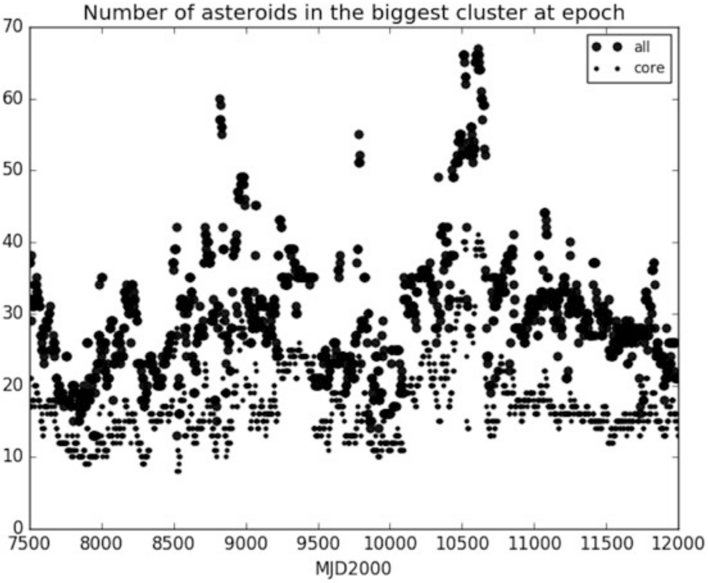


Fig. 6 Size of the largest cluster at epoch

asteroid clusters setting $\epsilon = 1650$ [m/s] and $m_{pts} = 5$. The result is visualized in Fig. 6 where, at each epoch, the size of the largest cluster found is reported as well as the number of core asteroids in it. Such a graph is proposed as a tool to help selecting the target area in the main belt and the time frame of a possible multiple rendezvous mission. In this particular case, for example, one notes that around the MJD2000 10,500 a special conjunction happens and relatively large clusters appear, having a size which is, on average, twice as much as clusters at other epochs. It comes then naturally that if a spacecraft was to be operating in one of these big cluster, it would have greater chances to find good transfer opportunities.

In Fig. 7 the actual clusters computed for $t_0 = 10{,}500$ MJD2000 are visualized together with some of the orbits of the main belt objects belonging to the data-set, the Earth and Jupiter orbit. The big cluster found by DBSCAN is clearly visible. All points in the cluster are guaranteed, in a first linear approximation, to be reachable from neighbours with a convenient orbital transfer having $\Delta V \leq \epsilon = 1650$ [m/s]. If they are core members of the cluster they are guaranteed to be reachable from at least m_{pts} other asteroids ($m_{pts} = 5$ in our case). Asteroids belonging to a cluster at t_0 are unlikely to form a cluster at different epochs as their orbital movement will tend to tear the cluster apart. Such an effect is directly proportional to ϵ as small values of ϵ imply similar orbital parameters.

Asteroid clusters are defined at a given epoch t_0 and, due to orbital motion they may disperse more or less quickly. To show how, in this case, a cluster persists in time for some years, in Fig. 8 the time evolution of a particular cluster is analyzed over a 6-year time span. The cluster we analyze is the biggest one detected by

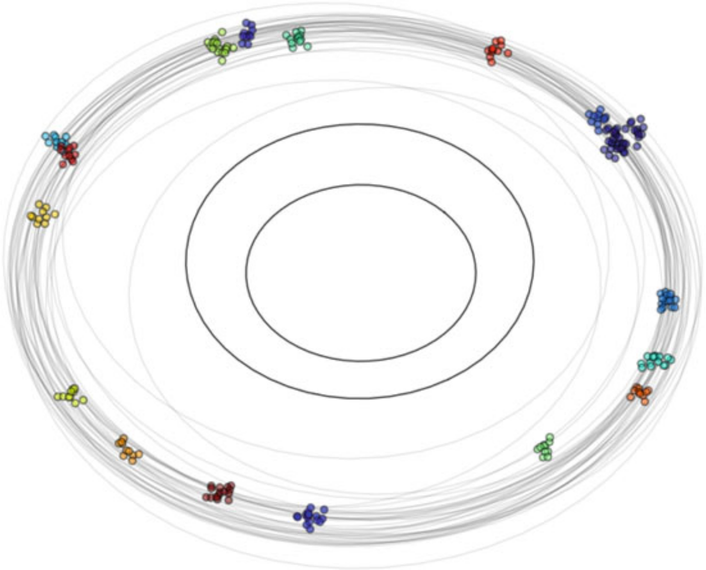


Fig. 7 Visualization of all clusters found by DBSCAN at $t_0 = 10{,}614$ MJD2000. The orbital metric is used. CLusters show in *different colors*. The Earth and Mars orbits are also shown as reference

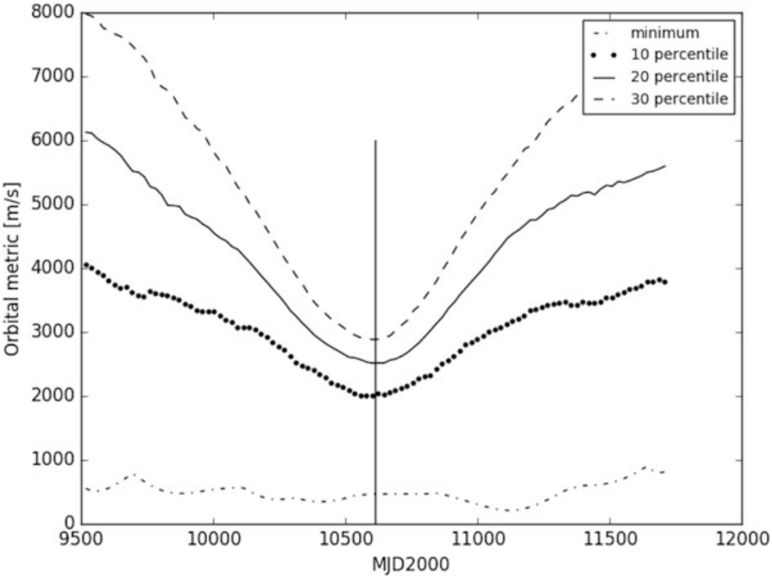


Fig. 8 Time evolution of the orbital metric computed for a cluster detected at $t_0 = 10{,}600$ [MJD2000]. All asteroid pairs are considered and an average over the best considered percentile is reported

DBSCAN and can be spotted in Fig. 7 as one big aggregate of points. A 6-year window is defined being centered at the epoch $t_0 = 10{,}614$ MJD2000 (when the cluster is detected). The orbital metric d_o is then computed for all pairs of asteroids in the cluster at each epoch in the defined window, and the minimum, the 10, 20 and 30 percentiles are reported. The plot shows how the cluster is resilient to being disrupted, at least within the considered time window. Since the minimum of the orbital metric d_o remains constantly low, at least two asteroids belonging to the cluster are always connected by an extremely advantageous orbital transfer, further more, while all the percentile plots present a minimum at the cluster epoch, they are only mildly increasing away from the cluster point. Similar features can be observed consistently for all clusters detected in the asteroid main belt.

3.4 Multi-Objective Beam-Search

Having developed the phasing value and asteroid clustering methods, we are now ready to build a procedure to search for complete multiple asteroid rendezvous missions. We approach the problem, as formally stated in Sect. 3.1, as a tree search problem. An algorithm based on the beam-search strategy is proposed, and its pseudo-code is shown as Algorithm 1. The algorithm, named MOBS, is a Multi-Objective Beam Search that accepts as inputs a starting epoch t_0 and a starting asteroid A_0 and searches for multiple rendezvous missions possible with the available resources. In MOBS, the non dimensional remaining mass $r_m = \frac{m-m_s}{m_p}$ and the non dimensional remaining time $r_t = 1 - \frac{t-t_0}{tof}$ are considered as the two resources available to the spacecraft. A node, representing a multiple-rendezvous trajectory, is defined as a triplet containing a list of visited asteroids $[A_0, A_1, A_2, \ldots A_n]$, and the two remaining resources r_t and r_m. The key elements of the algorithm are the Branch procedure and the Beam procedure. The Branch procedure is used to create branches from any particular node, which is equivalent to compute new transfers to

Algorithm 1 MOBS algorithm

```
procedure MOBS($t_0 \in [\bar{t}_0, \underline{t}_0], A_0 \in \mathscr{A}$)
    $\mathscr{B} = \{[[A_0], 1, 1]\}$, best = $[[A_0], 1, 1]$  ▷ Both resources are fully available at the beginning
    while $\mathscr{B} \neq \emptyset$ do
        $\mathscr{L} = \emptyset$
        for each $\mathscr{N} \in \mathscr{B}$ do
            $\mathscr{L} = \mathscr{L} \cup$ Branch($\mathscr{N}$)        ▷ Branch() creates maximum $BF$ new nodes
        end for
        best = UpdateBest($\mathscr{L}$)
        $\mathscr{B}$ =Beam($\mathscr{L}$)                              ▷ Beam() selects maximum $BS$ nodes
    end while
    return best
end procedure
```

one or more asteroids, increasing the overall objective of visiting as many asteroids as possible before the end of the mission. The Beam procedure is then used to select the most promising trajectories to be branched.

3.4.1 Branch Procedure

The Branch procedure takes a node (that is a multiple rendezvous trajectory), selects a maximum of BF (branching factor) possible asteroid targets and returns a list of new nodes created adding one of the target asteroids to the list of visited asteroids and updating r_t and r_m accordingly. The BF asteroids are selected among the ones having the largest (i.e. best) phasing value. A phasing value surrogate is used to speed up the computation, so that the BF asteroids will be the ones having the closest distance in the orbital metric d_o. Bodies already belonging to the list of visited asteroids are excluded as targets. The use of a k-d tree data structure makes the k-NN ($k = BF$) computations very efficient as mentioned in Sect. 3.2.2. For each of the target asteroids selected, the minimum arrival epoch optimal control problem is then solved first. If a solution is found, t^* being the earliest epoch the spacecraft can reach the target asteroid, also fixed arrival time, minimum mass optimal transfers are computed with $t_f = t^* + idt$, $i = 1..\ell$. For each feasible solution thus found, one can then compute the remaining resources r_t and r_m and insert a new node to the branched trajectory list. This branching procedure will return at most $\ell \cdot BF$ new trajectories visiting one more asteroid with respect to the parent node.

3.4.2 Beam Procedure

The Beam procedure takes a set of nodes and returns one of its subsets having at maximum BS members. In other words, it selects from a list of multiple rendezvous trajectories having equal depth, the most promising BS to be carried forward in the search. This selection is crucial to the performance of the overall scheme and is made introducing a node value r, computed for each node. The best BS nodes with respect to this value are returned. A trajectory should be considered good when it made clever use of the spacecraft's available resources, thus both r_m and r_t should be considered in defining the node value r. The first trivial choice would be to use directly r_m or r_t as a definition for the node value. This way, trajectories having spent a minimum amount of propellant, or time would be considered for further expansion. Since the phasing value is used to branch nodes, trajectories in the list already have been indirectly pre-selected with respect to a multi-objective criteria (the phasing value is defined with respect to the hypervolume), thus such a trivial choice would be less greedy than it appears and in fact, it works reasonably well when one knows upfront that one of the two available resources is particularly scarce. In the general case, though, a multi-objective aggregation of the two objectives seems like a more promising option to directly select good candidate trajectories. A number of options are thus proposed and summarized in Table 1 and

Table 1 Node value definitions

$r_1 = r_m$	Best mass
$r_2 = r_t$	Best time
$r_3 = \frac{1}{2}r_m + \frac{1}{2}r_t$	Average
$r_4 = \frac{r_m r_t}{r_m + r_t}$	Soft min
$r_5 = \min(r_m, r_t)$	Hard min
r_6	Pareto dominance

visualized in Fig. 9. A first direct approach is to consider a node value aggregating the two resources into one number via the expression $r = \lambda_m r_m + \lambda_t r_t$. The weights λ_m and λ_t implicitly define a priority on the two resources. One could thus consider them to be equal, in which case a simple *average* node rank is defined. The average node rank allows for a node having one of the resources almost completely depleted and the other fully available to rank equally to a node having both resources consumed half-way, which does not seem as a good choice. This suggests to introduce the *soft min* [18] case where the weights λ are adaptively modified according to how much a certain trajectory has used of a certain objective via the expression $\lambda_j = \left(1 - \frac{r_j}{r_m+r_t}\right)$. It is easy to recognize that this adaptive weight scheme is equivalent to use $r = \frac{r_m r_t}{r_m+r_t}$ for ranking. One possible problem with the *soft min* approach is that the weights, though adaptive, are still implicitly defining the importance of the two resources (mass and time) in a somewhat arbitrary fashion. A different approach is to use Pareto dominance concepts, where the top nodes are determined through a combination of non-dominated sorting and usage of a *Crowded Comparison* operator [10].

4 Experiments

We evaluate the overall performance of MOBS to search for multiple asteroid rendezvous missions. We consider the GTOC7 data so that the asteroids are moving along well defined Keplerian orbits. We also set a minimum waiting time on the asteroid $t_w = 30$ [days], an initial spacecraft mass $m_0 = 2000$ [kg], an initial propellant mass $m_p = 1200$ [kg], a maximum thrust $\tau_M = 0.3$ [N] and a specific impulse $I_{sp} = 3000$ [s]. We consider a maximum total mission duration of $tof = 6$ [years] and a starting epoch $t_0 \in [7500, 12{,}000]$ [mjd2000]. These values create a well defined instance of the multiple asteroid rendezvous problem defined in Sect. 3.1.

Table 1 shows the node value estimates used to evaluate MOBS. For all experiments we set *BF* to 10 and use the orbital metric for clustering as well as *k*-NN search. Additionally, MOBS requires a starting epoch t_0 and a starting asteroid A_0. We evaluate two different ways to provide such an initial condition:

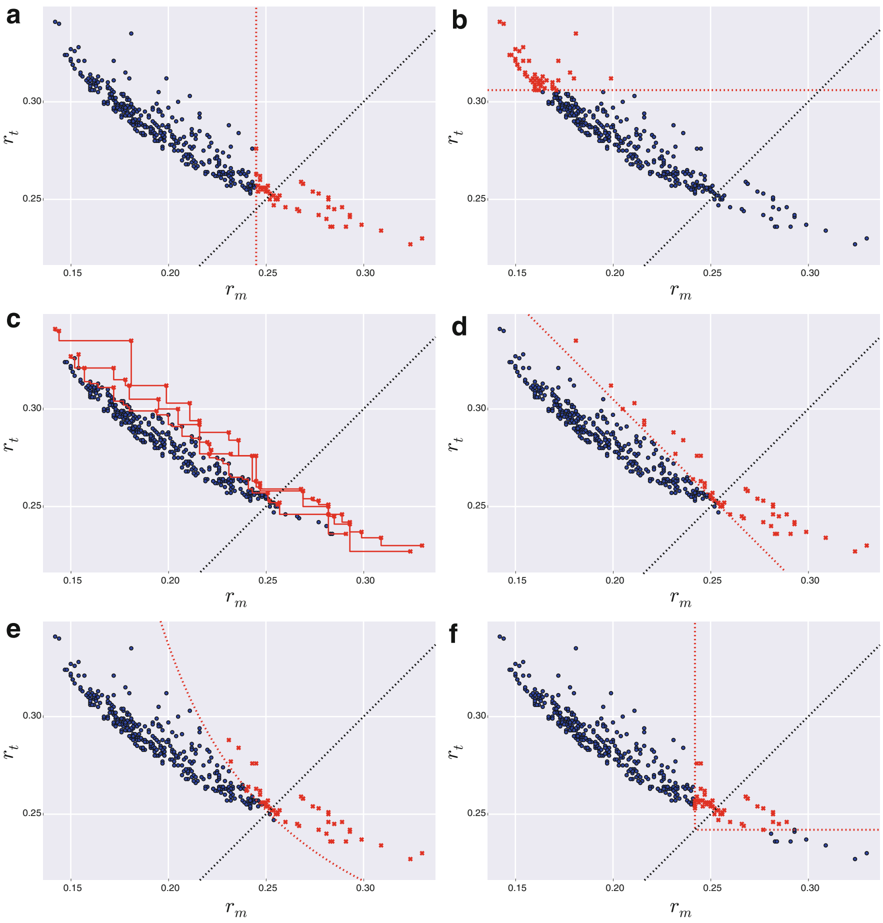


Fig. 9 Possible rankings for beam search: top 50 nodes at a given tree's current depth, as determined by the rating functions in Table 1. (**a**) r_1: best mass, (**b**) r_2: best time, (**c**) r_6: Pareto dominance, (**d**) r_3: average, (**e**) r_4: soft min, (**f**) r_5: hard min

- sampling t_0 uniformly from the launch window and choosing a random asteroid A_0 from the set of all 16,256 asteroids
- sampling t_0 uniformly from the launch window; perform clustering at epoch t_0 and select A_0 from the core of the largest cluster.

For each initial condition and node value estimator $r \in \{r_1, r_2, r_4, r_6\}$, we run 500 tree searches. A search results in a number of solutions of which we record the longest sequence. Table 2 shows the percentage of MOBS runs that resulted in a sequence of length at least $8, 9, \ldots, 14$ asteroids. The total number of solutions that reached a given length is reported in Fig. 10. On average, one full MOBS run took 1 h so that the entire experimental campaign here reported, involving 4000

Table 2 Multi-objective beam search (MOBS) performance

	Node value	≥8	≥9	≥10	≥11	≥12	≥13	≥14
With cluster	Mass	79.4	10.2	0.0	0.0	0.0	0.0	0.0
	Time	95.0	95.0	87.2	57.6	23.2	3.8	0.0
	Pareto	94.8	93.2	70.2	33.4	7.4	0.6	0.0
	Soft min	95.0	94.8	83.0	47.2	10.0	0.6	0.0
Without cluster	Mass	67.4	6.0	0.2	0.0	0.0	0.0	0.0
	Time	93.6	92.6	86.0	49.0	12.2	0.6	0.0
	Pareto	93.4	91.0	64.0	22.4	2.2	0.2	0.0
	Soft min	93.6	92.8	79.0	35.2	5.4	0.2	0.0

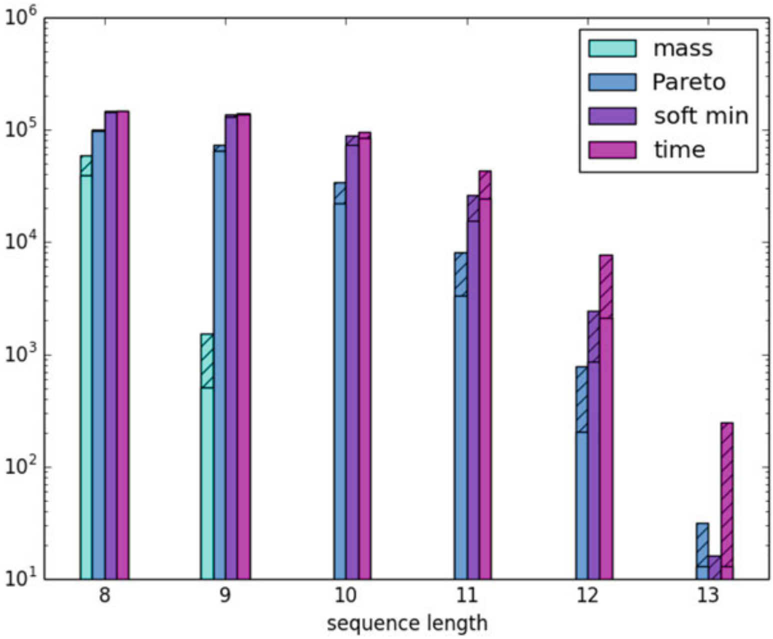


Fig. 10 Number of sequences generated by 500 searches for each node value estimator. The hatched part corresponds to the starting condition with clustering

tree searches, was run on a machine having 20 cores at 3.1 GHZ (40 parallel hyperthreads) during a period of, roughly, 4 days. During the runs, an approximate number of 30,000,000 OCPs are solved.

The maximum asteroid sequence length reported by participants of the GTOC7 as part of their final solutions was 13 (as reported during the GTOC7 workshop hosted in Rome in May, 2015). It is still an open question whether a sequence of length 14 exists, under the given settings, though it appears plausible. The proposed

MOBS algorithm is able to find sequences of length 12 with ease and it does find, in a small percentage of runs, longer sequences of asteroids to visit. MOBS never returned, neither during the experiments nor during the original runs made within the competition time-frame, a valid sequence of length 14.

Table 2 shows that the best node value estimator is time, while using mass result in extremely poor performances. While difficult to know upfront, this implies that the mission resources r_m and r_t are not evenly balanced: the spacecraft has comparatively more propellant than mission time in order to achieve its goals. Under different initial conditions, propellant could be a resource as scarce as time in which case other indicators should deliver better trajectories by exploiting the multi-objective trade-offs. The use of asteroid clustering greatly improves the chances to find good opportunities, focusing the search in the most promising areas of the asteroid belt at a given epoch.

We conclude this book chapter reporting, in Fig. 11, some visual information on one of the missions designed by MOBS and visiting 13 asteroids. We show the perihelion and aphelion for each of the asteroids visited overlapped to the background population of asteroids considered. We also show the osculating Keplerian elements during the entire mission as well as the spacecraft mass. The mission operates in the outer part of the main belt, acquiring a minimum distance of roughly 2.4 AUs and a maximum of 2.9 AUs. From the rapid increases and drops of the osculating semi-major axis during the same asteroid to asteroid transfers, a lot of thrust is used to cope with the non-perfect phasing. A control strategy that, though clearly sub-optimal when considering a single leg alone, allows to visit a greater number of asteroids when adopted to assemble the whole sequence.

5 Conclusions

The design of complex interplanetary trajectories is the subject of the Global Trajectory Optimization Competitions. Participating in these events requires a solid knowledge of basic astronomical problems such as the Kepler's problem, the Lambert's problem, the perturbed Kepler's problem, a certain familiarity with optimal control theory and algorithms as well as the development of original and innovative methods tailored for the particular problem to be solved. In the case of the 7th edition of this competition, the possibility to design multiple asteroid rendezvous missions was part of the problem assigned. We have found that a phasing value, defined as the hypervolume of the Pareto front of the multi-objective Lambert's transfer can be conveniently introduced and approximated using a surrogate orbital indicator. The phasing value approximation can be used to rank possible transfer opportunities and as a metric to define asteroid clusters in the main asteroid belt. We use these ideas to assemble a multi-objective tree search able to consistently design, in the GTOC7 data set, multiple rendezvous missions visiting up to 13 asteroids.

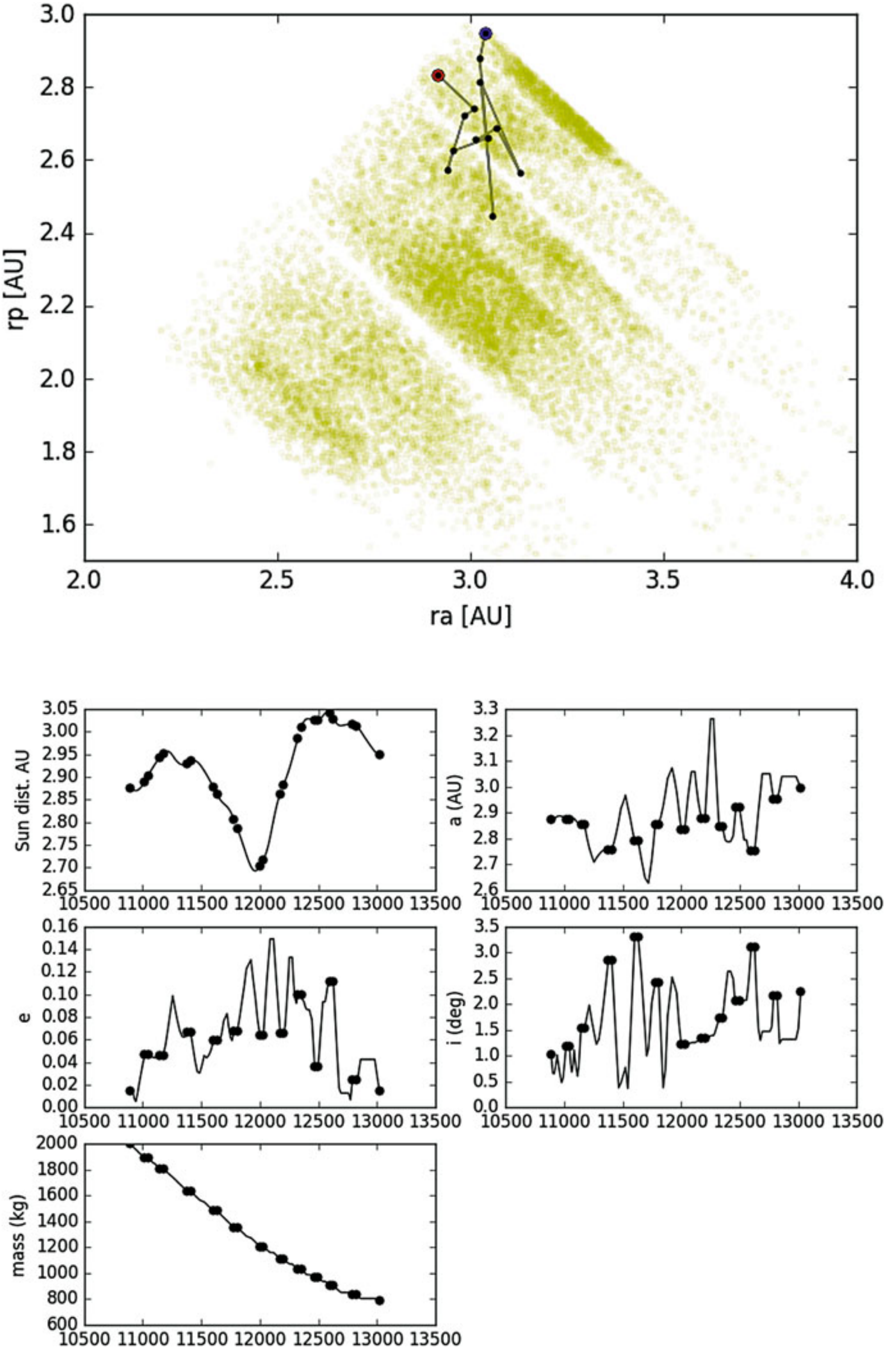


Fig. 11 Visualization of a multiple asteroid rendezvous mission visiting 13 bodies. *Top*: Aphelion and perihelion of the 13 asteroids visited in one of the mission designed by MOBS. The sequence starts at the *blue dot*. Asteroids from the main-belt population are also shown as background for reference. *Bottom*: some details of one of the trajectories found by MOBS. Each *dot* corresponds to a departure or arrival at one of the asteroids. Orbital parameters are the Keplerian osculating parameters

Acknowledgement Luís F. Simões was supported by FCT (Ministério da Ciência e Tecnologia) Fellowship SFRH/BD/84381/2012.

References

1. Bäck, T., Hoffmeister, F., Schwefel, H.: A survey of evolution strategies. In: Proceedings of the 4th International Conference on Genetic Algorithms, pp. 2–9 (1991)
2. Battin, R.H.: An Introduction to the Mathematics and Methods of Astrodynamics. American Institute of Aeronautics and Astronautics, Reston (1999)
3. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Commun. ACM **18**(9), 509–517 (1975)
4. Büskens, C., Wassel, D.: The ESA NLP solver WORHP. In: Modeling and Optimization in Space Engineering, pp. 85–110. Springer, New York (2013)
5. Casalino, L., Colasurdo, G.: Problem Description for the 7th Global Trajectory Optimisation Competition. http://areeweb.polito.it/gtoc/gtoc7_problem.pdf (2014). [Online. Accessed 10 Mar 2016]
6. Chaslot, G., Saito, J.T., Bouzy, B., Uiterwijk, J., Van Den Herik, H.J.: Monte-carlo strategies for computer go. In: Proceedings of the 18th BeNeLux Conference on Artificial Intelligence, Namur, pp. 83–91. Citeseer (2006)
7. Conway, B.A.: Spacecraft Trajectory Optimization, vol. 29. Cambridge University Press, Cambridge (2010)
8. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, vol. 5(3), p. 55. MIT Press, London (2001)
9. D'Arrigo, P., Santandrea, S.: The APIES mission to explore the asteroid belt. Adv. Space Res. **38**(9), 2060–2067 (2006)
10. 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)
11. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol. 96, pp. 226–231 (1996)
12. Gill, P.E., Murray, W., Saunders, M.A.: Snopt: an SQP algorithm for large-scale constrained optimization. SIAM J. Optim. **12**(4), 979–1006 (2002)
13. Grabmeier, J., Rudolph, A.: Techniques of cluster algorithms in data mining. Data Min. Knowl. Disc. **6**(4), 303–360 (2002)
14. Hennes, D., Izzo, D.: Interplanetary trajectory planning with monte carlo tree search. In: Proceedings of the 24th International Conference on Artificial Intelligence, pp. 769–775. AAAI Press, Palo Alto, California, USA (2015)
15. Izzo, D.: PyGMO and PyKEP: open source tools for massively parallel optimization in astrodynamics (the case of interplanetary trajectory optimization). In: Proceedings of the Fifth International Conference on Astrodynamics Tools and Techniques, ICATT (2012)
16. Izzo, D.: Revisiting lambert's problem. Celest. Mech. Dyn. Astron. **121**(1), 1–15 (2014)
17. Izzo, D., Simões, L.F., Märtens, M., de Croon, G.C.H.E., 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 (GECCO 2013), pp. 1301–1308. ACM Press, New York (2013)
18. Izzo, D., Simões, 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). doi:10.2420/AF08.2014.45
19. Jorba, À., Zou, M.: A software package for the numerical integration of odes by means of high-order taylor methods. Exp. Math. **14**(1), 99–117 (2005)
20. Kuhn, H.W.: Nonlinear programming: a historical view. In: Traces and Emergence of Nonlinear Programming, pp. 393–414. Springer, Basel (2014)

21. McAdams, J.V., Dunham, D.W., Farquhar, R.W., Taylor, A.H., Williams, B.G.: Trajectory design and maneuver strategy for the messenger mission to mercury. J. Spacecr. Rocket. **43**(5), 1054–1064 (2006)
22. Petropoulos, A.E., Bonfiglio, E.P., Grebow, D.J., Lam, T., Parker, J.S., Arrieta, J., Landau, D.F., Anderson, R.L., Gustafson, E.D., Whiffen, G.J., Finlayson, P.A., Sims, J.A.: GTOC5: results from the Jet Propulsion Laboratory. Acta Futura **8**, 21–27 (2014). doi:10.2420/AF08.2014.21
23. Racca, G., Marini, A., Stagnaro, L., Van Dooren, J., Di Napoli, L., Foing, B., Lumb, R., Volp, J., Brinkmann, J., Grünagel, R., et al.: Smart-1 mission description and development status. Planet. Space Sci. **50**(14), 1323–1337 (2002)
24. Sims, J.A., Flanagan, S.N.: Preliminary design of low-thrust interplanetary missions. In: AAS/AIAA Astrodynamics Specialist Conference, AAS Paper, pp. 99–338 (1999)
25. Vallado, D.A., McClain, W.D.: Fundamentals of Astrodynamics and Applications, vol. 12. Springer Science and Business Media, Berlin (2001)
26. Wächter, 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)
27. Wolf, A.A.: Touring the saturnian system. Space Sci. Rev. **104**(1–4), 101–128 (2002)
28. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)
29. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - a comparative case study. In: Parallel Problem Solving from Nature–PPSN V, pp. 292–301. Springer, New York (1998)

Satellite Constellation Image Acquisition Problem: A Case Study

Krishna Teja Malladi, Snezana Mitrovic Minic, Daniel Karapetyan, and Abraham P. Punnen

Abstract This chapter deals with the image acquisition scheduling of Earth observing satellites that revolve around the Earth in specific orbits and take images of prescribed areas requested by the clients. Often a satellite cannot acquire the images of a requested area in a single pass and it is necessary to divide the area into multiple strips each of which can be acquired in one satellite pass. Each satellite might have several image acquisition opportunities for each strip as the satellites can take images using different incidence angles. Then the Satellite Image Acquisition Scheduling Problem (SIASP) is to select the opportunities to acquire as many images as possible, without repetition, within a planning horizon while considering the image priorities and energy constraints. The proposed SIASP model employs a piecewise linear objective function to favor completion of an image acquisition request over partial acquisition of many requests. Extensive experimental study has been carried out using realistic randomly generated instances based on the forecasted statistics provided by MDA, Richmond, Canada. These experiments are

This work was done when Krishna Teja Malladi and Daniel Karapetyan were at the Department of Mathematics, Simon Fraser University, Surrey, BC, Canada

K.T. Malladi
Industrial Engineering Research Group, Department of Wood Science, University of British Columbia, Vancouver, BC, Canada V6T 1Z4

A.P. Punnen
Department of Mathematics, Simon Fraser University, Surrey, BC, Canada V3T 0A3

S.M. Minic (✉)
Department of Mathematics, Simon Fraser University, Surrey, BC, Canada V3T 0A3

MDA Systems Ltd., Richmond, BC, Canada V6V 2J3
e-mail: snezanam@sfu.ca

D. Karapetyan
ASAP Research Group, School of Computer Science, University of Nottingham, Nottingham, NGB 1BB, UK

Institute for Analytics and Data Science, University of Essex, Colchester, England, UK

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization
and Its Applications 114, DOI 10.1007/978-3-319-41508-6_7

intended as a preliminary investigation of the image acquisition scheduling for the Canadian RADARSAT Constellation Mission (RCM), a constellation of three satellites to be launched in 2018.

Keywords Clique • Satellite Image Acquisition Scheduling • Integer programming • Heuristic

1 Introduction

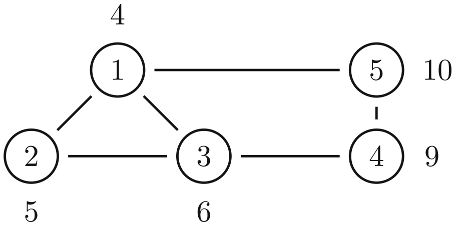
In this chapter we describe the Satellite Image Acquisition Scheduling Problem (SIASP) based on a case study of the Canadian RADARSAT Constellation Mission (RCM). We model SIASP as an optimization problem on a graph seeking a clique that maximizes a special objective function. We call this problem *Cluster Restricted Maximum Weight Clique Problem* (CRCP) [12] to reflect the fact that the graph is clustered and its structure affects the objective function. Although SIASP is closely related to the well known Maximum Clique Problem, it needs new solution approaches which we describe in this chapter.

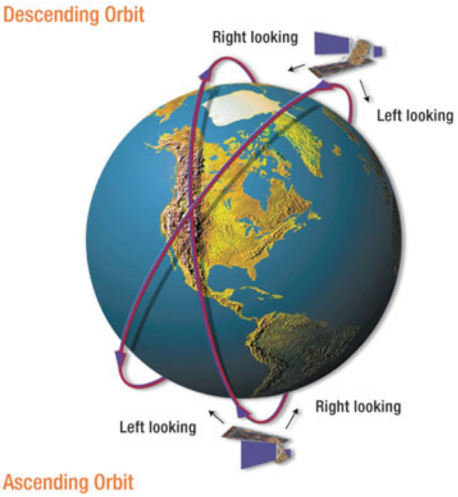
Let $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$ is the node set and E is the edge set. Each node $i \in V$ has an associated weight w_i. A clique is a complete subgraph in the given graph. For a clique Q of the graph G, the weight of the clique is defined by $\sum_{i \in Q} w_i$. The Maximum Weight Clique Problem (MWCP) deals with finding the clique with the maximum weight [3]. The Maximum Clique Problem (MCP) deals with finding the clique of the maximum size (with the maximum number of nodes). Thus, if $w_i = 1$ for all nodes $i \in V$, MWCP reduces to MCP.

For an illustration, consider a graph with five nodes and corresponding node weights as shown in Fig. 1. In the given graph, $w_1 = 4, w_2 = 5, w_3 = 6, w_4 = 9$ and $w_5 = 10$. The maximum weight clique in this graph is $\{4, 5\}$ with the weight of the clique equal to 19, whereas the maximum clique is $\{1, 2, 3\}$ with clique size equal to three.

SIASP deals with scheduling acquisition of images of regions on the Earth as requested by customers. This image acquisition mission is carried out by Earth observing satellites which generally orbit the Earth at low altitudes. The satellites in our case study use Synthetic Aperture RADAR (SAR) which allows acquisition of images of the Earth surface under any weather conditions and at any time of day. An example of a SAR satellite is given in Fig. 2.

Fig. 1 Example of MCP and MWCP





ORBIT CHARACTERISTICS	
Altitude (average)	798 kilometres
Inclination	98.6 degrees
Period	100.7 minutes
Ascending node	18 hours (±15 minutes)
Sun-synchronous	14 orbits per day
Repeat cycle	24 days
COVERAGE ACCESS USING 500 KM SWATH WIDTH	
North of 70°	Daily
Between 48° and 70°	Every 1-2 days
Equator	Every 2-3 days

Fig. 2 Orbit characteristics of RADARSAT-2, a Canadian Earth observing satellite, provided by MDA. Canadian Space Agency, 2015. All Rights Reserved. RADARSAT-2 and RCM are official marks of the Canadian Space Agency. Retrieved from www.asc-csa.gc.ca/eng/satellites/radarsat2/description.asp

The Canadian RADARSAT Constellation Mission (RCM) consists of three identical satellites. Each satellite can acquire images of areas which are located to the right of its ground track. The altitude of each satellite orbit is around 600 km. Each satellite takes about 96 min to complete one revolution around the Earth and has the capability to perform 12 min on average of imaging per orbit. The RCM would be able to cover on average 95 % of the world's surface daily. It is planned to launch these satellites in 2018. Each of these satellites has an estimated lifetime of 7 years.

Figure 3 shows various characteristics of the RCM satellites. Six different beam modes (out of ten available) along with the areas that can be covered by these modes have been indicated. Low resolution modes can acquire wider areas (swaths) whereas high resolution modes can only acquire more narrow areas (swaths). Within a given beam mode, there are different beam positions (sub-modes) that have particular incidence angles. (The incidence angle is the angle between the line joining the satellite and the Earth, orthogonal to the Earth surface, and the line joining the satellite and the area (swath) to be imaged.) Different incidence angles for higher resolution modes allow imaging along different swaths (as shown in Fig. 3). For example, for Very High Resolution 3 m mode, shown in red, six different swaths are shown and they correspond to different incidence angles. The satellite velocity vector represents the satellite's trajectory.

There has been a substantial amount of research work done on SIASP. In 1996, Bensana et al. [1] showed that SIASP is NP-hard. They proposed solutions and approaches, including a branch and bound based exact algorithm and heuristics

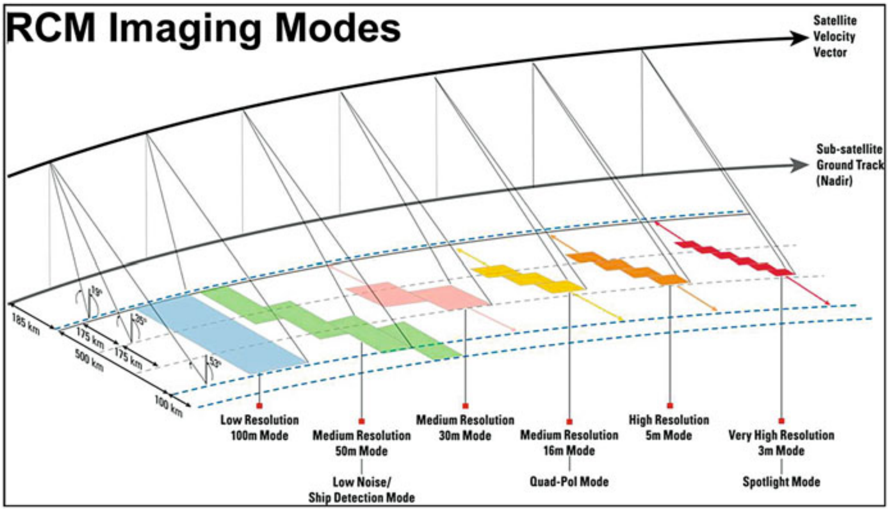

RCM Imaging Modes
Satellite Velocity Vector
Sub-satellite Ground Track (Nadir)
185 km
175 km
175 km
500 km
100 km
19°
35°
53°
Low Resolution 100m Mode
Medium Resolution 50m Mode
Low Noise/ Ship Detection Mode
Medium Resolution 30m Mode
Medium Resolution 16m Mode
Quad-Pol Mode
High Resolution 5m Mode
Very High Resolution 3m Mode
Spotlight Mode

Fig. 3 Characteristics of RCM. Canadian Space Agency, 2014. All Rights Reserved. RADARSAT-2 and RCM are official marks of the Canadian Space Agency. Retrieved from www.asc-csa.gc.ca/eng/satellites/radarsat/radarsat-tableau.asp

based on greedy algorithm and tabu search. They also compared their results with the solutions obtained by solving an integer programming formulation using the general purpose solver CPLEX. In 1997, Gabrel et al. [6] studied the problem of scheduling image acquisition by a low altitude satellite. They considered acquisition scheduling for a non-agile satellite, proposed a graph-theoretic model for the problem and suggested solution approaches that exploit the structure of the model. In 2002, Lemaître et al. [8] presented several algorithms to solve the Earth observation selecting and scheduling problem for agile satellites as a part of the French PLEIADES project. They provided a complete problem description and proposed a greedy algorithm, a dynamic programming algorithm, a constraint programming approach and a local search method to solve the problem. They considered a simplified version of the problem where their planning horizon was a single track or half orbit of the satellite. Cordeau and Laporte [4] proposed a tabu search heuristic to maximize the value of the Earth observation satellite orbit in 2005. In 2007, Bianchessi et al. [2] proposed a tabu search algorithm for a constellation of agile satellites. In 2011, Wang et al. [16] developed a priority-based heuristic for a Chinese satellite constellation which incorporated both image acquisition and downlink scheduling problems. In 2009, Li et al. studied image acquisition scheduling for a satellite constellation [9]. Their solution approach had two phases where in the first phase several tasks were assigned to each satellite of the constellation and in the second phase, a scheduling problem was solved for each satellite separately. In 2012, Tangpattanakul et al. [13] proposed a random-key genetic algorithm for the satellite image acquisition scheduling problem formulated as a multi-objective optimization problem. Their formulation has two objective functions, one is to maximize the total profit and the other one is to minimize the difference between profits of different users in order to ensure fairness among users. The problem that deals with scheduling the downlinks of images from satellites to the ground stations is called Satellite Downlink Scheduling Problem (SDSP). There is a substantial amount of literature on SDSP, see [7].

In this chapter, we describe the SIASP model for RCM and propose several data pre-processing strategies to improve the system's efficiency. The chapter is organized as follows. Section 2 describes SIASP and its constraints. Section 3 describes a mathematical formulation of the problem. Section 4 introduces graph models of SIASP. Section 5 describes how we generated realistic random test instances. Section 6 describes several pre-processing procedures. Section 7 gives the computational results. Finally, Sect. 8 provides the conclusions of the work.

2 SIASP

We are given a set of image acquisition orders for 1 year. Each order is a set of image acquisition requests, where each request corresponds to a set of Areas of Interest (AOI) on the ground whose images are to be taken. The following information is provided about each order:

- *Relative priority* defines the priority of the request.
- *Season of the year* during which the request is active.
- *Revisit frequency* defines the frequency with which the satellite constellation revisits a particular AOI. The time window within which the image of the region has to be acquired is defined by this revisit frequency.
- *AOI size* is a measure of the area of the region which is to be acquired.
- *Beam mode* specifies the resolution in which the image of the region has to be acquired.

Relative priority of an order is a number between 1 and 9, 1 being the lowest and 9 being the highest. Images of some AOIs are required only during certain seasons. RCM plans to provide complete coverage of Canada's land and oceans, offering a daily revisit. Revisit frequency of a request could be daily, weekly, bi-weekly, monthly, half-yearly and yearly and it may define the time window of the associated requests. For example, a request with daily revisit frequency has to be acquired completely within 1 day. The area of the region to be imaged could be as small as 2500 km^2 to as large as 11,400,000 km^2.

Each image acquisition request has a set of AOIs whose images are to be acquired. We call each such region a *target*. A target to be imaged may be too large to be acquired in one pass of a satellite. Such large targets are called *polygon targets*. Targets that can be acquired within one pass of a satellite are called *spot targets*. Since a polygon target cannot be acquired in one pass, it has to be divided into several *strips*. The width of each strip depends on the resolution defined in the image acquisition order. For a given resolution, the width of each strip is fixed. A spot target has one strip. Thus, each region may have one or several strips associated with it.

The satellites can acquire images of the regions that are to the right of their trajectories. For different beam modes, the images can be acquired with different incidence angles (see Fig. 3).

The actual strip image acquisition time and the duration of acquisition depend on the satellite's revolution around the Earth. Each time interval when an image of a strip can be acquired is called an *opportunity*. Each opportunity has a duration and an angle of incidence associated with it. The duration of an opportunity is equal to the time required to acquire the corresponding strip (i.e. the time that it takes the satellite to fly over the corresponding region of the Earth surface). Thus, all the opportunities corresponding to the same strip have the same duration. The constellation may have several opportunities to acquire the image of a strip within the planning horizon. Due to the freedom in the angle of incidence of the satellite, one strip may have opportunities in several consecutive orbits of a satellite. Since the satellites are non-agile, each strip has at most one opportunity per orbit of each satellite. The goal of SIASP is then to select a set of opportunities out of all opportunities that satisfy the constraints of the problem.

2.1 Constraints

The following are the constraints involved in the problem:

1. Each AOI has to be acquired within the request time window.
2. The image must meet the beam mode requirements stated in the order.
3. Each strip of AOI has to be acquired at most once.
4. It takes a constant amount of time, δ, for a satellite to switch between beam modes.
5. During one orbit, a satellite can acquire images for 12 min on average but not exceeding 20 min in any single revolution.
6. Each satellite can take only one image acquisition opportunity at a time.

Each strip has a finite number of opportunities for its image acquisition. The goal of SIASP is to select a subset from the set of all opportunities that satisfy the above mentioned constraints that maximizes the quality of the solution. Details of the objective function are given in Sect. 3.2.

3 Mathematical Modeling of the Problem

For simplicity, we assume that each AOI is represented by a single target. The same formulation can be used for the case where requests may have several targets.

3.1 Notations

The following notations are used for formulating this problem as an Integer Program.

- H is the planning horizon for the problem.
- Γ is the set of all satellites.
- R is the set of all image acquisition requests.
- S_r is the set of all strips for the request $r \in R$.
- A_j is the area of a strip $j \in S_r$.
- A^r is the total area of a request r, i.e. $A^r = \sum_{j \in S_r} A_j$.
- π_r' is the total area of strips S_r already acquired before the current planning horizon.
- $S = \bigcup_{r \in R} S_r$ is the set of all strips.
- P_j is the set of all opportunities when a strip $j \in S_r$ can be acquired within the current planning horizon.
- P is the set of all image acquisition opportunities within the current planning horizon. Thus, $P = \bigcup_{j \in S} P_j$.
- P^r is the set of all image acquisition opportunities for all the strips of a request r within the current planning horizon. Thus, $P^r = \bigcup_{j \in S_r} P_j$.
- s_i is the starting time of the image acquisition opportunity $i \in P$.
- λ is the time required for a satellite to orbit around the Earth. This time is a constant for all the satellites in the Constellation. We define one revolution as a time interval of duration λ. Starting from time $t = 0$, the interval $[0, \lambda]$ is the first revolution, the interval $[\lambda, 2\lambda]$ is the second revolution and so on. Thus the kth revolution is the interval $[(k-1)\lambda, k\lambda]$.
- T^k is the set of all opportunities that start within the k^{th} revolution: $T^k = \{i \in P : (k-1)\lambda \leq s_i < k\lambda\}$.
- κ is the number of revolutions of each satellite in a planning horizon. κ can be obtained from H.
- T_s is the set of all opportunities of image acquisition for the satellite $s \in \Gamma$ within the planning horizon H.
- C is the set of ordered pairs (u, v) where u and v are two conflicting opportunities. Opportunities u and v are called conflicting if acquiring both u and v by the same satellite is infeasible. We describe the procedure to construct this set C in Sect. 3.2.
- T_1 is the peak acquisition time allowed for a satellite during a revolution. $T_1 = 20$ min in our case study.
- T_2 is the average acquisition time allowed for a satellite per revolution. $T_2 = 12$ min in our case study.
- d_i is the duration of opportunity $i \in P$, i.e. the time required to acquire corresponding image.
- x_i is a decision variable such that $x_i = 1$ if opportunity i is used and 0 otherwise.

3.2 Integer Programming Formulation

In this section, we present the image acquisition problem as an integer programming model. The planning horizon is H. Thus the number of revolutions, κ, in this formulation range from 1 to $\lceil \frac{H}{\lambda} \rceil$.

$$\begin{aligned}
\text{Maximize } & \sum_{r \in R} c_r \\
\text{subject to } & \sum_{i \in T^k \cap T_s} d_i \cdot x_i \leq T_1 \;\; \forall\, k \text{ and } \forall\, s \in \Gamma, \\
& \sum_{i \in T_s} d_i \cdot x_i \leq \kappa \cdot T_2 \;\; \forall\, s \in \Gamma, \\
& \sum_{i \in P_j} x_i \leq 1 \;\; \forall\, j \in S_r \text{ and } r \in R, \\
& x_i + x_j \leq 1 \;\; \forall\, (i,j) \in C, \\
& x_i \in \{0, 1\} \;\; \forall\, i \in P.
\end{aligned}$$

In the above formulation, c_r, $r \in R$, reflects the importance of a request r. Let π_r be the maximum total area of request r that can theoretically be acquired by the end of the current planning horizon, i.e. $\pi_r = \pi_r{}' + \sum_{l \in P^r} A_j \cdot x_l$, where $l \in P_j$ and $j \in S_r$ and $\pi_r{}'$ is the area of the request that has been acquired till the end of the previous planning horizon. Then we calculate c_r as $c_r = p_r \cdot f(\frac{\pi_r}{A^r})$, where p_r is the priority of r and $f(\frac{\pi_r}{A^r})$ accounts for the progress of acquiring images of all the strips of request r.

By adjusting the shape of the function $f(x)$ we can achieve special effects. If $f(x)$ is a monotonically increasing convex function (see Fig. 4 for example), it encourages completion of requests which have been partially acquired during previous planning horizons instead of starting acquisition of completely new requests. This way we can avoid situation when many requests are partially fulfilled but only a few are completed. Similar approach has been used in the objective function in [4, 8, 16]. In what follows we call such functions $f_1(x)$.

In a situation where we have some requests whose start time and end time lie within the current planning horizon and we wish to acquire the strips of these requests only if we could acquire them completely, we could use a different function $f_2(x)$ as shown in Fig. 5. Such cases arise when we do not want to acquire images partially. The break point of $f_2(x)$ depends on the request for which this kind of function is considered. For such request r, the break point in $f_2(\frac{\pi_r}{A^r})$ is $1 - (A_t/A^r)$ where $t \in S_r$ is the strip which has maximum area out of all the strips in S_r.

Linearization of piecewise linear functions has been considered by several studies in the literature [10, 11, 14, 15, 17]. The following paragraphs describe two different ways to embed a piecewise linear function $f_1(x)$ shown in Fig. 4 into an

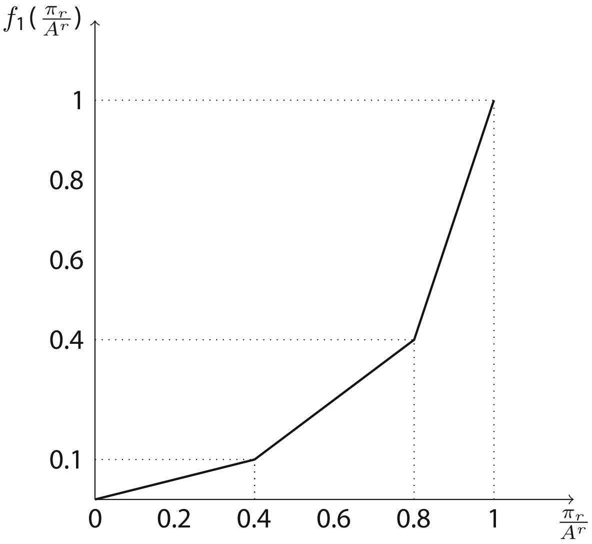


Fig. 4 $f_1(x)$: piecewise linear function used in the objective function of SIASP

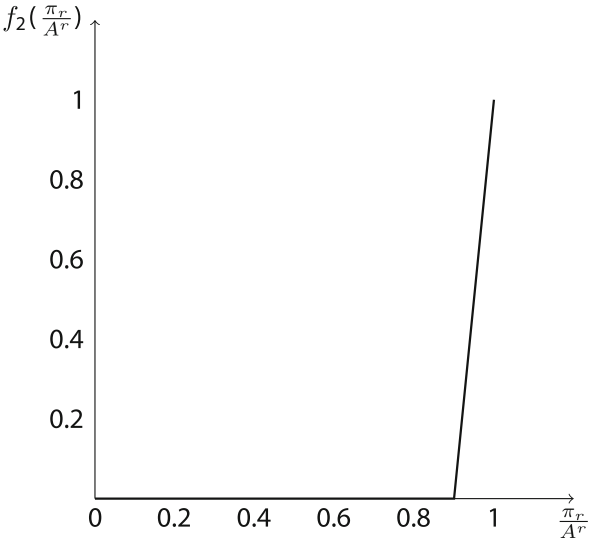


Fig. 5 $f_2(x)$: piecewise linear function used in the objective function of SIASP

integer linear program, one using the big-M method [10] and the other using special ordered set constraints of type 2 [17]. Similar approaches can be applied for $f_2(x)$ shown in Fig. 5. Let y denote the value of the function $f_1(x)$.

The first approach using the big-M method is as follows:

$$\begin{aligned}
&\text{Maximize } y\\
&\text{subject to } z_1 > x - 0.4,\\
&z_2 > x - 0.8,\\
&z_1 + z_2 \leq \frac{x}{0.4},\\
&y \leq 0.25x + z_1 \cdot M + z_2 \cdot M,\\
&y \leq 0.75x - 0.2 + (1 - z_1) \cdot M + z_2 \cdot M,\\
&y \leq 3x - 2 + (1 - z_1) \cdot M + (1 - z_2) \cdot M,\\
&0 \leq x, y \leq 1,\\
&z_1, z_2 \in \{0, 1\}.
\end{aligned}$$

In the above formulation, M is a large constant, and z_1 and z_2 are binary variables which act as indicators to which piece of the piecewise linear function the current value of x belongs to. For $x \in [0, 0.4)$, $(z_1, z_2) = (0, 0)$. For $x \in [0.4, 0.8)$, $(z_1, z_2) = (1, 0)$ and for $x \in [0.8, 1]$, $(z_1, z_2) = (1, 1)$. The values of z_1 and z_2 will be governed by the constraints $z_1 > x - 0.4$, $z_2 > x - 0.8$ and $z_1 + z_2 \leq \frac{x}{0.4}$. $y = 0.25x$, $y = 0.75x{-}0.2$ and $y = 3x{-}2$ are the equations of the three parts of $f_2(x)$. In order to avoid the strict inequalities in the first two constraints, we can introduce a very small constant m as $z_1 \geq x{-}0.4{+}m$ and $z_2 > x{-}0.8{+}m$. These strict inequalities are to be avoided in the formulation to ensure that the integer program has an optimal solution at an extreme point of the feasible region. Presence of these strict inequalities may lead to a situation where we can approach the optimal solution but cannot find it exactly. It is due to this reason that CPLEX requires that none of the constraints of the problem be strict inequalities.

The second approach to model $f_1(x)$ as linear constraints using special order subset type 2 constraints is as follows:

$$\begin{aligned}
&y = 0.1x_1 + 0.4x_2 + x_3,\\
&x = 0.4x_1 + 0.8x_2 + x_3,\\
&x_i \leq p_i,\ i = 1, 2, 3,\\
&p_1 + p_2 + p_3 \leq 2,\\
&p_1 + p_3 \leq 1,\\
&0 \leq x, y \leq 1,\\
&p_1, p_2, p_3 \in \{0, 1\},\\
&0 \leq x_1, x_2, x_3 \leq 1.
\end{aligned}$$

In this model, since x is continuous, x_i is also continuous for $i = 1, 2, 3$. The value of the piecewise linear function y is defined as $y = 0.1x_1 + 0.4x_2 + x_3$ where the coefficients are the break points of the function. This definition of y may not calculate the correct values of the piecewise linear function. To ensure correct calculations we need to restrict that at most two x_i can be non-zero and these two x_i must be adjacent. This is achieved by the introduction of binary variables p_1, p_2 and p_3. This way, any value of x between 0 and 1 can be obtained from specific values of x_1, x_2 and x_3 and the corresponding $f_1(x)$ can be found from $0.1x_1 + 0.4x_2 + x_3$. We observed in our preliminary experiments that this formulation converged to an optimal solution quicker than the one with the big-M. Similar results on the poor convergence of the big-M method have been stated in [15]. Thus, we use the formulation with special order subset type 2 constraints in our computational experiments.

Recall that the set C contains ordered pairs (i,j), where $i,j \in P$ conflict with each other. Two opportunities i and j conflict with each other when they have different beam modes or different angles of incidence and do not have enough time gap for the setup between them, i.e. $s_j - (s_i + d_i) < \delta$. If $(i,j) \in C$ then at most one of i and j can be acquired.

4 Graph Models of SIASP

A SIASP instance can be represented in a graph form such that a feasible solution is either a path or a clique in that graph, depending on the model used. In both these models, we assume the energy constraints of the problem are relaxed. Let $\{r_1, r_2, \ldots, r_n\}$ be the set of all opportunities of a SIASP problem instance. Let each opportunity r_i have a weight a_i and a time of possible acquisition t_i associated with it.

If the objective function is to maximize the sum of weights of all acquired opportunities, this problem can be modeled as a problem of finding the longest path in a directed acyclic graph $G(V, E)$. Let there be n vertices $\{v_1, v_2, \ldots, v_n\}$ in the graph, where a vertex v_i represents an opportunity r_i. Two opportunities r_i and r_j are compatible if both r_i and r_j can be acquired by the constellation during the same planning horizon. Let there be a directed edge (v_i, v_j) from v_i to v_j if $t_i \leq t_j$ for the compatible opportunities r_i and r_j. A graph constructed this way is acyclic and directed. Any path in G is a set of compatible opportunities. Thus any feasible solution to SIASP corresponds to a path in G. But some paths in the graph G may not correspond to a feasible solution to SIASP. This situation occurs when two opportunities of the same strip are compatible and the nodes representing these opportunities appear in one path. In such solutions, the acquisition of the same strip is repeated. Thus, certain paths in G, in which opportunities of the same strip are not repeated, correspond to feasible solutions of SIASP. Paths in G which obey an ordering of nodes can be proved to be feasible to SIASP. Thus, by constructing the edges of G which obey an ordering of the nodes, one may solve SIASP by solving the longest path problem in G. Due to the restriction in the ordering of nodes, SIASP may not be solved optimally using this model but good solutions can be obtained faster because solving the longest path problem in a directed acyclic graph can be done in polynomial time [5]. This model of SIASP is described in [8].

SIASP can also be modeled as a clique problem in a graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ and each node v_i represents an opportunity r_i. For two compatible opportunities r_i and r_j, let there be an edge (v_i, v_i) in the graph. There will be no edge between two nodes representing the opportunities of the same strip and there will be no loops in the graph. In the graph thus constructed, any clique represents a feasible solution to SIASP. All the opportunities of the same region can be clustered together and this will create a partition of the nodes. An optimal solution to SIASP would be a clique that maximizes the weight function as defined in the objective function of SIASP. We call this problem of finding a clique that maximizes $f(\frac{\pi_r}{A^r})$ the Cluster Restricted Maximum Weight Clique Problem (CRCP) and it is described in [12].

Our computational experiments have shown that the constraints for average acquisition and maximum acquisition times are not tight. The actual average and maximum acquisition times are less than 12 and 20 min, respectively. Thus, relaxing these constraints does not affect the final solution. Thus, by solving CRCP on a graph, we can solve SIASP and vice versa. Several solution strategies for solving CRCP are described in [12].

The remainder of this chapter solves SIASP using the approach described in Sect. 3.

5 Test Instances Generator

The north-south and east-west positions of a point on the Earth's surface are generally defined by latitudes and longitudes, respectively. Assuming Earth to be a perfect sphere, we use spherical co-ordinates to define the position of points on the Earth. Since the radius of the Earth is assumed to be a constant, we need only two co-ordinates to define a point. Let the equatorial plane be depicted by the xy plane and let the axis from the center of the Earth to the north pole be the z axis. Let θ be the angle made by the projection of the position vector of a point on the xy plane with the x axis and ϕ be the angle made by its position vector with the z axis. Thus, a point can be uniquely defined by the pair (θ, ϕ), where $\theta \in [0°, 360°)$ and $\phi \in [0°, 180°)$. For a point in the western hemisphere $\theta \in [180°, 360°)$ and for a point in eastern hemisphere $\theta \in [0°, 180°)$. $\phi \in [0°, 90°)$ represents a point in the northern hemisphere and $\phi \in [90°, 180°)$ represents a point in the southern hemisphere. This way, for a given latitude-longitude of any point, we can find the corresponding (θ, ϕ) and vice versa.

5.1 Data

Each image acquisition request has a set of targets on the ground to be imaged. The center of a target is defined by (θ, ϕ) at its center. Each target is assumed to be a region encapsulated by a pair of latitudes and longitudes. For each request we define the following attributes:

- A set of targets of the request.
- Area which is the total of the areas of all the targets (entire AOI) of the request.
- Time window during which the image acquisition has to be done.
- Priority between 1 (least important) to 9 (most important).
- Beam which represents the resolution in which the image has to be acquired.

We define each target by the (θ, ϕ) at its center and call them θ_{center} and ϕ_{center}. For a given center and area of a request, we can calculate the values of the θ and ϕ of its vertices. The calculations involved in finding these values of θ and ϕ are shown in Sect. 5.2. As we have described earlier, each target has one or more strips associated with it. Each target is defined by the following attributes:

- Center of the target defined by $(CenterTheta, CenterPhi)$.
- Area of the target.
- A set of strips of the target.

Each strip is defined by the θ and ϕ that bound the strip. Depending upon the satellite's orbit, the time and duration when a satellite can acquire the image of a particular strip can be computed. Each time interval when an image of a strip can be acquired is called an *opportunity*. Each strip is defined by the following attributes:

- The pairs of θ and ϕ which bound the strip.
- A set of opportunities for acquisition of the strip.

Each strip image acquisition opportunity has specific start and end times. The duration of an opportunity is equal to the time required by the satellite to acquire the image of the strip. Each opportunity has the following attributes:

- Satellite which has this opportunity of acquisition.
- Start time of the opportunity.
- End time of the opportunity.

5.2 Instance Generation

We are given a set of image acquisition orders for 1 year. Each order specifies the area of interest (AOI) or region, relative priority, season of the year, revisit frequency, area of the AOI and the beam requirements. We consider each revisit of the AOI to be an image acquisition request. The time window of a request is defined by its revisit frequency. For example, if the revisit frequency of a region is 1 week, then the time window within which the image has to be acquired is 1 week. About 66 % of the requests are within Canada. In the following paragraphs, we describe the procedure adopted to define the AOIs and targets from the center and the area size of the AOI. We make use of several formulas using spherical coordinates. The requests and AOIs are randomly generated from the statistical data we obtained. The exact position of the AOIs of each request were not provided due to confidentiality of information.

Depending on the shape of a AOI, we divide it into several targets. The AOIs comprised of several disconnected regions are considered to be a set of distinct targets. Thus, each AOI has a set of targets. Each target is identified by its center and area. If r is the radius of the Earth, ϕ_{top}, ϕ_{bottom}, θ_{left} and θ_{right} are the θ and ϕ that bound the target and $\rho = \theta_{left} - \theta_{right}$, then the area A of the target is given by the formula $A = r^2[cos(\phi_{top}) - cos(\phi_{bottom})] \times \rho \times \frac{180}{\pi}$. Since ϕ_{center} is defined for each target, by using $\delta = \phi_{bottom} - \phi_{top}$, we can calculate the values of the ϕ_{top} and ϕ_{bottom} using the formulas $\phi_{top} = \phi_{center} - \delta/2$ and $\phi_{bottom} = \phi_{center} + \delta/2$. To find the value of δ, we need to know the vertical length of the target. In order to ensure that the difference between the length and width of the target is not very large, we assume that the length of the target is $\sqrt{A}$. Thus, by using the formula $r \times \delta \times \frac{\pi}{180} = \sqrt{A}$ we can find the value of δ and thus the values of ϕ_{top} and ϕ_{bottom}. Since $A = r^2[\cos(\phi_{top}) - \cos(\phi_{bottom})] \times \rho \times \frac{180}{\pi}$ and the only unknown value in this formula now is ρ, we can find the value of ρ. Using this value of ρ, $\theta_{left} = \theta_{center} + \rho/2$ and $\theta_{right} = \theta_{center} - \rho/2$ can be calculated.

For requests to the extreme north or south, ϕ_{top} or ϕ_{bottom} may go beyond the interval $[0°, 180°]$. In such cases we fix the ϕ_{top} to 180° if calculated $\phi_{top} > 180°$ and we fix ϕ_{bottom} to 0° if the calculated $\phi_{bottom} < 0$. We also ensure that θ_{left} and θ_{right} of each request lies within the interval $[0°, 360°]$, *i.e.*, if the calculated θ_{left} and θ_{right} are out of the interval $[0°, 360°]$, we assign to them their equivalent values within the interval. This way, for a given target and its area, we can approximately find the corners of the target border. In the future, if the corners of each target are specified, we can use those values directly.

For each request given in the orders, we generate the center of the region randomly such that $2/3$rd of the requests are within Canada. We also make sure that each day there is a request for the water bodies around Canada.

Each target may have one or more strips. Each strip can be considered as a spot target. Thus, a strip also has θ_{left}, θ_{right}, ϕ_{top} and ϕ_{bottom}. The ϕ_{top} and ϕ_{bottom} of the strips of a target are the same as those of the target. The distance between points (θ_1, ϕ) and (θ_2, ϕ), given by $r \times \sin(\phi) \times |\theta_1 - \theta_2| \times \frac{\pi}{180}$, depends on ϕ. Thus for a target, we select a $\phi_\circ$ where $r \times \sin(\phi_\circ) \times |\theta_{left} - \theta_{right}| \times \frac{\pi}{180}$ is maximum, where $\phi_\circ \in [\phi_{top}, \phi_{bottom}]$ and divide the target along this $\phi_\circ$. If w_b is the width of each image as defined by the beam b and if $w_m = r \times \sin(\phi_\circ) \times |\theta_{left} - \theta_{right}| \times \frac{\pi}{180}$, then the number of strips into which the request is divided is $\left\lceil \frac{w_m}{w_b} \right\rceil$. This way by dividing along $\phi_\circ$, we make sure that the strips generated completely cover the target. If the target can be acquired in one pass, then we assume that it has one strip. Thus, each target is a set of strips and these strips cover the complete area of the target.

Each of the three satellites orbits the Earth in 96 min. These satellites follow near-polar orbits. The satellites have a certain angle of inclination with the longitudes (yaw angle). For simplicity, we assume that this angle on inclination is 0. It means that the satellites orbit the Earth along the longitudes. This is an approximation to the actual trajectory of the satellites. We assume that each orbit of the satellite is divided into two halves, the first half is along a longitude θ and the other half is along the longitude $(180 + \theta) \mod 360$. The path of the satellite from the North pole to the South pole is called *descending pass* and that from the South pole to the North pole is called *ascending pass*. For uniformity, we assume that an orbit of a satellite has its descent followed by its ascent. The θ along the descent is called the $\theta_{descent}$ and along the ascent is called the θ_{ascent}. The difference between the $\theta_{descent}$ of two consecutive orbits of a satellite is constant. The satellites are looking right within a visibility range, *i.e.* they can only acquire the strips that are located to the right of the satellites ground track at a distance not greater than its visibility range. It takes a constant amount of time for the satellite to change the incidence angle. The strips of targets must be parallel to the satellites ground track and this is how we generated them.

Each strip has at most one image acquisition opportunity during an orbit of a satellite. Since we know the initial position of the satellites at the beginning of the planning horizon, we can predict their position as a function of time and the details of the orbits such as the number of the orbit, start time of the orbit, $\theta_{descent}$ and θ_{ascent}. With this information, we can compute the opportunities when a strip can be acquired. Let $T_\circ$ be the time taken by a satellite to complete one orbit around the Earth. Let a strip s be acquired during orbit k of the planning horizon and let α be the difference between $\theta_{descent}$ of two consecutive orbits of a satellite. If s is acquired during the descent of the orbit, then $BeginTime(s) = (k-1)T_\circ + (\frac{\phi_{top}}{360})T_\circ$. If s is acquired during the ascent, then $BeginTime(s) = (k-\frac{1}{2})T_\circ + (\frac{180-\phi_{bottom}(s)}{360})T_\circ$.

6 Preprocessing the Instances

In Sect. 5, we discussed a method to divide the polygon targets into strips. Those strips have their vertical length equal to the vertical length of the polygon target. The width of each strip is equal to the width as defined by the beam mode, except for the last strip whose width might be smaller. Let w_b be the maximum width of an image that can be acquired with beam b. Thus, for a request with k strips along a latitude θ and with a requirement of beam b, the width of the first $k-1$ strips is equal to w_b and the width of the last strips is at most w_b. We call strips generated this way *large strips*. From now on, we will represent these large strips as simple requests, instead of the regions enclosed by a pair of latitudes and longitudes. This situation is explained in Fig. 6 where a polygon target is divided into four strips. In Fig. 6, the widths of s_1, s_2 and s_3 are equal to w_b along the θ of division. The width of s_4 along the θ of division is at most w_b. If w_{s_4} is the width of s_4, then the width of the target along the θ of division is equal to $3 \cdot w_b + w_{s_4}$.

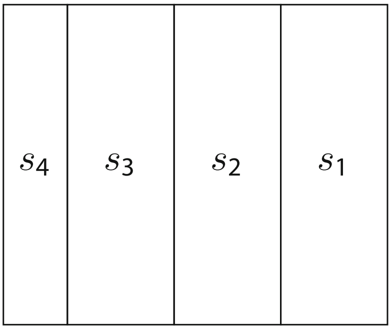


Fig. 6 A polygon target divided into strips

The division of large requests into large strips may not be an optimal way of generating the strips. In case of a conflict between two large strips for the satellite resources, according to the constraints of the problem, only one of them can be acquired and the other has to be dropped. In these situations, it would be reasonable to cut these large strips into smaller ones. Let two strips s_1 and s_2 be bounded by $(\theta_1, \phi_1, \theta_2, \phi_2)$ and $(\theta_3, \phi_3, \theta_4, \phi_4)$ respectively. If $\theta_1, \theta_2, \theta_3, \theta_4$ are such that both these strips are visible to the satellite during the same orbit, both during the ascent or both during the descent, then the corresponding opportunities of these strips conflict if $\phi_1 \leq \phi_3 \leq \phi_2$. Two such situations where conflicts between opportunities arise are shown in Fig. 7. In such a case, we will cut the strips into shorter ones, at the ϕ of the intersection. These divisions are depicted in Fig. 8. We will call these strips obtained from this procedure *small strips*. Now, with 66 % of the image acquisition requests within Canada, there are high chances of conflicts between many strips. For each of such conflicts, if we divide the large strips into small ones, we may end up in a situation where we have a very large number of very small strips. In order to avoid this situation, we make sure that we divide two conflicting large strips s_1 and s_2 into small strips only if the length of both strips is at least 100 km. By dividing these strips into small strips as shown in Fig. 8, the satellite may be able to acquire a larger total area than before. In Fig. 8, we can see that there is still a conflict between two strips, but these conflicting strips are smaller in length than before.

The satellites are right-looking, i.e. a satellite can acquire images only of strips that are to the right of its tracks. Even if only a fraction of the strip is to the left of the satellite, it cannot be acquired. To handle that, it is reasonable to divide such a strip into two narrow strips so that one of them lies entirely to the right of the satellite's track and the other one entirely to the left. Let $\theta_\circ$ be the θ of the satellite's orbit and θ_1 and θ_2 be the left and right boundaries of the strip. The situation described above arises if the longitude corresponding to $\theta_\circ$ lies between those corresponding to θ_1 and θ_2. We then divide this strip into new narrow strips s_1 and s_2 whose left and right boundaries are $(\theta_1, \theta_\circ)$ and $(\theta_\circ, \theta_2)$, respectively. We call such strips *narrow strips* (see Fig. 9). If we keep on dividing strips in this manner, we may end up in a situation where many strips will be created with narrow strips. Under this situation, even if we acquire large number strips, the total area of the regions acquired may remain small. To avoid this situation, we must control the number of narrow strips

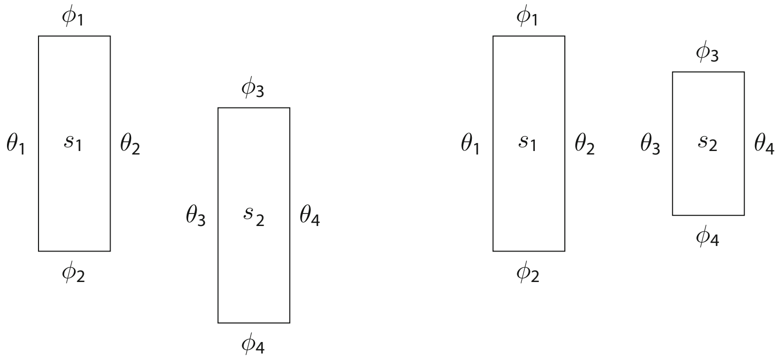


Fig. 7 Two situations where two strips will conflict with each other

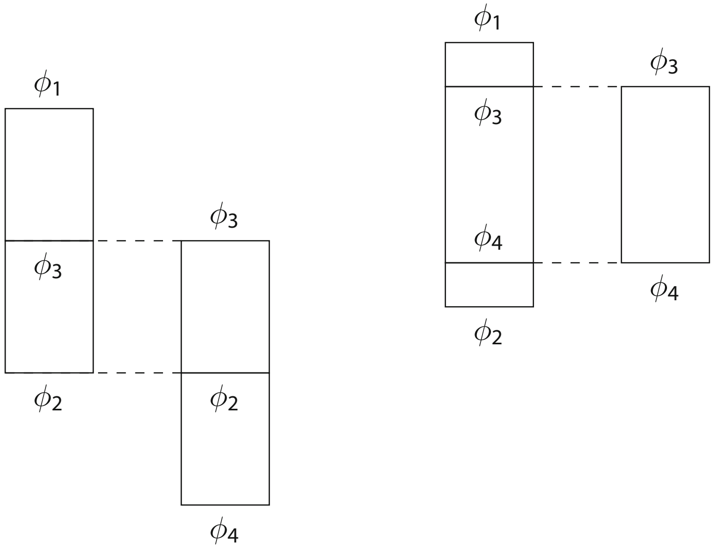


Fig. 8 Two situations where the large strips are cut into small ones

created. For a beam b, if w_b is the maximum width of an image possible using the beam b, then we will divide a strip into narrower strips only if the new strip created entirely to the right of the track has width at least $\frac{3}{4}w_b$.

7 Computational Study

We conducted a series of experiments in order to evaluate our various strategies of splitting the large strips and modeling the piecewise linear function. The piecewise linear function used for each request in the objective function is shown in Fig. 4.

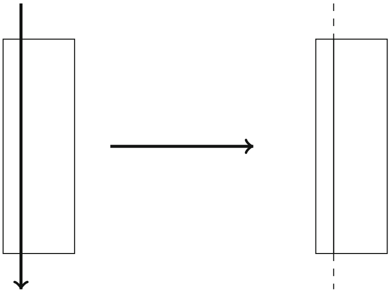


Fig. 9 Division of a wide strip into narrow strips if the strip is not completely to the *right* of the satellite's ground track. The *dark arrow* shows the satellite's ground track. After the division we get two new strips

Table 1 The number of acquired strips and average and maximum acquisition times (in minutes) for the instances with large strips, narrow strips and small strips

	Large strips			Narrow strips			Small strips		
# day	S.No.	Avg.	Max.	S.No.	Avg.	Max.	S.No.	Avg.	Max.
50	354	7.38	15	417	7.56	15.16	2045	10.16	17.34
100	381	8.24	14.36	438	8.31	13.6	2129	10.4	15.47
150	360	7.47	15.2	413	7.69	15.4	1995	9.46	16.62
200	344	7.28	14.8	398	7.57	13.9	1982	9.23	15.87
250	355	7.57	16.0	402	6.76	14.3	1862	9.65	18.23
300	336	8.3	12.9	385	8.6	13.7	1838	11.04	15.79
350	314	7.99	13.4	365	8.12	14	1751	11.01	16.31

Table 1 shows the results obtained from CPLEX for 7 test instances, each instance corresponding to 1 day of year. In each instance we considered all the requests whose time window overlaps with the day that is being considered. The number of image acquisition requests are around 400 in each instance. The results are provided for instances generated with large, small and narrow strips. We report the number of strips that are acquired in each problem instance and the average and maximum acquisition times of the satellites in each of the three cases. There are around 1250 strips in instances with large strips, around 1600 strips in instances with narrow strips and around 9500 strips in instances with small strips. Each strip has 4–5 image acquisition opportunities on average. The average and maximum acquisition times are in minutes. It can be observed that the utilisation of satellites resources is highest if small strips are used and lowest if large strips are used. Each instance is solved to optimality within few seconds of running time. Instances with large strips were of the smaller size and for them CPU time to reach optimality was less than a second, whereas the instances with small strips were of larger size and took around 10 s to reach optimality.

In Table 2, we report the number and total area of requests that are completely acquired (in km^2) out of those requests that have their deadline on the given day. We report the results for the instances with large, narrow and small strips. Except for the day 250, it can be observed that the total area acquired is larger with small strips as compared to the other two strategies. However, the number of requests that are

Table 2 The number of requests that are completely acquired out of all the requests whose deadline is the given day and the total area acquired of these requests for the instances with large, narrow and small strips

	Large strips		Narrow strips		Small strips	
# day	R.No.	Total area	R.No.	Total area	R.No.	Total area
50	58	14,640,986	54	14,183,614	65	14,955,658
100	62	13,647,071	58	14,351,111	72	16,109,986
150	62	13,329,081	61	13,362,688	72	13,434,589
200	43	14,540,530	40	14,705,978	50	15,276,139
250	36	13,452,578	34	13,467,191	49	13,321,412
300	45	11,040,703	45	11,075,610	53	11,549,353
350	45	11,009,681	39	11,096,491	49	11,466,283

Total area is measured in km^2

Table 3 Table showing the number of strips acquired, the average and maximum acquisition times of the satellites, CPLEX run time in seconds for the instances with both narrow and small strips

	Narrow and small strips			
# day	S.No.	Avg.	Max.	Run time (s)
50	2190	10.49	16.66	63
100	2330	10.87	16.27	21.6
150	2096	9.71	15.94	16
200	2104	9.57	16.69	14
250	2040	10.23	18.19	505.5
300	1912	10.81	16.08	73.1
350	1910	11.38	16.64	14.3

completely acquired is greater with small strips than with the other two strategies. This shows that our model favors the complete acquisition of requests over partial acquisition of several requests. It can also be observed that even though the total number of completely acquired requests using narrow strips does not increase as compared to using large strips, total area acquired increases in most of the instances.

Table 3 shows the number of strips acquired, average acquisition times and maximum acquisition times of the satellites for instances where the strips are both small and narrow. We can see that the average acquisition times for the satellites is more in this case than with large, narrow or small strips. Except for the day 300, the total area of acquisition is greater with both small and narrow strips than only with small strips, while the total number of completely acquired requests remain almost the same with both the strategies. This means that the satellite resources are used more efficiently with both narrow and small strips. The total number of strips generated in each instance roughly range between $11,000$ and $13,000$ and the total number of opportunities lie in between $35,000$ and $50,000$. With around 66 % of the requests lying in Canada, the number of conflicts in opportunities are in the order of around $90,000$ conflicts per problem instance.

Table 4 shows the number of completely acquired requests that end on the given day and the total acquired area of requests that end on the given day for the instances

Table 4 The total area and the number of requests that are completely acquired out of all the requests whose deadline is the given day. We show four different lengths of planning horizons for each day. For example, the 4 Day Plan for Day 50 begins on day 47 and ends on day 50. Total area is measured in km^2

	1 Day Plan		2 Day Plan		3 Day Plan		4 Day Plan	
Day	R.No.	Total area	R.No.	Total area	R.No.	Total area	R.No.	Total area
50	60	14,994,647	59	16,235,955	58	17,163,090	56	18,568,120
100	72	16,157,898	64	19,885,732	59	21,039,840	66	21,446,711
150	73	13,491,978	73	14,684,035	76	14,879,687	76	14,913,424
200	50	15,344,293	51	17,136,993	55	17,763,900	55	18,044,350
250	45	13,629,296	48	15,611,762	49	16,105,064	48	16,570,289
300	50	11,471,757	55	12,546,429	50	13,156,272	52	13,698,709
350	46	11,640,585	53	12,474,465	56	12,931,489	49	13,628,414

with small and narrow strips. We show the results for planning horizons of length 1, 2, 3 and 4 days. For a planning horizon of length d and a given day d', we sequentially solve the problem from day $d' - d + 1$ till day d'. It can be seen that as the length of the planning horizon increases, the total acquired area of requests that end on the given day increases considerably while the number of requests that are completely acquired remains almost the same. Instances with 4-day planning horizon were solved to optimality within 20 min of running time.

8 Conclusions

In this work, we studied the image acquisition scheduling problem for satellite constellations. An integer programming model was presented and solved using a commercial MIP solver. We conducted computational experiments for the case study of the Canadian RADARSAT Constellation Mission (RCM). We suggested several pre-processing techniques that increased the efficiency of the system. We also proposed two piece-wise linear objective functions to model the preference for the completion of already partially served requests. These objective functions have potential to increase the possibility of fully serving even the large-area requests of higher priority—a desirable capability in the satellite industry. The results of the computational study on pseudo-real-world instances were encouraging achieving the 1-day image acquisition schedules within several minutes and achieving the 4-day image acquisition schedules within 20 min for around 400 requests and 1000 requests, respectively, with tens of thousands of strips and opportunities.

Acknowledgements This research work was supported by an NSERC CRD grant awarded to Abraham P. Punnen with the MDA Systems Ltd. as the collaborating industrial partner.

References

1. Bensana, E., Verfaille, G., Agnese, J., Bataille, N., Blumestein, D.: Exact and inexact methods for the daily management of an earth observation satellite. In: International Symposium on Space Mission Operations and Ground Data Systems, Munich (1996)
2. Bianchessi, N., Cordeau, J., Desrosiers, J., Laporte, G.: A heuristic for the multi-satellite, multi-orbit and multi-user management of Earth observation satellites. Eur. J. Oper. Res. **177**(2), 750–762 (2007)
3. Bomze, I., Budinich, M., Pardolos, P., Pelillo, M.: The maximum clique problem. In: Du, D., Pardolos, P.M. (eds.) Handbook of Combinatorial Optimization, pp. 1–74. Springer, New York (1999)
4. Cordeau, J.-F., Laporte, G.: Maximizing the value of an earth observation satellite orbit. J. Oper. Res. Soc. **56**(8), 962–968 (2005)
5. Cormen, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. MIT, Cambridge, MA (1990)
6. 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)
7. Karapetyan, D., Mitrovic-Minic, S., Malladi, K., Punnen, A.: Satellite downlink scheduling problem: a case study. Omega **53**, 115–123 (2015)
8. Lemaître, M., Verfaillie, G., Jouhaud, F., Lachiver, J.-M., Bataille, N.: Selecting and scheduling observations of agile satellites. Aerosp. Sci. Technol. **6**, 367–381 (2002)
9. Li, J., Yao, F., Bai, B., He, R.: A decomposition-based algorithm for imaging satellites scheduling problem. In: International Conference on Information Engineering and Computer Science, Wuhan (2009)
10. Li, H.-L., Lu, H.-C., Huang, C.-H., Hu, N.-Z.: A superior representation method for piecewise linear functions. INFORMS J. Comput. **21**(2), 314–321 (2009)
11. Lin, M.-H., Carlsson, J., Ge, D., Shi, J.: A review of piecewise linearization methods. Math. Probl. Eng. **2013**, 1–8 (2013)
12. Malladi, K.T.: Cluster restricted maximum weight clique problem and linkages with satellite image acquisition scheduling. Master of Science Thesis, Simon Fraser University, Burnaby (2014)
13. Tangpattanakul, P., Jozefowiez, N., Lopez, P.: Multi-objective optimization for selecting and scheduling observations by agile earth observing satellites. In: Coello, C.A., Cutello, V., Deb, K., Forrest, S., Nicosia, G., Pavone, M. (eds.) Parallel Problem Solving from Nature - PPSN XII, pp. 112–121. Springer, Berlin, Heidelberg (2012)
14. Vielma, J., Nemhauser, G.: Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. Math. Program. **128**(1), 49–72 (2011)
15. Vielma, J., Ahmed, S., Nemhauser, G.: A note on "A Superior Representation Method for Piecewise Linear Functions". INFORMS J. Comput. **22**(3), 493–497 (2010)
16. Wang, P., Reinelt, G., Gao, P., Tan, Y.: A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation. Comput. Ind. Eng. **61**(2), 322–335 (2011)
17. Williams, H.: Model Building in Mathematical Programming, 5th edn. Wiley, New York (2013)

Re-entry Test Vehicle Configuration Selection and Analysis

Erwin Mooij

Abstract A small and low-cost re-entry vehicle can be a good means for doing hypersonic research, testing new heat-resistant materials, and qualifying newly developed subsystems in a realistic environment. To establish the optimal vehicle shape a response-surface methodology using design-of-experiments techniques is proposed. With these techniques the effects of changing several geometric design parameters in an 'all-at-the-same-time' approach can be studied, instead of the more traditional 'one-at-a-time' approach. Each of the design iterations includes an aerodynamic analysis based on the Modified Newtonian method and a three-degrees-of-freedom trajectory analysis. Generating response surfaces for each of the performance indices and optimising them with a multi-objective optimisation method, a set of geometric parameters is found that gives the best alternative for each of the performance indices. Two fundamentally different vehicle shapes are considered, i.e., one based on a trapezoidal cross section and a sharp, water-cooled nose, for an increased lift-to-drag ratio, and one being a blunted bi-cone that is simple to manufacture, has good stability properties and good potentials for various aerodynamic and material experiments. The developed methodology leads to significant insight in the design space and provides sub-optimal vehicle shapes at a limited computational cost. It may serve as a good starting point for more detailed analysis of a sub-region of the original design space.

Keywords Computer-supported design • Design of experiments • Re-entry systems • Conceptual design • Vehicle-shape optimization

1 Introduction

For the development of reusable launchers, new technology has to be developed and tested in a hypersonic environment that cannot be reproduced in ground-based facilities. Not only is it impossible to simultaneously obtain all conditions occurring

E. Mooij (✉)
Faculty of Aerospace Engineering, Delft University of Technology, P.O. Box 5058,
2600 GB Delft, The Netherlands
e-mail: e.mooij@tudelft.nl

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization
and Its Applications 114, DOI 10.1007/978-3-319-41508-6_8

during a hypersonic flight in these facilities, but also the measurement times are very short, typically in the order of milliseconds. Small re-entry vehicles can fulfill the need of hypersonic experiments supporting this technology development. Such experiments concern aerodynamic phenomena to expand the hypersonic database for verification of software, material tests in the chemically reactive hypersonic flow, tests of instrumentation, new guidance, navigation, and control (GNC) concepts and many other urgent experiments and tests.

Flight experiments in the hypersonic regime have an ongoing interest. Typical examples are the German Sharp Edge Flight Experiment (SHEFEX) missions [21, 29], ESA's Intermediate eXperimental Vehicle (IXV) [20], and DARPA's Hypersonic Test Vehicles (HTV-2) [28]. In the conceptual design phase, to establish the best aerodynamic shape of these vehicles that meets all the requirements, a trade-off between several concepts is needed.[1] To accurately model each aspect and to do a full-blown numerical optimization is at that early stage of the project not a wise step to take, considering the time and money involved. What one wants to do is a quick screening of a limited number of options to see in what direction the design should go, such that design efforts can be concentrated. Of course, when the screening process shows that due to non-linearities in the system's behaviour the number of design points is by no means sufficient, a more global approach should be pursued. Examples of such a global-optimization approach are the continuous shape optimization of winged entry vehicles [4], and the robust multi-disciplinary optimization of unmanned entry capsules [18, 19].

For a constrained system, arbitrary variation of design parameters generally does not lead to a feasible solution, let alone an optimal one. It may thus be clear that a more systematic design approach is needed to generate the best conceptual design. It is the objective of this study to present a methodology to investigate a wide range of possible shapes and to find the most promising one for a re-entry test vehicle. This vehicle would be the best compromise with a minimum effort and without using complex time-consuming design tools, and thus saving time and costs during the preliminary design. Once the general direction of the design has been established the outcome can be used as input to the next design phase (not presented here), such that all effort can be concentrated on the refinement of the design using more detailed design tools. The approach that we will follow is one from the field of *design of experiments*, in combination with a *response-surface methodology* from the field of regression analysis.

To illustrate the approach we will look at the aerodynamic design of two (hypersonic) flight-test vehicles with fundamentally different base shapes, as discussed by Mooij et al. [11] and Sudmeijer and Mooij [26]. The first vehicle assumes the availability of a low-cost launch facility that enables a maximum Mach number between 10 and 12, a set of mission requirements, and the definition of the experiments and corresponding equipment. The initial vehicle geometry is based on

[1] In fact, this could apply to the design of any complex (sub-)system and could be extended to detailed design as well.

a trapezoidal cross section and a sharp, water-cooled nose to increase the lift-to-drag ratio, a reaction-control system and four flaps mounted at its base for aerodynamic control. Several performance indices related to controllability, mission constraints and experiment performance (e.g., duration of flight above Mach 8) will be taken into account.

For the second generic module shape, a blunted bi-cone has been selected that is simple to manufacture, has good stability properties and potential for various aerodynamic and material experiments. The cooling of the spherical nose is based on nucleate pool boiling of water. For this second vehicle, the focus will be on the thermal loading of certain elements of the vehicle while doing a ballistic entry. The entry conditions come from a (sub-orbital) launch, and the vehicle reaches maximum Mach numbers close to 20. Variations in entry conditions and vehicle mass are included in an integral approach combined with the shape variations to establish the best shape for different mission types.

The layout of this chapter is as follows. In Sect. 2, the parametric design and analysis method is discussed. This methodology is successively applied in two test cases. Section 3 focuses on the aerodynamic design of a test vehicle with trapezoidal cross section, whereas in Sect. 4 the integrated shape and trajectory analysis of a biconic entry capsule is discussed. Section 5 concludes this chapter with some final remarks.

2 Parametric Design and Analysis

2.1 Design of Experiments

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 behavior 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. Furthermore, the design should be non-collapsing. 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 this means that effectively the same point is evaluated twice, and for deterministic simulation models this is not a desirable situation. Therefore, two design points should not share any coordinate values when it is not known a-priori which dimensions are important. However, from a preliminary analysis we know this situation will not occur in the treated examples. It could be useful, though, to keep this in mind for future reference when the design parameters change.

Table 1 Orthogonal array L_8 with seven factors (A–G) on two levels; "−1" represents the normalized 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

Straightforward factorial design, i.e., varying one parameter at a time and executing all combinations, rapidly leads to a large number of simulations.[2] A fractional factorial method was found in the field of design and production-process optimisation, called the Taguchi Method [17, 27]. This method 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* [17]. 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 [27] 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 1 (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 four 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. In a full factorial design 2^7 (= 128) experiments would be required. Note that the L_8 design is non-collapsing.

The use of orthogonal arrays is based on application of the so-called *D-optimality* criterion, which is easiest explained with the definition of a response surface (also known as a least-squares fit) of a performance index $\eta = \mathbf{X}\boldsymbol{\beta}$, where $\mathbf{X}$ is the regression matrix and $\boldsymbol{\beta}$ is the vector with regression coefficients. D-optimality is based on the notion that the experimental design should be chosen so as to achieve certain properties in the moment matrix $\mathbf{M}$, which is proportional to $\mathbf{X}^T\mathbf{X}$. Since the inverse of $\mathbf{M}$ contains variances and covariances of the regression coefficients $\boldsymbol{\beta}$, it means that the determinant of $\mathbf{X}^T\mathbf{X}$ is inversely proportional to the square of the

[2] Variation of k parameters with two (three) possible values, also called levels, results in a total of 2^k (3^k) combinations.

volume of the confidence region on the regression coefficients [13]. The application of orthogonal arrays to define $\mathbf{X}$ will maximise the determinant of $\mathbf{X}^T\mathbf{X}$, which means that for a given definition of the response surface this will result in the most accurate estimate of the regression coefficients.

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 [17]. For the array given in Table 1, 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.

In case the so-called main-factor effects are much larger than the effect of the interaction, the latter can usually be ignored, though. But if the system under study is not known one cannot say beforehand which of the interactions between the input parameters do not have an effect on a selected response. If two factors A and B are varied at the same time, the interaction AxB can be studied by not assigning any factor to the appropriate column as specified by the related interaction table. However, when a factor C is assigned to that column, the corresponding factor variation will be influenced or *confounded* by the interaction AxB. When for a two-level array the factors are assigned to the columns such that all the two-level interactions are free of confounding with other two-level interactions and main effects, this results in a so-called *Resolution-V* array, see also [5]. Basically, this is required when no information about the interactions is available.

To study only linear effects, factor variations over two levels will suffice. In case one is also interested in quadratic effects, variation over three levels is required, which leads to larger orthogonal arrays. However, also another approach is possible. After conducting a matrix experiment, it is possible to compute a response surface, i.e., a polynomial function in one or more dimensions that describes the relation between a response and the applied factors. To study quadratic effects, a second-order response surface may be used:

$$\eta = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{ij} x_i x_j \tag{1}$$

for which the coefficients $\boldsymbol{\beta}$ 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.

One particular and efficient design to generate the responses to solve the coefficients of Eq. (1) was introduced by Box and Wilson [2], i.e., the (empirical) class of Central Composite Designs (CCDs):

1. a complete (or fraction of a) *Resolution-V* 2^k factorial design, where the factor levels are coded to the usual -1, $+1$ values (the *factorial portion* of the design),
2. n_0 centre points ($n_0 \geq 1$), and
3. two axial points on the axis of each design variable at a distance α from the design centre (the *axial portion* of the design). To determine α, let n_{tot} be the total number of designs and n_f the corresponding number of the factorial portion. For a so-called orthogonal design, i.e., a design for which the variance of the coefficient estimates is minimised, Khuri and Cornell [6] state that

$$\alpha = \sqrt{\frac{\sqrt{n_{tot} n_f} - n_f}{2}}$$

Montgomery [8] mentions that a CCD is made *rotatable*,[3] the preferred class of second-order response-surface designs, by choosing $\alpha = n_f^{0.25}$. Combining both expressions would lead to $n_0 \approx 4\sqrt{n_f}+4-2k$ to make the design both orthogonal *and* rotatable.

Since many engineering problems exhibit a second-order behaviour, i.e., they contain linear, quadratic and first-order interaction effects, this design can effectively be used in many situations.

As an example, consider a system design with three ($k = 3$) design variables. The factorial portion is taken as a full factorial design, such that $n_f = 2^3 = 8$. For a rotatable design, $\alpha = n_f^{0.25} = 1.6818$, and to make the design orthogonal as well, $n_0 = 4\sqrt{n_f}+4-2k = 9.3$, which is rounded off to $n_0 = 9$. For three design variables with nominal values of $x_1 = 5$, $x_2 = 3$ and $x_3 = -8$, and corresponding ranges of $\Delta x_1 = \pm 2$, $\Delta x_2 = \pm 1$ and $\Delta x_3 = \pm 1.5$, this means that the corresponding minimum and maximum values for these parameters—the -1 and $+1$ settings in the orthogonal arrays—would be: $x_{1,min} = 3$, $x_{1,max} = 7$, $x_{2,min} = 2$, $x_{1,max} = 4$, $x_{3,min} = -9.5$, and $x_{3,max} = 6.5$. The axial points, however, are at a greater distance (at $\pm 1.6818 \Delta x$) from the nominal design values. For x_1, for instance, this would mean that $x_{1,\alpha,min} = 1.6364$ and $x_{1,\alpha,max} = 8.2728$.

A typical example of a central composite design from literature is the rocket-powered single-stage space-plane configuration selection and design [25], as well as its propulsion-system optimization [24], and such a CCD will also be used here for the aerodynamic-shape selection. In general, such a design requires fewer experiments than using a stand-alone three-level orthogonal array.

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:

[3] A rotatable design is the most effective from a variance point-of-view, and all points at the same radial distance from the center point have the same magnitude of prediction error (uniformity of variance).

$$\bar{y} = \frac{1}{N}\sum_{j=1}^{N} y_j = \frac{T}{N} \quad \sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2 \tag{2}$$

where $T = \sum_{j=1}^{N} y_j$ is the total sum. 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.

$$S_T = \sum_{i=1}^{N}(y_i - \bar{y})^2 \tag{3}$$

In case the response data are fitted by a response surface, the mean value is represented by β_0, see Eq. (1). The relative values of the other coefficients give the sensitivity of this mean response to a variation in the individual factors and interactions. Of course it should be clear that the computed response surface has to fit the responses well. The *residual* r_i is the difference between the measured and predicted response, i.e., $r_i = y_i - \hat{y}_i$. The *predicted response* $\hat{y}_i, i = 1, \cdots, N$, is the response value computed with the response surface using the same factor combination that resulted in the measured value. Three sums of squares can thus be defined, the total sum of squares (S_T), given by Eq. (3), as well as the sum of squares due to regression, S_R, and the sum of squares unaccounted for by the response surface, S_E:

$$S_R = \sum_{i=1}^{N} (\hat{y}_i - \bar{y})^2, \quad S_E = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{4}$$

with the mean value of the observed responses, $\bar{y}$, given by Eq. (2).

A first indication of the goodness of fit is given by the *coefficient of determination*, R^2, defined by the ratio of the sum of squares of the residual and the total sum of squares:

$$R^2 = \frac{S_R}{S_T} \tag{5}$$

although this does not say anything about the response surface in between the nodal points. In some cases the fit *through* the response values is very good (with R^2 close to its maximum of 1), but the response surface can exhibit oscillatory behaviour in between.

2.2 Vehicle-Design Methodology

The design approach followed is common to both cases to be discussed hereafter. To begin with, one needs a set of (generic) mission and system requirements that may drive the design process of flight-test vehicles. In general, they are derived from a set of user requirements. These can be divided into two essentially different groups, i.e., resulting from the use of (1) the vehicle itself to do tests in a hypersonic environment that cannot be done in ground-based facilities, and (2) the internal volume of the vehicle to do experiments in a better micro-gravity environment than achieved in, for instance, a parabolic flight. The user requirements that will lead to the set up of mission requirements can be categorised in the following sub-sets.

1. *External-flow measurement experiments and CFD-code validation.*

   (a) Investigation of shock-layer properties, i.e., gas composition, density, pressure and temperature,
   (b) Investigation of shock-wave/boundary layer interaction, boundary layer separation and reattachment, and
   (c) Investigation of boundary-layer transition, determination of permissible surface roughness and waviness.

   To allow for the above experiments, the following mission requirements can be set-up. To assure the occurrence of dissociation the chemical-kinetics parameter $\rho_\infty R_N$ (product of free-stream air density and nose radius of the vehicle), and the relative free-stream velocity V_∞ should be simultaneously covering $\rho_\infty R_N \in [0.1,10]\,\text{g/m}^2$ and $V_\infty \geq 2.5\,\text{km/s}$. In addition, the aerodynamic performance of hypersonic vehicles is characterised using the numbers of Reynolds and Mach, Re_∞ and M_∞. For valuable data the flight test should be executed in the following flight ranges: $Re_\infty \geq 2.6\times10^5 L_{ref}$ and $M_\infty \geq 5$. Here, L_{ref} is a characteristic reference length.

2. *Vehicle navigation and control experiments.*

   (a) Testing of control surfaces, i.e., to provide data to validate and improve prediction for control-flap effectiveness, study the influence of gaps on heat loads and flap performance, and study hot-spot effects due to viscous interaction (specifically in flow separation and reattachment regimes),[4]
   (b) Investigation on RCS, i.e., to improve prediction and efficiency for the performance of RCS, to study the interaction between the plume and the surrounding flow field, and verification of the RCS-thruster,
   (c) Navigation and measurement system (combination of global positioning system (GPS) and inertial measurement unit (IMU)), i.e., the investigation of advanced GPS for IMU update, investigation of accuracy of the GPS in

[4] To test the characteristics of a flap in the hypersonic flow, the Mach number should be larger than 5 at an altitude of about 60 km (typical re-entry trajectory).

high velocities during re-entry, to study the integration of GPS antennas in the thermal protection system (TPS) and the antenna gain, and to study the ground-link communication black-out during re-entry,

(d) In-flight assessment of GNC software performance, and

(e) Testing instrumentation, i.e., the verification of sensor accuracy on the real flight environment, verification of sensor reusability, investigation on smart and small micro sensor applied in extreme flow conditions, and testing of a new type of optical air-data system.

3. *Micro-gravity experiments*. The micro-g phase is defined as that part of the trajectory where the residual acceleration along all axes is less than 10^{-4} g. In general, the experiment module shall not generate forces leading to higher accelerations. Fields for which these measurements could be relevant are:

   (a) Life science, where the phenomenon itself is the most important element to study; these experiments always have a statistical nature, and

   (b) Materials and fluid science, for which a lot of theory has to be verified; these experiments are always very dedicated and unique.

Of course, there may be specific requirements for particular sub-systems, which should lead to an optimal design solution. Examples are the nose-cooling system, part of the TPS of the test vehicle discussed in Sect. 3, or the analysis of TPS thermal loads covered in Sect. 4.

The general design and analysis approach is now as follows. For the chosen family of configurations we derive analytical expressions for the vehicle shape and identify the geometrical parameters that we want to vary. The variation of the selected parameters will be done according to a central composite design, as discussed in Sect. 2.1, so we determine the ranges and assign the parameters to the corresponding columns of the orthogonal array. For each of the design configurations following from the CCD, first a surface mesh has to be generated that serves as input to an aerodynamic engineering code based on modified Newtonian flow for hypersonic speed. This code is used to build an aerodynamic database for an adequate Mach and—if required—angle-of-attack range. Subsequently, this database is linked with a flight-simulation software[5] that can compute both lifting and ballistic three-degrees-of-freedom trajectories. The output of the flight-simulation software is processed to derive the performance indices and to serve as input for the computation of the response surfaces. This output is also used as input to dedicated models used for the sub-system analyses mentioned above.

It is stressed that this methodology will not work for very complex shapes, because then we will not be able to create an analytical description of the geometry. In that case the shape variation becomes far more complex: one can think of defining the surface mesh by means of, for instance, Hermite polynomials, as was done by

[5] The flight-dynamics model has been developed for a rotating, flattened Earth; the atmosphere model is the United States Standard Atmosphere (1976), and the gravitational model is a central field model with a correction for the Earth's flattening.

Dirkx and Mooij [4]. However, the downside is the large number of parameters defining the geometry, which will prevent an effective use of design of experiments. Dirkx and Mooij [4] applied an evolutionary algorithm for the shape variation, at the expense of a large increase in CPU time.

3 Shape Optimization

For the first design case we consider the conceptual design of a small low-cost re-entry testbed for hypersonic experiments, designated *Hyperion-2*. This vehicle is an improved version of an earlier concept [10]. As such, the arrangement of the control flaps has been changed to prevent undesirable roll/yaw coupling of the previous design. It was also observed that the dynamic stability parameter, $C_{n_{\beta,dyn}}$, was too small causing an undesirable sensitivity for Dutch roll, indeed a classical problem for lifting bodies. This problem has been considerably reduced by an angle of dihedral (a V-shape) of the windward side of the module, see also Fig. 1.

Another important modification is the reduction of the nose radius R_N to substantially reduce the drag and subsequently increase the flight time at $M > 8$. As a consequence the stagnation heat flux on the nose increased and has led to a water-cooled nose to keep the temperature of the nose below 1300 °C. This is the temperature limit of the selected material PM1000, a metal oxide-dispersion strengthened (ODS) superalloy.

The flap effectiveness has been improved and the undesired roll/yaw coupling of the previous model could be avoided by applying four rather than three control flaps. The two bottom flaps are used for roll and pitch control, the two upper flaps are for yaw control and trim, but do not induce a roll moment when deflected.

Concerning the choice of launcher, sounding rockets do not reach orbital velocity, but for hypersonic experiments only a Mach number in the order of 10 is required for a reasonable amount of time. Most sounding rockets do not meet this requirement, but (at the time of the study) the Brazilian VS-40 is an available non-military rocket

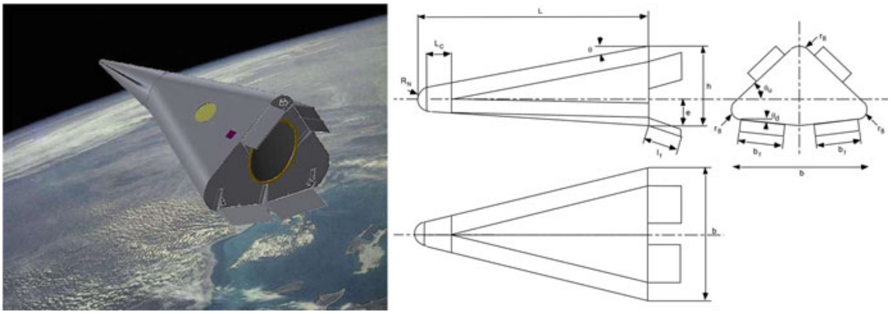


Fig. 1 *Hyperion-2*: artist impression (*left*) and generic geometry (*right*)

with sufficient performance.[6] So for the conceptual design of the present *Hyperion* re-entry module the VS-40 has been selected as launcher, with as a primary goal that the module should achieve a trajectory on which it reaches at least Mach 8, at an altitude, which resembles the actual re-entry trajectory. The burnout conditions of the VS-40 are such that at an altitude of $h = 120\,\text{km}$ the corresponding velocity relative to the atmosphere is $V = 3300\,\text{m/s}$, whereas the flight-path angle is $\gamma_0 = -10°$. It has to be confirmed, though, that these conditions are suitable for doing hypersonic experiments.

With all these design changes, we want to know the following. Can the proposed TPS protect the vehicle from the induced thermal environment? Can the dynamic stability be improved? Do we get sufficient measurement time in the Mach range of interest? To answer these questions a number of response objectives will be defined, and variation of the baseline shape will be studied to see the effect on these responses. The mission considered is one starting from the initial conditions mentioned above and entering the atmosphere at the configuration for maximum L/D until the flight-path angle γ is zero, and successively a flight with zero flight-path-angle rate, $\dot{\gamma}$, that is forced on the vehicle by angle-of-attack modulation. Conversion of the commanded γ to a commanded α follows directly from the related equation of motion, as discussed in, for instance [9]. A detailed discussion on the complete mission of *Hyperion-2* is given by Mooij et al. [12].

3.1 Vehicle Description

As mentioned earlier, the vehicle's nose TPS is based on water cooling. The water takes up the heat and starts boiling after reaching the evaporation temperature. During re-entry, the water damp will be forced towards the tank rear side due to the deceleration. Here, a valve is mounted, which is set to open at 5 bar. A tube leads the water damp towards the vehicle back side, where it is vented in the base flow of the vehicle. This concept prevents the vehicle boundary layer from being polluted with cooling mass, which would be undesirable for many of the anticipated experiments. The influence on the vehicle motion of the coolant dumping is considered to be controllable, due to the small cooling mass flow involved.

The nose heat flux, $q_{c,nose}$, is proportional to the inverse square root of the nose radius. However, since the nose area simultaneously decreases to the second power of the nose radius, the total heat load for the spherical nose decreases. Furthermore, the heat flux near the stagnation point decreases approximately with the cosine of the spherical nose angle before decreasing with the inverse running length from the stagnation point a bit further [23]. This already indicates that for a small nose radius

[6] The VS-40 has been successfully used to launch SHEFEX-2, a DLR-operated re-entry vehicle for hypersonic flight experiments [29].

the conical part will require water cooling as well to avoid reaching the maximum operating temperature of PM1000/PM2000.

An engineering-analysis model is established to quantify the required water coolant mass during the re-entry orbit of *Hyperion*. It is based on heat transfer methods of the stagnation point, laminar and turbulent layer flow after [31], which involves the computation of the boundary layer momentum-thickness. Similar methods after [1, 7] are modelled as well for reasons of verification of the analysis code.

The above models require knowledge about the boundary-layer edge conditions, for which two different methods are implemented, i.e., the so-called normal shock/isentropic expansion model after [1], and one based on the modified Newton method combined with a simple flow-angle-deflection velocity model. The former covers the stagnation point region more realistic, whereas the latter has a better representation of the decreased total pressure loss in the aft region of the nose, where the entropy layer becomes small compared to the boundary-layer thickness. The laminar heat flux of the latter model is slightly higher, but more important, due to earlier transition on the vehicle the modified Newton model gives an increased surface-integrated heat load compared to the normal shock/isentropic expansion model.

The outer geometry of the re-entry vehicle that served as a generic baseline for the aerodynamic design is build up with twelve independent parameters, i.e., eight for the body and three for the control flaps, that completely define the geometry (Fig. 1). Out of the eight body parameters, five parameters that have a major influence on the aerodynamic performance have been selected for variation, namely: the semi-cone angle, θ_c, the side angle of the upper panels, θ_u, the body-axis parameter, μ, the rounded radius of the base, r_B, and the angle of dihedral of the bottom, θ_l. The aforementioned body-axis parameter defines the position of the body axis, located along the X-axis in the plane of symmetry (X-Z plane). The body axis is fixed in the triangle defined by the centres of the rounded radii at the base by the fraction μ of the height of the triangle (and has, therefore, always a value between 0 and 1).

3.2 Aerodynamic Design

As mentioned, the five selected design parameters will be varied according to a CCD. For the factorial portion of the design, Taguchi's L_{16} array will be used, which is *Resolution-V* for a maximum of five parameters, column assignment is 1, 2, 4, 8 and 15 [5]. The axial parameter α will be chosen such that the design is orthogonal, i.e., a design for which the variance of the coefficient estimates is minimised. This means, for $n_0 = 1$ centre point and 16 designs (L_{16}) in the factorial portion, that $\alpha = 1.5467$. The total number of designs comes to $16+5\times2+1 = 27$. The nominal parameter settings and their variations are chosen to be representative to obtain a broad family of designs, see also Table 2. Since the range in the L_{16} array

Table 2 Parameter variations in CCD

Factor	Nominal value	Maximum range L_{16}	Maximum range axial ($\alpha = 1.5467$)
θ_c	13.0°	±0.647°	±1.0°
θ_u	50.0°	±6.465°	±10.0°
μ	0.4	±0.129	±0.2
r_B	0.2 m	±0.065 m	±0.1 m
θ_l	5.0°	±3.233°	±5.0°

is given by ±1, this means that the maximum physical range should be divided by 1.5467 to get the corresponding values. Note that the other three parameters are dependent on the five selected ones, and will change accordingly during the parameter variation. The vehicle configuration cannot change to extra-proportional dimensions, so constraints have been imposed on the vehicle length ($L_{max} = 2.7\,\text{m}$), the vehicle width ($b_{max} = 1.7\,\text{m}$), and the vehicle height ($h_{max} = 1.0\,\text{m}$). If one of the constraints is met during the parameter variation, the height is adjusted such that the constraints are no longer violated.

The performance indices used in this study can be divided into several categories, i.e.,

1. Flight experiments:
   (a) The altitude at which the vehicle is flying horizontally, h_{hor},
   (b) Flight time between Mach 10 and 8, $M_{10\rightarrow 8}$
   (c) Flight time between Mach 8 and 6, $M_{8\rightarrow 6}$
   (d) Chemical-kinetics parameter $\rho_\infty R_N$
   (e) Reynolds number at Mach = 8
2. Controllability:

   From analysing the eigenvalues of the characteristic polynomial, it appears that the restoring moment

$$C_{n_{\beta,dyn}} = C_{n_\beta} - C_{l_\beta} \tan\alpha_0 \frac{I_{zz}}{I_{xx}} \tag{6}$$

   is an important index for the sensitivity for Dutch roll [30], and will therefore be selected as objective as well. However, it will only be defined for one flight condition, i.e., Mach = 8 and $\alpha_0 = 5°$. Note that a positive value is required.

3. Trajectory constraints and vehicle related:
   (a) Maximum *g-load*
   (b) Maximum convective heat flux, q_c, for equilibrium wall temperature
   (c) Integrated heat load, Q
   (d) Maximum mass flow of evaporated water, $\dot{m}_{H_2O}$
   (e) Total cooling-water mass, m_{H_2O}

Furthermore, some additional parameters such as the maximum L/D, the initial Mach number at horizontal flight, the total flight time and the final altitude have been defined as auxiliary parameters to analyse the results.

3.3 Results

Evaluating the 27 designs of the CCD as discussed in the previous section[7] a selection of the results (i.e., the performance indices) can be found in Table 3. Unfortunately, it is not possible to discuss each of the results in great detail, so we have to restrict ourselves to a limited number.

It is clear from inspecting Table 3, that there is a large variation in L/D, with a minimum of 0.46 (#12) and a maximum of 2.21 (#5). It seems logical to expect a large variation in the other performance indices too, as can indeed be observed. As an example, $q_{c_{min}}$ is encountered by concept #4 (2255 kW/m^2), whereas $q_{c_{max}}$ is met by concept #5 (5483 kW/m^2). Two remarks: (1) the optimal configurations for L/D and q_c do not correspond, which would lead to a compromise if, for instance, L/D_{max} and $q_{c_{min}}$ should both be met; (2) $q_{c_{max}}$ = 5483 kW/m^2 is very high and leads to very high temperatures. Moreover, the large L/D seems to contradict this high q_c, but we will come back to that later.

Note that for each of the concepts the maximum *g-load*, not listed here, varies between 9.28 (#19) and 10.74 (#5), and these peak values all occur at more or less the same time ($t \approx$ 90–100 s), some 5–10 s before the vehicle starts flying horizontally. Obviously, the constant flight at $\alpha_{L/D_{max}}$ results in a pull-up with significant thermal and mechanical loads. In a next iteration step of the design process, the pull-up manoeuvre should be optimised such that the loads are decreased. On the other hand, the total amount of cooling water for the highest heat flux is 'only' 6.19 kg, partly also because the peak is of a relatively short duration (the cooling-water mass for the other configurations is lower; the variation of this mass is between 1.09 and 7.75 kg, indicating that if no technological problems are encountered, water cooling may be very efficient).

With respect to the chemical kinetics parameter and the Reynolds number (evaluated at Mach = 8), the analysis may be brief. In all 27 cases, $\rho_\infty R_N$ is in the required interval of [0.1,10] g/m^2, whereas the Reynolds number (based on the vehicle length, varying between 1.7 and 2.7 m) easily fulfills the criterion listed in Sect. 2.2. Note that some of the configurations start the horizontal flight at a lower Mach number than 8, hence the '–' (= no value) in Table 3. Obviously, also dt ($M_{10\rightarrow 8}$) is zero ('–') in that case.

[7] We have assumed that each vehicle can be trimmed throughout the flight. Verification of this has shown that by shifting the centre of mass more or less in Z-direction (vertical) this can indeed be achieved. At the moment we do not focus on optimizing the centre-of-mass location, and because the flap contribution to the aerodynamics is in the same range for each configuration, we have ignored this.

Table 3 Results from the central composite design (minimum and maximum values are grey shaded)

	max.	max.					$M_{10\rightarrow 8}$	$M_{8\rightarrow 6}$	M=8	max.	
	L/D	q_c	Q	$C_{n\beta,dyn}$	h_{hor}	$\rho_\infty R_N$	dt	dt	Re	$\dot{m}_{H_2O}$	m_{H_2O}
nr	(−)	(kW/m^2)	(MJ/m^2)	(1/rad)	(km)	(g/m^2)	(s)	(s)	(10^8)	(g/s)	(kg)
01	1.98	3944	301.7	0.686	32.2	0.33	32.0	57.0	1.88	68.9	3.45
02	1.03	2955	140.6	1.034	32.5	0.32	0.2	25.0	1.68	171.8	4.22
03	1.19	3044	155.9	0.526	32.9	0.30	5.0	28.0	1.18	43.7	1.39
04	0.63	2255	87.2	0.828	32.7	0.31	-	-	-	92.7	2.10
05	2.21	5483	432.3	0.916	28.5	0.59	38.8	61.0	3.41	108.6	6.19
06	1.28	3900	199.9	1.277	30.3	0.45	7.6	30.0	2.56	255.1	7.75
07	1.47	4012	236.8	0.657	30.6	0.43	14.3	39.0	1.88	66.4	2.74
08	0.66	2926	109.2	1.061	29.8	0.48	-	1.6	-	170.3	3.58
09	1.71	3652	244.9	0.634	32.5	0.32	21.8	47.0	1.58	60.1	2.56
10	1.00	2676	128.4	0.938	33.5	0.28	-	23.6	-	130.4	3.13
11	1.12	2800	141.4	0.496	33.6	0.27	2.2	27.0	1.02	36.2	1.09
12	0.46	2294	79.5	0.769	30.1	0.47	-	-	-	104.1	2.10
13	2.09	5365	378.4	0.871	28.6	0.59	31.7	52.0	3.25	103.4	5.41
14	1.04	3562	163.8	1.177	30.2	0.46	0.9	24.0	2.24	221.9	6.11
15	1.24	3814	197.7	0.631	30.3	0.45	7.6	30.0	1.68	62.1	2.28
16	0.61	2792	101.5	0.952	29.8	0.49	-	-	-	150.7	3.12
17	1.35	3504	197.2	0.895	31.9	0.35	10.8	35.0	1.79	118.5	4.00
18	1.11	3105	156.0	0.800	32.3	0.34	2.4	28.0	1.47	102.5	2.99
19	1.08	2612	131.9	0.696	34.2	0.24	1.4	26.0	1.10	34.8	1.54
20	1.34	4115	224.8	1.030	29.8	0.48	10.5	35.0	2.32	147.6	5.29
21	1.69	3919	265.0	0.985	31.6	0.37	22.3	48.0	2.13	147.0	5.92
22	0.74	2662	106.8	0.690	31.7	0.36	-	6.8	-	73.6	1.69
23	1.90	4519	301.9	0.551	30.3	0.45	25.7	47.0	2.16	47.1	2.29
24	0.64	2655	99.7	1.076	30.8	0.42	-	0.9	-	188.8	3.75
25	1.29	3410	181.1	0.840	31.9	0.35	7.7	31.0	1.73	110.3	3.57
26	1.15	3429	169.5	0.860	31.3	0.39	3.9	28.0	1.76	112.3	3.34
27	1.22	3415	175.4	0.844	31.6	0.37	6.4	29.0	1.73	111.0	3.59

Finally, the minimum critical flap deflection above which a strong interaction between the shock wave and the boundary layer occurs (and hence creating a hot spot on the control surface which might lead to structural problems due to the high thermal load) is in all cases relatively small: it varies between 1.8° and 2.6°, encountered at the peak dynamic pressure, or, equivalently, the peak *g-load*. This means that at this point in the trajectory, the total flap deflection, i.e., the deflection due to trim plus the increment due to control-system activities for corrective control should be smaller than this critical value. Of course, in general this is true for other

Table 4 Response-surface coefficients for L/D_{max}

Constant	Linear		Interaction		Quadratic	
1.2241	θ_c	-0.0747	$\theta_c \times \theta_u$	-0.0063	θ_c^2	0.0021
	θ_u	0.0906	$\theta_c \times \mu$	0.0088	θ_u^2	-0.0063
	μ	-0.3096	$\theta_c \times r_B$	0.0125	μ^2	-0.0042
	r_B	-0.3972	$\theta_c \times \theta_l$	-0.0063	r_B^2	0.0190
	θ_l	-0.0412	$\theta_u \times \mu$	-0.0200	θ_l^2	-0.0021
			$\theta_u \times r_B$	-0.0338		
			$\theta_u \times \theta_l$	0.0025		
			$\mu \times r_B$	0.0613		
			$\mu \times \theta_l$	0.0050		
			$r_B \times \theta_l$	-0.0013		

points in the trajectory as well, and therefore this δ_{crit} as a function of time should be included in the formulation of the trim algorithm, and the design of the control surfaces and the control system.

After this global discussion of the results, we try to establish a means to find the optimum vehicle configuration that fits the user requirements from Sect. 2.2. By simply picking the best-performing configuration, the first step in the design process could be finalised. However, by using all information available in Table 3, it is possible to optimise the performance even more.

In Sect. 2.1, Eq. (1) defines a response surface, i.e., a description of a performance index as a function of the independent vehicle parameters. Such response surfaces have been computed for each of the performance indices. By optimising these surfaces, the optimum vehicle configuration, as defined by the response surface, can be determined. It is clear that this optimal configuration should be verified by a design cycle, because the response surface is only an approximation with a finite accuracy (up to quadratic terms). Another point of attention is that the applied optimisation algorithm, a truncated Newton method for bounded optimisation developed by Nash [14], finds a local rather than a global optimum.

The first response surface that we discuss here is the one for L/D_{max}. In Table 4, the related coefficients are listed. The constant coefficient (= 1.2241), represents the value of the nominal configuration (all normalised parameters equal to '0', or the nominal values of Table 2). To address the importance of each of the terms, the contribution to the nominal value can be computed if each of the linear, interaction and quadratic terms is either -1 or 1, and the absolute contribution would be checked against 1.2241 (e.g., the contribution of μ would be 25.3 %, the interaction $\mu \times r_B$ is 5 %, and so on). By adding the contribution of each group of coefficients, the linear terms contribute 82.7 %, the interactions 14.3 % and the quadratic 3.1 %. Although individual coefficients may be neglected it is clear that the total group of interactions is too large to ignore. Major linear contributors are μ (25.3 %) and r_B (32.5 %), the three largest interactions are $\mu \times r_B$ (5 %), $\theta_u \times r_B$ (2.8 %) and $\theta_u \times \mu$ (1.6 %), whereas the largest quadratic term is r_B^2 (1.6 %). Note that the so-called

Table 5 Optimal vehicle-configurations (design parameters are grey shaded)

Vehicle parameter	Unit	L/D_{max}	dt_{max}	$C_{n_{\beta,dyn}}$	Multi
Semi-cone angle	(°)	12.000	12.603	12.496	12.000
Angle upper panels	(°)	60.000	60.000	60.000	40.000
Body axis parameter	(−)	0.200	0.200	0.200	0.200
Rounded radius base	(m)	0.100	0.100	0.300	0.238
Angle of dihedral	(°)	5.737	10.000	10.000	0.110
Radius top nosecone	(m)	0.025	0.025	0.025	0.025
Body width base	(m)	0.914	0.957	1.014	1.538
Radius base nose cone	(m)	0.100	0.100	0.300	0.238
Body height base	(m)	0.855	0.924	1.000	0.923
Nose radius	(m)	0.026	0.026	0.026	0.026
Nose length	(m)	0.020	0.020	0.020	0.020
Length conical nose	(m)	0.353	0.335	1.241	1.003
Triangular body length	(m)	2.327	2.345	1.295	1.676
Body length	(m)	2.700	2.700	2.556	2.700

coefficient of determination, i.e., a measure for the goodness of fit, is 99.9 %, indicating a good representation of the 27 data points. Deviations in between data points are of course still possible, hence the need for the verification step in the design cycle.

Optimising the response surface gives a maximum L/D of 2.69, with the corresponding configuration listed in Table 5. While doing a verification design cycle, it was found that L/D_{max} was 2.81, not too much different from the predicted value. Trajectory simulation showed that the peak value $q_{c_{max}}$ increased to the very large value of 7659 kW/m^2, leading to very high nose and flap temperatures (nose cooling to an acceptable level still requires no more than 10 kg of water!). As before, the reason is that despite the fact that the L/D is large, in absolute sense the lift and drag have small values because of the small drag area. Basically, the vehicle is long (L = 2.7 m) and small (b = 0.855 m), which means that it will dive deep into the atmosphere, and will start flying horizontally (h = 25.1 km) at a high speed (Mach = 10.6). On the other hand, because of this high speed and low drag, the time it flies between Mach = 8 and 10 has increased to 64.9 s! On the downside, also the total flight time down to Mach 3 has increased, so the total integrated heat load (0.765 MJ/m^2) may become a problem for the overall structure. This remains to be studied in more detail. Conclusion is that at least the pull-up manoeuvre should be refined as to increase the altitude at which the vehicle flies horizontally.

Finally, a point of attention with respect to the difference between predicted and verified optimum is that the response surface may not predict the response values with equal accuracy in each direction of the independent variables. As we mentioned before, the applied CCD was made orthogonal by selecting an axial parameter of α = 1.5467. By choosing α = 2, the design can be made rotatable, which means that it will be equidirectional. By defining 10 centre points (which, in the case discussed here, would just lead to 10 repetitions of the same design cycle, but this, of course, affects the statistics), the CCD is both orthogonal and rotatable. It remains to be studied whether this will improve the prediction of the optimal values, and, by equal maximum variation of the vehicle parameters, how reducing the range for the L_{16} experiments will influence the results.

The response surface for $dt\,(M_{10\rightarrow 8})$ shows coefficient contributions of 64.6 % (linear), 23.5 % (interactions) and 12.0 % (quadratic), which indicates, with a coefficient of determination close to 100 %, that a full second-order response surface has been the right choice. The computed maximum dt, however, is only 11.4 s, which indicates that a local maximum is found. This is obvious considering a very close goodness of fit and individual data points that have $dt = 38.8$ s (concept #5). So, at least this data point should have been found as maximum. A possible approach to avoid this problem would be to do the same optimisation with more (and different) starting estimates of the optimum.

Optimising the flight time $dt\,(M_{8\rightarrow 6})$, on the other hand, gives a predicted maximum of 61.3 s for the configuration listed in Table 5. Verifying this particular flight time showed a configuration performance similar to the one for L/D_{max}, albeit with a lower L/D (= 2.69). The verified $dt\,(M_{8\rightarrow 6})$ = 82 s is, whereas $dt\,(M_{10\rightarrow 8})$ = 59.4 s! The latter result stresses that special care should be taken that one is convinced of the global nature of the optimum value.

The last performance index to be discussed, $C_{n_{\beta,dyn}}$, has a maximum value of 1.403 with a verified value of 1.468 (the minimum is 0.189, still positive, so in principle the whole concept family should have relatively good flying properties). A follow-up controllability study must address the flying qualities of the resulting configuration (Table 5) in more detail, to indicate whether they are sufficient, because at this moment it is not clear what the target value should be. It may be possible that the maximum value is too much, resulting in a somewhat wild behaviour.

In Table 5, three different 'optimum' configurations were found, that are conflicting in some parameters of the vehicle geometry. In addition, especially the L/D and dt_{max} configuration result in very large heat fluxes. By simultaneously optimising multiple performance indices and taking constraints into account, theoretically a better feasible design may be derived that at least meets with more criteria and gives a consistent vehicle geometry. A first step in this direction is done with three performance indices, i.e., $q_{c_{max}}$ that should be lower than 3500 kW/m^2, the height for horizontal flight (as large as possible; goal has been defined as 35 km) and the flight time between Mach 10 and 8 (as large as possible; goal = 50 s). To solve the non-linear programming problem for this multi-objective optimisation, Goal Attainment (with equal weights for the objectives) is used, which is an improved form of Sequential Quadratic Programming.[8]

[8] The algorithm used is one of the local-search methods implemented in the Matlab$^{\delta}$ Optimization Toolbox.

The results, however, are striking. Since we wanted to avoid the risk of reaching a local maximum for the flight time, it was decided to take the 27 combinations of the original CCD as starting values for the Goal Attainment. All 27 cases converged to the same optimum of $q_{c_{max}} = 3500\,\mathrm{kW/m^2}$, $h = 33.5\,\mathrm{km}$ and $dt\,(M_{10\rightarrow 8}) = 9.7\,\mathrm{s}$—the relatively low $q_{c_{max}}$ prevents that $dt\,(M_{10\rightarrow 8})$ can reach higher values. However, the corresponding vehicle configurations were all different although some groups with identical values could be identified. This means that these performance indices are to a certain extent relatively insensitive to changes in vehicle geometry. Consequentially, we should focus on one (or several) additional performance indices that matter to include in the optimisation process. As an example the optimisation is repeated for similar goals for $q_{c_{max}}$ and h, and $dt\,(M_{8\rightarrow 6}) = 50\,\mathrm{s}$ instead of $dt\,(M_{10\rightarrow 8})$.

The resulting configuration is listed in the last column of Table 5. The performance of this configuration is quite reasonable, namely a height for horizontal flight of 33.4 km, whereas the time between Mach 6 and 8 is 32.3 s. The heat-flux constraint is just met, so $q_{c_{max}} = 3500\,\mathrm{kW/m^2}$. A verification design cycle shows that $q_{c_{max}} = 3189\,\mathrm{kW/m^2}$, $h = 33.0\,\mathrm{km}$ and $dt\,(M_{8\rightarrow 6}) = 36\,\mathrm{s}$, which is matching quite well. Note that $dt\,(M_{10\rightarrow 8}) = 9.5\,\mathrm{s}$, which indeed indicates that this flight time is relatively insensitive for certain vehicle variations, and that it is possible to focus on another performance index without major consequences.

As found before, $dt\,(M_{10\rightarrow 8})$ can be significantly increased, albeit at the expense of a much higher $q_{c_{max}}$, and consequently, a large impact on the structural design. If the technological problems of nose cooling can be solved, and in addition C-SiC (that can withstand high temperatures) is applied to part of the windward side of the vehicle, a unique experimental platform can be obtained.

3.4 Conclusions

An improved vehicle design is obtained for doing low-cost hypersonic re-entry testing and micro-gravity experiments, which will be launched by a Brazilian sounding rocket. This is established by applying a Response Surface Methodology incorporating orthogonal arrays centred around a number of performance indices that are derived from a set of well-defined user and mission requirements.

Essential geometric parameters of a generic vehicle, with a water-cooled nose, dihedral bottom and four control flaps, are optimised to such an extent, that:

1. hypersonic experiments can be performed around a flight Mach number of 8 for at least 20 s in a interesting flight regime regarding parameters like Reynolds number and chemical kinetics,
2. the vehicle will be aerodynamically stable during this flight phase, and
3. it will encounter thermal heat loads, which can be met with the foreseen structural design.

The maximum flap deflection to avoid strong shock-wave boundary-layer interaction has an obvious impact on control-system performance. The next step would therefore be to detail a flight control system and study its performance, e.g., verifying that the controllability is assured despite this flap constraint. Moreover, the simultaneous optimisation of performance indices including vehicle and mission constraints should be studied in more detail, such that the discussed design methodology can be a valuable tool in the (pre-)design phase of small re-entry vehicles.

4 Integrated Shape and Trajectory Analysis

For the second design case we will look at a different vehicle shape that allows for other aerodynamic experiments, but due to this shape it also encounters a different thermal environment. The vehicle shape may appear simple, i.e., it is a biconic capsule, but it has some favourable characteristics in terms of internal volume, stability characteristics, and ease of manufacturing. To show more of the capabilities of Design of Experiments, we will not only use this method to optimize the vehicle shape, but also do an integrated analysis of multiple re-entry missions.

4.1 The Re-entry Module Concept

In an earlier study two different bi-conical vehicles were analysed [16], the REVolution module studied at Delft University of Technology[9] and the Russian Volan module (see Fig. 2). The REVolution module allows for aerodynamic experiments with strong Shock-Wave Boundary-Layer (SWBL) interaction, but as a result there are high thermal fluxes on the flare. The heat fluxes on the Volan are lower, because of the larger nose radius and weak SWBL interaction.

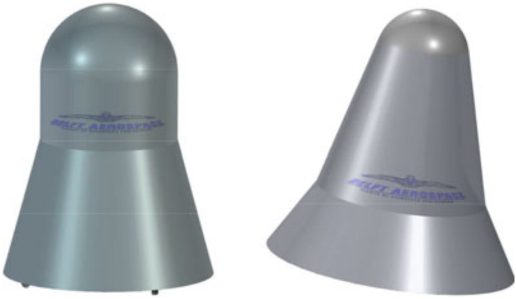


Fig. 2 Volan (*left*) and DART (*right*)

[9] This module is also known as DART, which stands for Delft Aerospace Re-entry Test Vehicle [3].

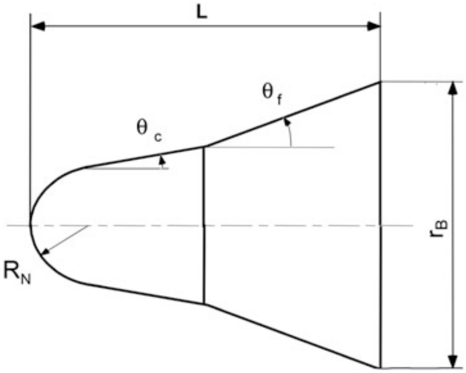


Fig. 3 General shape of the re-entry module

The best design would probably be something in between these two concepts, but certainly closer to the Volan than to the REVolution concept, because strong SWBL interaction would cause heat fluxes too high for the wall cooling system of the flare. As a second demonstration of the proposed methodology we will investigate a wide range of possible shapes to find the most promising shape of the module. Common to all vehicles is an outer skin consisting of PM1000 that sets a constraint to the design with respect to a maximum allowable skin temperature of 1300 °C. This is a major design driver for the shape of the re-entry module and as a result of this constraint additional cooling devices have to be included for the nose and the flare. The limitations of the cooling devices lead to additional constraints and requirements that make such a design very attractive to demonstrate the versatility of the RSM.

The outer geometry of the (biconic) re-entry capsule is fully determined by five parameters (Fig. 3): the nose radius, R_N, the base radius, r_B, the semi cone-angle, θ_c, the semi flare-angle, θ_f, and the vehicle length, L. The base diameter has a fixed dimension of 1.12 m and the maximum vehicle length shall not be more than 1.6 m, because of the available space inside the fairing of the selected launcher (the submarine-launched VOLNA, a former Russian ICBM).

The re-entry module has some novel features, such as a fully metallic outer skin, a water-cooled nose and a new enhanced radiation-cooling system for the body to keep the body temperature below 1300° C. To avoid exceeding this safe temperature limit the aerodynamic design has to assure that the heat flux on the body will not be more than 600 kW/m^2, the maximum capacity of the wall-cooling system. The heat flux caused by the strong SWBL interaction will certainly exceed 600 kW/m^2, so it is also required that no separation after the cone-flare junction occurs and that weak interaction is assured.

4.2 Theoretical Model

Nose cooling by nucleate pool boiling needs only 8–15 kg water (depending on the nose radius) due to the high evaporation heat of water, but there exists a maximum value for the heat flux to avoid film boiling [15], given by the Rohsenow limit:

$$q_{\max} = \frac{\pi}{24}\rho_v h_{f_g}\left(\frac{\sigma^* a_g(\rho_l - \rho_v)}{\rho_v^2}\right)^{\frac{1}{4}}\left(1 + \frac{\rho_v}{\rho_l}\right)^{\frac{1}{2}} \tag{7}$$

If this limit heat flux is exceeded a vapour film will develop between water and nose wall that decreases the heat transfer and as a result the temperature of the nose will jump up to unacceptable high values. Note: in Eq. (7) the following symbols have been used: ρ_v and ρ_l are the water density in the vapour and liquid phase, h_{f_g} is the latent heat of vaporisation, a_g is the gravitational acceleration, and σ^* is the surface tension of liquid-vapour interface.

The shape of the module shall be designed such that the stagnation heat flux will not exceed the Rohsenow limit of Eq. (7). The heat flux of the cone and flare shall not exceed 295 kW/m^2 for a non-cooled skin or 600 kW/m^2 for a cooled skin. These maximum allowable heat fluxes practically exclude strong SWBL interactions, because of the extreme heat flux at reattachment of the boundary layer. As a result a constraint is defined for the maximum allowable cone-flare angle to avoid the strong interaction. Thus, the maximum cone-flare angle without separation of the boundary layer at the cone-flare junction is [23]:

$$\theta_f - \theta_c = 80\sqrt{M_e}\left(\frac{\mu_e^* T_e}{\mu_e T_e^*}\,\frac{1}{3\mathrm{Re}_{x,e}}\right)^{\frac{1}{4}} \tag{8}$$

where T_e and μ_e are the temperature and the viscosity at the edge of the boundary layer of the flare, respectively, and the asterix denotes evaluation at the reference temperature

$$T_e = T_\infty\left(\frac{2\gamma(\gamma-1)}{(\gamma+1)^2}\right)M_\infty^2 \sin^2\beta \tag{9}$$

$$\mu_e = \frac{B_\mu T_e^{\frac{3}{2}}}{T_e + S_\mu} \tag{10}$$

$$T_e^* = 0.28\ T_e + 0.5\ T_w + 0.22\ T_{aw} \tag{11}$$

In Eq. (10), Sutherland's equation, $B_\mu = 1.458\times10^{-6}$ Pa s K$^{-1/2}$ is a constant depending on the gas and S_μ = 110.4 K is Sutherland's constant.

The streamlines at the edge of the boundary layer passed an oblique shock wave. By shock-wave theory the shock angle β is

$$\beta = \frac{\theta_c F(M_\infty)}{M_\infty^2} + \sqrt{\left(\frac{\theta_c F(M_\infty)}{M_\infty^2}\right)^2 + \frac{1}{M_\infty^2}} \tag{12}$$

with

$$F(M_\infty) = (\gamma + 1)M_\infty^2 + 2 \tag{13}$$

Note that Eq. (12) is valid for a two-dimensional wedge flow. However, the shock angle of a cone is smaller than for a wedge but the shock of a blunted cone is curved and as a result higher shock angles are expected. The Reynolds number in Eq. (8) is influenced considerably by the entropy layer caused by the blunt nose that reduces the velocities at the edge of the boundary layer. The Reynolds numbers are corrected for this entropy effect by a method based on measurements [7].

The heat fluxes on the nose, the cone and the flare for the complete re-entry flight are determined by engineering approximations based on existing literature. The heat flux in the stagnation point on the nose can be estimated by the classical Chapman relation for a wall temperature of 300 K:

$$q_{c300} = \frac{C}{\sqrt{R_N}} \sqrt{\frac{\rho_\infty}{\rho_0}} \left(\frac{V_\infty}{V_c}\right)^3 \tag{14}$$

with $C = 1.06584 \times 10^8 \sqrt{\mathrm{m}}$ and $V_c = 7905$ m/s. The stagnation heat flux corrected for the actual outer wall temperature is

$$q_c(t) = \frac{q_{c300}(t)}{T_{aw}(t) - 300}[T_{aw}(t) - T_w(t)] \tag{15}$$

The heat flux on the cone is determined by

$$q_{c,cone} = St \rho_e V_\infty \cos\theta_c c_p [T_{aw}(t) - T_w(t)]\sqrt{3} \tag{16}$$

where the Stanton number St and the adiabatic wall temperature T_{aw} are calculated from

$$St = \frac{0.332}{\sqrt{\mathrm{Re}_x^*}}(Pr^*)^{\frac{2}{3}} \tag{17}$$

$$T_{aw}(t) = T_\infty + \frac{rV_\infty^2}{2c_p} \tag{18}$$

The asterix of the Reynolds and Prandtl numbers refers to the evaluation of these numbers at the reference temperature, defined by Eq. (11).

The heat flux on the flare can be estimated by

$$q_{c,flare}(z) = c_p \rho_f v_f \left(T_{aw} - T_f\right) A \left(\frac{T_f}{T_f^*}\right)^{1-2n} \left(\frac{T_f \mu^*}{T_f^* \mu_f}\right)^n \left(\frac{\mu_f}{\rho_f v_f z}\right)^n \tag{19}$$

where z is the length of the boundary layer over the flare along a meridian [22]. The boundary layer of the flare past the shock wave of the cone-flare junction is assumed to be turbulent, so in Eq. (19) the values $n = 0.2$ and $A = 0.575$ can be used. The temperature on the flare can be calculated by

$$T_f = T_\infty \left(1 + \frac{\gamma - 1}{2} M_\infty \sin^2 \theta_f\right) \left\{1 + \left(\frac{T_e}{T_\infty} - 1\right) \left(1 - e^{c\left(\frac{z}{\delta_{bl}}+1\right)}\right)\right\} \tag{20}$$

with c = -0.412. Moreover, T_f^* is evaluated by Eq. (11), whereas the boundary-layer thickness of the cone near the cone-flare junction is given by

$$\delta_{bl} = \frac{5L_c}{\cos\theta_c \sqrt{\mathrm{Re}_{L_{lam}}}} \tag{21}$$

A correction of the specific heat for high temperature effects is applied with [7]

$$c_p = c_{p\infty} \left(\frac{T}{T_\infty}\right)^{0.1} \tag{22}$$

4.3 *Aerodynamic Design*

4.3.1 Design Setup

As discussed before, the outer geometry of the bi-conic re-entry vehicle that serves as a generic base line for the aerodynamic design is defined by five independent body parameters, i.e., R_N, θ_c, θ_f, r_B and L. In the exploratory phase of the research, it was found that varying all five parameters gave results that were highly non-linear, and could not be fitted properly by a second-order response surface. Adding cubic terms to the response surfaces improved the fit through the data points considerably (indicated by a larger coefficient of determination), but the predictive quality *in between* the data points was still bad (indicated by the relatively large uncertainty in the computed coefficients of the response surface). Two approaches were left to follow, i.e., decreasing the ranges of the design variables such that the system would become more linear, or to remove a design variable that would possibly account for (part of) the non-linearity. We have selected the latter approach to cover an as large design space as possible.

Two possible candidates for the non-linearities in the system appeared to be the vehicle length and base radius. This second parameter has been selected to be frozen, since basically one wants a base diameter that is as large as possible to maximise the internal volume. So, this leaves four parameters for variation according to a CCD as discussed before. For the factorial portion of the design, Taguchi's L_{16} array will be used, as was also done in Sect. 3.2. Also now the axial parameter α will be chosen such that the design is orthogonal, which means that for $n_0 = 1$ centre point and 16 designs (L_{16}) in the factorial portion $\alpha = 1.4142$. The total number of designs comes to $n_{tot} = 16+4*2+1 = 25$. The nominal parameter settings and their variations are chosen to be representative to obtain a broad family of (viable) designs, see also Table 6. The maximum parameter variation is, of course, defined by the axial points. As for the previous design case, the range in the L_{16} array is given by ±1; this means that the maximum physical range should be divided by 1.4142 to get the corresponding values. The nominal configuration has been depicted in Fig. 4, whereas in Table 7 the settings of the independent variables for all 25 configurations have been listed.

Table 6 Parameter variations in CCD ($\alpha = 1.4142$)

Factor	Nominal value	Maximum range L_{16}	Maximum range axial
R_N (m)	0.3	0.071	0.10
θ_c (°)	5	3.53	5
θ_f (°)	20	3.53	5
L (m)	1.45	0.106	0.15

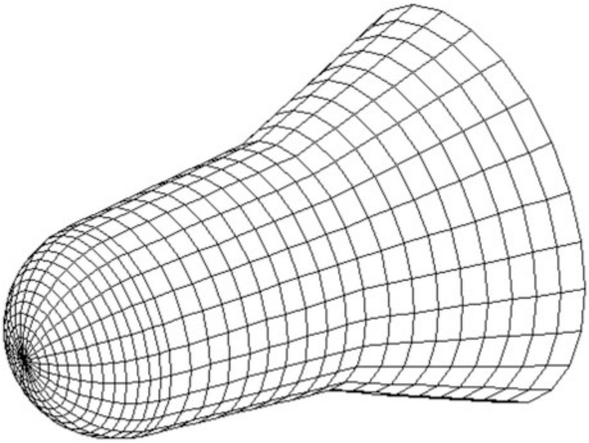


Fig. 4 Nominal configuration of the bi-conic re-entry capsule

Table 7 Central composite design of bi-conic re-entry capsule

Design	R_N (m)	θ_c (°)	θ_f (°)	L (m)
L16 row #01	0.229	1.464	16.464	1.344
L16 row #02	0.229	1.464	16.464	1.556
L16 row #03	0.229	1.464	23.536	1.344
L16 row #04	0.229	1.464	23.536	1.556
L16 row #05	0.229	8.536	16.464	1.344
L16 row #06	0.229	8.536	16.464	1.556
L16 row #07	0.229	8.536	23.536	1.344
L16 row #08	0.229	8.536	23.536	1.556
L16 row #09	0.371	1.464	16.464	1.344
L16 row #10	0.371	1.464	16.464	1.556
L16 row #11	0.371	1.464	23.536	1.344
L16 row #12	0.371	1.464	23.536	1.556
L16 row #13	0.371	8.536	16.464	1.344
L16 row #14	0.371	8.536	16.464	1.556
L16 row #15	0.371	8.536	23.536	1.344
L16 row #16	0.371	8.536	23.536	1.556
-Axial #01	0.200	5.000	20.000	1.450
Axial #01	0.400	5.000	20.000	1.450
-Axial #02	0.300	0.000	20.000	1.450
Axial #02	0.300	10.000	20.000	1.450
-Axial #03	0.300	5.000	15.000	1.450
Axial #03	0.300	5.000	25.000	1.450
-Axial #04	0.300	5.000	20.000	1.300
Axial #04	0.300	5.000	20.000	1.600
Centre point	0.300	5.000	20.000	1.450

the *grey*-shaded cells represent the extreme values of the design space

4.3.2 Performance Indices

The performance indices that have been defined can be divided into several categories. In this study, we will focus on the structure-related performance indices, i.e., the maximum heat flux in the nose (including the integrated heat load), on the cone and on the flare, as well as two aerodynamics-related performance indices, i.e., the location of the centre of pressure and the maximum cone-flare angle for which only weak SWBL interaction is assured. The maximum fluxes and heat load should all be as small as possible, of course, whereas the cone-flare angle should be as large as possible. The centre of pressure should be located as far backward as possible, to enhance the stability properties of the bi-cone.

4.3.3 Mission Analysis

It is noted that the variation of entry velocity and entry angle, and vehicle mass will be taken into account. At this stage, no detailed layout studies have been done, which would give input to defining the mass properties of the vehicle. Therefore, a mass range in between 150 and 250 kg has been assumed that is representative for this type of vehicle. The mission profile for the bi-conic vehicle consists of a ballistic re-entry at zero angle of attack. The state vector of the vehicle at an altitude of 120 km is given by the following ranges, depending on the chosen launcher: velocity range [4500,6000] m/s, with a nominal value V_e = 5250 m/s, and a flight-path angle range [$-10°$,–3°], with a nominal value $\gamma_e = -6.5°$.[10]

It is assumed that these parameters will vary over a minimum, nominal and maximum level, equally spaced around the nominal value. However, not all possible combinations (3^3 = 27) will be simulated, because for the total of 25 concepts it would generate too much data to process in an efficient way. As a representative alternative Taguchi's L_9 orthogonal array will be used to determine the parameter combinations. This array has nine rows (i.e., nine simulations) and allows for the variation of up to four independent parameters. Although it is likely that the three selected parameters are not independent, interactions will not be taken into account, because we are only interested in the main effect of the parameter variation. In Table 8, the parameter settings are listed. Note that by assigning the parameters to columns #2 to #4, row 2 represents the nominal value of each of the three parameters.

Table 8 Orthogonal array L_9 with four factors on three levels

Run	Not used	V_e (m/s)	γ_e (°)	m (kg)	Load case
1	-1	4500	-10	150	–
2	-1	5250	-6.5	200	Nominal
3	-1	6000	-3	250	–
4	0	4500	-6.5	250	–
5	0	5250	-3	150	–
6	0	6000	-10	200	Maximum
7	1	4500	-3	200	Minimum
8	1	5250	-10	250	–
9	1	6000	-6.5	150	–

[10] The remaining initial conditions (longitude τ, latitude δ, and heading χ) at the atmospheric interface are (arbitrarily) defined to be: $\tau = \delta = 0°$, and $\chi = 90°$ to define an equatorial re-entry.

4.4 Results

The CCD comprises a total of 25 concepts that have to be analysed (Table 7). Per concept, nine simulations are executed to account for the range of entry conditions and vehicle mass. This means that a total of 225 trajectory simulations have been executed. It can easily be understood that this amount of data is too much to be discussed in detail. Therefore, we only highlight some of the results.

After inspecting the results and the parameter combinations given by Table 8, a minimum, nominal and maximum load case can be identified: row 7 is the minimum load case (resulting in the minimum maximum heat flux), row 2 is the nominal load case, and row 6 is the maximum load case. These cases will provide part of the data for analysis.

To get an impression for the kind of mission we are discussing here, in Fig. 5a the nine trajectories have been plotted for the nominal configuration. Indicated is the Mach number as a function of altitude. Clearly, the influence of three different initial velocities is shown. Basically, there is no influence of the varying mass and entry flight-path angle down to an altitude of about 60 km. Then, three trajectories per entry velocity can be discerned, showing the impact of the two other parameters. Note that the simulation stops at $M = 3$, albeit at different altitudes, because of the trajectory variation.

In Fig. 5b, one of the performance indices, i.e., the flare flux, has been plotted as a function of altitude for the nominal configuration. The thermal load on the flare gradually increases when the vehicle dives deeper into the atmosphere. The maximum flare flux occurs at different altitudes for each of the nine trajectories (in between 40 and 33 km), obviously also with different peak values (ranging from 650 to 1850 kW/m^2). With a maximum allowable flare flux of about 600 kW/m^2, it can be concluded that the nominal configuration does not meet with this constraint for each of the nine trajectories, not even the minimum-load trajectory.

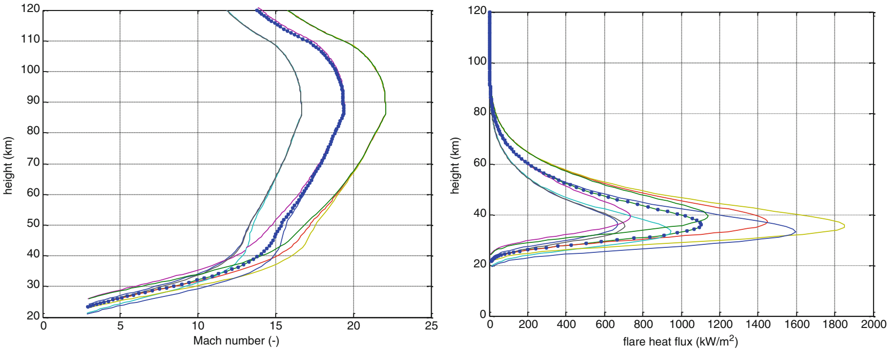


Fig. 5 Results for the nominal configuration and nine different design trajectories. The nominal-load trajectory has been indicated with a *dotted line*. *Left*: Height as a function of Mach number. *Right*: Maximum flare flux versus height

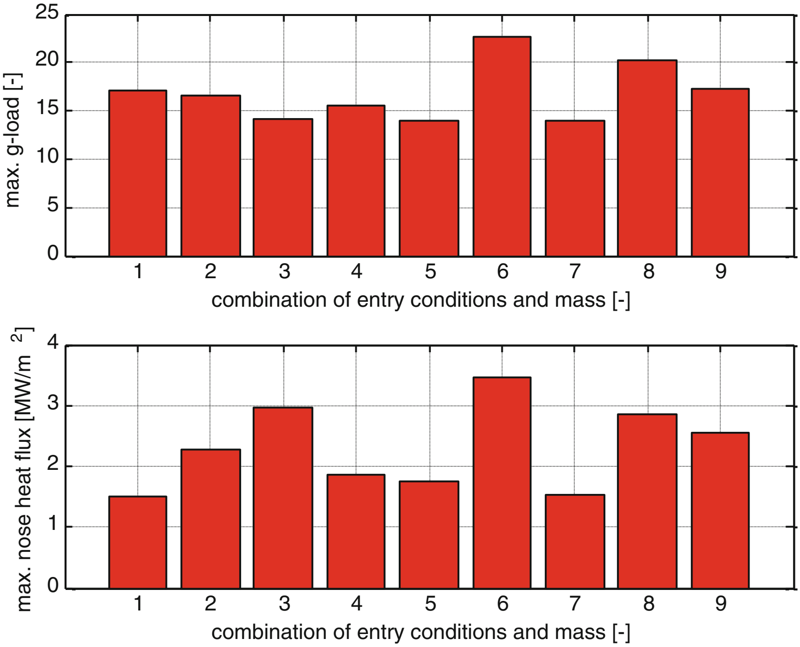


Fig. 6 Maximum *g-load* (*top*) and maximum nose heat flux for the nominal configuration and nine different combinations of entry conditions and mass

Figure 5b shows the importance of the flare-flux profile for the design process. However, as such this figure does not give the right representation to be included in an automated design-optimisation process. In Fig. 6, two of the selected performance indices have been plotted for the nominal configuration, i.e., the maximum *g-load* and the maximum nose heat-flux. These bar charts show the variation of the maximum values for each of the trajectories. Combining the two charts, also confirms the definition of the minimum- and maximum-load trajectories. It may be concluded, that irrespective of the thermal load on the vehicle, also the mechanical load is too much for some of the trajectories. Obviously, once the shape has been optimised to minimise the thermal loads, attention should be paid to the mechanical loads as well. Since the mechanical load is less shape dependent than the thermal load, the former will not be considered here any further.

Although not included in the current study, some results will be presented here for future reference. It may be possible to derive some optimisation criteria based on the dissociation parameter $\rho_\infty R_N$. As an example, in Fig. 7 the dissociation parameter has been plotted versus the velocity. Also included in the figure is a box, which assures the occurrence of dissociation: the chemical-kinetics parameter $\rho_\infty R_N$ and the relative velocity V_∞ should be simultaneously covering $0.1\ \mathrm{g/m^2} \leq \rho_\infty R_N \leq 10\ \mathrm{g/m^2}$ and $V_\infty \geq 2.5\ \mathrm{km/s}$ (Sect. 2.2).

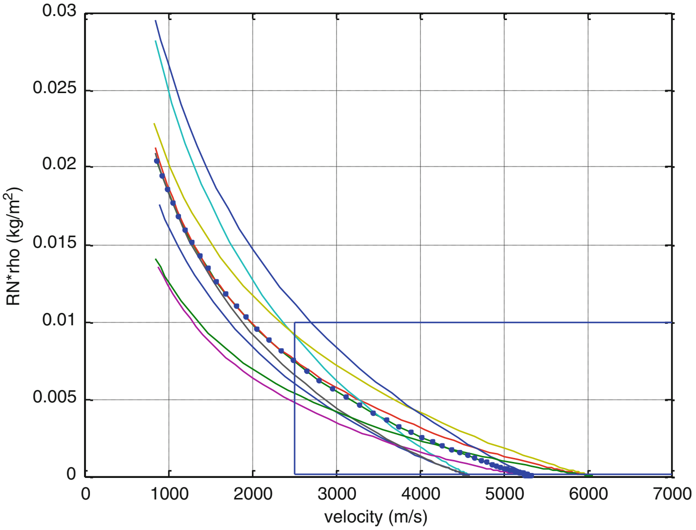


Fig. 7 Dissociation parameter versus velocity (nominal concept, nine load cases). *Dotted line*: nominal load case; *rectangular box*: flight region of interest

Optimising the vehicle shape is in principle done for a single trajectory. One might think, that if the vehicle is optimised for all nine trajectories, the result may be nine different vehicles. A preliminary analysis has shown, however, that for a biconic shape optimisation varying only R_N, θ_c and $\theta_{f/c} = \theta_f - \theta_c$, the optimised shape was the same for the minimum-, nominal- and maximum-load trajectory. Therefore, initially we will focus on the nominal-load trajectory and do a verification for the other two.

In Fig. 8, two of the performance indices have been plotted for each of the 25 configurations, i.e., the x-location of the c.o.p., x_{cop}, relative to the vehicle length and the maximum occurring flare flux. From these results, it shows that concept #19 is best with respect to c.o.p. and will be the most stable. However, even though this concept is not the worst with respect to the flare flux, it is by no means the best one. This would be concept #21, with a 60 % lower flare flux than concept #19. The conflict in optimising both performance indices simultaneously has thus been demonstrated. Note that both concepts outperform the nominal concept.

As a further illustration, in Fig. 9 the maximum flare flux and the maximum cone-flare angle have been plotted for the minimum- and maximum-load trajectories. It is concluded that the variation over the concepts seems to be more or less the same for the two trajectories, which confirms our earlier findings. Looking in more detail at the results, it shows that only one concept (#21) can withstand the thermal load on the flare in the maximum-load trajectory. This makes it likely that the mission that is to be flown by the vehicle should be more tailored towards the nominal-load trajectory, otherwise different materials and/or thermal protection should be considered.

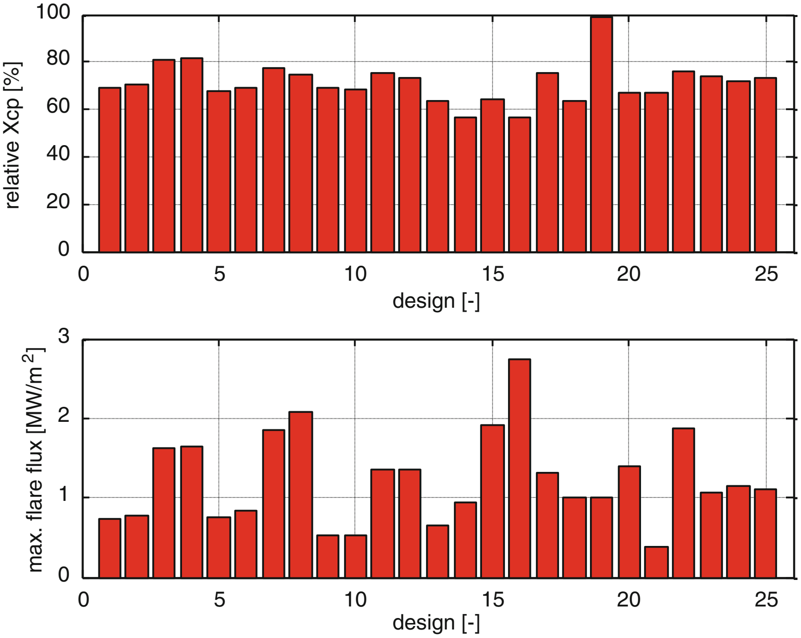


Fig. 8 Relative X location of the centre of pressure (*top*) and maximum flare heat flux (*bottom*) for each of the 25 designs (nominal-load trajectory)

For each of the load cases, the individual performance indices are fit into response surfaces (i.e., a least-squares fit). As an example response surface, the maximum flare flux (nominal-load trajectory) is (nominally) represented by the coefficients listed in Table 9.

Note that the response surface has been computed for normalised independent variables, i.e., each of them varies between the minimum and maximum axial value ($\alpha = \pm 1.4142$), which means that the relative contribution to, for instance, the mean value is 1.4142 times larger than the indicated coefficient values.

It can easily be seen that the contribution of each of the terms cannot be neglected. Of the linear terms, θ_c and θ_f contribute up to 100 % to the mean value. This means, that by varying these two parameters the maximum flare flux can be decreased significantly, which is important design information. The relatively large interaction and quadratic terms stress the non-linear nature of the flare flux in response to a vehicle design variation.

The optimum configuration of the re-entry vehicle can be found by either a single-objective or multi-objective optimisation of the related response surfaces. Previously, we have already identified that optimising, for instance, the flare flux and the x_{cop} will lead to conflicting configurations. Therefore, for this case study only multi-objective optimisation using goal attainment with equal weight for the objectives will be applied. The five criteria that have been selected to be optimised are the maximum $q_{c,nose}$, $q_{c,cone}$ and $q_{c,flare}$ (these responses should be minimised) and $\theta_{f/c}$ and x_{cop} (which should be maximised).

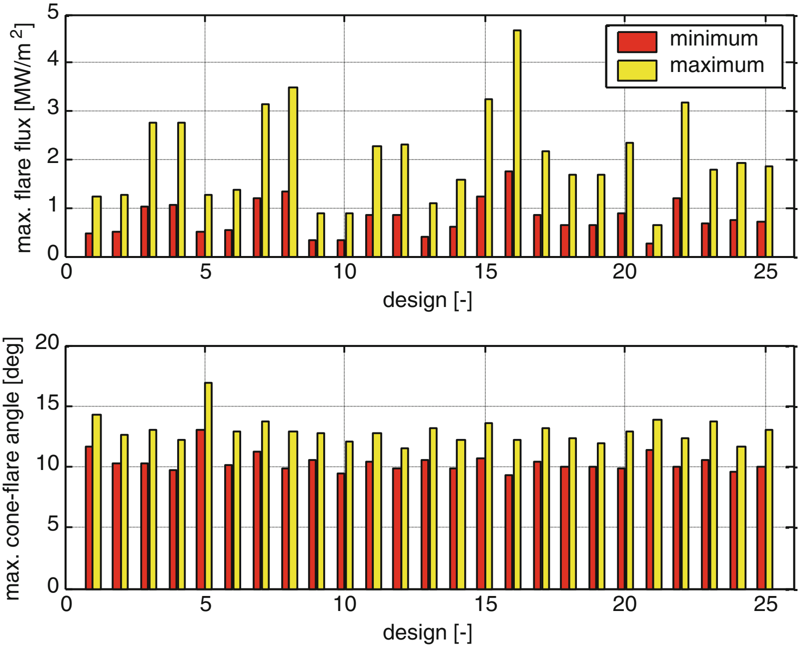


Fig. 9 Maximum flare flux (*top*) and maximum cone-flare angle (*bottom*) for each of the 25 designs (minimum- and maximum-load trajectory)

For the nominal-load trajectory, the following optimised configuration is found: $R_N = 0.353\,\text{m}$, $\theta_c = 2.69°$, $\theta_f = 15.00°$, and $L = 1.404\,\text{m}$, with predicted responses of $q_{c,nose} = 1.83\,\text{MW/m}^2$, $q_{c,cone} = 57.4\,\text{kW/m}^2$, $q_{c,flare} = 312.8\,\text{kW/m}^2$, $\theta_{f/c} = 12.2°$, and $x_{cp} = 67.2\,\%$. The final step in this optimisation process is to generate a panel grid for this configuration, compute the aero-dynamic coefficients and to compute the responses with a non-linear trajectory analysis. In doing so, the following results were obtained: $q_{c,nose} = 1.94\,\text{MW/m}^2$, $q_{c,cone} = 65.8\,\text{kW/m}^2$, $q_{c,flare} = 408.8\,\text{kW/m}^2$, $\theta_{f/c} = 11.9°$, and $x_{cp} = 0.94352\,\text{m}= 67.2\,\%$. As can be seen, the results match quite well, apart from the flare flux. The differences are attributed to the finite accuracy of the response-surface approximation.

Finally, we will check the optimal configuration for the minimum and maximum-load trajectories. Optimising the response surfaces yields $R_N = 0.278\,\text{m}$, $\theta_c = 3.15°$, $\theta_f = 15.00°$, and $L = 1.366\,\text{m}$ for the minimum-load trajectory, and $R_N = 0.4\,\text{m}$, $\theta_c = 1.60°$, $\theta_f = 15.00°$, and $L = 1.434\,\text{m}$ for the maximum-load trajectory.

The three configurations are shown in Fig. 10, and obviously they are not the same. Apart from the flare angle, which is constant for the three vehicles, we find an increasing nose radius from minimum-, to nominal- to maximum-load trajectory, a decreasing cone angle and an increasing vehicle length. Comparison of predicted and verified responses gives similar difference between the two as for the nominal-load trajectory. However, if we compare the results of all nine trajectories for each of

Table 9 Nominal response-surface coefficients for the maximum flare flux, with relative contribution to the mean value (coefficient b_0)

	Absolute value (kW/m^2)	Relative value (%)
Linear terms ($b_0, \cdots, b_k$)		
Constant term	1051.4	–
R_N (m)	−37.3	5.0
θ_c (°)	189.2	25.4
θ_f (°)	−545.8	73.4
L (m)	78.6	10.6
Interaction terms ($b_{12}, \cdots, b_{(k-1)k}$)		
$R_N \times \theta_c$	108.9	14.6
$R_N \times \theta_f$	37.4	5.0
$R_N \times L$	50.7	6.8
$\theta_c \times \theta_f$	123.5	16.6
$\theta_c \times L$	83.6	11.2
$\theta_f \times L$	42.7	5.7
Quadratic terms ($b_{11}, \cdots, b_{kk}$)		
R_N (m)	56.8	7.6
θ_c (°)	80.4	10.8
θ_f (°)	45.7	6.2
L (m)	33.0	4.4

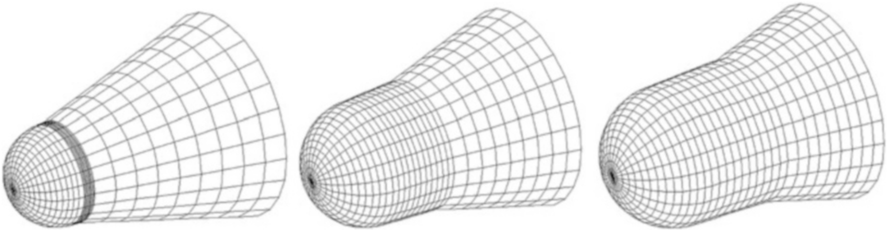


Fig. 10 Optimal configurations for three trajectory load cases: minimum (*left*), nominal (*middle*), and maximum (*right*)

the three configurations, it appears that in case of the flare flux, the response value of the ‘maximum’ configuration is consistently smaller than the corresponding values of the other two concepts, even for the trajectory for which those concepts were optimised (see Fig. 11a). This means, that regarding the flare flux, the ‘maximum’ concept is the better concept.

This is not the case for the maximum cone-flare angle (see Fig. 11b). Looking at the results, it appears that the opposite is the case, the ‘minimum’ concept is the better one. Another observation that can be made is that the variation of the maximum cone-flare angle in each of the nine trajectories is not the same as for the flare flux. For instance, compare the results of trajectory #9, where the flare flux is by no means the largest whereas the cone-flare angle is. In fact, the minimum-load trajectory #7 (almost) gives the smallest flare flux, but in case of the cone-flare angle

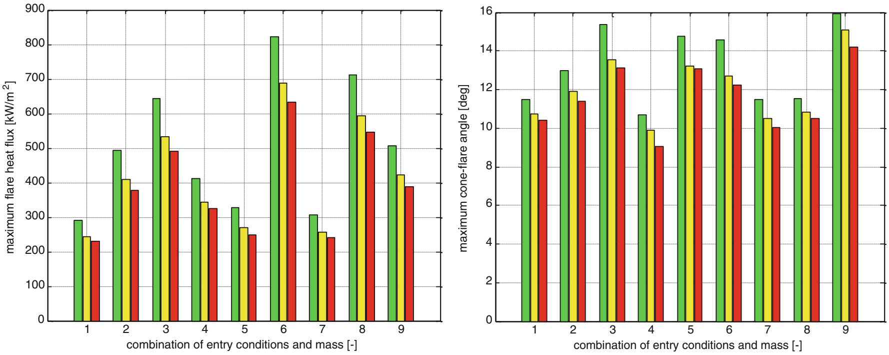


Fig. 11 Results for each of the nine combinations of entry conditions and mass, plotted for the minimum-, nominal- and maximum-load configuration (*left-to-right*). *Left*: Maximum flare heat flux. *Right*: Maximum cone-flare angle

this would be trajectory #4. So, with some reservation, it may be concluded that the addition of the cone-flare angle as one of the optimisation criteria, the trajectory-independent optimisation has been changed into a dependent one.

This trajectory dependency can be further analysed by looking at the signal-to-noise ratio for each of the concepts. Let $\bar{y}$ be the mean response of nine trajectory runs for a particular configuration, and let σ be the corresponding standard deviation. The signal-to-noise ratio of any response can be defined by

$$S/N = 10 \log_{10} \left(\frac{\bar{y}^2}{\sigma^2} \right) \tag{23}$$

A larger value for S/N indicates a smaller sensitivity to trajectory and mass variations. Plotting $\bar{y}$, σ and S/N for the maximum flare flux (Fig. 12a) we find a large variation in $\bar{y}$ and σ for the 25 configurations, meaning a large variation in absolute values due to the trajectory and mass variations, but an almost constant η. This means, that each configuration will respond in the same relative manner to the trajectory and mass variations, with, of course, differences in the absolute values of the flare flux. In case of the maximum cone-flare angle, on the other hand, a much smaller variation in $\bar{y}$ and σ is seen (Fig. 12b), but a larger variation in S/N. This confirms the earlier findings that due to trajectory and mass variations the maximum-cone flare angle has such an impact on the optimisation process that different optimal configurations have been found.

Focusing on the optimal configuration for the nominal-load trajectory, and comparing this configuration with the Volan (see Fig. 2), the resemblance is striking. The Volan has also been optimised to avoid the strong SWBL interaction, as well as to minimise the thermal loads, a configuration that is confirmed in the current study with a relatively small design effort.

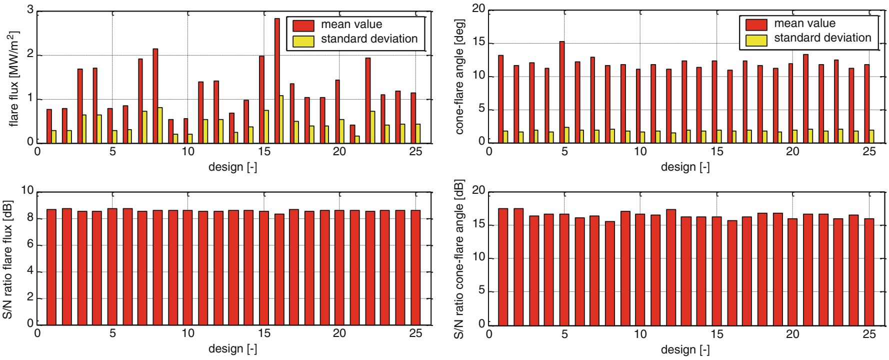


Fig. 12 Mean value and standard deviation for each of the 25 designs (*top*), as well as the signal-to-noise ratio to reflect trajectory sensitivity (*bottom*). *Left*: Maximum flare heat flux. *Right*: Maximum cone-flare angle

4.5 Conclusions

The second case study was focussed on improving the design of a biconic reentry vehicle, and study the effects of different initial conditions and vehicle mass on the shape. Three optimal configurations were obtained for different combinations of trajectory and vehicle mass. Comparison of the nominal and maximum-load configurations with the Russian Volan design shows a striking resemblance. It was demonstrated, however, that not all investigated combinations of entry velocity, flight-path angle and vehicle mass are feasible with respect to the thermo-mechanical loads, most notably the flare flux exceeds the allowable value for the maximum-load configuration and for combination #6 even for all three optimal configurations.

5 Concluding Remarks

The current study has led to a methodology to investigate a wide range of possible shapes and to find the most promising one for a re-entry test vehicle. Whereas normally in a conceptual-design phase one would only look at a limited number of variations, the use of techniques from Design of Experiments enables the designer to study a broad range of shapes within one class of vehicles. The use of engineering tools gives results rapidly, yet still provides the required accuracy to lead to design conclusions. So the "optimum" vehicle would be the best compromise with a minimum effort and without using complex time-consuming design tools, and thus saving time and costs in the preliminary design phase. It can rapidly provide insight in the main design problems and thus rule out the many useless configurations.

The applied design method is a combination of Central Composite Design (CCD) and a Response Surface Methodology, enabling to fit a second-order response surface through the defined performance indices. Even with the good results achieved, there are limitations as well. The number of (independent) design parameters should not be too large, as to avoid interactions between them. Also strong non-linearities in the responses may lead to less than optimal results. In the second case of the bi-conic vehicle some of the response surfaces appeared to be troublesome with respect to accuracy. Higher-order approximations may be required, but since CCD is developed to derive second-order approximations, it is not suitable for higher orders. And unless one has insight in the nature of the higher-order terms, which would allow for building a higher-order response surface, the preferred method may be a full factorial design. This would obviously lead to a rapid increase of design options.

Another approach to avoid such problems could be an evolutionary process with “genetic manipulation”. The configurations would be generated directly in large numbers and processed, which would have the advantage that the set of optimal alternatives in multi-objective optimisation (the so-called Pareto front) is automatically formed. An essential assumption in the use of evolutionary algorithms is, of course, that all design tools are linked together, and that data transfer between the individual tools is fully automated.

Acknowledgements The past and present cooperation and many fruitful discussions with Kees Sudmeijer and Frans Kremer is gratefully acknowledged. Without them the outcome would probably have been different, and for sure far less enjoyable. Thank you!

References

1. Bertin, J.J.: Hypersonic Aerothermodynamics. AIAA Education Series, American Institute of Aeronautics and Astronautics, Washington, D.C. (1994)
2. Box, G.E.P., Wilson, K.B.: On the experimental attainment of optimum conditions. J. R. Stat. Soc. B **13**, 1–38 (1951)
3. Buursink, J., van Baten, T.J., Mooij, E., Sudmeijer, K.J.: DART: the Delft aerospace re-entry test vehicle. IAF-00-V.4.07. In: 51st International Astronautical Congress, Rio de Janeiro, 2–6 October 2000
4. Dirkx, D., Mooij, E.: Optimization of entry-vehicle shapes during conceptual design. Acta Astronaut. **94**, 198–214 (2014) (online 2013). doi:10.1016/j.actaastro.2013.08.006
5. Fowlkes, W.Y., Creveling, C.M.: Engineering Methods for Robust Product Design: Using Taguchi Methods in Technology and Product Development. Addison-Wesley, Reading, MA (1995)
6. Khuri, A.I., Cornell, J.A.: Response Surfaces: Designs and Analyses. Statistics: Textbooks and Monographs, vol. 81. Dekker, New York (1987)
7. Krasnov, N.F.: Aerodynamics of Bodies of Revolution. American Elsevier, New York (1970)
8. Montgomery, D.C.: Design and Analysis of Experiments, 2nd edn. Wiley, New York (1984)
9. Mooij, E.: Heat-flux tracking for thermal-protection system testing. AIAA-2014-4141. In: AIAA/AAS Astrodynamics Specialist Conference, San Diego, 4–7 August 2014
10. Mooij, E., Marée, A.G.M., Sudmeijer, K.J.: Aerodynamic controllability of a selected re-entry test vehicle. IAF-95-V.4.04. In: 46th International Astronautical Congress, Oslo, October 2–6 1995

11. Mooij, E., Kremer, F.J.G., Sudmeijer, K.J.: Aerodynamic design of a low-cost re-entry test vehicle using a Taguchi approach. AIAA-1999-4831. In: AIAA 9th International Space Planes and Hypersonic Systems and Technologies Conference, Norfolk, VA, November 1–4 1999
12. Mooij, E., Kremer, F.J.G., Sudmeijer, K.J.: Mission analysis of a low-cost re-entry test vehicle. AIAA-99-4935. AIAA 9th International Space Planes and Hypersonic Systems and Technologies Conference, Norfolk, VA, November 1–4, 1999
13. Myers, R.H., Montgomery, D.C.: Response Surface Methodology: Process and Product Optimization Using Designed Experiments. Wiley Series in Probability and Statistics. Wiley, New York (1995)
14. Nash, S.G.: Newton-Type Minimization via the Lanczos method. SIAM J. Numer. Anal. **21**, 770–788 (1984)
15. Osizik, M.N.: Heat Transfer. McGraw-Hill, New York, 1985
16. Ottens, H.B.A.: Preliminary computational investigation on aerodynamic phenomena on Delft aerospace re-entry test vehicle (DART). In: 4th European Symposium on Aerothermodynamics for Space Vehicles, Capua (2001). Published in: ESA SP-487, 2002, pp. 207–213
17. Phadke, M.S.: Quality Engineering Using Robust Design. Prentice-Hall, Englewood Cliffs, NJ (1989)
18. Ridolfi, G., Mooij, E., Corpino, S.: Complex-systems design methodology for systems-engineering collaborative environment. In: Systems Engineering. Theory and Applications. InTech, Rijeka (2012). ISBN 979-953-307-410-7
19. Ridolfi, G., Mooij, E., Dirkx, D., Corpino, S.: Robust multi-disciplinary optimization of unmanned entry capsules. AIAA-2012-5006. In: AIAA Modeling and Simulation Technologies Conference, Minneapolis (2012)
20. Rufolo, G., Pereira, C., Camarri, F.: ESA intermediate experimental vehicle in-flight experimentation. objectives, experiment, implementation, qualification and integration. IAC-14.D2.6.3. In: 65th International Astronautical Congress, Toronto (2014)
21. Sagliano, M., Samaan, M., Theil, S., Mooij, E.: SHEFEX-3 optimal feedback entry guidance. AIAA-2014-4208. In: AIAA SPACE 2014 Conference and Exposition, San Diego (2014)
22. Simeonides, G.: Experimental and computational investigation of hypersonic flow about compression ramps. J. Fluid Mech. **283**, 17–42 (1995)
23. Simeonides, G.: Simple theoretical and semi-empirical convective heat transfer predictions for generic aerodynamic surfaces. YPA/1576/GS, ESA/ESTEC (1995)
24. Stanley, D.O., Unal, R. Joyner, C.R.: Application of Taguchi methods to propulsion system optimisation for SSTO vehicles. J. Spacecr. Rocket. **29**(4), 453–459 (1992)
25. Stanley, D.O., Engelund, W.C., Guinta, A.A., Unal, R.: Rocket-powered single-stage vehicle configuration selection and design. AIAA-1993–1053. In: AIAA/AHS/ASEE Aerospace Design Conference, Irvine, CA, 16–19 February 1993
26. Sudmeijer, K.J., Mooij, E.: Shape Optimisation for a small experimental re-entry module. AIAA-2002-5261. In: AIAA/AAAF 11th International Space Planes and Hypersonic Systems and Technologies Conference, Orleans (2002)
27. Taguchi, G.: System of Experimental Design. Engineering Methods to Optimise Quality and Minimise Costs, vol. 1, 2nd edn. UNIPUB/Kraus International Publications, White Plains, NY (1988)
28. Walker, S., Sherk, J., Shell, D., Schena, R., Bergmann, J., Gladbach, J.: The DARPA/AF Falcon program: the hypersonic technology vehicle #2 (HTV-2) flight demonstration phase. AIAA-2008-2539. In: 15th AIAA International Space Planes and Hypersonic Systems and Technologies Conference, Dayton, OH, April 29–May 1 2008
29. Weihs, H., Turner, J., Hörschgen-Eggers, M.: SHEFEX II – the next step within flight testing of re-entry technology. IAC-06-D2.5.03. In: 57th International Astronautical Congress, Valencia, 2–6 October 2006
30. Yonemoto, K., Inatani, Y.: Analytical interpretation on lateral/directional stability and controllability of high angle-of-attack reentry flight. Report No. 630. The Institute of Space and Astronautical Science, Tokyo (1988)
31. Zoby, E.V., Moss, J.N., Sutton, K.: Approximate convective-heating equations for hypersonic flow. J. Spacecr. Rocket. **18**(1), 64–70 (1981)

Rigorous Global Optimization for Collision Risk Assessment on Perturbed Orbits

Alessandro Morselli, Roberto Armellin, Pierluigi Di Lizia, and Franco Bernelli-Zazzera

Abstract In this chapter, a method to assess the occurrence of impacts between objects (either spacecraft or space debris) orbiting around the Earth is presented. The method is based on the computation of the minimum distance between two evolving orbits by means of a rigorous global optimizer. Analytical solutions of artificial satellite motion are utilized to account for perturbative effects of Earth's zonal harmonics, atmospheric drag, and third body. It is shown that the method can effectively compute the intersection between perturbed orbits and hence identify pairs of space objects on potentially colliding orbits. Test cases considering sun-synchronous, low perigee and earth-synchronous orbits are presented to assess the performances of the method.

Keywords Minimum orbit intersection distance • Space debris • Taylor models • Global optimization • Orbital perturbations

List of Acronyms

MOID	Minimum orbital intersection distance
FFT	Fast Fourier transform
DA	Differential algebra
LDB	Linear dominated bounder

A. Morselli (✉)
European Space Agency, European Space Operations Centre, Robert-Bosch-Straße 5, 64293 Darmstadt, Germany
e-mail: alessandro.morselli@esa.int

R. Armellin
Departamento de Matemáticas y Computación, Universidad de La Rioja, C/Luis de Ulloa s/n, 26004 Logroño, Spain
e-mail: roberto.armellin@unirioja.es

P. Di Lizia • F. Bernelli-Zazzera
Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy
e-mail: pierluigi.dilizia@polimi.it; franco.bernelli@polimi.it

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization and Its Applications 114, DOI 10.1007/978-3-319-41508-6_9

QFB	Quadratic fast bounder
ECI	Earth Centered Inertial
UT	Universal time

1 Introduction

The probability of a close encounter or an impact between two bodies moving on orbits around the Earth can be assessed by computing the minimum distance between their orbits. In astrodynamics this quantity is usually referred to as minimum orbital intersection distance (MOID), and is usually computed by looking at all the stationary points of the square of the Euclidean distance, d^2, between two points on the first and the second orbit, respectively. Several algorithms have been proposed for the solution of this problem [9, 15, 30]. These algorithms are mainly affected by the difficulty in dealing with a nonlinear one-dimensional equation appearing when a component of the critical points of d^2 is sought for. In [22] the problem was algebraically solved in the case of two Keplerian elliptic orbits by finding all the critical points of a trigonometric polynomial of degree eight, obtained with Gröbner bases theory. Furthermore, it was proven that a trigonometric polynomial of degree less than eight does not exist. Later in [4] the method was extended to all types of conic sections. In [13] an algorithm based on the resultant theory and the Fast Fourier Transform (FFT) is introduced to perform the elimination of one variable; an upper bound on the maximum number of critical points (if they are finitely many) is also obtained by using Newton's polytopes and Bernstein's theorem. Several improvements to the algorithm were later presented in [14] as well as the extension to unbounded Keplerian orbits. In [2], the computation of MOID for Keplerian orbits is approached as a global optimization problem. The rigorous global optimizer COSY-GO [28] is run on either the square distance function or the square of its gradient for the computation of the MOID or all the stationary points of d^2, respectively.

All the aforementioned methods make the assumption of Keplerian orbits. In this chapter, the approach presented in [2] is applied to non-Keplerian orbits. The two-body approximation for space objects orbiting the Earth can be too crude, even when small time intervals are considered. Perturbations, such as non-symmetrical gravity field, atmospheric drag, solar radiation pressure, and luni-solar perturbation, act on the orbiting bodies and, as a consequence, the motion is no longer Keplerian. Analytical approximations suitable for different orbital regimes (see [1, 18, 19]) can be used to efficiently describe the dependence of the orbital parameters on time. As a result, the square distance function d^2 becomes time dependent. Thus, two true anomalies and an epoch, which determines the current orbital configuration, are necessary to identify the MOID. Note that the MOID computation remains a geometrical problem, as time is used to describe the evolution of the orbital parameters only. No information on minimum distance between trajectories is gained (we refer to this as the synchronization problem); but a small MOID in the perturbed dynamics indicates that, during the considered time window, the perturbations modify the orbits in such a way that a conjunction is possible. This approach can be possibly useful to pick out, over long time intervals, threatening configurations, otherwise missed in a two-body approximation.

The chapter is organized as follows. In Sects. 2 and 3 some notes on the theory of Taylor models are given and the rigorous global optimizer COSY-GO is briefly described. The problem formulation is introduced in Sect. 4, underlining the main characteristics of the analytical solutions considered as well as the procedure for objective function computation. Some numerical experiments are presented and discussed in Sect. 5. Final remarks conclude the chapter.

2 Notes on Taylor Models

Verified global optimization needs the determination of rigorous upper and lower bounds of the objective function in order to implement a branch-and-bound method [21]. The commonly used interval approach has excelled in solving this problem elegantly from both a formal and an implementational viewpoint. However, there are situations where the method has limitations for extended or complicated calculations because of the dependency problem, which is characterized by a cancellation of various sub-parts of the function that cannot be detected by direct use of interval methods. This effect often leads to pessimism and sometimes even drastic overestimation of range enclosure. Furthermore, the sharpness of intervals resulting from calculations typically scales linearly with the sharpness of the initial discretization intervals. For complicated problems, and in particular higher dimensions, this sometimes significantly limits the sharpness of the result that can be obtained [26].

The Taylor model approach enables the computation of fully mathematically rigorous range enclosures while largely avoiding many of the limitations of the conventional interval method [25]. The method is based on the inductive local modelling of functional dependencies by a polynomial with a rigorous remainder bound, and as such represents a hybrid between formula manipulation, interval methods, and methods of computational differentiation [7, 12].

An n-th order Taylor model of a multivariate function f that is $(n+1)$-times continuously partially differentiable on the domain D, consists of the n-th order multivariate Taylor polynomial P expanded around a point $\boldsymbol{x}_0 \in D$ and representing a high-order approximation of the function f, and a remainder error interval I for verification such that

$$\forall \boldsymbol{x} \in D, \quad f(\boldsymbol{x}) \in P(\boldsymbol{x} - \boldsymbol{x}_0) + I. \tag{1}$$

From Taylor's theorem, it is clear that the width of the remainder interval I can be chosen to scale with the domain size proportional to $|\boldsymbol{x} - \boldsymbol{x}_0|^{n+1}$. The practical computation of P and I is based on Taylor model arithmetic, which carries P and I

through all the operations comprising I. By choosing the size $|\boldsymbol{x} - \boldsymbol{x}_0|$ sufficiently small and the order n sufficiently high, the size of the remainder interval I can be kept very small in practice. The bulk of the functional dependency is kept in the polynomial part P with point coefficients, and there is no interval arithmetic associated inflation that happens in the polynomial part. Thus, the interval related overestimation is rather optimally suppressed with the Taylor model method [26]. The implementation of the method in the code COSY Infinity [6, 25] supports binary operations and standard intrinsic functions, as well as the antiderivative operation which widens the applications of the method. Note that when only the polynomial part P of the Taylor model is considered, also the analytic operation of differentiation can be introduced, so finalizing the definition of a differential algebraic (DA) structure [5].

The Taylor model approach has the following important properties:

1. The ability to provide rigorous enclosures of any function given by a finite computer code list by a Taylor polynomial and a remainder bound with a sharpness that scales with order $(n + 1)$ of the width of the domain.
2. The computational expense increases only moderately with order, allowing the computation of sharp range enclosures even for complicated functional dependencies with significant dependency problem.
3. The computational expense of higher dimensions increases only very moderately, significantly reducing the "curse of dimensionality".

The structure of Taylor models naturally represents a rich resource of information. In particular, the coefficients of the polynomial part P of a Taylor model are nothing but the derivatives up to order n. Consequently, when representing a function f by a Taylor model (P, I) on a computer, we also obtain the local slope, Hessian and higher order derivatives. When a task is focused on range bounding, those pieces of information become particularly useful.

While naive range bounding of Taylor models, namely merely evaluating each monomial of P using interval arithmetic then summing up all the contributions as well as the remainder interval I [27], already exhibits the superiority over the mere interval arithmetic and the more advanced centered form [25], the active utilization of those additional pieces of information in Taylor models has a lot of potential of developing efficient range bounders. Based on this observation, various kinds of Taylor model based range bounders have been developed [8], and among them the linear dominated bounder (LDB) and the quadratic fast bounder (QFB) are the backbones of Taylor model based verified global optimizer COSY-GO that will be discussed afterward.

The linear dominated bounder (LDB) is based on the fact that for Taylor models with sufficiently small remainder bound, the linear part of the Taylor model dominates the behavior, and this is also the case for range bounding. The linear dominated bounder utilizes the linear part as a guideline for iterative domain reduction to bound Taylor models. Around an isolated interior minimizer, the Hessian of a function f is positive definite, so the purely quadratic part of a Taylor model (P, I) which locally represents f, has a positive definite Hessian matrix.

The quadratic fast bounder provides a lower bound of a Taylor model cheaply when the purely quadratic part is positive definite. More details on polynomial bounders are given in [28].

3 COSY-GO

COSY-GO [8] is a branch-and-bound optimization algorithm employing local domain reduction techniques exploiting the bounding performances assured by Taylor model methods. Should the global minimum of a sufficiently regular scalar function f on a given domain $A \subseteq \Re^m$ wished to be evaluated, the algorithm starts with an initial value for the global optimum, the *cut-off* value, and then proceeds on analyzing at each step a subdomain for possible elimination or reduction. At each step the following tasks are performed.

1. A rigorous lower bound l of the objective function is obtained on the subdomain of interest using various bounding schemes hierarchically with the hope of showing that l lies above the already established cut-off value, which will allow elimination of the subdomain. A first assessment is made whether the remainder bound of the Taylor model at hand is sufficiently small; if it is not, then the underlying function exhibits too much detail for modeling by local estimators, and the subdomain is split in the direction of fastest change of the function.
2. If the remainder bound is sufficiently small, as a first test the polynomial part of the objective function is evaluated in interval arithmetic. When it fails to eliminate the box, the LDB bounder is applied. If it also fails to eliminate the box, and if the quadratic part of the polynomial representation of the objective function P is positive definite, the QFB bounder is applied.
3. If the just studied subdomain of interest cannot be eliminated, but is seen to have a lower bound close to the current cut-off values, domain reduction techniques are brought to bear based on the LDB and QFB algorithms to reduce the subdomain in size. Once these methods are applicable, they will allow to cut the subdomain of interest and rapidly reduce the active volume.
4. The cut-off value is updated using various schemes. First, the linear and quadratic parts of the Taylor polynomial are utilized to obtain a potential cut-off update. In particular, if the quadratic part of the polynomial is positive definite, the minimizer of the quadratic polynomial is tested. If the quadratic part is not positive definite, the minimizer of the quadratic part in the direction of the negative gradient is tested. For objective functions of nontrivial cost, as in the example at hand, also more sophisticated local searches within and near the current subdomain may be carried out.

The algorithm continues to reduce and examine the domain until the minimum dimension allowed is reached. The result of the optimization is the validated enclosure of the global minimum of the problem.

4 Problem Formulation

In [2], the MOID is computed by running the optimizer COSY-GO on either the square distance function between two bodies moving on perturbed Earth orbits or the square of its gradient. The considered orbits are Keplerian and the objective function is the square of the Euclidean distance d^2 between two generic points belonging to the first and the second orbit respectively. Since the two orbits are Keplerian, their five Keplerian elements a_i, e_i, I_i, Ω_i, and ω_i, where $i = 1, 2$, are constants and d^2 is a function of the true anomalies ν_1 and ν_2 only (see Fig. 1).

The position of an object in the Earth Centered Inertial (ECI) reference frame is computed from its orbital elements using the following equations

$$\begin{aligned} r_i &= \frac{a_i\left(1-e_i^2\right)}{1+e_i\cos(\nu_i)} \\ \boldsymbol{r}_i &= \begin{Bmatrix} r_{I_i} \\ r_{J_i} \\ r_{K_i} \end{Bmatrix} = r_i \begin{Bmatrix} \cos(\Omega_i)\cos(\omega_i+\nu_i) - \sin(\Omega_i)\cos(I_i)\sin(\omega_i+\nu_i) \\ \sin(\Omega_i)\cos(\omega_i+\nu_i) + \cos(\Omega_i)\cos(I_i)\sin(\omega_i+\nu_i) \\ \sin(I_i)\sin(\omega_i+\nu_i) \end{Bmatrix}. \end{aligned} \tag{2}$$

The square distance d^2 is given by

$$d^2 = (r_{I_1} - r_{I_2})^2 + (r_{J_1} - r_{J_2})^2 + (r_{K_1} - r_{K_2})^2 , \tag{3}$$

and, for Keplerian orbits, it is a function of the two true anomalies ν_1 and ν_2 only.

In a perturbed two-body problem, the Keplerian elements a_i, e_i, I_i, Ω_i, and ω_i of the two objects, and, consequently, the square distance of their orbits

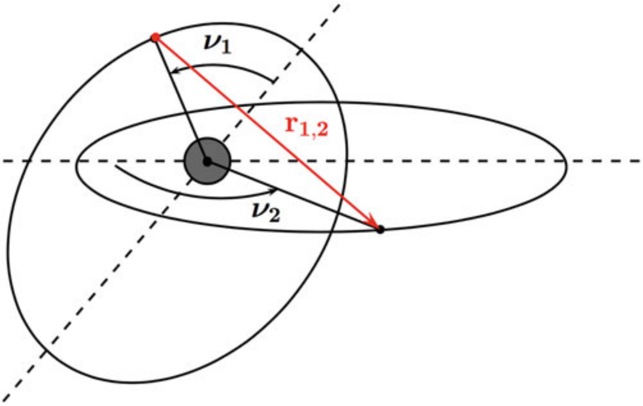


Fig. 1 Distance between two orbits: given the Keplerian elements of both orbits, the distance is univocally determined by the pair (ν_1, ν_2)

become functions of time t. Thus, the objective function evaluation must include a mathematical model to account for the dependence on t. An analytical model is used in this chapter. This section is devoted to illustrate the approach and its use within the formulation of the objective function.

4.1 Analytical Representation of Orbital Dynamics

The time dependence of the Keplerian elements is obtained by means of analytical theories for satellite motion. Thus, the value of the orbital elements at time t is computed through the evaluation of analytical expressions. Usually the perturbative effects are divided into long-period and short-period. For the former, explicit functions of time are available, whereas the latter can only be evaluated solving Kepler's equation.

The analytical solutions adopted in this chapter are

- Aksnes's solution [1]: zonal harmonics from J_2 to J_5;
- HANDE [18]: zonal harmonics from J_2 to J_4, atmospheric drag;
- SGP4 [19]: zonal harmonics from J_2 to J_4, luni-solar perturbation, daily resonance with tesseral harmonics $J_{2,2}$, $J_{3,1}$, $J_{3,3}$.

Based on the perturbations included in each model, HANDE is suitable for low-Earth orbits, where the effect of atmosphere is not negligible. It is worth highlighting that HANDE's density model is arbitrary and even tabulated values can be adopted. Thus, depending on the considered orbit, the best fitting model can be chosen. SGP4 is used for geostationary orbits, where atmospheric drag is negligible and luni-solar perturbations are comparable with zonal harmonic J_2 perturbative acceleration. Aksnes' solution is considered for intermediate orbits, for which the atmospheric drag can be neglected and the luni-solar perturbation is still orders of magnitude lower than zonal harmonics perturbations. No matter which model is selected, the orbital evolution of one object is represented by a single analytical solution. It is thus assumed that the selected model can accurately predict the motion of the object for the entire time window at hand. The details of these methods are reported in the following sections.

4.1.1 Aksnes Zonal Harmonics Solution

The Aksnes zonal harmonics solution was originally developed in 1971 [1]. Since the model is used in this chapter to compute the square distance of two orbiting objects, this section focuses on the description of the terms for the computation of the position vector.

Given the initial mean Keplerian elements (a_0, e_0, I_0, $l_0 = M_0$, $g_0 = \omega_0$, $h_0 = \Omega_0$) the following constants are computed

$$\begin{aligned}
c &= \cos(I_0) \\
s &= \sin(I_0) \\
\eta &= \sqrt{1 - {e_0}^2} \\
p &= a_0\,\eta^2 \\
n_0 &= \sqrt{\frac{\mu_\oplus}{{a_0}^3}} \\
\gamma &= J_2\left(\frac{R_\oplus}{p}\right)^2 \\
\gamma_j &= \frac{J_j}{J_2^2}\left(\frac{R_\oplus}{p}\right)^{j-4}, \qquad j = 3, 4, 5.
\end{aligned} \tag{4}$$

where $\mu_\oplus$ and $R_\oplus$ are the gravitational parameter and mean radius of the Earth respectively; $J_j, j = 2, 3, 4, 5$, are the zonal harmonics.

The constant rates of the Delaunay's variables l, g and h are given by

$$\begin{aligned}
\dot{l} = n_0 \Bigg\{ 1 - \frac{3}{4}\gamma\,\eta \Bigg[1 - 3c^2 - \frac{1}{32}\gamma\,\{10\,(1 - 6c^2 + 13c^4) - 5\,(5 - 18c^2 + 5c^4)\,e_0^2\ + \\
+16\eta\,(1 - 6c^2 + 9c^4) - 15\,\gamma_4\,(3 - 30c^2 + 35c^4)\,e_0^2\} \Bigg]\Bigg\}
\end{aligned} \tag{5}$$

$$\begin{aligned}
\dot{g} = -\frac{3}{4}\gamma\,n_0 \Bigg[1 - 5c^2 + \frac{1}{32}\gamma\,\{2\,(5 + 43c^2)\,(1 - 5c^2) + (25 - 126c^2 + 45c^4)\,e_0^2\ + \\
-24\,\eta\,(1 - 8c^2 + 15c^4) + 20\,\gamma_4\,(3 - 36c^2 + 49c^4) + 45\,\gamma_4\,(1 - 14c^2 + 21c^4)\,e_0^2\} \Bigg]
\end{aligned} \tag{6}$$

$$\begin{aligned}
\dot{h} = -\frac{3}{2}\gamma\,c\,n_0 \Bigg[1 - \frac{1}{16}\gamma\,\{4 - 40c^2 - (9 - 5c^2)\,e_0^2 + 12\eta\,(1 - 3c^2)\ + \\
-5\gamma_4\,(3 - 7c^2)\,(2 + 3e_0^2)\} \Bigg].
\end{aligned} \tag{7}$$

Thus, their mean values can be computed as

$$\begin{aligned}
h &= h_0 + \dot{h}\,t \\
g &= g_0 + \dot{g}\,t \\
l &= l_0 + \dot{l}\,t\,,
\end{aligned} \tag{8}$$

where t is time since reference epoch.

The longitude $\tilde{\lambda}$ and latitude $\tilde{\varphi}$, measured from the Vernal point meridian and the equator respectively, are obtained through the following equations

$$\begin{aligned} I &= \arccos\left(c\right) \\ \tilde{\varphi} &= \arcsin\left(\sin(I)\cos(u)\right) \\ \tilde{\lambda} &= \arctan\left(\frac{\cos(I)\sin(u)}{\cos(u)}\right) + h\ , \end{aligned}$$

where $u = g + \nu$, with ν being the true anomaly.

The position in ECI is obtained as

$$\mathbf{r} = r\,\hat{\mathbf{u}}, \tag{9}$$

where

$$\hat{\mathbf{u}} = \begin{Bmatrix} \cos(\tilde{\lambda})\cos(\tilde{\varphi}) \\ \sin(\tilde{\lambda})\cos(\tilde{\varphi}) \\ \sin(\tilde{\varphi}) \end{Bmatrix}. \tag{10}$$

4.1.2 HANDE

The mathematical procedure for the formulation of the analytical solution of the HANDE model can be found in [18] and [16]. The method for the long-period contribution to the position vector is reported hereafter.

First of all, the solution is initialised by computing the secular variations of the six Keplerian elements due to atmospheric drag, as well as the derivatives of mean motion n and eccentricity e. To this aim, the mean Keplerian elements at epoch are required, together with satellite ballistic coefficient B. The first derivatives of mean motion and eccentricity can be computed by means of the integrals

$$\begin{aligned} f^{(2)}_{n,D} &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{3}{2}\, B\, n\, \beta^{-2}\, \rho\, v \left[1 + e^2 + 2e\cos(\nu) - \frac{\omega_a}{n}\,\beta^3\,\cos(I)\right] dM \\ f^{(2)}_{e,D} &= -\frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{1}{2}\, B\, \rho\, v \left[2\left(e + \cos(\nu)\right) - \frac{\omega_a}{n}\,\beta^3\,\cos(I)\frac{\left(2\cos(\nu) + e + e\,\cos^2(\nu)\right)}{\left(1 + e\cos(\nu)\right)^2}\right] dM\ , \end{aligned} \tag{11}$$

where ω_a is the rotational velocity of Earth's atmosphere, $\beta = \sqrt{1 - e^2}$, ρ is the atmospheric density at the satellite altitude above Earth's surface (hence a function of true anomaly ν), and the satellite velocity v with respect to atmosphere is given by

$$v = \frac{n\,a}{\beta}\sqrt{\left(1 + e^2 + 2\,e\,\cos(\nu)\right) - 2\,\frac{\omega_a}{n}\,\beta^3\,\cos(I) + \frac{{\omega_a}^2}{n^2}\beta^6\frac{1 - \sin^2(I)\sin^2(u)}{\left(1 + e\cos(\nu)\right)^2}}. \tag{12}$$

The integration is facilitated by the change of variables

$$dM = \frac{\beta^3}{(1 + e\cos(\nu))^2} d\nu \ .$$

The integrals, also referred to as *drag functions*, are evaluated using a 13-points Gauss-Legendre formula (see [23]). In this approach there is no constraint on the choice of the density model, that can be either mathematical (e.g. power density or power function) or tabulated. Also the higher-order derivatives are obtained numerically, using a 7-points central difference formula. Hence, values of n, $\dot{n}$, e, and $\dot{e}$ at the instants $\pm 3\tau$, $\pm 2\tau$, $\pm\tau$, as well as their values at epoch are required. Denoting mean motion and eccentricity, as well as their first derivatives at epoch, with a subscript 0, the values at $\pm\tau$ are

$$n_{\pm\tau} = n_0 \pm \dot{n}_0 \tau \ \text{ and } \ e_{\pm\tau} = e_0 \pm \dot{e}_0 \tau \ .$$

Given the values of n and e at time $\pm\tau$, the derivatives $\dot{n}_{\pm\tau}$ and $\dot{e}_{\pm\tau}$ are obtained by evaluating the integrals (11) with the updated values of mean motion and eccentricity. The values at the other instants can be obtained using the same strategy. When all seven values of eccentricity and mean motion are available, the derivatives $\dot{e}$, $\ddot{e}$, $\dddot{e}$, $\dot{n}$, $\ddot{n}$ and $\dddot{n}$ can be computed using formulae in [11]. To compute $\ddddot{n}$ the formula for the third derivative should be employed, using the first derivatives of n at times $\pm 3\tau$, $\pm 2\tau$, $\pm\tau$.

The secular variations of the other four Keplerian elements can be computed from the drag functions

$$f^{(2)}_{I,D} = -\frac{1}{2} \int_{-\pi}^{\pi} \frac{1}{2} B \rho v \frac{\omega_a}{n} \beta^{-1} \sin(I) \frac{r^2}{a^2} \cos^2(u)\, dM \tag{13}$$

$$f^{(2)}_{\omega,D} = \frac{1}{2} \int_{-\pi}^{\pi} \left[\frac{1}{2} B \rho v \frac{\omega_a}{n\beta} \cos(I) \frac{r^2}{a^2} \left(sin(\nu)\cos(\nu) + \sin(u)\cos(u) \right) - \Delta M_D \right] dM \tag{14}$$

$$f^{(2)}_{\Omega,D} = -\frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{1}{2} B \rho v \frac{\omega_a}{n} \beta^{-1} \frac{r^2}{a^2} \sin(u)\cos(u) dM \tag{15}$$

$$\begin{aligned} f^{(2)}_{M,D} = \frac{1}{2} \int_{-\pi}^{\pi} \Bigg[\frac{1}{2} B \rho v \Bigg\{ & 2e\sin(E) - \frac{2e}{1+\beta} \sin(\nu) \ + \\ & - \frac{\omega_a}{n} \cos(I) \frac{r^2}{a^2} \left[\frac{-2e}{\beta(1+\beta)} + \cos(\nu) \right] \sin(f) \Bigg\} + \Delta M_D \Bigg] dM \ , \end{aligned} \tag{16}$$

where $u = \nu + \omega$, $E(\nu)$ is the eccentric anomaly $r(\nu)$ the orbit radius and

$$\Delta M_D = B\rho v \left[1 - \frac{\omega_a}{n\,\beta} \cos(I)\, \frac{r^2}{a^2} \right] \frac{\sin(\nu)}{e} \ .$$

When $e \rightarrow 0$ the values of ΔM_D becomes singular, due to arbitrary definition of the location of perigee in circular orbits. Since this term exactly vanishes in the sum $M + \omega$, numerical singularities are avoided setting $\Delta M_D = 0$ when $e < 1 \times 10^{-6}$.

For highly eccentric orbits density decreases rapidly away from the pericentre. In this case the integrals can be approximated by

$$I = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\nu) d\nu \simeq \frac{1}{2\pi} \int_{-b}^{b} f(\nu) d\nu \ .$$

The value of b can be computed as

$$\begin{aligned} \cos(b) &= \frac{a\,e\ (1-e) - \Delta r}{a\,e\,(1-e) + e\,\Delta r} && \text{for} \Delta r < 2\,a\,e \\ \cos(b) &= -1 && \text{for} \Delta r \geqslant 2\,a\,e \ , \end{aligned}$$

with

$$\begin{aligned} \Delta r &= q \left[A_0 + A_1\,q + A_2\,q^2 + A_3\,q^3 + A_4\,q^4 + A_5\,q^5 + A_6\,q^6 \right] \\ q &= a_0\,(1 - e_0) - R_\oplus \\ A_0 &= -3.1301240 \\ A_1 &= 6.1710434 \times 10^{-2} \\ A_2 &= -3.4111266 \times 10^{-4} \\ A_3 &= 8.7321429 \times 10^{-7} \\ A_4 &= -1.1225340 \times 10^{-9} \\ A_5 &= 7.1123451 \times 10^{-13} \\ A_6 &= -1.7765750 \times 10^{-16} \ , \end{aligned}$$

where q is the perigee height above Earth's mean surface in kilometres and Δr is the altitude change in kilometres for a drop in density by a factor 100. This approximation avoids the underestimation of Keplerian elements secular rates that would occur computing numerically the integrals on the whole domain $[-\pi, \pi]$.

The total secular rates of Keplerian elements due to zonal harmonics and drag are computed using the following equations

$$\dot{I}_0 = f^{(2)}_{I,D} \tag{17}$$

$$\begin{aligned}\dot{\omega}_0 =& f^{(2)}_{\omega,D} + \frac{3}{4} n_0 J_2 \left(\frac{R\oplus}{p_0}\right)^2 \left(4 - 5\sin^2(I_0)\right) + \frac{3}{16} n_0 J_2^2 \left(\frac{R\oplus}{p_0}\right)^4 \left[\left(4 - 5\sin^2(I_0)\right)\left(12 + \right.\right.\\ & \left.\left. -\frac{43}{4}\sin^2(I_0) + 6\beta_0\left(1 - \frac{3}{2}\sin^2(I_0)\right)\right) + \left(7 - \frac{9}{2}\sin^2(I_0) - \frac{45}{8}\sin^4(I_0)\right) e_0^2\right] + \\ & -\frac{15}{32} n_0 J_4 \left(\frac{R\oplus}{p_0}\right)^4 \left[16 - 62\sin^2(I_0) + 49\sin^4(I_0) + \right.\\ & \left.\left(18 - 63\sin^2(I_0) + \frac{189}{4}\sin^4(I_0)\right) e_0^2\right]\end{aligned} \tag{18}$$

$$\begin{aligned}\dot{\Omega}_0 =& f^{(2)}_{\Omega,D} - \frac{3}{2} n_0 J_2 \left(\frac{R\oplus}{p_0}\right)^2 \cos(I_0) - \frac{3}{8} n_0 J_2^2 \left(\frac{R\oplus}{p_0}\right)^4 \cos(I_0)\left[9 - 10\sin^2(I_0) + \right.\\ & \left. + e_0^2\left(1 + \frac{5}{4}\sin^2(I_0)\right) + 6\beta_0\left(1 - \frac{3}{2}\sin^2(I_0)\right)\right] + \\ & + \frac{15}{16} n_0 J_4 \left(\frac{R\oplus}{p_0}\right)^4 \cos(I_0)\left(4 - 7\sin^2(I_0)\right)\left(1 + \frac{3}{2}e_0^2\right).\end{aligned} \tag{19}$$

The constants of the long-period periodic variations of the osculating Keplerian elements are computed as well, since they depend only on the mean elements at epoch:

$$e_{LP_1} = C_1\, e_0\, \beta_0^2\, \sin^2(I_0) \tag{20}$$

$$e_{LP_2} = -C_2\, \beta_0^2\, \sin(I_0) \tag{21}$$

$$I_{LP_1} = -C_1\, e_0^2\, \sin(I_0)\, \cos(I_0) \tag{22}$$

$$I_{LP_2} = C_2\, e_0\, \cos(I_0) \tag{23}$$

$$\begin{aligned}\omega_{LP_1} =& -C_1\left(\sin^2(I_0) - e_0^2 + \frac{9}{2}e_0^2\sin^2(I_0)\right) + \\ & -\frac{1}{8}J_2\left(\frac{R\oplus}{p_0}\right)^2 \frac{1}{4 - 5\sin^2(I_0)}\left(e_0^2\sin^2(I_0)\right)\left(1 + 5\frac{J_4}{J_2^2}\right)\end{aligned} \tag{24}$$

$$\omega_{LP_2} = -C_2 e_0 \sin(I_0) \frac{35\cos^2(I_0)}{4 - 5\sin^2(I_0)} + \Delta\omega - \Delta M \tag{25}$$

$$\Omega_{LP_1} = -C_1 e_0^2 \cos(I_0) + \frac{5}{8} J_2 \left(\frac{R\oplus}{p_0}\right)^2 \frac{1}{4 - 5\sin^2(I_0)} \left(e_0^2 \cos(I_0)\sin^2(I_0)\right)\left(3 + 7\frac{J_4}{J_2^2}\right) \tag{26}$$

$$\Omega_{LP_2} = \frac{C_2}{4 - 5\sin^2(I_0)} e_0 \sin(I_0)\left[15\cos(I_0) + \frac{4}{1 + \cos(I_0)}\right] - \Delta\omega. \tag{27}$$

where the constants C_1, C_2, and $\Delta\omega$ are defined as

$$C_1 = \frac{1}{8} J_2 \left(\frac{R\oplus}{p_0}\right)^2 \frac{1}{4 - 5\sin^2(I_0)} \left[14 - 15\sin^2(I_0) + \frac{J_4}{J_2^2}\left(30 - 35\sin^2(I_0)\right)\right] \tag{28}$$

$$C_2 = \frac{1}{2}\frac{J_3}{J_2}\left(\frac{R\oplus}{p_0}\right) \tag{29}$$

$$\begin{cases} \Delta\omega = C_2 \dfrac{4\,e_0}{\sin(I_0)} \dfrac{1}{4 - 5\sin^2(I_0)} & \text{if} I_0 \geqslant 1 \times 10^{-6} \\ \Delta\omega = 0 & \text{if} I_0 < 1 \times 10^{-6} \,. \end{cases} \tag{30}$$

The last constants computed during initialization are

$$\frac{\dot{\beta}_0}{\beta_0} = -\frac{e_0\,\dot{e}_0}{\beta_0^2} \tag{31}$$

$$\frac{\ddot{\beta}_0}{\beta_0} = -\frac{1}{\beta_0^4}\left(\dot{e}_0^2 + e_0\,\ddot{e}_0\,\beta_0^2\right) \tag{32}$$

$$D_{1,-3} = \frac{1}{2}\left(\frac{7}{3}\frac{\dot{n}_0}{n_0} - 3\frac{\dot{\beta}_0}{\beta_0}\right) \tag{33}$$

$$D_{1,-4} = \frac{1}{2}\left(\frac{7}{3}\frac{\dot{n}_0}{n_0} - 4\frac{\dot{\beta}_0}{\beta_0}\right) \tag{34}$$

$$D_{2,-3} = \frac{1}{6}\left(\frac{28}{9}\frac{\dot{n}_0^2}{n_0^2} + \frac{7}{3}\frac{\ddot{n}_0}{n_0} - 14\frac{\dot{n}_0}{n_0}\frac{\dot{\beta}_0}{\beta_0} + 12\frac{\dot{\beta}_0^2}{\beta_0^2} - 3\frac{\ddot{\beta}_0}{\beta_0}\right) \tag{35}$$

$$D_{2,-3} = \frac{1}{6}\left(\frac{28}{9}\frac{\dot{n}_0^2}{n_0^2} + \frac{7}{3}\frac{\ddot{n}_0}{n_0} - \frac{56}{3}\frac{\dot{n}_0}{n_0}\frac{\dot{\beta}_0}{\beta_0} + 20\frac{\dot{\beta}_0^2}{\beta_0^2} - 4\frac{\ddot{\beta}_0}{\beta_0}\right)\,. \tag{36}$$

The second step of the method consists in applying both secular and long-period periodic variations of the mean elements, through the equations

$$n' = n_0 + \dot{n}_0\, t + \frac{1}{2}\ddot{n}_0\, t^2 + \frac{1}{6}\dddot{n}_0\, t^3 + \frac{1}{24}\ddddot{n}_0\, t^4 \tag{37}$$

$$\omega_{\text{sec}} = \omega_0 + \dot{\omega}_0\, t \tag{38}$$

$$e' = e_0 + \dot{e}_0\, t + \frac{1}{2}\ddot{e}_0\, t^2 + \frac{1}{6}\dddot{e}_0\, t^3 + e_{LP_1}\left(\cos^2(\omega_{\text{sec}}) - \cos^2(\omega_0)\right) + e_{LP_2}\left(\sin(\omega_{\text{sec}}) - \sin(\omega_0)\right) \tag{39}$$

$$I' = I_0 + \dot{I}_0\, t + I_{LP_1}\left(\cos^2(\omega_{\text{sec}}) - \cos^2(\omega_0)\right) + I_{LP_2}\left(\sin(\omega_{\text{sec}}) - \sin(\omega_0)\right) \tag{40}$$

$$\begin{aligned}\Omega' = {} & \Omega_0 + \dot{\Omega}_0\, t + \Omega_{LP_1}\left(\sin(\omega_{\text{sec}})\cos(\omega_{\text{sec}}) - \sin(\omega_0)\cos(\omega_0)\right) + \\ & + \Omega_{LP_2}\left(\cos(\omega_{\text{sec}}) - \cos(\omega_0)\right) - \frac{3}{2}\, n_0\, J_2 \left(\frac{R_\oplus}{p_0}\right)^2 \cos(I_0)\left(D_{1,-4}\, t^2 + D_{2,-4}\, t^3\right)\end{aligned} \tag{41}$$

$$\omega' = \omega_{\text{sec}} + \omega_{LP_1}\left(\sin(\omega_{\text{sec}})\cos(\omega_{\text{sec}})\sin(\omega_0)\cos(\omega_0)\right) + \omega_{LP_2}\left(\cos(\omega_{\text{sec}}) - \cos(\omega_0)\right)\ . \tag{42}$$

From the primed variables computed above the following change of variables is made

$$a' = \left(\frac{\mu_\oplus}{n'^2}\right)^{\frac{1}{3}} \tag{43}$$

$$\beta' = \sqrt{1 - e'^2} \tag{44}$$

$$p' = a'\, \beta^2 \tag{45}$$

$$u' = \nu' + \omega' \tag{46}$$

$$r' = \frac{a'\, \beta'^2}{1 + e' \cos(\nu')}\ , \tag{47}$$

where ν' is the true anomaly.

The position in Earth centred inertial (ECI) reference frame is given by

$$\mathbf{r} = r'\, \hat{\mathbf{u}}\ , \tag{48}$$

where

$$\hat{\mathbf{u}} = \begin{Bmatrix} -\sin(\Omega')\cos(I')\sin(u') + \cos(\Omega')\cos(u') \\ \cos(\Omega')\cos(I')\sin(u') + \sin(\Omega')\cos(u') \\ \sin(I')\sin(u') \end{Bmatrix}\ . \tag{49}$$

4.1.3 SGP4/SDP4

The Simplified General Perturbations #4 (SGP4) model is one of the orbit propagator developed during the 1970s by NORAD and U.S. Air Force Space Command. The SGP4 method is optimized to work with Two Line Elements (TLE), an

ASCII representation of the orbital parameters required to describe the motion of an Earth-orbiting object. The USSTRATCOM maintains a catalog of containing TLEs for all resident space objects that is accessible through Space-Track[1] and Celes-Track[2]websites. The details of SGP4 implementation can be found in [15, 17, 19, 20, 31]. The algorithm presented in this section is a simplified version of the SGP4 algorithm. All terms related to atmospheric drag have been dropped, as well as effects of 12 h resonances. This version is thus intended to be used with space objects with orbital periods larger than 225 min or orbiting in geostationary regime, and is indeed quite close to the Simplified Deep Space Perturbations #4 (SDP4). This theory, although initially developed as a stand-alone propagator, is now commonly embedded into SGP4 implementations, which is also referred to as SGP4/SDP4.

The orbital elements provided by the a TLE are mean elements, computed using Kozai mean motion. The first step is to recover from them the Brower mean motion through the equations

$$a_k = \left(\frac{\mu_\oplus}{n_k^2}\right)^{\frac{1}{3}} \tag{50}$$

$$\delta_1 = \frac{3}{2}\,\frac{k_2}{a_k^2}\,\frac{3\cos^2(I_0)-1}{\left(1-e_0^2\right)^{3/2}} \tag{51}$$

$$a_2 = a_1\left(1-\frac{1}{3}\delta_1-\delta_1^2-\frac{134}{81}\,\delta_1^3\right) \tag{52}$$

$$\delta_0 = \frac{3}{2}\,\frac{k_2}{a_2^2}\,\frac{3\cos^2(I_0)-1}{\left(1-e_0^2\right)^{3/2}} \tag{53}$$

$$n_0 = \frac{n_k}{1+\delta_0} \tag{54}$$

$$a_0 = \left(\frac{\mu_\oplus}{n_0^2}\right)^{\frac{1}{3}}\ . \tag{55}$$

where n_k is the Kozai mean motion and the coefficient k_2 is related to the second zonal harmonic and can be computed as

$$k_2 = \frac{1}{2}\,J_2R_\oplus^2. \tag{56}$$

[1] http://www.space-track.org.

[2] http://www.celestrack.org.

The next step is computing the secular effects of Earth's zonal harmonics

$$\dot{M}_z = \left[\frac{3}{2}\, k_2\, \frac{-1 + 3\cos^2(I_0)}{a_0^2\, \beta_0^3} + \frac{3}{16}\, k_2^2\, \frac{13 - 78\cos^2(I_0) + 137\cos^4(I_0)}{a_0^4\, \beta_0^7}\right] n_0 \tag{57}$$

$$\begin{aligned} \dot{\omega}_z = \Bigg[& -\frac{3}{2} k_2 \frac{1 - 5\cos^2(I_0)}{a_0^2\, \beta_0^4} + \frac{3}{16}\, k_2^2\, \frac{7 - 114\cos^2(I_0) + 395\cos^4(I_0)}{a_0^4\, \beta_0^8} + \\ & + \frac{5}{4}\, k_4\, \frac{3 - 36\cos^2(I_0) + 49\cos^4(I_0)}{a_0^4\, \beta_0^8} \Bigg] n_0 \end{aligned} \tag{58}$$

$$\dot{\Omega}_z = \left[-3k_2 \frac{\cos(I_0)}{a_0^2\, \beta_0^2} + \frac{3}{2}\, k_2^2\, \frac{4\cos(I_0) - 19\cos^3(I_0)}{a_0^2\, \beta_0^8} + \frac{5}{2} k_4 \cos(I_0) \frac{3 - 7\cos^2(I_0)}{a_0^4\, \beta_0^8}\right] n_0, \tag{59}$$

in which k_4 is related to the fourth zonal harmonics and is defined as

$$k_4 = -\frac{3}{8}\, J_4 R_{\oplus}^4. \tag{60}$$

The secular effects of luni-solar perturbations are also computed during initialization. For this purpose, the orbital elements of the Sun and the Moon at the TLE reference epoch are required. Following the procedure in [19] and [10], Moon's right ascension of ascending node (RAAN) $\Omega_{\leftmoon}$, inclination $I_{\leftmoon}$, and argument of pericentre $\omega_{\leftmoon}$ are computed. Sun's RAAN $\Omega_{\odot}$ and argument of pericentre $\omega_{\odot}$ are treated as constants instead.

For both the Sun and the Moon the following coefficients are calculated

$$\begin{aligned} a_{1x} &= \cos(\omega_x)\cos(\Omega_0 - \Omega_x) + \sin(\omega_x)\cos(I_x)\sin(\Omega_0 - \Omega_x) \\ a_{3x} &= -\sin(\omega_x)\cos(\Omega_0 - \Omega_x) + \cos(\omega_x)\cos(I_x)\sin(\Omega_0 - \Omega_x) \\ a_{7x} &= -\cos(\omega_x)\sin(\Omega_0 - \Omega_x) + \sin(\omega_x)\cos(I_x)\cos(\Omega_0 - \Omega_x) \\ a_{8x} &= \sin(\omega_x)\sin(I_x) \\ a_{9x} &= \sin(\omega_x)\sin(\Omega_0 - \Omega_x) + \cos(\omega_x)\cos(I_x)\cos(\Omega_0 - \Omega_x) \\ a_{10x} &= \cos(\omega_x)\sin(I_x) \\ a_{2x} &= a_{7x}\cos(I_0) + a_{8x}\sin(I_0) \\ a_{4x} &= a_{9x}\cos(I_0) + a_{10x}\sin(I_0) \\ a_{5x} &= -a_{7x}\sin(I_0) + a_8\cos(I_0) \\ a_{6x} &= -a_{9x}\sin(I_0) + a_{10x}\cos(I_0) \end{aligned} \tag{61}$$

$$
\begin{aligned}
X_{1x} &= a_{1x}\cos(\omega_0) + a_{2x}\sin(\omega_0)\\
X_{2x} &= a_{3x}\cos(\omega_0) + a_{4x}\sin(\omega_0)\\
X_{3x} &= -a_{1x}\sin(\omega_0) + a_{2x}\cos(\omega_0)\\
X_{4x} &= -a_{3x}\sin(\omega_0) + a_{4x}\cos(\omega_0)\\
X_{5x} &= a_{5x}\sin(\omega_0)\\
X_{6x} &= a_{6x}\sin(\omega_0)\\
X_{7x} &= a_{5x}\cos(\omega_0)\\
X_{8x} &= a_{6x}\cos(\omega_0)
\end{aligned}
\tag{62}
$$

$$
\begin{aligned}
Z_{31x} &= 12X_{1x}^2 - 3X_{3x}^2\\
Z_{32x} &= 24X_{1x}X_{2x} - 6X_{3x}X_{4x}\\
Z_{33x} &= 12X_{2x}^2 - 3X_{4x}^2\\
Z_{1x} &= 6\left(a_{1x}^2 + a_{2x}^2\right) + \left(1 + e_0^2\right)Z_{31x}\\
Z_{3x} &= 6\left(a_{3x}^2 + a_{4x}^2\right) + \left(1 + e_0^2\right)Z_{33x}\\
Z_{11x} &= -6a_{1x}\,a_{5x} + e_0^2\,(-24X_{1x}\,X_{7x} - 6X_{3x}\,X_{5x})\\
Z_{13x} &= -6a_{3x}\,a_{6x} + e_0^2\,(-24X_{2x}\,X_{8x} - 6X_{4x}\,X_{6x})\\
Z_{21x} &= 6a_{2x}\,a_{5x} + e_0^2\,(24X_{1x}\,X_{5x} - 6X_{3x}\,X_{7x})\\
Z_{23x} &= 6a_{4x}\,a_{6x} + e_0^2\,(24X_{2x}\,X_{6x} - 6X_{4x}\,X_{8x})\\
Z_{22x} &= 6a_{4x}\,a_{5x} + 6a_{2x}\,a_{6x} + e_0^2\,(24X_{2x}\;X_{5x} + 24X_{1x}\,X_{6x} - 6X_{4x}\,X_{7x} - 6X_{3x}\,X_{8x})\\
Z_{12x} &= -6a_{1x}\,a_{6x} - 6a_{3x}\,a_{5x} - e_0^2\,(24X_{2x}\,X_{7x} + 24X_{1x}\,X_{8x} + 6X_{3x}\,X_{6x} + 6X_{4x}\,X_{5x})\ ,
\end{aligned}
\tag{63}
$$

where subscript x stands for the considered perturbing body, while subscript 0 indicates the satellite mean elements at reference epoch. The secular rates for each body are given by

$$\dot{a}_x = 0 \tag{64}$$

$$\dot{e}_x = -15\,C_x\,n_x\,\frac{e_0\,\beta_0}{n_0}\,(X_{1x}\,X_{3x} + X_{2x}\,X_{4x}) \tag{65}$$

$$\dot{I}_x = -\frac{C_x\,n_x}{2\,n_0\,\beta_0}\,(Z_{11x} + Z_{13x}) \tag{66}$$

$$\dot{M}_x = -\frac{C_x\,n_x}{n_0}\left(Z_{1x} + Z_{3x} - 14 - 6e_0^2\right) \tag{67}$$

$$\dot{\Omega}_x = \begin{cases} \dfrac{C_x\, n_x\, \beta_0}{2\, n_0\, \beta_0 \sin(I_0)}\, (Z_{21x} + Z_{23x}) & \text{if} \quad |I_0| \leqslant 3\text{deg} \\ 0 & \text{if} \quad |I_0| > 3\text{deg} \end{cases} \tag{68}$$

$$\dot{\omega}_x = \frac{C_x\, n_x\, \beta_x}{n_0}\, (Z_{31x} + Z_{33x} - 6) - \dot{\Omega}_x \cos(I_0)\ . \tag{69}$$

where n_x is the perturbing body mean motion; $C_{☾}$ and $C_{\odot}$ are lunar and solar perturbation coefficients listed in [19]. For a nearly geosynchronous satellite or debris, whose period in minutes is in the interval $[1200, 1800]$, it is necessary to calculate the functions of inclination $F(I)$

$$F_{220} = \frac{3}{4}\,(1 + \cos(I_0))^2 \tag{70}$$

$$F_{311} = \frac{15}{16}\sin^2(I_0)\,(1 + 3\cos(I_0)) - \frac{3}{4}\,(1 + \cos(I_0)) \tag{71}$$

$$F_{330} = \frac{15}{8}\,(1 + \cos(I_0))^3\ , \tag{72}$$

and eccentricity function $G(e)$

$$G_{200} = 1 - \frac{5}{2}e_0^2 + \frac{13}{16}\,e_0^4 \tag{73}$$

$$G_{310} = 1 + 2e_0^2 \tag{74}$$

$$G_{300} = 1 - 6e_0^2 + \frac{423}{64}\,e_0^4\ . \tag{75}$$

The coefficients of the resonance terms are subsequently computed:

$$\delta_{S1} = 3\,\frac{n_0^2}{a_0^3}\,F_{311}\,G_{310}\,Q_{31} \tag{76}$$

$$\delta_{S2} = 6\,\frac{n_0^2}{a_0^2}\,F_{220}\,G_{200}\,Q_{22} \tag{77}$$

$$\delta_{S3} = 9\,\frac{n_0^2}{a_0^3}\,F_{330}\,G_{300}\,Q_{33}\ , \tag{78}$$

where Q coefficients are listed in Table 1.

To compute the resonance effect, a numerical integration scheme is required. The 1-day period initial conditions are computed during initialization and are given by

$$\lambda_{i_0} = M_0 + \omega_0 + \Omega_0 - \theta_{G0} \tag{79}$$

$$n_{i_0} = n_0 \tag{80}$$

Table 1 SGP4 tesseral and sectoral constants

n	2	3	3
m	2	1	3
Q_{nm}	1.7891679×10^{-6}	2.1460748×10^{-6}	2.2123015×10^{-7}
λ_{nm} [rad]	2.88431980	0.13130908	0.37448087

$$\dot{\lambda}_{i_0} = \dot{M}_z + \dot{M}_\odot + \dot{M}_☾ + \dot{\Omega}_z + \dot{\Omega}_\odot + \dot{\Omega}_☾ + \dot{\omega}_z + \dot{\omega}_\odot + \dot{\omega}_☾ - \omega_\oplus \tag{81}$$

$$\dot{n}_{i_0} = \delta_{S1} \sin(\lambda_{i_0} - \lambda_{31}) + \delta_{S2} \sin(2\lambda_{i_0} - \lambda_{22}) + \delta_{S3} \sin(3\lambda_{i_0} - \lambda_{33}) \tag{82}$$

$$\ddot{\lambda}_{i_0}/2 = \dot{n}_{i_0}/2 \tag{83}$$

$$\ddot{n}_{i_0}/2 = \left(\dot{\lambda}_{i_0}/2\right) [\delta_{S1} \sin(\lambda_{i_0} - \lambda_{31}) + 2\delta_{S2} \sin(2\lambda_{i_0} - \lambda_{22}) + 3\delta_{S3} \sin(3\lambda_{i_0} - \lambda_{33})] \ , \tag{84}$$

where λ_{22} and λ_{33} are the tesseral harmonics coefficients, and θ_{G0} is GST angle measured at epoch. The GST is obtained evaluating a third degree polynomial at the universal time (UT) of interest [3].

Once all constants are defined, the osculating elements can be obtained from the mean Keplerian elements. The secular variations due to zonal harmonics and lunisolar attraction are given by

$$M_{\text{sec}} = M_0 + n_0\, t + \left(\dot{M}_z + \dot{M}_\odot + \dot{M}_☾\right) t \tag{85}$$

$$\omega_{\text{sec}} = \omega_0 + (\dot{\omega}_z + \dot{\omega}_\odot + \dot{\omega}_☾)\, t \tag{86}$$

$$\Omega_{\text{sec}} = \Omega_0 + \left(\dot{\Omega}_z + \dot{\Omega}_\odot + \dot{\Omega}_☾\right) t \tag{87}$$

$$I_{\text{sec}} = I_0 + \left(\dot{I}_\odot + \dot{I}_☾\right) t \tag{88}$$

$$e_{\text{sec}} = e_0 + (\dot{e}_\odot + \dot{e}_☾)\, t\ . \tag{89}$$

The next step is computing resonance effect of Earth's gravity through numerical integration. The equations to integrate with an Euler-Maclaurin scheme are

$$\lambda_i = \lambda_{i-1} + \dot{\lambda}_i\, \Delta t + \left(\ddot{\lambda}_i/2\right)\ \Delta t \tag{90}$$

$$n_i = n_{i-1} + \dot{n}_i\, \Delta t + (\ddot{n}_i/2)\ \Delta t\ , \tag{91}$$

where n is the mean motion and λ is defined as

$$\lambda = M + \Omega + \omega - \theta_G\ . \tag{92}$$

The time step Δt is 12 h. At the first step the values λ_{i_0} and n_{i_0} and their derivatives computed during initialization are used. At each step the derivatives of λ_i and n_i are updated with the relations

$$\dot{\lambda}_i = n_i + \dot{\lambda}_{i_0} \tag{93}$$

$$\dot{n}_i = \delta_{S1}\sin(\lambda_i - \lambda_{31}) + \delta_{S2}\sin(2\lambda_i - \lambda_{22}) + \delta_{S3}\sin(3\lambda_i - \lambda_{33}) \tag{94}$$

$$\ddot{\lambda}_i = \dot{n}_i/2 \tag{95}$$

$$\ddot{n}_i/2 = \left(\dot{\lambda}_i/2\right)\left[\delta_{S1}\ \sin(\lambda_i - \lambda_{31}) + 2\,\delta_{S2}\ \sin(2\lambda_i - \lambda_{22}) + 3\delta_{S3}\ \sin(3\lambda_i - \lambda_{33})\right]\ . \tag{96}$$

When λ_i and n_i are obtained at the time of interest, the mean motion and mean anomaly are given by

$$n' = n_i \tag{97}$$

$$M_{\text{sec}} = \lambda_i - \Omega_{\text{sec}} - \omega_{\text{sec}} + \theta_G\ , \tag{98}$$

where θ_G is Greenwich hour angle at time t.

The long-period periodic effects of luni-solar perturbation can now be applied, knowing the mean anomaly M_x of the body x at time t. The true anomaly of the perturbing body is approximated by

$$\nu_x = M_x + 2e_x \sin M_x \tag{99}$$

Defining for both the Sun and the Moon

$$F_{2x} = \frac{1}{2}\sin^2(\nu_x) - \frac{1}{4} \qquad\qquad F_{3x} = -\frac{1}{2}\sin(\nu_x)\cos(\nu_x)\ , \tag{100}$$

the long-period variations of the secular elements due to body x written in non-singular variables are

$$\delta e_x = -\left(30\,\beta_0\,C_x\,\frac{e_0}{n_0}\right)\left[F_{2x}\,(X_{2x}X_{3x} + X_{1x}X_{4x}) + F_{3x}\,(X_{2x}X_{4x} - X_{1x}X_{3x})\right] \tag{101}$$

$$\delta I_x = -\frac{C_x}{n_0\,\beta_0}\left[F_{2x}Z_{12x} + F_{3x}\,(Z_{13x} - Z_{11x})\right] \tag{102}$$

$$\delta M_x = -2\,\frac{C_x}{n_0}\left[F_{2x}Z_{2x} + F_{3x}\,(Z_{3x} - Z_{1x}) - 3\,e_x\ \sin(\nu_x)\left(7 + 3\,e_0^2\right)\right] \tag{103}$$

$$(\delta\omega_x + \cos(I_x)\,\delta\Omega_x) = 2\,\beta_0\,\frac{C_x}{n_0}\left[F_{2x}Z_{32x} + F_{3x}\,(Z_{33x} - Z_{31x}) - 9\,e_x\ \sin(\nu_x)\right] \tag{104}$$

$$\sin(I_x)\,\delta\Omega_x = \frac{C_x}{n_0\,\beta_0}\left[F_{2x}Z_{22x} + F_{3x}\,(Z_{23x} - Z_{21x})\right]\ , \tag{105}$$

where the subscripts x on the right side of the equation are referred to the perturbing body. The combined contribution for the two body are applied directly to secular elements when $I' \geqslant 0.2$ rad

$$e' = e_{\text{sec}} + \delta e_{\odot} + \delta e_{☾} \tag{106}$$

$$I' = I_{\text{sec}} + \delta I_{\odot} + \delta I_{☾} \tag{107}$$

$$M' = M_{\text{sec}} + \delta M_{\odot} + \delta M_{☾} \tag{108}$$

$$\Omega' = \Omega_{\text{sec}} + (\delta\Omega_{\odot} + \delta\Omega_{☾}) \tag{109}$$

$$\omega' = \omega_{\text{sec}} + \big(\delta\omega_{\odot} + \cos(I')\delta\Omega_{\odot} + \delta\omega_{☾} + \cos(I')\delta\Omega_{☾}\big) - (\delta\Omega_{\odot} + \delta\Omega_{☾})\cos(I')\ . \tag{110}$$

It is important to underline that the long-period variations are usually non-null at reference epoch, as initial elements are mean elements. To have zero values of long-period perturbation at initial time, and hence initial perturbed values equal to mean Keplerian elements, the initial values of long-period variations can be used as an offset and subtracted to δ quantities. When $I' < 0.2$ rad the perturbations can not be applied directly, as the presence of small divisor leads to singular values. In this case the Lyddane[24] modification is applied to RAAN and argument of pericentre, while the other three elements are computed as above. The following quantities are computed when $I' \geqslant 0$

$$\alpha = \sin(I')\sin(\Omega_{\text{sec}}) + \sin(I')\,(\delta\Omega_{\odot} + \delta\Omega_{☾})\cos(\Omega_{\text{sec}}) + \cos(I')\sin(\Omega_{\text{sec}})\,(\delta I_{\odot} + \delta I_{☾}) \tag{111}$$

$$\beta = \sin(I')\cos(\Omega) - \sin(I')\,(\delta\Omega_{\odot} + \delta\Omega_{☾})\sin(\Omega_{\text{sec}}) + \cos(I')\cos(\Omega_{\text{sec}})\,(\delta I_{\odot} + \delta I_{☾})\ , \tag{112}$$

whereas when $I' < 0$, α and β are

$$\alpha = -\sin(I')\sin(\Omega_{\text{sec}}) + \sin(I')\,(\delta\Omega_{\odot} + \delta\Omega_{☾})\cos(\Omega_{\text{sec}}) + \cos(I')\sin(\Omega_{\text{sec}})\,(\delta I_{\odot} + \delta I_{☾}) \tag{113}$$

$$\beta = -\sin(I')\cos(\Omega) - \sin(I')\,(\delta\Omega_{\odot} + \delta\Omega_{☾})\sin(\Omega_{\text{sec}}) + \cos(I')\cos(\Omega_{\text{sec}})\,(\delta I_{\odot} + \delta I_{☾})\ . \tag{114}$$

Then the mean longitude L is computed

$$\begin{gathered} L' = M' + \omega_{\text{sec}} + \cos(I')\Omega_{\text{sec}} - \Omega_{\text{sec}}\sin(I')\,(\delta I_{\odot} + \delta I_{☾})\ + \\ + \big(\delta\omega_{\odot} + \cos(I')\delta\Omega_{\odot} + \delta\omega_{☾} + \cos(I')\delta\Omega_{☾}\big) \end{gathered} \tag{115}$$

and the primed values of Ω and ω are given by

$$\Omega' = \arctan_2\left(\frac{\alpha}{\beta}\right) \qquad \omega' = L' - M' - \cos(I')\Omega' \, . \tag{116}$$

These passages are necessary because at zero inclination and eccentricity the argument of pericentre and right ascension of the ascending node are not defined uniquely.

Subsequently, the long-period periodic effects of Earth's gravity are added, using the quantities

$$a_{xN} = e' \cos(\omega') \tag{117}$$

$$a_{yN} = e' \sin(\omega') + \frac{1}{4}\frac{A_{3,0} \sin(I')}{k_2\, a'\, \beta'^2} \tag{118}$$

$$u = \omega' + \nu' + \frac{1}{8}\frac{A_{3,0} \sin(I')}{k_2\, a'\, \beta'^2}\left(e' \cos(\omega')\right)\left(\frac{3 + 5\cos(I')}{1 + \cos(I')}\right) \tag{119}$$

$$r = \frac{a'\left(1 - e'^2\right)}{1 + e' \cos \nu}\left[1 - \frac{3}{2}k_2\frac{\sqrt{1 - e'^2}}{p_L^2}\left(3\cos^2(I') - 1\right)\right], \tag{120}$$

where a' is the semi-major axis obtained with mean motion n'; $A_{3,0}$, e, and p_L are defined as

$$A_{3,0} = -J_3\, R_\oplus^3 \tag{121}$$

$$e = \sqrt{a_{xN}^2 + a_{yN}^2} \tag{122}$$

$$p_L = a'\left(1 - e'^2\right). \tag{123}$$

The position in Earth centred inertial reference frame is given by

$$\mathbf{r} = r\,\hat{\mathbf{u}} \, , \tag{124}$$

where

$$\hat{\mathbf{u}} = \begin{Bmatrix} -\sin(\Omega)\cos(I)\sin(u) + \cos(\Omega)\cos(u) \\ \cos(\Omega)\cos(I)\sin(u) + \sin(\Omega)\cos(u) \\ \sin(I)\sin(u) \end{Bmatrix} . \tag{125}$$

4.2 Objective Function

Based on the analytical approaches described above, the computation of the MOID for two Earth-orbiting objects in a perturbed dynamical framework goes through the following steps:

1. The perturbed values of the orbital elements are evaluated at the desired time t, taking into account only long-period perturbations. Neglecting short-period oscillation terms introduces an error of a few kilometers on the object position [29]. Thus, if these oscillations were considered, it would be more appropriate to study the synchronization problem rather than identifying the MOID.
2. The position vectors of the two objects and the square distance d^2 are computed using Eq. (2) and (3) respectively.
3. The MOID is obtained by running the global optimizer COSY-GO to minimize d^2. The search space is $[-180; +180]$ deg for the true anomalies and $[0, n_d]$ days for time (n_d stands for the number of days), measured from the epoch at which the initial orbital elements are given.

5 MOID of Perturbed Orbits: Numerical Experiments

This section is devoted to assess the performances of the procedure proposed in Sect. 4 on a set of test cases. If not otherwise mentioned, the minimum box size of the optimizer is 1E-2 on all variables. The expansion order is set to 6, as higher order coefficients of the Taylor expansion become too small and their contribution is moved to the remainder interval. A time window of 1 year is considered to allow for a large evolution of the orbital parameters.

In Table 2 the orbits of the two objects analyzed in the first test case are defined. The first orbit (e.g. a debris) is sun-synchronous and the perturbation of the first four zonal harmonics is modeled through Aksnes' solution. The second object is supposed to move on a Keplerian orbit by adopting a proper station keeping strategy (e.g. an operative satellite). It is worth noting that, if both orbits were considered as Keplerian, the MOID would be 1880.083 km. According to this value, the possibility of an impact between the two objects would be ruled out. However, due to zonal harmonics perturbation, the orbital plane of the sun-synchronous orbit rotates around an axis which is perpendicular to Earth's equatorial plane and, as a result, the actual MOID can be lower.

Table 2 Test case #1: orbits definition

			a	e	I	Ω	ω
Orbit #	Orbit type	Dynamical model	[km]		[°]	[°]	[°]
1	Sun-syncr.	Aksnes	6878.136	0.0	97.0	110.0	70.0
2	MEO	Kepler	11,130.227	0.4	6.5	300.0	73.0

Table 3 Test case #1: enclosure of objective function minimum

Test case #	d^2 [km^2]
1	[−0.22250739E−307, 0.68795338E−014]

Table 4 Test case #1: enclosure of the four intersections

ν_1 [°]	ν_2 [°]	t [days]
[155.990229, 155.990231]	[26.0738534, 26.0738536]	[119.068571, 119.068573]
[118.290326, 118.290328]	[−26.0738536, −26.0738534]	[260.826736, 260.826738]
[−39.7805915, −39.7805913]	[26.0738534, 26.0738536]	[318.632662, 318.632664]
[−40.9823329, −40.9823327]	[−26.0738536, −26.0738534]	[61.7190874, 61.7190876]

COSY-GO is used to identify the MOID in the perturbed motion. The interval enclosure of the minimum of the objective function is reported in Table 3. The two bounds of the enclosure are small numbers and the lower bound is negative. Thus, the null square distance belongs to this interval and intersections between these orbits could occur in the considered time span. The minimum of the objective function occurs at four different orbital configurations. Each of them is defined by a set of values of ν_1, ν_2, and t, whose interval enclosures are reported in Table 4.

The configuration of the two orbits at the four intersections are illustrated in Fig. 2. The contour plot of the objective function along ν_1-ν_2 sections at the four intersections are presented in Fig. 3. The algorithm can compute the minimum distance between two orbits, one of which is perturbed by Earth's zonal harmonics. The presented example also shows that the MOID computed with the approximation of Keplerian orbits is not always sufficient to exclude the occurrence of intersections.

The second and third test cases are aimed at showing the effect of atmospheric drag. They involve two perturbed orbits, whose parameters are listed in Table 5. The orbit #1 is similar to a Molnyia orbit, although having lower semi-major axis and eccentricity. The two optimizations are run on the time span of 1 year. In the second test case both orbits are modeled by Aksnes' solution, i.e. considering the effect of zonal harmonics from second to fourth, only. In the third test case, the orbit #1 is modeled by means of HANDE formulation, thus including atmospheric drag. The ballistic coefficient B for the object on orbit #1 is reported in Table 5 as well. The enclosures of the global minimum obtained for the two cases are listed in Table 6. It can be observed that there is no intersection for test case #2 and the MOID is 17.15 km, whereas an intersection is found in the other test case. The reason is that the apogee height of the first orbit decreases due to the atmospheric drag, which is modelled in HANDE algorithm. The enclosure of the three variables is listed in Table 7. The two orbits at their minimum distance are illustrated in Fig. 4. The contour plots of the search domain at the times of minimum distance are represented in Fig. 5 for the two test cases.

The fourth test case considers two geostationary orbits, whose Keplerian elements are listed in Table 8. Orbit #2 represents a controlled satellite, whereas a space debris moves on orbit 1, and SGP4 analytical solution is used. Two intersections

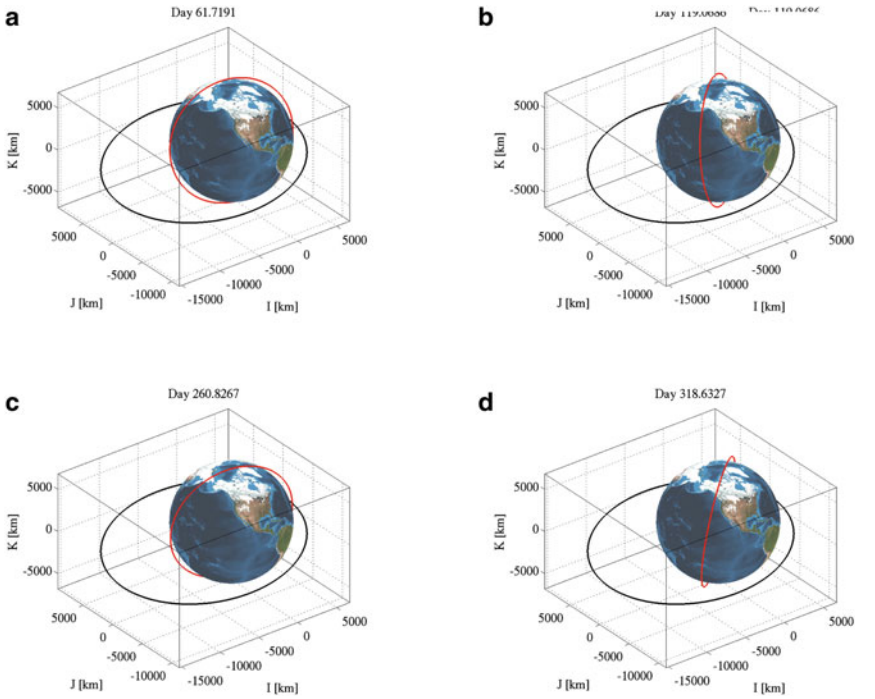


Fig. 2 Representation of orbits at intersections: (**a**) first intersection, (**b**) second intersection, (**c**) third intersection, (**d**) fourth intersection

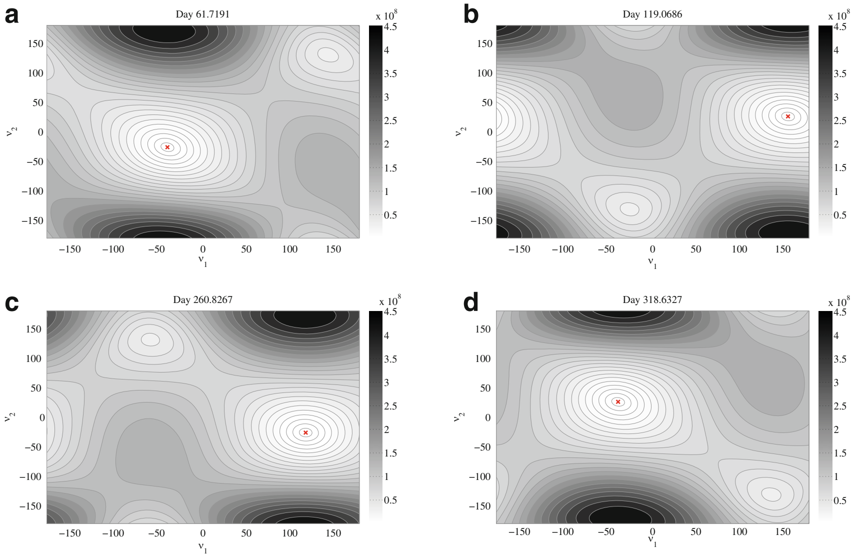


Fig. 3 Test case #1: MOID computation. Objective function contour plots: (**a**) first intersection, (**b**) second intersection, (**c**) third intersection, (**d**) fourth intersection

Table 5 Test cases #2 and #3: orbits definition

			a	e	I	Ω	ω	B
Orbit #	Orbit type	Dynamical model	[km]		[°]	[°]	[°]	[m^2/kg]
1	Molnyia-like	Aksnes/HANDE	9825.909	0.3	63.43	276.6	168.7	0.04
2	MEO	Aksnes	12,559.681	0.0	10.0	0.0	0.0	–

Table 6 Test cases #2 and #3: enclosure of objective function minimum

Test case #	d^2 [km^2]
2	[294.108777, 294.111215]
3	[−0.22250739E−307, 0.25157065E−018]

Table 7 Test cases #2 and #3: enclosure of stationary points

Test case #	ν_1 [°]	ν_2 [°]	t [days]
2	[−164.464846, −164.460773]	[−104.832006, −104.824949]	[359.995569, 360.000001]
3	[−163.805136, −163.805134]	[−52.4528862, −52.4528823]	[336.907751, 336.907755]

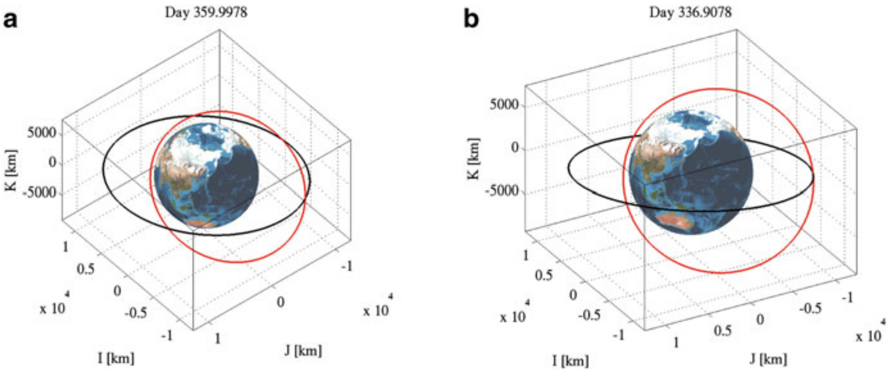


Fig. 4 Representation of orbits at their minimum distance: (**a**) test case #2, (**b**) test case #3

are found in the time span of 1 year. More specifically, 41 boxes smaller than the minimum allowed size remain at the end of the optimization process for the first intersection and 56 boxes for the second. The boxes are grouped together to obtain the enclosures reported in Table 9, since intervals neighbor each other in the search domain. The contour plot of the objective function can be observed in Fig. 6. The relative geometry of the two orbits is such that, at each time, a line of objective function values close to the MOID is identified on the search space. Thus, the solution of the problem requires many objective function evaluations. In Fig. 6b and d the search domain is sectioned along planes parallel to domain boundaries and whose intersection identifies the position of the MOID.

The last test case involves two perturbed orbits that do not intersect in the considered time span. The first orbit has low perigee and thus atmospheric drag is accounted for. The second orbit is a MEO and it is propagated by means of

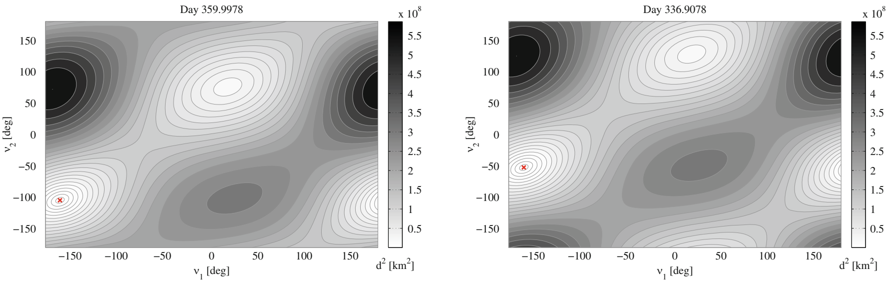


Fig. 5 MOID computation. Objective function contour plot at intersection time: (**a**) test case #2, (**b**) test case #3

Table 8 Test case #4: orbits definition

Orbit #	Orbit type	Dynamical model	a [km]	e	I [°]	Ω [°]	ω [°]
1	GEO	SGP4	42,164.136	0.04	−0.86	0.00	0.00
2	GEO	Kepler	42,164.136	0.00	0.00	0.00	0.00

Table 9 Test case #4: enclosure of intersections

ν_1 [°]	ν_2 [°]	t [days]
[92.296701, 92.563286]	[7.204308, 6.9307441]	[243.935364, 243.944814]
[−92.438383, −92.197266]	[−177.802735, −177.561034]	[243.932431, 243.943692]

Table 10 Test case #5: orbits definition

Orbit #	Orbit type	Dynamical model	a [km]	e	I [°]	Ω [°]	ω [°]	B [m²/kg]
1	Molnyia-like	HANDE	9825.909	0.3	63.4	276.6	171.5	0.04
2	MEO	Aksnes	11,278.136	0.0	25.0	110.0	200.0	–

Table 11 Test case #5: enclosure of objective function minimum

Test case #	d^2 $\left[\mathrm{km}^2\right]$
5	[1566860.77, 1566861.40]

Aksnes' solution. The initial osculating Keplerian elements are listed in Table 10. The enclosure of the minimum of the objective function computed by COSY-GO is listed in Table 11. In this case the square distance is comprised between two large positive numbers and thus no intersection occurs. Three contiguous boxes remain at the end of the optimization run, whose size is lower than the minimum box size. The remaining boxes are grouped together to obtain the enclosures of ν_1, ν_2, and t corresponding to the global minimum shown in Table 12.

The computational time associated to each test case is reported in Table 13. The table refers to the computational time obtained by running the code on an Intel

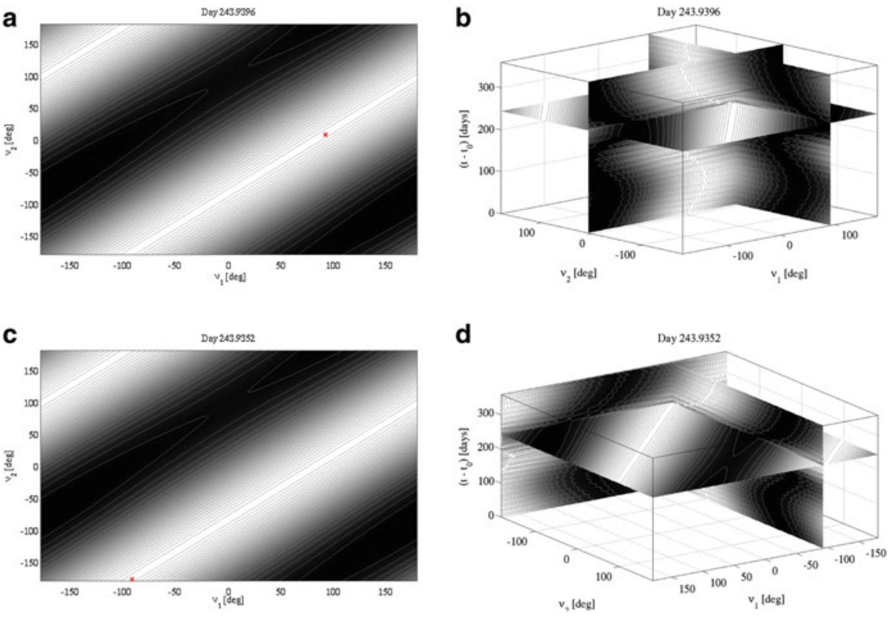


Fig. 6 MOID computation for test case #4: (**a**) objective function contour plot at the first intersection, (**b**) search domain dissection at the first intersection, (**c**) objective function contour plot at second intersection, (**d**) search domain dissection at the second intersection

Table 12 Test case #5: enclosure of intersections

ν_1 [°]	ν_2 [°]	t [days]
[−162.265526, −162.258591]	[−31.9152062, −31.9107101]	[−0.2225074E−307, 0.5562744E−003]

Table 13 CPU time in seconds required for the computation of the MOID for the different test cases

Test case #:	1	2	3	4	5
Computational time [s]:	116.74	63.27	57.12	4466.98	216.07

Pentium M 1.73 GHz with 1 GB RAM, Sabayon Linux 5.3. It can be observed that the computational time increases if

- many orbital intersections occur (e.g., test case #1);
- the objective function is flat near the MOID in large portion of the search space (e.g., test cases #4 and 5).

In the latter case the computational cost is magnified by setting a small value for the minimum box size dimension. This is evident in case #4 and it is due to the larger number of boxes that must be processed until the end of the optimization. In addition, the computational time increases with analytical model complexity, since the number of terms to be evaluated is higher.

Table 14 Test case #3: computational time for different expansion orders

Cut-off [km^2]	Box size	Expansion order	Elapsed time [s]
–	0.1	6	57.60
100	0.1	6	49.58
–	0.1	4	31.74
100	0.1	4	27.38
–	0.1	2	31.39
100	0.1	2	26.79

Table 15 Test case #5: computational time for different box sizes and expansion orders

Cut-off [km^2]	Box size	Expansion order	Elapsed time [s]
–	0.01	6	216.07
100	0.01	6	15.88
100	0.1	6	15.55
100	0.1	4	10.74
100	0.1	2	8.02

The test cases presented above shows that the developed approach can identify rigorous and sharp enclosures of orbit intersections. The algorithm can be effectively used to assess which pairs of orbits can potentially intersect. Nevertheless, the computational time of a single optimization run turns out to be relevant, since hundreds of seconds are required.[3] However, the computational time can be significantly reduced by setting larger values of the minimum box size, thus avoiding many unnecessary box splitting. Furthermore, a suitable cut-off value can be set at the beginning of the optimization to obtain an efficient pruning of the regions of the search space where the orbits are far apart.

In Table 14, the computational times for test case #3 are reported. The minimum box size is kept constant, whereas the Taylor model expansion order is changed. The computational time required by the optimization process decreases with the expansion order. In addition, if a cut-off value is introduced, a further reduction of computational time is achieved. The effect of cut-off value and box size is shown in Table 15. The minimum d^2 in this case is large and the introduction of a cut-off value produces a drastic reduction in the computational time.

[3] The computational time can indeed be reduced almost linearly performing parallel computation on many processors as COSY-GO has a fully parallel implementation.

6 Final Remarks

This chapter described a method for the computation of the global minimum of the distance function between two perturbed orbits based on Taylor models. The orbital evolution of the orbiting objects has been computed through analytical solutions, thus accounting for zonal harmonics, atmospheric drag, and luni-solar long-period perturbations.

A global optimization problem has been formulated and solved rigorously by means of the global optimizer COSY-GO to obtain a validated enclosure of the solutions. Five sets of orbital parameters have been used as test cases. It has been shown that potential collisions could occur between Sun-synchronous and other LEO orbits due to orbital plane rotation caused by J_2 perturbations. In addition, test cases #2 and #3 demonstrated the capability of the method to catch semi-major axis reduction. The method has also been applied to a GEO case using the SGP4 analytical solution. The case of no intersection has been considered with test case #5. The numerical experiments performed have shown that orbital conjunctions can occur also for orbits with large initial MOID. In these cases a two-body approximation would miss the occurrence of threatening conditions.

An analysis of the effects of expansion order, cut-off value, and minimum box size on computational time has also been performed. A considerable reduction of the computational effort can be mainly achieved by lowering the expansion order and setting proper cut-off values.

When orbits have similar orbital elements, as in test case #4, many similar MOID can be found at each time. It is thus convenient to approach the risk assessment by computing the distance between the trajectories, which requires the validated solution of Kepler's equation. In this case an optimization problem in a single variable, i.e. time, can be formulated.

References

1. Aksnes, K.: On the use of Hill variables in artificial satellite theory: Brouwer's theory. Astron. Astrophys. **17**, 70–75 (1972)
2. Armellin, R., Di Lizia, P., Berz, M., Makino, K.: Computing the critical points of the distance function between two Keplerian orbits via rigorous global optimization. Celestial Mech. Dyn. Astron. **107**, 377–395 (2010)
3. Astronomical Almanac for the Year 2007, United States Government Printing Office (2006)
4. Baluyev, R.V., Kholshevnikov, K.V.: Distance between two arbitrary unperturbed orbits. Celestial Mech. Dyn. Astron. **91**, 287–300 (2005)
5. Berz, M.: Modern Map Methods in Particle Beam Physics. Academic Press, San Diego (1999)
6. Berz, M., Makino, K.: COSY INFINITY Version 9 reference manual. MSU Report MSUHEP-060803, Michigan State University, East Lansing, MI 48824, pp. 1–84 (2006)
7. Berz, M., Bischof, C., Corliss, G., Griewank, A.: Computational Differentiation: Techniques, Applications, and Tools, pp. 1–419. SIAM, Philadelphia (1996)
8. Berz, M., Makino, K., Kim, Y.: Long-term stability of the tevatron by verified global optimisation. Nucl. Instrum. Methods **A558**, 1–10 (2005)

9. Dybczynski, P.A., Jopek, T.J., Serafin, R.A.: On the minimum distance between two Keplerian orbits with a common focus. Celestial Mech. Dyn. Astron. **38**, 345–356 (1986)
10. Explanatory Supplement to the Astronomical Ephemeris and the American Ephemeris and Nautical Almanac (1961)
11. Fornberg, B.: Generation of finite difference formulas on arbitrarily spaced grids. Math. Comput. **51**, 699–706 (1988)
12. Griewank, A., Corliss, G.F.: Automatic Differentiation of Algorithms, pp. 25–31. SIAM, Philadelphia (1991)
13. Gronchi, G.F.: On the stationary points of the squared distance between two ellipses with a common focus. SIAM J. Sci. Comput. **24**, 61–80 (2002)
14. Gronchi, G.F.: An algebraic method to compute the critical points of the distance function between two Keplerian orbits. Celestial Mech. Dyn. Astron. **93**, 295–329 (2005)
15. Hoot, F.R.: Reformulation of the Brower geopotential theory for improved computational efficiency. Celestial Mech. **24**, 367–375 (1981)
16. Hoots, F.R.: An analytical satellite theory using gravity and a dynamic atmosphere. In: AIAA/AAS Astrodynamics Conference (1982)
17. Hoots, F.R., Roehrich, R.L.: Models for Propagation of the NORAD Elements Sets, Project SPACETRACK, Rept. 3 (1980)
18. Hoots, F.R., France, R.G.: An analytical satellite theory using gravity and a dynamic atmosphere. Celestial Mech. **40**, 1–18 (1987)
19. Hoots F.R., Schumacher, P.W., Glover, R.A.: History of analytical orbit modeling in the U.S. space surveillance system. J. Guid. Control Dyn. **27**, 174–185 (2004)
20. Hujsak, R.S.: A restricted four body solution for resonating satellites with an oblate earth. In: American Institute of Aeronautics and Astronautics Conference (1979)
21. Kearfott, R.B.: Rigorous Global Search: Continuous Problems, pp. 169–199. Kluwer Academic Publishers, Dordrecht (1996)
22. Kholshevnikov, K.V., Vassiliev, N.N.: On the distance function between two Keplerian elliptic orbits. Celestial Mech. Dyn. Astron. **75**, 75–83 (1999)
23. Lowan, A.N., Davis, N., Levenson, A.: Table of the zeros of the Legendre Polynomials of order 1–16 and the weight coefficients for Gauss' mechanical quadrature formula. Bull. Am. Math. Soc. **48**, 739–743 (1942)
24. Lyddane, R.H.: Small eccentricities or inclinations in the Brower theory of the artificial satellite. Astron. J. **68**, 555–558 (1963)
25. Makino, K.: Rigorous Analysis of Nonlinear Motion in Particle Accelerators. Ph.D. Thesis, Michigan State University, East Lansing, MI, pp. 76–136 (1998)
26. Makino, K., Berz, M.: Efficient control of the dependency problem based on Taylor model methods. Reliab. Comput. **5**, 3–12 (1999)
27. Makino, K., Berz, M.: Taylor models and other validated functional inclusion methods. Int. J. Pure Appl. Math. **6**, 239–316 (2003)
28. Makino, K., Berz, M.: Verified global optimization with Taylor model-based range bounders. Trans. Comput. **11**, 1611–1618 (2005)
29. Morselli A.: The space debris problem: collision risk assessment for perturbed orbits via rigorous global optimization, M.Sc. Thesis, Politecnico di Milano, Milan (2011)
30. Sitarski, G.: Approaches of the parabolic comets to the outer planets. Acta Astronaut. **18**, 171–19 (1968)
31. Vallado, D.A., Crawford, P., Hujsak, R., Kelso, T.S.: Revisiting SPACETRACK Report # 3: Rev. 1, AIAA/AAS Astrodynamics Specialist Conference and Exhibit (2006)

Optimal Robust Design of Hybrid Rocket Engines

Dario Pastrone and Lorenzo Casalino

Abstract Hybrid rocket engines are flexible, safe, reliable, and low-cost and can be used in many aerospace applications. The engine design and operation are contingent on the type of designated mission and engine design is strictly related to trajectory optimization. In real-world applications, uncertainties affect propulsion system performance: mission goals and constraints may be not fulfilled by an engine designed using a deterministic approach. Uncertainty of the coefficient and mass flux exponent in the classical regression rate correlation are here taken into account, as they are the ones that more remarkably deviate delivered propulsion system performance from expected nominal values. The upper-stage of a small launcher is considered. The engine has a partially regulated pressure-fed system. First, the deterministic optimal design is obtained by means of a nested direct–indirect optimization procedure, and launcher performance are evaluated considering non-nominal regression rate correlations. The height of the attainable orbit results to be strongly jeopardized when the regression rate is larger than that of the nominal case (large oxidizer residual). In contrast, when regression rate is smaller than nominal, residual propellant consists of fuel and a less severe performance degradation occurs. Some improvements in the off-design behavior can be obtained if the engine design is optimized for values of the regression rate correlation coefficient that are larger than the nominal ones. Results shows that robustness of deterministic solutions is not adequate (e.g., insertion within 100 km from the desired orbit altitude). An evolutionary optimization code is then used to define the optimal robust design. The fitness of each individual of the population is evaluated as a linear combination of payload and an index that quantifies the effective reaching of the target orbit under uncertainty (based either on the worst case scenario or on the average performance). Results show that close matching of the required performance (e.g., within 10 km from the desired orbit altitude) can be obtained with a moderate (below 5 %) penalty on the payload.

Keywords Hybrid rocket engines • Multidisciplinary design optimization • Robust optimization

D. Pastrone (✉) • L. Casalino
Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129 Torino, Italy
e-mail: dario.pastrone@polito.it; lorenzo.casalino@polito.it

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization
and Its Applications 114, DOI 10.1007/978-3-319-41508-6_10

1 Introduction

Hybrid rocket engines (HRE) are promising propulsion systems. Their performance is similar to that of storable or semi-cryo liquid rocket engines and they share appealing features of both solid rocket motors and liquid rocket engines. Moreover, they are cheaper and safer than liquid and solid rockets, and, in many cases, they are more environmentally friendly than storable liquid and solid rockets. Due to these reasons, many research programs focus on the development of HREs. Different applications are being investigated, including microgravity platforms, upper stage for small launchers, debris removal and commercial space flights.

Budget and time required for the development of HREs can be reduced if a proper conceptual design is carried out. Multidisciplinary design optimization can be performed to improve performance. The propulsion needs are strictly related to the mission to be carried out and a coupled optimization of the propulsion system and trajectory is required. This is especially true for HREs, which are one-lever control engines and usually exhibit varying thrust profile and mixture ratio shifting. Of course, the optimization process is simplified if the design variables and the model parameters are assumed to have deterministic values. On the other hand uncertainties, inherently present in real life, may cause severe deviation of the performance from nominal values, so that vehicle performance are jeopardized or even mission failure may occur. Therefore, it is important to take uncertainties into account from the beginning of the design phase [20, 28]. The goal is to improve the robustness of the design, i.e. to minimize the effects that uncertainty or variation of design parameters may have on system performance, without eliminating the causes of uncertainty or variation [3, 21, 24, 26].

Previous studies highlighted that a hybrid rocket upper stage is a viable option for small/low-cost launchers [17]. The authors carried out deterministic multidisciplinary optimizations and demonstrated that the use of HREs can provide a very good margin of payload improvement [8, 9, 12, 13]. The present work concerns the robust optimal design of a hybrid rocket upper stage for a small launcher, focusing on the effects of the uncertainty of the parameters of the regression rate correlation. In literature, few works have concerned robust design of HREs, and pointed out that regression rate uncertainty is the most important factor that influences the propulsion system performance [29, 30]. In fact, the fuel regression rate highly conditions the design and the performance of HREs [22] and the parameters that appear in the classical correlations that describe regression rate behavior may be affected by relevant uncertainty. As a result, the deterministic optimum solution may not be robust enough to guarantee a reliable insertion of the given payload to the prescribed orbit [10].

In a previous work [9], the authors presented an approach to the deterministic optimization of hybrid rocket design and trajectory, with several example of suitable applications. In this chapter, a robust optimization procedure is presented. Reference is made to the Vega launcher [16], which has three solid-propellant stages and a fourth liquid propellant stage. The fourth stage is first ignited to complete the boost phase and then performs a second burn for the injection into the final orbit. A HRE is considered to replace the third and fourth stages. It is designed to perform the injection into the final orbit by means of two burns. Hydrogen-peroxide (HP)/ polyethylene (PE) is the propellant combination. A partially regulated gas-pressure feed system is adopted because of its simplicity and lower cost compared to pump-fed engines. First, a sensitivity analysis is performed. This analysis is pursued via a multidisciplinary optimization approach, which couples the optimization of propulsion system and 3D trajectory. The set of design parameters is optimized by means of a direct method (mathematical programming), in conjunction with an indirect procedure to optimize the trajectory. The performance index to be maximized is the payload inserted into a reference orbit. The optimal values for the propulsion system parameters are sought, while holding the lift-off mass and the performance of the lower stages of the launcher unchanged (i.e., mass, altitude and velocity at the upper stage ignition are assigned). The effects of the regression rate on launcher performance, engine design parameters and engine operation are investigated. Then the off-design performance are evaluated for different engine designs as a function of regression rate parameter values, in order to find a design which is more insensitive to regression rate uncertainty. Finally, an evolutionary optimization code is used to define the optimal robust design. The fitness of each individual of the population is evaluated as a linear combination of payload and an index that quantifies the effective reaching of the target orbit under uncertainty (based either on the worst case scenario or on the average performance).

2 Grain Geometry and Ballistic Model

In HREs, oxidizer and fuel are separated and stored in two different physical phases. A HP/PE combination is here considered. The solid fuel PE is stored as a cylindrical grain in the combustion chamber and the liquid oxidized HP is stored in a tank and injected into the combustion chamber. The fuel grain presents one or more perforations, called ports, through which the oxidizer flows. Combustion takes place through diffusive mixing of oxidizer and fuel coming from the solid grain surface. The fuel mass flow results to be proportional to the regression rate $\dot{y}$ and the perimeter P of the holes. Due to the low fuel regression rate and the high trust level required, a multi-port grain geometry is here adopted in order to avoid grains with unacceptable length. The regression rate is assumed to be uniform along the port axis.

The geometry of the circular-section grain [2]. described in Fig. 1, is defined by the number of ports N, the web thickness w and the grain outer radius R_g. The parameters x, h, and β are introduced

$$x = \pi/N \ \sin^{-1}[w/(R_g - w)] \tag{1}$$

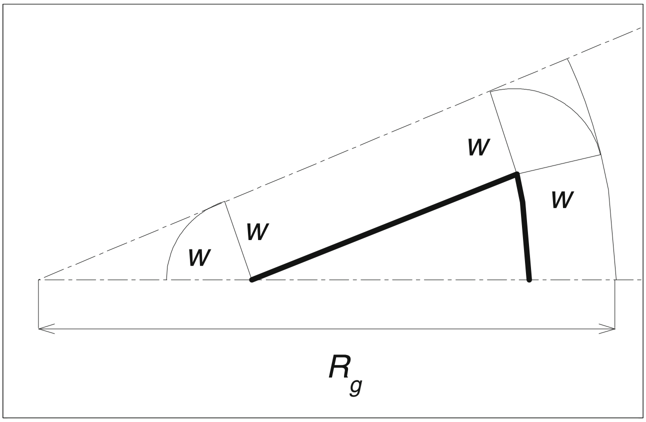


Fig. 1 Grain geometry

$$h = \sqrt{(R_g - w)^2 - w^2} - w \tan(\pi/2 - \pi/N) \tag{2}$$

$$\beta = \pi/2 + x\pi/N \tag{3}$$

to evaluate the grain geometry. The initial (subscript i) port area A_p is

$$(A_p)_i = 2N\left[(R_g - w)^2(1 - x)\pi/(2N) - hw/2)\right] \tag{4}$$

For a given burning distance y ($0 \leq y \leq w$), one easily computes the burning perimeter P

$$P = 2N\left[(R_g - w + y)(1 - x)\pi/N + \beta y + h + (\pi/2 - \pi/N)y\right] \tag{5}$$

and the port area

$$\begin{aligned} A_p = (A_p)_i + 2N\big\{&\left[(R_g - w + y)^2 - (R_g - w)^2\right] \\ &(1 - x)\pi/(2N) + \beta y^2/2 + hy + (\pi/2 - \pi/N)y^2/2\big\} \end{aligned} \tag{6}$$

No pyrolysis of the lateral ends is considered. Pressure losses inside the combustion chamber are taken into account by relating the chamber head-end pressure p_1 to the chamber nozzle-stagnation pressure p_c. An approximate relation, similar to that proposed by Barrere et al. [1] for side-burning grains, is used

$$p_1 = \left[1 + 0.2\left(\frac{A_{th}}{A_p}\right)^2\right] p_c \tag{7}$$

where A_{th} is the throat area.

The regression rate is determined by the oxidizer mass flow rate $\dot{m}_O$ and grain geometry

$$\dot{y} = a\,(\dot{m}_O/A_p)^n \tag{8}$$

with nominal values for the regression rate correlation [18, 27] $a = 7 \cdot 10^{-6}$ and $n = 0.8$, when SI units are used. The hydraulic resistance Z in the oxidizer flow path from the tank to the combustion chamber determines the oxidizer flow rate. Under the assumption of incompressible turbulent flow

$$\dot{m}_O = \sqrt{(p_t - p_1)/Z} \tag{9}$$

where p_t is the oxidizer tank pressure. The value of Z is assumed to be constant during engine operation. The fuel mass flow $\dot{m}_F$ is obtained as

$$\dot{m}_F = \rho_F \dot{y} A_b = \rho_F \dot{y} L_b P \tag{10}$$

where ρ_F is the fuel grain density, A_b is the burning area, and L_b is the cylindrical grain length. The mixture ratio α is

$$\alpha = \frac{\dot{m}_O}{\dot{m}_F} \propto \dot{m}_O^{1-n} A_p^n / A_b \tag{11}$$

An isentropic expansion in the nozzle is assumed, and the chamber nozzle-stagnation pressure p_c is determined by

$$p_c = \frac{(\dot{m}_O + \dot{m}_F)c^*}{A_{th}} \tag{12}$$

The performance of the propellant combination is evaluated [19] as a function of the mixture ratio α, assuming $p_c = 10\,\text{bar}$. Even though the actual pressure in the combustion chamber can span over a wide range during engine operations, the error is small for chamber pressures and mixture ratios considered in this chapter. Frozen equilibrium expansion is assumed; the exhaust gas maintains throughout the nozzle the composition that it has in the combustion chamber. This conservative assumption of frozen equilibrium expansion is adopted to account for the low combustion efficiency of HREs; in addition a 0.96 c^*-efficiency [25] is introduced. Third-degree polynomial curves fitting the characteristic velocity and specific heat ratio are embedded in the code to compute the proper values as the mixture ratio changes during engine operations.

3 Deterministic Design and Optimization

A suitable set of variables must be used to define the engine geometry and the propulsion system. In an initial design phase, variables that have a clear physical meaning and can be easily estimated are preferred. According to the chosen ballistic model, the design of the HRE is defined by the initial thrust level F_i, the initial mixture ratio α_i, the nozzle expansion ratio E, the initial value of tank pressure $(p_t)_i$, the initial value of chamber pressure $(p_c)_i$, and the ratio J of the throat area to the initial port area. Some of these variables are however here constrained. The initial chamber pressure is assigned by imposing $(p_c)_i = 0.4\,(p_t)_i$; actually, the ratio p_t/p_c varies during operation, but the assumed initial ratio is usually sufficient to guarantee $p_t/p_c > 1.5$ and to avoid coupling between the hybrid engine and the oxidizer feed system. The initial port area to throat area ratio J should be as large as possible but not exceed 0.5 to avoid excessive pressure losses and nonuniform grain regression: $J = 0.5$ is therefore assumed throughout. The initial tank pressure would assume rather low values if left unconstrained, so its value is here fixed at 25 bar.

A partially regulated feed system is considered: A phase with constant tank pressure, maintained by means of helium flowing from an auxiliary tank, is introduced, followed by a blowdown phase. The initial ullage volume is assumed to be 3 % of the oxidizer volume, in order to have a stable regulator response when the out flow starts [4]. In this case two additional parameters are the auxiliary gas tank volume V_a and the initial pressurizing gas pressure p_a; the latter is fixed at $p_a = 200$ bar, even though improved performance could be obtained by increasing the gas tank pressure. The parameter V_a is conveniently replaced by the exhausted oxidizer mass at the beginning of the blowdown phase $(m_O)_{BD}$. When the tank pressure is kept constant $p_t = (p_t)_i$, whereas p_t is calculated assuming an isentropic expansion of the pressurizing gas in the tank during the subsequent blowdown phase. By indicating with subscript BD the values at the beginning of the blowdown phase, one has

$$p_t = (p_t)_i \left[\frac{(V_g)_{BD}}{V_g} \right]^{\gamma_g} \tag{13}$$

where the gas volume in the tank $V_g = (V_g)_i + m_O/\rho_O$ depends on the oxidizer mass that has been exhausted (m_O), $(V_g)_{BD} = (V_g)_i + (m_O)_{BD}/\rho_O$, and γ_g is the specific heat ratio of the pressurizing gas. The design parameters are optimized by means of a direct method, described in the following.

Given the set of design parameters, the engine geometry is initially determined and the trajectory is then optimized to determine the payload. From α_i, the relevant properties of the combustion gases can be computed and the thrust coefficient C_F can then be evaluated by assuming an isentropic one-dimensional expansion to the exit conditions (subscript e), with constant specific heat ratio γ (a 0.98 correction factor introduced to modify the vacuum thrust coefficient)

$$C_F = 0.98\left\{\sqrt{\frac{2\gamma^2}{\gamma-1}\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{\gamma-1}}\left[1-\left(\frac{p_e}{p_c}\right)^{\frac{\gamma}{\gamma-1}}\right]}+E\frac{p_e}{p_c}\right\}-E\frac{p_0}{p_c} \tag{14}$$

where the term related to the atmospheric pressure p_0 is always small, as the third stage always flies at high altitude. From F_i, the mass flow rates at rocket ignition (i.e., at $t = 0$) are found

$$(\dot{m}_p)_i = (1+\alpha_i)(\dot{m}_F)_i = \frac{1+\alpha_i}{\alpha_i}(\dot{m}_O)_i = \frac{F_i}{c_i^*(C_F)_i} \tag{15}$$

and throat and initial port areas A_{th} and $(A_p)_i$ are then determined

$$A_{th} = \frac{(\dot{m}_p)_i}{(p_c)_i c_i^*} \quad ; \qquad (A_p)_i = \frac{A_{th}}{J} \tag{16}$$

The nozzle throat area A_{th} is considered to be constant during operation. One also finds

$$(A_b)_i = \frac{(A_p)_i^n}{a\rho_F}\,\frac{(\dot{m}_F)_i}{(\dot{m}_O)_i^n} \tag{17}$$

The grain geometry can then be derived by means of an iterative procedure. A tentative value is assumed for R_g and Eqs. (1)–(4) are numerically solved for x, β, h, and w given the required initial port area. Equation (5) at ignition ($y = 0$) gives the initial perimeter P_i to compute the grain length $L_b = (A_b)_i/P_i$. Equation (8) is integrated up to burnout during the optimization of the ascent trajectory. The optimization procedure corrects the tentative value for R_g to match the necessary condition $y_f = w$ at burnout.

The head-end pressure is computed with Eq. (7) and, knowing the initial tank pressure, also the hydraulic resistance Z can be determined by applying Eq. (9) at $t = 0$. The engine geometry is completely defined and, once the initial ullage volume in the propellant tank has been assumed and the pressurization system has been specified, the engine performance can be evaluated during operation.

The tank pressure is either $p_t = (p_t)_i$ or it is provided by Eq. (13). Numerical integration of Eqs. (8)–(10), allows for the evaluation of the fuel grain geometry, the exhausted masses of oxidizer and fuel, and their mass flow rates. At each instant t, once the tank pressure p_t and the engine geometry are known, the regression rate, the propellant flow rates (and their ratio α), c^*, p_c and p_1 are computed by numerically solving Eqs. (8)–(12) while the curve fit for c^* as a function of α is used. Then, the thrust level $F = p_c A_{th} C_F$ is determined by evaluating C_F at the actual altitude via Eq. (14), in order to integrate the trajectory equations. At burnout the overall propellant is finally evaluated, and an estimation of the structural masses can be obtained. The oxidizer volume is required to determine the initial ullage volume. A tentative value is assumed for the overall oxidizer mass and is corrected by the indirect method during the trajectory optimization, in order to match the value obtained by integrating Eq. (9) up to the orbit insertion.

The optimization procedure aims at finding the engine design parameters and the corresponding trajectory that maximize the mission performance index, that is, the payload inserted into a prescribed orbit. A mixed optimization procedure [6] is here adopted. An indirect method [11] optimizes the trajectory for each choice of the engine parameters. These are instead optimized by means of a direct procedure [5]. Both methods have been developed at the Politecnico di Torino.

Tentative values are initially assumed for the design parameters, i.e., F_i, α_i, E, and $(m_O)_{BD}$. For each set of parameters the fast and accurate indirect procedure provides the optimal trajectory and the corresponding performance index; few seconds are required when a 2 GHz PC is used. The design parameters are then varied by small quantities to numerically evaluate the derivatives of the performance index with respect to the design parameters. To find the maximum performance index, a procedure based on Newton's method is used to determine the set of design parameters which simultaneously nullify the index partial derivatives. Only a few minutes are sufficient to obtain the optimal design and the corresponding trajectory.

A point mass rocket is considered for the trajectory optimization. The state equations [6] provide the derivative of: Position **r** (radius, latitude and longitude), velocity **v** (radial, eastward, and northward components) and rocket mass M. In a vectorial form one has

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \mathbf{v} \qquad \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{g} + \frac{\mathbf{F} - \mathbf{D}}{m} \qquad \frac{\mathrm{d}M}{\mathrm{d}t} = -\frac{|\mathbf{F}|}{c^* C_F} \tag{18}$$

An inverse-square gravity field **g** is assumed. An interpolation of the standard atmosphere is used to evaluate density and pressure as a function of the rocket altitude to compute the aerodynamic drag **D**.

The equations of motion are written in non-dimensional form to improve the integration's numerical accuracy. The trajectory is split into phases with homogeneous control law. In the present case, the trajectory consists of four arcs: First burn, split into constant-pressure phase and blowdown phase, coasting, and second burn. The initial rocket conditions (position, velocity, and mass) are given.

The details of the indirect optimization procedure can be found in Ref. [6] and are here only summarized. An adjoint variable is associated to each equation; the theory of optimal control provides the Euler-Lagrange equations for the adjoint variables, algebraic equations that determine the control variables (i.e., the thrust direction), and the boundary conditions for optimality, which also implicitly define the engine switching times. The multipoint boundary value problem, which arises from the application of the theory of optimal control, is solved by a procedure [14] based on Newton's method. Tentative values are initially chosen for the problem unknowns and progressively modified to fulfill the boundary conditions. The unknown parameters are the time lengths of each phase, the initial values of five adjoint variables (the variable corresponding to longitude is null, the one corresponding to the mass is fixed at one, as the problem is homogeneous in the adjoint variables, which can therefore be arbitrarily scaled). The overall oxidizer mass and the grain radius are additional unknowns.

No constraints (dynamic pressure, heat flux, acceleration) are explicitly imposed during the trajectory optimization. However, unconstrained optimal trajectories tend to penetrate deeply into the atmosphere during the coasting arc, causing excessive thermal loads (it is here supposed that, at the ignition of the third stage, the fairing has already been jettisoned). A rigorous analysis of this problem is not carried out here; only, a constraint, which forces the velocity to be horizontal at the end of the first burn, is added. An additional unknown (the adjoint variable corresponding to the horizontal velocity component has a free discontinuity at the end of the first burn) is introduced in the trajectory optimization procedure. This constraint is sufficient to guarantee a trajectory that does not reenter into the atmosphere and has limited thermal loads. Typically, a 10 kg penalty is associated to the introduction of this constraint.

4 Numerical Results for Deterministic Design

The design of a hybrid propellant third stage is considered, with the aim of maximizing the payload delivered into a 700-km polar orbit. Data and boundary conditions are detailed in Ref. [7]. Fixed conditions at the ignition of the third stage, consistent with a launch from Kourou, have been assumed: height $h = 86.88\,\text{km}$, latitude $\varphi = 9.11°$, velocity components in the radial northward and eastward directions $u = 0.142\,\text{km/s}$, $v = 0.230\,\text{km/s}$, $w = 4.146\,\text{km/s}$, respectively, mass 14,522 kg. A 8-port grain is assumed.

The indirect trajectory optimization maximizes the final mass (initial mass minus exhausted propellant) given the propulsion system design, for assigned value of a and n. This is equivalent to maximize the payload, which is evaluated by subtracting the mass of the propulsion system, i.e., the masses of combustion chamber, nozzle, tanks, rocket casing and propellant sliver, from the final mass; these masses are estimated by means of suitable assumptions and approximations [8]. The direct optimization procedure for the design variables determines the values that maximize the payload, i.e., the nozzle expansion area ratio, the oxidizer mass of the constant-pressure phase, and the initial values of thrust and mixture ratio. Very small values of tank pressure would be required for optimal performance, with the risk of poor combustion due to low regression rate. For this reason the tank pressure is here fixed at 25 bar.

Nominal values of the regression rate correlations, that is, Eq. (8), are $a = 7 \cdot 10^{-6}\,\text{m}^{2n+1}\,\text{s}^{n-1}\,\text{kg}^{-n}$ and $n = 0.8$. A sensitivity analysis is first carried out to assess how changes of ballistic properties (a and n) affect the deterministic optimal design. Variations between $6.9 \cdot 10^{-6}$ and $7.1 \cdot 10^{-6}\,\text{m}^{2n+1}\,\text{s}^{n-1}\,\text{kg}^{-n}$ are assumed for a, whereas n varies between 0.79 and 0.81. The optimization procedure for the same initial conditions and final orbit is repeated for the nominal values and for the values

Table 1 Optimal deterministic designs and performance for different ballistic properties

a $\mathrm{m}^{2n+1}\,\mathrm{s}^{n-1}\,\mathrm{kg}^{-n}$	n	m_u kg	F_i kN	α_i	E	$(m_O)_{BD}$ kg
$7.0 \cdot 10^{-6}$	0.80	1955.3	327.7	6.30	15.14	3961.6
$6.9 \cdot 10^{-6}$	0.79	1955.0	317.5	6.38	15.29	3976.2
$6.9 \cdot 10^{-6}$	0.81	1955.0	334.1	6.23	15.04	3951.5
$7.1 \cdot 10^{-6}$	0.79	1955.5	321.5	6.37	15.23	3971.8
$7.1 \cdot 10^{-6}$	0.81	1955.2	338.4	6.22	14.98	3947.1

Table 2 Off-design attainable orbit height (km) of deterministic designs

	Design values					
	Case A $a = 7.0 \cdot 10^{-6}\,\mathrm{m}^{2n+1}\,\mathrm{s}^{n-1}\,\mathrm{kg}^{-n}$ $n = 0.80$			Case B $a = 7.1 \cdot 10^{-6}\,\mathrm{m}^{2n+1}\,\mathrm{s}^{n-1}\,\mathrm{kg}^{-n}$ $n = 0.81$		
a, $\mathrm{m}^{2n+1}\,\mathrm{s}^{n-1}\,\mathrm{kg}^{-n}$	$n = 0.79$	$n = 0.80$	$n = 0.81$	$n = 0.79$	$n = 0.80$	$n = 0.81$
$6.9 \cdot 10^{-6}$	564	669	266	431	534	638
$7.0 \cdot 10^{-6}$	594	700	109	461	565	699
$7.1 \cdot 10^{-6}$	625	511	–	491	596	700

at the extremes of the variation intervals. Results in terms of payload and design variables are summarized in Table 1.

Ballistic properties greatly affect the evolution of thrust and mixture ratio during operation, and relevant changes (up to 7 %) in some of the design variables are required to guarantee almost the same payload, in the presence of even small differences in a and n. The optimal design changes are a clear sign that uncertainties in the knowledge of the ballistic properties may have a large influence on the performance of a given design and justify the need of a robust design approach.

To this purpose, the performance of a design which is optimal for specific a and n values, must be evaluated when the ballistic properties assume different values. For a given design, the rocket mass budget (including payload), grain and feed system features are now all fixed. The optimization procedure maximizes the orbit height to which the payload is delivered (700 km in the nominal case). Five equally spaced values in the variation intervals are evaluated for a and n. Results are presented only for two cases: Case A assumes nominal values, that is, $a = 7.0 \cdot 10^{-6} \mathrm{m}^{2n+1} \mathrm{s}^{n-1}\,\mathrm{kg}^{-n}$ and $n = 0.80$; case B has $a = 7.1 \cdot 10^{-6}\,\mathrm{m}^{2n+1} \mathrm{s}^{n-1}\,\mathrm{kg}^{-n}$ and $n = 0.81$ and exhibits the most robust performance among the deterministic designs. Table 2 shows the orbit height that can be reached by a given (optimal) design as a function of the actual values of a and n used to compute the ascent trajectory.

When a and n assume values that differ from those used to optimize the design, performance is depleted. With similar relative variation ranges, the effect of n (mass flux exponent) is larger than the one of a. The performance of the nominal design rocket (case A) when a and n assume different values shows an evident asymmetry, depending of whether the burning rate is larger (a and n greater than the nominal values) or lower compared to the nominal one. For a large burning rate the fuel is completely burned before all the oxidizer has been exhausted. A large residual oxidizer mass remains on board and greatly penalizes the performance. When $a = 7.1 \cdot 10^{-6}\,\mathrm{m}^{2n+1}\mathrm{s}^{n-1}\,\mathrm{kg}^{-n}$ and $n = 0.81$ the rocket designed for nominal values (case A) cannot even achieve a sufficient ΔV to reach any orbit. On the contrary, for values of the regression rate lower than those used to design the rocket, there is a smaller performance depletion, as the oxidizer is exhausted first, and only a relatively small fuel mass remains on board. This diverse behavior is caused by the relatively large value of mixture ratio; the same residual percentage (which one can assume to be produced by changes in the ballistic properties) corresponds to a much larger residual mass for the oxidizer than the fuel. These observations suggest that a design which is carried out for the ballistic properties corresponding to the fastest burning rates would show good performance for any combination of the actual ballistic coefficients. This supposition is confirmed by the results in Table 2, which shows that, even with a remarkable penalty (up to 270 km), orbit can be achieved even for ballistic coefficients at the opposite extreme of the variation interval.

5 Robust Optimization

A robust design must be such that satisfactory performance can be obtained for any values of a and n. The solutions found using the approach described in the previous section do not have satisfactory off-design performance. The most appealing case (i.e., B) presented in Table 2, may possibly insert the payload 270 km below the required altitude. It is necessary to give up some payload to achieve robustness, that is, the capability of achieving a suitable orbit altitude under non-nominal conditions. The engine must have more fuel and more oxidizer to adapt to both fast and slow burning rates, and it is necessary to enlarge the set of design variables to determine the required values. In the optimal design procedure, oxidizer and fuel masses are implicitly defined to assume the minimum values that allow to achieve the prescribed orbit for the assigned values of a and n (thus maximizing the payload). In the robust design they must assume larger values that must be properly determined.

An evolutionary optimization algorithm developed at Politecnico di Torino [23] is employed to this purpose. The algorithm can employ in parallel different algorithms (genetic algorithm, differential evolution, particle swarm optimization), but only particle swarm optimization with 20 particles is used here, since it exhibits a good compromise between probability of success (i.e., finding the optimum) and convergence speed. The set of design variables is conveniently re-defined to comprise R_g, w, L_b, Z, $(m_O)_f$, and, again, $(m_O)_{BD}$ and E. This set of variables completely determines the propulsion system configuration and the payload. Ranges for the variables are chosen to comprise the optimal design values and are shown in Table 3. Some of the optimization runs produce values that lie on the boundary.

Table 3 Variable ranges for robust optimization

Variable	R_g	w	L_b	Z	$(m_O)_f$	$(m_O)_{BD}$	E
	m	m	m	–	kg	kg	–
Lower boundary	0.45	0.045	3.2	90	9440	3780	14
Upper boundary	0.53	0.056	3.8	140	9730	4360	20

Usually, small improvements are obtained if the variable range is enlarged, but these cases are not discussed here in detail.

A 3×3 grid is used to assess off-design performance (as in Table 2) with $a_i \cdot 10^6 = 6.9, 7, 7.1\,\mathrm{m}^{2n+1}\mathrm{s}^{n-1}\,\mathrm{kg}^{-n}$ for $i = 1, 2, 3$, respectively, and $n_j = 0.79, 0.8, 0.81$ for $j = 1, 2, 3$, respectively. For each individual the altitude of the attained orbit h_{ij} is evaluated for the nine combinations of a and n.

Optimal robust design requires large payloads while assuring satisfactory off-design performance in terms of orbit altitude. Since two objectives are relevant (i.e., payload and altitude), an ϵ-constraint approach [15] is adopted to find the Pareto front of robust solutions. Both the worst-case scenario and the average performance are considered. The worst case scenario evaluates the maximum altitude constraint violation, that is the difference between the imposed altitude ($h^* = 700\,\mathrm{km}$) and the minimum achieved altitude (only when it is below h^*) $\Delta_{max} = \max_{ij}(0,\ h^* - h_{ij})$. The minimum altitude $h_{min} = h^* - \Delta_{max}$ is constrained to a specific value ϵ by maximizing the performance index

$$J_1 = m_u - k_1\ \max(0,\ \epsilon - h_{min}) \tag{19}$$

with a sufficiently large penalty weight k_1 (e.g. 2). On the other hand, the average constraint violation $\Delta_{avg} = \sum_{ij} p_i p_j \max_{ij}(0,\ h^* - h_{ij})$ can be considered (a binomial distribution is assumed giving $p_1 = p_3 = 0.25$ and $p_2 = 0.5$). The average altitude is then $h_{avg} = h^* - \Delta_{avg}$ and the index

$$J_2 = m_u - k_2\ \max(0,\ \epsilon - h_{avg}) \tag{20}$$

is maximized selecting $k_2 = 20$ to force the average altitude at ϵ. Different values are selected for ϵ to evaluate the trade-off between payload and robustness. The ϵ-constraint method works flawlessly and allows to find the non-convex Pareto front. It is almost a straight line for the worst case (continuous line in Fig. 2), with $\mathrm{d}m_u/\mathrm{d}h_{min}$ nearly constant: about 2 kg of payload are lost for a 3 km increase of the minimum altitude. The behavior of average-altitude Pareto front is instead convex (dashed line in Fig. 2). The results of the robust optimization are summarized in Fig. 2 and are compared to the optimal deterministic designs in Tables 4 and 5.

Design variables of the robust optimization exhibit a somehow irregular behavior: exact convergence to the optimum would require large computational times with marginal improvements. The evolutionary algorithms is stopped after 100 steps and the algorithm typically converges to a set of variables which is capable of assuring

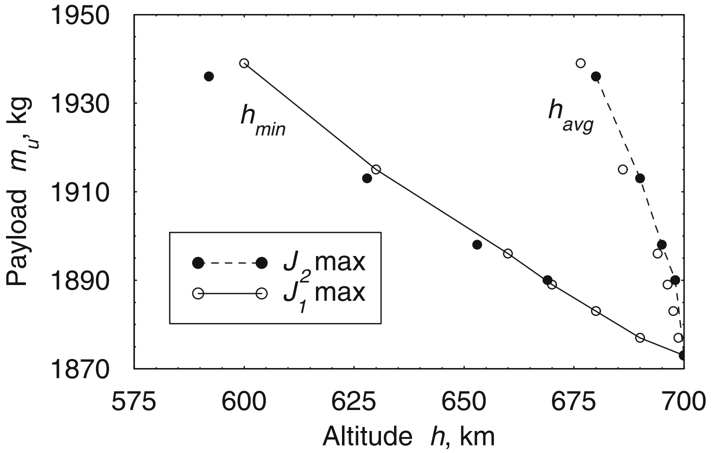


Fig. 2 Minimum and average altitudes of optimal robust designs

Table 4 Performance of deterministic and robust designs

Case	m_u kg	Δ_{max} km	Δ_{avg} km
Deterministic			
A	1955	700	198
B	1955	269	135
J_1 **max**			
$\epsilon = 600$ km	1939	100	23.5
$\epsilon = 630$ km	1915	70	13.9
$\epsilon = 660$ km	1896	40	6.0
$\epsilon = 690$ km	1877	10	1.3
J_2 **max**			
$\epsilon = 680$ km	1936	108	20.0
$\epsilon = 690$ km	1913	72	10.0
$\epsilon = 695$ km	1898	47	5.0
$\epsilon = 698$ km	1890	31	2.0
$\epsilon = 700$ km	1873	0	0.0

sufficient performance. A comparison to the deterministic designs confirm that robustness requires the increase of fuel (mainly) and oxidizer (to a lesser extent) masses. Also, the nozzle expansion ration is increased to improve the specific impulse, thus reducing the propellant requirement (which is subject to uncertainty) at the expense of a heavier nozzle. As the robustness requirements are strengthened, the propellant mass is increased. Performance show that the required 700-km altitude can be assured in any case with an 82-kg penalty in terms of payload, which decreases as the robustness requirement is loosened. It is worth noting that small design changes are required, but they have a quite relevant effect on performance.

Table 5 Comparison of deterministic and robust designs

Case	R_g	w	L_b	Z	$(m_O)_f$	$(m_O)_{BD}$	E
	m	m	m	–	kg	kg	–
Deterministic							
A	0.503	0.0447	3.58	133.7	9541	3961	15.1
B	0.513	0.0459	3.43	125.6	9537	3947	15.0
J_1 **max**							
$\epsilon = 600\,\text{km}$	0.482	0.0559	3.37	90.0	9515	3993	18.6
$\epsilon = 630\,\text{km}$	0.484	0.0547	3.42	90.0	9518	3928	18.9
$\epsilon = 660\,\text{km}$	0.483	0.0557	3.41	90.0	9543	3985	18.1
$\epsilon = 690\,\text{km}$	0.483	0.0556	3.45	90.0	9538	3953	18.4
J_2 **max**							
$\epsilon = 680\,\text{km}$	0.483	0.0560	3.35	90.0	9500	3960	20.0
$\epsilon = 690\,\text{km}$	0.483	0.0560	3.33	90.0	9540	3964	20.0
$\epsilon = 695\,\text{km}$	0.483	0.0555	3.44	90.0	9511	4034	19.7
$\epsilon = 698\,\text{km}$	0.482	0.0557	3.43	90.0	9514	4036	20.0
$\epsilon = 700\,\text{km}$	0.482	0.0560	3.41	90.0	9533	3993	20.0

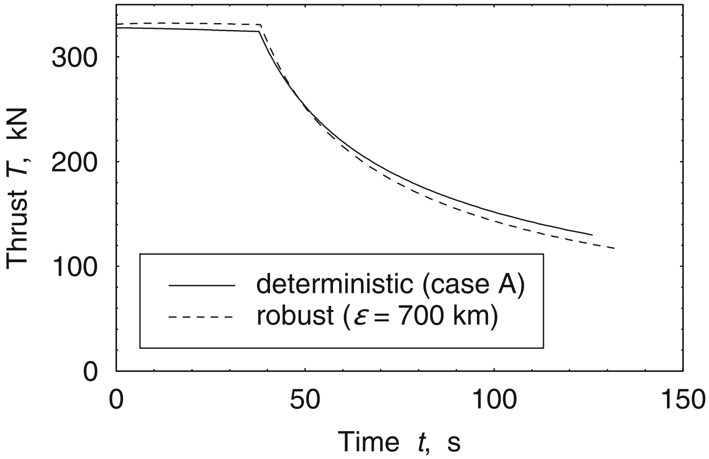


Fig. 3 Thrust history for deterministic design (case A) and optimal robust design ($\epsilon = 700\,\text{km}$)

Figures 3 and 4 compare thrust and mixture ratio histories for the deterministic optimum and the optimal robust design with $\epsilon = 700\,\text{km}$ in the case of nominal values of a and n. The robust design exhibits slightly larger changes of thrust magnitude and mixture ratio shifting, due to the reduced grain radius and larger web thickness, which cause enhanced changes of port and burning area. Differences are however quite limited, as, in this case, robustness can be obtained with moderate changes in the engine design.

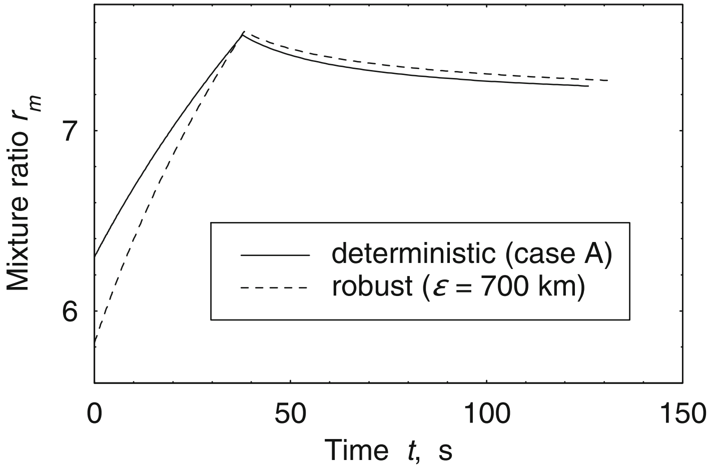


Fig. 4 Mixture ratio history for deterministic design (case A) and optimal robust design ($\epsilon = 700\,\text{km}$)

6 Conclusions

The robust design of a hybrid rocket engine to be used as the third stage of a three-stage launcher is discussed in order to take the uncertainties of the coefficients in the regression rate correlation into account. An analysis of the optimal deterministic designs, which are obtained by means of a coupled direct–indirect optimization method, shows that the required altitude may be missed by a large margin in off-design conditions and justifies the need of a robust design procedure.

A robust optimization procedure based on an evolutionary algorithm is introduced. An ϵ-constraint approach is adopted to maximize payload while minimizing constraint violation either in terms of worst-case scenario or average performance. The procedure proves to be effective and relatively fast, as an optimization run requires roughly 2 h on a standard 2.13 GHz PC. Results prove that robustness can be achieved at the expense of a relatively small payload reduction: a penalty of 82 kg (4 %) with respect to the deterministic optimum is encountered to guarantee a 100 % success in achieving the required altitude.

References

1. Barrere, M., Jaumotte, A., De Veubeke, B.F., Vandenkerckhove, J.: Rocket Propulsion. Elsevier Publishing Company, New York (1960)
2. Ben-Yakar, A., Gany, A.: Hybrid engine design and analysis. Paper AIAA 93–2548. AIAA, Reston, VA (1993)
3. Box, G.E.P., Fung, C.: Is your robust design procedure robust? Qual. Eng. **6**, 503–514 (1993)

4. Brown, C.D.: Spacecraft Propulsion. AIAA Education Series. AIAA, Washingtonn, DC, USA, (1992), p. 82
5. Casalino, L., Pastrone, D.: Oxidizer control and optimal design of hybrid rockets for small satellites. J. Propul. Power **21**, 230–238 (2005)
6. Casalino, L., Pastrone, D.: Optimal design and control of hybrid rockets for access to space. Paper AIAA 2005–3547 (2005)
7. Casalino, L., Pastrone, D.: Optimal design of hybrid rocket motors for launchers upper stages. Paper AIAA 2008–4541 (2008)
8. Casalino, L., Pastrone, D.: Optimal design of hybrid rocket motors for launchers upper stages. J. Propul. Power **26**, 421–427 (2010)
9. Casalino, L., Pastrone, D.: Integrated design-trajectory optimization for hybrid rocket motors. In: Fasano, G., Pinter, J.D. (eds.) Modeling and Optimization in Space Engineering. Springer, New York/Heidelberg/Dordrecht/London (2013)
10. Casalino, L., Pastrone, D.: A straightforward approach for robust design of hybrid rocket engine upper stage. Paper AIAA 2015–4202 (2015)
11. Casalino, L., Colasurdo, G., Pastrone, D.: Optimal low-thrust escape trajectories using gravity assist. J. Guid. Control Dyn. **22**, 637–642 (1999)
12. Casalino, L., Letizia, F., Pastrone, D.: Optimization of hybrid upper stage motor with coupled evolutionary/indirect procedure. J. Propul. Power **30**, 1390–1398 (2014)
13. Casalino, L., Pastrone, D., Simeoni, F.: Approximate and exact approaches for the optimization of hybrid-rocket upper stage. J. Propul. Power **31**, 765–769 (2015)
14. Colasurdo, G., Pastrone, D.: Indirect optimization method for impulsive transfer. Paper AIAA 94–3762 (1994)
15. Haimes, Y., Lasdon, L., Wismer, D.: On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Trans. Syst. Man Cybern. **1**, 296–297 (1971)
16. Isakowitz, S.J., Hopkins, J.A., Hopkins, J.A.Jr.: International Reference Guide to Space Launch Systems, 4th edn. AIAA, Reston, VA (1994)
17. Karabeyoglu, A., Stevens, J., Geysel, D., Cantwell, B., Micheletti, D.: High performance hybrid upper stage motor. AIAA Paper 2011–6025 (2011)
18. Maisonneuve, Y., Godon, J.C., Lecourt, R., Lengelle, G., Pillet, N.: Hybrid propulsion for small satellites: design logic and test. In: Combustion of Energetic Materials. Begell House, New York (2002)
19. Mc Bride, B.J., Reno, M.A., Gordon, S.: CET93 and CETPC: an interim updated version of the NASA Lewis computer program for calculating complex chemical equilibria With applications, NASA TM-4557 (1994)
20. Noor, A.K.: Nondeterministic approaches and their potential for future aero-space systems. NASA/CP-2001-211050. Langley Research Center (2001)
21. Park, G.J., Lee, T.H., Lee, K.H., Hwang, K.H.: Robust design: an overview. AIAA J. **44**, 181–191 (2006)
22. Pastrone, D.: Approaches to low fuel regression rate in hybrid rocket engines. Int. J. Aerosp. Eng. **12**, 1–12 (2012)
23. Rosa Sentinella, M., Casalino, L.: Hybrid evolutionary algorithm for the optimization of interplanetary trajectories. J. Spacecr. Rockets **46**, 365–372 (2009)
24. Shu, N.P.: Axiomatic Design: Advances and Applications. Oxford University Press, New York (2001)
25. Sutton, G.P., Biblarz O.: Rocket Propulsion Elements, 7th edn. Wiley, New York (2001)
26. Taguchi, G., Chowdhury, S., Taguchi, S.: Robust Engineering. McGraw-Hill, New York (2000)
27. Wernimont, E.H., Heister, S.D.: Combustion experiments in hydrogen peroxide/polyethylene hybrid with catalytic ignition. J. Propul. Power **16**, 318–326 (2000)

28. Yao, W., Chen, X., Luo, W., van Tooren, M., Guo, J.: Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicle. Prog. Aerosp. Sci. **47**, 450–479 (2011)
29. Zhu, H., Tian, H., Cai, G.B., Bao, W.M.: Uncertainty analysis and probabilistic design optimization of hybrid rocket motors for manned lunar landing. Sci. China: Technol. Sci. **58**(7), 1234–1241 (2015)
30. Zhu, H., Tian, H., Cai, G.B., Bao, W.M.: Uncertainty analysis and design optimization of hybrid rocket motor powered vehicle for suborbital flight. Chin. J. Aeronaut. **28**(3), 676–686 (2015)

Nonlinear Regression Analysis by Global Optimization: A Case Study in Space Engineering

János D. Pintér, Alessandro Castellazzo, Mariachiara Vola, and Giorgio Fasano

Abstract The search for a better understanding of complex systems calls for quantitative model development. Within this development process, model fitting to observational data (calibration) often plays an important role. Traditionally, local optimization techniques have been applied to solve nonlinear (as well as linear) model calibration problems numerically: the limitations of such approaches in the nonlinear context—due to their local search scope—are well known. In order to properly address this issue, global optimization strategies can be used to find (in practice, to approximate) the best possible model parameterization. This work discusses an application of nonlinear regression model development and calibration in the context of space engineering. We study a scientific instrument, installed on-board of the International Space Station and aimed at studying the Sun's effect on the Earth's atmosphere. A complex sensor temperature monitoring objective has motivated the adoption of an *ad hoc* calibration methodology. Due to the apparent non-convexity of the underlying regression model, a global optimization approach has been implemented: the LGO software package is used to carry out the numerical optimization required periodically for each stage of the analysis. We report computational performance results and offer related insight. Our case study shows the robust and efficient performance of the global scope model calibration approach.

J.D. Pintér (✉)
Lehigh University, Bethlehem, PA, USA

PCS Inc., Halifax, NS, Canada
e-mail: janos.d.pinter@gmail.com; jdp416@lehigh.edu

A. Castellazzo • M. Vola
Altran Italia S.p.A., Consultant c/o Thales Alenia Space Italia S.p.A., Turin, Italy
e-mail: alessandro.castellazzo@altran.com; mariachiara.vola@altran.com

G. Fasano
Exploration and Science, Thales Alenia Space, Turin, Italy

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization
and Its Applications 114, DOI 10.1007/978-3-319-41508-6_11

Keywords Regression analysis • Nonlinear model fitting to data • Global optimization • Space engineering applications • Composite periodic functions • Trend analysis • Failure detection

MSC Classification (2000): 62J02, 90C26, 90C30, 90C90, 90-02, 90-08

1 Introduction

Regression analysis [1–7] is an important subject across a broad range of studies in econometrics, engineering, and the sciences. Nonlinear regression is a general framework for regression analysis in which the observational data are modeled by a postulated nonlinear function: this function is then parameterized according to a stated optimality criterion. A quick Internet search for the key words "Nonlinear Regression" returns close to 2,700,000 results (as of March 2016, using Google's search engine), clearly indicating a substantial interest towards the subject.

The most frequently used classical optimization method to find the parameters of a nonlinear regression model (based on the minimization of the corresponding least squares error function) is the Levenberg–Marquardt algorithm (LMA). The LMA is a modification of the Gauss-Newton method, proposed by Levenberg [8] and rediscovered by Marquardt [9]: consult e.g. the related discussions in Press et al. [10], Björck [2], and Kelley [11]. In the LMA a linearized local approximation of the nonlinear model is used sequentially, and—based on a suitable initial solution "guess"—the model parameters are iteratively refined.

The model fitting exercise can become a hard numerical challenge when the conjectured regression model includes highly nonlinear functions. This study will discuss such a case with compositions of trigonometric functions in the regression model. In similar cases, the error function can be multi-extremal: hence, different initial solution "guesses" can lead to locally best model fitting results of broadly varying quality—calling for a global scope calibration strategy.

Model development studies in which a proper global optimization approach is required arise in numerous real-world applications: consult e.g., Pintér ([12, 13]), Van der Molen and Pintér [14], Finley et al. [15] for related examples and case studies. The substantial advances in global optimization witnessed in recent decades support the application of global optimization algorithms and software to handle challenging nonlinear model fitting problems. Without going into details on the subject of global optimization that are outside of the scope of the present discussion, we refer e.g. to [12, 16–21].

The chapter is organized as follows. The subjects of model calibration, global optimization and information regarding the LGO software package are concisely presented and discussed in Sect. 2. Following these brief expositions that serve as the technical basic of our modeling and solution approach, we present a trend analysis and failure detection case study arising in a current space engineering application (Sect. 3). Section 4 presents concluding notes, followed by a list of references.

Let us mention that a broad range of space engineering case studies is discussed in the volumes edited by Fasano and Pintér [22, 23]: several of these studies include also various model calibration tasks as important ingredients.

2 Global Optimization for Nonlinear Model Fitting

2.1 The Model Calibration Problem

Model development is an essential research tool in many quantitative studies. In very general terms, the following main phases of such development can be distinguished:

1. formulation of model objectives
2. determination of the model structure (functional form selection) based on domain specific knowledge and expertise
3. data collection and analysis, to support model development
4. model fitting to data (calibration, parameterization)
5. validation and sensitivity study
6. applications in analysis, forecasting, management, and so on.

Hence, model calibration is an important stage of the process of understanding and managing complex (chemical, engineering, environmental, physical, or other) systems.

In order to present a general model calibration problem statement, we introduce the following notation:

$t = 1, \ldots, T$ time moments of system observations; T is the number of data used

x model parameters (to be selected according to some chosen optimality criterion); x is assumed to be a real n-vector

D set of admissible (feasible) model parameterizations

M continuous (real-valued, scalar) model function; the values of M depend on x, for each value of $t = 1, \ldots, T$

m_t model output data at time t; $m_t = M(x,t)$; their sequence is $\{m_t\}$, for $t = 1, \ldots, T$

o_t measurement data at time t corresponding to m_t; their sequence is $\{o_t\}$, $t = 1, \ldots, T$

f continuous error function that expresses the discrepancy between the sequences $m = \{m_t\}$ and $o = \{o_t\} : f = f(\{m_t\}, \{o_t\})$.

Applying these notations, the generic model calibration problem can be formulated as

$$\begin{aligned} &\min f(\{m_t\}, \{o_t\}) \\ &m_t = M(x, t) \qquad t = 1, \ldots, T \\ &x \in D \subset R^n. \end{aligned} \tag{1}$$

In order to specify the general model (1), next we present some frequently used model types. The set of feasible parameter settings D can be defined by explicit finite lower and upper bounds (n-vectors l and u) regarding x, as well as by an optional set of $k \geq 0$ additional constraints written in summary form as $g(x) \leq 0$ (g denotes a continuous k-dimensional vector function when such constraints are present):

$$D = \{x : l \leq x \leq u, g(x) \leq 0\}. \tag{2}$$

Based on these conditions, D is a bounded subset of the n-dimensional Euclidean space; we will assume that D is non-empty.

The aggregate model error function f is often defined using a suitably chosen l_p-norm to measure the discrepancy between the vectors m and o:

$$f = f(\{m_t\}, \{o_t\}) = ||m - o||_p \qquad 1 \leq p \leq \infty. \tag{3}$$

Various extensions of this model can be introduced to handle more general formulations, including consideration for uncertainties and/or for multiple model calibration objectives: consult, e.g. Van der Molen and Pintér [14], Pintér [12].

In the context of our discussion, let us point out that the general nonlinear model calibration problem (1)–(3) could well be multi-extremal: cf. e.g. Pintér [12] Chapter 4.5, and several environmental modeling case studies discussed in the same work that illustrate this aspect. For this reason, we have been introducing and using global optimization technology to handle nonlinear model calibration problems across a range of application areas.

2.2 The Global Optimization Model

The model calibration problem (1)–(3) belongs to the general class of continuous global optimization models stated as

$$\min f(x) \tag{4}$$

$$D = \{x : l \leq x \leq u, g(x) \leq 0\} \tag{5}$$

$$f \text{ and } g \text{ (the latter component-wise) are continuous functions in } [l, u]. \tag{6}$$

Notice the absence of the usual convexity assumptions in the above general model formulation that would justify the use of local optimization tools. In (4)–(6) not only the objective f could be multi-extremal, but the feasible region D could also be non-convex. At the same time, the above stated key assumptions already guarantee that the optimal solution set X^* of model (1)–(3) is non-empty. For additional technical details, we refer to Pintér [12].

2.3 LGO Solver Suite for Nonlinear Optimization

The traditional numerical optimization methods used for model calibration seek only for local optima (tacitly assuming the availability of a sufficiently good initial parameter vector). In the general framework presented here this may not be a realistic assumption: therefore global scope search strategies will be required to parameterize (possibly multi-extremal) nonlinear regression models.

Specifically, we will use the Lipschitz Global Optimizer (LGO) solver suite for constrained nonlinear—both global and local—optimization. LGO can handle models with merely continuous structure (without asking for higher order—gradient, Hessian—information); and its operations are based on model function values. This feature makes LGO a suitable choice to tackle a broad range of model calibration problems, including completely "black box" models, in addition to standard (analytically defined) models.

LGO has been discussed in other works, cf. e.g. Pintér [12, 21, 24, 25]: therefore here we present only a summary description. The design of LGO is based on the flexible combination of several nonlinear optimization algorithms, each with corresponding theoretical (provable) global and local convergence properties. It should be noted that the name LGO reflects the original (first) global solver component embedded in the software. (Note in passing that even this solver component uses only model function values, without requiring exact—typically unknown—Lipschitz-continuity information.)

Next, we briefly describe the overall algorithmic structure of LGO. LGO includes a local solver (LS) option which precedes all global search options. LS can be started either from an initial solution point provided by the user, or from a default point determined by LGO. The LS search mode can be also used without a subsequent global search phase. Following the LS phase, two quick global pre-solvers are launched: each of these is followed by LS from the current best point, if an improved solution has been found. The overall purpose of these solver components is to provide a reasonable quality solution with a relatively small global search effort. Next, one of three theoretically "exhaustive" global search options is invoked based on the LGO user's preference: the methods to choose from are branch-and-bound (BB), single-start partially randomized search (RS), and multi-start partially randomized search (MS). Each of BB and RS is followed by a LS phase, while each major MS iteration is followed by a corresponding LS phase.

Based on the solver options summarized above, LGO—as a stand-alone solver suite—can be used for both global and local constrained nonlinear optimization. Without going into further details, we refer to Pintér [12] for an in-depth discussion of the theoretical results leading to the global search options BB, RS and MS. The relatively inexpensive first global pre-solver is described in Pintér and Horváth [26]; the second one is an unpublished heuristic strategy. The LS method is a generalized reduced gradient algorithm implementation: for background, consult e.g. Edgar et al. [27].

In the practical context of numerical optimization—that is, in resource-limited computations—each one of LGO's "exhaustive" global search options generates a global solution estimate(s) that is (are) refined by the seamlessly following local search mode(s). This way, the expected overall result of using LGO is global and/or local search based high-quality feasible solution that satisfies at least the local optimality criteria. (To guarantee theoretical local optimality, standard local smoothness conditions need to apply—at least whenever LS is invoked.)

At the same time, one should keep in mind that no global—or, in fact, any other—optimization software will always work satisfactorily, with default settings and under resource limitations related to model size, time, model function evaluation, or other usage limits. With this cautionary remark in mind, extensive numerical tests and a growing range of practical applications demonstrate that LGO and its platform-specific implementations can find high-quality numerical solutions to complicated and sizeable GO problems. For details, consult e.g. Pintér [12, 19, 25, 28], Pintér and Kampas [29], with references to a range of applications—including also real-world model fitting problems.

LGO is available for use with a range of compiler platforms (C/C++/C#, Fortran 77/90/95), with seamless links to several optimization modeling languages (currently, AMPL, GAMS, MPL), to Excel, and to the leading high-level technical computing systems Maple, *Mathematica*, and MATLAB.

The structure of the compiler-based core LGO implementation used in our study is shown by Fig. 1: a brief explanation of the symbols displayed follows below.

LGOMAIN is a driver program that defines or retrieves from the input file (called LGO.IN) LGO's static calling parameters before activating LGO. The adjective static refers to model descriptor and solver option information that is defined (or read) only once and then remains unchanged during a specific LGO run. LGOMAIN may also include additional user actions such as links to other program files and to external applications, to report generation and to the further optional use of LGO results.

LGOFCT serves to define the dynamic components of an optimization problem: these are defined by the model objective *f* and constraint functions *g*. Here dynamic means that this file will be called at every algorithmic iteration step of LGO, to evaluate its functions depending on the algorithmically generated sequence of input

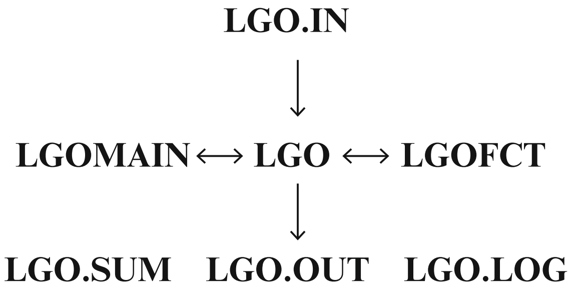


Fig. 1 LGO application program structure

variable arguments x. Again, this file may include calls to other application programs (as needed), in order to evaluate the model functions.

LGO.IN is an optionally used LGO input parameter (text) file that stores LGO's static calling parameters (unless these are directly defined by LGOMAIN).

The source code files LGOMAIN and LGOFCT are to be compiled and linked to the LGO (object or dynamic link library) file. Upon launching the generated executable program, LGOMAIN invokes the LGO solver suite; then LGO iteratively calls LGOFCT.

LGO's operations can be partially controlled by the static input parameter file LGO.IN, or by changing LGOMAIN: this structure supports repeated LGO runs under various model specifications and/or solver option settings. Of course, LGOFCT can also be changed if necessary to test different model variants. LGOMAIN optionally reads LGO.IN when launched; in the opposite case all calling parameters are directly defined in LGOMAIN.

LGO optionally generates result text files, on different levels of detail specified by the user. The first one of these files, called LGO.SUM, presents only a concise summary of the results obtained. The second file, called LGO.OUT, provides more detailed information pertinent to the optimization process. The third file, called LGO.LOG, reports the entire sequence of all arguments x generated and the resulting function values f and g.

For additional details, we refer to the earlier listed references, especially to the current LGO manual [25].

3 A Regression Model Case Study in Space Engineering

3.1 Introduction

Columbus is a science laboratory that is part of the International Space Station (ISS): it is the largest contribution to the ISS made by the European Space Agency (ESA). For information related to the ISS, consult NASA [30]; regarding Columbus, see ESA [31]. The Columbus laboratory carries an extensive collection of instruments. These instruments—referred to as payloads—are aimed at performing various requested scientific experiments, and can be located either internally or externally.

The SOLAR (external) payload (ESA [32], see Fig. 2) has the scope of studying the Sun with extremely high accuracy across most of its spectral range. Its scientific contributions are mainly focused on solar and stellar physics, as well as on the Sun's interaction with the Earth's atmosphere. Its monitoring activity has been in continuous operation since its installation outside the ESA Columbus module in February 2008.

SOLAR consists of three instruments that complement each other, to allow measurements of the solar spectral irradiance virtually throughout the whole electromagnetic spectrum (from 17 nm to 100 μm) in which 99 % of all solar

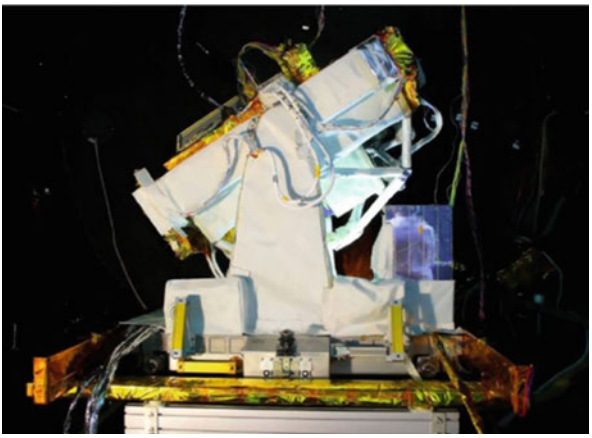


Fig. 2 SOLAR. Copyright © ESA; http://www.esa.int/ESA_Multimedia/Images/2008/01/SOLAR

energy is emitted. These instruments are referred to as SOL-ACES (SOLar Auto-Calibrating Extreme UV/UV Spectrophotometers; see NASA [33]), SOLSPEC (SOLar SPECtral Irradiance measurements; see NASA [34]) and SOVIM (SOlar Variable and Irradiance Monitor; see NASA [35]).

The present discussion is focused on monitoring the SOLAR sensor temperature. Relevant data are retrieved continuously from the ISS to the Earth, in order to carry out a dedicated trend analysis and failure detection activity. This is accomplished periodically (every 3 months), applying regression analysis as described in the following subsections.

3.2 Trend Analysis and Failure Detection

From the point of view of regression modeling, the trend analysis and failure detection activity essentially consists of deriving (repeatedly) the analytical expression representing the sensor temperature trend, from the data available for each time period analysed. (Actually, for safety reasons—in order to increase data reliability—two sensors are utilized and the average of their measurements is considered.) This analysis is then used in conjunction with a reference function to identify possible deviations from the nominal state, together with the identification of possibly occurring anomalies, as well as to predict (through extrapolation) the future behaviour of the system, with respect to the temperature control. A further goal is to verify that the expected temperature trend stays inside the admissible operational range.

At its nominal state, the temperature trend is expected to have two leading modes: a primary (carrier) and a secondary (modulating) periodic mode, as depicted by Fig. 3.

The first mode is associated with the nodal precession of the ISS, with a period of about 2 months [30]. The secondary mode is determined by the orbital motion of the ISS around the Earth, with a period of about 90 min. Both modes are therefore

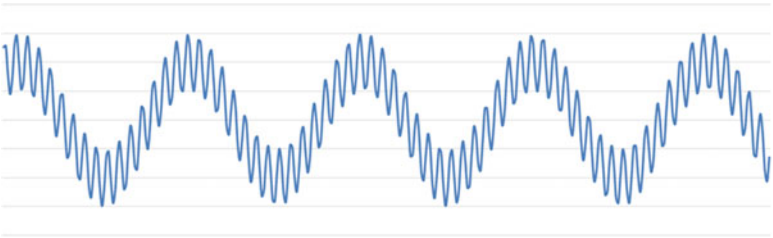


Fig. 3 Primary periodical (carrier) mode and secondary periodical (modulating) mode

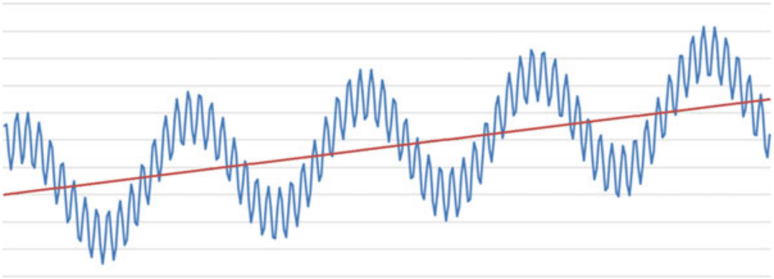


Fig. 4 Possible systematic (linear) degradation function: an example

assumed to have approximately a sinusoidal nominal trend. A possible systematic physical degradation of the SOLAR thermal protection system (due to ambient radiation) is hypothesized next: for the sake of simplicity, the corresponding trend function is assumed to be linear, see Fig. 4.

3.3 The Model Calibration Problem

The analytical formulation of the regression model outlined above leads to an optimization problem, defined by the following objective function

$$\min_{\substack{A_1, A_2,\\ T_{01}, T_{02},\\ K_1, K_2,\\ R, S}} \sum_{i \in I} \{D_i - [A_1 \sin(T_{01} + K_1 T_i) + A_2 \sin(T_{02} + K_2 T_i) + R T_i + S]\}^2 \tag{7}$$

We also consider the following box constraints:

$$\begin{aligned} &A_1 \in \left[\underline{A}_1, \overline{A}_1\right], A_2 \in \left[\underline{A}_2, \overline{A}_2\right], T_{01} \in \left[\underline{T}_{01}, \overline{T}_{01}\right], T_{02} \in \left[\underline{T}_{02}, \overline{T}_{02}\right], \\ &K_1 \in \left[\underline{K}_1, \overline{K}_1\right], K_2 \in \left[\underline{K}_2, \overline{K}_2\right], R \in \left[\underline{R}, \overline{R}\right], S \in \left[\underline{S}, \overline{S}\right]. \end{aligned} \tag{8}$$

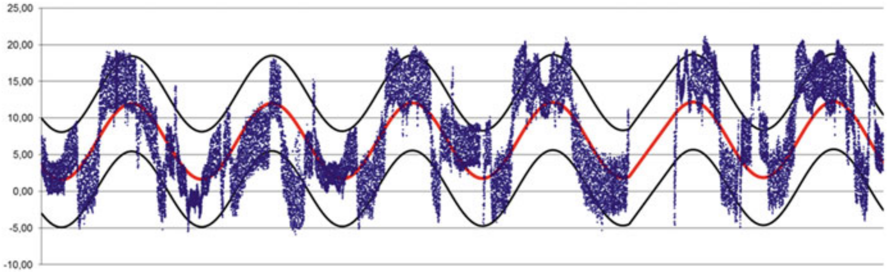


Fig. 5 Solution typology: graphical representation

In model (7)–(8) $A_1, T_{01}, K_1, A_2, T_{02}, K_2, R, S$ are model parameters to estimate; I denotes the set of sampling time moments, and D_i are the corresponding temperature measurements. The objective function terms $A_1 \sin(T_{01} + K_1 t)$ are related to the primary mode, the terms $A_2 \sin(T_{02} + K_2 t)$ are related to the secondary mode, and $Rt + S$ is the possible linear degradation.

The computational difficulty of this global optimization problem is dictated by its highly multi-modal objective function. Evidently (after considering the specific notations), model (7)–(8) is a special case of both generic model formulations (1)–(3) and (4)–(6).

3.4 Solving the Regression Model

This section provides some insights and details regarding the actual application of the globally optimized model calibration approach. Our experimental results will be merely outlined, due to confidentiality restrictions: nonetheless, what follows will suffice to illustrate the efficiency of the methodology adopted.

Let us point out that the amount of telemetry data to handle is huge. Approximately 150 million sample points (observations affected by abnormal gaps and spikes that are to be properly filtered) have been retrieved since 2011. This circumstance has induced the need to develop a dedicated pre-processing package which will not be discussed here.

Figure 5 illustrates a typical solution extracted from the set of those obtained so far, considering 365 days of observation on the horizontal axis and temperature trend expressed in centigrade degrees on the vertical axis.

The experimental analysis that has been performed since 2008 to date has highlighted a slight increment of the amplitude A_1 of the primary mode. A linear degradation rate of about 1.35 degC/year has furthermore been detected based on the mean value since 2008 to date. By extrapolating the latter information, compliance with the currently given tolerance limits would be guaranteed until 2025, well beyond the mission deadline.

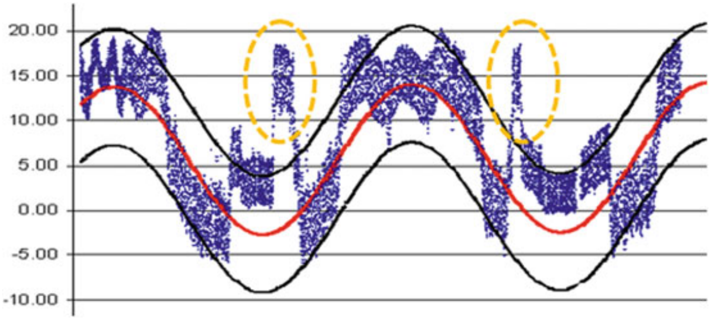


Fig. 6 An example of explained anomalies

No actual anomalies have been identified so far, although apparently some occurred: in fact, these were related to non-nominal manoeuvres of the ISS itself. Figure 6 shows an example of such explainable anomalies, pointing out the supposed deviation by circling the relevant two sets of measurements. In all observed cases, the event times corresponded exactly to specific non-standard control actions performed by the ISS.

In our study, the LGO solver suite has been regularly (sequentially) used to solve model calibration problems of the type (7)–(8). Since the frequencies and amplitudes corresponding to the primary and secondary modes respectively are characterized by pronouncedly different scales, in practice the relevant parameters could be estimated separately. A first analysis is therefore performed regarding the primary set of terms including a possible degradation A_1, T_{01}, K_1, R, S, while neglecting the secondary terms A_2, T_{02}, K_2. Next, a number of short-term subintervals is selected, and then an evaluation of the secondary parameters can be carried out, while keeping the results obtained for the primary terms. The average values of the secondary terms, derived considering the entire set of subintervals selected, will yield the final estimation. At each step of this analysis, the numerical results obtained in the previous step for the entire set of model parameters $(A_1, T_{01}, K_1, A_2, T_{02}, K_2, R, S)$ have been utilized as initial solution “guess” values.

In order to provide an overview of the solution quality obtained by the LGO solver, observational data related to the time period January 2013 to December 2014 have been considered and analysed according to two different temporal frameworks: 365 days and 90 days. A set of 9 model fitting test case results is summarized in Table 1, indicating also the number of observational data used, the computational effort (expressed as hours:minutes:seconds), and the solution quality obtained. The number of the optimized (primary) variables is 5, in each case shown.

For both timeframes considered, LGO finds parameter settings that lead to remarkably well-fitted models. Two examples of the solutions found are shown by Figs. 7 and 8 respectively: the scattered dots (representing actual data) are compared with the continuous line representing the expected trend according to the parameterized model resulting from the analysis.

Table 1 Illustrative experimental results

Test case number	Number of variables	Time period		Number of data entries	Program execution time	Normalized least squares error	Global optimality status
		Days	Dates				
TEST 1	5	365	01/01/2013 → 12/31/2014	2186	01:42:06	21.97	Optimum reached
TEST 2	5	365	04/01/2013 → 03/31/2014	2191	03:16:28	22.17	Optimum reached
TEST 3	5	365	07/01/2013 → 06/30/2014	2270	02:24:37	25.92	Optimum reached
TEST 4	5	365	10/01/2013 → 09/30/2014	2193	01:07:01	23.24	Optimum reached
TEST 5	5	365	01/01/2014 → 12/31/2014	2183	00:43:04	23.12	Optimum reached
TEST 6	5	90	01/01/2014 → 03/31/2014	542	05:52:42	20.04	Optimum reached
TEST 7	5	90	04/01/2014 → 06/30/2014	1083	11:53:49	23.96	Optimum reached
TEST 8	5	90	07/01/2014 → 09/30/2014	1636	18:08:29	18.65	Optimum reached
TEST 9	5	90	10/01/2014 → 12/31/2014	2183	02:12:37	13.88	Optimum reached

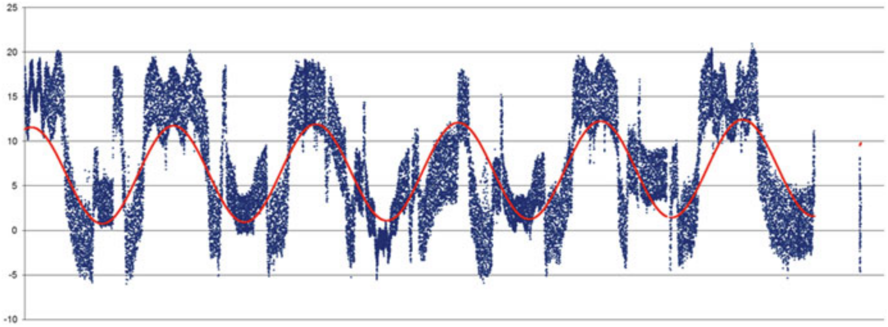


Fig. 7 Example of analysis for the carrier mode (TEST 1, 365 days)

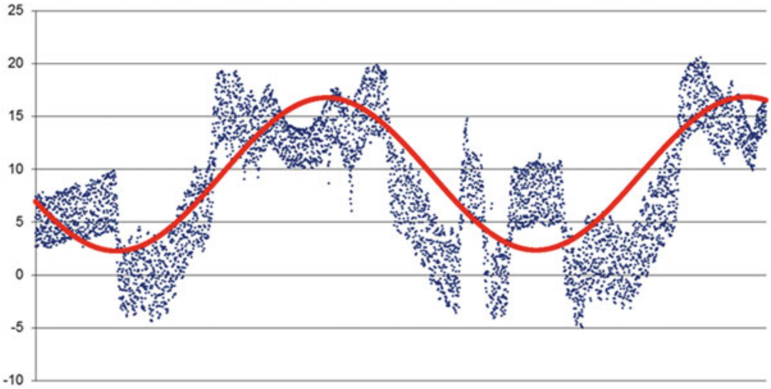


Fig. 8 Example of analysis for the carrier mode (TEST 9, 90 days)

In all examples presented, a high-quality numerical global optimum (estimate) is reached. Additional statistical analysis of the residuals (normalized with respect to the mean value of the model function) exhibits an apparently "normal-like" distribution: see Fig. 9. This finding is in line with the underlying statistical assumptions of the least squares based model fitting paradigm.

4 Conclusions

This work discusses a nonlinear regression model development and calibration study in the context of an application in space engineering. We study the SOLAR payload, installed on-board of the International Space Station. This scientific device is aimed at studying the Sun's effect on the Earth's atmosphere. Due to the apparent non-convexity of the underlying mathematical model, a global optimization approach has been proposed for model calibration. The LGO solver suite is used to carry out

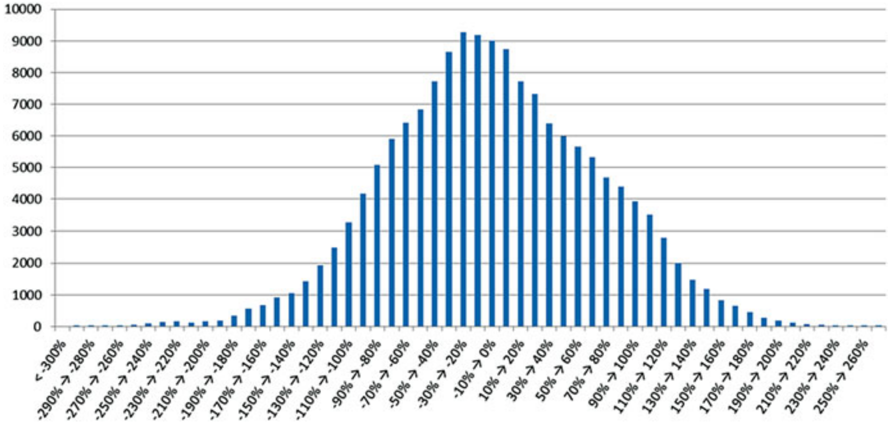


Fig. 9 Normal-like distribution of residuals for carrier mode

the numerical optimization required periodically for each analysis stage. Insights regarding the experimental results and computational performance are provided. Our case study demonstrates the efficiency of the approach proposed, as well as of the software used.

Regarding the SOLAR mission, future research can be directed towards optimizing further model parameters relevant to the payload, such as the voltage/current involved. Extensions to other scientific instruments on-board the International Space Station can also be foreseen, as well as applications related to future scenarios including the anticipated challenge of interplanetary missions.

References

1. Bates, D.M., Watts, D.G.: Nonlinear Regression Analysis and Its Applications. Wiley, New York (1988)
2. Björck, A.: Numerical Methods for Least Squares Problems. Society for Industrial and Applied Mathematics, Philadelphia (1996)
3. Chatterjee, S., Hadi, A.S.: Regression Analysis by Example, vol. 5. Wiley, Hoboken, NJ (2012)
4. Greene, W.H.: Econometric Analysis, 7th edn. Pearson, Harlow, UK (2012)
5. Kleijnen, J.P.C.: Design and Analysis of Simulation Experiments. Springer, New York (2015)
6. Seber, G.A.F., Wild, C.J.: Nonlinear Regression. Wiley, New York (1989)
7. Sen, A., Srivastava, M.: Regression Analysis: Theory, Methods, and Applications, 4th edn. Springer, Berlin (2011)
8. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. Q. Appl. Math. **2**, 164–168 (1944)
9. Marquardt, D.: An algorithm for least-squares estimation of nonlinear parameters. SIAM J. Soc. Ind. Appl. Math. **11**(2), 431–441 (1963)
10. Press, W.H., Teukolsky, S.A., Wetterling, W.T., Flannery, B.: Numerical Recipes in FORTRAN: The Art of Scientific Computing. Cambridge University Press, Cambridge (1992)

11. Kelley, C.T.: Iterative Methods for Optimization. Society for Industrial and Applied Mathematics, Philadelphia (1999)
12. Pintér, J.D.: Global Optimization in Action. Kluwer Academic, Dordrecht (1996)
13. Pintér, J.D.: Globally optimized calibration of nonlinear models: techniques, software, and applications. Optim. Methods. Softw. **18**(3), 335–355 (2003)
14. Van der Molen, D.T., Pintér, J.D.: Environmental model calibration under different problem specifications: an application to the model SED. Ecol. Model. **68**, 1–19 (1993)
15. Finley, J.R., Pintér, J.D., Satish, M.G.: Automatic model calibration applying global optimization techniques. Environ Model Assess **3**, 117–126 (1998)
16. Horst, R., Pardalos, P.M. (eds.): Handbook of Global Optimization, vol. 1. Kluwer Academic, Dordrecht (1995)
17. Liberti, L., Maculan, N. (eds.): Global Optimization: From Theory to Implementation. Springer, New York (2005)
18. Pardalos, P.M., Romeijn, H.E. (eds.): Handbook of Global Optimization, vol. 2. Kluwer Academic, Dordrecht (2002)
19. Pintér, J.D.: Global optimization: software, test problems, and applications. In: Pardalos, P.M., Romeijn, H.E. (eds.) Handbook of Global Optimization, vol. 2, pp. 515–569. Kluwer Academic, Dordrecht (2002)
20. Pintér, J.D. (ed.): Global Optimization: Scientific and Engineering Case Studies. Springer, New York (2006)
21. Pintér, J.D.: Software development for global optimization. In: Pardalos, P.M., Coleman, T.F. (eds.) Global Optimization: Methods and Applications. Fields Institute Communications, vol. 55, pp. 183–204. American Mathematical Society, Providence, RI (2009)
22. Fasano, G., Pintér, J.D. (eds.): Modeling and Optimization in Space Engineering. Springer, New York (2013)
23. Fasano, G., Pintér, J.D. (eds.): Space Engineering: Modeling and Optimization with Case Studies. Springer, New York (2016)
24. 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, Dordrecht (1997)
25. Pintér, J.D.: LGO—A Model Development and Solver System for Global–local Nonlinear Optimization. User's Guide. Current edition. Pintér Consulting Services, Canada (2015)
26. Pintér, J.D., Horváth, Z.: Integrated experimental design and nonlinear optimization to handle computationally expensive models under resource constraints. J Glob Optim **57**, 191–215 (2013)
27. Edgar, T.F., Himmelblau, D.M., Lasdon, L.S.: Optimization of Chemical Processes, 2nd edn. McGraw-Hill, New York (2001)
28. Pintér, J.D.: How difficult is nonlinear optimization? A practical solver tuning approach, with illustrative results. (Submitted for publication) Preprint available for download at http://www.optimization-online.org/DB_HTML/2014/06/4409.html (2014)
29. Pintér, J.D., Kampas, F.J.: Benchmarking nonlinear optimization software in technical computing environments: global optimization in *Mathematica* with *MathOptimizer Professional*. TOP **21**, 133–162 (2013)
30. National Aeronautics and Space Administration: International Space Station. www.nasa.gov/mission_pages/station (2016). Accessed 10 Feb 2016
31. European Space Agency: http://www.esa.int/Our_Activities/Human_Spaceflight/Columbus/Columbus_laboratory (2016). Accessed 10 Feb 2016
32. European Space Agency: http://www.esa.int/Our_Activities/Human_Spaceflight/Columbus/SOLAR (2016). Accessed 10 Feb 2016
33. National Aeronautics and Space Administration: Sun Monitoring on the External Payload Facility of Columbus—SOLar Auto-Calibrating EUV/UV Spectrophotometers. http://www.nasa.gov/mission_pages/station/research/experiments/407.html (2016). Accessed 10 Feb 2016

34. National Aeronautics and Space Administration: Sun Monitoring on the External Payload Facility of Columbus—SOLar SPECtral Irradiance Measurements. http://www.nasa.gov/mission_pages/station/research/experiments/200.html (2016). Accessed 10 Feb 2016
35. National Aeronautics and Space Administration: Sun Monitoring on the External Payload Facility of Columbus—SOlar Variable and Irradiance Monitor. http://www.nasa.gov/mission_pages/station/research/experiments/170.html (2016). Accessed 10 Feb 2016

Regression-Based Sensitivity Analysis and Robust Design

Guido Ridolfi and Erwin Mooij

Abstract This paper presents the Regression-Based global Sensitivity Analysis method (RBSA). It is an approach for quantitative, variance-based, sensitivity analysis of mathematical models used for design purposes. The method is based on the subdivision of the global variance into its components, due to the design-factor contributions, using general polynomial regression models. The performance of the RBSA is compared to other methods commonly used in engineering for computing sensitivity, namely, the method of Sobol', the Fourier amplitude sensitivity test, the method of Morris, and the standardized regression coefficients. It was found that RBSA, under certain circumstances, provides very accurate results with a significant reduction of the number of required model evaluations. A test case, using the mathematical models of two subsystems of a spacecraft, demonstrates how RBSA facilitates the discovery and understanding of the effects of the design choices on the performance of the system.

Keywords Computer-supported design • Decision making • System(s) design • Space systems • Global sensitivity analysis • Conceptual design

1 Introduction

Sensitivity Analysis (SA) is a technique used in many scientific and technical environments with different purposes, such as the determination of the quality of a certain model, validation of assumptions, or as a method to identify important design factors. The SA method proposed in this chapter is intended to support the system-design activities, where the system of interest is represented by its mathematical model. A mathematical representation of the system to be designed (being very preliminary or detailed, depending on the type of analysis to be performed) is fundamental to understand the result of the decisions taken during the design on its final performance (cause → effect), even before the system is built and operated.

G. Ridolfi • E. Mooij (✉)
Faculty of Aerospace Engineering, Delft University of Technology, P.O. Box 5058, 2600 GB Delft, The Netherlands
e-mail: guido.ridolfi@gmail.com; e.mooij@tudelft.nl

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization
and Its Applications 114, DOI 10.1007/978-3-319-41508-6_12

In this context, SA can be described as the study of the *effect* of a certain model input X_i (or group of inputs) on a given model output Y_j. It allows to identify design drivers, i.e., those factors or group of factors that shall be carefully assessed by the design team, because those factors will be the principal responsible for determining the performance of the system. The *effect* mentioned before can be the result of a local measure. It can be, for instance, the measure of a derivative, e.g., $\left(\partial Y_j/\partial X_i\right)_{X_i=x^*}$, which requires an infinitesimal variation of the input X_i around a specific value x^*.

The measure of sensitivity can also be obtained when the input varies over a specified finite interval ΔX_i. In this case, SA is valid over the entire interval of variation spanned by the input factor (i.e., the design region of interest) rather than only directly around a single (operating) point. Therefore, this type of SA is often called *global*. The *global* importance of a factor X_i can be determined on the basis of the reduction of the variance of the output Y_j, $V\left(Y_j\right)$, given that X_i, normally varying over an interval ΔX_i, is fixed to x^* [17, 18].

The computation of global sensitivity based on the variance of the model output is a growing (and also logical) practice. It allows for taking the dimensions of the design region of interest into account to provide multi-dimensional averaged information on the effect of the factors on the output. The advantages of using global sensitivity analysis and factors prioritization during the preliminary design of space systems were already demonstrated [12]. Indeed, the knowledge of the *importance* of the factors in their contribution to the output variance, is fundamental information for engineers and can be used to identify and fix the non-influential factors (or those with a limited influence) on the determination of the output of the model. The most important ones may be ranked and their effect may be studied in more detail.

Sampling the design space is the first step necessary when the mathematical model of a system needs to be studied. A sample is a set of points in the design space (a k-dimensional hyperspace), whose coordinates are the values of the design variables taken from their variability ranges. The model is executed using each sample point as input. The corresponding output, i.e., the performance, can then be studied in detail to draw conclusions on the correlation between input and output. Key requirements for the chosen sampling method are, for a certain required coverage of the design space, the total number of sample points, and the ability to address both continuous and discrete design variables. One such method is the so-called mixed-hypercube approach that was especially developed for this [2, 11, 12, 14]. With some modifications this method can also be used to address the robustness of the design, i.e., to make the design least sensitive to uncertainties that cannot be controlled by the designer.

To address the above mentioned aspects of sensitivity and robustness, this chapter is organized as follows. A literature survey of related work is provided in Sect. 2. In Sect. 3 the Regression Based Sensitivity Analysis (RBSA) method is described in detail. In Sect. 4 a comparison of RBSA with other methods for SA is presented. In Sect. 5 we will introduce the augmented mixed-hypercube approach to be used for robust design. A test case with the step-by-step implementation of RBSA is discussed in Sect. 6, where we address the sensitivities in the design of a satellite's communication and power system, as well as its robustness. Finally, conclusions are provided in Sect. 7.

2 Related Work

Global SA can be computed using qualitative or quantitative methods; it depends on the purpose of the analysis, on the complexity of the problem and on the available computational resources.

A qualitative approach, like the method of Morris [8], allows to determine the importance of the factors with a relatively limited computational effort. It is based on the so-called elementary effect, which is a measure of the sensitivity in the form of incremental ratios, i.e., an approximation of a local gradient within a finite interval of variation of the variable. As such, the elementary effect is a local measure of sensitivity. However, in the method of Morris, the final value attributed to the sensitivity of each design variable is obtained by averaging several elementary effects computed at different points of the input space [8]. The method of Morris provides two qualitative measures of sensitivity, namely the mean, μ, and the standard deviation, σ, of the elementary effects. Large values of μ indicate that a factor has a prominent overall influence on the output. Large values of σ, instead, are the result of interactions of the factors with other factors or non-linear effects on the output. It was recognised that the method of Morris may present some limitations with non-monotonic problems. Campolongo et al. [1] propose an alternative measure of the parameter μ, namely μ^*, to avoid misleading results with non-monotonic models. This is the measure that we also use throughout this chapter. For more information on the method of Morris we refer the reader to the original literature.

The method of Morris, and other qualitative methods, can only rank input factors in order of importance. If from one hand qualitative methods are unable to determine a quantitative measure of the contribution of the factors to the variability of the performance, from the other hand quantitative techniques usually require a large number of model evaluations to perform SA. This may be a limitation when a large number of input factors are taken into account, or when the model is computationally time consuming.

The sensitivity indices introduced by Sobol', for instance, are quantitative [17, 21]. Consider, for instance, $Y = f(X)$ as the mathematical model of the system of interest. Y is the output vector while $X = (x_1, x_2, \cdots, x_k)$ is the vector of the k independent input factors.

To compute the sensitivity with the method of Sobol', a sample of N points is taken from $Y = f(X)$ by evaluating the model N times. The unconditional variance $V(Y)$ can be decomposed as follows [21]:

$$V(Y) = \sum_i V_i + \sum_i \sum_{j>i} V_{ij} + \cdots + V_{12\cdots k} \tag{1}$$

where V_i is the variance of Y due to factor i, V_{ij} is the variance of Y due to the interaction between factor i and factor j. All the terms of this relationship are conditional variances of the factors indicated by the subscripts. For factor i, for instance, $V_i = V(E(Y|x_i))$. For the interaction factor V_{ij}, instead, $V_{ij} = V\left(E\left(Y\left|x_i, x_j\right.\right)\right) - V_i - V_j$, which is the combined effect of the factors x_i and x_j minus their individual conditional variances V_i and V_j.

Since the following relationship holds, $V(Y) = V(E(Y|x_i)) + E(V(Y|x_i))$, and since the unconditional variance $V(Y)$ is constant, an important factor will lead to a small value of $E(V(Y|x_i))$, as anticipated before, or equivalently to a large value of $V(E(Y|x_i))$ [17]. Therefore, each term in Eq. (1) can be used as a measure of global sensitivity. Indeed, Sobol' sensitivity indices are defined as follows [21]:

$$S_i = \frac{V(E(Y|x_i))}{V(Y)} \tag{2}$$

S_i, one of the two measures of sensitivity introduced by Sobol', is sometimes called the *first-order sensitivity index* to distinguish it from *higher-order sensitivity indices* (S_{ij}, S_{ijw}, or S_{ii}, which represent the effects of the interactions between factors or the effect of higher-order terms on the unconditional variance).

Another measure of sensitivity is represented by the so-called total-effect sensitivity indices, S_{Ti}. A total-effect sensitivity index takes the contribution to the unconditional variance of a certain variable X_i into account, considering the first-order and all higher-order effects that involves it. A total sensitivity index provides an indication of the overall effect of a certain variable on the response of the model. The total-effect sensitivity indices can be computed as follows [17]:

$$S_{Ti} = 1 - \frac{V(E(Y|X_{-i}))}{V(Y)} \tag{3}$$

where $V(E(Y|X_{-i}))$ indicates the contribution to the variance due to all factors with the exception of x_i. The vector $X = [x_1, x_2, \ldots, x_i, \ldots, x_k]$ contains all the design factors. The vector $X_{-i} = [x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k]$ contains all the factors except x_i.

Using an analogy with the analysis of signals in the frequency domain, [3] were able to develop an algorithm to compute the sensitivity indices as indicated in Eq. (2). Indeed, the Fourier Amplitude Sensitivity Test (FAST) foresees that each factor x_i is associated with a certain frequency ω_i.

Saltelli et al. [16] proposed the Extended-FAST as an improved version of FAST. The limitation of FAST is that it allows for computing the *first-order* sensitivity indices only [3]. With EFAST the total-effect sensitivity indices as indicated in Eq. (3) can be estimated as well [16].

Many alternative approaches have been developed in the past years for the computation of sensitivity indices for computer models. A thorough discussion of all of them is beyond the scope of the current chapter. Helton and Davis [5] present an analysis of the methods that are most widely used in engineering. The study provides results on the comparison of the performances of the following procedures and measures of sensitivity: correlation coefficients, rank correlation coefficients, common means, common locations, common medians, statistical independence, standardized regression coefficients, partial correlation coefficients, standardized rank regression coefficients, partial rank correlation coefficients, stepwise regression analysis and scatter plots. Despite the limited computational effort required by most of the mentioned procedures, many of them provide local measures of sensitivity while many other only provide a qualitative indication on the ranking of the importance of the design variables in the determination of the output.

Regression analysis is a popular method to assess the effects of input factors on performances. In particular, least-squares procedures are used to construct linear regression models in the following form:

$$\hat{Y} = \hat{\beta}_0 + \sum_{i=1}^{k} \hat{\beta}_i x_i \tag{4}$$

where $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k$ are the estimated regression coefficients. The $\hat{\beta}$ are often used as measure of sensitivity. In general, they do not provide global sensitivity indications, it is the case only for linear models. The same happens when the regression coefficients are expressed as Standardized Regression Coefficients (SRCs), i.e., normalized coefficients, to eliminate the effect of the units in which Y and x_i are expressed, and the effect of the range of variation of the variables.

In general, a linear regression model like Eq. (4), very often results to be poor in approximating the behavior of the model Y, thus the regression coefficients undergo the risk of being quantitatively meaningless and sometimes also qualitatively misleading.

3 Regression-Based Sensitivity Analysis Method

The Regression-Based Sensitivity Analysis method is general enough to be applicable using regression models of any order. However, the choice of the regression order depends on several aspects that will be discussed throughout this section. For ease of discussion the method will be explained using a second-order model as a reference:

$$\hat{Y} = \hat{\beta}_0 + \sum_{i=1}^{k} \hat{\beta}_i x_i + \sum_{i=1}^{k} \hat{\beta}_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \hat{\beta}_{ij} x_i x_j \tag{5}$$

Here, $\hat{\beta}_i$, $\hat{\beta}_{ii}$ and $\hat{\beta}_{ij}$ are the estimated regression coefficients that are calculated by fitting a response surface, using least squares, through the points sampled from the model.

3.1 A Review of the Least-Squares Method

Let us consider a general mathematical model using a compact notation:

$$Y = \beta_0 + \sum_{j=1}^{l} \beta_j x_j \tag{6}$$

where x_j represents any functional involving any of the design variables, for instance $x_j = x_2^2$ or $x_j = x_1 x_2$. In this case the coefficients β_j are the true (unknown) ones, which will be estimated by the coefficients $\hat{\beta}_j$.

Using a least-squares method to estimate the l regression coefficients of the model, at least $N \geq l$ samples are needed. The least-squares method computes an estimation of the regression coefficients minimizing the sum of squares of the errors ϵ_i:

$$Y_i = \beta_0 + \sum_{j=1}^{l} \beta_j x_j + \epsilon_i, \qquad i = 1, 2, \ldots, N \tag{7}$$

In Eq. (7), Y_i represents the observed response for the ith design-variable set $\mathbf{X}_i$. If the model in Eq. (7) is rewritten with matrix notation, i.e., $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ the least-squares method is easier to present and to implement. Here, we have used the following definitions:

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix}, \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1l} \\ 1 & x_{21} & x_{22} & \cdots & x_{2l} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{Nl} \end{bmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_l \end{pmatrix}, \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_N \end{pmatrix}$$

The least-squares estimate of the regression coefficients is computed as follows:

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1} \mathbf{X}^T\mathbf{Y} \tag{8}$$

The utilization of a decomposition method, such as QR factorization or singular value decomposition (SVD), to work with the matrix $\mathbf{X}^T\mathbf{X}$ in Eq. (8) is highly recommended. That matrix may be close to be singular in some cases, also said *ill-conditioned*, and these factorization or decomposition methods are considered numerically stable also with ill-conditioned matrices. The least-squares model is therefore represented by the following relationship:

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\boldsymbol{\beta}} \tag{9}$$

In general, the error between the regression model and the observations may have two main sources. The first source of error is the lack-of-fit of the regression model, when the model for which the regression has been computed does not have enough parameters to explain the data. The second source of error is the measurement performed to collect the sample; in this case it is called *pure error*, or *measurement error*. Since in this case regression analysis is applied to deterministic mathematical models, the pure error is zero. Indeed, given a certain combination of design variables values, the response will always be the same.

In the case of computer experiments, the utilization of least squares for regression analysis may be questionable by some, because of the lack of independent random errors as in physical experiments. In this case, however, least squares are only viewed as a curve fitting tool. This does not imply the assumption of having the residuals behaving like white noise (as for physical experiments). Other methods are considered more suitable for regression analysis of computer experiments (e.g., the kriging method also called Gaussian process). These are non-parametric approaches that do not treat the variables individually in favour of using sets of functions that best interpolate the available data [23]. This aspect makes kriging methods unsuitable for using them for SA purposes.

3.2 Decomposition of the Variance

The total sum of squares of a set of observations of a mathematical model can be expressed as follows:

$$SS_T = \sum_{i=1}^{N} (Y_i - E(\mathbf{Y}))^2 \tag{10}$$

The sum of squares of the regression only, instead, can be computed as follows:

$$SS_R = \sum_{i=1}^{N} \left(\hat{Y}_i - E(\mathbf{Y})\right)^2 \tag{11}$$

SS_R represents the portion of the total variability that can be explained by the regression model. In case the regression model perfectly fits the data then $SS_T = SS_R$. When residuals are present the portion of the total variability not explained by the regression model can be computed in the form of the error sum of squares, SS_E:

$$SS_E = \sum_{i=1}^{N} \left(Y_i - \hat{Y}_i\right)^2 \tag{12}$$

The following relationship holds between the total, regression, and error sum-of-squares:

$$SS_T = SS_R + SS_E \tag{13}$$

To obtain the sensitivity indices of all the factors that contribute to the total variability of the regression model, the regression sum of squares SS_R should be partitioned in its components. The main idea is to associate a sensitivity index to the additional variability calculated when a factor is added to the regression model. To do so, a matrix notation for the sum of squares is now introduced. Combining Eqs. (8) and (9), the regression model can be expressed as follows:

$$\hat{\mathbf{Y}} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y} = \mathbf{H}\mathbf{Y} \tag{14}$$

The matrix $\mathbf{H} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$ is called the *hat* matrix. It transforms the vector of the observed responses $\mathbf{Y}$ into the vector of the fitted values $\hat{\mathbf{Y}}$. Using the hat matrix, the total, regression and error sums of squares can be expressed with the following relationships [7]:

$$SS_T = \mathbf{Y}^T\left[\mathbf{I} - \frac{1}{N}\mathbf{J}\right]\mathbf{Y};\ SS_R = \mathbf{Y}^T\left[\mathbf{H} - \frac{1}{N}\mathbf{J}\right]\mathbf{Y};\ SS_E = \mathbf{Y}^T\left[\mathbf{I} - \mathbf{H}\right]\mathbf{Y} \tag{15}$$

where $\mathbf{I}$ is an $N \times N$ identity matrix, and $\mathbf{J}$ is an $N \times N$ matrix of ones.

In literature there are several methods that are most widely used to obtain the variance decomposition of Eq. (5) [4]. As one of the possible models that Eq. (5) can describe, let us consider the following, with 3 factors:

$$\begin{aligned}\hat{Y} =&\hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3 + \hat{\beta}_{11} x_1^2 + \hat{\beta}_{22} x_2^2 + \hat{\beta}_{33} x_3^2 + \\ &+ \hat{\beta}_{12} x_1 x_2 + \hat{\beta}_{13} x_1 x_3 + \hat{\beta}_{23} x_2 x_3\end{aligned} \tag{16}$$

In the following discussion, $SS(Y_{x_1})$ represents the sum of squares associated with the model computed with only the factor x_1 (i.e., $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 x_1$). $SS(x_2|\, Y_{x_1})$ represents the sum of squares associated with a regression model where x_2 is added to the model given that x_1 is already present, it will also be indicated as $SS(x_2)$ since it is the sum of squares associated with x_2 only. This indicates the additional variability explained by adding x_2 to the model.

The Type-II sum-of-squares decomposition, or classical sum of squares, indicates the change in the variability explained by the regression model due to adding an extra term to the model, given that all other terms have been added except for the terms that contain the effect under test. For instance, the sum of squares of factor x_3, with x_1 and x_2 in the model, with all interactions (two and three factor-interactions) can be computed as follows:

$$SS(x_3) = SS(x_3|\, Y_{x_1 x_2 x_{12}}) = SS(Y_{x_1 x_2 x_3 x_{12} x_{13} x_{23}}) - SS(Y_{x_1 x_2 x_{12}}) \tag{17}$$

Another method for sum-of-squares decomposition computes the contribution to the variability explained by the regression model due to adding an extra term, given that all other terms are already in the model, including the interactions and higher-order factors involving the term under investigation. The sum of squares of Type-III for the factor x_3 of the model in Eq. (16) would be as follows:

$$SS(x_3) = SS(x_3|\, Y_{x_1x_2x_{12}}) = SS(Y_{x_1x_2x_3x_{12}x_{13}x_{23}}) - SS(Y_{x_1x_2x_{12}x_{13}x_{23}}) \tag{18}$$

In Eqs. (17) and (18) the term $Y_{(\cdot)}$ represents the regression model with all the factors and interactions indicated by the subscripts. Given the sum of squares associated with every factor of the regression model, the sensitivity indices can be computed with a relationship that is similar to that presented in Eqs. (2) and (3):

$$S_i = \frac{SS(x_i)}{SS_R + SS_E} \tag{19}$$

Indeed, the sensitivity measures computed using Eq. (19) can be interpreted in terms of the first-order and total-order sensitivity indices. When $SS(x_i)$ is computed with the Type-II decomposition, Eq. (17), it describes the contribution of a factor considering, simultaneously, all the interactions and higher-order effects involving it. Thus, it provides information on the total effect of that factor. Using the Type-III decomposition to compute $SS(x_i)$, Eq. (18), instead, we obtain the contribution of each term of the polynomial regression model (e.g., x_1, x_1^2, or x_1x_2) to the total variability computed with the regression model. This allows to compute the contribution to the variance of individual effects in a way that is not allowed with other approaches discussed in the previous section. And this is possible with no additional simulations.

In the case of RBSA we call the effects of the individual factors [computed with Eqs. (18) and (19)] *first-order* effects. In these cases it would be more appropriate calling them *individual-order* effects since they refer to individual terms in the regression model, therefore also the quadratic (e.g., x_1^2) or interaction (e.g., x_1x_2) terms. With the Sobol' method, or FAST, it is only possible to compute the actual first-order sensitivity indices (e.g., sensitivity indices of x_1, x_2, etc.).

3.3 The Algorithm for RBSA

The RBSA algorithm begins with an educated hypothesis on the behavior of the model in the design region of interest. Eq. (5) could be used, for instance, as an initial assumption. However, if later in the process *inacceptable* lack-of-fit is detected, this assumption could be reviewed by modifying the regression model and adding cubic (e.g, x_i^3) or higher-order interaction terms (e.g, $x_ix_jx_k$), for instance. For the moment, let us use the model presented in Eq. (5).

The second step consists in the creation of an input sample matrix **M**, made of k columns (the number of design variables taken into account) and N rows:

$$\mathbf{M} = \begin{pmatrix} x^{11} & x^{12} & \cdots & x^{1k} \\ x^{21} & x^{22} & \cdots & x^{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x^{N1} & x^{N2} & \cdots & x^{Nk} \end{pmatrix} \tag{20}$$

Each row represents a design vector with a value for each design variable: each row represents a *sample point*. The sample size N shall be larger than the number of coefficients to estimate. For instance, $N > 2k + k(k-1)/2$ samples are needed for the regression analysis on the model of Eq. (5). The output vector $\mathbf{Y}$ is obtained by executing the mathematical model with the rows of $\mathbf{M}$ as inputs.

The next step is to build the matrix $\mathbf{X}$ that will be used to compute the sum of squares and the sensitivity indices. The construction of $\mathbf{X}$, and the methodology to compute the sensitivity indices, will only be presented specifically for the model in Eq. (5). The derivation for regression models of different orders is similar. First, the two matrices $\mathbf{R_1}$ and $\mathbf{R_2}$ with dimensions $N \times k(k-1)/2$ shall be obtained by a re-arrangement of the columns of $\mathbf{M}$.

$$\mathbf{R}_1 = \left(\begin{array}{cccccccc} \mathbf{M}_{(1,1)} & \ldots & \mathbf{M}_{(1,1)} & \mathbf{M}_{(1,2)} & \ldots & \mathbf{M}_{(1,2)} & \ldots & \mathbf{M}_{(1,k\text{-}1)} \\ \mathbf{M}_{(2,1)} & \ldots & \mathbf{M}_{(2,1)} & \mathbf{M}_{(2,2)} & \ldots & \mathbf{M}_{(2,2)} & \ldots & \mathbf{M}_{(2,k\text{-}1)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{M}_{(N,1)} & \ldots & \mathbf{M}_{(N,1)} & \mathbf{M}_{(N,2)} & \ldots & \mathbf{M}_{(N,2)} & \ldots & \mathbf{M}_{(N,k\text{-}1)} \\ \multicolumn{3}{c}{\text{k-1}} & \multicolumn{3}{c}{\text{k-2}} & & \text{1} \end{array}\right)$$

$$\mathbf{R}_2 = \left(\begin{array}{cccccccc} \mathbf{M}_{(1,2)} & \ldots & \mathbf{M}_{(1,k)} & \mathbf{M}_{(1,3)} & \ldots & \mathbf{M}_{(1,k)} & \ldots & \mathbf{M}_{(1,k)} \\ \mathbf{M}_{(2,2)} & \ldots & \mathbf{M}_{(2,k)} & \mathbf{M}_{(2,3)} & \ldots & \mathbf{M}_{(2,k)} & \ldots & \mathbf{M}_{(2,k)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{M}_{(N,2)} & \ldots & \mathbf{M}_{(N,k)} & \mathbf{M}_{(N,3)} & \ldots & \mathbf{M}_{(N,k)} & \ldots & \mathbf{M}_{(N,k)} \\ \multicolumn{3}{c}{\text{k-1}} & \multicolumn{3}{c}{\text{k-2}} & & \text{1} \end{array}\right)$$

The matrices $\mathbf{R_1}$ and $\mathbf{R_2}$ can be visualized in blocks. The first $k-1$ columns of $\mathbf{R_1}$ are $k-1$ replications of the first column of $\mathbf{M}$. The second block of $k-2$ columns is made of the replication of the second column of $\mathbf{M}$, and so on until the last-but-one column of $\mathbf{M}$, which appears only once. $\mathbf{R_2}$ is built with a different approach, but the visualization by blocks is still possible. The first $k-1$ columns of $\mathbf{R_2}$ are replications of the second to last column of $\mathbf{M}$. The second block of $k-2$ columns consists of the third to last column of $\mathbf{M}$, and so on until the last column of $\mathbf{M}$, which appears only once. Therefore, the elements of $\mathbf{R_1}$ and $\mathbf{R_2}$ can be interpreted as follows: $\mathbf{M}_{(1,1)} = x^{11}$, $\mathbf{M}_{(1,k)} = x^{1k}$, and $\mathbf{M}_{(N,k)} = x^{Nk}$.

The coefficient-wise (i.e., Hadamart, indicated by ∘) product of $\mathbf{R_1}$ and $\mathbf{R_2}$ gives the matrix $\mathbf{R}$:

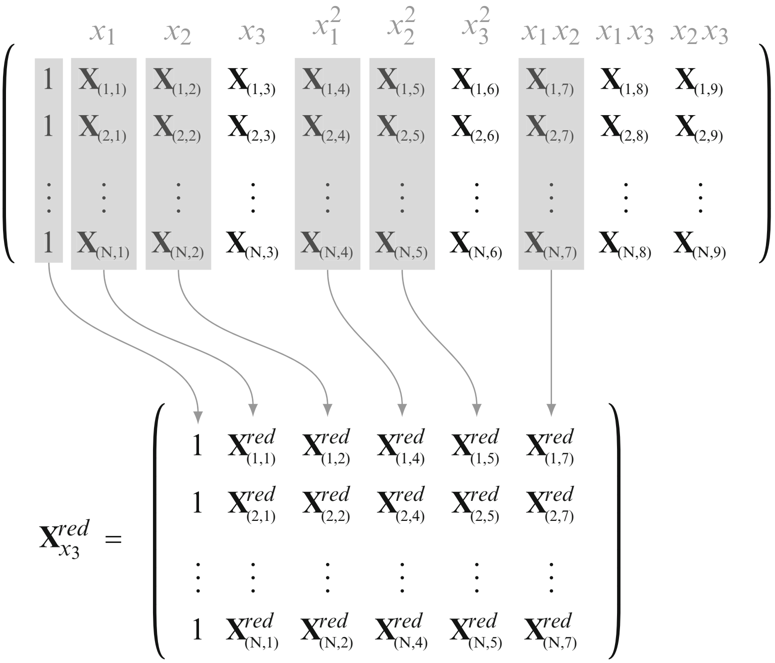
$$\mathbf{R} = \mathbf{R_1} \circ \mathbf{R_2}$$

Each element of $\mathbf{R}$ is obtained by multiplying the corresponding elements of $\mathbf{R_1}$ and $\mathbf{R_2}$, i.e., $\mathbf{R}_{ij} = \mathbf{R}_{1(ij)} \times \mathbf{R}_{2(ij)}$. $\mathbf{R}$ will be used to compute the interaction effects for the sensitivity indices. The matrix $\mathbf{X}$ to be used for the regression analysis is obtained by re-arranging the columns of $\mathbf{M}$ and $\mathbf{R}$:

$$\mathbf{X} = \begin{pmatrix} 1 & \mathbf{M}_{(1,1)} & \dots & \mathbf{M}_{(1,k)} & \left(\mathbf{M}_{(1,1)}\right)^2 & \dots & \left(\mathbf{M}_{(1,k)}\right)^2 & \mathbf{R}_{(1,1)} & \dots & \mathbf{R}_{(1,k(k\text{-}1)/2)} \\ 1 & \mathbf{M}_{(2,1)} & \dots & \mathbf{M}_{(2,k)} & \left(\mathbf{M}_{(2,1)}\right)^2 & \dots & \left(\mathbf{M}_{(2,k)}\right)^2 & \mathbf{R}_{(2,1)} & \dots & \mathbf{R}_{(2,k(k\text{-}1)/2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & \mathbf{M}_{(N,1)} & \dots & \mathbf{M}_{(N,k)} & \left(\mathbf{M}_{(N,1)}\right)^2 & \dots & \left(\mathbf{M}_{(N,k)}\right)^2 & \mathbf{R}_{(N,1)} & \dots & \mathbf{R}_{(N,k(k\text{-}1)/2)} \end{pmatrix}$$

Once $\mathbf{X}$ is available, SS_T, SS_R, and SS_E can be computed using the *hat* matrix $\mathbf{H}$ and the relationships presented in Eq. (15).

Total-order sensitivity indices are computed for every design variable in the model. For each of the k design variables a reduced version of $\mathbf{X}$, namely $\mathbf{X}^{red}$, needs to be build. $\mathbf{X}^{red}$ is obtained by removing the columns of $\mathbf{X}$ that are related to all terms involving the related design variable for which the total sensitivity index is computed. For instance, consider the model of Eq. (5) with three design variables. The construction of $\mathbf{X}^{red}_{x_3}$ for the variable x_3 would be as shown below, by removing the *white* columns:


x_1
x_2
x_3
x_1^2
x_2^2
x_3^2
x_1x_2
x_1x_3
x_2x_3
1 $\mathbf{X}_{(1,1)}$ $\mathbf{X}_{(1,2)}$ $\mathbf{X}_{(1,3)}$ $\mathbf{X}_{(1,4)}$ $\mathbf{X}_{(1,5)}$ $\mathbf{X}_{(1,6)}$ $\mathbf{X}_{(1,7)}$ $\mathbf{X}_{(1,8)}$ $\mathbf{X}_{(1,9)}$
1 $\mathbf{X}_{(2,1)}$ $\mathbf{X}_{(2,2)}$ $\mathbf{X}_{(2,3)}$ $\mathbf{X}_{(2,4)}$ $\mathbf{X}_{(2,5)}$ $\mathbf{X}_{(2,6)}$ $\mathbf{X}_{(2,7)}$ $\mathbf{X}_{(2,8)}$ $\mathbf{X}_{(2,9)}$
1 $\mathbf{X}_{(N,1)}$ $\mathbf{X}_{(N,2)}$ $\mathbf{X}_{(N,3)}$ $\mathbf{X}_{(N,4)}$ $\mathbf{X}_{(N,5)}$ $\mathbf{X}_{(N,6)}$ $\mathbf{X}_{(N,7)}$ $\mathbf{X}_{(N,8)}$ $\mathbf{X}_{(N,9)}$
$\mathbf{X}^{red}_{x_3} =$
1 $\mathbf{X}^{red}_{(1,1)}$ $\mathbf{X}^{red}_{(1,2)}$ $\mathbf{X}^{red}_{(1,4)}$ $\mathbf{X}^{red}_{(1,5)}$ $\mathbf{X}^{red}_{(1,7)}$
1 $\mathbf{X}^{red}_{(2,1)}$ $\mathbf{X}^{red}_{(2,2)}$ $\mathbf{X}^{red}_{(2,4)}$ $\mathbf{X}^{red}_{(2,5)}$ $\mathbf{X}^{red}_{(2,7)}$
1 $\mathbf{X}^{red}_{(N,1)}$ $\mathbf{X}^{red}_{(N,2)}$ $\mathbf{X}^{red}_{(N,4)}$ $\mathbf{X}^{red}_{(N,5)}$ $\mathbf{X}^{red}_{(N,7)}$

Using $\mathbf{X}^{red}$ the regression sum of squares, SS_R^{red}, can be computed using Eqs. (14) and (15):

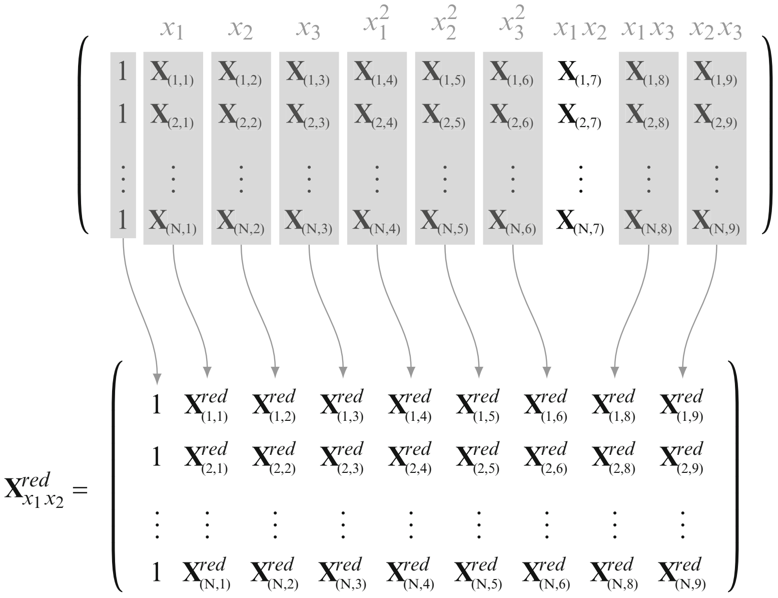
$$\mathbf{H}^{red} = \mathbf{X}^{red}\left(\mathbf{X}^{Tred}\mathbf{X}^{red}\right)^{-1}\mathbf{X}^{T,red} \tag{21}$$

$$SS_R^{red} = \mathbf{Y}^T\left[\mathbf{H}^{red} - \frac{1}{N}\mathbf{J}\right]\mathbf{Y} \tag{22}$$

SS_R^{red} of a certain design variable x_i indicates the variability that the model without the contribution of the terms that involve x_i is able to explain. The difference between the regression sum of squares computed with Eqs. (14) and (15) using $\mathbf{X}$ and the *reduced* regression sum of squares indicates the overall contribution of the design variable x_i to the variability detected by the full model (i.e., Type-II sum of square). Thus, a total-order sensitivity index may be computed as follows:

$$S_{Ti} = \frac{SS_R - SS_R^{red}}{SS_R + SS_E} \tag{23}$$

First-order sensitivity indices can be obtained in a very similar fashion for each term of the model, including interactions and higher-order terms. For each term of the model $\mathbf{X}^{red}$ needs to be build, as in the previous case. $\mathbf{X}^{red}$ is again a *reduced version* of the matrix $\mathbf{X}$, but in this case it is obtained by removing only the column of $\mathbf{X}$ that is related to the term of interest. For instance, the construction of $\mathbf{X}^{red}_{x_1x_2}$ for the interaction term x_1x_2 would be as follows:



x_1 x_2 x_3 x_1^2 x_2^2 x_3^2 x_1x_2 x_1x_3 x_2x_3
1 $\mathbf{X}_{(1,1)}$ $\mathbf{X}_{(1,2)}$ $\mathbf{X}_{(1,3)}$ $\mathbf{X}_{(1,4)}$ $\mathbf{X}_{(1,5)}$ $\mathbf{X}_{(1,6)}$ $\mathbf{X}_{(1,7)}$ $\mathbf{X}_{(1,8)}$ $\mathbf{X}_{(1,9)}$
1 $\mathbf{X}_{(2,1)}$ $\mathbf{X}_{(2,2)}$ $\mathbf{X}_{(2,3)}$ $\mathbf{X}_{(2,4)}$ $\mathbf{X}_{(2,5)}$ $\mathbf{X}_{(2,6)}$ $\mathbf{X}_{(2,7)}$ $\mathbf{X}_{(2,8)}$ $\mathbf{X}_{(2,9)}$
1 $\mathbf{X}_{(N,1)}$ $\mathbf{X}_{(N,2)}$ $\mathbf{X}_{(N,3)}$ $\mathbf{X}_{(N,4)}$ $\mathbf{X}_{(N,5)}$ $\mathbf{X}_{(N,6)}$ $\mathbf{X}_{(N,7)}$ $\mathbf{X}_{(N,8)}$ $\mathbf{X}_{(N,9)}$
$\mathbf{X}^{red}_{x_1x_2} =$
1 $\mathbf{X}^{red}_{(1,1)}$ $\mathbf{X}^{red}_{(1,2)}$ $\mathbf{X}^{red}_{(1,3)}$ $\mathbf{X}^{red}_{(1,4)}$ $\mathbf{X}^{red}_{(1,5)}$ $\mathbf{X}^{red}_{(1,6)}$ $\mathbf{X}^{red}_{(1,8)}$ $\mathbf{X}^{red}_{(1,9)}$
1 $\mathbf{X}^{red}_{(2,1)}$ $\mathbf{X}^{red}_{(2,2)}$ $\mathbf{X}^{red}_{(2,3)}$ $\mathbf{X}^{red}_{(2,4)}$ $\mathbf{X}^{red}_{(2,5)}$ $\mathbf{X}^{red}_{(2,6)}$ $\mathbf{X}^{red}_{(2,8)}$ $\mathbf{X}^{red}_{(2,9)}$
1 $\mathbf{X}^{red}_{(N,1)}$ $\mathbf{X}^{red}_{(N,2)}$ $\mathbf{X}^{red}_{(N,3)}$ $\mathbf{X}^{red}_{(N,4)}$ $\mathbf{X}^{red}_{(N,5)}$ $\mathbf{X}^{red}_{(N,6)}$ $\mathbf{X}^{red}_{(N,8)}$ $\mathbf{X}^{red}_{(N,9)}$

Using $\mathbf{X}^{red}$ the regression sum of squares SS_R^{red} is again obtained with Eqs. (21) and (22). The first-order sensitivity index for each term of the model may be computed as presented before:

$$S_i = \frac{SS_R - SS_R^{red}}{SS_R + SS_E} \tag{24}$$

In this case the difference between the regression sum of squares computed with Eqs. (14) and (15) using $\mathbf{X}$ and the *reduced* regression sum of squares indicates the Type-III sum of squares indicated in Eq. (18).

This approach to compute the sensitivity indices, based on regression analysis provides some advantages. First of all, the number of model evaluations, that is usually the most resource-consuming part of the analysis, is reduced (a numerical comparison is provided in Sect. 4). Second, the RBSA provides quantitative information (rather than qualitative as most of the screening or sample-based SA methods) also on the effects of interactions and higher-order terms on the performance of interest (rather than only first-order and total sensitivity indices as the method of Sobol' or FAST). The fact that higher-order models are implemented, rather than linear models only, allows to explain a larger part of variability when compared to the SRCs method, for instance.

One possible drawback of RBSA is that the validity of the results depends on the lack-of-fit of the regression model with respect to the sample data. Indeed, special attention must be paid to the ratio between the regression and the total sum of squares. If SS_R is close to SS_T, then the regression model is able to account for a large part of the output variance, and as a consequence the sensitivity indices are meaningful measures. If this is not the case, lack-of-fit is present meaning that important terms are missing from the initially assumed regression model. Lack-of-fit is important to decide whether to proceed with SA anyway or to modify the initial assumption and increase the order of the regression model by adding extra terms, e.g., higher-order terms like cubic or higher-order interactions.

3.4 Testing for Model Adequacy

Testing for model adequacy is a fundamental step, since it is a means to validate the results of the SA, allowing to mitigate the effect of the lack-of-fit on the sensitivity indices by an iterative approach (see also Sect. 3.5). The presence of lack-of-fit could be related to the fact that important terms have been neglected, or simply that the chosen polynomial regression model is not entirely adequate to reproduce the relationships between the design variables, e.g., in case of higher-order effects, exponential or sinusoidal effects.

The coefficients of determination, R^2, and often its adjusted version R^2_{adj} allow to detect the fraction of the model output variance accounted for by the regression model [7]:

$$R^2 = \frac{\sum_{i=1}^{N}\left(\hat{Y}_i - \bar{Y}\right)^2}{\sum_{i=1}^{N}\left(Y_i - \bar{Y}\right)^2} = \frac{SS_R}{SS_T} \qquad 0 \leq R^2 \leq 1 \tag{25}$$

Very often the *adjusted* coefficient of determination, R^2_{adj}, is used instead of R^2:

$$R^2_{adj} = 1 - \left(\frac{N-1}{N-(l+1)}\right)\left(1 - R^2\right) \qquad 0 \leq R^2_{adj} \leq 1 \tag{26}$$

where l indicates the total number of regressors in the polynomial model (without the constant term β_0).

Values of R^2 or R^2_{adj} larger than 0.9 usually suggest a good fit of the data. The extreme case in which R^2 or R^2_{adj} are equal to one, indicates that the regression model is able to account for all the variability of the model output, but this does not always mean that the regression model perfectly matches the true one in all points of the design region. Having R^2 or R^2_{adj} equal to one may also be caused by using regression model of a lower order than the real one. Increasing the order of the regression model could substantially help for a better reconstruction of the underlying relationships between the design variables. To reduce lack-of-fit increasing the sample size alone is in general only partially beneficial. A better approximation is obtained when there is also an increase in the order of the regression model.

Concluding, there is not a general and guaranteed approach to identify lack-of-fit. It is advised, though, to build the regression models with a number of samples that exceeds the actual number of terms needed to build the model. In this way, more degrees of freedom for the estimation of the error are provided, avoiding to obtain misleading values for R^2 or R^2_{adj}. Here, we will use the R^2_{adj} as a measure of lack-of-fit. The discussion on the model adequacy provided in this section is limited to the implementation needed for the proposed RBSA methodology. For a more complete analysis the interested readers may consider the books of [4] and [7].

3.5 The Iterative Approach to RBSA

In Table 1 a list of suggested regression models of increasing order, with the minimum number of samples required to compute all the coefficients, is presented. This particular choice is merely indicative, it shall be considered as an example to explain the iterative approach to RBSA. The minimum number of samples for every regression model is equal to the number of factors present in the model plus extra sample points equal to the number of variables of the model. The decision to modify the initial assumptions on the regression model depends on the adequacy of the current one, determined by R^2_{adj}.

At the beginning of the process, the minimum number of samples for fitting a linear model is collected. If R^2_{adj} is lower than a certain threshold value, e.g.,

Table 1 Suggested regression models for the iterative procedure

Model order	Regression model	Minimum number of samples
1	$Y_1 = \beta_0 + \sum \beta_i x_i$	$2 + k$
$1^1/_2$	$Y_{^1/_2} = Y_1 + \sum \beta_{ij} x_i x_j$	$2 + k + \binom{k}{2}$
2	$Y_2 = Y_{^1/_2} + \sum \beta_{ii} x_i^2$	$2 + 2k + \binom{k}{2}$
3	$Y_3 = Y_2 + \sum \beta_{iii} x_i^3 + \sum \beta_{ijw} x_i x_j x_w$	$2 + 3k + \binom{k}{2} + \binom{k}{3}$
4	$Y_4 = Y_3 + \sum \beta_{i^4} x_i^4$	$2 + 4k + \binom{k}{2} + \binom{k}{3}$
5	$Y_5 = Y_4 + \sum \beta_{i^5} x_i^5$	$2 + 5k + \binom{k}{2} + \binom{k}{3}$
6	$Y_6 = Y_5 + \sum \beta_{i^6} x_i^6$	$2 + 6k + \binom{k}{2} + \binom{k}{3}$
7	$Y_7 = Y_6 + \sum \beta_{i^7} x_i^7$	$2 + 7k + \binom{k}{2} + \binom{k}{3}$

k is the number of design variables

0.9, the sample size is increased (by a multiple of k, for instance), and R^2_{adj} is computed again. During the iterations, each time that the number of samples is sufficient to evaluate the next higher-order regression model, see Table 1, also R^2_{adj} of that model is tested. This procedure is repeated until at least one regression model provides satisfactory results, or if for increasing regression-model order and increasing sample size the value of R^2_{adj} does not significantly improve. RBSA is then computed with the regression model having the best performance in terms of R^2_{adj}.

At first sight, this iterative approach may seem inefficient, due to the re-sampling of the design region. However, with particular care on the sampling technique, the samples taken in one iteration can be re-used also for the subsequent one. For instance, the sampling technique developed by Sobol', the LP_τ sequence, is a quasi-random sequence of numbers [20]. The LP_τ algorithm provides a sequence of sampling points for which it is known at any stage how successive points will fill in the gaps in the previously generated distribution [9]. That means a reuse of earlier points such that there is only a limited additional computational load. We therefore advise the reader to use such a sampling method for the iterative RBSA.

4 Validation of RBSA

In this section the methods for global SA mentioned in the introduction, including RBSA, are tested against five problems of increasing complexity, derived from [5]. The purpose is to evaluate the performance of RBSA in determining the sensitivity indices of the various factors, comparing it with the method of Sobol', FAST, the method of Morris, and the SRCs. The comparison is based on the number of model evaluations, indicated with N, needed to obtain the sensitivity indices, and their accuracy. For a given model, a smaller number of evaluations indicates that the computational time needed to obtain the sensitivity indices is lower. It is useful to remember that the evaluation of the model is considered the computationally expensive part of the analysis. The analysis of the data to perform SA is relatively fast in all cases presented here.

The main purpose of this comparison is to demonstrate that RBSA is able to successfully provide quantitative sensitivity indices (as the Sobol' and the FAST approach) with a low number of model evaluations (as most of the screening methods, such as the method of Morris). The method of Morris is executed with increasing levels (P) and increasing number of samples-per-level (R) for the same purpose. The values of the SRC are reported from the original study of Helton and Davis [5] for comparison with the results obtained with the other methods. The RBSA method is executed with models of increasing order and with an increasing number of sample points until a satisfactory level of R^2_{adj} is obtained, as discussed in Sect. 3. We stress the fact that the order of the model is not fixed a-priori, but rather determined automatically by the iterative process discussed in Sect. 3.5.

The sensitivity indices obtained with the method of Sobol', FAST and RBSA are only reported in terms of total-order sensitivity indices, S_{Ti}. The methods are executed on each problem with an increasing number of sample points to determine the minimum number of model evaluations that allows to stabilize the value of the sensitivity indices. By *stable* it is intended that the sensitivity indices do not change significantly for increasing sample size, i.e., they are constant to the second meaningful decimal digit. To obtain the sensitivity indices with the methods of Sobol', FAST, and Morris, the *Simlab* software suite was used [19]. In the comparison presented in this section, we consider the converged values from the method of Sobol' and FAST to be the correct results for the sensitivity indices. With RBSA, we try to obtain the same results, in a computationally cheaper way.

The first test problem considered is linear with only three uniformly distributed variables (Problem 1, [5]):

$$f(\mathbf{x}) = \sum_{i=1}^{3} x_i, \qquad \mathbf{x} = [x_1, x_2, x_3] \tag{27}$$

with $x_i : U(\bar{x_i} - \sigma_i, \bar{x_i} + \sigma_i), \bar{x_i} = 3^{i-1}, \sigma_i = 0.5\bar{x_i}$ for $i = 1, 2, 3$.

The results of the comparison are summarized in Table 2. Considering the low complexity of the problem, the method of Sobol' and FAST converge to a stable value of the sensitivity indices with a relatively large sample size, 1000 model evaluations, while the RBSA provides satisfactory results already after 20 model evaluations.

This is demonstrated with the graphs in Fig. 1. They show the trend of the sensitivity indices computed with the method of Sobol' (a), FAST (b), and RBSA (c), as a function of the number of model evaluations. The first two methods provide a definite distinction between the effects of the three factors already with few sample points but the values of the sensitivity indices are stable only after many more model evaluations. This effect will be more evident in the presence of more complex problems. Far less model evaluations are needed by the method of Morris to obtain a qualitative measure of sensitivity, i.e., the ranking of the factors

Table 2 Comparison of SA methods. Problem 1 [see Eq. (27)]

	Sobol' $N = 1000$	FAST $N = 1000$	Morris method $N = 8$			SRC[a] $N = 100$	RBSA[b] $N = 20$
Variable name	S_{Ti}	S_{Ti}	Rank	μ^*	σ	Value	S_{Ti}
x_1	0.011	0.014	3	9	0	0.105	0.013
x_2	0.099	0.101	2	3	0	0.316	0.097
x_3	0.892	0.890	1	1	0	0.946	0.890

[a]Standardized Regression Coefficients. Data adapted from [5]

[b]Linear regression model with 2-factors interaction terms. $R^2_{adj} = 1.00$

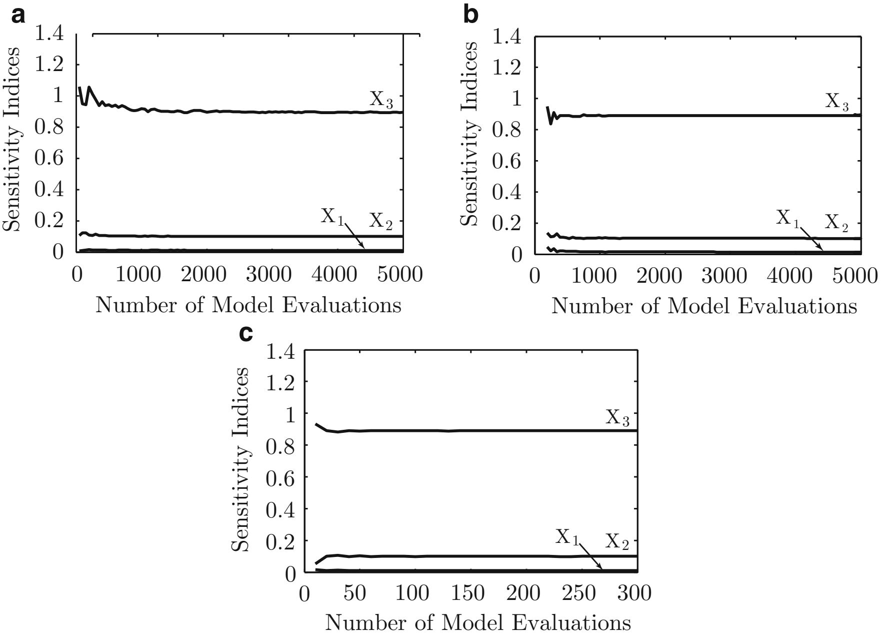


Fig. 1 Total-order sensitivity indices as a function of the sample size. Problem 1 [see Eq. (27)]. (**a**) Method of Sobol' . (**b**) FAST. (**c**) RBSA

according to their importance in the determination of $f(\mathbf{x})$. RBSA provides very precise quantitative sensitivity indices with a reduced computational effort when compared to the method of Sobol' and FAST. Indeed, only 20 model evaluations are required to obtain in practice the same results as the method of Sobol' and FAST. The SRCs provide a correct ranking of the relevance of the factors, but the sensitivity indices are much different from these provided by the other methods. Indeed, x_2 results to be much more important than it actually is.

The second test problem is again a linear one, but with a larger number, i.e., 22, of uniformly distributed variables (Problem 2, [5]):

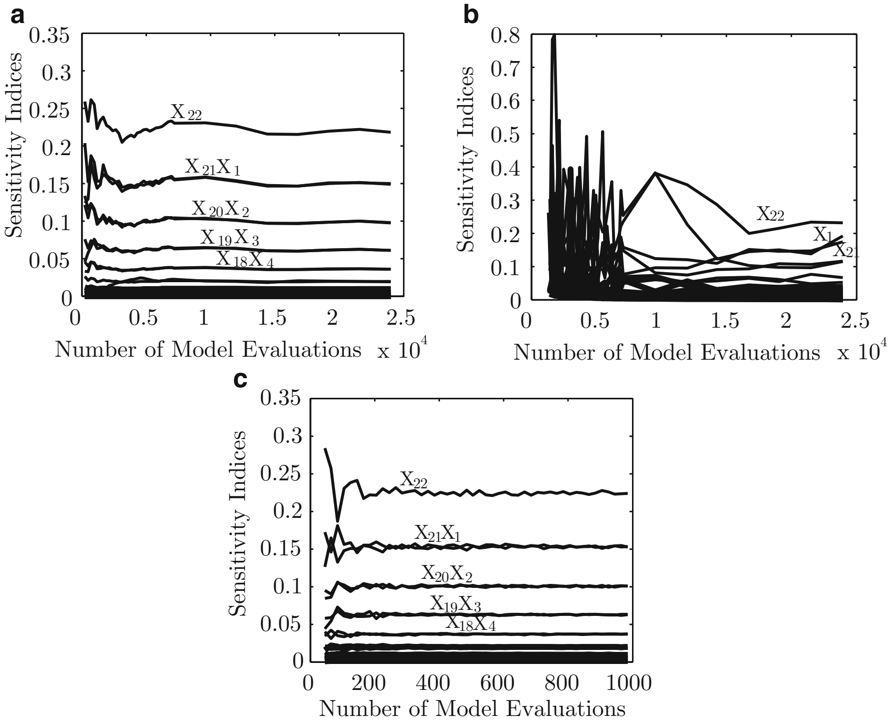


Fig. 2 Total-order sensitivity indices as a function of the sample size. Problem 2 [see Eq. (28)]. (**a**) Method of Sobol' . (**b**) FAST. (**c**) RBSA

$$f(\mathbf{x}) = \sum_{i=1}^{22} c_i\,(x_i - 0.5)\,, \qquad \mathbf{x} = [x_1, x_2, \cdots, x_{22}] \tag{28}$$

with $x_i : U(0, 1)$ and $c_i = (i - 11)^2$ for $i = 1, 2, \cdots, 22$.

The large number of variables of Problem 2 causes the method of Sobol' and FAST to converge to a stable value for the sensitivity indices only after 10,000 and 24,000 model evaluations, respectively. However, a clear distinction between the factors is already in place after 5000 samples in the case of the method of Sobol', see Fig. 2a.

The method of Morris provides excellent results in ranking the factors with a very low number of simulations. This is due to the fact that Problem 2 is linear, and a precise estimation of the variability of the data using the elementary effect is already possible with two sample points per variable. The RBSA provides very precise quantitative sensitivity indices, see Table 3, already after 600 model evaluations.

In both Problems 1 and 2 the method of Morris is able to correctly rank the factors and to correctly indicate the absence of interactions or non-linear effects (since the value of σ is zero for all factors). Also the SRCs provide a correct indication

Table 3 Comparison of SA methods. Problem 2 [see Eq. (28)]

	Sobol'	FAST	Morris method			SRC[a]	RBSA[b]
	$N = 10{,}000$	$N = 24{,}000$	$N = 46$			$N = 100$	$N = 600$
Variable name	S_{Ti}	S_{Ti}	Rank	μ^*	σ	Value	S_{Ti}
x_1	0.149	0.192	2	100	0	0.381	0.152
x_2	0.0979	0.115	3	81	0	0.308	0.100
x_3	0.0619	0.0660	4	64	0	0.243	0.0633
x_4	0.0369	0.0372	5	49	0	0.186	0.0369
x_5	0.0199	0.0212	6	36	0	0.136	0.0198
x_6	0.0093	0.0130	7	25	0	0.0951	0.0096
x_7	0.0038	0.0081	8	16	0	0.0608	0.0039
x_8	0.0012	0.0023	9	9	0	0.0342	0.0012
x_9	0.0002	0.0013	10	4	0	0.0152	0.0002
x_{10}	0	0.0011	11	1	0	0.0038	0
x_{11}	0	0.0011	12	0	0	0	0
x_{12}	0	0.0011	11	1	0	0.0038	0
x_{13}	0.0002	0.0012	10	4	0	0.0152	0.0002
x_{14}	0.0012	0.0026	9	9	0	0.0342	0.0012
x_{15}	0.0038	0.0059	8	16	0	0.0609	0.0039
x_{16}	0.0093	0.0076	7	25	0	0.0951	0.0096
x_{17}	0.0199	0.0160	6	36	0	0.136	0.0197
x_{18}	0.0367	0.0390	5	49	0	0.186	0.0371
x_{19}	0.0619	0.0520	4	64	0	0.243	0.0627
x_{20}	0.0980	0.116	3	81	0	0.307	0.100
x_{21}	0.149	0.174	2	100	0	0.380	0.153
x_{22}	0.218	0.232	1	121	0	0.460	0.224

[a] Standardized Regression Coefficients. Data adapted from [5]

[b] Linear regression model. $R^2_{adj} = 1.00$

on the relative importance of the factors but they do not provide any information on the presence (or not) of higher-order effects. The method of Sobol' and FAST provide quantitative sensitivity indices at the expenses of a large computational effort. The RBSA method provides very precise quantitative sensitivity indices, even in problems with a large number of variables, as Problem 2, at a computational cost that is much lower when compared to the method of Sobol' and FAST.

The third problem is monotonic, non-linear, with six uniformly distributed variables (Problem 3, [5]):

$$f(\mathbf{x}) = \exp\left(\sum_{i=1}^{6} b_i x_i\right) - \prod_{i=1}^{6} \frac{(e^{b_i} - 1)}{b_i}, \qquad \mathbf{x} = [x_1, x_2, \ldots, x_6], \tag{29}$$

with $x_i : U(0, 1)$ for $i = 1, 2, \ldots, 6$ and $b_1 = 1.5, b_2 = b_3 = \cdots = b_6 = 0.9$.

Table 4 Comparison of SA methods. Problem 3 [see Eq. (29)]

	Sobol'	FAST	Morris method			SRC[a]	RBSA[b]
	$N = 7500$	$N = 7500$	$N = 2870$			$N = 100$	$N = 500$
Variable name	S_{Ti}	S_{Ti}	Rank	μ^*	σ	Value	S_{Ti}
x_1	0.399	0.391	1	36.97	1.22	0.522	0.392
x_2	0.166	0.161	6	22.37	0.82	0.295	0.153
x_3	0.153	0.165	4	22.61	0.84	0.297	0.157
x_4	0.164	0.156	3	22.70	0.90	0.344	0.159
x_5	0.158	0.173	5	22.46	0.81	0.351	0.157
x_6	0.155	0.170	2	23.41	0.92	0.284	0.156

[a] Standardized Regression Coefficients. Data adapted from [5]

[b] Cubic regression model. $R^2_{adj} = 0.99$

The results of the comparison are shown in Table 4. The method of Sobol' and FAST converge to stable values for the sensitivity indices after 7500 model evaluations. The RBSA is able to account for almost all the variability of Problem 3 with a cubic regression model ($R^2_{adj} = 0.99$), and already with 500 sample points the estimation of the sensitivity indices is very precise, see Table 4.

In this case the factor x_1 is identified as the most relevant one already with $R = 10$, thus with a sample size of 70. The relative ranking of the factors x_2 to x_6 keeps changing with increasing R. For this reason it was decided to report only the results for $R = 410$ in Table 4.

In the case of non-linear monotonic problems the method of Sobol' and FAST provide as accurate results as in the linear case. The method of Morris has shown one potential weakness that arises when non-linear problems are taken into account: the results are very sensitive to the number of levels P and the number of sample points per level R. The SRCs perform well, even with non-linear monotonic problems. Linear approximations of the non-linear monotonic models provide a good indication of the general trends of the output, but this cannot be considered true in general. The RBSA demonstrates excellent performance also with this class of problems. Indeed, it provided very precise quantitative sensitivity indices at a relatively low computational cost.

The fourth problem is non-monotonic with eight uniformly distributed variables (Problem 4, [5]):

$$f(\mathbf{x}) = \prod_{i=1}^{8} \frac{|4x_i - 2| + a_i}{1 + a_i} \qquad \mathbf{x} = [x_1, x_2, \ldots, x_8], \tag{30}$$

with $x_i : U(0, 1)$ for $i = 1, 2, \ldots, 8$ and $a_1 = 0, a_2 = 1, a_3 = 4.5, a_4 = 9, a_5 = a_6 = a_7 = a_8 = 99$.

The results of the comparison are presented in Table 5. The first aspect worth mentioning is that the SRCs are not able to distinguish any of the variables effects. This is probably an expected result since the model of Problem 4 presents an

Table 5 Comparison of SA methods. Problem 4 [see Eq. (30)]

	Sobol'	FAST	Morris method			SRC[a]	RBSA[b]
	$N = 3000$	$N = 5000$	$N = 90{,}000$			$N = 100$	$N = 1000$
Variable name	S_{Ti}	S_{Ti}	Rank	μ^*	σ	Value	S_{Ti}
x_1	0.792	0.794	1	0.0060	0.0260	~ 0	0.704
x_2	0.244	0.239	2	0.0048	0.0151	~ 0	0.175
x_3	0.0338	0.0355	4	0.0014	0.0059	~ 0	0.0214
x_4	0.0104	0.0114	3	0.0028	0.0033	~ 0	0.0120
x_5	0.0001	0.0006	7	0.0003	0.0003	~ 0	0.0075
x_6	0.0001	0.0006	6	0.0004	0.0003	~ 0	0.0110
x_7	0.0001	0.0006	5	0.0007	0.0003	~ 0	0.0118
x_8	0.0001	0.0006	8	0.0000	0.0003	~ 0	0.0075

[a] Standardized Regression Coefficients. Data adapted from [5]

[b] Fifth-order regression model. $R^2_{adj} = 0.938$

absolute value, which causes the linear model to be deceived. The method of Sobol' and FAST provide stable results after 3000 and 5000 model evaluations, respectively. As already anticipated in the brief description of the methods, and as demonstrated in this test case, they do not suffer the highly non-linear behavior of the problem under investigation in the design region of interest. The polynomial regression models of the RBSA cannot perfectly cope with a functional like the absolute value, by definition. However, with a fifth-order model and 1000 sample points the RBSA can already account for almost 94 % of the variability of the data, providing a good quantitative distinction between the effects of the factors, and quantitative sensitivity indices that are close to the actual ones.

As reported in Table 5, the method of Morris presents the same type of problem encountered with the SRCs. However, a certain qualitative distinction between the factors' importance may still be identified. This is mainly due to the asymmetry of the absolute value of Eq. (30) in the variability interval determined by the variable ranges. The results are obtained with $R = 10{,}000$, thus a sample size of 90,000.

The last problem is non-monotonic with 3 uniformly distributed variables (Problem 5, [5]):

$$f(\mathbf{x}) = \sin x_1 + A \sin^2 x_2 + B x_3^4 \sin x_1 \qquad \mathbf{x} = [x_1, x_2, x_3]\,, \tag{31}$$

with $x_i : U(-\pi, \pi)$ for $i = 1, 2, 3$ and $A = 7, B = 0.1$.

Also in this case, the SRCs and the method of Morris are not able to detect the correct contribution of the factors to the variability of the performance, see Table 6. The method of Sobol' and FAST confirm the fact that the results they provide are not sensitive to the nature of the underlying model. Indeed in Table 6 it is shown that they provide a stable estimate of the sensitivity indices for 3000 and 5000 model evaluations, respectively. The RBSA, using a seventh-order model provided a coefficient of determination of 0.75. In this case this result is not as good as the

Table 6 Comparison of SA methods. Problem 5 [see Eq. (31)]

	Sobol' $N = 3000$	FAST $N = 5000$	Morris method $N = 40{,}000$			SRC[a] $N = 100$	RBSA[b] $N = 1000$
Variable name	S_{Ti}	S_{Ti}	Rank	μ^*	σ	Value	S_{Ti}
x_1	0.556	0.536	1	7.99	0.0988	~ 0	0.417
x_2	0.445	0.487	3	0.0055	0.0284	~ 0	0.330
x_3	0.237	0.242	2	0.1157	0.0781	~ 0	0.0054

[a] Standardized Regression Coefficients. Data adapted from [5]

[b] Seventh-order regression model. $R^2_{adj} = 0.75$

previous examples, leading to a misleading value for the absolute sensitivity index of the variable x_3. However, at least a correct ranking of the importance of the factors can be identified.

The Regression-Based Sensitivity Analysis method has shown good performance with different types of models. The sensitivity indices for linear and non-linear monotonic models can be precisely computed with a very reduced number of model evaluations, when compared to other methods. In the case of non-monotonic problems, the polynomial representation shows its limitations. RBSA provides less accurate quantitative results in these cases, but still it provides insight in the ranking the factors according to their importance, also when other qualitative methods fail. A polynomial function does not cope well with terms like $\sin x$, $\cos x$, e^x, and $\frac{1}{x}$, for instance. Therefore, it is hard to obtain a value for the coefficient of determination that is close *enough* to one. These non-polynomial terms could be included in the representation of the model of Eq. (5), but then the sensitivity indices would indicate the effect of the terms $\sin x$, $\cos x$, e^x, and $\frac{1}{x}$ rather than the effect of the factor x, which is what designers are usually interested in.

The method of Sobol' and the RBSA are in general valid for independent (i.e., non-correlated) input factors. The case with correlated inputs implies that the correlation structure must be taken into account during the sampling of the design space, leading to higher computational cost on one hand and to a non-direct applicability of the method on the other hand [17]. An effective technique for imposing the correlation between input variables has been proposed by [6].

The application of RBSA to a realistic, space-engineering related problem will be discussed in Sect. 6. There, we will study the design of the communication and power subsystems of a satellite.

5 Robust Design

So far we have looked at the sensitivity of a design's response to variations in the design parameters. In the related method, RBSA, we used a smart sampling method to reduce the number of sampling points when iteratively increasing the order of the regression model. A suitable and smart sampling method, however, does not only do

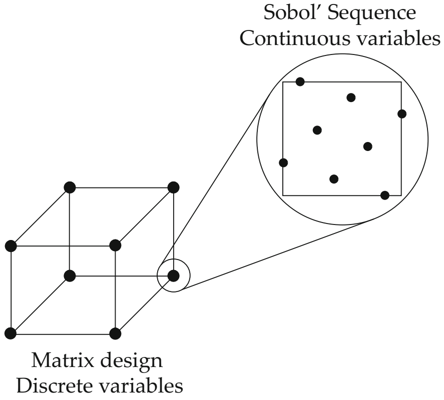


Fig. 3 Mixed-hypercube sampling with three discrete and two continuous variables

that, but also allows for varying different types of design parameters. Such a method is the mixed-hypercube approach that we can also efficiently apply to robust design.

The mixed-hypercube approach is a mixed sampling method that can take both continuous and discrete variables into account. In particular, with the mixed-hypercube approach we use both sampling for continuous variables and elements from Design of Experiments. The main idea is to separate the continuous and discrete variables into two groups. A matrix design is then created for the discrete variables, while for every row of the matrix design (i.e., for every design point of the factorial design), a Sobol' sequence is generated for the continuous variables. An example with three discrete and two continuous variables is presented in Fig. 3.

The advantage of using a matrix design instead of a space-filling technique for the discrete variables is that it allows to deterministically select the levels of the factors. When only few factor levels can be selected (e.g., in a database there is a certain number of batteries, or only a limited number of thrusters is considered in the analysis of a satellite system) the maximum number of simulations is determined by a full-factorial design. Then, depending on the type of analysis, and the available resources, one could choose for a fractional-factorial design. This will allow for the reduction of the computational effort while avoiding to disrupt the balance characteristics of the sampling matrix. The modification of a random or pseudo-random technique for sampling only at certain levels does not immediately provide such a balance, especially when the number of samples is kept low. On the other hand, in case of continuous variables matrix designs alone are less flexible in *filling* the design region and less suitable for the *re-sampling* process than the Sobol' technique.

The proposed mixed-hypercube sampling approach allows for covering the design region more uniformly when compared to other techniques already with a low number of samples, such as Latin hypercube sampling, factorial design, and orthogonal arrays. The sensitivity-analysis technique described in Sect. 3, will directly benefit from these characteristics, since convergence of the variance is obtained with a reduced computational effort, for instance. Another aspect of using specific implementations of the mixed-hypercube sampling method in combination with the design approaches discussed in this chapter is found in the area of robust design.

Robustness is a concept that can be seen from two different perspectives. One can define robustness of the system with respect to the effect of uncontrollable factors (aleatory and/or epistemic) and, if interested in obtaining a robust design, one can select that combination of controllable design-factor values that minimizes the variance while optimizing the performance. This concept is the most common way of thinking of robust design.

However, robustness can also be defined as the insensitivity of a certain design baseline to modification of the design variables in subsequent phases of the design process, thus providing an intrinsic design-baseline robustness figure. The modification of the levels of the design variables is likely to happen, especially when the baseline is at an early stage of the design process (phase 0/A). In this sense, robustness can be linked to the programmatic risk encountered when modifying a set of design parameters at later stages of the design process [15]. In the first case, instead, robustness is more related to the operational-life risk of the system (if the uncertainties derive from the operational environment, for instance).

In this section we introduce the Augmented Mixed Hypercube (AMH) as a mixed sampling techniques that takes into account continuous and discrete variables, where continuous variables can be deterministic (i.e., controllable) or probabilistic (i.e., uncontrollable). Discrete design factors are always considered deterministic here. For system design, discrete variables describe *architectures* of the system, and system architectures are fully controllable during the design.

The AMH is presented in Fig. 4 as an extension of the mixed hypercube shown in Fig. 3. In the AMH we take all types of design factors mentioned in this chapter into account. When the purpose of the analysis is to study the settings of controllable factors that are able to cope with the uncertainties introduced by the uncontrollable factors (stochastic and epistemic) then the AMH of Fig. 4a shall be used. There, for each combination of the levels of the controllable design variables, an uncertainty analysis can be executed using the unified sampling method to obtain the performance of the system, and the relative statistics, due to uncertain factors. When the purpose is only to propagate uncertainty into the model, then the AMH in the form presented in Fig. 4b shall be used instead. In the next section AMH shall be used for the robust design of the satellite's communication and power subsystems.

6 Test Case: Design of Communication and Power Subsystems of a Satellite

The main purpose of the discussion in this section is to better explain the utilization of the iterative RBSA method and to show, step-by-step, its implementation to

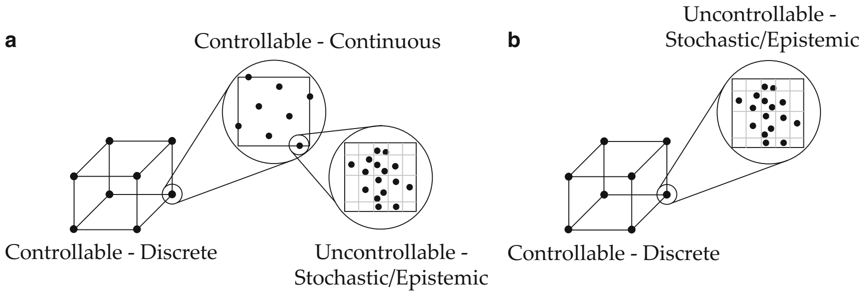

a
Controllable - Continuous
Controllable - Discrete
Uncontrollable - Stochastic/Epistemic
b
Uncontrollable - Stochastic/Epistemic
Controllable - Discrete

Fig. 4 Augmented Mixed Hypercube sampling procedure for robust design, (**a**) Study of controllable-factor settings. (**b**) Uncertainty propagation

Table 7 Settings of the design variables for the design of the communication and power subsystems

Design variables		Code	Intervals		Levels
			Min	Max	
Output RF power	[W]	A	1	50	–
Antenna diameter	[m]	B	0.05	1	–
Type of antenna	[–]	C	1	2	2
Type of solar array	[–]	D	1	3	3
Type of transmitter	[–]	E	1	2	2

compute the sensitivity indices. Moreover, in the second half, we will address the subsystems' robustness.

The model used for the analysis is intentionally focussed on the interactions existing between the communication and the power subsystem of a satellite. In particular, the model of the communication subsystem is used to estimate the uplink and the downlink budget between the satellite and the ground station, and its mass and power consumption. The model of the power subsystem, instead, is used to estimate the mass and power consumption of the power subsystem. The results presented in this section are obtained by using mathematical models available from Wertz and Larson [22], and Ridolfi et al. [10, 13].

In Fig. 5 we show a schematic, an N^2 chart, with the interactions between the communication and power subsystems. Besides links with the subsystem experts and with other subsystems and disciplines, there is one point of attention that is the loop created between the required power (from the communication subsystem) and the available power (from the power subsystem). This type of loops makes the design process iterative and correlates the performance of the two satellite subsystems.

We set up an analysis of the communication and power subsystems using five design variables, see Table 7, two performance indicators (namely, the down-link margin and the total mass of the two subsystems) and one constraint represented by the down-link margin itself demanded to be larger than 4 dB. More details on the settings of other design factors that influence the performance of these two subsystems are provided in the appendix.

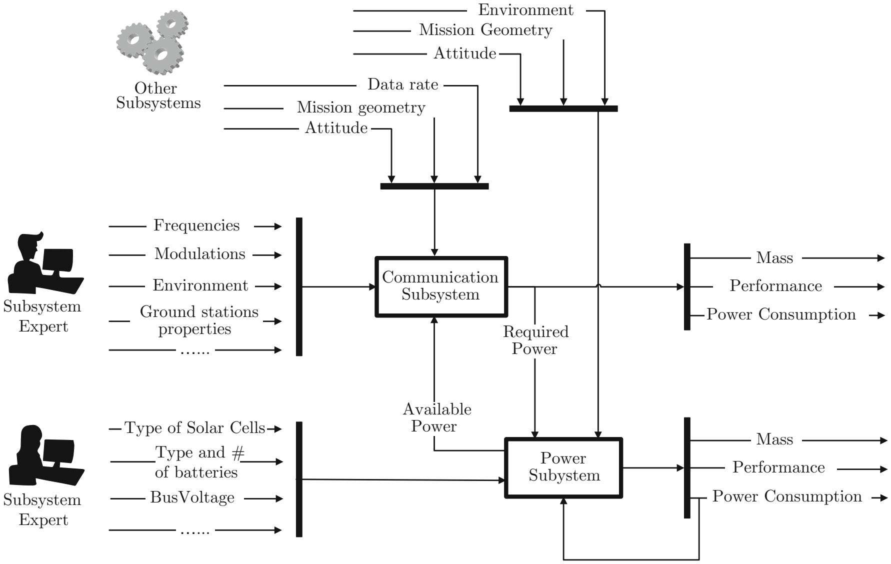


Fig. 5 Schematic representation of the interactions between the communication and power subsystems' models with the subsystem experts and other subsystems and disciplines

The three discrete variables give rise to a three-dimensional factorial design with 12 possible factor-level combinations in total. For each combination of discrete-variable levels the RBSA routine initially generates 7 (2+5, see Table 1) sample points using a Sobol' sequence. With seven sample points, a linear regression model is computed for both performances.

The coefficients of determination indicate that the subsystems mass is well represented by a linear relationship, $R^2_{adj} = 0.995$. The down-link margin, instead, could be better approximated using a higher-order model. Indeed in this case $R^2_{adj} = 0.920$. The decision whether to *re-sample* or continue with the RBSA shall be based on the value of R^2_{adj}. In this case a threshold of $R^2_{adj} = 0.95$ is used, which induces the iterative RBSA to add sample points to the analysis. A linear regression model with interaction terms is not sufficient to reach the threshold, which can be met only with a quadratic regression model. In this case the number of sample points is increased from 7 to 22 (2+10+10, see Table 1). The indications of Table 1 are only for the minimum number of sample points. The actual number of sample points to use, for each model order, is up to the user of RBSA.

The coefficients of determination ($R^2_{adj} = 0.998$ for the mass and $R^2_{adj} = 0.992$ for the down-link margin) confirm that a quadratic regression model is suitable for representing the variability of the performances in the design region of interest

With the same process described for the first combination of discrete design variables, the mathematical models of the communication and power subsystems are executed on the sample points for the other discrete-variable combinations. Then,

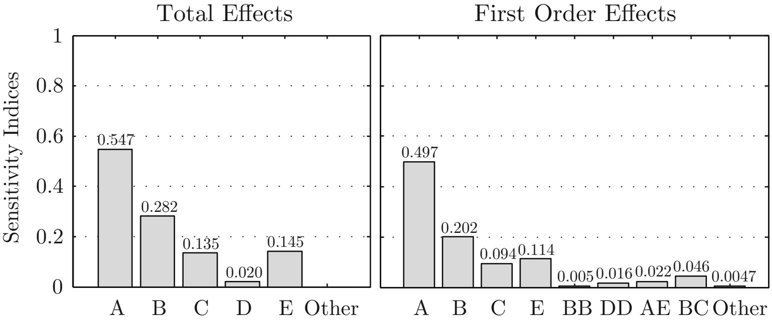


Fig. 6 Sensitivity indices obtained with the regression based sensitivity analysis. Sub-systems mass

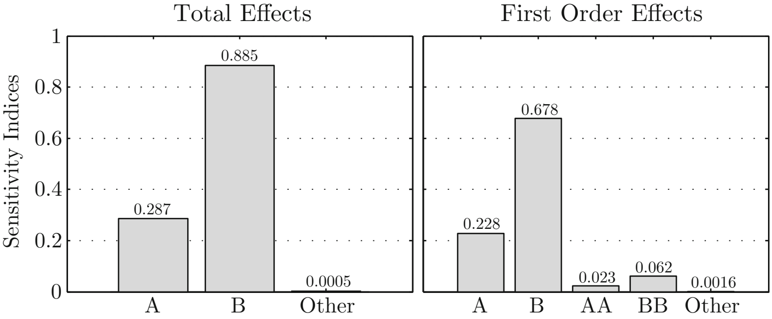


Fig. 7 Sensitivity indices obtained with the regression based sensitivity analysis. Down-link margin

with the RBSA the sensitivity indices can be estimated, using the relationship of Eqs. (23) and (24). In Figs. 6 and 7 the *total-order* and the *first-order* sensitivity indices for the subsystems' mass and down-link margin, respectively, are shown. The bars represent the sensitivity indices, i.e., the contribution of the factors indicated on the horizontal axis of the graphs, their interactions (when the product of two factors is indicated), and their quadratic effects (when the product of the factor by itself is indicated) to the variability of the performances. A sensitivity index equal to 0.2, for instance, indicates a contribution of that factor to the variance of the performance of interest equal to 20 %. The contribution of all other effects that are not explicitly shown in the bar plots, including the regression error, are encapsulated in the bars named *Other*.

The *Output RF Power* and the *Antenna Diameter* contribute for more than 50 % of the variability of the subsystems mass while they influence almost all the variability of the *down-link margin*. The *Type of Antenna* (C) and the *Type of Transmitter* (E) affect the mass of the subsystems because of their power density with respect to the aperture diameter and the output power. These interactions are evident in the bars of Fig. 6 named (BC) and (AE), respectively. The *Type of Solar*

Array (D) contributes to a limited extent to the variability of the mass of the two subsystems. This is due to the relatively low difference between the values of the power density of the solar cells selected from the data base for the analysis, see Table 9. Their contribution is mainly quadratic, correctly indicating that there is a minimum (in this case) of mass when the selected array is the one corresponding to the second level of the discrete design variable. The contribution of the *Type of Antenna* (C) to the *down-link margin* is very limited, and hidden in the *Other* bar of Fig. 7. For a given diameter, in the given frequency range, the aperture and horn antenna have similar performances in terms of gain, which lead to similar performances in terms of *down-link margin*. On the other hand, the influence on the *mass* of the *Type of Antenna* is more significant, with the aperture antenna being lighter than the horn one for a given reference diameter.

A quantitative indication of the importance of the factors for the determination of the performances, provides the engineering team with fundamental information to understand the effects of the design choices on the final design. In this case, for instance, one may easily conclude that the *Type of Solar Array* does not affect the performances much, thus it might be frozen to a particular type, based, for instance, on the availability at the moment of implementation, or its cost, or based on experience on past space missions. The *Type of Antenna* can be selected on the basis of its sole contribution to the mass (the aperture antenna minimizes the mass for a given *down-link margin* performance). This reduces *de facto* the dimensions of the design space allowing the design team to channel the effort on the more relevant design parameters. Very often the expert designers, or the developers of the mathematical model themselves, are already able to predict in advance the effects of the design choices on the output. However, this does not have to be the case, especially in the presence of less expert engineers or team members, or those who were not directly involved in the development of the mathematical model.

The next step in the analysis of the communication and power subsystems will be its robust design using the Augmented Mixed Hypercube approach. The above results obtained with the RBSA suggest that the linear graphs and contour plots that retain most of the variability of the performances are those presented in Fig. 8. As shown there, the trends corroborates the initial insight in the problem gained with the sensitivity analysis.

With these settings of the design variables, a confirmation experiment was performed on the model. The simulation provided a mass of the coupled subsystems of 160.2 kg and a down-link margin of 4.96 dB. The reason for performing a confirmation experiment is that the design point selected from the contour plot may not be very precise eventually due to the presence of lack-of-fit in the regression model. To get the results without the bias caused by the lack-of-fit, a confirmation experiment is needed.

The purpose of the analysis presented here is to draw some conclusions on the robustness to controllable and uncontrollable factors variations of the various architectures, using the AMH approach. A tabular representation of the AMH used for the analysis is presented in Table 8. The two continuous design variables are considered with a certain degree of uncertainty with respect to their baseline value. The other uncontrollable factors in Table 8 encompass many aspects related to the

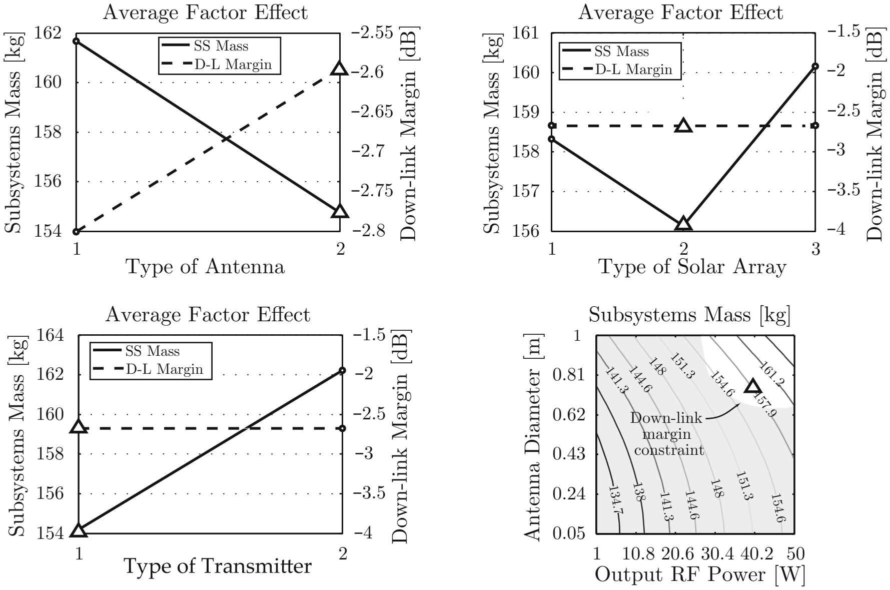


Fig. 8 Main results of the Communication and Power subsystems analysis. Δ is a tentative selected baseline. The *light-gray area* represents the down-link margin constraint-violation conditions

design and the operative life of the satellite for which there is uncertainty on one side, and the impossibility of controlling them directly on the other side. The results of the robust design on the Communication and Power subsystems, presented in Fig. 9, are computed using the AMH sampling procedure as shown in Fig. 4b.

In Fig. 9a, b the most robust and least robust configurations of the architectural variables are presented. In this case, the optimal configuration selected as a tentative baseline is also the most robust one (see the black probability density function). The least robust configuration, the one with the largest variance, is instead represented by the one having the *horn* antenna, the *triple junction* type of solar cell, and the *SSPA* type of transmitter. The sensitivity analysis presented in Fig. 9c, d reports the uncertain-factors contribution to these results. The *transmitter output-power* and the *transmission efficiencies* are the factors that influence most the sensitivity of the subsystem mass to the uncertainties (design and environmental). This means that the transmitter output power shall be carefully controlled in subsequent phases of the design process to maintain the *as-designed* performances. This also means that the margin that shall be applied to the subsystem mass is strongly dependent on the uncertainties that the engineering team has on the efficiencies with which the power is transmitted on board. Further, other sources of uncertainty will not affect the design much from the mass point-of-view. In Fig. 9a, the black vertical arrow represents the 20 % margin applied to the mean (nominal) value of the

Table 8 Settings of the design variables

Uncertain variables		Code	Intervals		Distribution
			Min	Max	
Output RF power	[W]	A	35	45	Uniform
Antenna diameter	[m]	B	0.75	0.85	Uniform
Satellite pointing error	[°]	C	1	4	Normal[d]
Implementation loss	[dB]	D	1	4	Epistemic[a]
Satellite antenna efficiency	[–]	E	0.45	0.55	Normal[d]
Antenna mass density	[Kg/m^2]	F	9	11.5	Log-Normal[e]
Ground antenna efficiency	[–]	G	0.45	0.55	Normal[d]
Ground antenna pointing error	[°]	H	0.1	1	Log-normal[e]
Transmission efficiency—sunlight	[–]	I	0.6	0.8	Epistemic[b]
Transmission efficiency—eclipse	[–]	J	0.6	0.8	Epistemic[c]
Solar cells η	[%]	K	Nom.[f]−10 %	Nom.[f]+10 %	Log-normal[e]
Solar array power dens.	[W/kg]	L	Nom.[f]−10 %	Nom.[f]+10 %	Log-normal[e]
Batteries energy dens.	[Wh/kg]	M	25	75	Log-normal[e]
Circular orbit altitude	[km]	N	990	1100	Normal[d]
Type of antenna	[–]		1	2	2 levels
Type of solar array	[–]		1	3	3 levels
Type of transmitter	[–]		1	2	2 levels

[a] Intervals [1, 1.75, 2.5, 3.25, 4], BPA [0.4, 0.25, 0.2, 0.15]
[b] Intervals [0.6, 0.667, 0.773, 0.8], BPA [0.25, 0.4, 0.35]
[c] Intervals [0.6, 0.667, 0.773, 0.8], BPA [0.25, 0.4, 0.35]
[d] $\mu = 0$ $\sigma = 1$, Min and Max are the 0.01 and 0.99 percentile respectively
[e] $\sigma = 1$, Max is the 0.99 percentile, Min corresponds to $X = 0$
[f] See nominal values in Table 9

subsystems mass. A classical margins-approach just providing the margin with respect to the mean value, will not convey any other kind of knowledge on the uncertainty structure and on the sensitivity with respect to the uncertain factors.

7 Conclusions

The Regression-Based Sensitivity Analysis (RBSA) proposed here uses general regression models obtained by adding higher-order terms (including interactions) to the standard linear model to minimize the lack-of-fit. However, it introduces a fundamental novel aspect by basing the computation of the sensitivity indices on the contribution to the variance of the various parameters, rather than simply relying on the regression coefficients, therefore providing global sensitivity information to the design team.

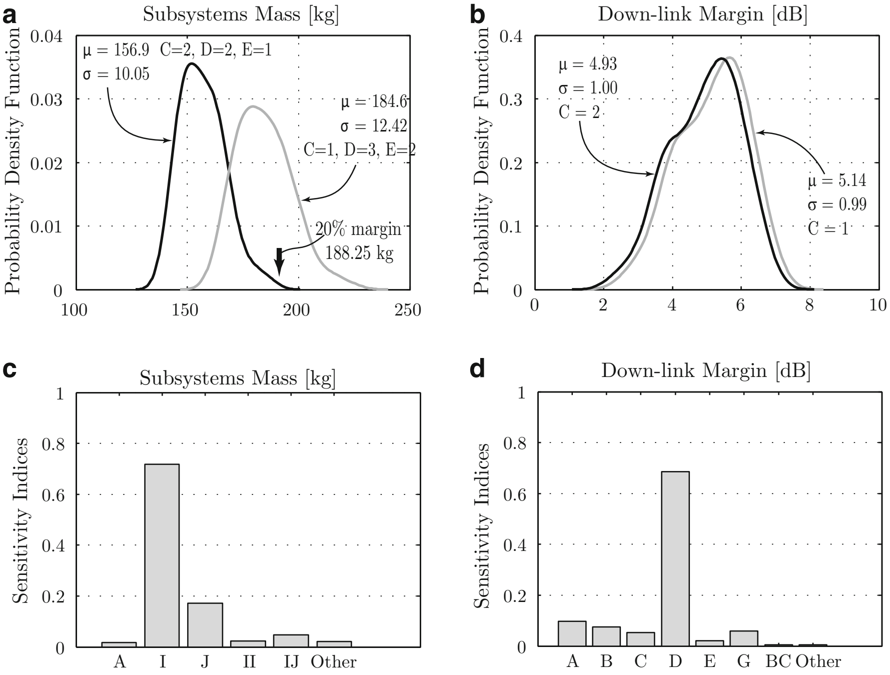


Fig. 9 Communication and power subsystems robust design and uncertainty analysis. Probability density function of the most robust (*black lines*) and least robust (*gray lines*) configuration on the (**a**) Subsystems mass, (**b**) Down-link margin. (**c**) Sensitivity analysis of the subsystems mass to the uncertain factors. (**d**) Sensitivity analysis of the down-link margin to the uncertain factors

The RBSA has been compared to other widely used approaches to global SA for designing purposes, and it demonstrated the characteristic of providing very precise quantitative information on the importance of the factors at a reduced computational effort in the case of linear and non-linear problems, even with a large number of variables. Further, it has been demonstrated the possibility of obtaining quantitative indices also of the single effects involving the design variables, information that is not available with the other SA methods. In case of highly non-linear and non-monotonic problems, the RBSA is able to provide at least a qualitative indication on the importance of the factors and their ranking, even when other qualitative screening methods fail. When designing a complex engineering system, with many variables to be taken into account, the RBSA could help in supporting the engineering team in quantitatively assess on the contribution of the design drivers, with a low computational cost and thus in a shorter time, which also means cost in most of the cases. This characteristic makes RBSA amongst the best candidates as a quantitative analysis technique to be used during design activities for the support of decision-making processes.

By combining the smart sampling method with an uncertainty analysis—the so-called augmented-hypercube sampling—one can successfully (and efficiently) generate response data to obtain the performance of the system, and the relative statistics, due to uncertain factors. This way the design can be fine-tuned to become least sensitive to disturbance sources that cannot be controlled by the designer.

One should realise that the applied augmented-hypercube sampling method is what we call a local method. Sometimes a single hypercube is sufficient to entirely cover the design space, sometimes instead a narrower hypercube might be needed to avoid major lack-of-fit conditions. In this case more than one hypercube may be implemented to study different regions of the design space as different alternative baselines of the system. In this case, the methodologies presented in this chapter will not only support the engineering team in selecting the best configuration for each single baseline, but will also allow to compare and trade between the baselines based on their performances, constraint-violation conditions and robustness.

Appendix: Communication and Power Subsystem

In Table 9 we describe the settings of the discrete variables used for the analysis of the communication and power subsystem. Further, the analysis of the communication and power subsystems cannot be performed considering them as separate from the other subsystems of the satellite and irrespectively of the orbit that the satellite will undergo. Some boundary conditions need to be set. In Table 10, the settings of all the parameters that significantly influence the performances of the communication and power subsystems are presented.

Table 9 Communication and power subsystem, discrete design variables settings

	Levels		
Type of antenna	Horn aperture		
Type of solar array	Silicon	GaAs	Triplejunction
	$\eta_{cell} = 0.148$	$\eta_{cell} = 0.24$	$\eta_{cell} = 0.20$
	$I_d = 0.77$	$I_d = 0.77$	$I_d = 0.60$
	$\eta_{degr/year} = 0.037$	$\eta_{degr/year} = 0.038$	$\eta_{degr/year} = 0.02$
	$\rho_{power} = 115\,\mathrm{W/kg}$	$\rho_{power} = 140\,\mathrm{W/kg}$	$\rho_{power} = 100\,\mathrm{W/kg}$
Type of transmitter	TWTA SSPA		

Data from [22]

Table 10 Communication system, settings of other factors influencing the performance

			Value
Mission and orbit	Orbit type	[–]	*Circular*
	Orbit altitude	[km]	1000
	Minimum elevation angle	[°]	30
	Mission duration	[years]	7
Attitude control	Antenna pointing offset	[°]	2
	Sun incidence angle	[°]	23
Payload	Average power consumption	[W]	160
Communication subsystem	Implementation losses	[dB]	2
	Ground antenna efficiency	[°]	0.55
	Ground antenna pointing offset	[°]	0.3
	Down-link frequency	[GHz]	2.2
	Down-link data rate	[Mbps]	100
Power subsystem	Transmission efficiency (sunlight)	[%]	71
	Transmission efficiency (eclipse)	[%]	62
	Solar flux	[W/m^2]	1367
	Batteries DoD	[%]	50
	Batteries energy density	[Wh/kg]	50

References

1. Campolongo, F., Cariboni, J., Saltelli, A.: An effective screening design for sensitivity analysis of large models. Environ. Model Softw. **22**(10), 1509–1518 (2007)
2. Cardile, D., Ridolfi, G., Mooij, E., Chiesa, S., Ferrari, G.: A system of systems approach for the concurrent design of space missions. In: IAC-10.B5.2.3, 61st International Astronautical Congress. Prague, CZ (2010)
3. Cukier, R., Levine, H., Shuler, K.: Nonlinear sensitivity analysis of multiparameter model systems. J. Comput. Phys. **1**(26), 1–42 (1978)
4. Draper, N., Smith, H.: Applied Regression Analysis, 3rd edn. Wiley, New York (1998)
5. Helton, J.C., Davis, F.J.: Illustration of sampling-based methods for uncertainty and sensitivity analysis. Risk Anal. **22**(3), 591–622 (2002)
6. Iman, R., Conover, W.: A distribution-free approach to inducing rank correlation among input variables. Commun. Stat. Simul. Comput. **3**(B11), 311–334 (1982)
7. Kuri, A., Cornell, J.: Response Surfaces. Design and Analyses. Marcel Dekker Inc., New York (1996). 2nd edn. Revised and Expanded
8. Morris, M.: Factorial sampling plans for preliminary computational experiments. Technometrics **33**(2), 161–174 (1991)
9. Press, W., Teukolsky, S., Vetterling, W., Flannery, B.: Numerical Recipes. The Art of Scientific Computing, 3rd edn. Cambridge University Press, New York (2007)
10. Ridolfi, G., Mooij, E., Corpino, S.: A system engineering tool for the design of satellite subsystems. In: Proceedings of the AIAA Modelling and Simulation Technologies Conference, Chicago. AIAA 2009-6037 (2009)
11. Ridolfi, G., Mooij, E., Cardile, D., Corpino, S., Stesina, F.: Orthogonal-array based design methodology for complex, coupled space systems. In: IAC-10.B5.2.4, 61st International Astronautical Congress, Prague (2010)

12. Ridolfi, G., Mooij, E., Corpino, S.: A methodology for system-of-systems design in support of the engineering team. Acta Astronaut. (2011). doi:10.1016/j.actaastro.2011.11.016
13. Ridolfi, G., Mooij, E., Chiesa, S.: A modelling framework for the concurrent design of complex space systems. In: Proceedings of the AIAA MST Conference Proceedings, Toronto AIAA 2010-7782 (2010)
14. Ridolfi, G., Mooij, E., Chiesa, S.: A parametric approach to the concurrent design of complex systems. In: ESA Workshop - SECESA, Lausanne, CH (2010)
15. Ridolfi, G., Mooij, E., Corpino, S.: Complex-systems design methodology for se collaborative environment. In: Systems Engineering. Theory and Applications (2012). ISBN:979-953-307-410-7
16. Saltelli, A., Tarantola, S., Chan, K.: A quantitative model-independent method for global sensitivity analysis of model output. Technometrics **41**(1), 39–56 (1999)
17. Saltelli, A., Tarantola, S., Campolongo, F., Ratto, M.: Sensitivity Analysis in Practice. Wiley, Chichester (2004)
18. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S.: Global Sensitivity Analysis. The Primer. Wiley, Chichester (2008)
19. Simlab: Software package for uncertainty and sensitivity analysis. Technical Report Last downloaded May 2010, Joint Research Centre of the European Commission (2010). Downloadable for free at: http://simlab.jrc.ec.europa.eu
20. Sobol', I.: On the systematic search in a hypercube. SIAM J. Numer. Anal. **16**(5), 790–793 (1979)
21. Sobol', I.M.: Sensitivity analysis for nonlinear mathematical models. Math. Model. Comput. Exp. **1**, 407–414 (1993)
22. Wertz, J., Larson, W.: Space Mission Analysis and Design, 3rd edn. Springer, New York (1999)
23. Williams, C.K.: Prediction with Gaussian processes: from linear regression to linear prediction and beyond. In: Learning in Graphical Models, pp. 599–612. MIT Press, Cambridge (1998)

Low-Thrust Multi-Revolution Orbit Transfers

Sven Schäff

Abstract This chapter presents a procedure to solve low-thrust orbit transfer with many orbital revolutions. One typical application is a transfer from the Geostationary Transfer Orbit (GTO) to the Geosynchronous Equatorial Orbit (GEO). Many telecommunication satellites to be located in the GEO ring are placed into an intermediate transfer orbit like the GTO. Just recently these spacecrafts are equipped more often with electric propulsion (EP) systems for the transfer. With respect to state of the art chemical apogee kick engine, EP provides just a small amount of thrust. As a result, the transfer needs many weeks or months and involves many orbital revolutions around the central body. The spacecraft has to be steered properly to match transfer constraints such as the final orbit. An approach is presented to solve this type of orbit transfers. After introducing the required spacecraft dynamics, several astrodynamical aspects like perturbations and environment conditions are highlighted. A direct collocation method is proposed to solve the optimal control problem. Furthermore few practical applications are shown to demonstrate the capabilities of the mentioned strategy.

Keywords Optimization models and methods • Astrodynamics • Low-thrust • Orbit transfer • Orbit-raising • Multi-revolution transfer • GTO-to-GEO • Trajectory optimization • Optimal control problem • Direct collocation • Nonlinear programming • Large-scale

1 Introduction

Telecommunication satellites located in the Geostationary Equatorial Orbit (GEO) are often injected in a Geostationary Transfer Orbit (GTO). Then they are transferred to the GEO using their own onboard propulsion system. State of the art for this type of transfer is the chemical propulsion. In the meantime few satellites consider Electric Propulsion (EP) for their orbit-raising, because it is very attractive to exploit the high specific impulse of EP technology to reduce the propellant consumption.

S. Schäff (✉)
Astos Solutions GmbH, Meitnerstrasse 8, Stuttgart, Germany
e-mail: sven.schaeff@astos.de

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization
and Its Applications 114, DOI 10.1007/978-3-319-41508-6_13

But electric orbit-raising requires much more complex maneuver sequences than what is needed for pure chemical transfers. And the optimization of such scenarios requires sophisticated modeling and optimization techniques. Since only small thrust magnitudes are provided, the transfer lasts many months. A careful planning of the spacecraft attitude maneuvers is required in advance to fulfill this mission. Unfortunately, most approaches lack the accuracy necessary to fully exploit all capabilities of electric orbit-raising. For example, the transfer trajectory has to avoid crossings of the GEO belt. Further, limitations and constraints related to different spacecraft subsystems, such as eclipse handling, and other possible limitations related to environmental aspects (e.g. radiation) have to be considered. For such sophisticated optimal problems it is very convenient to utilize direct transcription of the optimal control problem into a nonlinear programming (NLP) problem by discretization, because it results in a sparse problem solved in short time.

Section 2 introduces the required spacecraft dynamics and details the modeling of the environment and astrodynamics. Details about the proposed optimization procedure are presented in Sect. 3. Few examples of typical low-thrust multi-revolution transfers are given in Sect. 4. Finally, Sect. 5 concludes this chapter.

2 Modeling

The modelling of the low-thrust orbit transfer scenario is the first essential step for its computation and/or optimization. It encompasses the dynamic system, the perturbations acting on the spacecraft during its travel as well as some additional environmental effects.

Furthermore this section introduces the independent variable and controls required to setup the optimal control problem.

2.1 Dynamics

Any two objects of mass m and M with distance $\mathbf{r}$ apart are attracting each other. This is known as Newton's law of gravitation. Assuming the mass M is fixed in inertial space and $m \ll M$ Newton's law yields

$$\ddot{\mathbf{r}} = -\frac{\mu}{\|\mathbf{r}\|^3}\mathbf{r} \tag{1}$$

where μ is the standard gravitational parameter of mass M and $\ddot{\mathbf{r}}$ is the acceleration vector of mass m relative to the inertial frame. It represents the motion of mass m in the gravity field of mass M. Here, the first one is the mass of the satellite, while the latter one is the mass of the central body.

Since (1) is only true for two-body systems, a disturbing acceleration vector **a** is introduced

$$\ddot{\mathbf{r}} = -\frac{\mu}{\|\mathbf{r}\|^3}\mathbf{r} + \mathbf{a} \tag{2}$$

It is required to include disturbing accelerations which can be caused by thrust and/or other effects like gravitational perturbations and solar radiation pressure. Assuming the magnitude of the disturbing acceleration being small, like it is the case for low-thrust transfers, the equation describes the motion of a spacecraft subject to perturbations.

Next, the Cartesian state representation is transformed into a set of orbital elements. Each classical Keplerian orbital element

- semi-major axis a,
- eccentricity e,
- inclination i,
- argument of periapsis ω,
- longitude of ascending node Ω and
- true anomaly ν

characterizes a physical property of the orbit shape (a, e), its orientation (i, ω, Ω) and the position of the body orbiting the central body (ν). Indeed, this is very helpful to understand the variation of orbital changes.

However, these classical orbital elements suffer from two singularities. First, with circular orbits, which have eccentricity of zero, the line of apsis is undefined. And second, with an equatorial orbit, which means an inclination of either 0° (prograde) or 180° (retrograde), both the ascending and descending nodes are ill-defined. Therefore a new set of orbital elements which eliminates these deficiencies is required: the equinoctial elements.

They are also the better choice for the optimization since the results are more precisely, the time needed for the optimization is less and the convergence of the computation is better toward the Keplerian elements [1, 2]. Unfortunately, the set of classical equinoctial elements does not accommodate orbits with e $\geq$ 1. To eliminate this deficiency, a set of modified equinoctial elements is used as proposed in [1].

Equinoctial element p is the semi-latus rectum of the orbit and is related to the semi-major axis and eccentricity. The modified equinoctial elements f and g represent the eccentricity vector, and h and k represent the inclination vector. Equinoctial element L is the true longitude of the spacecraft. All six elements describe the position and velocity of the spacecraft expressed as state vector **x**. The relationship between the modified equinoctial elements and the Keplerian orbital elements is given in the following equations:

$$p = a\left(1 - e^2\right) \tag{3}$$

$$f = e\cos(\omega + \Omega) \tag{4}$$

$$g = e\sin(\omega + \Omega) \tag{5}$$

$$h = \tan\left(\frac{i}{2}\right)\cos\Omega \tag{6}$$

$$k = \tan\left(\frac{i}{2}\right)\sin\Omega \tag{7}$$

$$L = \Omega + \omega + \nu \tag{8}$$

Using the inverse transformation the Keplerian orbital elements are defined by:

$$a = \frac{p}{(1 - f^2 - g^2)} \tag{9}$$

$$e = \sqrt{f^2 + g^2} \tag{10}$$

$$i = 2\tan^{-1}\sqrt{h^2 + k^2} \tag{11}$$

$$\omega = \tan^{-1}\frac{g}{f} - \tan^{-1}\frac{k}{h} \tag{12}$$

$$\Omega = \tan^{-1}\frac{k}{h} \tag{13}$$

$$\nu = L - \tan^{-1}\frac{g}{f} \tag{14}$$

To obtain the inertial Cartesian coordinates for position **r** and velocity **v** the following equations can be used:

$$\mathbf{r} = \begin{bmatrix} \frac{r}{s^2}\left(2hk\sin L + \left(1 + \alpha^2\right)\cos L\right) \\ \frac{r}{s^2}\left(\left(1 - \alpha^2\right)\sin L + 2hk\cos L\right) \\ \frac{2r}{s^2}\left(h\sin L - k\cos L\right) \end{bmatrix} \tag{15}$$

$$\mathbf{v} = \begin{bmatrix} -\frac{1}{s^2}\sqrt{\frac{\mu}{p}}\left(\left(1 + \alpha^2\right)(\sin L + g) - 2hk(\cos L + f)\right) \\ -\frac{1}{s^2}\sqrt{\frac{\mu}{p}}\left(2hk(\sin L + g) + \left(\alpha^2 - 1\right)(\cos L + f)\right) \\ \frac{2}{s^2}\sqrt{\frac{\mu}{p}}\left(k(\sin L + g) + h(\cos L + f)\right) \end{bmatrix}, \tag{16}$$

where

$$w = 1 + f\cos L + g\sin L \tag{17}$$

$$r = \frac{p}{w} \tag{18}$$

$$\propto^2 = h^2 - k^2 \tag{19}$$

$$s^2 = 1 + h^2 + k^2. \tag{20}$$

2.2 Equations of Motion

When using the modified equinoctial element, it is convenient to express the disturbing acceleration vector a in a rotating frame. Its origin is the center of mass of the spacecraft. This reference frame is defined with respect to the inertial frame by the following principal axes:

$$i_r = \frac{\mathbf{r}}{\|\mathbf{r}\|} \tag{21}$$

$$i_t = \frac{(\mathbf{r} \times \mathbf{v}) \times \mathbf{r}}{\|(\mathbf{r} \times \mathbf{v}) \times \mathbf{r}\|} \tag{22}$$

$$i_n = \frac{\mathbf{r} \times \mathbf{v}}{\|\mathbf{r} \times \mathbf{v}\|} \tag{23}$$

Index r indicates the radial component, t is the transverse (along-track) component and n is the normal (cross-track) component. The r-axis points in the same direction as the position vector and the t-axis lies in the orbital plane pointing in direction of flight. Both axes span the orbital plane. Note that the t-axis is not necessarily parallel to the velocity vector. The out-of-orbit-plane axis n is perpendicular to the orbital plane and points in the direction of the angular momentum. This coordinate frame is also called RTN or LVLH (local vertical, local horizontal). In [3] it is known as RSW.

The corresponding transformation matrix given by

$$\mathbf{R} = \begin{bmatrix} i_r \; i_t \; i_n \end{bmatrix} \tag{24}$$

describes the transformation from the rotating RTN-frame to the inertial frame, i.e. the International Celestial Reference System (ICRS).

Next, the disturbing acceleration vector in the rotating RTN-frame is introduced:

$$\Delta = \mathbf{R}^{\mathrm{T}}\mathbf{a} \tag{25}$$

With the acceleration vector components defined in the rotating frame, the equinoctial dynamics have to be defined. These are the state derivatives which are integrated over the independent variable. For the states describing the position and velocity of the spacecraft these are simply the derivatives of the modified equinoctial elements (see [2]):

$$\dot{p} = \sqrt{\frac{p}{\mu}}\Delta_t \frac{2p}{w} \tag{26}$$

$$\dot{f} = \sqrt{\frac{p}{\mu}}\left\{\Delta_r \sin L + \Delta_t \frac{1}{w}\left[(w+1)\cos L + f\right] - \Delta_n \frac{g}{w}\left[h \sin L - k \cos L\right]\right\} \tag{27}$$

$$\dot{g} = \sqrt{\frac{p}{\mu}}\left\{-\Delta_r \cos L + \Delta_t \frac{1}{w}\left[(w+1)\sin L + g\right] + \Delta_n \frac{f}{w}\left[h \sin L - k \cos L\right]\right\} \tag{28}$$

$$\dot{h} = \sqrt{\frac{p}{\mu}}\Delta_n \frac{s^2}{2w}\cos L \tag{29}$$

$$\dot{k} = \sqrt{\frac{p}{\mu}}\Delta_n \frac{s^2}{2w}\sin L \tag{30}$$

$$\dot{L} = \sqrt{p\mu}\left(\frac{w}{p}\right)^2 + \sqrt{\frac{p}{\mu}}\Delta_n \frac{1}{w}\left(h \sin L - k \cos L\right) \tag{31}$$

In case the disturbing acceleration is zero ($\Delta = 0$), all equations of motion except the one of the true longitude ($\dot{L}$) become zero:

$$\dot{p} = \dot{f} = \dot{g} = \dot{h} = \dot{k} = 0 \tag{32}$$

It implies these modified equinoctial elements are simply constant, while the equation of motion related to the true longitude describes the movement of the point mass according to the two-body equation:

$$\dot{L} = \sqrt{p\mu}\left(\frac{w}{p}\right)^2 \tag{33}$$

2.3 Perturbations

Also impacting the equations of motion are all perturbing forces acting on the spacecraft during its transfer in the gravity well of the central body. Mainly it is the thrust acceleration of the on-board propulsion system. Besides, available perturbations include gravitational accelerations and non-conservative perturbations, among others. All perturbing forces are summed up and the resulting disturbing acceleration vector is given by

$$\Delta = \Delta_T + \Delta_G + \Delta_{3rd} + \Delta_D + \Delta_{SRP} + \Delta_{SW} \tag{34}$$

where acceleration $\varDelta_T$ is due to thrust, $\varDelta_G$ is due to central body gravity perturbations, $\varDelta_{3rd}$ is due to third body gravity perturbations, $\varDelta_D$ is due to atmospheric drag, $\varDelta_{SRP}$ is due to solar radiation pressure and $\varDelta_{SW}$ is due to solar wind.

2.3.1 Thrust

In case of a low-thrust orbit transfer the motion of a spacecraft is mainly affected by its propulsion system. In contrast to natural forces it is active controlled and long propulsion periods, also known as thrust arcs, are one of the characteristics of low-thrust transfers.

Since the thrust acceleration vector can be defined in the rotating RTN frame, it is given by

$$\mathbf{\Delta}_T = \frac{T}{m}\mathbf{a}_T \tag{35}$$

where T is the thrust magnitude of the propulsion system, m is the mass of the spacecraft and $\mathbf{a}_T$ is the vector defining the direction of the thrust magnitude in the rotating RTN frame according to

$$\mathbf{a}_T = \begin{pmatrix} a_r \\ a_t \\ a_n \end{pmatrix} \tag{36}$$

It is a time-varying vector in its Cartesian representation and has to have unit length

$$\|\mathbf{a}_T\| = \sqrt{a_r^2 + a_t^2 + a_n^2} = 1 \tag{37}$$

Further, the mass flow rate is defined as

$$\dot{m} = -\frac{T}{v_{ex}} \tag{38}$$

where v_{ex} is the exhaust velocity given by

$$v_{ex} = g_0 I_{sp} \tag{39}$$

where g_0 is the standard acceleration due to gravity at Earth's surface and I_{sp} is the specific impulse. The minus sign of the mass flow rate describes the loss of propellant. Integrating the mass flow rate yields the consumed propellant mass. The higher the specific impulse, the less propellant needed for a given delta-v increment.

2.3.2 Central Body Gravity

For a point mass, the potential function of the gravity is

$$\Phi = \frac{\mu}{r} \tag{40}$$

where μ is the gravitational parameter of the central body and r is the distance of the spacecraft to the center of the central body. It indicates that the gravity potential at a certain point is directly proportional to the mass of the center body and inversely proportional to the distance of the spacecraft to the central body.

In case of a spheroidal central body, the potential of the gravity has to be integrated. Chobotov [4] shows that

$$\Phi = \frac{\mu}{r}\sum_{n=0}^{\infty}\sum_{q=0}^{n}\left(\frac{R_0}{r}\right)^n P_n^q\left(\sin\varphi\right)\left(C_{n,q}\cos q\lambda + S_{n,q}\sin q\lambda\right) \tag{41}$$

where R_0 is the equatorial radius of the central body, φ is the planeto-centric latitude (also known as declination), λ is the longitude towards east, $C_{n,q}$ and $S_{n,q}$, are the coefficients of the potential of degree n and order of m and P_n are the Legendre polynomials.

Considering a spheroid of revolution with $q = 0$ yields the potential function

$$\Phi = -\frac{\mu}{r}\left(1 - \sum_{n=2}^{\infty} J_n \left(\frac{R_0}{r}\right)^n P_n\left(\sin\varphi\right)\right) \tag{42}$$

where J_n are the dimensionless potential coefficients ($J_n = -C_{n,0}$). They are also known as zonal harmonic coefficients and only depend on the latitude. Another representation of the potential function is

$$\Phi = -\frac{\mu}{r} + B\left(r, \lambda, \varphi\right) \tag{43}$$

with the gravity perturbation

$$B\left(r, \lambda, \varphi\right) = \frac{\mu}{r}\sum_{n=2}^{\infty} J_n \left(\frac{R_0}{r}\right)^n P_n\left(\sin\varphi\right) \tag{44}$$

The gravitational disturbing acceleration vector is given by

$$\mathbf{a}_G = -\nabla B\left(r, \lambda, \varphi\right) = -\begin{pmatrix} \frac{\partial B}{\partial r} \\ \frac{1}{r\sin\varphi}\frac{\partial B}{\partial \lambda} \\ \frac{1}{r}\frac{\partial B}{\partial \varphi} \end{pmatrix} \tag{45}$$

Using (44) and (45) yields

$$\mathbf{a}_G = -\begin{pmatrix} \frac{\mu}{r^2}\sum\limits_{n=2}^{\infty}\left(n+1\right) J_n \left(\frac{R_0}{r}\right)^n P_n\left(\sin\varphi\right) \\ 0 \\ \frac{\mu}{r^2}\cos\varphi \sum\limits_{n=2}^{\infty} J_n \left(\frac{R_0}{r}\right)^n P_n^{'}\left(\sin\varphi\right) \end{pmatrix} \tag{46}$$

where $P_n^{'}(\sin\varphi)$ is the derivative of the n-th order Legendre polynomial. Since only zonal harmonics are considered here there is no longitudinal acceleration. In general, the Legendre polynomials of order n and degree of m are given by

$$P_{nm}(x) = \frac{\left(1-x^2\right)^{\frac{m}{2}}}{2^n n!} \cdot \frac{d^{(n+m)}\left(x^2-1\right)^n}{dx^{(n+m)}} \tag{47}$$

For convenience, the Legendre polynomials up to order 6 and degree 0 are

$$P_2 = \frac{3}{2}\sin^2\lambda - \frac{1}{2} \tag{48}$$

$$P_3 = \frac{5}{2}\sin^3\lambda - \frac{3}{2}\sin\lambda \tag{49}$$

$$P_4 = \frac{35}{8}\sin^4\lambda - \frac{15}{4}\sin^2\lambda + \frac{3}{8} \tag{50}$$

$$P_5 = \frac{63}{8}\sin^5\lambda - \frac{35}{4}\sin^3\lambda + \frac{15}{8}\sin\lambda \tag{51}$$

$$P_6 = \frac{231}{16}\sin^6\lambda - \frac{315}{16}\sin^4\lambda + \frac{105}{16}\sin^2\lambda + \frac{5}{16} \tag{52}$$

and their corresponding derivatives are

$$P_2^{'} = 3\sin\lambda \tag{53}$$

$$P_3^{'} = \frac{15}{2}\sin^2\lambda - \frac{3}{2} \tag{54}$$

$$P_4^{'} = \frac{35}{2}\sin^3\lambda - \frac{15}{2}\sin\lambda \tag{55}$$

$$P_5^{'} = \frac{315}{8}\sin^4\lambda - \frac{105}{4}\sin^2\lambda + \frac{15}{8} \tag{56}$$

$$P_6^{'} = \frac{693}{8}\sin^5\lambda - \frac{315}{4}\sin^3\lambda + \frac{105}{8}\sin\lambda \tag{57}$$

Finally, the acceleration vector needs to be transformed from the local reference frame to the rotating RTN frame given as

$$\mathbf{\Delta}_G = \mathbf{R}_L^{\mathrm{T}}\mathbf{a}_G \tag{58}$$

where $\mathbf{R}_L$ is the corresponding transformation matrix from the rotating RTN-frame to the local reference frame (in case of Earth it is the International Terrestrial Reference System). Oblateness of the central body is one of major perturbations for low-thrust trajectories. Most impact can be observed in the longitude of the ascending node Ω and the argument of periapsis ω.

2.3.3 Third Body Gravity

Perturbing accelerations caused by third bodies have to be considered for low-thrust transfers as well. In some situations this acceleration might be even larger than the thrust acceleration of the spacecraft itself. To consider perturbations of the third bodies the following formulation is used:

$$\mathbf{a}_{3rd} = -\sum_k \mu_k \left(\frac{\mathbf{d}_k}{\|\mathbf{d}_k\|^3} + \frac{\mathbf{s}_k}{\|\mathbf{s}_k\|^3} \right) \tag{59}$$

where μ_k is the standard gravitational parameter of the kth third body, $\mathbf{d}_k$ is the vector from the kth third body to the spacecraft and $\mathbf{s}_k$ is the vector from the central body of the spacecraft to the kth third body. It follows that

$$\mathbf{d}_k = \mathbf{r} - \mathbf{s}_k \tag{60}$$

Equation (59) can be used to compute the gravitational perturbation vector. However, Battin [5] suggests using the following

$$f(q_k) = \frac{\mathbf{d}_k}{\mathbf{s}_k} - 1 = q_k \left(\frac{3 + 3q_k + q_k^2}{1 + (1 + q_k)^{3/2}} \right) \tag{61}$$

where

$$q_k = \frac{\mathbf{r}^{\mathrm{T}} (\mathbf{r} - 2\mathbf{s}_k)}{\mathbf{s}_k^{\mathrm{T}} \mathbf{s}_k} \tag{62}$$

Finally (59), (60) and (61) yield

$$\mathbf{a}_{3rd} = -\sum_k \frac{\mu_k}{\|\mathbf{d}_k\|^3} (\mathbf{r} + f(q_k)\, \mathbf{s}_k) \tag{63}$$

Since this acceleration vector is given in the inertial frame, the following transformation is used to obtain it in the RTN frame

$$\boldsymbol{\Delta}_{3rd} = \mathbf{R}^{\mathrm{T}} \mathbf{a}_{3rd} \tag{64}$$

where **R** is the transformation matrix from the RTN frame to the inertial frame.

2.3.4 Atmospheric Drag

Atmospheric drag mostly influences the spacecraft trajectory in low altitudes. This effect can be dominant with respect to other perturbations like central body oblateness. On the other side, in higher altitudes (i.e. more than 1000 km) the atmospheric drag becomes very small and is dominated by solar radiation pressure and third body perturbations. The atmospheric drag is caused by the particles of the atmosphere and depends on its density as well as the velocity of the spacecraft with respect to the atmosphere.

The perturbing force vector induced by the atmospheric drag is defined by

$$\mathbf{F}_D = -pC_D A \frac{\mathbf{v}_{rel}}{\|\mathbf{v}_{rel}\|} \tag{65}$$

where p is the atmospheric pressure, C_D is the drag coefficient of the spacecraft, A is the cross-sectional area of the spacecraft and $\mathbf{v}_{rel}$ is the velocity vector of the spacecraft relative to the atmosphere. The atmospheric pressure is obtained from the Bernoulli equation

$$p = \frac{1}{2}\rho\|\mathbf{v}_{rel}\|^2 \tag{66}$$

where ρ is the atmospheric density. Further, the velocity vector of the spacecraft relative to the rotating atmosphere is defined by

$$\mathbf{v}_{rel} = \mathbf{v} - \boldsymbol{\omega} \times \mathbf{r} \tag{67}$$

where $\boldsymbol{\omega}$ is the spin vector of the central body. The atmospheric density is quite difficult to be determined since the density of the upper atmosphere changes due to the following:

- molecular composition
- solar flux
- interactions with the magnetic field

There are many factors affecting the density of the atmosphere, such as variations in the position (longitude, latitude) and cyclic variations (diurnal, solar rotation cycle, solar cycle, etc.) [3]. One of the most precise atmosphere models for Earth is the Jacchia-Bowman 2008 model [6]. Much faster in computation, but less accurate, is an exponential model. Such a basic model varies the density of the atmosphere according to

$$\rho = \rho_0 e^{\left(-\frac{h-h_0}{H}\right)} \tag{68}$$

where ρ_0 is the reference density specified at the reference altitude h_0, h is the actual altitude of the spacecraft and H is the scale height. Quite moderate results are achieved with the exponential atmosphere model suggested in [3]. It uses the U.S. Standard Atmosphere (1976) and CIRA-72 including exospheric temperatures.

Finally, the resulting acceleration vector is the atmospheric drag force over mass, and with equations (65) and (66) it follows that

$$\mathbf{a}_D = -\frac{1}{2}\rho\frac{C_D A}{m}\|\mathbf{v}_{rel}\|^2\frac{\mathbf{v}_{rel}}{\|\mathbf{v}_{rel}\|} \tag{69}$$

Transforming it to the rotating RTN frame yields

$$\boldsymbol{\Delta}_D = \mathbf{R}^{\mathrm{T}}\mathbf{a}_D \tag{70}$$

where $\mathbf{R}$ is the transformation matrix.

2.3.5 Solar Radiation Pressure

Due to nuclear fusion reaction the Sun continuously emits radiant energy, called solar radiation. It includes all electromagnetic waves emitted by the Sun and the electromagnetic energy has a spectrum close to that of a black body with a temperature of 5800 K. The solar radiation pressure (SRP) is the pressure exerted by the solar radiation on objects within its reach, i.e. spacecrafts. Its effect is strongest for light objects with large reference areas. If the solar photons are completely absorbed the SRP is defined as

$$p_{SRP} = \frac{S}{c} \tag{71}$$

where S is the solar radiation flux density and c is the speed of light in vacuum. Because the radiation flux of the Sun, respectively a point light source, is proportional to the inverse square of distance, the solar radiation pressure becomes

$$p_{SRP} = \frac{S_0}{c}\left(\frac{r_0}{\|\mathbf{r}_S\|}\right)^2 \tag{72}$$

Where S_0 is the solar constant at a distance of 1 astronomical unit (AU), r_0 is the distance between Sun and Earth defined as 1 AU and $\mathbf{r}_s$ is the vector from the spacecraft to the Sun. The solar constant is not a physical constant. It varies by about 0.1 % over each solar cycle (sunspot cycle) of about 11 years, discovered by astronomer Schwabe.

The perturbing force vector exerted due to solar radiation pressure is defined by

$$\mathbf{F}_{SRP} = -p_{SRP} C_R A \frac{\mathbf{r}_S}{\|\mathbf{r}_S\|} \tag{73}$$

where C_R is the coefficient of reflectivity of the spacecraft and A is its reference area. Valid values for C_R are between 0 and 2 where C_R equals 1 for a perfect absorber (black body) and in case of a perfect reflector it equals 2. Translucent materials have a value between 0 and 1. For typical SEP spacecrafts a value of about 1.3 is used [2].

Since the acceleration is defined by force over mass, and with (72) and (73), it yields

$$\mathbf{a}_{SRP} = -C_R \frac{AS_0}{mc} r_0^2 \frac{\mathbf{r}_S}{\|\mathbf{r}_S\|^3} \tag{74}$$

where m is the mass of the spacecraft. To obtain the acceleration in the rotating RTN frame, it follows that

$$\mathbf{\Delta}_{SRP} = \mathbf{R}^{\mathrm{T}} \mathbf{a}_{SRP} \tag{75}$$

where $\mathbf{R}$ is the transformation matrix.

2.3.6 Solar Wind

The solar wind was postulated to explain the bending of a comets plasma tail and the auroras. It is a stream of charged particles, also known as plasma, expelled from the Sun and consists mainly of protons, high-energy electrons, alpha particles and heavy ions. These charged particles may also affect sensor performance of the spacecraft and/or ground-spacecraft communication. With dominating hydrogen and helium the composition of the solar wind is identical to the corona of the Sun.

Two different kinds of the solar wind exist: a fast and a slow solar wind. In the plane of the ecliptic, and therefore also next to the planets, the solar wind is slower and denser. The typical speed is between 200 and 600 km/s with an average value of 400 km/s [7]. It is also notable that daily fluctuations by a factor of two exist. Outside the ecliptic plane the solar wind is faster with typical speeds between 600 and 800 km/s. Both the slow and the fast solar wind can be interrupted by interplanetary coronal mass ejections, known as solar storms. When arriving at Earth they temporarily deform its magnetic field.

Any impact of energetic particles from the solar wind on a spacecraft exerts a force which can be compared to that from the atmospheric drag, since both are typical drag forces. The drag force induced by the solar wind is given by

$$\mathbf{F}_{SW} = -p_{SW} C_D A \frac{\mathbf{r}_S}{\|\mathbf{r}_S\|} \tag{76}$$

where p_{SW} is the pressure exerted by the solar wind, C_D is the drag coefficient of the spacecraft, A is the reference area of the spacecraft and $\mathbf{r}_s$ is the vector from the spacecraft to the Sun. The solar wind pressure is obtained from the Bernoulli equation

$$p_{SW} = \frac{1}{2} \rho v_{SW}^2 \tag{77}$$

where ρ is the density and v_{SW} is the velocity of the solar wind. Assuming a constant density in all directions of the solar system, the density is defined by

$$\rho = \frac{\dot{m}_{sun}}{v_{SW} A_s} \tag{78}$$

Where $\dot{m}_{sun}$ is the mass flow rate of the Sun and A_s is the area defined by

$$A_s = 4\pi \|\mathbf{r}_s\|^2 \tag{79}$$

It is the surface area of a sphere centered at the center of the Sun and with a radius given by the distance between Sun and spacecraft. Again, the acceleration is defined by force over mass and with (76), (77), (78) and (79) it yields

$$\mathbf{a}_{SW} = -C_D \frac{\dot{m}_{sun}}{8\pi} \frac{A}{m} v_{SW} \frac{\mathbf{r}_S}{\|\mathbf{r}_S\|^3} \tag{80}$$

The acceleration defined in the RTN frame is

$$\mathbf{\Delta}_{SW} = \mathbf{R}^{\mathrm{T}} \mathbf{a}_{SW} \tag{81}$$

where **R** is the transformation matrix.

Just like the solar radiation pressure the influence of the solar wind is given for all orbital elements and, of course, the impact is larger for spacecrafts with low ballistic coefficients. Nevertheless, the acceleration induced by the solar wind is expected to be few magnitudes less than by the solar radiation pressure. Hence its modelling is only required for very accurate trajectory propagations.

2.4 Environment

The definition of the environment is the setup of the central body as well as other celestial bodies required to model the scenario. Each body is typically defined by its equatorial radius and its gravity field defined as point mass. Only the central body might require a more detailed definition by means of polar radius to represent the flattening and by means of spherical harmonics for the gravity field.

Further it is vital to know the positions of the celestial bodies to each other, known as ephemerides. Another environmental aspect is the radiation caused by the radiation belt(s). In case of Earth, these are the inner and outer Van Allen radiation belts.

Additionally, eclipses need to be considered to model the solar electric propulsion system which is typically used for low-thrust orbit transfers.

2.4.1 Ephemerides

Certain accelerations like solar radiation pressure and third body perturbations require the position of additional celestial bodies. In general, position and velocity of celestial bodies are known as ephemerides. Each ephemeris can be either obtained from pre-computed sources like dynamic libraries or from analytical computation.

Widely used is the ephemeris data DE405 calculated by numerical integration at the Jet Propulsion Laboratory (JPL). The data file provides position and velocity of the Sun, the major planets as well as the Moon, among others.

Another source for ephemerides is the Astronomical Almanac [8]. It provides geocentric ephemerides of Sun, Moon and the planets. Besides, also the heliocentric osculating orbital elements are provided.

2.4.2 Radiation Belt

Depending on the initial and final orbit of the low-thrust transfer, the spacecraft might have to pass through the radiation belt of the central body many times. Every time the spacecraft crosses the radiation belt the total time spent there is accumulated. A basic implementation is given by

$$r_{RB,l} \leq r \leq r_{RB,u} \tag{82}$$

where $r_{RB,l}$ is the lower radius of the radiation belt and $r_{RB,u}$ is the upper radius of the radiation belt.

In case the stay time in the radiation belt is subject to be minimized, it is required to integrate the duration spent inside the belt according to

$$\dot{t}_{RB} = \begin{cases} 0 & r < r_{RB,l} \\ 1 & r_{RB,l} \leq r \leq r_{RB,u} \\ 0 & r > r_{RB,u} \end{cases} \tag{83}$$

2.4.3 Eclipses

Since typical low-thrust orbit transfers are achieved with solar electric propulsion, it is essential to consider eclipsing effects. Almost every spacecraft orbiting a celestial body except the Sun encounters eclipses during its orbits. The following kinds of eclipses are distinguished:

- Umbra is a total eclipse when the Sun is completely blocked by the central body.
- Penumbra is a partial eclipse and only a portion of the Sun is obscured by the central body. This is typically the case before and after the umbra, in other words it is the transition from 100 % sunlight to 100 % shadow.
- Antumbra is a partial eclipse as well but here the central body is entirely contained within the Sun-disc and a ring of the Sun is visible around the central body. The spacecraft experiences an annular eclipse.

Eclipses do not only impact the power generation and therefore the available thrust magnitude of a solar electric propulsion spacecraft, also perturbations caused by solar radiation pressure and solar wind are affected. A comprehensive description of the eclipse geometry as well as their conditions and computation is found in [3, 9].

2.5 Independent Variable

Usually a spacecraft trajectory is integrated over time and for most applications it is eminently suitable. In case of low-thrust planetary orbit transfers the spacecraft travels several weeks or even months while using its propulsion system to bend the trajectory. This type of orbital transfer is also called multi-revolution transfer because of the high number of revolutions.

In such a low-thrust multi-revolution transfer two problems are faced which are related to each other:

1. During a transfer starting in a low-altitude orbit and spiraling up for example to the GEO belt, the orbital period is changing from about 2 h at the beginning to 24 h when reaching the belt. It is obvious that a fixed integration step size (equidistant grid nodes, e.g. each 30 min) would result in only 1/12 integration steps over one orbital revolution in a low altitude orbit in comparison to the final GEO orbit.
2. When starting the transfer in a geostationary transfer orbit the initial eccentricity is quite high. With equidistant integration steps the accuracy during the apoapsis passage would be good while during the periapsis passage the accuracy is so poor that the result is not reliable anymore. In other words, the discretization of the spacecraft dynamics is not good enough.

Both aspects can be solved by simply increasing the number of integration steps and reducing the step size. Although the pure integration of the trajectory is working, the optimal control problem is not only enormously oversized. Also an optimization method is required handling such huge optimization problems with probably tens or hundreds of millions of parameters. Furthermore the computational effort is tremendous.

A second solution would be to place the grid points not equidistant over time but only where required. In that way the number of optimizable parameters can be kept small. However, during the optimization the shape of the orbit is changing. Thus, it might happen that perfectly and densely placed grid nodes during the periapsis passage are now during the apoapsis passage and vice versa. As a result only few nodes would be present at the periapsis passage resulting in poor discretization and a loss of accuracy.

Finally, the integration over time is very critical for low-thrust trajectories with many revolutions, typically several hundreds, especially in combination with dramatically changing orbital periods or in case of high-eccentric orbits. One solution is using almost the same number of grid nodes in each revolution and to fairly distribute them over periapsis and apoapsis passages. It results in a good discretization and can be achieved by integrating over an orbital angle: the equinoctial element L also known as the true longitude. It is the sum of true anomaly, argument of periapsis and the right ascension of the ascending node. A change of 2π in this variable represents one orbit revolution of the spacecraft.

As a result the state vector $\mathbf{x}$ representing the spacecraft dynamics

$$\mathbf{x}^{\mathrm{T}} = [p, f, g, h, k, L] \tag{84}$$

becomes

$$\mathbf{x}^{T} = [p, f, g, h, k, t] \tag{85}$$

where the true longitude L is replaced by the time t. The derivative for the state time $t(L)$ is the reciprocal of the derivative of $L(t)$:

$$\frac{dt}{dL} = \frac{1}{\dot{L}} \tag{86}$$

According to

$$\frac{dx}{dL} = \frac{dx}{dt}\frac{dt}{dL} = \dot{x}\frac{1}{\dot{L}} \tag{87}$$

it follows for the remaining state derivatives

$$\frac{dp}{dL} = \dot{p}\frac{1}{\dot{L}} \tag{88}$$

$$\frac{df}{dL} = \dot{f}\frac{1}{\dot{L}} \tag{89}$$

$$\frac{dg}{dL} = \dot{g}\frac{1}{\dot{L}} \tag{90}$$

$$\frac{dh}{dL} = \dot{h}\frac{1}{\dot{L}} \tag{91}$$

$$\frac{dk}{dL} = \dot{k}\frac{1}{\dot{L}} \tag{92}$$

2.6 Controls

Controls represent optimizable parameters as function of the independent variable. But in contrary to states they do not have equations of motion. Still, they are impacting the dynamics of the optimal control problem. For low-thrust orbit transfer it is suitable to use the spacecraft attitude or at least the thrust direction as control. In case of the thrust direction, the third degree of freedom of the spacecraft attitude is usually not modelled since it increases the complexity and size of the optimal control problem.

Few representations of the direction and magnitude of the thrust vector exist

- Cartesian vector components
- Spherical vector components (i.e. angles) and throttle
- Quaternion and throttle

The first two have three control variables, while the latter one results in five control variables. Of course, quaternions have several advantages. Most important is the proper handling of the gimbal lock. But there are also other benefits for example in memory consumption, interpolation and transformation efficiency. However, two additional control variables increase the problem size.

A second option is to describe the thrust vector with a reduced set of Euler angles: the out-of-orbit-plane angle yaw and the in-plane angle pitch. As both angles describe the direction of the thrust vector a third control is required for its magnitude: the throttle. They define the spherical vector components. Because angular quantities are periodic by nature (e.g. $\sin\alpha = \sin\alpha \pm 2\pi$) they show a wrapping behaviour in the optimization. In summary, they are fast in computation but not very robust.

Therefore it is most convenient using the Cartesian representation of a vector as control. It describes magnitude and direction of the thrust vector. Two possible frames for the vector representation exist. First, in case of the inertial frame the control vector is defined as

$$\mathbf{u} = \begin{pmatrix} u_x \\ u_y \\ u_z \end{pmatrix} \tag{93}$$

Since the thrust acceleration is required in the rotating RTN-frame it yields

$$\Delta_T = \frac{T}{m}\mathbf{R}^{\mathrm{T}}\frac{\mathbf{u}}{\|\mathbf{u}\|} = \frac{T}{m}\mathbf{R}^{\mathrm{T}}\frac{\left(u_x\ u_y\ u_z\right)^{\mathrm{T}}}{\left\|\left(u_x\ u_y\ u_z\right)^{\mathrm{T}}\right\|} \tag{94}$$

The second possibility is to define the control vector directly in the rotating RTN-frame

$$\mathbf{u} = \begin{pmatrix} u_r \\ u_t \\ u_n \end{pmatrix} \tag{95}$$

Now, the thrust acceleration vector becomes

$$\Delta_T = \frac{T}{m}\mathbf{u} = \frac{T}{m}\frac{\left(u_r\ u_t\ u_n\right)^{\mathrm{T}}}{\left\|\left(u_r\ u_t\ u_n\right)^{\mathrm{T}}\right\|} \tag{96}$$

Without transformation matrix this approach is faster in computation as long as only the dynamics are considered.

3 Optimization

To optimize a low-thrust orbit transfer the optimal control problem (OCP) has to be solved. Very efficient is a direct transcription of the OCP into a nonlinear programming problem by discretization. In a next step the NLP problem is solved by using sequential quadratic programming (SQP). The drawback of this approach is the high number of parameters, typically in the order of several 10,000 or 100,000. Therefore a sparse NLP solver is required.

This section briefly describes few discretization methods applicable to solve the optimal control problem of low-thrust orbit transfers. Furthermore the constraints and objectives are discussed. At the end of this section some details are provided about the optimization convergence and the accuracy of the solution. But before the creation of an initial guess is explained in more detail, since it provides an initial solution of the problem.

3.1 Initial Guess

Computing low-thrust trajectories from "scratch" which can be used as initial guess for the optimization process is very crucial due to the complexity of the spacecraft attitude during its multi-revolution orbit transfer. To provide suitable initial guesses two different methods can be identified:

1. Using analytic control laws to describe the spacecraft attitude.
2. Using a previously computed (sub-)optimal spacecraft attitude.

In the first method analytic control laws are applied like the ones described by Pollard [10]. These attitude laws change either one or multiple of the following orbital parameters:

- semi-major axis
- eccentricity
- inclination
- argument of periapsis
- right ascension of the ascending node.

Furthermore some laws can be enhanced by an out-of-plane component so that the inclination changes at the same time. For example, such a strategy is very convenient when for the orbital elements semi-major axis and eccentricity. An internal scheme following predefined rules can be applied to change the control laws automatically during the initial guess creation. Besides the initial and final orbit conditions have to be considered as well to match the user defined transfer characteristics.

A second approach to provide an initial guess is to use the attitude of an already computed (sub-)optimal trajectory. Especially for very complex orbit transfers it might be required since the problem description is changing between multiple optimization runs. However in both cases the states are purely integrated according to the equations of motion introduced in the previous section and considering the provided attitude.

3.2 Discretization

Discretization methods are required to transcribe the optimal control problem into a nonlinear programming problem. Since the entire optimization problem is discretized on a grid it is also known as direct transcription. The grid contains the so called collocation nodes where state and control values are known. But the state values are not integrated. They are approximated using polynomials depending on the applied discretization method. Due to this approximation a small discontinuity is present at each node for every state: the discretization error.

Several different discretization schemes exist, such as

- Trapezoidal
- Hermite-Simpson compressed
- Hermite-Simpson separated

The trapezoidal method is a basic discretization scheme of order 2 whereas the Hermite-Simpson discretization schemes are of order 4. They last two use third-order polynomials and introduce additional NLP variables at the mid-point: the compressed scheme adds control variables only and the separated scheme adds control and state variables at the mid-point. In both cases not only the number of NLP variables increases but also the number of constraints since the defects at the mid-points have to be evaluated additionally. Thus, the trapezoidal discretization method is faster in computation but less robust than the Hermite-Simpson methods.

3.3 Constraints

It is important to postulate some requirements which have to be met and they are typically modelled as constraints. The following kinds of constraints are distinguished:

- initial boundary constraints (at t_0/L_0)
- path constraints (between t_0/L_0 and t_f/L_f)
- final boundary constraints (at t_f/L_f)

The first and the last one are also known as point constraints and they are evaluated at the initial and final value of the independent variable, respectively. Initial boundary constraints have to meet the initial orbital properties, while the final boundary constraints have to meet the final or target orbit conditions.

Since the control vector is defined as unit vector it must be assured that the length of the control vector equals one at each value of the independent variable where the control vector is present. Thus, this is modelled as path constraint

$$0 = 1 - \sqrt{u_r^2 + u_t^2 + u_n^2} \tag{97}$$

in case the attitude control is defined in the rotating RTN-frame.

3.4 Objectives

The objective of the optimal control problem is subject to be minimized while all constraints have to be fulfilled. In principle, any property of the optimal control problem might be formulated as objective. While there is only one objective in single-objective optimization problems, it might contain several cost terms which are, for example, simply summed up.

Similar to the constraints different kinds of cost functions are distinguished:

- initial cost (at t_0/L_0)
- Lagrange cost (between t_0/L_0 and t_f/L_f)
- final cost (at t_f/L_f)

Initial and final cost are known as Mayer cost terms and evaluated at the initial and final value of the independent variable, respectively. The final cost is also known as terminal cost. A Lagrange cost is an integrated cost term from initial to final independent variable.

Typical final cost terms are the transfer duration or the propellant consumption, depending on the scenario. The path constraints for the unit vector length can be also reformulated as Lagrange cost.

3.5 Convergence and Precision

Convergence of the optimal control problem is achieved when few conditions are satisfied. For example, the violation of the constraints must be within a specified tolerance (typically in the order of 1e^{-7}). Further, the norm of the projected gradient of the partial derivatives of$f$ must be within a specified tolerance of e.g. 1e^{-7}. It is a measure for the optimality of the OCP. But also the error in the discretization must be smaller than a certain tolerance (e.g. 1e^{-6}). The discretization error depends on

Table 1 Example for convergence of optimizer

Mesh number	Number of grid nodes	Error in equations
1	5001	$2.4e^{-5}$
2	10,001	$9.0e^{-7}$
3	10,001	$1.1e^{-7}$
4	10,354	$3.0e^{-8}$

Table 2 Accuracy of the final spacecraft position for a 6-month low-thrust GTO-GEO transfer

	Optimization (km)	Propagation (km)
x-Position	42,138.5	42,198.8
y-Position	−1471.5	−1473.6
z-Position	0.0	9.3

the discretization scheme, the variable (states and controls) values, their derivatives as well as the grid. Since state derivatives are given by the dynamics of the optimal control problem, the discretization accuracy, in other words the accuracy of the approximation, strongly depends on the grid and the interval length between two grid nodes. More precisely the amount of change of a variable within a discretization interval is a measure for the discretization error. When variables are not changing much, longer intervals are fine, while shorter intervals are required in case at least one variable changes rapidly. This is a reason why proper selection of spacecraft dynamics and control variables is crucial.

During the optimization process the optimizer is converging by introducing additional grid nodes where the requested discretization accuracy is not achieved. It refines the discretization and each time additional nodes are added to the grid is understood as mesh refinement. Usually the optimization algorithm utilizes automatic mesh refinements. One example of the discretization error and the number of grid nodes for several mesh refinements is summarized in Table 1. A final discretization error of about $3e^{-8}$ is almost machine accuracy. More information about discretization accuracy, mesh refinement and convergence is provided in [11, 12].

Considering the spacecraft dynamics and controls of the optimal control problem, mesh number 4 results in about 200,000 parameters and about 160,000 constraints.

An example for the accuracy of the final spacecraft position after successful mesh refinement of an end-to-end low-thrust orbit transfer from GTO to GEO is given in Table 2. The result of the optimization is exactly the requested target orbit position, whereas the propagation of the optimized attitude profile is slightly different. Of course, it must be considered that at each node of the grid a small discretization (less than $3e^{-8}$) error is present which results in a different propagated position. Further is has to be remembered that the integrated trajectory lasts about 6 months and about 260 orbital revolutions. Finally, the difference in x-position is about 60 km, or 0.14 %. In the y and z components the position error is much smaller with only a few kilometers.

4 Examples of GTO-GEO Transfers

Few examples are presented based on the modeling and optimization concept already introduced. The application is a transfer from the GTO to the GEO. Certain results like time and propellant optimality are shown as well as the impact of phasing on the optimal solution. How to avoid the crossing the GEO ring is shown as well.

For the presented examples it is assumed to have a spacecraft of 1,000 kg with a thrust magnitude of 150 mN. Initial and final orbit conditions are summarized in Tables 3 and 4. Perturbations are not considered in the given examples. Their impact on the performance parameters (time, propellant consumption) is negligible for this type of transfer. But the control histories would be actually different, in particular the thrust direction. Anyway, the impact on the shown figures is almost not traceable.

4.1 Time Optimal Transfer

Before showing the results for a time optimal low-thrust transfer from GTO to GEO the computation of the initial guess is shown. For sake of convenience the attitude control laws are augmented by an out-of-plane control strategy for efficient inclination change. These laws are to change the semi-major axis or eccentricity, for example. Other possible control laws include very basic control strategies like constant thrust in radial, normal or transverse/tangential orbit direction.

For a standard GTO to GEO transfer with an initial inclination of 27° the attitude control is shown in Fig. 1. Two different phases can be identified: first half of the transfer the orbit energy is increased at a maximum rate with only a small portion for the inclination change. Once the desired orbit energy, here GEO orbit altitude,

Table 3 Initial orbit conditions

Apoapsis altitude	35,786 km
Periapsis altitude	250 km
Inclination	27 deg
True anomaly	180 deg
Argument of periapsis	178 deg
RAAN	0 deg

Table 4 Final orbit conditions

Semimajor axis	42,164.137 km
Eccentricity	0
Inclination	0 deg
Argument of periapsis	0 deg
RAAN	0 deg
True anomaly	0 deg
Relative longitude	37 deg

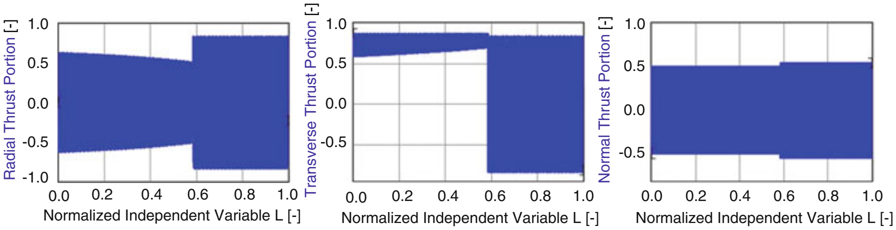


Fig. 1 Control history of radial (*left*), transversal (*middle*), and normal component (*right*) using enhanced attitude control laws

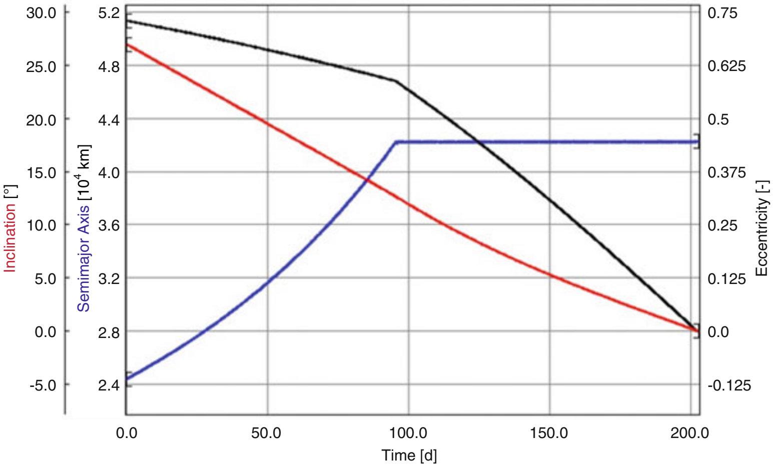


Fig. 2 History of the orbital elements semi-major axis (*blue*), eccentricity (*black*) and inclination (*red*) of a propagated GTO to GEO low-thrust transfer using augmented attitude control laws

Table 5 Comparison of a GTO-GEO transfer using augmented attitude control laws and the time optimal solution

	Transfer duration (days)	Fuel consumption (kg)
Initial guess with control laws	202.94	134.10
Time optimal solution	188.95	124.85

is achieved, the control of the remaining transfer circularizes the orbit shape and reduces the inclination to zero. At the end GEO orbit is reached (Fig. 2).

The presented initial guess is an excellent starting point for the optimization of the transfer while minimizing the transfer duration. After the optimization the transfer performance is increased by about 7 % (see Table 5). The optimal control history of the spacecraft attitude is shown in Fig. 3 and the optimized orbit elements are presented in Fig. 4.

For the computation of the initial guess where analytic laws are used less than 1 min of a single-core CPU (Central Processing Unit) is required. The converged

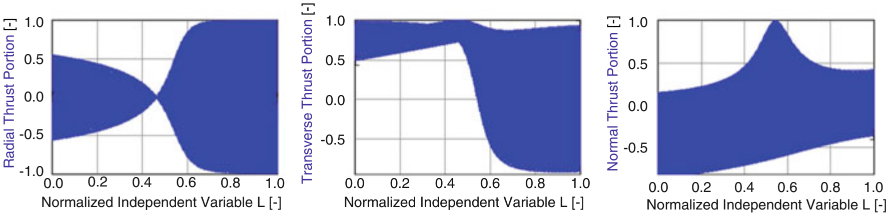


Fig. 3 Control history of a time optimal GTO to GEO transfer after converged optimization: the radial thrust vector control component is shown to the *left*, the transversal component in the *middle*, and the normal component to the *right*

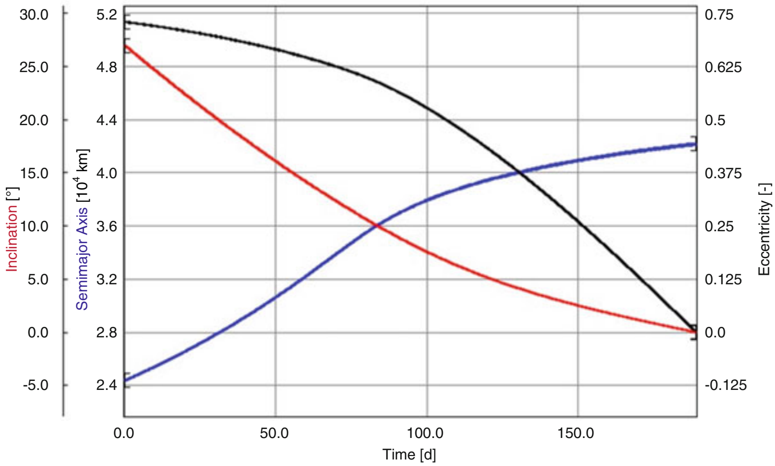


Fig. 4 Semi-major axis (*blue*), eccentricity (*black*) and inclination (*red*) of time optimal transfer

solution using the sparse NLP optimizer is achieved in few minutes. In case of a more sophisticated model including many perturbing forces like third body perturbations, atmospheric drag and solar radiation pressure the computation time is increasing due to many computations of the ephemerides to retrieve the position of the spacecraft with respect to other orbital bodies. Anyway, the converged solution is achieved in less than 30 min on a today standard desktop computer using one CPU core.

4.2 Propellant Optimal Transfer

For the minimization of the propellant consumption the transfer duration needs to be extended. The longer the transfer with respect to the time-optimal solution the more propellant can be saved. But there is a minimum propellant consumption required to bend the trajectory to the desired target orbit. An example is shown

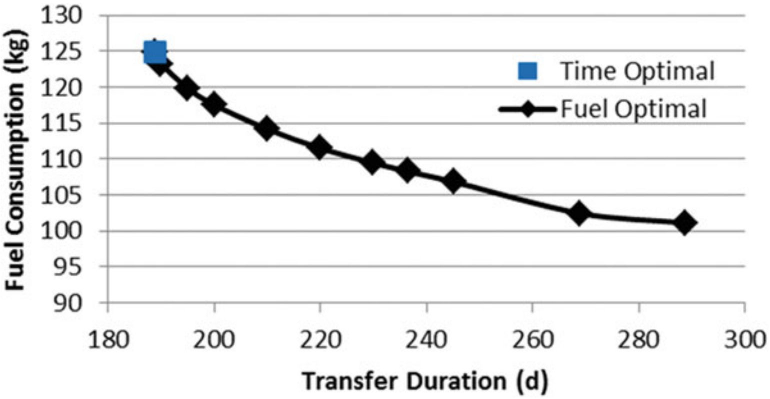


Fig. 5 Pareto front of the fuel consumption versus transfer duration of propellant optimal transfers

Table 6 Seasoning effect of eclipses caused by Earth on GTO-GEO transfers

Epoch	21st of March	21st of June	21st of September	21st of December
Duration of eclipses (h)	68.2	58.0	81.0	92.2

in Fig. 5. All presented solutions define the Pareto front of this optimal control problem. Here, the propellant consumption of 125 kg for a time optimal transfer can be reduced to about 100 kg when the transfer duration is extended by more than 50 %. It seems that further extended mission duration will not significantly reduce the fuel consumption.

It is also known as multi-objective optimization problem since we have two objectives: the time and the propellant consumption. There is not one single optimum but a whole solution class known as the Pareto front.

4.3 Eclipses

As the propulsions system is fed by solar energy the effect of eclipses during the many months lasting low-thrust transfer cannot be neglected. But eclipses strongly depend on the seasons and therefore the initial date when the transfer starts. An example for an Ariane 5 GTO to GEO transfer lasting about 6 months is given in Table 6. The difference between the minimum and the maximum time spent in the shadow of Earth is more than 50 %.

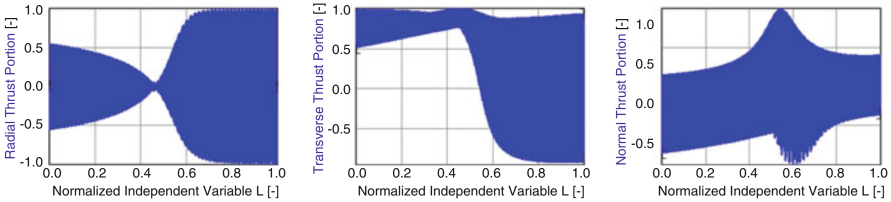


Fig. 6 Attitude profiles for time-optimal transfer with GEO box targeting

4.4 Phasing

Phasing means the targeting of a certain longitude in the GEO, also known as GEO box/slot. The relative (geographic) longitude of the final spacecraft position has to match a required GEO box. Phasing couples the final orbit with a certain time. In the worst case the transfer duration is increasing by almost 1 day with respect to the time optimal one when targeting a specific GEO slot, because the Earth rotates once in 24 h.

The required phasing strongly depends on the initial launch epoch, orbit and target GEO box longitude. In the given example in Fig. 6 the spacecraft targets a longitude of 15° east while in the time-optimal transfer it was approaching at about 13° east. Thus, the transfer needs to be extended by almost one day for the phasing. The changes in radial and normal attitude control component are quite obvious in comparison to the time optimal solution (see Fig. 3). Alternatively, the spacecraft could start its transfer also about 2/360 days earlier to arrive at 15° East with the time optimal transfer and no phasing would be required.

4.5 GEO Belt Crossings

The GEO belt, or GEO ring, is understood as the area in space where most of the operational satellites in geosynchronous orbit are located. Typically they have an inclination and eccentricity of almost zero, but arbitrary geographic longitudes.

In general, a spacecraft might cross the GEO ring at the beginning, mid of the transfer, and at the end. But it strongly depends on the orbital parameters of the transfer trajectory. For example, most influence has the initial and final orbit as well as the argument of periapsis during the transfer. Especially at the end of a low-thrust orbit transfer to the GEO the spacecraft may cross the GEO belt several times since the spacecraft targets zero eccentricity and inclination at GEO altitude.

To avoid the risk of a possible collision with assets in the GEO belt, a condition is formulated as cost function forbidding the spacecraft to travel through the GEO ring. The situation is illustrated in Fig. 7 in a co-rotating frame. The x-axis is the direction from the center of Earth towards the projected spacecraft position

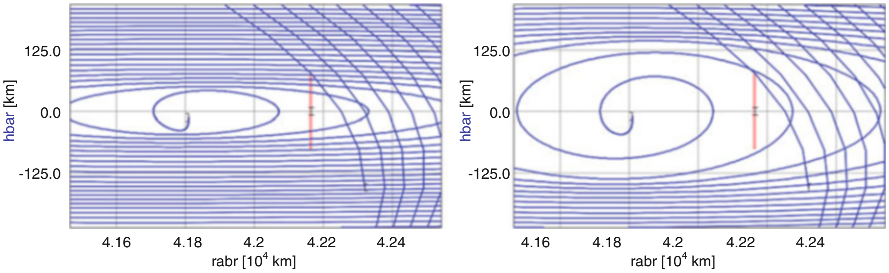


Fig. 7 Super-synchronous transfer (*blue*) without GEO belt avoidance (*left*) and with active GEO belt avoidance (*right*). The location of the GEO ring is indicated by the *red rectangle*

Table 7 Performance table for time optimal transfers regarding GEO belt crossings

	GEO ring crossings	Transfer duration (%)	Propellant consumption (%)
Super-synchronous	7	100.0	100.0
GEO crossing condition	0	100.02	100.02
Sub-synchronous	0	110.9	110.9

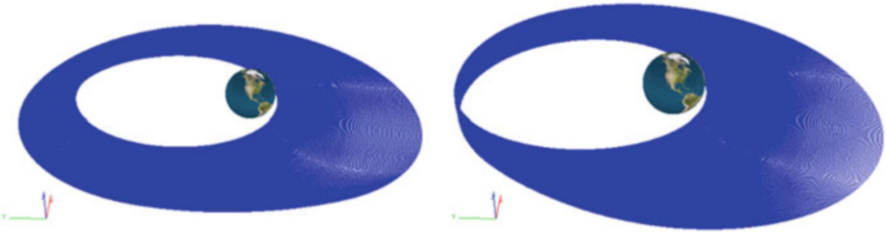


Fig. 8 3d plot of super-synchronous (*left*) and sub-synchronous transfer (*right*)

in the equator plane, also known as r-bar. The y-axis is the out-of-equator plane component pointing north, labelled h-bar. The red rectangle indicates the GEO ring location in the co-rotating frame. In the right figure it can be seen that the spacecraft circumnavigates the GEO belt once the formulated condition is considered in the optimization problem. The fuel consumption and transfer duration is increasing by less than 0.02 % (see Table 7). In this example, a target orbit with an altitude of 500 km below the GEO orbit was targeted to demonstrate zero crossings of the belt. When directly targeting, there is one “crossing” of the belt when the spacecraft enters it to meet the final position located inside the GEO belt.

An alternative to reduce the number of crossings is a sub-synchronous transfer with the spacecraft staying below GEO altitude during the whole transfer (Fig. 8). Here, the transfer duration and propellant consumption is increased by e.g. about 11 % in comparison to a super-synchronous transfer with active GEO belt crossing avoidance (see also Table 7).

5 Conclusions

In this chapter the modeling and optimization of low-thrust multi-revolution orbit transfers was outlined. First, the modeling of the aerospace problem was introduced. It encompasses the spacecraft dynamics. Using modified equinoctial elements to represent the position and velocity of the spacecraft is perfectly suited for this type of application. Next, all typical perturbations affecting the spacecraft dynamics were presented. Besides the thrust, it is vital to consider atmospheric drag, solar radiation pressure as well as gravitational perturbations like central body oblateness and third bodies.

For the optimization of such large-scale optimal control problems a direct collocation method is used to transcribe it into an NLP problem using sequential quadratic programming. Creation of a suitable initial guess is possible with simple analytic attitude control laws. Furthermore a brief description of the constraints and objective function was given and the typical performance and accuracy was presented.

Several examples of GTO to GEO transfers have shown the practical application of the presented procedure. It was shown that time and propellant optimal transfers can be solved even when considering complex transfer characteristics like phasing and avoidance of GEO ring crossings.

Once the optimal control problem was formulated, it is not required to reformulate it when the orbit transfer characteristics change. For example, changing the initial or final orbit, thrust to mass ratio or considered perturbations do not require any reformulation of the original OCP. Therefore this presented approach is perfectly suited for mission analysis engineers.

References

1. Betts, J.T.: Very low thrust trajectory optimization using a direct SQP method. J. Comput. Appl. Math. **120**, 27–40 (2000)
2. Schäff, S.: Re-optimization of a perturbed low-thrust GTO to GEO transfer for operational purpose. Diploma Thesis, Astos Solutions GmbH and University of Stuttgart, Stuttgart, Germany (2007)
3. Vallado, D.A.: Fundamentals of Astrodynamics and Applications. Space Technology Library, 3rd edn. Springer, New York (2007)
4. Chobotov, V.A.: Orbital Mechanics. AIAA Education Series, 3rd edn. American Institute of Aeronautics and Astronautics, Reston, VA (2002)
5. Battin, R.H.: An Introduction to the Mathematics and Methods of Astrodynamics. AIAA Education Series, Rev. edn. American Institute of Aeronautics and Astronautics, Reston, VA (1999)
6. Bowman, B.: Jacchia-Bowman thermospheric density model. http://sol.spacenvironment.net/jb2008/ (2012). Accessed 6 Oct 2012
7. Ryabova, G.O.: On the dynamical consequences of the Poynting-Robertson drag caused by solar wind, dynamics of populations of planetary systems. Proceedings IAU Colloquium No. 197, International Astronomical Union (2005)

8. The Statery Office: The Astronomical Almanac for the Year 2008. Statery Office, United Kingdom (2006)
9. Wertz, J.R.: Spacecraft Attitude Determination and Control. Kluwer Academic, Dordrecht (1978)
10. Pollard, J.E.: Simplified Analysis of Low-Thrust Orbital Maneuvers. Aerospace Corporation, El Segundo, CA (2000)
11. Becerra, V.M.: Practical direct collocation methods for computational optimal control. In: Fasano, G., Pinter, J.D. (eds.) Modeling and Optimization in Space Engineering. Springer, New York (2013)
12. Betts, J.T.: Practical Methods for Optimal Control Using Nonlinear Programming. Society for Industrial and Applied Mathematics, Philadelphia (2001)

Balance Layout Problems: Mathematical Modeling and Nonlinear Optimization

Yuriy Stoyan, Tatiana Romanova, Alexander Pankratov, Anna Kovalenko, and Peter Stetsyuk

Abstract The paper studies the optimal layout problem of 3D-objects (solid spheres, straight circular cylinders, spherocylinders, straight regular prisms, cuboids and tori) in a container (a cylindrical, a parabolic, or a truncated conical shape) with circular racks. The problem takes into account a given minimal and maximal allowable distances between objects, as well as, behaviour constraints of the mechanical system (equilibrium, moments of inertia and stability constraints). We call the problem the Balance Layout Problem (BLP) and develop a continuous nonlinear programming model (NLP-model) of the problem, using the *phi*-function technique. We also consider several BLP subproblems; provide appropriate mathematical models and solution algorithms, using nonlinear programming and nonsmooth optimization methods, illustrated with computational experiments.

Keywords Layout problems • Behaviour constraints • Distance constraints • *Phi*-functions • Quasi-*phi*-functions • NLP-models • Optimization algorithms

1 Introduction

3D layout optimization problems have a wide spectrum of practical applications. In particular, these problems arise in space engineering for rocketry design. Their distinctive feature consists of taking into account behaviour constraints of a satellite system. Behaviour constraints specify the requirements for system's mechanical properties such as equilibrium, inertia, and stability. Many publications analyze problems of the equipment layout in modules of spacecraft or satellites (see, e.g.

Y. Stoyan • T. Romanova (✉) • A. Pankratov • A. Kovalenko
Department of Mathematical Modeling and Optimal Design, Institute for Mechanical Engineering Problems of the National Academy of Sciences of Ukraine, 2/10 Pozharskyi Str., 61146 Kharkov, Ukraine
e-mail: tarom27@yahoo.com

P. Stetsyuk
Department of Methods of Nonsmooth Optimization, Glushkov Institute of Cybernetic of the National Academy of Sciences of Ukraine, 40 Glushkova Ave., 03187 Kyiv, Ukraine

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization and Its Applications 114, DOI 10.1007/978-3-319-41508-6_14

[1, 2]). For example, objects layout problems for a simplified scheme of satellite module taken into account behaviour constraints were considered in [3–8]. These problems are NP-hard [9].

To construct adequate mathematical models of the layout optimization problems in the form of nonlinear programming problems, analytical description of special constraints is important: placement constraints (non-overlapping of objects, containment of objects in a container with regard for the minimal and maximal allowable distances) and behaviour constraints (equilibrium, moments of inertia, and stability constraints).

The *phi*-function technique is generally known to be an efficient tool of mathematical modeling of geometric objects relations in the class of placement problems. This technique allows nonlinear programming methods to solve the placement optimization problems. The studies [1, 2] present radical-free *phi*-functions and quasi-*phi*-function for classes of 2D and 3D objects. With the use of these functions, mathematical models of some types of layout optimization problems described in [1, 10] are proposed.

In the paper we consider the Balance Layout Problem (called the BLP problem) in the following statement: arrange 3D objects in a container with circular racks taking into account special constraints so that the objective function attains its extreme value. The objects are solid spheres, straight circular cylinders, spherocylinders, straight regular prisms, cuboids and tori. As a container we choose a cylindrical, parabolic or truncated conical shape.

The purpose of the study is to create an exact mathematical model of the balance layout of 3D-objects as a nonlinear programming problem. Such model can be used to obtain various variants of the BLP problem, which are determined by the variety of spatial forms of objects and containers, forms of the objective function, and the presence of the special constraints mentioned above.

Our chapter is organized as the following. Section 2 introduces shapes of objects and types of containers, provides analytical descriptions of the placement and behaviour constraints considered in the BLP problem. In Sect. 3 we develop the exact NLP-model of the BLP problem and propose a general solution strategy. Section 4 is devoted to modeling and solving of some variants of the BLP problem. Here we present new algorithms to construct feasible starting points, involved in our multistart strategy. Here we give computational results illustrated with pictures. Section 5 completes our chapter with some conclusions.

2 Problem Formulation

2.1 *Objects and Containers*

Let $\Omega = \{(x, y, z) \in \mathbb{R}^3 : \ \mathrm{G(x, y, z)} \geq 0\}$ be a container of given height H. We consider the following types of containers: 1) $\Omega \equiv \mathbf{C}$, $\mathbf{C}$ is a straight circular cylinder with a base of radius R, $G(x, y, z) = \min\{-x^2 - y^2 + R^2, -z + H, z\}$; 2)

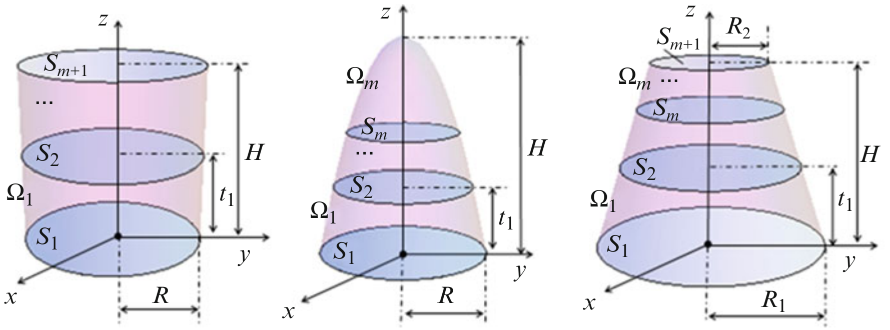


Fig. 1 Types of containers

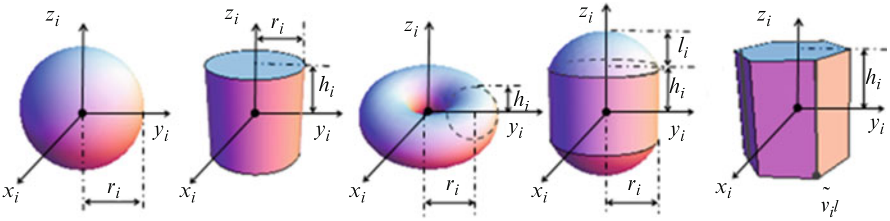


Fig. 2 Shapes of objects

$\Omega \equiv \mathbf{\Lambda}$, $\mathbf{\Lambda}$ is a paraboloid of revolution with a base of radius $R = \sqrt{H}$, $G(x, y, z) = \min\{-z - x^2 - y^2 + H, z\}$; 3) $\Omega \equiv \mathbf{E}$, $\mathbf{E}$ is a straight circular blunted cone with lower and upper bases of radii R_1 and $R_2 < R_1$ respectively, $G(x, y, z) = \min\{-z - H(\sqrt{x^2 + y^2} - R_1)/(R_1 - R_2), -z + H, z\}$. Suppose that Ω is divided by circular racks S_k, $k = 1, 2, \ldots, m + 1$, into subcontainers Ω^k, $k = 1, 2, \ldots, m$. We assume that S_1 is a base of Ω. Between racks S_k and S_{k+1} the distance t_k is given.

We specify $Oxyz$ as a local coordinate system of container Ω, where Oz is the longitudinal axis of symmetry of Ω. The origin O of system $Oxyz$ is the center of symmetry of the lower base S_1 of Ω (Fig. 1).

Family $A = \{A_i, i \in I_n\}$, $I_n = \{1, 2, .., n\}$, involves the following shapes of objects: solid spheres $\mathbb{S}_i$ of radius r_i; straight circular cylinders $\mathbb{C}_i$ of radius r_i and height $2h_i$; tori $\mathbb{T}_i$ with metric characteristics (r_i, h_i), where r_i is the distance from the center of generating circle to the axis of revolution, $2h_i$ is the height of $\mathbb{T}_i$, h_i is the radius of the generating circle; spherocylinders $\mathbb{S}_{\mathbb{C}i}$ with metric characteristics (l_i, r_i, h_i), where l_i is the height of ball segments, r_i is the radius and $2h_i$ is the height of cylinder; straight regular prisms and cuboids $\mathbb{K}_i$ with metric characteristics $(h_i, \tilde{v}_{il},)$, where $2h_i$ is the height of $\mathbb{K}_i$, $\tilde{v}_{il} = (\tilde{x}_{il}, \tilde{y}_{il})$, $l = 1. \ldots, \mathrm{s_i}$, are vertices of the base of $\mathbb{K}_i$ (which is a convex polygon K_i), s_i is the number of vertices of K_i.

We specify the local coordinate system $O_i x_i y_i z_i$ of object A_i. The axes of the system we denote by $O_i x_i$, $O_i y_i$, $O_i z_i$. The origin O_i of the coordinate system is at the center of object A_i. We note that $O_i z_i \| Oz$ (Fig. 2).

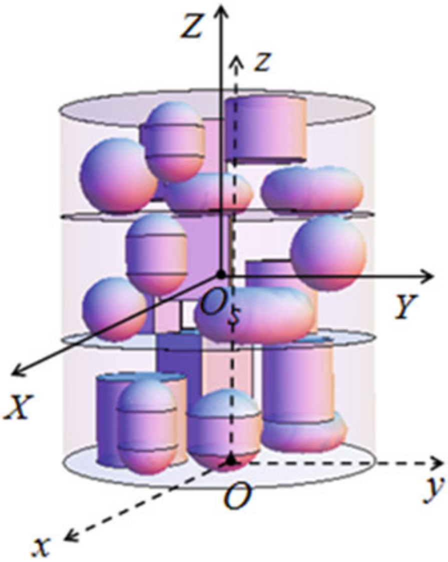


Fig. 3 Example of system Ω_A

Each object A_i is a homogeneous rigid body of given mass m_i.

Assume that a partition of A into subsets $A^k = \{A_i, i \in I^k\}$, $k = 1, 2, \ldots, m$, $I^1 \cup I^2 \cup \ldots \cup I^k \cup \ldots I^m = I_n$, with respect to the placement of the subsets of objects inside subcontainers Ω^k, $k = 1, 2, \ldots, m$, is given.

In turn each subset A^k is divided into two subsets $A^k_+ = \{A_i, i \in I^k_+\}$ and $A^k_- = \{A_i, i \in I^k_-\}$, where A^k_+ is subset of objects, which have to be placed *on* the rack S_k, A^k_- is subset of objects, which have to be placed *under* the rack S_{k+1} inside subcontainer Ω^k.

Any arrangement of object $A_i \in A$ inside container Ω is defined by vector $u_i = (v_i, \theta_i)$ with respect to the coordinate system $Oxyz$, where $v_i = (x_i, y_i, z_i)$ is a variable translation vector of object A_i, θ_i is a variable rotation angle of object A_i in the plane $O_i x_i y_i$. Thus, a vector of variables $u = (p, u_1, u_2, \ldots, u_n)$, defines the arrangement of object family A inside container Ω where p is a vector of variable parameters of container. Container Ω with the objects packed in it is called a system Ω_A (Fig. 3).

Balance Layout Problem (BLP): Pack 3D objects $A_i \in A$, $i = 1, 2, \ldots, n$, sliding on (above or below) assigned racks S_k, $k = 1, 2, \ldots, m$, inside container Ω, so that the given objective function $F(u)$ attains its extreme value with regard for special constraints.

Let us define the special constraints, which embrace *placement constraints* and *behaviour constraints*.

2.2 Placement Constraints

The placement constraints in the BLP problem are generated by non-overlapping of objects A_i, A_j, $i > j \in I^k$, $k = 1, \ldots, m$, which have to be placed inside subcontainer Ω^k, and containment of object A_i in container Ω, $i \in I_n$. In addition, the minimal ρ_{ij}^{-} and maximal $\rho_{ij}^{+} \geq \rho_{ij}^{-}$ allowable distances between objects $A_i, A_j \in A^k$, $i > j \in I^k$, may be specified. Also, the minimal allowable distance ρ_i^{-} between object $A_i \in A$, $i \in I_n$, and the lateral surface of container Ω may be given. Without loss of generality we set $\rho_{\mathrm{ij}}^{-} = 0$ (or $\rho_{ij}^{+} = \varpi$) if a minimal (or a maximal) allowable distance between objects A_i and A_j is not given, $i > j \in I^k$. Here ϖ is a given sufficiently great number. In particular, the condition $\rho_{ij}^{+} = \rho_{ij}^{-}$ provides the arrangement of objects A_i and A_j on the exact distance. We also set $\rho_i^{-} = 0$ if a minimal allowable distance between object A_i and the lateral surface of subcontainer Ω^k is not given.

Placement constraints in the BLP problem may be presented as the following:

$$\rho_{ij}^{-} \leq \operatorname{dist}(A_i, A_j) \leq \rho_{ij}^{+}, \quad i > j \in I^k, k = 1, 2, \ldots, m,$$

and

$$\operatorname{dist}(A_i, \Omega^*) \geq \rho_i^{-}, i = 1, \ldots, n,$$

where $\Omega^* = \mathbb{R}^d \backslash \operatorname{int}\Omega$, $d = 2, 3$.

To describe the placement constraints analytically we employ the *phi*-function technique (see, e.g., [11, 12]). We offer the reader some needed definitions illustrated with examples in Appendix 1.

Adjusted phi-functions and quasi-phi-functions. Let $A, B \subset \mathbb{R}^{\mathrm{d}}$ be two *phi*-objects (see, e.g., [11] and Appendix 1), $d = 2, 3$. Assume, that at least one of them is bounded. And let u_A and u_B be placement parameters of A and B, respectively. Let minimal ρ^{-} and maximal ρ^{+} allowable distances between objects A and B be given, i.e.

$$\operatorname{dist}(A, B) \geq \rho^{-}, \operatorname{dist}(A, B) \leq \rho^{+}$$

where $\operatorname{dist}(A, B) = \min\limits_{t_1 \in A, t_2 \in B} d(t_1, t_2)$, $d(t_1, t_2)$ is the Euclidean distance between two points t_1 and t_2.

To formalize the mentioned above distance constraints we employ adjusted *phi*-functions (see, e.g., [11]) and adjusted quasi-*phi*-functions (see, e.g., [2, 13]).

Definition 1. Everywhere defined and continuous function $\overset{\frown}{\Phi}^{-}(u_A, u_B)$ (or $\overset{\frown}{\Phi}^{+}(u_A, u_B)$) is called an adjusted *phi*-function for objects $A(u_A)$ and $B(u_B)$, if

$$\begin{array}{ll}
\overset{\frown}{\Phi}^{-}(u_A, u_B) > 0, \text{ if } \operatorname{dist}(A, B) > \rho^{-} & (\overset{\frown}{\Phi}^{+}(u_A, u_B) > 0, \text{ if } \operatorname{dist}(A, B) < \rho^{+}), \\
\overset{\frown}{\Phi}^{-}(u_A, u_B) = 0, \text{ if } \operatorname{dist}(A, B) = \rho^{-} & (\overset{\frown}{\Phi}^{+}(u_A, u_B) > 0, \text{ if } \operatorname{dist}(A, B) = \rho^{+}), \\
\overset{\frown}{\Phi}^{-}(u_A, u_B) < 0, \text{ if } \operatorname{dist}(A, B) < \rho^{-} & (\overset{\frown}{\Phi}^{+}(u_A, u_B) < 0, \text{ if } \operatorname{dist}(A, B) > \rho^{+}).
\end{array}$$

Definition 2. A function $\overset{\frown}{\Phi}{}'(u_A, u_B, u')$ is called an adjusted quasi-*phi*-function for objects $A(u_A)$ and $B(u_B)$, if function $\max\limits_{u' \in \mathbb{R}^d} \overset{\frown}{\Phi}{}'(u_A, u_B, u')$ is an adjusted *phi*-function $\overset{\frown}{\Phi}(u_A, u_B)$ for objects $A(u_A)$ and $B(u_B)$. Here u' is a vector of extra variables. The dimension of Euclidean space $\mathbb{R}^{\mathrm{d}}$ depends on the object shapes.

By analogy we discriminate the adjusted quasi-*phi*-functions for modeling minimal and maximal allowable distances between objects A and B, and denote these functions by $\overset{\frown}{\Phi}{}'^{-}$ and $\overset{\frown}{\Phi}{}'^{+}$.

Thus, $\max\limits_{u' \in U} \overset{\frown}{\Phi}{}'^{-} \geq 0 \Leftrightarrow \operatorname{dist}(A, B) \geq \rho^{-}$ and $\max\limits_{u' \in U} \overset{\frown}{\Phi}{}'^{+} \geq 0 \Leftrightarrow \operatorname{dist}(A, B) \leq \rho^{+}$. Using the properties of quasi-*phi*-functions [13], we may conclude that $\overset{\frown}{\Phi}{}'^{-} \geq 0 \Rightarrow \operatorname{dist}(A, B) \geq \rho^{-}$, $\overset{\frown}{\Phi}{}'^{+} \geq 0 \Rightarrow \operatorname{dist}(A, B) \leq \rho^{+}$.

Placement constraints in terms of phi-functions. Using the definitions of the adjusted *phi*-function and the adjusted quasi-*phi*-function our placement constraints in the BLP problem may be presented in the following form:

$$\overset{\frown}{\Phi}{}_{ij}^{-} \geq 0 \text{ or } \overset{\frown}{\Phi}{}_{ij}^{\prime -} \geq 0 \Rightarrow \operatorname{dist}(A_i, A_j) \geq \rho_{ij}^{-}, i > j \in I^k, k = 1, 2, \ldots, m,$$

$$\overset{\frown}{\Phi}{}_{ij}^{+} \geq 0 \text{ or } \overset{\frown}{\Phi}{}_{ij}^{\prime +} \geq 0 \Rightarrow \operatorname{dist}(A_i, A_j) \leq \rho_{ij}^{+}, i > j \in I^k, k = 1, 2, \ldots, m,$$

$$dist(A_i, \Omega^*) > \rho_i^{-} \Rightarrow \overset{\frown}{\Phi}{}_{i}^{-} \geq 0,\ i = 1, \ldots, n.$$

Now we introduce two functions

$$\Upsilon_1(u, u') = \min\{\Upsilon_{ij}^{-}, (i,j) \in \Xi_{-}^{k}, \Upsilon_{ij}^{+}, (i,j) \in \Xi_{+}^{k}, k = 1, 2, \ldots, m\} \tag{1}$$

where $u = (p, u_1, u_2, \ldots, u_n)$, $u' = (u_{ij}^{\prime -}, (\mathrm{i}, \mathrm{j}) \in \Xi_{-}^{k}, u_{ij}^{\prime +}, (\mathrm{i}, \mathrm{j}) \in \Xi_{+}^{k}, k = 1, 2, \ldots, m)$,

$$\Xi_{-}^{k} = \{(i,j) : |z_i - z_j| < h_i + h_j + \rho_{ij}^{-}, i > j \in I^k\},\ \Xi_{+}^{k} = \{(i,j) : \rho_{ij}^{+} < \varpi, i > j \in I^k\},$$

$$\Upsilon_{ij}^{-} \in \{\overset{\frown}{\Phi}{}_{ij}^{-}, \overset{\frown}{\Phi}{}_{ij}^{\prime -}\}, (i,j) \in \Xi_{-}^{k},\ \Upsilon_{ij}^{+} \in \{\overset{\frown}{\Phi}{}_{ij}^{+}, \overset{\frown}{\Phi}{}_{ij}^{\prime +}\}, (i,j) \in \Xi_{+}^{k},$$

and

$$\Upsilon_2(u) = \min\{\overset{\frown}{\Phi}{}_{i}^{-}, i \in I^k, k = 1, 2, .., m\}. \tag{2}$$

Then the inequality

$$\Upsilon(u, u') = \min\{\Upsilon_1(u, u'), \Upsilon_2(u)\} \geq 0 \tag{3}$$

describes placement constraints in the BLP problem.

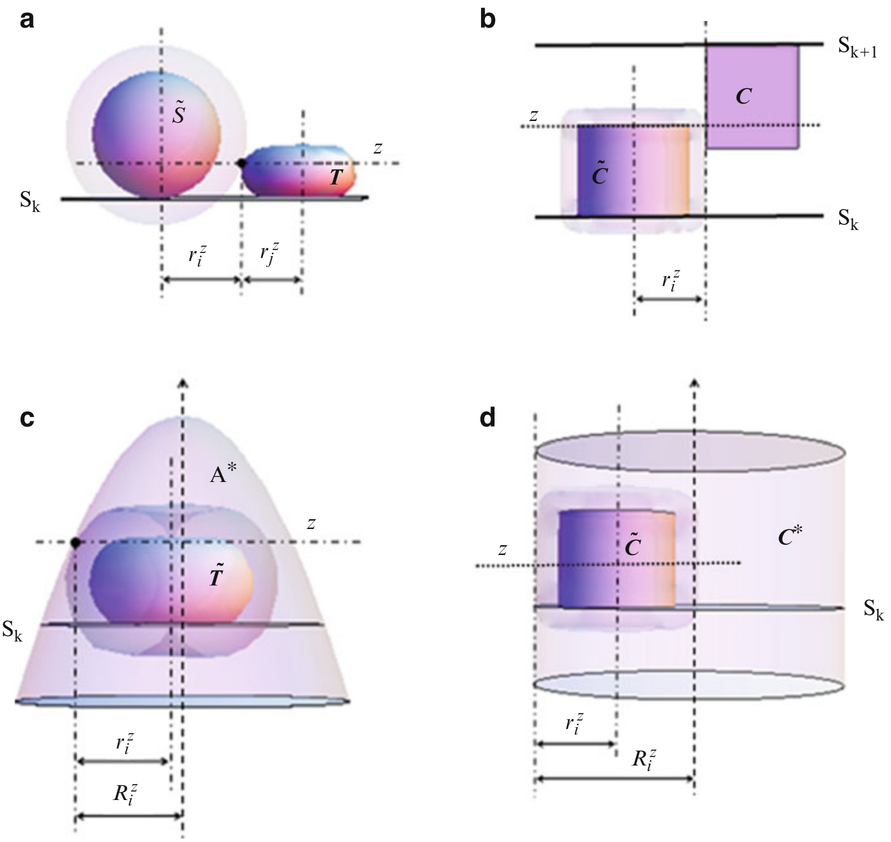


Fig. 4 Contact of objects → $\widehat{A}_i$ and A_j: (**a**) contact of → $\widehat{\mathbb{S}}$ and $\mathbb{T}$, (**b**) contact of → $\widehat{\mathbb{C}}$ and $\mathbb{C}$, (**c**) contact of → $\widehat{\mathbb{T}}$ and $\mathbf{\Lambda}^*$, (**d**) contact of → $\widehat{\mathbb{C}}$ and $\mathbf{C}^*$

Below we define the explicit forms of our adjusted *phi*-functions and adjusted quasi-*phi*-functions involved in (3).

Modeling of constraints on minimal allowable distances between placement objects. To construct adjusted *phi*-functions $\widehat{\Phi}^{-}_{ij}$ and adjusted quasi-*phi*-functions $\widehat{\Phi}'^{-}_{ij}$ for objects A_i and A_j, $(i, j) \in \Xi^k_-$, in (1) we derive $z_{ij} = z$ of a point of contact for objects $\widehat{A}_i = A_i \oplus S(\rho^-_{ij})$ and A_j, where $S(\rho^-_{ij})$ is a solid sphere of radius ρ^-_{ij}, $\oplus$ is a symbol of Minkovski sum. There are two cases of contact: (a) a unique point of contact (Fig. 4a), (b) a continuum of contact points (Fig. 4b). In case (b) we set $z_{ij} = \min\{z_i + h_i, z_j + h_j\}$.

(1) Let $A_i, A_j \in \{\mathbb{S}, \mathbb{C}, \mathbb{T}, \mathbb{S}_{\mathbb{C}}\}$. We consider cross-sections of objects $\widehat{A}_i$ and A_j by plane Oxy provided that $z_{ij} = z$. We denote the radius of the section of object $\widehat{A}_i$

by r_i^z, and the radius of the section of object A_j by r_j^z. Then distance constraint $\mathrm{dist}(A_i, A_j) \geq \rho_{ij}^-$, can be described by means of an adjusted *phi*-function of the form

$$\overset{\frown}{\Phi}{}_{ij}^{-}(u_i, u_j) = (x_j - x_i)^2 + (y_j - y_i)^2 - (r_i^z + r_j^z)^2. \tag{4}$$

(2) Let $A_i, A_j \in \{\mathbb{K}\}$. Now we consider two objects $\overset{\frown}{\mathbb{K}}_i = \mathbb{K}_i \oplus S(\rho_{ij}^-)$ and $\mathbb{K}_j$, and the appropriate cross-sections of the objects by plane *Oxy* provided that $z_{ij} = z$. We denote the appropriate cross-sections by $\overset{\frown}{K}_i = K_i \oplus C(\rho_{ij}^{-z})$ and K_j. It should be noted that if objects $\mathbb{K}_i$ and $\mathbb{K}_j$ have a continuum of contact points then $\rho_{ij}^{-z} = \rho_{ij}^-$. Then distance constraint $\mathrm{dist}(\mathbb{K}_i, \mathbb{K}_j) \geq \rho_{ij}^-$ can be described by means of an adjusted quasi-*phi*-function of the form

$$\overset{\frown}{\Phi}{}_{ij}^{\prime -}(u_i, u_j, u_P) = \min\{\Phi^{K_iP}(u_i, u_P), \Phi^{K_jP^*}(u_j, u_P)\} - 0.5\rho_{ij}^{-z}, \tag{5}$$

where $\Phi^{K_iP}(u_i, u_P) = \min\limits_{1 \leq l \leq s_i} \psi_P(\tilde{v}'_{il})$ is a *phi*-function of K_i and halfplane $P(u_P)$, $\Phi^{K_jP^*}(u_j, u_P) = \min\limits_{1 \leq 1 \leq s_j} (\psi_P(\tilde{\mathrm{v}}'_{\mathrm{j1}}))$ is a *phi*-function of K_j and halfplane $P^*(u_P) = \boldsymbol{R}^2 \backslash \mathrm{int} P(u_P)$, K_i and K_j are bases of $\mathbb{K}_i$ and $\mathbb{K}_j$. Here and after we set $P(u_P) = \{(x, y) : \psi_P = \alpha \cdot x + \beta \cdot y + \gamma_P \leq 0\}$, $u_P = (\theta_P, \gamma_P)$, $\alpha = \cos\theta_P$, $\beta = \sin\theta_P$. Here and after $\tilde{v}'_{il} = (\tilde{x}'_{il}, \tilde{y}'_{il})$, $l = 1, \ldots, s_i$, $\tilde{v}'_{jl} = (\tilde{x}'_{jl}, \tilde{y}'_{jl})$, $l = 1, \ldots, s_j$, $\tilde{x}'_{il} = x_i + \tilde{x}_{il}\cos\theta_i + \tilde{y}_{\sin\theta_i}$, $\tilde{y}'_{il} = y_i - \tilde{x}_{il}\sin\theta_i + \tilde{y}_{il}\cos\theta_i$, $\tilde{x}'_{jl} = x_j + \tilde{x}_{jl}\cos\theta_j + \tilde{y}_{\sin\theta_j}$, $\tilde{y}'_{jl} = y_j - \tilde{x}_{jl}\sin\theta_j + \tilde{y}_{jl}\cos\theta_j$.

(3) Let $A_i \in \{\mathbb{S}, \mathbb{C}, \mathbb{T}, \mathbb{S}_{\mathbb{C}}\}$, $A_j \in \{\mathbb{K}\}$. Now we consider two objects $\rightarrow \overset{\frown}{A}_i$ and $\mathbb{K}_j$ and the appropriate cross-sections of the objects by plane *Oxy* provided that $z_{ij} = z$. We denote the radius of the cross-section of $\rightarrow \overset{\frown}{A}_i$ by r_i^z. Then distance constraint $\mathrm{dist}(A_i, \mathbb{K}_j) \geq \rho_{ij}^-$ can be described by means of an adjusted quasi-*phi*-function of the form

$$\overset{\frown}{\Phi}{}_{ij}^{\prime -}(u_i, u_j, u_P) = \min\{\Phi^{KP}(u_j, u_P), \overset{\frown}{\Phi}{}^{CP^*}(u_i, u_P)\}, \tag{6}$$

where $\Phi^{KP}(u_j, u_P) = \min\limits_{1 \leq l \leq s_j} \psi_P(\tilde{v}'_{jl})$ is a *phi*-function of K_j and P, $\overset{\frown}{\Phi}{}^{CP^*}(u_i, u_P) = -\psi_P(u_i) - r_i^z$ is a *phi*-function of $\overset{\frown}{C}_i$ and P^*, K_j is the base of $\mathbb{K}_j$.

Modeling of constraints on maximal allowable distances between placement objects. In order to describe maximal allowable distances in (1) we use adjusted *phi*-functions $\overset{\frown}{\Phi}{}_{ij}^{+}$, or adjusted quasi-*phi*-functions $\overset{\frown}{\Phi}{}_{ij}^{\prime +}$ for objects A_i and A_j, $(i, j) \in \Xi_+^k$.

Let $A_i, A_j \in \{\mathbb{S}, \mathbb{C}, \mathbb{T}, \mathbb{S}_{\mathbb{C}}, \mathbb{K}\}$. We denote the appropriate cross-sections of objects $\overset{\frown}{A}_i = A_i \oplus S(\rho_{ij}^{+})$ and A_j with plane Oxy provided that $z_{ij} = z$ in the point of contact by $\overset{\frown}{A_i^z}$ and A_j^z .

Then distance constraint $\mathrm{dist}(A_i, A_j) \leq \rho_{ij}^{+}$ can be defined by means of the following functions:

- if $\overset{\frown}{A_i^z} \equiv C_i^z, A_j^z \equiv C_j^z$ then we use an adjusted *phi*-function of the form

$$\overset{\frown}{\Phi}{}_{ij}^{+}(u_i, u_j) = -(x_j - x_i)^2 - (y_j - y_i)^2 + (r_i^z + r_j^z)^2, \tag{7}$$

where r_i^z, r_j^z are radii of circles C_i^z and C_j^z.

- if $\overset{\frown}{A_i^z} \equiv \overset{\frown}{K_i^z} = K_i \oplus C(\rho_{ij}^{+z}), A_j^z \equiv K_j^z$ then we use an adjusted quasi-*phi*-function of the form

$$\overset{\frown}{\Phi}{}_{ij}^{\prime +}(u_i, u_j, u') = \min\{(\rho_{ij}^{+z})^2 - dist^2(p_i, p_j), f_i(p_i), f_j(p_j)\}, \tag{8}$$

where $u' = (p_i, p_j), p_i = (x_{p_i}, y_{p_i}), p_j = (x_{p_j}, y_{p_j})$,

$$\mathrm{dist}^2(p_i, p_j) = (x_{p_j} - x_{p_i})^2 + (y_{p_j} - y_{p_i})^2,$$

$$f_i(p_i) = \min\{\chi'_{il}(p_i), l = 1, \ldots, s_i\}, \quad \chi'_{il}(p_i) = A'_{il}(x_{p_i}) - B'_{il}(y_{p_i}) + C'_{il},$$
$$A'_{il} = \tilde{y}'_{i(l+1)} - \tilde{y}'_{il}, \quad B_{il} = \tilde{x}'_{i(l+1)} - \tilde{x}'_{il}, \quad C'_{il} = \tilde{y}'_{il} \cdot \tilde{x}'_{i(l+1)} - \tilde{y}'_{i(l+1)} \cdot \tilde{x}'_{il},$$
$$f_j(p_j) = \min\{\chi'_{jl}(p_j), l = 1, \ldots, s_j\}, \quad \chi'_{jl}(p_j) = A'_{jl}(x_{p_j}) - B'_{jl}(y_{p_j}) + C'_{jl},$$
$$A'_{jl} = \tilde{y}'_{j(l+1)} - \tilde{y}'_{jl}, \quad B_{jl} = \tilde{x}'_{j(l+1)} - \tilde{x}'_{jl}, \quad C'_{jl} = \tilde{y}'_{jl} \cdot \tilde{x}'_{j(l+1)} - \tilde{y}'_{j(l+1)} \cdot \tilde{x}'_{jl},$$

$p_i = (x_{p_i}, y_{p_i}), p_j = (x_{p_j}, y_{p_j}) \in \mathbb{R}^2$, such that $f_i(p_i) \geq 0$ ($f_j(p_j) \geq 0$), if $p_i \in A_i^z$ ($p_j \in A_j^z$) (and $f_i(p_i) < 0$ ($f_j(p_j) < 0$) otherwise), $\chi'_{il} = 0$ ($\chi'_{jl} = 0$) are the equations of straightlines, passing through vertices $\tilde{v}'_{il}$ and $\tilde{v}'_{i(l+1)}$ of polygon K_i, $l = 1, 2, \ldots, s_i$ (or vertices $\tilde{v}'_{jl}$ and $\tilde{v}'_{j(l+1)}$ of polygon K_j, $l = 1, 2, \ldots, s_j$). Here K_i (K_j) is the base of $\mathbb{K}_i$ ($\mathbb{K}_j$);

- if $\overset{\frown}{A_j^z} \equiv C_j, A_i^z \equiv K_i$ then we use a quasi-*phi*-function of the form

$$\overset{\frown}{\Phi}{}_{ij}^{\prime +}(u_i, u_j, u') = \min\{f_{ij}(p_i, u_j), f_i(p_i)\}, \tag{9}$$

where $u' = (p_i), p_i = (x_{p_i}, y_{p_i}), f_{ij}(p_i, u_j) = -(x_j - p_{xi})^2 - (y_j - p_{yi})^2 + (r_j^z)^2$, r_j^z is the radius of C_j^z, and function $f_i(p_i)$is defined in (8).

Modelling containment constraints taking into account minimal allowable distances. To describe function (2) we use adjusted *phi*-functions $\overset{\frown}{\Phi}{}_i^{-}$ for objects A_i, $i \in I^k$, and Ω^{*k}.

(1) Let $A_i \in \{\mathbb{S},\ \mathbb{C},\ \mathbb{T},\ \mathbb{S}_{\mathbb{C}}\}$, $\overset{\frown}{A}_i = A_i \oplus S(\rho_i^-)$. We consider cross-section C^* of subcontainer Ω^k and cross-section $\overset{\frown}{A}_i$ of object $\overset{\frown}{A}_i$ by plane *Oxy*, provided that $z = z_i^{\Omega} = const$. The value of z_i^{Ω} is defined explicitly by metric characteristics of $\overset{\frown}{A}_i$ and Ω^k, if $\Omega \in \{\mathbf{\Lambda}, \mathbf{E}\}$ (Fig. 4c). We set $z_i^{\Omega} = z_i$ if $\Omega \in \{\mathbf{C}\}$ (Fig. 4d). Then a containment of object A_i in subcontainer Ω^k taking into account minimal allowable distance ρ_i^- we define, using the following adjusted *phi*-function

$$\overset{\frown}{\Phi}_i(u_i) = -x_i^2 - y_i^2 + (R_i^z - r_i^z)^2, \tag{10}$$

where r_i^z is the radius of $\overset{\frown}{A}_i$, $R_i^z \geq r_i^z$ is the radius of C*.

(2) Let $A_i \in \{\mathbb{K}\}$. We consider cross-section $\overset{\frown}{A}^*$ of object $\overset{\frown}{\Omega}^k = \mathbb{R}^3 \setminus int\overset{\frown}{\Omega}^{k*}$ and cross-section K_i of $\mathbb{K}_i$ by plane *Oxy*, provided that $z_{ij} = z$, where $\overset{\frown}{\Omega}^{k*} = \Omega^{k*} \oplus S(\rho_i^-)$, $\Omega^{k*} = \mathbb{R}^3 \setminus int\Omega^k$. Then a containment of object $\mathbb{K}_i$ in subcontainer Ω^k taking into account minimal allowable distance ρ_i^- we define, using the following adjusted *phi*-function

$$\overset{\frown}{\Phi}_i(u_i) = \min\{-(\tilde{x}'_{il})^2 - (\tilde{y}'_{il})^2 + (R_i^z)^2, l = 1, \ldots, s_i\}, \tag{11}$$

where $(\tilde{x}'_{il}, \tilde{y}'_{il})$, $l = 1, \ldots, s_i$, are coordinates of vertices of K_i and R_i^z is the radius of $\overset{\frown}{C}^*$.

2.3 Behaviour Constraints

Let m_0 be the mass of container Ω (we neglect masses of racks and the base of the container). We denote the center of mass of Ω in the fixed coordinate system *Oxyz* by (x_0, y_0, z_0). Assume that the density of the lateral surface of Ω is a constant. For each considered type of our container Ω the point (x_0, y_0, z_0) belongs to axis *Oz* of its symmetry, therefore $x_0 = 0$, $y_0 = 0$ and z_0 is defined as the following:

$$z_0 = \frac{H}{2} \text{ for } \mathbf{C}, z_0 = \frac{2H}{5} \text{ for } \mathbf{\Lambda}, z_0 = \frac{H}{3}\frac{R_1 + 2R_2}{R_1 + R_2} \text{ for } \mathbf{E}.$$

The center of mass of each object A_i is at origin O_i of the coordinate system of object A_i.

We denote the center of mass of system Ω_A by $O_s = (x_s, y_s, z_s)$, where

$$x_s(u) = \frac{1}{M}\sum_{i=1}^{n} m_i x_i, \quad y_s(u) = \frac{1}{M}\sum_{i=1}^{n} m_i y_i, \quad z_s(u) = \frac{1}{M}\sum_{i=1}^{n} m_i z_i,$$

$M = \sum_{i=0}^{n} m_i$ is the mass of system Ω_A.

Now we define coordinate system O_sXYZ of Ω_A. The origin of the system is at O_s and $O_sX\| Ox$, $O_sY\| Oy$, $O_sZ\| Oz$ (Fig. 3.).

Let us consider the constraints of mechanical characteristics of system Ω_A.

The equilibrium constraints are defined by the following system of inequalities:

$$\mu_{11}(u) = \min\{-(x_s(u) - x_e) + \Delta x_e, (x_s(u) - x_e) + \Delta x_e\} \geq 0,$$

$$\mu_{12}(u) = \min\{-(y_s(u) - y_e) + \Delta y_e, (y_s(u) - y_e) + \Delta y_e\} \geq 0,$$

$$\mu_{13}(u) = \min\{-(z_s(u) - z_e) + \Delta z_e, (z_s(u) - z_e) + \Delta z_e\} \geq 0,$$

where (x_e, y_e, z_e) is the expected position of O_s, $(\Delta x_e, \Delta y_e, \Delta z_e)$ are admissible deviations from the point (x_e, y_e, z_e).

The constraints of moments of inertia are defined as the following:

$$\mu_{21}(u) = -J_X(u) + \Delta J_X \geq 0,$$

$$\mu_{22}(u) = -J_Y(u) + \Delta J_Y \geq 0,$$

$$\mu_{23}(u) = -J_Z(u) + \Delta J_Z \geq 0,$$

where $J_X(u), J_Y(u), J_Z(u)$ are the moments of inertia of the system Ω_A with respect to the axes of coordinate system O_sXYZ, ΔJ_X, ΔJ_Y, ΔJ_Z are admissible values for $J_X(u), J_Y(u), J_Z(u)$, where

$$J_X(u) = J_{x_0} + \sum_{i=1}^{n} (J_{x_i} \cos^2 \theta_i + J_{y_i} \sin^2 \theta_i) + \sum_{i=1}^{n} (y_i^2 + z_i^2)m_i - M(y_s^2 + z_s^2),$$

$$J_Y(u) = J_{y_0} + \sum_{i=1}^{n} (J_{x_i} \sin^2 \theta_i + J_{y_i} \cos^2 \theta_i) + \sum_{i=1}^{n} (x_i^2 + z_i^2)m_i - M(x_s^2 + z_s^2),$$

$$J_Z(u) = \sum_{i=0}^{n} J_{z_i} + \sum_{i=1}^{n} (y_i^2 + z_i^2)m_i - M(x_s^2 + y_s^2),$$

$J_{x_0}, J_{y_0}, J_{z_0}$ are the moments of inertia of container Ω with respect to the axes of the coordinate system $Oxyz$, $J_{x_i}, J_{y_i}, J_{z_i}$, $i \in I_n$, are the moments of inertia of object $\acute{A}_i$ with respect to the axes of coordinate system $O_ix_iy_iz_i$ (see Appendix 2).

The stability constraints are defined by the following system of inequalities:

$$\mu_{31}(u) = \min\{-J_{XY}(u) + \Delta J_{XY}, J_{XY}(u) + \Delta J_{XY}\} \geq 0,$$

$$\mu_{32}(u) = \min\{-J_{YZ}(u) + \Delta J_{YZ}, J_{YZ}(u) + \Delta J_{YZ}\} \geq 0,$$

$$\mu_{33}(u) = \min\{-J_{XZ}(u) + \Delta J_{XZ}, J_{XZ}(u) + \Delta J_{XZ}\} \geq 0,$$

where $J_{XY}(u), J_{YZ}(u), J_{XZ}(u)$ are the products of inertia of system Ω_A with respect to the axes of the coordinate system O_sXYZ, ΔJ_{XY}, ΔJ_{YZ}, ΔJ_{XZ} are admissible values for $J_{XY}(u), J_{YZ}(u), J_{XZ}(u)$, respectively,

$$J_{XY}(u) = \frac{1}{2}\sum_{i=1}^{n}(J_{x_i} - J_{y_i})\sin 2\theta_i + \sum_{i=1}^{n} x_i y_i m_i - Mx_s y_s,$$

$$J_{YZ}(u) = \sum_{i=1}^{n} y_i z_i m_i - My_s z_s, \quad J_{XZ}(u) = \sum_{i=1}^{n} x_i z_i m_i - Mx_s z_s.$$

Behaviour constraints of the BLP problem we define as the system of inequalities

$$\mu_1(u) \geq 0, \mu_2(u) \geq 0, \mu_3(u) \geq 0,$$

where

$$\mu_1(u) = \min\{\mu_{11}(u), \mu_{12}(u), \mu_{13}(u)\}, \tag{12}$$

$$\mu_2(u) = \min\{\mu_{21}(u), \mu_{22}(u), \mu_{23}(u)\}, \tag{13}$$

$$\mu_3(u) = \min\{\mu_{31}(u), \mu_{32}(u), \mu_{33}(u)\}. \tag{14}$$

3 A Mathematical Model and a Solution Strategy

Using (1)–(14), a mathematical model of the BLP problem can be presented in the form

$$\min F(u) \text{ s.t. } (u, u') \in W \tag{15}$$

$$W = \{(u, u') \in \mathbb{R}^{\xi} : \Upsilon(u, u') \geq 0, \mu(u) \geq 0, \zeta \geq 0\}, \tag{16}$$

where $\Upsilon(u, u')$ is defined in (3), $\mu(u) = \min\{\mu_s(u), s \in U_t\}$, $U_t \in P(U)$, $P(U)$ is the power set of $U = \{1, 2, 3\}$, functions $\mu_1(u), \mu_2(u), \mu_3(u)$ are given in (12)–(14), $\zeta \geq 0$ is the system of additional constraints of metric characteristics of container Ω and placement parameters of objects. If $s = \emptyset$, i.e. behaviour constraints are not involved in (16), then our objective function $F(u)$ meets mechanical characteristics of system Ω_A.

Depending on the form of objective function different variants of mathematical model (15) and (16) can be generated. The most frequently occurring objective functions found in related publications are the following: (1) size of container Ω; (2) deviation of the center of mass of system Ω_A from a given point; (3) moments of inertia of system Ω_A (see, e.g., [3–8]).

Problem (15) and (16) for a given U_t is a multiextremal nonlinear programming problem. Feasible region W is described by the system of N inequalities with nonsmooth functions, where $N = N_a + N_b$, N_a is the number of adjusted *phi*-functions and adjusted quasi-*phi*-functions for non-overlapping constraints, $N_a = \sum_{k=1}^{m} N_k$, N_k is the number of adjusted *phi*-functions and adjusted quasi-*phi*-functions for containment constraints, $N_k = n^{2k}$, $k = 1, 2, \dots, m$, N_b is the number of functions for behaviour constraints, $N_b \leq 15$. The adjusted *phi*-functions and adjusted quasi-*phi*-functions in (3) are composed generally of min- and max-operations of nonlinear functions. As a result, set W of feasible solutions is non-convex, leading to many local extrema.

One of the important features of our feasible region (16) is the following: $W = W_1 \cup \dots \cup W_s \cup \dots \cup W_\tau$, where each subregion W_s is specified by a system of inequalities with differentiable functions (see, e.g., [2, 10])

Problem (15) and (16) can be reduced to the following optimization problem:

$$F(u^*, u'^*) = \min\{F(u^{s*}, u'^{s*}), s = 1, 2, \dots, \tau\}, \tag{17}$$

where

$$F(u^{s*}, u'^{s*}) = \min F(u, u') \text{ s.t. } (u, u') \in W_s \subset \mathbb{R}^{\xi}. \tag{18}$$

The model requires a comprehensive search for local extrema on all subregions and provides the global minimum provided each subproblem (18) can be solved optimally. Subproblems (18) are nonlinear programming problems and they may be directly solved by means of global NLP-solvers (at least theoretically).

Based on the features of adjusted *phi*-functions and adjusted quasi-*phi*-functions defined in (4)–(11), and the forms of our functions for behaviour constraints (12)–(14) the feasible region (16) can be described by a system of inequalities with differentiable functions, i.e.$\tau = 1$.

Problem (15) and (16) can also be transformed into the nonconstrained optimization problem of the form

$$\min f(u, u'), \tag{19}$$

where $f(u, u')$ is the almost everywhere differentiable function

$$f(u, u') = F(u) + P_1 \sum_{l=1}^{N_a} \max\{0, -\Phi_l(u, u')\} + P_2 \sum_{k=1}^{N_b} \max\{0, -\mu_k(u)\} + P_3 \max\{0, -p + p_{low}\},$$

$P_i, i = 1, 2, 3$, are penalty coefficients, Φ_l, $l = 1, \dots, N_a$, are *phi*-functions from (1) and (2), μ_k, $k = 1, \dots, N_b$, are functions of the form (12)–(14), p_{low} is the evident lower bound of variable metrical characteristic p of container Ω.

To solve problem (15) and (16) we apply a multi-start strategy, which involves the following procedures:

- generation of starting points;
- solving nonlinear optimization problem (18) (or problem (19)) for each starting point obtained at the previous step;
- selection of the best of local minima obtained at the previous step as a local-optimal solution of problem (15) and (16).

To solve our NP-hard constrained optimization problem (18) we combine our strategy with a clever choice of feasible starting points and our special local NLP solver. We develop special fast algorithms (BSPA) to construct feasible starting points depending on the form of objective function $F(u)$ and types of constraints used in (16). It should be noted that the finding of a feasible starting point by BSPA is granted only when every nonlinear subproblem, involved in BSPA, is solved optimally (with using global NLP-solvers).

In order to reduce computational costs (time and memory) we employ a modification of LOFRT algorithm proposed in [2, 13]. The algorithm allows us to reduce large dimension problem (18) with a large number of inequalities to sequence subproblems with a considerably smaller number of variables and inequalities.

We apply IPOPT for solving nonlinear programming problems in our algorithms, which is available in the open access noncommercial software depository (https://projects.coin-or.org/Ipopt) and is based on the interior point method described in [14].

To solve nonconstrained nonlinear optimization problem (19) we use randomly generated starting points. To search for local minima of the almost everywhere differentiable function $f(u)$ we employ the nonsmooth optimization method based on Shor's r-algorithm [15, 16] (see Appendix 3) and program ralgb5 [16].

4 Variants of BLP Problems

Here we formulate some types of the BLP problem, provide their mathematical models as realizations of problem (15) and (16), develop the appropriate algorithms to generate feasible starting points and provide computational results for each realization. To search for feasible starting points we use homothetic transformations of our objects, assuming that object homothetic coefficients $\lambda_i = \lambda$, $0 \leq \lambda \leq 1$, $i \in I_n$, are variable.

4.1 BLP1 Problem

Place a family of cylinders $\mathbb{C}_i$ of the same height $2h_i$ and different radii r_i, $i \in I_n$, into container $\Omega \equiv \mathbf{C}$ of variable radius R and given height $H = 2h_i$, taking into account the behaviour constraints. Here we minimize R.

Since $H = 2h_i$ and $z_i = 0$, $i \in I_n$, then $O_s = (x_s(u), y_s(u), 0)$. Assume that the origin of system $Oxyz$ of Ω is located in the center of its symmetry, $(x_0, y_0, z_0) = (0, 0, 0)$ and $m_0 = 0$.

Taking into account the peculiatrities of BLP1 problem, axial and axifugal moments of inertia of system Ω_A in (13), (14) can be simplified as the following:

$$J_X(v) = \frac{1}{12}\sum_{i=1}^{n} m_i(3r_i^2 + H^2) + \sum_{i=1}^{n} y_i^2 m_i – M(y_s(v))^2,$$

$$J_Y(v) = \frac{1}{12}\sum_{i=1}^{n} m_i(3r_i^2 + H^2) + \sum_{i=1}^{n} x_i^2 m_i - M(x_s(v))^2,$$

$$J_Z(v) = \frac{1}{2}\sum_{i=1}^{n} m_i r_i^2 + \sum_{i=1}^{n} (x_i^2 + y_i^2)m_i - M((x_s(v))^2 + (y_s(v))^2), \tag{20}$$

$$J_{XY}(v) = \sum_{i=1}^{n} x_i y_i m_i - M x_s(v) y_s(v), \quad J_{XZ}(v) = 0, \quad J_{YZ}(v) = 0. \tag{21}$$

Now mathematical model (15) and (16) for BLP1 problem takes the form

$$\min R \text{ s.t. } u \in W,$$

$$W = \{u \in \mathbb{R}^{2n+1} : \Upsilon_1(v) \geq 0, \Upsilon_2(u) \geq 0, \mu(v) \geq 0, \zeta \geq 0\},$$

where $u = (R, v)$, $v = (v_1, \ldots, v_n)$, $\Upsilon_1(v)$ is defined by (1), provided that $k = 1$, $\rho_{ij}^- = 0$, $\rho_{ij}^+ = \varpi$ (i.e. $\Xi_-^1 = \{(i,j) : i < j \in I^1 = \{1, 2, \ldots, n\}\}$, $\Xi_+^1 = \{\emptyset\}$), $\Upsilon_2(u) = \min\{\Upsilon_{i2}(R_i^z = R, v_i), i \in I_n\}$, provided that $\rho_i^- = 0$, $U_t \in P(U)\backslash\emptyset$, $\mu_1(v), \mu_2(v), \mu_3(v)$ are defined in (12)–(14), taking into account (20) and (21), $\zeta = R - \max_{i=1,\ldots,n} r_i$.

Feasible starting point algorithm (BSPA1) for BLP1 problem. BSPA1 algorithm involves the following steps.

Step 1. Set sufficiently great starting value of radius $R = R^0$ of our container (e.g., $R^0 = R_{up} = \sum_{i=1}^{n} r_i$).

Step 2. Generate a collection of random points $v_i^0 = (x_i^0, y_i^0) \in \Omega^0$, $i \in I_n$. Form vector $v^0 = (x_1^0, y_1^0, \ldots, x_n^0, y_n^0)$.

Step 3. Take feasible starting point and solve the following auxiliary nonlinear problem:

$$\lambda^* = \max \lambda, \text{ s.t. } v_\lambda \in W_\lambda \tag{22}$$

$$W_\lambda = \{v_\lambda \in \mathbb{R}^{2n+1} : \Upsilon_1(v_\lambda) \geq 0, \Upsilon_2(R_{up}, v_\lambda) \geq 0, 1 - \lambda \geq 0, \lambda \geq 0\}, \tag{23}$$

where $v_\lambda = (v, \lambda)$, $\Upsilon_1(v_\lambda)$ and $\Upsilon_2(R_{up}, v_\lambda)$ are defined by analogy with functions $\Upsilon_1(u)$, $\Upsilon_2(u)$ in (1), (2), taking into account variable homothetic coefficient λ under $\theta_i = 0, i \in I_n$. We denote a point of the global maximum of problem (22) and (23) by $v_\lambda^* = (v^*, \lambda^* = 1)$.

Step 4. Derive $\mu(v^*)$. If $\mu(v^*) < 0$, then go to Step 5, otherwise go to Step 6.

Step 5. Take feasible starting point and solve the following auxiliary nonlinear problem:

$$\alpha^* = \max \alpha, \text{ s.t. } v_\alpha \in W_\alpha, \tag{24}$$

$$W_\alpha = \{v_\alpha \in \mathbb{R}^{2n+1} : \Upsilon_1(v) \geq 0, \Upsilon_2(R_{up}, v) \geq 0, \mu(v) - \alpha \geq 0, -\alpha \geq 0\}, \tag{25}$$

where α is the auxiliary discrepancy-variable, $v_\alpha = (v, \alpha)$.

If $\alpha^* = 0$ then point $= (v^*, \alpha^*)$ of the global maximum of problem (24) and (25) is found and we go to Step 5. If $\alpha^* < 0$ then a feasible starting point for BLP1 problem could not be found, since behaviour constraints are "disrupted". In the case we return to Step 2.

Step 6. Form starting feasible point $u^0 = (R_{up}, v^*, \theta^0) \in W$ for BLP1 problem.

Computational experiments for BLP1 problem

Instance 1. Let $H = 1$, $m = 1$, $n = 5$, $A = \{\mathbb{C}_i, i \in I_5\}$, $h_i = 1$, $i \in I_5$, $\{r_i, i \in I_5\}$ =$\{0.1, 0.2, 0.3, 0.5, 0.8\}$, $\{m_i, i \in I_5\}$ =$\{0.0785, 0.314, 0{,}7065, 1.9625, 5{,}024\}$, $\Xi_+^1 = \{\emptyset\}$, $(x_e, y_e, z_e) = (0, 0, 0)$, $(\Delta x_e, \Delta y_e) = (10^{-4}, 10^{-4})$, $(\Delta J_X, \Delta J_Y, \Delta J_Z) = (5, 5, 5)$.

The best local-optimal solution under $U_t = \{1\}$ found both by Shor's $r(\alpha)$-algorithm and IPOPT is $F(u^*) = R^* = 1.316108$ (see, Fig. 5a). The found point u^* is a point of the global minimum (the rigorous proof of the fact one can find in [17]).

The best local-optimal solution under $U_t = \{1, 2\}$ found by IPOPT is $F(u^*) = R^* = 1.362501$ (see, Fig. 5b).

Instance 2. Let $H = 700$, $m = 1$, $n = 40$, $A = \{\mathbb{C}_i, i \in I_{40}\}$, $h_i = 700$, $i \in I_{40}$, $\{r_i, i = 1, \ldots, 40\}$ =$\{106, 112, 98, 105, 93, 103, 82, 93, 117, 81, 89, 92, 109, 104,$

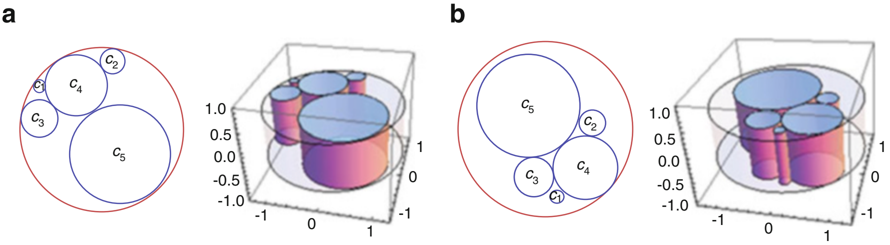


Fig. 5 Local-optimal placements in Instance 1: (**a**) a global-optimal placement under $U_t = \{1\}$, (**b**) a local-optimal placement under $U_t = \{1, 2\}$

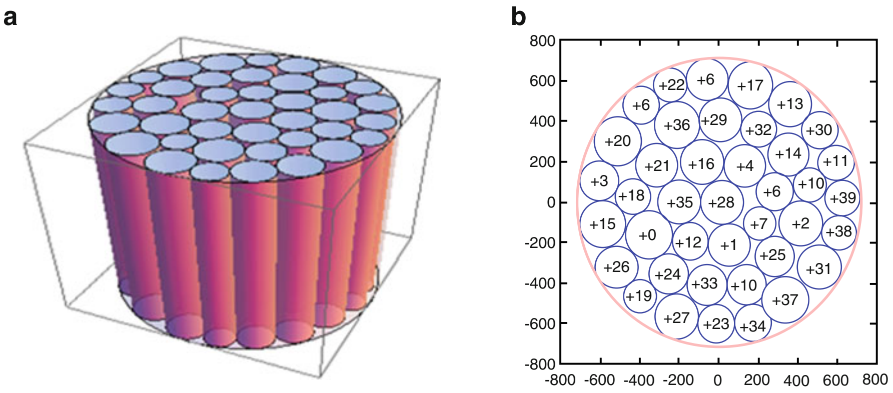


Fig. 6 The local-optimal placement of cylinders in Instance 2: (**a**) system Ω_A, (**b**) the projection of Ω_A on *Oxy*

115, $\{m_i, i = 1, \ldots, 40\} = \{11, 12, 9, 11, 8, 10, 6, 8, 13, 6, 7, 8, 11, 10, 13, 12, 12, 7, 6, 14, 11, 7, 8, 10, 10, 11, 12, 11, 11, 8, 12, 8, 10, 8, 11, 12, 13, 7, 7, 9\}$, $\Xi^1_+ = \{\emptyset\}$, $(x_e, y_e, z_e) = (0, 0, 0)$, $(\Delta x_e, \Delta y_e) = (10^{-7}, 10^{-7})$.

The best local-optimal solution under $U_t = \{1\}$ found by Shor's $r(\alpha)$-algorithm is $F(u^*) = R^* = 711.9522$ (Fig. 6).

4.2 BLP2 Problem

Place a family of cylinders $\mathbb{C}_i$ of height $2h_i$ and different radii r_i, $i \in I_n$, into container $\Omega \in \{\mathbf{C}, \mathbf{\Lambda}, \mathbf{E}\}$ taking into account the behaviour constraints. Here we minimize a deviation of the center of mass of system Ω_A.

Now mathematical model (15) and (16) for BLP2 problem takes the form

$$\min\left((x_s(v))^2 + (y_s(v))^2 + (z_s - z_e)^2\right), \text{ s.t. } v \in W,$$

$$W = \{v \in \mathbb{R}^{2n} : \Upsilon_1(v) \geq 0, \Upsilon_2(v) \geq 0, \mu(v) \geq 0\},$$

where $v = (x_1, y_1, \ldots, x_n, y_n)$, $(x_s(v), y_s(v), z_s)$ is the center of mass of system Ω_A, function $\Upsilon_1(v)$ has form (1) provided that $\rho^-_{ij} = 0$, $\Xi^-_k = \{(i,j) : i < j \in I^k\}$, $\rho^+_{ij} = \varpi$ (i.e. $\Xi^k_+ = \{\emptyset\}$), function $\Upsilon_2(v)$ has form (2) provided that $\rho^-_i = 0$, $\Upsilon_2(v) = \min\{\Upsilon_{i2}(R^z_i = const, v_i), i \in I_n\}$, axial and axifugal moments of inertia of sytem Ω_A are defined by (12)–(14), $U_t \in \{\emptyset, \{2\}, \{3\}, \{2, 3\}\}$.

It should be noted that if the value of the objective function is equal to $(z_s - z_e)^2$ then the optimal solution of BLP2 problem is found.

Feasible starting point algorithm (BSPA2) for BLP2 problem. BSPA2 algorithm involves the following steps.

Step 1. Generate a collection of random points $v_i^0 = (x_i^0, y_i^0)$ belonging to the appropriate cross-section circles of radii R_i^z, $i \in I_n$. Form vector $v^0 = (x_1^0, y_1^0, \ldots, x_n^0, y_n^0)$.

Step 2. Take feasible starting point and solve the following nonlinear auxiliary problem:

$$\lambda^* = \max \lambda, \text{ s.t. } v_\lambda \in W_\lambda, \tag{26}$$

$$W_\lambda = \{v_\lambda \in \mathbb{R}^{2n+1} : \Upsilon_1(v_\lambda) \geq 0, \Upsilon_2(v_\lambda) \geq 0, 1 - \lambda \geq 0, \lambda \geq 0\}, \tag{27}$$

where $v_\lambda = (v, \lambda)$, functions $\Upsilon_1(v_\lambda)$ and $\Upsilon_2(v_\lambda)$ are defined by analogy with functions $\Upsilon_1(u)$ and $\Upsilon_2(u)$ in (1), (2) taking into account homothetic coefficient λ, provided that $\theta_i = 0$, $i \in I_n$. We denote a point of local maxima of problem (26) and (27) by $v_\lambda^* = (v^*, \lambda^*)$. If $\lambda^* = 1$ go to Step 3, otherwise we return to Step 1.

Step 3. Derive $\mu(v^*)$. If $\mu(v^*) < 0$, then go to Step 4, otherwise go to Step 5.

Step 4. Take feasible starting point and solve the following nonlinear auxiliary problem:

$$\alpha^* = \max \alpha, \text{ s.t.} v_\alpha \in W_\alpha, \tag{28}$$

$$W_\alpha = \{v_\alpha \in \mathbb{R}^{2n+1} : \Upsilon_1(v) \geq 0, \Upsilon_2(v) \geq 0, \mu(v) - \alpha \geq 0, -\alpha \geq 0\}, \tag{29}$$

where α is the auxiliary *discrepancy*-variable, $v_\alpha = (v, \alpha)$.

If $\alpha^* = 0$ then point (v^*, α^*) of the global maximum of problem (28) and (29) is found and we go to Step 5. If $\alpha^* < 0$ then a feasible starting point for BLP2 problem could not be found, since the behaviour constraints are "disrupted". In the case we return to Step 1.

Step 5. Form starting feasible point $u^0 = (v^*, \theta^0) \in W$ for BLP2 problem.

Instance 3. Let $\Omega \equiv \mathbf{E}$, $m = 2$, $(v, u') \in W$, $R_1 = 0.5$, $R_2 = 0.3$, $t_1 = 0.3$, $n = 8$, $A = \{\mathbb{C}_i, i = 1, \ldots, 8\}$, $\{r_i, i = 1, \ldots, 8\}$ =$\{$0.1, 0.1, 0.1, 0.075, 0.075, 0.06, 0.05, 0.045$\}$, $\{h_i, i = 1, \ldots, 8\}$ =$\{$0.12, 0.09, 0.1, 0.1, 0.1, 0.075, 0.1, 0.08$\}$, $\{m_i, i \in I_8\}$ =$\{$26.62, 16.97, 18.85, 10.6, 10.6, 5.09, 4.71, 3.05$\}$, $A_-^1 = \{\mathbb{C}_1, \mathbb{C}_2, \mathbb{C}_3, \mathbb{C}_4\}$, $A_+^2 = \{\mathbb{C}_5, \mathbb{C}_6, \mathbb{C}_7, \mathbb{C}_8\}$, $U_t = \{2, 3\}$, $(x_e, y_e, z_e) = (x_0, y_0, z_0) = (0, 0, 0.275)$, $(\Delta J_X, \Delta J_Y, \Delta J_Z) = (5, 5, 5)$, $(\Delta J_{XY}, \Delta J_{YZ}, \Delta J_{XZ}) = (0, 0, 0)$.

The best local-optimal solution found by IPOPT is $F(v^*) = 0.000819642$ (Fig. 7).

Instance 4. Let $\Omega \equiv \mathbf{\Lambda}, H = 70, m = 3, t_1 = 18.5, t_2 = 14, n = 45, A = \{\mathbb{C}_i, i = 1, \ldots, 45\}$, $h_i = 1.85$, $i = 1, \ldots, 45$, $\{r_i, i = 1, \ldots, 45\}$ =$\{$2.0, 2.4, 0.8, 1.1, 1.3, 0.7, 0.7, 1.5, 2.4, 1.8, 1.5, 1.7, 1.7, 1.4, 1.6, 1.8, 0.5, 2.1, 2.1, 1.3, 0.8, 1.4, 0.8, 1.5, 1.1, 1.7, 2.1, 1.6, 0.6, 1.8, 2.4, 1.3, 2.0, 1.0, 1.5, 2.0, 2.2, 1.7, 1.7, 0.7, 2.1, 1.1, 0.5, 2.3, 0.8$\}$, $\{m_i, i = 1, \ldots, 45\}$ =$\{$86, 72, 81, 54, 29, 94, 92, 41, 57, 77, 40, 67, 31, 47, 39, 61, 73, 83, 11, 20, 75, 29, 36, 58, 75, 32, 98, 52, 76, 85, 59, 18, 85, 36, 12,

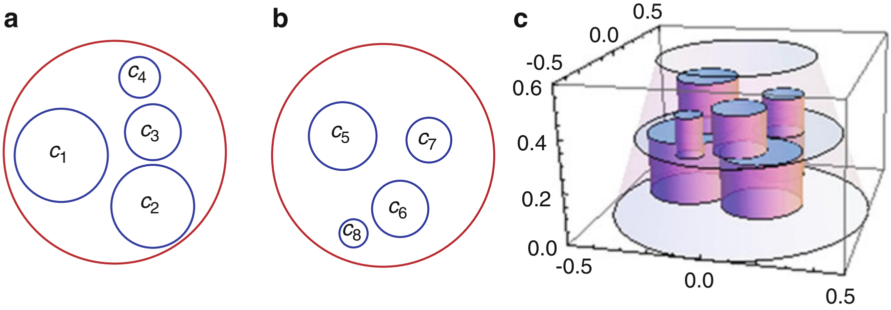


Fig. 7 The local-optimal placement of cylinders in Instance 3: (**a**) the projection of A^1_- on rack S_2, (**b**) the projection of A^2_+ on rack S_2, (**c**) system Ω_A

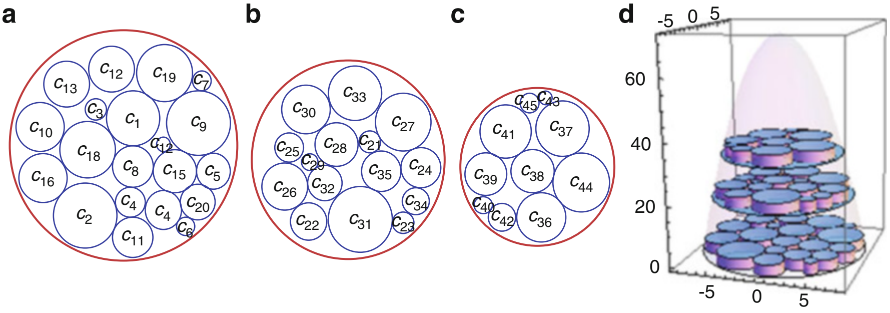


Fig. 8 The optimal placement of cylinders in Instance 4: (**a**) the projection of A^1_+ on rack S_1, (**b**) the projection of A^2_+ on rack S_2, (**c**) the projection of A^3_+ on rack S_3, (**d**) system Ω_A

35, 61, 49, 89, 68, 80, 93, 82, 70, 20}, $\Xi^1_+ = \{\emptyset\}$, $U_t = \emptyset$, $(x_e, y_e, z_e) = (0, 0, z_s)$, $A^1_+ = \{\mathbb{C}_1, \dots, \mathbb{C}_{20}\}$, $\acute{A}^2_+ = \{\mathbb{C}_{21}, \dots, \mathbb{C}_{35}\}$, $\acute{A}^3_+ = \{\mathbb{C}_{36}, \dots, \mathbb{C}_{45}\}$.

The best local-optimal solution found by IPOPT is $F(v^*) = 0$ (Fig. 8).

4.3 BLP3 Problem

Place a family of cylinders $\mathbb{C}_i$ of height $2h_i$ and different radii r_i, $i \in I_n$, into container $\Omega \equiv \mathbf{C}$ under $I^k = I^k_+$, $k = 1, \dots, m$, taking into account the behaviour constraints. Here we minimize both the radius of $\mathbf{C}$ and the deviation of the center of mass of system Ω_A.

Now mathematical model (15) and (16) for BLP3 problem takes the form

$$\min \left(\kappa_1 R + \kappa_2 \left((x_s(u))^2 + (y_s(u))^2 + (z_s - z_e)^2\right)\right), \text{ s.t. } u \in W,$$

$$W = \{u \in \mathbb{R}^{2n+1} : \Upsilon_1(v) \geq 0, \Upsilon_2(u) \geq 0, \mu(u) \geq 0, \zeta \geq 0\},$$

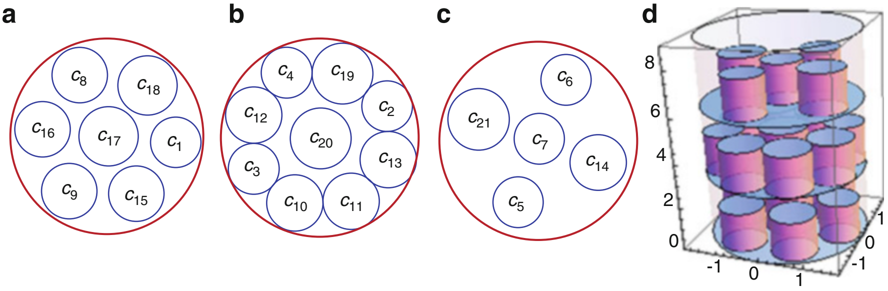


Fig. 9 The local-optimal placement of cylinders in Instance 5: (**a**) the projection of A^1_+ on rack S_1, (**b**) the projection of A^2_+ on rack S_2, (**c**) the projection of A^3_+ on rack S_3, (**d**) system Ω_A

where $v = (x_1, y_1, \ldots, x_n, y_n)$, $(x_s(v), y_s(v), z_s)$ is the center of mass of system Ω_A, κ_1, κ_2 are the given weighting coefficients, function $\Upsilon_1(v)$ is defined by (1) provided that $\rho^-_{ij} = 0$, $\rho^+_{ij} = \varpi$, $\Upsilon_2(R, v) = \min\{\Upsilon_{i2}(R^z_i = R, v_i), i \in I_n\}$ provided that $\rho^-_i = 0$, $\zeta = R - \max\limits_{i=1,\ldots,n} r_i$, $U_t \in \{\emptyset, \{2\}, \{3\}, \{2, 3\}\}$, functions $\mu_2(u)$ and $\mu_3(u)$ are described by (13), (14).

In order to search for feasible starting points for BLP3 problem we employ BSPA1 algorithm. However, at Step 6 we form point $u^0 = (R_{up}, v^*, \theta^0) \in W$ for BLP3 problem.

Instance 5. Let $\Omega \equiv \mathbf{C}$, $m = 3$, $H = 9$, $t_1 = 3$, $t_2 = 3$, $n = 21$, $A = \{\mathbb{C}_i, i = 1, \ldots, 21\}$, $h_i = 0.88$, $i = 1, \ldots, 21$, $A^1_+ = \{\mathbb{C}_1, \mathbb{C}_8, \mathbb{C}_9, \mathbb{C}_{15}, \mathbb{C}_{16}, \mathbb{C}_{17}, \mathbb{C}_{18}\}$, $A^2_+ = \{\mathbb{C}_2, \mathbb{C}_3, \mathbb{C}_4, \mathbb{C}_{10}, \mathbb{C}_{11}, \mathbb{C}_{12}, \mathbb{C}_{19}, \mathbb{C}_{20}\}$, $A^3_+ = \{\mathbb{C}_5, \mathbb{C}_6, \mathbb{C}_7, \mathbb{C}_{14}, \mathbb{C}_{21}\}$, $(x_e, y_e, z_e) = (0, 0, z_s)$, $r_i = 0, 45$, $m_i = 3.1416$, for $i = 1, \ldots, 7$, $r_i = 0, 5$, $m_i = 3.8013$, for $i = 8, \ldots, 14$, $r_i = 0.54$, $m_i = 4.5239$, for $i = 15, \ldots, 21$, $U_t = \emptyset$.

The best local-optimal solution is found by IPOPT under $R^* = 1.7554$ and $(x_s(v^*))^2 + (y_s(v^*))^2 + (z_s - z_e)^2 = 0$ (Fig. 9), i.e. $F(v^*) = 1.7554$

Instance 6. Let $\Omega \equiv \mathbf{C}$, $H = 9$, $m = 2$, $n = 35$, $t_1 = 4$, $\acute{A} = \{\mathbb{C}_i, i = 1, \ldots, 35\}$, $h_i = 1, 85$, $i \in I_{35}$, $(x_e, y_e, z_e) = (0, 0, z_s)$, $U_t = \emptyset$, $A^1_+ = \{\mathbb{C}_i, i = 1, \ldots, 20\}$, $\{r_i, i = 1, \ldots, 35\} =$\{20, 24, 8, 11, 13, 7, 7, 15, 24, 18, 15, 17, 17, 14, 16, 18, 5, 21, 21, 13, 8, 14, 8, 15, 11, 17, 21, 16, 6, 18, 24, 13, 20, 10, 15\}, $\{m_i, i = 1, \ldots, 35\} =$\{86, 72, 81, 54, 29, 94, 92, 41, 57, 77, 40, 67, 31, 47, 39, 61, 73, 83, 11, 20, 75, 29, 36, 58, 75, 32, 98, 52, 76, 85, 59, 18, 85, 36, 12\}.

The best local-optimal solution is found by IPOPT under $R^* = 80.716254$ and $(x_s(v^*))^2 + (y_s(v^*))^2 + (z_s - z_e)^2 = 0$ (Fig. 10), i.e. $F(v^*) = 80.716254$

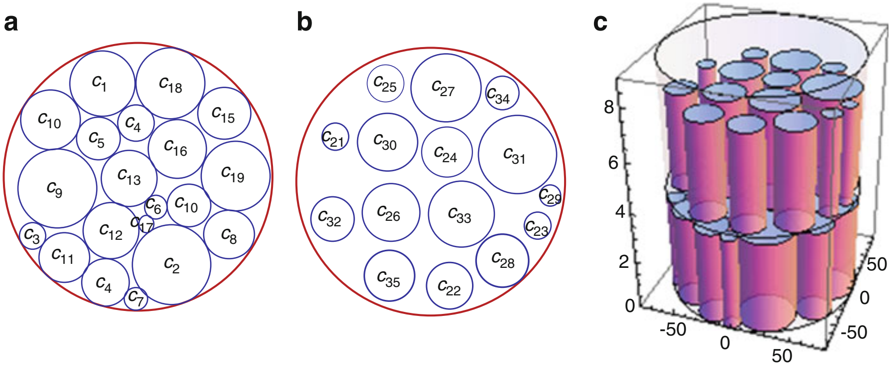


Fig. 10 The local-optimal placement of cylinders in Instance 6: (**a**) the projection of A^1_+ on rack S_1, (**b**) the projection of A^2_+ on rack S_2, (**c**) system Ω_A

4.4 BLP4 Problem

Place a family of 3D-objects (solid spheres, right circular cylinders, tori, spherocylinders, cuboids and right-angle prisms) into container $\Omega \in \{\mathbf{C}, \mathbf{\Lambda}, \mathbf{E}\}$ taking into account minimal and maximal allowable distances and behaviour constraints. Here we minimize a deviation of the center of mass of system Ω_A.

Now mathematical model (15) and (16) for BLP4 problem takes the form

$$\min\left((x_s(v))^2 + (y_s(v))^2 + (z_s - z_e)^2\right), \text{ s.t. } (v, u') \in W,$$

$$W = \{(v, u') \in \mathbb{R}^{\xi} : \Upsilon_1(v, u') \geq 0, \Upsilon_2(v) \geq 0, \mu(v) \geq 0\},$$

where $v = (x_1, y_1, \ldots, x_n, y_n)$, $(x_s(v), y_s(v), z_s)$ is the center of mass of Ω_A, function $\Upsilon_1(v, u')$ has form (1) and is described by means of adjusted *phi*-functions and adjusted quasi-*phi*-functions (4)–(11), $\Upsilon_2(v) = \min\{\Upsilon_{i2}(R^z_i = const, v_i), i \in I_n\}$, $U_t \in \{\emptyset, \{2\}, \{3\}, \{2, 3\}\}$.

It should be noted that if the value of the objective function is equal to $(z_s - z_e)^2$ then the optimal solution of BLP4 problem is found.

Feasible starting point algorithm (BSPA4) for BLP4 problem. BSPA4 algorithm involves the following steps.

Step 1. Generate a collection of random points $v^0_i = (x^0_i, y^0_i)$ belonging to the appropriate cross-section circles of radii R^z_i, $i \in I_n$. Form vector $v^0 = (x^0_1, y^0_1, \ldots, x^0_n, y^0_n)$. Fix rotation parameters $\theta_i = \theta^0_i = 0$, $i \in I_n$.

Step 2. Let $\lambda = \lambda_i$ be a homothetic coefficient for objects A_i, $i \in I_n$. Using clear geometric constructions we define the vector of additional variables u'^0 of τ-dimension, such that each our adjusted quasi-*phi*-function in (1) will reach its maximal value by additional variables u'^0 at point (u^0_λ, u'^0), where $u^0_\lambda = (v^0, \theta^0, \lambda^0)$, $\lambda^0 = 0$, $v^0 = (v^0_1, \ldots, v^0_n)$, $\theta^0 = (\theta^0_1, \ldots, \theta^0_n)$.

Step 3. Derive $\alpha^0 = \min\{\Upsilon_1(u_\lambda^0, u'^0), \Upsilon_2(u_\lambda^0)\}$. If $\alpha^0 < 0$, then go to Step 4, otherwise form point and go to Step 5.

Step 4. Set $\lambda = 0$, $\theta_i = \theta_i^0 = 0$, $i \in I_n$ and use as a starting point to solve the following auxiliary nonlinear problem:

$$\alpha^* = \max \alpha, \text{ s.t. } u_\alpha \in W_\alpha, \tag{30}$$

$$W_\alpha = \{u_\alpha \in \mathbb{R}^{3n+\tau+1} : \Upsilon_1(u_\lambda, u') - \alpha \geq 0, \Upsilon_2(u_\lambda) - \alpha \geq 0, -\alpha \geq 0\}, \tag{31}$$

where $= (u_\lambda, u', \alpha)$.

If $\alpha^* = 0$ then point of the global maximum of problem (30) and (31) is found and we go to Step 5. If $\alpha^* < 0$ then a feasible starting point for problem (30) and (31) could not be found, since placement constraints for BLP4 problem are "disrupted" under $\lambda = 0$. In the case we return to Step 1.

Step 5. Assume that parameters θ_i, $i \in I_n$, are variable. Generate randomly starting values of rotation parameters $\theta_i^* \in [0, 2\pi)$, $i \in I_n$, involved in vector.

Step 6. Take feasible starting point (u_λ^*, u'^*) using and solve the following auxiliary nonlinear problem:

$$\lambda^* = \max \lambda, \text{ s.t. } (u_\lambda, u') \in W_\lambda, \tag{32}$$

$$W_\lambda = \{(u_\lambda, u') \in \mathbb{R}^{3n+\tau+1} : \Upsilon_1(u_\lambda, u') \geq 0, \Upsilon_2(u_\lambda) \geq 0, 1 - \lambda \geq 0, \lambda \geq 0\}. \tag{33}$$

If $\lambda^* = 1$ then point $(u_\lambda^*, u'^*) = (v^*, \theta^*, \lambda^*, u'^*)$ of the global maximum of problem (32) and (33) is found and we go to Step 7. If $\lambda^* < 1$ then go to Step 1.

Step 7. Derive $\mu(v^*, \theta^*)$. If $\mu(v^*, \theta^*) < 0$, go to Step 8, otherwise go to Step 9.

Step 8. Starting from point , solve the following auxiliary nonlinear problem:

$$\beta^* = \max \beta, \text{ s.t. } u_\beta \in W_\beta, \tag{34}$$

$$W_\beta = \{u_\beta \in \mathbb{R}^{3n+1} : \Upsilon_1(u, u') \geq 0, \Upsilon_2(u) \geq 0, \mu(u) - \beta \geq 0, -\beta \geq 0\}, \tag{35}$$

where β is a *discrepancy*-variable, $u_\beta = (u, u', \beta)$, $u = (v, \theta)$.

If $\beta^* = 0$ then point $(v^*, \theta^*, u'^*, \beta^*)$ of the global maximum of problem (34) and (35) is found and we go to Step 9. If $\beta^* < 0$ then go to Step 1.

Step 9. Form starting feasible point $u^0 = (v^*, \theta^*, u'^*) \in W$ for BLP4 problem.

Instance 7. Let $\Omega \equiv \mathbf{E}$, $m = 2$, $H = 0.6$, $R_1 = 0.5$, $R_3 = 0.3$, $t_1 = 0.3$, $n = 10$, $A = \{\mathbb{S}_1, \mathbb{S}_2, \mathbb{C}_3, \mathbb{C}_4, \mathbb{T}_5, \mathbb{T}_6, \mathbb{S}_{\mathbb{C}7}, \mathbb{S}_{\mathbb{C}8}, \mathbb{K}_9, \mathbb{K}_{10}\}$, $A_-^1 = \{\mathbb{S}_1, \mathbb{C}_3, \mathbb{T}_5, \mathbb{S}_{\mathbb{C}7}, \mathbb{K}_9\}$, $A_+^2 = \{\mathbb{S}_2, \mathbb{C}_4, \mathbb{T}_6, \mathbb{S}_{\mathbb{C}8}, \mathbb{K}_{10}\}$, $\rho_{ij}^- = 0.03$, $i > j \in I_{10}$, $\rho_{39}^+ = 0.1$, $\rho_{26}^+ = 0.08$, $(x_e, y_e, z_e) = (0, 0, 0.275)$,

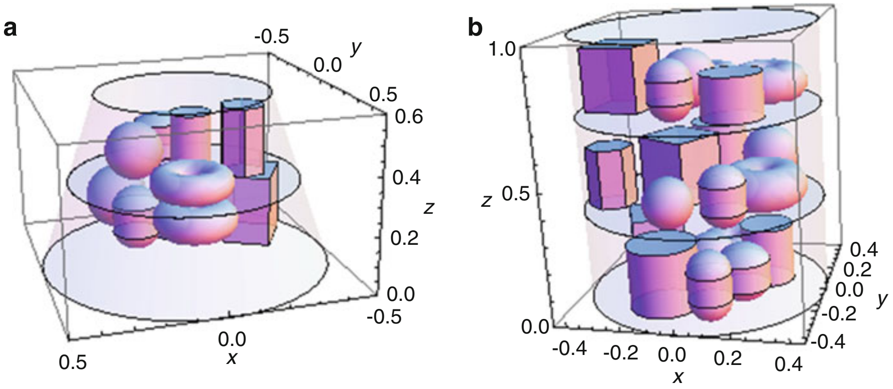


Fig. 11 The local-optimal placement of 3D-objects: (**a**) system Ω_A in Instance 6, (**b**) system Ω_A in Instance 7

$\{z_i, i = 1, \dots, 10\}$=$\{0.19, 0.4, 0.19, 0.41, 0.24, 0.35, 0.19, 0.39, 0.18, 0.42\}$, $\{m_i, i = 1, \dots, 10\}$=$\{27.8764, 20.944, 34.5575, 16.9332, 28.4245, 22.2066, 17.2159, 19.2265, 38.4, 19.9532\}$, $r_1 = 0.11$, $r_2 = 0.1$, $r_3 = 0.1$, $h_3 = 0.11$, $r_4 = 0.07$, $h_4 = 0.11$, $r_5 = 0.08$, $h_5 = 0.06$, $r_6 = 0.09$, $h_6 = 0.05$, $r_7 = 0.08$, $h_7 = 0.05$, $l_7 = 0.06$, $h_8 = 0.06$, $l_8 = 0.03$, $s_9 = 4, h_9 = 0.12$, $\tilde{v}_{91} = (0.08, 0.1)$, $\tilde{v}_{92} = (0.08, -0.1)$, $\tilde{v}_{93} = (-0.08, -0.1)$, $\tilde{v}_{94} = (-0.08, 0.1)$, $s_{10} = 6$, $h_{10} = 0.12$, $\tilde{v}_{(10)1} = (0.04, 0.07)$, $\tilde{v}_{(10)2} = (0.08, 0)$, $\tilde{v}_{(10)3} = (0.04, -0.07)$, $\tilde{v}_{(10)4} = (-0.04, -0.07)$, $\tilde{v}_{(10)5} = (-0.08, 0)$, $\tilde{v}_{(10)6} = (-0.04, 0.07)$.

The local-optimal solution under $U_t = \{2, 3\}$ found by NLP-solver in CAS Wolfram Mathematica 9.0 (Fig. 11a) is $F(u^*, u'^*) = 1.12726 \times 10^{-6}$.

Instance 8. Let $\Omega \equiv \mathbf{C}$, $m = 3$, $H = 1$, $R = 0.45$, $t_2 = 0.35$, $n = 20$, $A = \{\mathbb{S}_i, i = 1, \dots, 4, \mathbb{C}_i, i = 5, \dots, 8, \mathbb{T}_i, i = 9, \dots, 12, \mathbb{S}_{\mathbb{C}i}, i = 13, \dots 16, \mathbb{K}_i, i = 17, \dots, 20\}$, $A^1_+ = \{\mathbb{S}_1, \mathbb{C}_5, \mathbb{C}_6, \mathbb{T}_9, \mathbb{S}_{\mathbb{C}14}, \mathbb{P}_{17}\}$, $A^2_+ = \{\mathbb{S}_2, \mathbb{S}_3, \mathbb{C}_7, \mathbb{T}_{10}, \mathbb{S}_{\mathbb{C}15}\mathbb{P}_{18}, \mathbb{K}_{20}\}$, $A^3_+ = \{\mathbb{S}_4, \mathbb{C}_8, \mathbb{T}_{11}, \mathbb{T}_{12}, \mathbb{S}_{\mathbb{C}16}, \mathbb{P}_{19}\}$, $U_t = \emptyset$, $\rho^-_{ij} = 0.02$, $i < j = 1, \dots, 20$, $(x_e, y_e, z_e) = (0, 0, 0.5)$, $\{z_i, i = 1, \dots, 20\}$=$\{0.1, 0.44, 0.46, 0.81, 0.11, 0.12, 0.46, 0.78, 0.06, 0.425, 0.76, 0.77, 0.11, 0.13, 0.46, 0.81, 012, 0.47, 0.82, 0.46\}$, $\{m_i, i = 1, .., 20\}$=$\{20.944, 15.2681, 27.8764, 34.5575, 63.7115, 41.8146, 30.4106, 28.4245, 49.9649, 24.8714, 38.6888, 26.2637, 20.7764, 17.2159, 16.8756, 52.8, 52.8, 52.8, 23.1489\}$, $r_1 = 0.1$, $r_2 = 0.09$, $r_3 = 0.11$, $r_4 = 0.11$, $r_5 = 0.1$, $h_5 = 0.11$, $h_6 = 0.12$, $r_7 = 0.11$, $r_8 = 0.11$, $h_8 = 0.08$, $r_9 = 0.08$, $h_9 = 0.07$, $r_{10} = 0.09$, $h_{10} = 0.075$, $r_{11} = 0.07$, $h_{11} = 0.06$, $r_{12} = 0.08$, $h_{12} = 0.07$, $r_{13} = 0.1$, $h_{13} = 0.05$, $l_{13} = 0.07$, $r_{14} = 0.05$, $h_{14} = 0.05$, $l_{14} = 0.08$, $r_{15} = 0.08$, $h_{15} = 0.05$, $l_{15} = 0.06$, $r_{16} = 0.08$, $h_{16} = 0.04$, $l_{16} = 0.07$, $s_i = 4$, $\tilde{v}_{i1} = (-0.11, -0.1)$, $\tilde{v}_{i2} = (0.11, -0.1)$, $\tilde{v}_{i3} = (0.11, 0.1)$, $\tilde{v}_{i4} = (-0.11, 0.1)$, $h_i = 0.12$,$i = 17, 18, 19$, $s_{20} = 6$, $\tilde{v}_{(20)1} = (0.045, 0.078)$, $\tilde{v}_{(20)2} = (0.09, 0)$, $\tilde{v}_{(20)3} = (0.045, -0.078)$,$\tilde{v}_{(20)4} = (-0.045, -0.078)$, $\tilde{v}_{(20)5} = (-0.09, 0)$, $\tilde{v}_{(20)6} = (-0.045, 0.078)$, $h_{20} = 0.11$.

The local optimal solution under $U_t = \emptyset$ found by NLP-solver in CAS Wolfram Mathematica 9.0 is $F(u^*, u'^*) = 0.001911$ (Fig. 11b).

Instance 9. Let $\Omega \equiv \mathbf{C}$, $m = 2$, $H = 20$, $R = 8.8$, $t_1 = 10$, $n = 80$, $A = \{\mathbb{C}_i, i = 1, \ldots 64, \mathbb{K}_i, i = 65, \ldots, 80\}$, $A^1_+ = \{\mathbb{C}_i, i = 1, \ldots, 16, \mathbb{K}_i, i = 65, \ldots, 68\}$, $A^1_- = \{\mathbb{C}_i, i = 17, \ldots, 32, \mathbb{K}_i, i = 69, \ldots, 72\}$, $A^2_- = \{\mathbb{C}_i, i = 49, \ldots, 64, \mathbb{K}_i, i = 77, \ldots, 80\}$, $A^2_+ = \{\mathbb{C}_i, i = 33, \ldots, 48, \mathbb{K}_i, i = 73, \ldots, 76\}$, $U_t = \emptyset, \{r_i, i = 1, \ldots, 64\}$ ={2.0, 2.4, 0.8, 1.1, 1.3, 0.7, 0.7, 1.5, 2.4, 1.8, 1.5, 1.7, 1.7, 1.4, 1.6, 2.1, 2.0, 2.4, 0.8, 1.1, 1.3, 0.7, 0.7, 1.5, 2.4, 1.8, 1.5, 1.7, 1.7, 1.4, 1.6, 2.1, 2.0, 2.4, 0.8, 1.1, 1.3, 0.7, 0.7, 1.5, 2.4, 1.8, 1.5, 1.7, 1.7, 1.4, 1.6, 2.1, 2.0, 2.4, 0.8, 1.1, 1.3, 0.7, 0.7, 1.5, 2.4, 1.8, 1.5, 1.7, 1.7, 1.4, 1.6, 2.1}, $s_i = 4, i = 65, \ldots, 80$, $\tilde{v}_{i1} = (-1.8, -1.8)$, $\tilde{v}_{i2} = (1.8, -1.8)$, $\tilde{v}_{i3} = (1.8, 1.8)$, $\tilde{v}_{i4} = (-1.8, 1.8)$, $i = 65, 69, 73, 77$, $\tilde{v}_{i1} = (-0.5, -0.5)$, $\tilde{v}_{i2} = (0.5, -0.5)$, $\tilde{v}_{i3} = (0.5, 0.5)$, $\tilde{v}_{i4} = (-0.5, 0.5)$, $i = 66, 70, 74, 78$, $\tilde{v}_{i1} = (-2.1, -2.1)$, $\tilde{v}_{i2} = (2.1, -2.1)$, $\tilde{v}_{i3} = (2.1, 2.1)$, $\tilde{v}_{i4} = (-2.1, 2.1)$, $i = 67, 71, 75, 79$, $\tilde{v}_{i1} = (-1.3, -1.3)$, $\tilde{v}_{i2} = (1.3, -1.3)$, $\tilde{v}_{i3} = (1.3, 1.3)$, $\tilde{v}_{i4} = (-1.3, 1.3)$, $i = 68, 72, 76, 80$, $\{h_i, i = 1, \ldots, 62\}$={1.5, 1.5, 1.5, 1.5, 1.5, 3.0, 1.5, 1.5, 1.5, 3.0, 1.5, 3.0, 1.5, 3.0, 3.0, 1.5, 1.5, 1.5, 3.0, 1.5, 3.0, 1.5, 1.5, 3.0, 1.5, 1.5, 3.0, 1.5, 1.5, 3.0, 1.5, 1.5, 1.5, 3.0, 1.5, 1.5, 1.5, 3.0, 1.5, 1.5, 1.5, 3.0, 1.5, 1.5, 1.5, 3.0, 3.0, 1.5, 1.5, 1.5, 3.0, 1.5, 3.0, 1.5, 1.5, 3.0, 1.5, 1.5, 3.0, 1.5, 1.5, 3.0, $\{h_i, i = 63, \ldots, 80\}$={1.5}, $\{m_i, i = 1, \ldots, 80\}$={86, 72, 81, 54, 29, 94, 92, 41, 57, 77, 40, 67, 31, 47, 39, 61, 73, 83, 11, 20, 86, 72, 81, 54, 29, 94, 92, 41, 57, 77, 40, 67, 31, 47, 39, 61, 73, 83, 11, 20, 86, 72, 81, 54, 29, 94, 92, 41, 57, 77, 40, 67, 31, 47, 39, 61, 73, 83, 11, 20, 86, 72, 81, 54, 29, 94, 92, 41, 57, 77, 40, 67, 31, 47, 39, 61, 73, 83, 11, 20}.

The local optimal solution found by IPOPT is $F(u^*, u'^*) = 0.000000$ (Fig. 12).

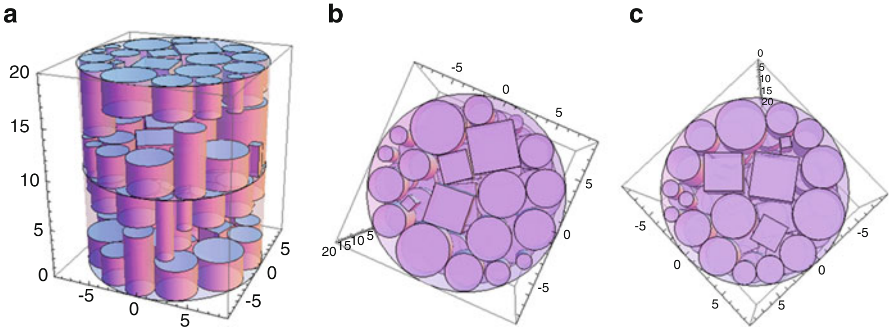


Fig. 12 The local-optimal placement of 3D-objects in Instance 8: (**a**) system Ω_A, (**b**) *top view* of Ω_A, (**c**) *bottom view* of Ω_A

5 Conclusions

In this chapter we formulate the optimization layout problem of 3D objects (solid spheres, right circular cylinders, tori, spherocylinders, cuboids and right-angle prisms) into a container (cylinder, blunted cone, paraboloid of revolution) with circular racks taking into account both distance (minimal and maximal allowable distances) constraints and behaviour (equilibrium, inertia and stability) constraints. We call the problem as Balance Layout Problem (BLP). In order to describe placement constraints analytically we derive a collection of radical-free adjusted *phi*-functions and adjusted quasi-*phi*-functions. These functions allow us to build an exact mathematical model of the BLP problem in the form of nonlinear programming problem with differentiable functions. We develop a solution strategy, which is based on the multistart method and involves nontrivial procedures to construct feasible starting points, nonlinear programming and nonsmooth optimization methods, employing NLP solvers. We also consider some variants of the BLP problem depending on the form of the objective function, shapes of objects and containers, types of distance and behaviour constraints. To show the efficiency of our approach we provide the collection of Instances.

Appendix 1: Phi-Functions and Quasi-Phi-Functions

Phi-Objects

Here we define a class of admissible objects for our models, called *phi*-objects (see, e.g., [11]). They must have interior ("main part") and boundary (frontier). Accordingly, we require each *phi*-object be the closure of its interior. (In mathematical topology, closed sets that are closures of their interior are said to be canonically closed; this is what our *phi*-objects are.) This requirement rules out such elements as isolated points, one-dimensional curves, etc.— they do not occur in realistic applications. Figure A1a shows an invalid *phi*-object— it has three one-dimensional

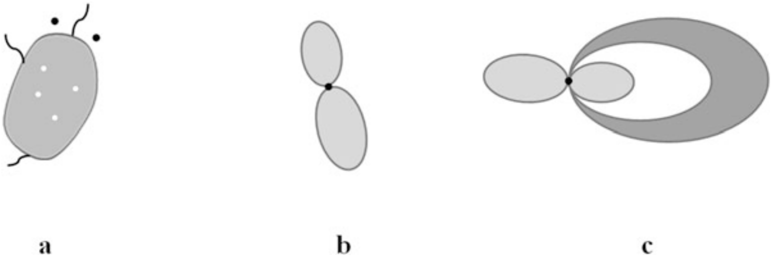


Fig. A1 Examples of invalid *phi*-objects: (**a**) object with 'whiskers', isolated and four punctured points, (**b**) object with self-intersections along its frontier, (**c**) the confusion case for two objects

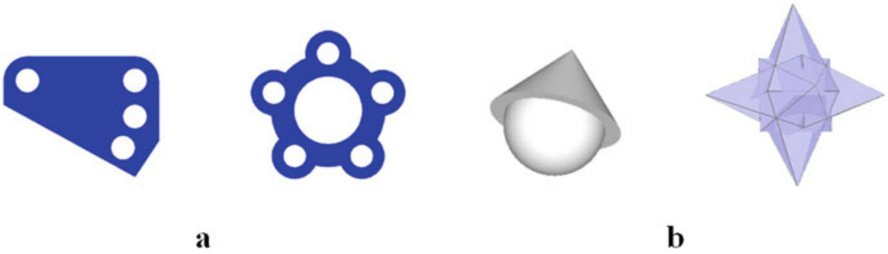


Fig. A2 Examples of valid *phi*-objects: (**a**) 2D *phi*-objects, (**b**) 3D *phi*-objects

'whiskers', two isolated points, and four punctured interior points (white dots). In addition, our *phi*-objects should not have self-intersections along their frontier, as shown in Fig. A1b because this may lead to confusion. For example, Fig. A1c shows a dark domain of which two ends touch each other like pincers; this must be prohibited. The reason is also demonstrated in the same figure: a similar object (the light grey "figure eight") is placed so that the two objects intersect each other only in their frontiers, which is generally allowed, but in this particular case we cannot place these objects as shown because one 'cuts' through the other.

Mathematically, the above requirement can be stated as the following: a *phi*-object and its interior must have the same homotopic type (the same number of connected components, the same number of interior holes, etc.). These requirements may sound too abstract, but their practical meaning should be clear from the above example. An important property of *phi*-objects is that if A is a *phi*-object, then the closure of its complement is a *phi*-object, too. Figure A2 shows the examples of valid *phi*-objects.

Phi-Functions

Let A and B be two *phi*-objects. The position of object A is defined by the vector of *placement parameters* (v_A, θ_A), where: $v_A = (x_A, y_A)$ is a translation vector and θ_A is a rotation angle if $A \subset \boldsymbol{R}^2$; $v_A = (x_A, y_A, z_A)$ is a translation vector and $\theta_A = (\theta_z, \theta_x, \theta_y)$ are rotation angles (from axis OX to OY, from axis OY to OZ and from axis OX to OZ) if $A \subset \boldsymbol{R}^3$. We denote the vector of variables for object A by $u_A = (v_A, \theta_A)$ and the vector of variables for object B by $u_B = (v_B, \theta_B)$. Object A rotated by θ_A and translated by vector v_A will be denoted by $A(u_A)$.

In order to feasibly place two *phi*-objects within a containing region, we need an analytical description of the relationships between a pair of objects A and B. We employ the *phi*-function technique for this. *Phi*-functions allow us to distinguish the following three cases: A and B are intersecting so that A and B have common interior points; A and B do not intersect, i. e. A and B do not have common points; A and B are in contact, i. e. A and B have only common frontier points.

Definition A1. Continuous and everywhere defined function $\Phi^{AB}(u_A, u_B)$ is called a *phi*-function for objects $A(u_A)$ and $B(u_B)$ if

$$\Phi^{AB} < 0, \text{ if } \mathrm{int}A(u_A) \cap \mathrm{int}B(u_B) \neq \emptyset;$$

$$\Phi^{AB} = 0, \text{ if } \mathrm{int}A(u_A) \cap \mathrm{int}B(u_B) = \emptyset \text{ and } frA(u_A) \cap frB(u_B) \neq \emptyset;$$

$$\Phi^{AB} > 0, \text{ if } A(u_A) \cap B(u_B) = \emptyset.$$

Here frA means the boundary (frontier) and int$A$ means the interior of object A.

Thus, $\Phi^{AB} \geq 0 \Leftrightarrow intA(u_A) \cap intB(u_B) = \emptyset$. We employ *phi*-functions for the description of the containment relation $A \subseteq B$ as the following: $\Phi_{AB^*} \geq 0$, where $B^* = \boldsymbol{R}^d \backslash intB$, $d = 2, 3$. We emphasize that according to Definition A1, *phi*-function Φ^{AB} for a pair of objects A and B can be constructed by many different formulas, and we can choose the most convenient ones for our optimization algorithms.

Quasi-Phi-Functions

In comparison with *phi*-functions we include auxiliary variables u', which take values in some domain $U \subset \boldsymbol{R}^n$ (it depends on the shapes of objects A and B), and introduce function $\Phi'^{AB}(u_A, u_B, u')$. The function must be defined for all values of u_A and u_B. It must be continuous in all its variables.

Definition A2. Continuous and everywhere defined 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 $\max\limits_{u' \in U} \Phi'^{AB}(u_A, u_B, u')$ is a *phi*-function for the objects.

Let us consider two convex objects $A(u_A)$ and $B(u_B)$ and let $P(u_P)$ be a half-space: $P(u_P) = \{(\mathrm{x}, \mathrm{y}, \mathrm{z}) : \psi_P = \alpha \cdot x + \beta \cdot y + \gamma \cdot z + \mu_P \leq 0\}$, $u_P = (\theta_{xP}, \theta_{yP}, \mu_P)$, $\alpha = \sin \theta_{yP}$, $\beta = -\sin \theta_{xP} \cdot \cos \theta_{yP}$, $\gamma = \cos \theta_{xP} \cdot \cos \theta_{yP}$ for 3D case;

$P(u_P) = \{(x, y) : \psi_P = \alpha \cdot x + \beta \cdot y + \gamma_P \leq 0\}$, $u_P = (\theta_P, \gamma_P)$, $\alpha = \cos \theta_P$, $\beta = \sin \theta_P$ for 2D case. A function defined by

$$\Phi'^{AB}(u_A, u_B, u' = u_P) = \min\{\Phi^{AP}(u_A, u_P), \Phi^{BP^*}(u_B, u_P)\},$$

is a quasi-*phi*-function for $A(u_A)$ and $B(u_B)$. Here $\Phi^{AP}(u_A, u_P)$ is a *phi*-function for $A(u_A)$ and a half-space $P(u_P)$ and $\Phi^{BP^*}(u_B, u_P)$ is a *phi*-function for $B(u_B)$ and $P^*(u_P) = \boldsymbol{R}^d \backslash intP(u_P)$, $d = 2, 3$.

The latter function meets all the requirements of Definition A2. First, function Φ'^{AB} is defined everywhere and is continuous in all its variables, since the *phi*-functions Φ^{AP} and Φ^{BP^*} enjoy the same properties. Based on the properties of a separated line (plane) for two convex objects the following is fulfilled:

1) $\max\limits_{u' \in R^d} \Phi'^{AB} < 0$, if $intA(u_A) \cap intB(u_B) \neq \emptyset$;
2) $\max\limits_{u' \in R^d} \Phi'^{AB} = 0$, if $intA(u_A) \cap intB(u_B) = \emptyset$ and $frA(u_A) \cap frB(u_B) \neq \emptyset$;
3) $\max\limits_{u' \in R^d} \Phi'^{AB} > 0$, if $A(u_A) \cap B(u_B) = \emptyset$.

It means that $\max\limits_{u' \in R^d} \Phi'^{AB}$ is a *phi*-function for objects A and B according to Definition A1.

Examples

Example 1. Let $v_i^1 = (x_i^1, y_i^1)$, $i = 1, \dots, m_1$, be the vertices of convex polygon $K_1(u_1)$, and $v_j^2 = (x_j^2, y_j^2)$, $j = 1, \dots, m_2$, those of convex polygon $K_2(u_2)$, and $K_1(u_1) = \{(x,y) : \phi_i \leq 0,\ i = 1, \dots, m_1\}$, $K_2(u_2) = \{(x,y) : \psi_j \leq 0,\ j = 1, \dots, m_2\}$, $\phi_i = \alpha_i' x + \beta_i' y + \gamma_i'$, $\psi_j = \alpha_j'' x + \beta_j'' y + \gamma_j''$, where $u_1 = (x_1, y_1, \theta_1)$ and $u_2 = (x_2, y_2, \theta_2)$ are the placement parameters of polygons K_1 and K_2.

It should be noted that each point $(\widetilde{x}, \widetilde{y})$ of non-translated and non-rotated convex polygon K is transformed into point (x, y):
$x = \widetilde{x} \cdot \cos\theta_K + \widetilde{y} \cdot \sin\theta_K + x_K, y = -\widetilde{x} \cdot \sin\theta_K + \widetilde{y} \cdot \cos\theta_K + y_K$,where $(x_K$,$y_K)$ is a translation vector and θ_K is a rotation angle of K.

A *phi*-function for K_1 and K_2 can be defined in the form

$$\Phi^{K_1K_2} = \max\{ \max_{1 \leq i \leq m_1} \min_{1 \leq j \leq m_2} \phi_{ij}, \max_{1 \leq j \leq m_2} \min_{1 \leq i \leq m_1} \psi_{ji} \}, \tag{36}$$

where $\phi_{ij} = \phi_i(v_j^2) = \alpha_i' x_j^2 + \beta_i' y_j^2 + \gamma_i'$, $\psi_{ji} = \psi_j(v_i^1) = \alpha_j'' x_i^1 + \beta_j'' y_i^1 + \gamma_j''$.

Example 2. Let us consider *convex polygons* K_1 and K_2 from Example 1.

A quasi-*phi*-function for K_1 and K_2 can be defined in the form

$$\Phi'^{K_1K_2}(u_1, u_2, u_P) = \min\{\Phi^{K_1P}(u_1, u_P), \Phi^{K_2P^*}(u_2, u_P)\}, \tag{37}$$

where $\Phi^{K_1P}(u_1, u_P) = \min\limits_{1 \leq i \leq m_1} \psi_P(v_i^1)$ is a *phi*-function of K_1 and halfplane $P(u_P)$, $\Phi^{K_2P^*}(u_2, u_P) = \min\limits_{1 \leq j \leq m_2} (-\psi_P(v_j^2))$ is a *phi*-function of K_2 and halfplane $P^*(u_P) = R^2 \backslash intP(u_P)$.

In general, each of our *phi*-functions (ordinary, adjusted) is formed by operations of minimum and maximum of continuous and everywhere defined functions. The more operations of maximum take part in forming of a *phi*-function the more nonlinear programming subproblems we need to solve.

For example, in order to reach the global minimum for the problem of packing of two convex polygons K_1 and K_2 in a rectangle of minimum area, using *phi*-function (36), we need to solve $m_1 + m_2$ nonlinear programming subproblems optimally. See details in [18].

Alternatively, in order to reach the global minimum of the latter problem, using quasi-*phi*-function (37), we need to solve only one nonlinear programming problem optimally. However, in the case the problem dimension is increased by two.

We may reasonably combine *phi*-functions and quasi-*phi*-functions in our models depending on types of our objects.

Appendix 2: Moments of Inertia for Containers and Objects

- For the lateral surface of container Ω we have:

$$\Omega \equiv \mathbf{C}: \ J_{x_0} = J_{y_0} = \frac{1}{6} m_0 (3R^2 + 2H^2), \ J_{z_0} = m_0 R^2;$$

$$\Omega \equiv \mathbf{\Lambda}: \ J_{x_0} = J_{y_0} = \frac{1}{70} m_0 H (21 + 16H), \ J_{z_0} = \frac{3}{5} m_0 H;$$

$$\Omega \equiv \mathbf{E}: \ J_{x_0} = J_{z_0} = \frac{m_0}{2} \left(\frac{H^2(R_1 + 3R_2)}{3(R_1 + R_2)} + \frac{R_1^2 + R_2^2}{2} \right), \ J_{z_0} = \frac{m_0(R_1^2 + R_2^2)}{2}.$$

- For a homogeneous object A_i we have:

$$A_i \equiv \mathbb{S}_i: \ J_{x_i} = J_{y_i} = J_{z_i} = \frac{2}{5} m_i r_i^2;$$

$$A_i \equiv \mathbb{C}_i: \ J_{x_i} = J_{y_i} = \frac{1}{12} m_i (3r_i^2 + 4h_i^2), \ J_{z_i} = \frac{1}{2} m_i r_i^2;$$

$$A_i \equiv \mathbb{T}_i: \ J_{x_i} = J_{y_i} = \frac{1}{8} m_i (4r_i^2 + 5h_i^2), \ J_{z_i} = \frac{1}{4} m_i (4r_i^2 + 3h_i^2);$$

$$A_i \equiv \mathbb{S}_{\mathbb{C}i}: \ J_{x_i} = J_{y_i} = \frac{m_i}{2} \left(\frac{h_i(2h_i(l_i^3 + 3l_i r_i^2) + (2l_i^4 + 4l_i^2 r_i^2 + 3r_i^4))}{l_i^3 + 6h_i r_i^2 + 3l_i r_i^2} + \right.$$

$$\left. + \frac{7l_i^5 + 5r_i^2(8h_i^3 + 3l_i^3 + 2l_i r_i^2)}{10(l_i^3 + 6h_i r_i^2 + 3l_i r_i^2)} \right), \ J_{z_i} = \frac{m_i(l_i^5 + 5l_i^3 r_i^2 + 30h_i r_i^4 + 10l_i r_i^4)}{10(l_i^3 + 6h_i r_i^2 + 3l_i r_i^2)}.$$

For objects $A_i \equiv \mathbb{K}_i$ moments of inertia depends on a type of the cross-section polygon. For instance, we have

$J_{x_i} = \frac{1}{12} m_i (l_i^2 + h_i^2)$, $J_{y_i} = \frac{1}{12} m_i (w_i^2 + h_i^2)$, $J_{z_i} = \frac{1}{12} m_i (l_i^2 + w_i^2)$ for cuboid, and $J_{x_i} = J_{y_i} = \frac{1}{24} m_i (5r_i^2 + 8h_i^2)$, $J_{z_i} = \frac{5}{12} m_i r_i^2$ for straight regular prism with the regular hexagon base.

Appendix 3: Shor's r-Algorithm

The r-algorithm is one of the Shor's subgradient-type methods with the space transformation of variables (the space dilation) for minimisation of nonsmooth convex functions. Shor's r-algorithms are based on two related ideas. The first idea lies in using the steepest descent method in the direction of antisubgradient of nonsmooth convex functions in the transformed space of variables. It ensures a monotonicity of a nonsmooth convex function for the minimizing the sequence which is constructed by r-algorithm. The second idea employs the operation of the space dilation in the direction of the difference of two subsequent subgradients in order to transform the space of variables; this permits to improve the properties of ravine-like functions in the transformed space. Combination of the ideas provides the accelerated convergence of r-algorithms for ravine-like functions ensuring their monotonicity (or almost monotonicity) under certain regulation of the step and the space dilation coefficients.

Let $f(x)$ be a convex function, x be a vector of n variables. We assume that space dilation coefficients $\{\alpha_k\}_{k=0}^{\infty}$ have to be greater than unity. Our r-algorithm for minimization of $f(x)$ is an iterative procedure for finding sequence of vectors $\{x_k\}_{k=0}^{\infty}$ and matrices $\{B_k\}_{k=0}^{\infty}$ by the following rule:

$$x_{k+1} = x_k - h_k B_k \xi_k, \quad B_{k+1} = B_k R_{\beta_k}(\eta_k), \quad k = 0, 1, 2, \ldots, \tag{38}$$

where

$$\xi_k = \frac{B_k^T g_f(x_k)}{\parallel B_k^T g_f(x_k) \parallel}, \quad h_k = \arg\min_{h \geq 0} f(x_k - hB_k\xi_k), \tag{39}$$

$$\eta_k = \frac{B_k^T r_k}{\parallel B_k^T r_k \parallel}, \quad r_k = g_f(x_{k+1}) - g_f(x_k), \quad \beta_k = \frac{1}{\alpha_k} < 1. \tag{40}$$

Here, x_0 is a starting point; $B_0 = I_n$ is a unity $n \times n$-matrix (B_0 is often taken to be diagonal matrix D_n with positive entries on a diagonal to make scaling of variables); h_k is a step multiplier (found from the condition of minimum of function $f(x)$ in the direction of the normed subgradient in the transformed space of variables); α is a coefficient of the space dilation; $R_\beta(\eta) = I_n + (\beta - 1)\eta\eta^T$ is an operator of contraction of space of subgradients in the normed direction η with coefficient $\beta = \frac{1}{\alpha} < 1$; $g_f(x_k)$ and $g_f(x_{k+1})$ are subgradients of function $f(x)$ at points x_k and x_{k+1}. If $g_f(x_k) = 0$, then x_k is a point of the minimum of function $f(x)$, and process (38) and (40) stops.

Among r-algorithms the most efficient is $r(\alpha)$-algorithm with $\alpha_k \equiv \alpha$ and adaptive regulation of step h_k. The value of h_k is related to the unidimensional descent procedure in the direction of the normed antigradient in the transformed space of variables. The procedure involves parameters h_0, q_1, n_h, q_2. Here h_0 is the value of an initial step (it is used on the first iteration, and this value is sequentially refined on each iteration); q_1 is a step decrease factor ($q_1 \leq 1$), if the descent stopping criterion is satisfied in one step; q_2 is a step increase factor ($q_2 \geq 1$); natural number n_h specifies the number of steps in one-dimensional descent ($n_h > 1$)—after this number of steps the step size will be taken q_2 times greater.

Guidance to the values of the space dilation coefficient as well as the parameters of adaptive regulation of a step is discussed in [15, pp. 104–105]. The values are aimed to improve the accuracy of finding of approximation to the minimum of the function, provided that the number of steps should not be too large (two-three per one iteration).

Stopping criteria in $r(\alpha)$-algorithm is described by parameters ε_x and ε_g: calculations come to the end at point x_{k+1}, if $\|x_{k+1} - x_k\| \leq \varepsilon_x$ (stopping criterion by argument) or if $\left\|g_f(x_{k+1})\right\| \leq \varepsilon_g$ (stopping criterion by normed gradient, which is used for smooth functions). Abnormal program termination can happen if either function $f(x)$ is not bounded below, or initial step h_0 is too small and should be increased.

The following values of parameters are recommended for minimization of nonsmooth functions: $\alpha = 2 \div 3$, $h_0 = 1.0$, $q_1 = 1.0$, $q_2 = 1.1 \div 1.2$, $n_h = 2 \div 3$. If the priory bound of the distance from starting point x_0 to the minimum point x^* is given, then it is reasonable to choose initial step h_0 to be approximately equal to $\| x_0 - x^* \|$.

For minimization of smooth functions the same parameters are recommended, except q_1, that should be taken $q_1 = 0.8 \div 0.95$. This can be explained in such a way: further step decreasing would provide finding a more accurate approximation to the minimum point of the function in the direction, and in the case of minimization of smooth functions this gives good rate of convergence. Under the parameters the number of descents is usually not greater than two, and after n steps the accuracy will be three-five times better. Stopping parameters $\varepsilon_x, \varepsilon_g \sim 10^{-6} \div 10^{-5}$ for minimization of a convex function (even the strongly ravine-like one) provide finding x_r^* which is a fairly good approximation to the minimum point of the function.

Usually the condition $\frac{f(x_r^*)-f(x^*)}{|f(x^*)|+1} \sim 10^{-6} \div 10^{-5}$ for nonsmooth functions (and $\sim 10^{-12} \div 10^{-10}$ for smooth functions) is satisfied. It is confirmed by the results of numerous tests and applied calculations in linear and nonlinear programming problems, block problems with different schemes of decompositions, minimax and matrix optimization problems. It is also used for calculation of Lagrangian dual bounds in multiextremal and combinatorial optimization problems.

Shor's $r(\alpha)$-algorithm with the adaptive step regulation is realized by a number of programs. One of the simplest programs is octave-program ralgb5, which requires $5n^2$ arithmetical operations for each iteration [16]. The program uses octave's function [f, g] = calcfg(x), which calculates values of function $f = f(x)$ and its subgradient $g = \partial f(x)$ at point x.

References

1. Fasano, G., Pintér, J. (eds.): Modeling and Optimization in Space Engineering. Series: Springer Optimization and Its Applications. vol. 73, XII, 404 pp. Springer, New York (2013)
2. Fasano, G., Pintér, J. (eds.): Optimized Packings and Their Applications. Springer Optimization and Its Applications, vol. 105, 326 pp. Springer, Berlin (2015)
3. Che, C., Wang Y., Teng, H.: Test problems for quasi-satellite packing: Cylinders packing with behaviour constraints and all the optimal solutions known. Opt. (2008). Online http://www.optimization-online.org/DB_HTML/2008/09/2093.html
4. 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, vol. 1, pp. 434–442. Springer, Berlin (2009)
5. Sun, Z., Teng, H.: Optimal layout design of a satellite module. Eng. Optim. **35**(5), 513–530 (2003)
6. Jingfa, L., Gang, L.: Basin filling algorithm for the circular packing problem with equilibrium behavioural constraints. Science China Inf. Sci. **53**(5), 885–895 (2010)
7. Oliveira, W.A., Moretti, A.C., Salles-Neto, L.L.: A heuristic for the nonidentical circle packing problem. Anais do CNMAC, **3**, 626–632 (2010)
8. Xu, Y.-C., Xiao R.-B., Amos, M.: A novel algorithm for the layout optimization problem. In: Proceedings of 2007 IEEE Congress on Evolutionary Computation (CEC07), pp. 3938–3942. IEEE Press, New York (2007)
9. Chazelle, B., Edelsbrunner, H., Guibas, L.J.: The complexity of cutting complexes. Discret. Comput. Geom. **4**(2), 139–181 (1989)
10. 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). doi:10.1007/s10559-015-9746-5
11. Chernov, N., Stoyan, Y., Romanova T.: Mathematical model and efficient algorithms for object packing problem. Comput. Geom. Theory Appl. **43**(5), 533–553 (2010)
12. Chernov, N., Stoyan, Y., Romanova, T., Pankratov, A.: *phi*-functions for 2D objects formed by line segments and circular arcs. Adv. Oper. Res. (2012). doi:10.1155/2012/346358
13. Stoyan, Y., Pankratov, A., Romanova, T.: Quasi-*phi*-functions and optimal packing of ellipses. J. Global Optim. (2015). doi:10.1007/s10898-015-0331-2
14. 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)
15. Shor, N.Z.: Nondifferentiable Optimization and Polynomial Problems, vol. 394. Kluwer Academic Publishers, Boston (1998)
16. Shor, N.Z., Stetsyuk, P.I.: Modified *r*-algorithm to find the global minimum of polynomial functions. Cybern. Syst. Anal. **33**(4), 482–497 (1997)
17. Stetsyuk, P., Romanova, T., Scheithauer, G.: On the global minimum in a balanced circular packing problem. Optim. Lett. (2015). doi:10.1007/s11590-015-0937-9
18. Bennell, J., Scheithauer, G., Stoyan, Y., Romanova, T., Pankratov, A.: Optimal clustering of a pair of irregular objects. J. Glob. Optim. **61**(3), 497–524 (2015)

Pilot-Induced-Oscillations Alleviation Through Anti-windup Based Approach

Sophie Tarbouriech, Isabelle Queinnec, Jean-Marc Biannic, and Christophe Prieur

Abstract The chapter is dedicated to the optimization of a well-known structure of compensators: the anti-windup scheme. This approach belongs to the saturation allowance control class which aims to exploit at the most the actuators capabilities. The objective of this chapter consists of adapting and developing the anti-windup compensator design to some particular classes of nonlinear actuators presenting both magnitude and rate saturations. It is illustrated on the lateral flying case for a civil aircraft in presence of aggressive maneuvering of the pilot. A complete methodology is then proposed comparing several approaches including given anti-PIO filters.

Keywords Magnitude and rate saturations • Anti-windup compensator • Convex optimization • PIO

1 Introduction

Control engineers, where possible, like to work under the assumption of linearity. The mathematics associated with the field of linear systems is well developed and underpins much of the control theory which is applied in industry. Even nonlinear techniques often attempt to generalize linear concepts, and frequently nonlinear systems are linearized to obtain linear models which locally yield good engineering approximations [18]. The problem with the assumption of linearity is that it is sometimes unrealistic and can lead to erroneous results. Actually, the increasing requirements in terms of operational reliability and performance ask to work beyond

S. Tarbouriech (✉) • I. Queinnec
LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
e-mail: tarbour@laas.fr; queinnec@laas.fr

J.-M. Biannic
ONERA, System Control and Flight Dynamics Department, Toulouse, France
e-mail: Jean-Marc.Biannic@onera.fr

C. Prieur
GIPSA-Lab, Department of Automatic Control, Grenoble Campus,
Saint Martin d'Hères, France
e-mail: christophe.prieur@gipsa-lab.fr

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization
and Its Applications 114, DOI 10.1007/978-3-319-41508-6_15

the linear behavior of the system. Hence, actuators saturations (both magnitude and rate saturations) represent a common nonlinear phenomenon in almost all physical applications, especially in space and aeronautical fields. There are many examples of saturation problems but perhaps the most notorious are those associated with so-called pilot-induced-oscillations (PIO's) in aeronautics (see, for example, [8, 19, 20]). These saturation-induced events have led to the crash of several aircrafts (the SAAB Grippen and the Boeing V22 Osprey are notorious examples [2]) and several near-misses with others. Actually, recall that a pilot-induced oscillation is a sustained or uncontrollable, undesired oscillation resulting from the action of the pilot to control the aircraft. A common nonlinearity leading to PIO is control surface rate limiting. Then this phenomenon can introduce a delayed response and then the action of the pilot implies that the airplane response is essentially opposite of the command wished by the pilot (see, in particular the recent chapter [7]) Thus the presence of saturation can lead to performance degradation from the mild to the severe and can also lead to loss of stability [16, 17]. Although this is not always critical, it is clear that some way of predicting the effects of saturation is required and, moreover, that some method of limiting the degradation that occurs is warranted. This reflects the need for the development of new and more complex control techniques in order to meet the new demands.

In the aeronautical literature, there exist some methods mainly based on the addition of filters or estimators designed to predict and reduce the risk of PIO (see, for example the OLOP criterion using describing functions studied in [12] or the use of a detector based on short time Fourier transform and autoregressive model [21]. In this chapter, we choose another route by considering the use of anti-windup technique, with the objective to provide constructive conditions (that is associating theoretical conditions to optimization routines in order to exhibit effective numerical solutions). More specifically, the approach proposed in this chapter is based on the optimization of a well-known structure of compensators: the anti-windup scheme (see e.g., [29, 32] for an introduction of this notion). This approach belongs to the saturation allowance control class which aims to exploit at the most the actuators capabilities. The basic concept consists of introducing an extra layer to the existing linear controller, accounting for the nonlinearities in order to mitigate the windup phenomenon created by the saturation [15]. This strategy, also called anti-windup design, allows the designer to keep the existing linear controller (already validated) and to introduce a compensator which is active only when the nonlinearity arises. In this framework, numerous works have emerged in the context of both magnitude and rate saturation constraints [6, 10, 11, 13, 25, 26]. Such an approach appears to be really attractive as the anti-windup loop may work with existing control laws (a priori designed by the engineers to answer to defined requirements). Indeed, it represents an interesting technique for the controller designers who can use familiar and intuitive techniques for them and then, simply add an extra layer, which will consider the nonlinear behavior in a second step. If originally, results on anti-windup design consisted on ad-hoc methods intended to work with PID controllers [3, 9], modern anti-windup methods have emerged during the last decade (see, for example, [14, 31]). Then, the design of such an additional compensator is generally carried out through a static optimization problem of the controller parameters. Thanks to the development of semi-definite programming and convex optimization [5], the anti-windup controller design problem can be formulated as the optimization of a multi-objective criterium (corresponding to closed-loop stability and performance specifications) subject to matrix inequalities constraints associated to the dynamical system. Many different techniques exist in control theory to synthesize such anti-windup controllers, among which static and dynamic linear anti-windup augmentation (see [32]) based on a generalized sector condition representing the saturation [29]. Other anti-windup augmentations are possible as nonlinear synthesis, in particular for control systems equipped with other nonlinearities than magnitude and rate saturations. One can consult [1] for control systems presenting two different sector conditions, and [30] for a control system with a memory-based input.

The objective of this chapter is to adapt and develop the anti-windup compensator design to a class of systems presenting both magnitude and rate saturations. The techniques proposed first include a modeling of the nonlinear actuator involved to further derive analysis and design conditions. It is illustrated on the lateral flying case for a civil aircraft in presence of aggressive maneuvering of the pilot. A complete methodology is then proposed comparing several approaches including given anti-PIO filters (borrowed mainly from [4, 22]).

The chapter is organized as follows. In Sect. 2, a complete model of plant, actuator and controller involved to address the stability and performance optimization problem is described. Then, the multi-objective problem to be solved in order to design anti-windup loops is stated. Section 3 pertains to the anti-windup design conditions in two cases depending on the signal used as the input of the anti-windup controller. Then, in order to alleviate the PIO risk for a civil aircraft in presence of aggressive maneuvering of the pilot, Sect. 4 depicts how the previous techniques are very interesting in comparison with classical anti-PIO filters to guarantee stability and performance of the closed-loop system. Several simulations illustrate the benefits provided by the anti-windup compensators, in terms of simple and systematic methods without needing a tuning parameters step. Finally, some concluding remarks end the chapter.

2 Model Description and Problem Formulation

Anti-windup strategies represent an appropriate framework to mitigate the undesired saturation effects [29, 32]. Thus, the general principle of the anti-windup scheme can be depicted in Fig. 1, where the (unconstrained) signal produced by the controller is compared to that which is actually fed into the plant (the constrained signal). This difference is then used to adjust the control strategy by preserving stability and performance.

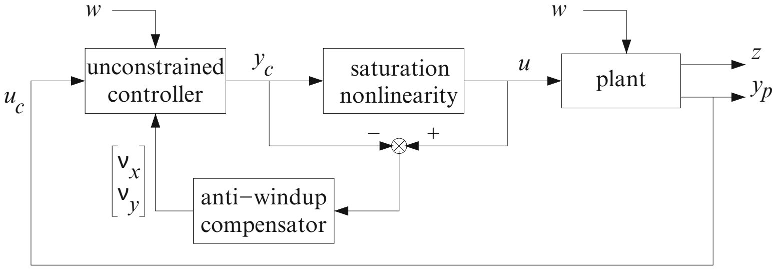


Fig. 1 Principle of anti-windup

The kind of anti-windup controller used in the chapter is specified later and is strongly depending on the considered plant, actuator and controller. Let us first describe the complete model.

2.1 Plant Model

Unlike most systems in the literature, the outputs of the controller are not affected in a same way by the nonlinear elements. Then, the vector $u \in \Re^m$ building the m inputs of the plant is decomposed into two subvectors: the first one, denoted $u_s \in \Re^{m_s}$, corresponds to m_s saturated inputs, whereas the second one, denoted $u_{ns} \in \Re^{m-m_s}$, corresponds to the linear inputs (unsaturated inputs). The plant model can be defined by:

$$\text{sysP} : \begin{cases} \dot{x}_p &= A_p x_p + B^s_{pu} u_s + B^{ns}_{pu} u_{ns} + B_{pw} w \\ y_p &= C_p x_p + D^s_{pu} u_s + D^{ns}_{pu} u_{ns} + D_{pw} w \\ z &= C_z x_p + D^s_{zu} u_s + D^{ns}_{zu} u_{ns} + D_{zw} w \end{cases} \qquad (1)$$

where $x_p \in \Re^{n_p}$ and $y_p \in \Re^p$ are the state and the measured output of the plant. $w \in \Re^q$ generally represents an exogenous perturbation but may also be used to represent a reference signal (or both). Furthermore, $z \in \Re^l$ represents the regulated output, which is used to evaluate the performance of the system with respect to the perturbation w via some pertinent optimization criteria.

2.2 Controller Model

Differently from the classical anti-windup loops, in which the output of the anti-windup controller is injected to the dynamics of the controller and/or the output of the controller, we consider here that the output of the anti-windup controller modifies only partially the dynamics of the controller and/or the output of the controller. Then, with this in mind, the dynamical controller is described as follows:

$$\text{sysC} : \begin{cases} \dot{x}_c &= A_c x_c + B_c u_c + B_{cw} w + B_{ca} v_x \\ y_{cs} &= C_c^s x_c + D_c^s u_c + D_{cw}^s w + D_{ca} v_y \\ y_{cns} &= C_c^{ns} x_c + D_c^{ns} u_c + D_{cw}^{ns} w \end{cases} \tag{2}$$

where $x_c \in \Re^{n_c}$ and $u_c \in \Re^p$ are the state and the input of the controller. The output of the controller is decomposed into two signals: $y_{cs} \in \Re^{m_s}$, which will be interconnected to u_s through a saturated actuator, and $y_{cns} \in \Re^{m-m_s}$, which will be interconnected with the linear (unsaturated) input u_{ns}. Moreover, v_x and v_y are the additional inputs that will be connected to the anti-windup controller.

B_{ca} and D_{ca} are matrices of dimensions $n_c \times n_{cr}$ and $m_s \times m_r$, and allow to specify what are the n_{cr} states and m_r outputs modified by the anti-windup action.

2.3 Actuator Model

There is an actuator block between the output of the controller y_c and the input of the plant u, which is decomposed into two blocks: the first one corresponding to the nonlinear (saturated) part and the second one corresponding to the linear (unsaturated) part. The nonlinear actuator part involves n_{dz} nested saturations, including the case of rate and magnitude saturations, as depicted in Fig. 2a. Such nonlinearities will be tackled via the use of dead-zone, denoted $\phi_i(.)$, $i = 1 \ldots n_{dz}$.

The dynamical model of the actuator is based on Fig. 2b as follows:

$$\text{sysACT} : \begin{cases} \dot{x}_a = v + \phi_1(v) \\ v \;= T_0 y_{cs} + T_0 \phi_0(y_{cs}) - T_0 x_a \\ u_s = x_a \end{cases} \tag{3}$$

with $\phi_0(y_{cs}) = sat_{u_0}(y_{cs}) - y_{cs}$ and $\phi_1(v) = sat_{u_1}(v) - v$, where $sat_{u_0}(.)$ and $sat_{u_1}(.)$ are classical saturation functions and u_0 and u_1 are the levels of saturation

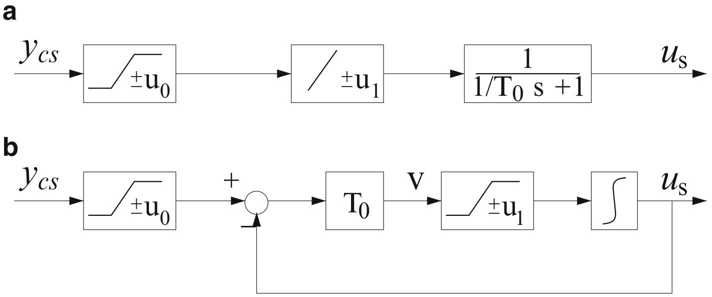


Fig. 2 (**a**) Actuator with rate and magnitude saturations. (**b**) Model used to represent such an actuator (scalar case)

in magnitude and in rate, respectively. The elements of the diagonal matrix $T_0 \in \Re^{m_s \times m_s}$ classically take values large enough in order to avoid affecting the linear dynamics of the closed-loop system.

2.4 Interconnections

The interconnections considered in the chapter can be described as follows:

- linear link between the output of the plant and the input of the controller: $u_c = y_p$;
- the first part of the output of the controller (y_{cs}) is linked to the corresponding inputs of the plant (u_s) through the actuator model (3);
- the second part of the output of the controller is directly connected to the corresponding inputs of the plant: $u_{ns} = y_{cns}$;
- v_x and v_y are built from the anti-windup compensator (and will be specified later).

2.5 Anti-windup Compensator

In the DLAW (Direct Linear Anti-Windup) strategy, the anti-windup controller uses as input the difference between the signals issued either from the input and the output of the whole actuator or from the input and the output of the nonlinear elements included in the actuator. Following this, we pursue two strategies to design the anti-windup loops.

- The first strategy is reported in [4] and considers the difference between the input and the output of the actuator defined by $e = u_s - y_{cs} \in \Re^{m_s}$. Additionally, one assumes that the anti-windup controller only acts on the dynamics of the controller, which corresponds to $v_y = 0$, or equivalently, $m_r = 0$. The anti-windup controller of order n_{aw}, with $v_x \in \Re^{n_{cr}}$, reads:

$$AW_e \ : \ \begin{cases} \dot{x}_{aw} &= A_{aw}x_{aw} + B^e_{aw}(u_s - y_{cs}) \\ v_x &= C_{aw}x_{aw} + D^e_{aw}(u_s - y_{cs}) \end{cases} \tag{4}$$

- The second strategy considers that the input of the anti-windup controller are the dead-zones associated to each saturation. Hence, the anti-windup controller of order n_{aw} reads:

$$AW_\phi \ : \ \begin{cases} \dot{x}_{aw} = A_{aw}x_{aw} + B^0_{aw}\phi_0(y_c) + B^1_{aw}\phi_1(v) \\ \begin{bmatrix} v_x \\ v_y \end{bmatrix} = C_{aw}x_{aw} + D^0_{aw}\phi_0(y_c) + D^1_{aw}\phi_1(v) \end{cases} \tag{5}$$

where v_x and v_y are of dimensions n_{cr} and m_r, respectively.

Remark 1. The interest of the second anti-windup structure resides in the simplicity of the design conditions. Indeed, the design of a static anti-windup gain (only matrices D_{aw}^{0} and D_{aw}^{1} are used) is issued from a fully linear problem. In the case of the design of a dynamical anti-windup controller, for a priori given matrices A_{aw} and C_{aw}, the determination of input and transmission matrices is also obtained by solving a linear problem. In the case where $n_{aw} = n_p + n_{m_s} + n_c$, the resolution of a linear problem can also be considered [29]. At the opposite, the first strategy is more adapted to provide analysis conditions but does not allow to simultaneously compute the matrices of the anti-windup and the matrix of the Lyapunov function through a linear optimization problem, even in the static anti-windup case.

Remark 2. The anti-windup model (5) imposes the assumption that the input and output signals of each saturation block is available. To overcome this assumption, alternative strategies can be investigated. For example, the anti-windup may use the difference between the nonlinear actuator and a linear fictitious one (with the same dynamics but without saturation blocks) to explicitly take into account the dynamics of the actuator (present in the rate limiter) [23]. Another option would be to build an observer to evaluate the internal state of the actuator [27]. In these cases, conditions can be derived in a simpler way than that ones issued from the strategy with (4), but they remain more complex than those due to the strategy with (5).

2.6 Standard Formulation

In [29], a standard formulation of the anti-windup design has been proposed for different kinds of actuators. In the current case, by considering an augmented state of dimensions $n = n_p + m_s + n_c + n_{aw}$ including the state of the plant, the state of the actuator, the state of the controller and the state of the anti-windup controller, the following standard model of the complete closed-loop system can be defined by:

$$\begin{cases} \dot{x} \; = \mathscr{A}x + \mathscr{B}_0\phi_0(y_c) + \mathscr{B}_1\phi_1(v) + \mathscr{B}_2 w \\ y_c = \mathscr{C}_0 x + \mathscr{D}_{00}\phi_0(y_c) + \mathscr{D}_{01}\phi_1(v) + \mathscr{D}_{0w} w \\ v \; = \mathscr{C}_1 x + \mathscr{D}_{10}\phi_0(y_c) + \mathscr{D}_{11}\phi_1(v) + \mathscr{D}_{1w} w \\ z \; = \mathscr{C}_2 x + \mathscr{D}_{20}\phi_0(y_c) + \mathscr{D}_{21}\phi_1(v) + \mathscr{D}_{2w} w \end{cases} \tag{6}$$

Then, depending of the anti-windup scheme under consideration, the matrices of the anti-windup controller are encapsulated into the matrices of system (6).

The design procedure of the anti-windup controller consists in optimizing some quantities as the size of the region of stability of the closed-loop system or the guaranteed level of performance. Several optimization problems are then of interest. In particular, the idea by adding the anti-windup loop is to maximize the basin of attraction of the origin for the closed-loop system and/or to minimize the $\mathscr{L}_2$ gain between w and z or to maximize the set of perturbation w, which can be rejected. Then, throughout the chapter, the signal of perturbation is supposed to be bounded in energy as follows:

$$\|w\|_2^2 = \int_0^\infty w'(t)w(t)dt \leq \delta^{-1}\ ; \quad 0 \leq \delta^{-1} < \infty \tag{7}$$

The problem we intend to address in the chapter can be summarized below.

Problem 1. Determine an anti-windup controller and a region $\mathscr{E}$, as large as possible, such that

- Internal stability. The closed-loop system (6) with $w = 0$ is asymptotically stable for any initial conditions belonging to $\mathscr{E}$ [which is a region of asymptotic stability (RAS)];
- Performance. The $\mathscr{L}_2$ gain between w and z is finite and equal to $\gamma > 0$.

The convex optimization problems associated to Problem 1 are specified in Sects. 3.2 and 3.3.

3 Main Anti-windup Design Conditions

3.1 Solution to Standard Anti-windup Design

The following proposition provides conditions of local stability and $\mathscr{L}_2$ performance for the closed-loop system (6). The result regards existence conditions to solve Problem 1.

Proposition 1. *If there exist a symmetric positive definite matrix $Q \in \Re^{n\times n}$, two matrices Z_0 and $Z_1 \in \Re^{m\times n}$, two positive diagonal matrices S_0 and $S_1 \in \Re^{m\times m}$ and a positive scalar γ such that the following conditions are verified:*

$$\begin{bmatrix} Q\mathscr{A}' + \mathscr{A}Q & \mathscr{B}_0S_0 - Q\mathscr{C}_0' - Z_0' & \mathscr{B}_1S_1 - Q\mathscr{C}_1' - Z_1' & \mathscr{B}_2 & Q\mathscr{C}_2' \\ \star & -2S_0 - \mathscr{D}_{00}S_0 - S_0\mathscr{D}_{00}' & -\mathscr{D}_{01}S_1 - S_0\mathscr{D}_{10}' & -\mathscr{D}_{0w} & S_0\mathscr{D}_{20}' \\ \star & \star & -2S_1 - \mathscr{D}_{11}S_1 - S_1\mathscr{D}_{11}' & -\mathscr{D}_{1w} & S_1\mathscr{D}_{21}' \\ \star & \star & \star & -I & \mathscr{D}_{2w}' \\ \star & \star & \star & \star & -\gamma I \end{bmatrix} < 0 \tag{8}$$

$$\begin{bmatrix} Q & Z_{0(i)}' \\ \star & \delta u_{0(i)}^2 \end{bmatrix} \geq 0, \quad i = 1, \ldots, m \tag{9}$$

$$\begin{bmatrix} Q & Z_{1(i)}' \\ \star & \delta u_{1(i)}^2 \end{bmatrix} \geq 0, \ i = 1, \ldots, m \tag{10}$$

then,

1. *when* $w = 0$*, the set* $\mathscr{E}(Q^{-1}, \delta) = \{x \in \Re^n; x'Q^{-1}x \leq \delta^{-1}\}$ *is RAS for the closed-loop system (6);*
2. *when* $w \neq 0$*, satisfying (7), and for* $x(0) = 0$*,*
   - *the trajectories of the closed-loop system remain bounded in the set* $\mathscr{E}(Q^{-1}, \delta)$*;*
   - *the* $\mathscr{L}_2$ *gain is finite and one gets:*

$$\int_0^T z(t)'z(t)\,dt \leq \gamma \int_0^T w(t)'w(t)\,dt, \forall T \geq 0 \tag{11}$$

In an analysis purpose (the anti-windup controller being given), conditions of Proposition 1 are linear and can be directly used to solve adequate optimization problems. Moreover, in the design context, conditions of Proposition 1 are non convex, matrices A_{aw}, B_{aw}, C_{aw} and D_{aw} being hidden in matrices $\mathscr{A}$, $\mathscr{B}_i$, $\mathscr{C}_i$, $\mathscr{D}_{ij}$, $i, j = 0, 1$. Depending on the problem studied, conditions linear in the decision variables can be obtained, more or less directly, by modifying a bit the original conditions or still by considering iterative procedures (including D-K iteration process) allowing to search for Lyapunov matrix and anti-windup matrices. These situations are detailed in the next sections.

Remark 3. In the sequel, one considers a set $\mathscr{X}_0$, defined by some directions in the plant state space $v_i \in \Re^{n_p}$, $i = 1, \ldots, q$, to provide a desired shape of the region $\mathscr{E}(Q^{-1}, \delta)$ to be maximized when solving Problem 1. Then, considering $\bar{v}_i = \begin{bmatrix} v_i' & 0 \end{bmatrix}' \in \Re^n$, $i = 1, \ldots, q$ and β a scaling factor such that $\beta\mathscr{X}_0 \subset \mathscr{E}(Q^{-1}, \delta)$, an additional condition to those of Proposition 1 have to be considered in the algorithms which follow:

$$\begin{bmatrix} \delta \frac{1}{\beta^2} & \delta \bar{v}_i' \\ \delta \bar{v}_i & Q \end{bmatrix} > 0, \; i = 1, \ldots, q \tag{12}$$

3.2 Algorithms for AW_e Case

From (1), (2), (3) and (4), matrices of system (6) read:

$$\begin{array}{l}
\mathscr{A} = \begin{bmatrix} \mathbb{A} & 0 \\ 0 & 0 \end{bmatrix} + B_{\nu B} A_{aw} C_{\nu A} + B_{\nu B} B^e_{aw} C_{\nu C} + B_{\nu D} C_{aw} C_{\nu A} + B_{\nu D} D^e_{aw} C_{\nu C} \\
\mathscr{B}_0 = \begin{bmatrix} B_{\phi 0} \\ 0 \end{bmatrix}; \; \mathscr{B}_1 = \begin{bmatrix} B_{\phi 1} \\ 0 \end{bmatrix}; \; \mathscr{B}_2 = \begin{bmatrix} B_2 \\ 0 \end{bmatrix} + B_{\nu B} B^e_{aw} C_{\nu W} + B_{\nu D} D^e_{aw} D_{0w} \\
\mathscr{C}_0 = \begin{bmatrix} C_0 & 0 \end{bmatrix}; \; \mathscr{C}_1 = \begin{bmatrix} C_1 & 0 \end{bmatrix}; \; \mathscr{C}_2 = \begin{bmatrix} C_2 & 0 \end{bmatrix} \\
\mathscr{D}_{00} = 0 \; ; \; \mathscr{D}_{01} = 0 \; ; \; \mathscr{D}_{10} = D_1 \; ; \; \mathscr{D}_{11} = 0 \\
\mathscr{D}_{20} = 0 \; ; \; \mathscr{D}_{21} = 0 \; ; \; \mathscr{D}_{0w} = D_{0w} \; ; \; \mathscr{D}_{1w} = D_{1w} \; ; \; \mathscr{D}_{2w} = D_{2w}
\end{array} \tag{13}$$

with

$$\mathbb{A} = \begin{bmatrix} A_p + B_{pu}^{ns}\Delta^{-1}D_c^{ns}C_p & B_{pu}^{s} + B_{pu}^{ns}\Delta^{-1}D_c^{ns}D_{pu}^{s} & B_{pu}^{ns}\Delta^{-1}C_c^{ns} \\ T_0 D_c^{s}(C_p + D_{pu}^{ns}\Delta^{-1}D_c^{ns}C_p) & \begin{array}{c} T_0(D_c^{s}D_{pu}^{s} - I_{ms} \\ \quad + D_c^{s}D_{pu}^{ns}\Delta^{-1}D_c^{ns}D_{pu}^{s}) \end{array} & T_0(C_c^{s} + D_c^{s}D_{pu}^{ns}\Delta^{-1}C_c^{ns}) \\ B_c C_p + B_c D_{pu}^{ns}\Delta^{-1}D_c^{ns}C_p & B_c D_{pu}^{s} + B_c D_{pu}^{ns}\Delta^{-1}D_c^{ns}D_{pu}^{s} & A_c + B_c D_{pu}^{ns}\Delta^{-1}C_c^{ns} \end{bmatrix}$$

$$B_2 = \begin{bmatrix} B_{pw} + B_{pu}^{ns}\Delta^{-1}(D_c^{ns}D_{pw} + D_{cw}^{ns}) \\ T_0(D_c^{s}D_{pw} + D_{cw}^{s} + D_c^{s}D_{pu}^{ns}\Delta^{-1}(D_c^{ns}D_{pw} + D_{cw}^{ns})) \\ B_{cw} + B_c D_{pw} + B_c D_{pu}^{ns}\Delta^{-1}(D_c^{ns}D_{pw} + D_{cw}^{ns}) \end{bmatrix}$$

$$B_{\phi 0} = \begin{bmatrix} 0 \\ T_0 \\ 0 \end{bmatrix} \; ; \; B_{\phi 1} = \begin{bmatrix} 0 \\ I_{m_s} \\ 0 \end{bmatrix} \; ; \; D_1 = T_0$$

$$\begin{aligned} C_0 &= \Big[\, D_c^{s}(I_p + D_{pu}^{ns}\Delta^{-1}D_c^{ns})C_p \quad D_c^{s}(I_p + D_{pu}^{ns}\Delta^{-1}D_c^{ns})D_{pu}^{s} \quad C_c^{s} + D_c^{s}D_{pu}^{ns}\Delta^{-1}C_c^{ns} \,\Big] \\ C_1 &= \Big[\, T_0 D_c^{s}(I_p + D_{pu}^{ns}\Delta^{-1}D_c^{ns})C_p \quad T_0 D_c^{s}(I_p + D_{pu}^{ns}\Delta^{-1}D_c^{ns})D_{pu}^{s} - T_0 \\ &\qquad\qquad\qquad\qquad\qquad\qquad T_0(C_c^{s} + D_c^{s}D_{pu}^{ns}\Delta^{-1}C_c^{ns}) \,\Big] \\ C_2 &= \Big[\, C_z + D_{zu}^{ns}\Delta^{-1}D_c^{ns}C_p \quad D_{zu}^{s} + D_{zu}^{ns}\Delta^{-1}D_c^{ns}D_{pu}^{s} \quad D_{zu}^{ns}\Delta^{-1}C_c^{ns} \,\Big] \end{aligned}$$

$$\begin{aligned} D_{0w} &= D_{cw}^{s} + D_c^{s}D_{pw} + D_c^{s}D_{pu}^{ns}\Delta^{-1}(D_c^{ns}D_{pw} + D_{cw}^{ns}) \\ D_{1w} &= T_0(D_{cw}^{s} + D_c^{s}D_{pw} + D_c^{s}D_{pu}^{ns}\Delta^{-1}(D_c^{ns}D_{pw} + D_{cw}^{ns})) \\ D_{2w} &= D_{zw} + D_{zu}^{ns}\Delta^{-1}(D_c^{ns}D_{pw} + D_{cw}^{ns}) \end{aligned}$$

$$B_{\nu B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ I_{n_{aw}} \end{bmatrix} \; ; \; B_{\nu D} = \begin{bmatrix} 0 \\ 0 \\ B_{ca} \\ 0 \end{bmatrix} \; ; \; C_{\nu A} = \begin{bmatrix} 0 \; 0 \; 0 \; I_{n_{aw}} \end{bmatrix}$$

$$\begin{aligned} C_{\nu C} = \Big[& -D_c^{s}(I_p + D_{pu}^{ns}\Delta^{-1}D_c^{ns})C_p \qquad I_m - D_c^{s}(I_p + D_{pu}^{ns}\Delta^{-1}D_c^{ns})D_{pu}^{s} \\ & \qquad\qquad\qquad\qquad\qquad\qquad -C_c^{s} - D_c^{s}D_{pu}^{ns}\Delta^{-1}C_c^{ns} \qquad 0 \Big] \end{aligned}$$

and $\Delta = I_{m-ms} - D_c^{ns}D_{pu}^{ns}$.

In the analysis context, conditions of Proposition 1 using the AW_e structure are linear in the decision variables and can be directly used. On the other hand, in the design context, conditions of Proposition 1 are nonlinear due to, in particular, the products between the Lyapunov matrix Q and the matrices of the anti-windup controller. Then, to address the design and solve Problem 1, some iterative procedure can be applied by considering at the first step a given static ($n_{aw} = 0$) or dynamic ($n_{aw} \neq 0$) anti-windup controller.

The following algorithms can be used.

Algorithm 3.1. *Analysis of a given AW_e anti-windup controller*

1. *Give matrices A_{aw}, B^e_{aw}, C_{aw} and D^e_{aw}.*
2. *Choose directions to be optimized $v_i \in \Re^{n_p}$, $i = 1, \ldots, q$ and a known perturbation with bound δ.*
3. *Solve*

$$\min_{Q,S_0,S_1,Z_0,Z_1,\gamma,\mu} \gamma + \mu$$
$$\textit{subject to LMI (8), (9), (10) and (12)}$$

   where γ is the $\mathscr{L}_2$ gain between w and z and $\mu = 1/\beta^2$.

Algorithm 3.2. *Design of a AW_e anti-windup controller*

1. *Select an initial guess for matrices A_{aw}, B^e_{aw}, C_{aw} and D^e_{aw}. of appropriate dimensions in order to build the desired anti-windup loop. A static anti-windup AW_e may also be used by considering $n_{aw} = 0$.*
2. *Choose directions to be optimized $v_i \in \Re^{n_p}$, $i = 1, \ldots, q$ and a known perturbation with bound δ.*
3. *Analysis step—Solve*

$$\min_{Q,S_0,S_1,Z_0,Z_1,\gamma,\mu} \gamma + \mu$$
$$\textit{subject to LMI (8), (9), (10) and (12)}$$

   where γ is the $\mathscr{L}_2$ gain between w and z and $\mu = 1/\beta^2$.
4. *If the solution obtained is satisfactory (some accuracy has to be fixed) or no more improved from the previous steps then STOP. Otherwise, go to the next iteration (the idea is to finish by an analysis step).*
5. *Synthesis step—Pick the solution Q obtained at Step 3 and solve*

$$\min_{A_{aw},B^e_{aw},C_{aw},D^e_{aw},S_0,S_1,Z_0,Z_1,\gamma} \gamma$$
$$\textit{subject to LMI (8), (9), (10) and (12)}$$

6. *Go to Step 3.*

Remark 4. The selection of an initial guess of anti-windup in the Algorithm 3.2 must take care of the dimension of each elements but must also verify that A_{aw} is Hurwitz. Actually, it is not possible to initialize the problem with null matrices of appropriate dimensions (for a given order of the anti-windup scheme n_{aw}) as the condition on $\mathscr{A}$ in the first block of inequality (8) imposes that both the closed-loop linear dynamics of the system and the anti-windup dynamics are asymptotically stable. An option may then be to select any stable dynamical matrix A_{aw} with matrices B^e_{aw}, C_{aw} and D^e_{aw} equal to null matrices of appropriate dimensions. This initial anti-windup scheme is ineffective but allows to solve the analysis steep and obtain a matrix Q to be used in the synthesis step.

3.3 Algorithms for AW$_\phi$ Case

As for the previous case, from (1), (2), (3) and (5), matrices of system (6) are defined by:

$$\begin{array}{c}
\mathscr{A} = \begin{bmatrix} \mathbb{A} & B_v C_{aw} \\ 0 & A_{aw} \end{bmatrix};\ \mathscr{B}_0 = \begin{bmatrix} B_{\phi 0} + B_v D_{aw}^0 \\ B_{aw}^0 \end{bmatrix};\ \mathscr{B}_1 = \begin{bmatrix} B_{\phi 1} + B_v D_{aw}^1 \\ B_{aw}^1 \end{bmatrix} \\
\mathscr{C}_0 = \begin{bmatrix} C_0 & C_{v0} C_{aw} \end{bmatrix};\ \mathscr{C}_1 = \begin{bmatrix} C_1 & C_{v1} C_{aw} \end{bmatrix};\ \mathscr{C}_2 = \begin{bmatrix} C_2 & 0 \end{bmatrix} \\
\mathscr{D}_{00} = C_{v0} D_{aw}^0\ ;\ \mathscr{D}_{01} = C_{v0} D_{aw}^1\ ;\ \mathscr{D}_{10} = D_1 + C_{v1} D_{aw}^0\ ;\ \mathscr{D}_{11} = C_{v1} D_{aw}^1 \\
\mathscr{B}_2 = \begin{bmatrix} B_2 \\ 0 \end{bmatrix}\ ;\ \mathscr{D}_{20} = 0\ ;\ \mathscr{D}_{21} = 0\ ;
\end{array} \tag{14}$$

Matrices $\mathbb{A}$, B_2, $B_{\phi 0}$, $B_{\phi 1}$, D_1, C_0, C_1, C_2, $\mathscr{D}_{0w}$, $\mathscr{D}_{1w}$ and $\mathscr{D}_{2w}$ remain unchanged. Matrices defining the interconnection between the anti-windup loop and the system are:

$$B_v = \begin{bmatrix} 0 \\ T_0 D_{ca} \begin{bmatrix} 0 & I_{m_r} \end{bmatrix} \\ B_{ca} \begin{bmatrix} I_{n_{cr}} & 0 \end{bmatrix} \end{bmatrix}\ ;\ C_{v0} = D_{ca} \begin{bmatrix} 0 & I_{m_r} \end{bmatrix};\ C_{v1} = T_0 D_{ca} \begin{bmatrix} 0 & I_{m_r} \end{bmatrix}$$

As in the previous case, the analysis problem (Algorithm 3.3) is linear and the synthesis problem of the anti-windup is nonlinear, including products between decision variables, and in particular between the Lyapunov matrix Q and the anti-windup elements. As for the AW_e strategy, a D-K iteration procedure may then be considered for the synthesis problem (Algorithm 3.4). However, differently for the AW_e strategy, the synthesis optimization problem may be partially linearized and, for given matrices A_{aw} and C_{aw}, the design of matrices B_{aw}^i and D_{aw}^i, $i = 0, 1$ can be handled via a linear optimization problem (Algorithm 3.5).

Algorithm 3.3. *Analysis of a given AW$_\phi$ anti-windup controller*

1. *Select matrices A_{aw}, B_{aw}^0, B_{aw}^1, C_{aw}, D_{aw}^0 and D_{aw}^1.*
2. *Choose directions to be optimized $v_i \in \Re^{n_p}$, $i = 1, \ldots, q$ and a known perturbation with the bound δ.*
3. *Solve*

$$\min_{Q, S_0, S_1, Z_0, Z_1, \gamma, \mu} \gamma + \mu$$
$$\text{subject to LMI (8), (9), (10) and (12)}$$

where γ is the $\mathscr{L}_2$ gain between w and z and $\mu = 1/\beta^2$.

Algorithm 3.4. *Design of a AW_ϕ anti-windup controller*

1. *Select matrices A_{aw}, B^0_{aw}, B^1_{aw} C_{aw}, D^0_{aw} and D^1_{aw} of appropriate dimensions in order to build the desired anti-windup loop. A static anti-windup AW_ϕ may also be used by considering $n_{aw} = 0$.*
2. *Choose directions to be optimized $v_i \in \Re^{n_p}$, $i = 1, \ldots, q$ and a known perturbation with bound δ.*
3. *Analysis step—Solve*

$$\min_{Q,S_0,S_1,Z_0,Z_1,\gamma,\mu} \gamma + \mu$$

subject to LMI (8), (9), (10) and (12)

where γ is the $\mathscr{L}_2$ gain between w and z and $\mu = 1/\beta^2$.

4. *If the solution obtained is satisfactory (some accuracy has to be fixed) or no more improved from the previous steps then STOP. Otherwise, go to the next iteration (the idea is to finish by an analysis step).*
5. *Synthesis step—Pick the solution Q obtained at Step 3 and solve*

$$\min_{S_0,S_1,Z_0,Z_1,B^0_{aw},B^1_{aw},D^0_{aw},D^1_{aw},\gamma} \gamma$$

subject to LMI (8), (9), (10) and (12)

6. *Go to Step 3.*

Algorithm 3.5. *Design of a AW_ϕ anti-windup controller with fixed dynamics*

1. *Give matrices A_{aw} and C_{aw}.*
2. *Choose directions to be optimized $v_i \in \Re^{n_p}$, $i = 1, \ldots, q$ and a known perturbation with the bound δ.*
3. *Solve*

$$\min_{Q,S_0,S_1,Z_0,Z_1,\bar{B}^0_{aw},\bar{B}^1_{aw},\bar{D}^0_{aw},\bar{D}^1_{aw},\gamma,\mu} \gamma + \mu$$

subject to LMI (8), (9), (10) and (12)

where γ is the $\mathscr{L}_2$ gain between w and z and $\mu = 1/\beta^2$.

4. *Compute $B^0_{aw} = \bar{B}^0_{aw} S_0^{-1}$, $B^1_{aw} = \bar{B}^1_{aw} S_1^{-1}$, $D^0_{aw} = \bar{D}^0_{aw} S_0^{-1}$ and $D^1_{aw} = \bar{D}^1_{aw} S_1^{-1}$.*

Remark 5. In Algorithm 3.5, condition (8) is not directly applied. The products between B^i_{aw} and D^i_{aw} with the matrices S_i are replaced by the change of variables $\bar{B}^i_{aw}$ and $\bar{D}^i_{aw}$, $i = 0, 1$, which allows to linearize the problem.

Remark 6. An interesting case is the static anti-windup one, for which matrices A_{aw} and C_{aw} are null matrices of appropriate dimensions. It implies that B^i_{aw}, $i = 0, 1$ are also null matrices of appropriate dimensions and only matrices D^i_{aw}, $i = 0, 1$ are computed in the linear Algorithm 3.5.

Remark 7. Matrices A_{aw} and C_{aw} to be used in Algorithm 3.5 may be selected as the solution of a full-order ($n_{aw} = n_p + n_c + m_s$) anti-windup compensator design where the actuator is just a saturation in magnitude (see, for example, the conditions provided in [29]), i.e. via a linear optimization problem. Eventually, an order-reduction step may also be considered in order to select matrices A_{aw} and C_{aw} (see Example 8.5 in [29]). Other procedures developed in [32] could be used.

4 Anti-windup and Its Use for PIO Alleviation

The design and analysis algorithms of Sect. 3 are now applied and compared in the realistic context of lateral maneuvers of a civil transport aircraft. A specific attention will be devoted to aggressive pilot's demands in conjunction with actuator loss.

In the following, the pilot's activity is modeled as a static gain K_{pil}. For this application, a normal activity would correspond to $K_{pil} = 1$. But, in stressful situations, notably in case of actuators loss, a more aggressive pilot's behavior is generally observed, resulting in much higher gains. Here, the gain is set to $K_{pil} = 3$.

4.1 Problem Setup and Objectives

The two anti-windup structures AW_e and AW_ϕ above described are compared in this applicative part of the chapter. Both nonlinear closed-loop Simulink implementations are sketched in Figs. 3 and 4, respectively. For each design strategy, the state-space models *sysP* and *sysC* are readily obtained from the Simulink

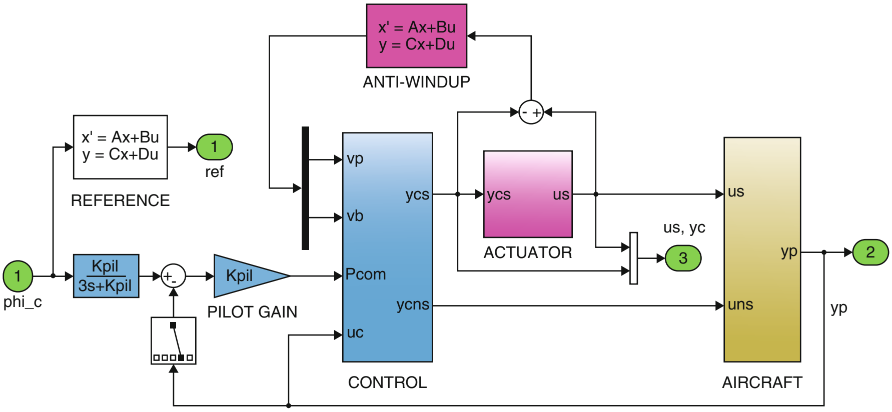


Fig. 3 Nonlinear closed-loop Simulink implementation of Anti-Windup *AWe* for lateral aircraft simulations

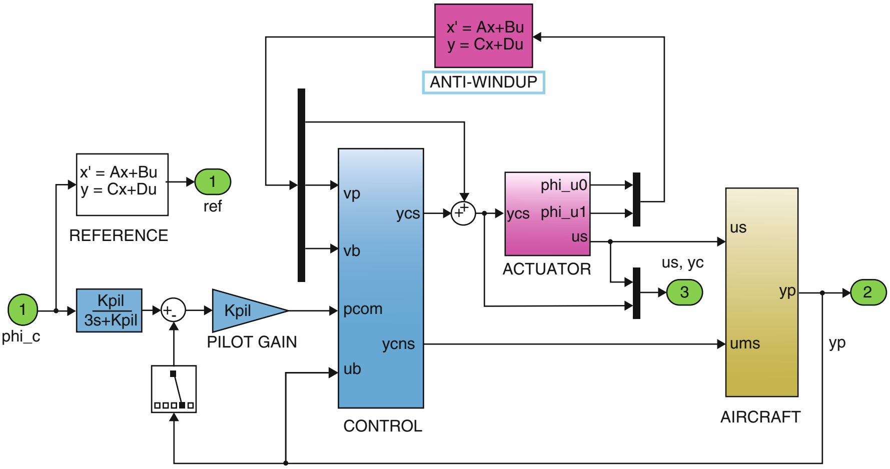


Fig. 4 Nonlinear closed-loop Simulink implementation of Anti-Windup AW_ϕ for lateral aircraft simulations

diagrams of Figs. 3 and 4 with the help of the Matlab **linmod** function. The plant corresponds to the "yellow box" depicting the aircraft system while the global controller (including pilot actions) is obtained by extraction of the three blue boxes. A standard balanced reduction technique is finally applied to obtain reasonably sized models. The obtained reduced orders, respectively $n_p = 8$ and $n_c = 20$, are compatible with the proposed algorithms. The aircraft system involves two inputs ($m = 2$) with only the ailerons deflection actuator which saturates ($m_s = 1$), and five outputs ($p = 5$), among which the performance output which is set as the roll angle ($l = 1$). The disturbance of the system is due to the saturation of the input, i.e., $B_{pw} = B^s_{pu}$ ($q = 1$).

In the AW_e strategy, the anti-windup input is a single scalar signal ($m_s = 1$) which only captures the difference between the input of the nonlinear ailerons deflection actuator and its output. In the AW_ϕ strategy, two signals (one for the magnitude limitation and one for the rate limitation) are used by the anti-windup device. Their generation is detailed in the Simulink implementation of Fig. 5.

Whatever the considered approach, both anti-windup controllers act similarly on the internal dynamics of the nominal lateral controller of the aircraft through two scalar signals v_p and v_b which respectively affect roll and sideslip angles dynamics ($v_x = \left[v_p \;\; v_b\right]'$ and $v_y = 0$, $n_{cr} = 2$, $m_r = 0$). Remark yet that the second strategy offers more flexibility with the possibility of a direct anti-windup action at the controller output. However, no significant improvement has been observed with this additional feature which has thus not been further considered in this application.

The main objective of this application is to design and evaluate anti-windup systems to improve the aircraft response to roll angle solicitations while limiting oscillations despite actuator loss [22]. During such maneuvers, a significant control activity is observed on the ailerons. This is why the effects of saturations are

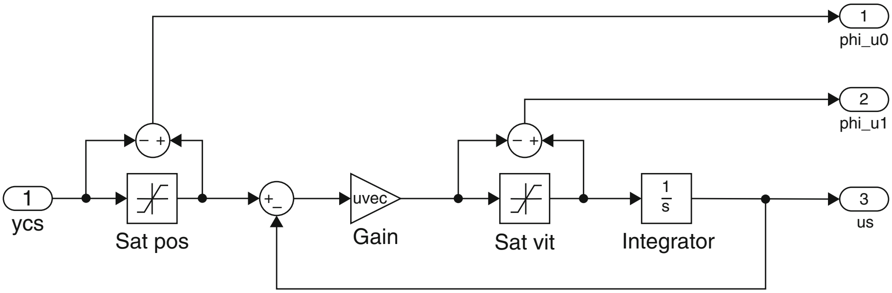


Fig. 5 Detailed view of the magnitude and rate limitations system

modeled and taken into account for these actuators in both diagrams of Figs. 3 and 4 while no saturation is introduced on the rudders. The effects of saturations become even more penalizing in case of a partial loss of control capability. Assume indeed that the aircraft is controlled by a pair of ailerons on each wing but only one is operational. In that case, the activity of the remaining actuators is doubled as well as the risk of magnitude and rate saturations. Then, the magnitude and rate limits in the following will be halved. We will consider $L_m = 10°$ (instead of 20 in normal conditions) and $L_r = 20°/\mathrm{s}$ (instead of 40).

In the sequel, five different anti-windup compensators are implemented and compared:

- A standard anti-PIO filter used in the industry. It is an "open-loop" solution which does not exploit the information relative to the saturation of the signal (see [4]). This may be considered as the basic solution from the industry. It corresponds to the block REFERENCE in Figs. 3 and 4;
- A dynamic $\mathscr{H}_\infty AW_e$ anti-windup built by using a structured $\mathscr{H}_\infty$ design method [22]. The advantage of such a strategy is that it circumvents some limitations of LMI-based strategies (limitation on the size problem when manipulating LMIs, conservatism of sufficient conditions) but to the detriment of the easiness of construction for engineers not always specialists of advanced control theories;
- A dynamic AW_e anti-windup designed with the Algorithm 3.2 initialized with the $\mathscr{H}_\infty AW_e$ anti-windup above-described;
- A dynamic AW_ϕ anti-windup designed with the Algorithm 3.5 using matrices A_{aw} and C_{aw} borrowed to the $\mathscr{H}_\infty AW_e$ anti-windup;
- A static AW_ϕ anti-windup designed with the Algorithm 3.5. This strategy is an alternative to the standard anti-PIO filter as it is very easy to implement (no additional dynamical system to introduce in the controller block).

4.2 Analysis and Design of Anti-windup AW_e

First, an analysis step (in terms of stability and performance) of the anti-windup controller proposed in [4] (denoted $\mathscr{H}_\infty AW_e$ anti-windup) is carried out. This anti-windup compensator has been built by using a structured $\mathscr{H}_\infty$ design method [22]. Such an anti-windup has provided very good numerical results but no proof of its stability in closed-loop was a priori ensured. By using Algorithm 3.1, one can verify that the conditions are feasible. The optimization problem is then solved by considering the bound on perturbation $\delta = 0.1$ and $v_1 = [C_p(4,:)\ \ 0]$ as the direction to be optimized over the set $\mathscr{E}(Q^{-1}, \delta)$. Algorithm 3.1 gives the optimal solution:

$$\text{Analysis } \mathscr{H}_\infty AW_e : \quad \gamma = 1.7167\ ; \quad \beta = 0.6254$$

It is interesting to do the same analysis for the closed-loop system without anti-windup. The feasibility is also obtained and the solution is:

$$\text{Analysis without anti-windup: } \gamma = 1.8929\ ; \quad \beta = 0.7851$$

The solution with the $\mathscr{H}_\infty AW_e$ anti-windup described through γ and β as performance indicators does not appear as much better than the one without anti-windup: indeed γ is actually decreased in the case with anti-windup but β is slightly degraded. However, simulations presented in [4] exhibited that the time responses with the $\mathscr{H}_\infty AW_e$ strategy were very close to the desired behavior (that is without saturation), differently from the case without anti-windup resulting in large overshoot and degraded time evolution. The meaning of this is that the considered criterion of optimization, which does not explicitly include the time response performance, does not exactly fit to the analysis or design of the anti-windup loop. Nevertheless, considering criteria on time response performance is a difficult task and the optimization criterion used here gives a reasonable trade-off between stability guarantee, performance and time response.

In a second step, Algorithm 3.2 is used by considering the previous anti-windup controller at the initialization step (Step 1). After one iteration (after the conditions become unfeasible for numerical reasons), one gets a new anti-windup controller AW_e such that:

$$\text{Design } AW_e : \quad \gamma = 1.6887\ ; \quad \beta = 0.6978$$

Now, we compare the results obtained in response to a step demand of 40° on the roll angle, by using the scheme given in Fig. 3.

In Fig. 6, the time evolutions obtained with the $\mathscr{H}_\infty AW_e$ anti-windup (used in the first step of analysis) and the AW_e anti-windup above designed are compared. The case without saturation is also plotted (denoted “reference” in the figure). The time evolution of inputs δ_{pc} is plotted in both cases and without saturation in Fig. 7.

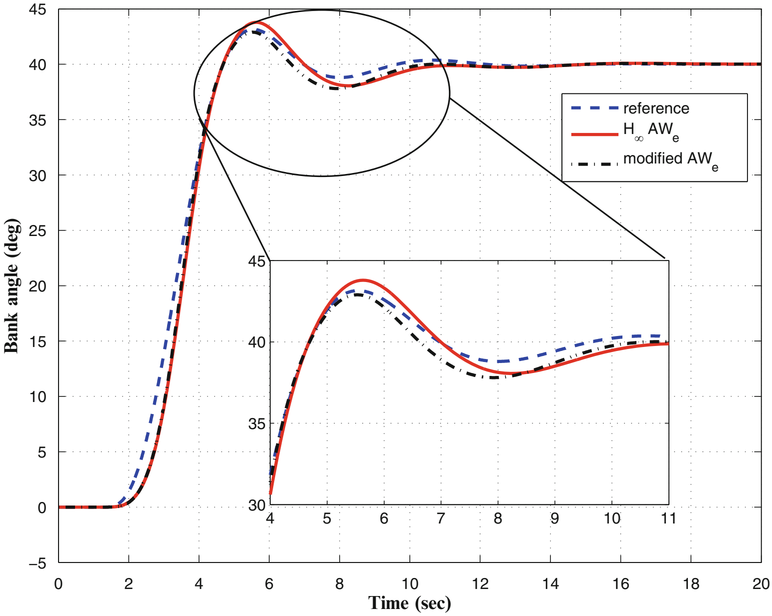


Fig. 6 Roll solicitation of 40°: comparison of the performance outputs for the cases with $\mathscr{H}_\infty AW_e$ anti-windup and designed anti-windup AW_e

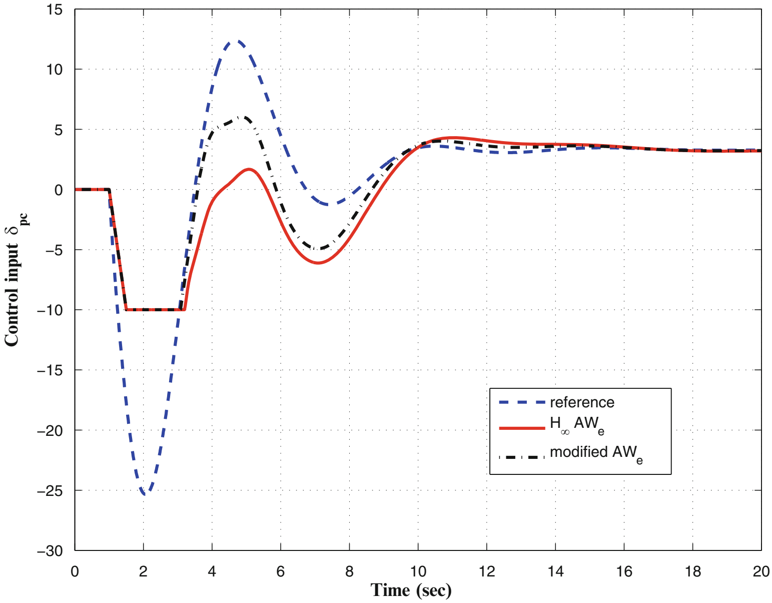


Fig. 7 Roll solicitation of 40°: comparison of the inputs for the cases with $\mathscr{H}_\infty AW_e$ anti-windup and designed anti-windup AW_e

Remark 8. Note that the level of performance obtained with the $\mathscr{H}_\infty AW_e$ anti-windup cannot be much improved with our strategy as this “initial” anti-windup had been cleverly designed with the structured $\mathscr{H}_\infty$ approach. The iterative procedure could be initialized with any other anti-windup controller of order n_{aw}, for example (as suggested before) solution of a design in the case of magnitude saturation only. The initial choice has however a significant influence on the iterative process and the attainable solution. Many tests have shown that the results obtained are often not much convincing (generally the solution extended from the initial $\mathscr{H}_\infty AW_e$ anti-windup shows better performance indexes) and then it seems that this anti-windup structure is not much adapted to a design from scratch.

4.3 Design of Anti-windup AW$_\phi$

In this section, we consider the second strategy, whose main advantage is that it is based on a systematic method without tuning parameter. Figure 4 shows how the AW_ϕ anti-windup is implemented.

As commented in Sect. 3, in the context of the AW_ϕ strategy, the design of matrices B^i_{aw} and D^i_{aw} is cast by solving a linear optimization problem. Then, for the same conditions on δ and $\mathscr{X}_0$ as in the previous case, and by considering matrices A_{aw} and B_{aw} of the $\mathscr{H}_\infty AW_e$ anti-windup, Algorithm 3.5 gives matrices B^i_{aw} and D^i_{aw}, $i = 0, 1$, and the following optimal solution

$$\text{Design of dynamical } AW_\phi : \quad \gamma = 1.8441 \; ; \quad \beta = 0.9013$$

As previously, we consider a Roll solicitation of 40° to compare the results. The time responses of the roll angle for the case without saturation, with the $\mathscr{H}_\infty AW_e$ anti-windup and the designed dynamic AW_ϕ anti-windup are plotted in Fig. 8. Similarly, the time evolutions of δ_{pc} in these cases are depicted in Fig. 9.

One can observe that the level of performance of the $\mathscr{H}_\infty AW_e$ anti-windup is slightly degraded in the case of the design of AW_ϕ, but it remains acceptable.

Now, we design a static AW_ϕ anti-windup (only matrices D^i_{aw}, $i = 0, 1$, have to be designed). The main advantage is that we do not need to initialize the algorithm as matrices A_{aw} and C_{aw} do not exist ($n_{aw} = 0$). Algorithm 3.5 provides the following optimal solution:

$$\text{Static } AW_\phi \text{ design} : \quad \gamma = 1.7846 \; ; \quad \beta = 0.7772$$

Figures 10 and 11 illustrate the time evolution of the closed-loop system to a roll solicitation of 40°. The responses are compared by considering the case without saturation, a standard anti-PIO strategy (see [4]) and the static AW_ϕ anti-windup strategy. It is important to underline that a simple static anti-windup strategy allows to obtain better performance than the standard anti-PIO case, which adds dynamics in the system.

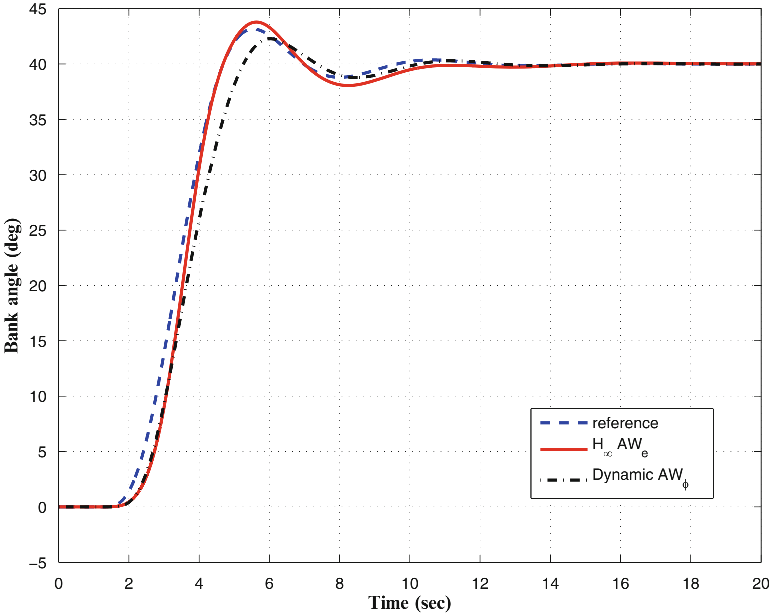


Fig. 8 Roll solicitation of 40°: comparison of the performance outputs for the cases with $\mathscr{H}_\infty AW_e$ anti-windup and designed anti-windup AW_ϕ

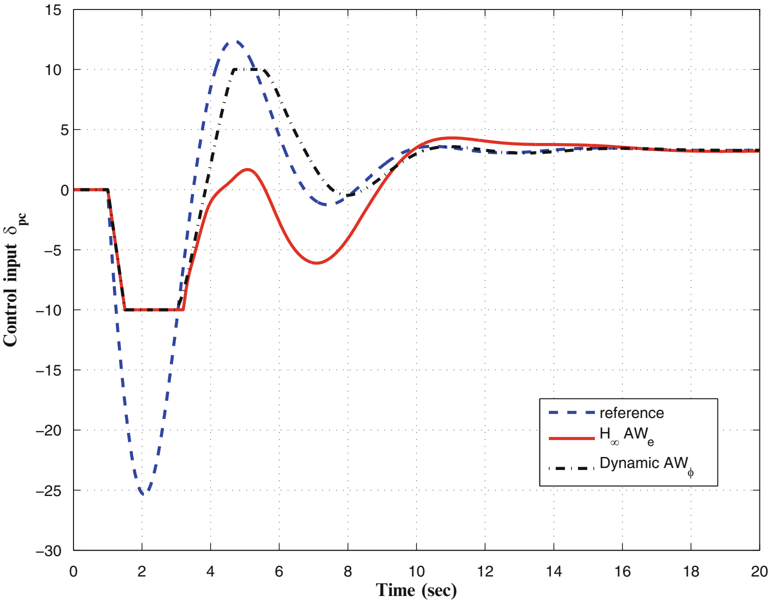


Fig. 9 Roll solicitation of 40°: comparison of the inputs for the cases with $\mathscr{H}_\infty AW_e$ anti-windup and designed anti-windup AW_ϕ

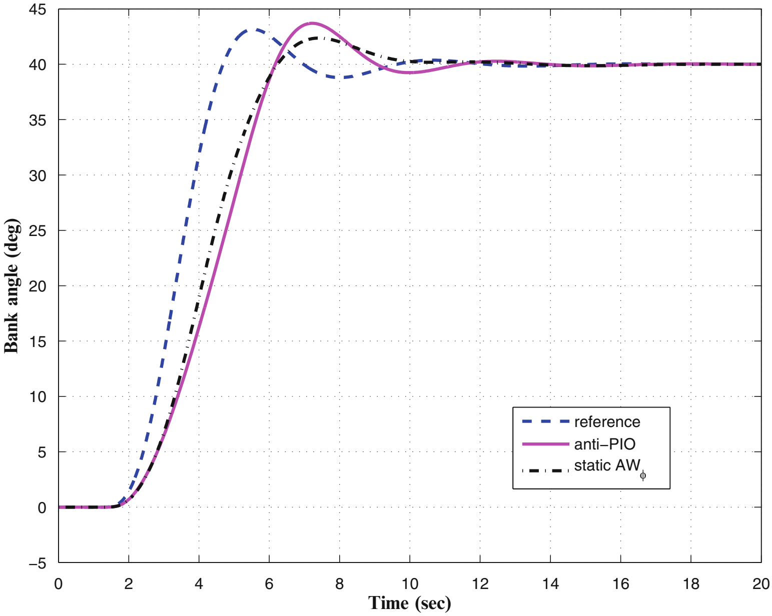


Fig. 10 Roll solicitation of 40°: comparison of the performance outputs for the cases with standard anti-PIO and static anti-windup AW_ϕ

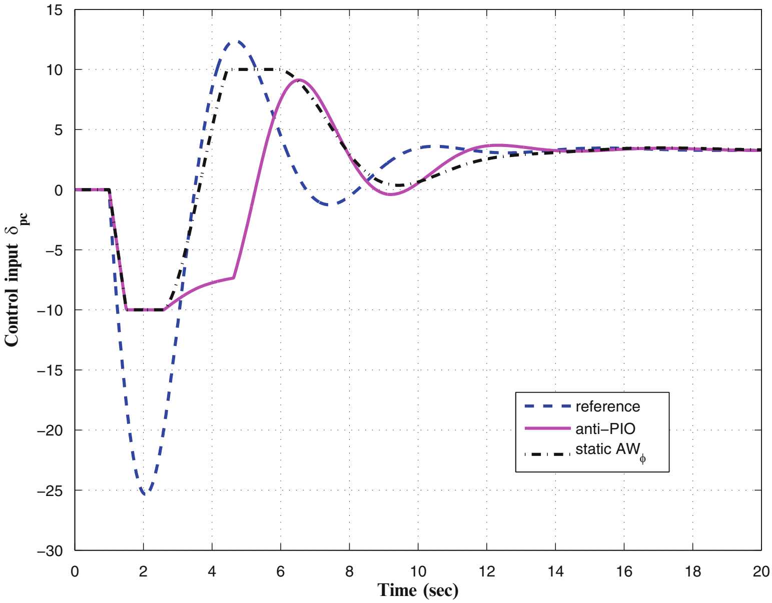


Fig. 11 Roll solicitation of 40°: comparison of the inputs for the cases with standard anti-PIO and static anti-windup AW_ϕ

5 Conclusion

In this chapter, an anti-windup analysis and design strategy has been proposed for systems involving both magnitude and rate saturations, and taking into consideration that such saturations elements only affect some of the inputs. Such a situation has much practical interest for many systems issued in particular from aerospace domain. It is illustrated here on a lateral flying model of a civil aircraft in presence of aggressive maneuvering of the pilot. Actually magnitude and rate saturations of the ailerons deflection actuator may lead to an undesirable behavior which is often called Pilot-Induced-Oscillation (PIO). For this class of nonlinear control systems, anti-windup compensators have been adapted through adequate convex optimization schemes. A comparison with given dynamic anti-PIO filters already used for this class of systems has also been provided. This work lets many questions open, such as the design of other anti-windup schemes. Other classes of fruitful anti-windup compensators may include the parameter-varying approach [24] or reset controllers [28].

Acknowledgement This work was supported by COCKPIT project, convention ONERA F/20 334/DA PPUJ.

References

1. Andrieu, V., Prieur, C., Tarbouriech, S., Arzelier, D.: Global asymptotic stabilization of systems satisfying two different sector conditions. Syst. Control Lett. **60**(8), 570–578 (2011)
2. Anon.: Why the gripen crashed. Aerosp. Am. **32**(2), 11 (1994)
3. Åström, K.J., Rundqwist, L.: Integrator windup and how to avoid it. In: American Control Conference, pp. 1693–1698, Pittsburgh, PA (1989)
4. Biannic, J.-M., Tarbouriech, S.: Analyse et ajustement de lois de commande en présence de saturations implantation de filtres anti-PIO générés par synthèse anti-windup. Technical Report, Rapport COCKPIT/OCKF/CO1.1, 2011
5. Boyd, S., Vandenberghe, S.P.: Convex Optimization. Cambridge University Press, Cambridge (2004)
6. Brieger, O., Kerr, M., Leissling, D., Postlethwaite, I., Sofrony, J., Turner, M.C.: Anti-windup compensation of rate saturation in an experimental aircraft. In: American Control Conference, pp. 924–929, New York (2007)
7. D'Andrea, R., Wyss, M., Waibel, M.: Challenges actuated wingsuit for controlled, self-propelled flight. In: Samad, T., Annaswamy, A.M. (eds.) The Impact of Control Technology, 2nd edn. IEEE-CSS, The Impact of Control Technology (2014)
8. Duda, H.: Prediction of pilot-in-the-loop oscillations due to rate saturation. J. Guid. Control. Dyn. **20**(3), 581–587 (1997)
9. Fertik, H.A., Ross, C.W.: Direct digital control algorithm with anti-windup feature. ISA Trans. **6**, 317–328 (1967)
10. Galeani, S., Onori, S., Teel, A.R., Zaccarian, L.: A magnitude and rate saturation model and its use in the solution of a static anti-windup problem. Syst. Control Lett. **57**(1), 1–9 (2008)
11. Galeani, S., Tarbouriech, S., Turner, M.C., Zaccarian, L.: A tutorial on modern anti-windup design. Eur. J. Control **15**(3–4), 418–440 (2009)

12. Gilbreath, G.P.: Prediction of PIO due to actuator rate limiting using the open-loop onset point (OLOP) criterion. Msc Thesis, Department of Aeronautics and Astronautics, AirForce Institute of Technology, Ohio, USA (2001)
13. Glattfelder, A.H., Schaufelberger, W.: Control Systems with Input and Output Constraints. Springer, London (2003)
14. Grimm, G., Hatfield, J., Postlethwaite, I., Teel, A.R., Turner, M.C., Zaccarian, L.: Anti-windup for stable linear systems with input saturation: an LMI based synthesis. IEEE Trans. Autom. Control **48**(9), 1509–1525 (2003)
15. Hippe, P.: Windup in control. Its effects and their prevention. Advances in Industrial Control. Springer, Germany (2006)
16. Hu, T., Lin, Z.: Control systems with actuator saturation: analysis and design. Birkhäuser, Boston (2001)
17. Kapila, V., Grigoriadis, K. (eds.): Actuator Saturation Control. Marcel Dekker, Inc., New York (2002)
18. Khalil, H.K.: Nonlinear Systems. MacMillan, New York (1992)
19. Klyde, D.H., McRuer, D.T., Myers, T.T.: Pilot-induced oscillation analysis and prediction with actuator rate limiting. J. Guid. Control. Dyn. **20**(1), 81–89 (1997)
20. Klyde, D.H., Richards, N., Cogan, B.: Use of active inceptor cueing to mitigate pilot-vehicle system loss of control. In: AIAA Guidance, Navigation, and Control Conference, Minneapolis, USA, August 2012
21. Liu, Q.: Pilot-induced-oscillation detection and mitigation. Ph.D. Thesis, Cranfield University (2012)
22. Puyou, G., Biannic, J.-M., Boada-Bauxell, J.: Application of robust antiwindup design to the longitudinal aircraft control to cover actuator loss. In: 19th IFAC Symposium on Automatic Control in Aerospace, University of Wurzburg, W ́urzburg, September 2013
23. Queinnec, I., Tarbouriech, S., Garcia, G.: Anti-windup design for aircraft control. In: IEEE Conference on Control Applications (CCA), Munich, Germany, 2006
24. Roos, C., Biannic, J-M., Tarbouriech, S., Prieur, C., Jeanneau, M.: On-ground aircraft control design using a parameter-varying anti-windup approach. Aerosp. Sci. Technol. **14**(7), 459–471 (2010)
25. Rundquist, L., Stahl-Gunnarsson, K.: Phase compensation of rate-limiters in unstable aircraft. In: IEEE Conference on Control Applications, 1996
26. Tarbouriech, S., Turner, M.C.: Anti-windup design: an overview of some recent advances and open problems. IET Control Theory Appl. **3**(1), 1–19 (2009)
27. Tarbouriech, S., Queinnec, I., Turner, M.C.: Anti-windup design with rate and magnitude actuator and sensor saturations. In: European Control Conference, Budapest, Hungary, 2009
28. Tarbouriech, S., Loquen, T., Prieur, C.: Anti-windup strategy for reset control systems. Int. J. Robust Nonlinear Control **21**(10), 1159–1177 (2011)
29. Tarbouriech, S., Garcia, G., Gomes da Silva Jr., J.M., Queinnec, I.: Stability and Stabilization of Linear Systems with Saturating Actuators. Springer, Berlin (2011)
30. Tarbouriech, S., Queinnec, I., Prieur, C.: Stability analysis and stabilization of systems with input backlash. IEEE Trans. Autom. Control **59**(2), 488–494 (2014)
31. Teel, A.R.: Anti-windup for exponentially unstable linear systems. Int. J. Robust Nonlinear Control **9**, 701–716 (1999)
32. Zaccarian, L., Teel, A.R.: Modern Anti-windup Synthesis. Princeton University Press, Princeton (2011)

Modeling and Optimization of Hybrid Transfers to Near Earth Objects

Francesco Topputo and Mauro Massari

Abstract Hybrid propulsion combines chemical propulsion and electric propulsion on the same platform. The brand-new hybrid transfers are thus achieved by sequential combination of high thrust (impulsive maneuvers) and low thrust (continuous arcs). In this chapter, hybrid propulsion transfers are applied to the trajectory optimization for missions to the Near Earth Objects (NEOs). These have been obtained with an optimization based on a direct transcription procedure. The problem is formulated as a nonlinear programming problem and solved for a finite set of variables, which maximize the final spacecraft mass. Effort has been put in modeling the propulsion subsystem. Realistic limitations on both the impulsive maneuvers and low-thrust magnitude have been considered, as well as gravity losses and variation of available electrical power with the distance from the Sun. The transfer to asteroid 162173/1999 JU3 is considered as case study. The designed hybrid propulsion transfers have been compared with purely chemical transfers proposed for the ESA's Marco Polo mission.

Keywords Hybrid transfer • Low-thrust transfer • Near Earth asteroids

1 Introduction

The concept of hybrid propulsion originates from the attempt to combine the features of low-energy transfers and those of low-thrust transfers [8]. Low-Energy transfers are special lunar and interplanetary transfers defined in the frame of the restricted *n*-body problem [3]. These trajectories take advantage of the highly nonlinear dynamics produced by n − 1 attractors to reduce the ΔV cost of a transfer, which in turn reduces the propellant mass. In the Earth–Moon scenario, the low-energy transfers are derived by exploiting the gravitational attractions of the Earth, the Sun, and the Moon. These three forces are modeled as all simultaneously acting upon the spacecraft. A low-energy transfer exploits the natural dynamics of the solar

F. Topputo (✉) • M. Massari
Department of Aerospace Science and Technology, Politecnico di Milano,
Via La Masa 34, 20156 Milano, Italy
e-mail: francesco.topputo@polimi.it; mauro.massari@polimi.it

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization
and Its Applications 114, DOI 10.1007/978-3-319-41508-6_16

system in a more efficient way. Low-energy transfers are achieved with impulsive ΔV's that intrinsically ask for chemical propulsion.

Low-thrust propulsion, which is usually obtained through Solar Electric Propulsion (SEP), represents a viable option to attain further reductions of the propellant mass fraction. However, with low-thrust propulsion only, the transfer time increases excessively. Moreover, using low-thrust propulsion to achieve hyperbolic escape in the framework of interplanetary transfers is not usually deemed a viable option due to the long time required, which is also mostly spent in the harsh radiation environment of the Van Allen belts.

An appealing option to improve the performances of the low-energy transfers without excessively increasing the transfer time is represented by the so-called *low-energy, low-thrust* transfers or hybrid propulsion transfers. This concept has been assessed in a number of works [7, 9, 10]. Hybrid propulsion couples the chemical propulsion escape with a subsequent low-thrust propulsion phase. In this way, both the benefits of low-energy and low-thrust can be exploited, reducing the long time limitation typical of low-thrust propulsion, which are limited in this case to the insertion into the final orbit only. The impact of this aspect on the mission is highly dependent on the particular target orbit required. The case of missions to NEOs is practically not affected by this limitation as the gravity attraction of the target is not sufficiently high to generate a sphere of influence. Therefore, encountering a NEO can be regarded as a rendezvous in heliocentric orbit. This assumption is also true for an eventual return trip from the NEO to the Earth, as the arrival at Earth usually consists in a direct reentry trajectory. For these reasons the adoption of hybrid propulsion transfers for sample return missions to NEOs could potentially allow peculiar performances in terms of mass returned to Earth, and thus consequently of NEO's sample mass.

The case study selected to assess the performances of hybrid propulsion transfers to NEOs is ESA's Marco Polo NEO sample return mission [2]. In Marco Polo, a spacecraft rendezvous with an asteroid, collects some samples, and returns them back to Earth with a direct re-entry trajectory. The most relevant constraints from the mission analysis point of view are the launcher, the launch window, the mission duration, and the hyperbolic Earth arrival velocity. The constraints reported above have been considered in the design of hybrid propulsion transfers for Marco Polo mission, using the same values considered in the preliminary mission design. To mimic the reference Marco Polo mission, hybrid propulsion transfers to the asteroid 162173/1999 JU3 have been searched in the years 2018 (baseline) and 2019 (back-up). Moreover, for each of the 2 years window, the possibility of exploiting a lunar gravity assist has been assessed.

2 Definition of Hybrid Propulsion Transfers

A hybrid propulsion transfer can be briefly described by the following sequence of events. First, the spacecraft is launched and placed into a low-Earth parking orbit. Then, an impulsive maneuver injects the spacecraft into an Earth-escape orbit toward the Sun–Earth L_1, L_2 Lagrange points region. The Earth-escape maneuver is accomplished by using chemical propulsion. From this point on the spacecraft can only rely on its low-thrust propulsion to reach the final target.

Few observations can be made from the definition of hybrid propulsion transfer. First of all, a lunar gravity assist (performed immediately after the chemical burn) can be considered. This option would improve the overall transfers performances [5–7] although the inclusion of a lunar swing-by is not straightforward and depends on the mission at hand. For instance, it can be planned only for missions having a dedicated launch. In the case of exploitation of a lunar gravity assist, it would be desirable to account for a small trajectory correction maneuver (TCM) between the initial impulsive injection and the Moon encounter, that can be performed using both chemical propulsion or SEP. When no lunar swingbys are envisaged, possible errors in the nominal trajectory (due to off-nominal impulsive injection) can be recovered in the subsequent low-thrust arc.

As the entity of the initial chemical injection maneuver can be considerably high, it is advisable to split it into several (e.g., two or three) lower ΔV maneuvers. This allows reducing considerably the gravity losses associated with the initial maneuver. Although this increases the total flight time, splitting the initial injection maneuver may allow for trajectory corrections and full spacecraft commissioning before the escape.

The chemical propulsion is used only to achieve the injection into escape trajectory. Enabling the use of chemical propulsion along the transfer is not convenient; the main reasons for this are related to the fact that the chemical propulsion is strictly necessary only in the first phase of the mission. After achieving escape, all the masses associated to the chemical propulsion subsystem are not needed anymore and could be ejected. In this way the low-thrust propulsion would be more effective on a higher thrust-to-mass ratio. Keeping the chemical propulsion subsystem on-board means having a less complex system design (e.g., no separation of propulsion stage is required).

With reference to Fig. 1, two different options can be considered for the system design: a single stage spacecraft or a dual stage spacecraft. In a single stage spacecraft, the two propulsion subsystems, as well as the main platform, constitute a single system. Although chemical propulsion is used in the first part of the transfer only, the spacecraft carries on all the masses associated to it (thruster, tanks, residual propellant, feeding lines, etc.). In principle, this solution is inefficient (the chemical propulsion is no longer used after the Earth departure and the possible TCM), while it eases the design, integration, and operations. The dual stage spacecraft is made of a chemical propulsion module (CPM) and by the main platform equipped with SEP. After having executed the Earth departure and the possible TCM, the CPM is jettisoned from the main spacecraft, which continues its mission by relying on SEP only. This solution is deemed efficient, as the thrust-to-mass ratio of the SEP arcs is higher than that of a single stage spacecraft. Moreover, in this case the existing technology of dual stage spacecraft can be reused. This is the case, for instance, of the propulsion modules of LISA Pathfinder [12]. This would reduce the costs and complexity of a dual stage choice.

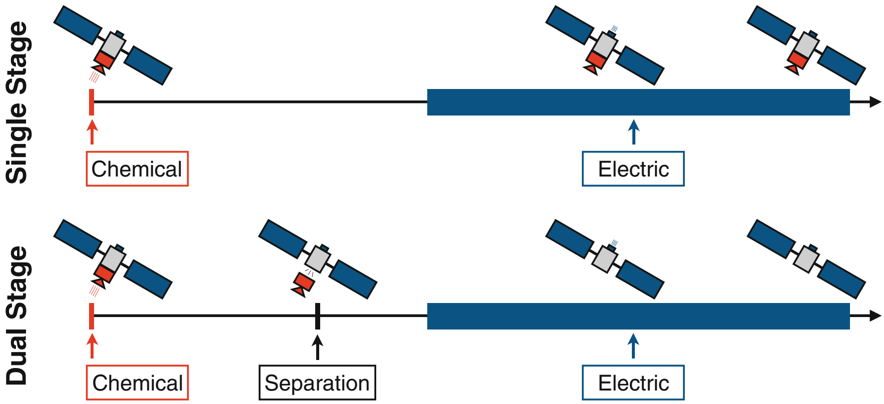


Fig. 1 Hybrid propulsion transfers for single and dual staged spacecraft

In the rest of the chapter the concept of a dual stage spacecraft is considered. For the mid-sized class of interplanetary missions to the NEOs, the potential benefits may overcome the increased complexity, especially considering that the design of the chemical propulsion stage could be easily shared with other missions directed to different targets.

3 Modeling Hybrid Propulsion Transfers

3.1 Chemical Propulsion

Chemical propulsion is modeled as producing instantaneous velocity changes. For the sake of evaluating the preliminary chemical propellant mass budget, the rocket equation is used:

$$m_{p_{ch}} = m_0 \left(1 - e^{-\frac{\Delta V}{I_{sp_{ch}} g_0}}\right)(1 + g_l) \tag{1}$$

where $m_{p_{ch}}$ is the chemical propellant mass, m_0 is the wet mass delivered in GTO by a Soyuz launcher (see Sect. 4.2), ΔV is the magnitude of the initial impulse, $I_{sp_{ch}}$ is the chemical engine specific impulse (assumed 316 s), g_0 is the gravitational acceleration at sea level, and g_l is the gravity loss factor. A value of gravity loss equal to 5 % has been assumed, which is compatible with a three-burn strategy for the initial impulse.

It is worth observing that in case of dual stage spacecraft (CPM jettisoned after the impulsive maneuver), the CPM mass has to be subtracted from m_0 in order to infer the initial mass for the SEP trajectory. As sizing the CPM is behind the scope of this work, it has been assumed that the dry mass of the CPM, $m_{dry_{CPM}}$, is a fixed fraction of the spacecraft dry mass, $m_{dry_{SC}}$. A mass fraction of 15 % is taken. Therefore, since $m_{dry_{SC}} = 1600\,\text{kg}$, we have

$$m_{dry_{CPM}} = 0.15\, m_{dry_{SC}} = 240\,\text{kg}. \tag{2}$$

The assumption in Eq. (2) is compatible with the statistical data regarding propulsion subsystem for interplanetary missions, and is in accordance with the available data of missions implementing a separated propulsion unit.

3.2 Electric Propulsion

Solar electric propulsion is modeled as a low-thrust acceleration. The propellant mass spent in the thrust arcs is obtained by integrating the following equation:

$$\dot{m} = -\frac{|\mathbf{T}|}{I_{sp_{lt}} g_0} \tag{3}$$

where $\dot{m}$ is the instantaneous mass-flow rate, $|T|$ is the instantaneous thrust magnitude, $I_{sp_{lt}}$ is specific impulse of the low-thrust engine. Depending on the technology implemented in the low-thrust engine, the figures of solar electric propulsion may vary sensitively.

The Snecma PPS 5000 Hall effect thruster has been considered for the low-thrust phase. The nominal values of thrust and specific impulse for this thruster have been considered in the trajectory optimization. These values, reported in Table 1, have been extrapolated from [4] by considering the performances at 5.5 kW.

In the optimization two different cases have been considered, with the Electrical Power Subsystem (EPS) sized to provide the required 5.5 kW at NEO's maximum distance from the Sun and at 1 AU. The first case is simple, because it is equivalent to considering the maximum thrust fixed for the entire trajectory, making the assumption that the necessary 5.5 kW power is always available. In the second case, the maximum thrust and I_{sp} are function of the available power, which is dependent on the actual distance from the Sun. In this work, it has been considered that the I_{sp} remains always constant and that the maximum thrust is proportional to the available power, which depends on the distance from the Sun as

$$T_{max} = T_{max_{1\,AU}} r^{-1.8} \tag{4}$$

Table 1 Snecma PPS 5000 figures

Element	Value	Unit
Maximum I_{sp}	1769	s
Maximum thrust at 5.5 kW	307	mN

where r is the distance from the Sun. The exponent 1.8 is considered in place of the classic 2 to account for better performances of the solar arrays associated to lower temperatures, which are achieved when moving far from the Sun.

3.3 Trajectory Optimization

The equations of motion used to model the orbital dynamics are

$$\ddot{\mathbf{r}} + \frac{\mu}{r^3}\mathbf{r} = \frac{\mathbf{T}}{m} \tag{5}$$

where r is the distance from the Sun and μ is the Sun gravitational parameter. The guidance law, $\mathbf{T}(t)$, is the unknown in the trajectory optimization. This has to be determined such that the boundary conditions are met, as well as the technological limitations, typically written in the form $|\mathbf{T}| - T_{max} \leq 0$.

In this work, a direct transcription procedure has been used to optimize the hybrid transfers. The dynamics in Eq. (5) is discretized over a uniform time grid and the constraints deriving from the equations of motion are enforced within each of the time intervals. The problem is formulated as a nonlinear programming problem using a Pseudo Spectral Method and solved for a finite set of variables. The optimization maximizes the final spacecraft mass while respecting the trajectory constraints. Details on the method used can be found in [5].

4 Marco Polo Reference Mission

To assess the usefulness and efficiency of hybrid propulsion transfers to NEOs, the Marco Polo mission has been taken as benchmark case. This is a NEO sample return mission. It is important to highlight that the work presented in this chapter considers the available data in the reference documentation of Marco Polo for what concerns the details of the mission and for comparing it with conventional propulsion [11]. It is deemed that the updates on Marco Polo mission (new asteroid, new launch window, use of low-thrust, etc.) do not add significant information. The data relevant from the Marco Polo mission analysis are reported in Table 2.

The most relevant constraints from the mission analysis point of view are the launcher capability, the launch window, the mission duration and the hyperbolic Earth arrival velocity. The figures in Table 2 have been considered for the design of a hybrid propulsion transfer for a NEOs sample return mission. The boundary conditions considered are represented by the initial orbit and by the target orbit, which are briefly described in the following.

Table 2 Marco Polo mission summary (taken from [11])

Launch	Direct escape using Soyuz Launcher
Target	Asteroid 162173/1999 JU3
Propulsion	Chemical propulsion
Launch window	2018 baseline, 2019 back-up
Mission duration	Less than 8 years
Near asteroid phase	Longer than 8 months
Earth arrival velocity	Less than 6 km/s

Table 3 Orbital elements of asteroid 162173/1999 JU3

Element	Value	Unit
Semi-major axis	1.18953379	AU
Eccentricity	0.19025925	–
Inclination	5.88404421	deg
RAAN	251.61712004	deg
Perihelion anomaly	211.42300069	deg
Mean anomaly at epoch	226.57102589	deg
Epoch	4655.5	MJD2000

4.1 Asteroid 162173/1999 JU3 Orbit

The asteroid 162173/1999 JU3 is an Apollo-type, Earth crossing asteroid. The orbital parameters of asteroid 162173/1999 JU3 are given in Table 3.

With these data, the perihelion and aphelion are 0.9632 and 1.4158 AU, respectively, so indicating that the asteroid may get close to the Earth and Mars orbits. This encounter does not occur in the time-frame of interest, and therefore the orbit considered is that specified by the above mentioned orbital parameters. The orbits of the Earth, Mars, and asteroid 162173/1999 JU3 are shown in Fig. 2.

4.2 Soyuz-Fregat GTO Orbit

For the hybrid transfer to asteroid 162173/1999 JU3, a dedicated launch from Kourou with Soyuz 2.1b and the Fregat upper stage has been considered. The orbital parameters of the reference GTO are summarized in Table 4.

The altitude of perigee and apogee is fixed, as well as the inclination and argument of perigee. The right ascension of ascending node (RAAN) is calculated as a function of the departure epoch (t_0) through

$$\Omega(t_0) = \Omega_{CM}(t_0) + 182^\circ \tag{6}$$

where $\Omega_{CM}(t_0)$ is the argument of the central meridian (Greenwich meridian) at the departure epoch]. As the dedicated launch strategy foresees a launch at any desired

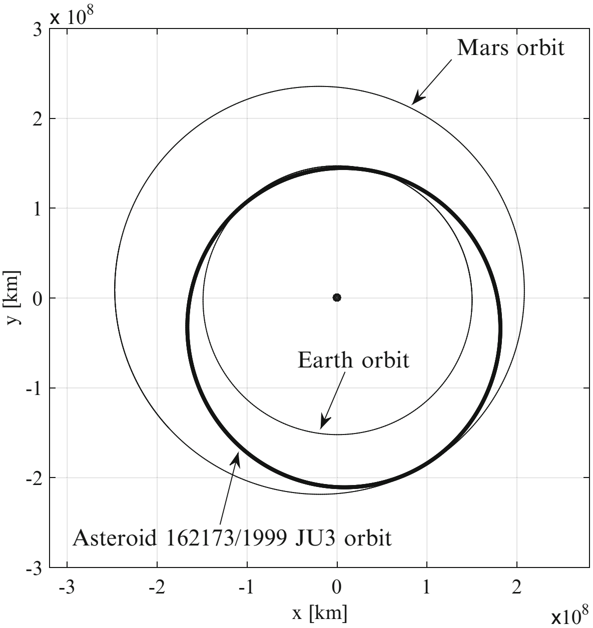


Fig. 2 The orbits of the Earth, Mars, and asteroid 162173/1999 JU3

Table 4 Orbital parameters of the Soyuz 2.1b Fregat GTO [1]

Element	Value	Unit
Perigee altitude	250	km
Apogee altitude	35,950	km
Inclination	6	deg
Argument of perigee	178	deg

moment (within the year and within the day), all values of RAAN can be achieved by properly selecting t_0.

5 Hybrid Propulsion Options for Marco Polo

The hybrid propulsion transfers for the sample return mission to NEOs that have been designed are all structured with the qualitative sequence of events reported in Fig. 3 and listed in Table 5.

The two gravity assists at the Moon and Earth are optional; i.e., they are not strictly needed to accomplish the transfer. In addition, the Earth return is foreseen with a direct re-entry with a limit on the hyperbolic excess velocity, like in the Marco Polo mission. The aim of the hybrid transfers is maximizing the final mass at asteroid arrival, which in turn maximizes the final mass returned to Earth. This is accomplished by minimizing both the total ΔV of the impulsive maneuvers as well as the propellant mass spent in the low-thrust arcs.

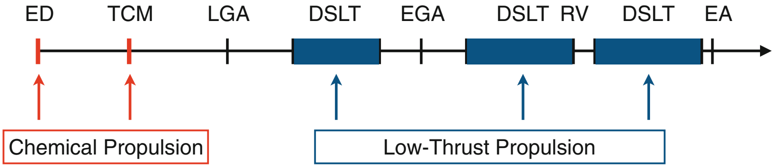


Fig. 3 Sequence of events for hybrid transfer to asteroid 162173/1999 JU3

Table 5 Sequence of events for hybrid transfer to asteroid 162173/1999 JU3

Event	Description
ED	Earth departure
TCM	Trajectory correction maneuver (optional)
LGA	Lunar gravity assist (optional)
DSLT	Deep space low-thrust
EGA	Earth gravity assist (optional)
RV	NEO rendez-vous
EA	Earth arrival

To mimic the reference Marco Polo mission, hybrid propulsion transfers to the asteroid 162173/1999 JU3 have been searched in the years 2018 (baseline) and 2019 (back-up). Moreover, for each of the two one-year windows, the possibility of exploiting a lunar gravity assist has been assessed. Marco Polo mission foresees the possibility of a direct injection into escape orbit, however, the concepts of hybrid propulsion transfer is based on the assumption of departure from LEO or GTO. Therefore, in order to fairly compare Marco Polo conventional propulsion transfer with the hybrid propulsion transfer, it has been decided to select as departure orbit the GTO orbit of the Soyuz 2.1b launched from Kourou and achieved with the Fregat upper stage (i.e., using the same launcher configuration used for Marco Polo).

5.1 The Escape Portion

The initial impulsive maneuver is needed to achieve escape from the Earth. This can be of two types, ballistic [13] or hyperbolic. Although ballistic escape may be more efficient than the classical hyperbolic escape, the latter has been considered in this study. However, the hyperbolic escape orbit is not achieved using only chemical propulsion and considering the analytic solution of the two-body problem. The full four-body model is considered for the dynamics, and the SEP is used in conjunction with the chemical propulsion to achieve the escape, exploiting in this way both the gravitational attractions of the Moon and the Sun, and the low-thrust propulsion. In this way, the solution obtained is more flexible and faster than a purely ballistic escape, but it is less costly than a purely hyperbolic escape.

5.2 The Low-Thrust Portion

Solar electric propulsion is used to rendezvous with the target asteroid. The SEP is switched on after the chemical maneuver, or in the case it is foreseen, after the lunar flyby, and it is used in the deep space to acquire the proper conditions at arrival. In this phase, the dynamics are those of a controlled two-body problem, where only the gravitational attraction of the Sun is considered.

The return leg is treated likewise, with the only difference that the arrival condition is constrained by a limit on the hyperbolic excess velocity, while the initial condition is considered to be equal to the asteroid state.

In general, the duration and the mass spent in the low-thrust phases depend on the type and the number of thrusters, on the thrusters I_{sp}, on the thrust magnitude and on the nominal guidance law. The low-thrust phases are designed such that the propellant mass is minimized, regardless of the transfer time, as in the considered reference mission only constraints on the maximum mission duration are foreseen.

6 Results

Hybrid transfers to the asteroid 162173/1999 JU3 have been searched with departure in the years 2018 (baseline) and 2019 (back-up). Moreover, the possibility of exploiting a lunar gravity assist has been assessed. Executing one or multiple Earth gravity assists has been deemed not useful in the hybrid propulsion transfer scenario as the 5.9° inclination change can be achieved efficiently with SEP, without constraining the trajectory to perform Earth encounters. All the cases analyzed have been optimized considering the EPS sized for maximum power both at the target NEO distance and at Earth distance (1 AU).

The return leg of the trajectory has been designed fixing a departure date for the return trip. This is common to all the cases considered. It allows the easing the trajectory design without jeopardizing the feasibility of the mission, as both the minimum stay at the asteroid and the maximum mission duration are guaranteed by the chosen date. In this way the return trip is optimized in one case, and obtained by continuation for the other cases, by just changing the initial mass. In summary, the eight cases summarized in Table 6 have been analyzed.

Table 6 Summary of solutions analyzed

Label	Year	Launch	EPS sizing
H18DN	2018	Direct transfer	@ NEO
H18LN	2018	Lunar gravity assist	@ NEO
H19DN	2019	Direct transfer	@ NEO
H19LN	2019	Lunar gravity assist	@ NEO
H18DE	2018	Direct transfer	@ 1 AU
H18LE	2018	Lunar gravity assist	@ 1 AU
H19DE	2019	Direct transfer	@ 1 AU
H19LE	2019	Lunar gravity assist	@ 1 AU

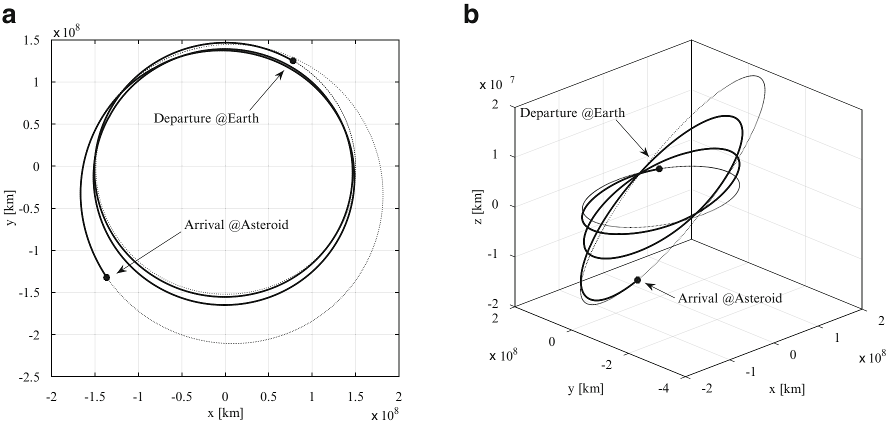


Fig. 4 Solution H18DN. (**a**) Ecliptic plane. (**b**) Out of plane

The details of the designed hybrid propulsion transfers are reported in Table 7. In Fig. 4a, b the heliocentric phase of H18DN is reported. From the figure it is possible to see that the change of inclination is performed gradually with the SEP during the heliocentric phase and that the need of multiple revolutions is essentially due to that.

In Fig. 5a, b both the heliocentric and geocentric phases of solution H18LN are reported. This solution performs a lunar gravity assist. However, the final mass does not improve at all. This is mainly due to the fact that a lunar gravity assist requires that the spacecraft encounter with the Moon happens at one of the Moon orbital nodes. This is required to avoid the necessity of a plane change maneuver between the equatorial plane of the GTO and the Moon orbital plane in the initial phase. This imposes phasing constraints to the geocentric trajectory which are in contrast with the constraints required by the subsequent heliocentric rendez-vous.

Figure 6a, b shows the heliocentric phases of solution H19DN. Both the transfer from Earth to the NEO and the return to Earth are reported. The heliocentric phase of solution H19LN is very similar to that of solution H19DN, so it has not been shown. It is worth to recall that the return trajectories of all the solutions are very similar as the departure date is the same and are all obtained by continuation of the Solution H18DN.

The return transfer requires considerably less propellant than the initial transfer to reach the NEO. This is mainly due to the fact that the constraints on the arrival velocity at Earth does not require neither a rendezvous nor that the spacecraft change the inclination to be on the orbital plane of the Earth, thus reducing considerably the amount of propellant needed.

The solutions for the cases considering the sizing of the EPS at 1 AU have been obtained by continuation, starting for the corresponding solution with EPS sized at the NEO. The final trajectories are almost identical to the cases already

Table 7 Details of solutions found

	Unit	H18DN	H18LN	H19DN	H19LN	H18DE	H18LE	H19DE	H19LE
Masses									
Initial wet mass	kg	3070	3070	3070	3070	3070	3070	3070	3070
Chemical propellant mass	kg	619	599	626	604	619	599	625	604
CPM wet mass	kg	859	839	866	844	859	839	865	844
Xenon mass to asteroid	kg	475	500	471	510	472	494	470	517
Xenon mass for return	kg	43	43	42	42	43	43	42	42
S/C mass at asteroid	kg	1736	1731	1733	1716	1739	1737	1735	1709
Earth return mass	kg	1693	1688	1691	1674	1696	1694	1693	1667
Orbits									
Initial orbit		GTO	GTO	GTO	GTO	GTO	GTO	GTO	GTO
Moon pericenter altitude	km	–	100	–	1855	–	100	–	1874
Final orbit		Re-entry	Re-entry	Re-entry	Re-entry	Re-entry	Re-entry	Re-entry	Re-entry
Times									
Departure	year	2018	2018	2019	2019	2018	2018	2019	2019
Departure	date	6 Nov	25 Oct	19 Apr	7 Apr	6 Nov	25 Oct	19 Apr	7 Apr
Earth–asteroid	days	923	825	713	728	923	825	713	728
Near asteroid phase	days	493	603	539	536	493	603	540	536
Asteroid–Earth	days	803	804	803	804	803	804	802	804
Total mission duration	years	6.08	6.11	5.63	5.66	6.08	6.11	5.63	5.66
Velocities									
Initial ΔV	m/s	697.8	673.0	706.9	679.0	697.6	672.9	705.7	678.8
Moon incoming velocity	km/s	–	1.02	–	1.06	–	1.08	–	1.06
Earth arrival velocity	km/s	4.82	4.89	4.83	4.88	4.85	4.89	4.83	4.88

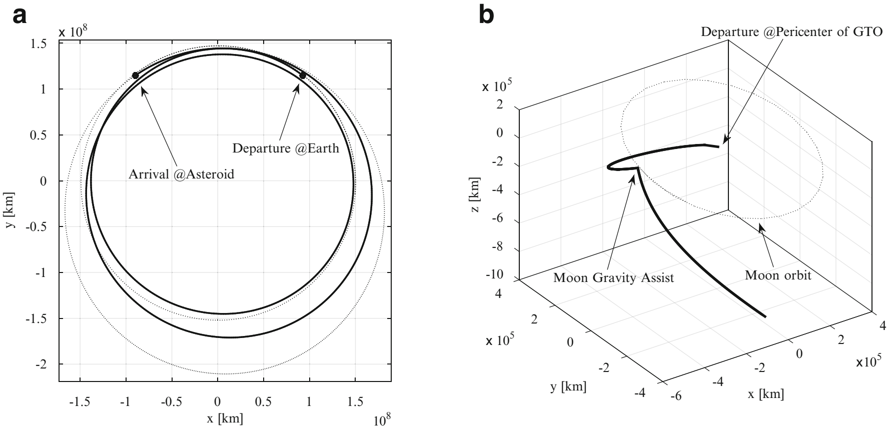


Fig. 5 Solution H18LN. (**a**) Heliocentric phase. (**b**) Geocentric phase

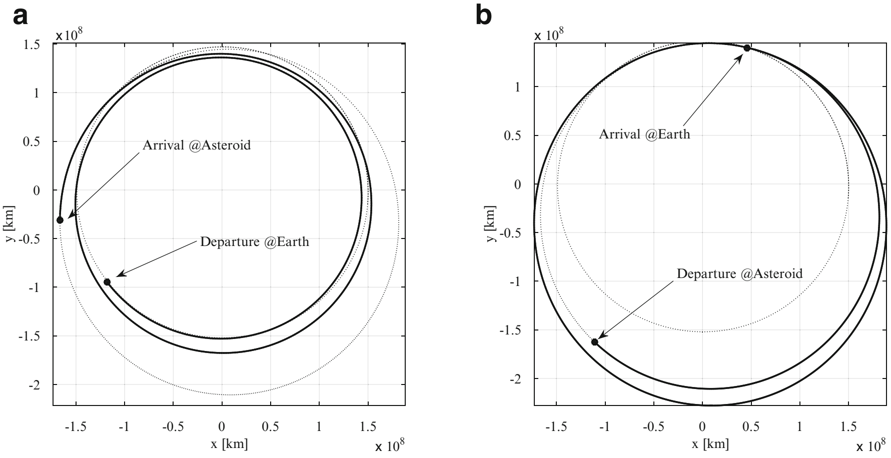


Fig. 6 Solution H19DN. (**a**) Earth to asteroid phase. (**b**) Asteroid to Earth phase

shown. Also the performances obtained in terms of final mass delivered to Earth are similar. This result was unexpected, as it seem reasonable that the reduced available maximum thrust should have affected more the trajectories. However, this can be easily explained, looking at the thrust profile obtained for solutions H18DN shown in Fig. 7. It is clearly visible that the thrust profile has a typical bang-bang structure, as it could have been expected maximizing the final mass. The interesting issue is that all the obtained solutions take advantage of the higher efficiency of the thrust at lower distance from the Sun, thus minimizing the effect of the maximum thrust reduction due to the more stringent EPS sizing.

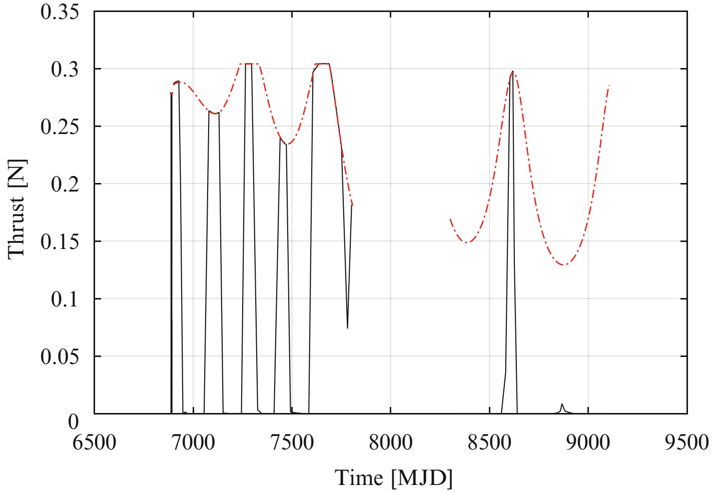


Fig. 7 Thrust profile for solution H18DE

7 In-depth Comparison and Discussion

The reference solutions for assessing hybrid transfers to NEOs are taken from Marco Polo reference documentation [11]. In these study a direct injection into an escape hyperbola has been considered. It is straightforward that the direct injection option with Soyuz launch involves a much lower mass delivered by the launcher with respect to the injection in GTO. This amounts roughly about 1650 kg, and depends on the hyperbolic escape velocity and declination of the escape hyperbola. It has been considered that even in the case of departure from GTO, for a purely chemical mission, the escape mass will not be too much different, as the required escape velocity should be reached in any case. All reference solutions consider chemical propulsion and a single stage spacecraft.

7.1 Marco Polo Mission: Direct Escape with Soyuz

In Marco Polo reference documentation two solutions are presented, the first covers a launch in 2018 (baseline), the second in 2019 (back-up). The details are reported in Table 8. The baseline solution has two Earth swing-by's and four deep-space maneuvers (DSM); the back-up solution foresees instead one Earth swing-by and three DSM. In all cases, the total transfer budget is 1394 m/s, roughly split in 850 m/s for the transfer to the asteroid, and 550 m/s for the Earth return. One of the most important figures is the maximum available mass at Earth return, whose value amounts to 929 kg in the baseline solution and 897 kg in the back-up.

Table 8 Details of Marco Polo reference solutions (BL: baseline, BK: backup) [11]

Solution	BL	BK
Departure epoch	Dec 2018	Dec 2019
Earth flyby 1 epoch	Dec 2019	Dec 2020
Earth flyby 2 epoch	Dec 2020	–
Asteroid encounter	Feb 2022	Feb 2022
Asteroid departure	Jul 2023	Jul 2023
Earth re-entry	Dec 2024	Dec 2024
Total mission duration	6 years	5 years
Hyperbolic escape velocity	3.2 km/s	3.2 km/s
Total ΔV	1394 m/s	1394 m/s
Earth escape mass	1661 kg	1629 kg
Earth return mass	929 kg	897 kg

7.2 Critical Comparison

A preliminary analysis of the hybrid solutions found and their consequences at system level can be performed by comparing the results above. On the one side, hybrid propulsion transfers have some drawbacks with respect to conventional propulsion: they require longer transfer times and higher power levels (i.e., larger solar arrays); they also involve an increased complexity of both the system and operations. However, on the other side, the potential savings in Earth return mass are dramatic. Moreover, if compared with a purely SEP transfer, a shorter durations are required, especially during the escape phase, so allowing a reduction of the cumulated radiation dose.

In Table 9, the most relevant figures of the hybrid propulsion transfers found (taken from Table 7) and those of the reference Marco Polo mission (taken from Table 8) are reported. In particular, the focus in on (1) the Earth escape mass (EM), which is the mass placed in hyperbolic Earth escape orbit, (2) the mass at asteroid (MA), which is the mass at asteroid arrival, (3) the Earth return mass (ERM), which is the mass at Earth re-entry, (4) the time to asteroid encounter (TAE), which is the time for the Earth–asteroid transfer, and (5) the Asteroid–Earth transfer time (AET), which is the time for the Earth return.

The hybrid solutions are able to place a higher mass in Earth escape orbit. This is due to the lower values of the impulse required. As for the mass at asteroid rendezvous, the hybrid solutions outperform the direct escape solutions (BL and BK). The best performances are obtained in terms of Earth return mass. Here the hybrid propulsion transfers assure an atmospheric entry mass that is approximately 600 kg higher than the direct escape case. This is an important result.

The same conclusions can be drawn by looking at Fig. 8, which shows the mass breakdown of the different solutions, where the ERM is summed up to the mass needed to escape, to reach the asteroid, and to come back to Earth. The reference mission solutions are characterized by a different launch option (direct escape) with respect to the hybrid propulsion transfers (GTO). Therefore, to make a fair

Table 9 Comparison of solutions (EM, MA, and ERM are in kg; TAE and AET are in days)

	H18DN	H18LN	H19DN	H19LN	H18DE	H18LE	H19DE	H19LE	BL	BK
EM	2211	2231	2204	2226	2211	2231	2205	2226	1661	1629
MA	1736	1731	1733	1716	1739	1737	1735	1709	1209	1177
ERM	1693	1688	1691	1674	1696	1694	1693	1667	929	897
TAE	923	825	713	728	923	825	713	728	1154	789
AET	803	804	803	804	803	804	802	804	518	518

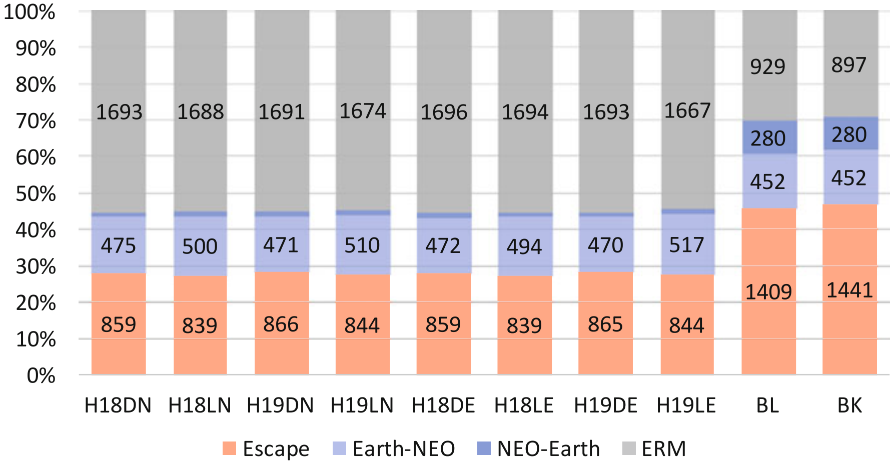


Fig. 8 Solutions mass breakdown (masses in kg)

comparison, it has been assumed that the same escape trajectory and mass can be obtained considering a pure chemical spacecraft starting from a GTO and thus with an initial mass of 3070 kg.

7.3 Discussion

Earth–NEO hybrid propulsion transfers from GTO have been computed in this study. Although these solutions can be re-computed and better refined considering a more accurate model of both the dynamics and the thruster, some observations can be done by analyzing the results in Table 9. In general, the hybrid propulsion transfers outperform the chemical propulsion transfers. The price to pay for higher transfer efficiency is the increased transfer time. In this case, this is mitigated by the fact that no Earth flyby are required in the hybrid propulsion transfers. All hybrid transfers for Marco Polo behave better than the reference solutions in terms of Earth return mass. Moreover, since lunar gravity assists do not improve the hybrid solutions, they can be neglected, thus reducing the complexity of the hybrid approach.

8 Conclusion

In this chapter, the concept of hybrid propulsion transfers for sample return mission to the NEOs has been presented. These transfers foresee the presence of both high-thrust chemical propulsion and low-thrust solar electric propulsion on the same platform. The hybrid propulsion transfers have been applied to the case of Marco Polo sample return mission. It has been demonstrated that the hybrid propulsion transfers are more convenient with respect to the reference conventional propulsion in terms of Earth return mass. The implications of the hybrid propulsion transfers on the system design have been briefly discussed, and even if the potential benefits have been shown, it is hardly possible to clearly state that the hybrid propulsion concept is better than conventional propulsion. However, the results shown indicate that this technology can enable considerable benefits for certain kind of mission, such as the missions to the NEOs.

Acknowledgements The work described in this chapter is inspired by the ESA ITT AO/1-6791/11/NL/AF *Hybrid propulsion transfer strategies*, which was executed by Dinamica Srl under Contract No. 4000105465 in 2012. The authors are grateful to Torsten Bieler, Michael Khan, Richard Long, and Giorgio Mingotti for their valuable support.

References

1. Arianespace: Soyuz User's Manual (Issue 2 - Revision 0) (2012) http://www.arianespace.com/wp-content/uploads/2015/09/Soyuz-Users-Manual-March-2012.pdf
2. Barucci, M., Yoshikawa, M., Michel, P., Kawagushi, J., Yano, H., Brucato, J., Franchi, I., Dotto, E., Fulchignoni, M., Ulamec, S.: Marco Polo: near earth object sample return mission. Exp. Astron. **23**, 785–808 (2009). doi:10.1007/s10686-008-9087-8
3. Belbruno, E., Miller, J.: Sun-perturbed Earth-to-moon transfers with ballistic capture. J. Guid. Control. Dyn. **16**, 770–775 (1993)
4. Duchemin, O., Dumazert, P., Clark, S., Mundy, D.: Development and testing of a high-power hall thruster. In: 28th International Electric Propulsion Conference, Toulouse, 17–21 March 2003
5. Massari, M., Bernelli-Zazzera, F.: Options for optimal trajectory design of a mission to NEOs using low-thrust propulsion. Adv. Astronaut. Sci. **119**, 541–552 (2005)
6. Massari, M., Bernelli-Zazzera, F., Vasile, M.: Trajectory optimization for a mission to NEOs, using low-thrust propulsion and gravity assist. Adv. Astronaut. Sci. **114**, 317–329 (2003)
7. Mingotti, G., Topputo, F.: Ways to the moon: a survey. Adv. Astronaut. Sci. **140**, 2531–2548 (2011)
8. Mingotti, G., Topputo, F., Bernelli-Zazzera, F.: Low-energy, low-thrust transfers to the moon. Celest. Mech. Dyn. Astron. **105**, 61–74 (2009). doi:10.1007/s10569-009-9220-7
9. Mingotti, G., Topputo, F., Bernelli-Zazzera, F.: Numerical methods to design low-energy, low-thrust sun-perturbed transfers to the moon. In: Proceedings of the 49th Israel Annual Conference on Aerospace Sciences, Tel Aviv – Haifa, 2009
10. Mingotti, G., Topputo, F., Bernelli-Zazzera, F.: Efficient invariant-manifold, low-thrust planar trajectories to the moon. Commun. Nonlinear Sci. Numer. Simul. **17**, 817–831 (2012). doi:10.1016/j.cnsns.2011.06.033
11. Marco Polo assessment study report (SRE-2009-3). Technical Report (2009). http://sci.esa.int/jump.cfm?oid=46019
12. Racca, G., McNamara, P.: The LISA pathfinder mission. Space Sci. Rev. **151**, 159–181 (2010). doi:10.1007/s11214-009-9602-x
13. Topputo, F., Belbruno, E., Gidea, M.: Resonant motion, ballistic escape, and their applications in astrodynamics. Adv. Space Res. **42**, 6–17 (2008). doi:10.1016/j.asr.2008.01.017

Probabilistic Safety Analysis of the Collision Between a Space Debris and a Satellite with an Island Particle Algorithm

Christelle Vergé, Jérôme Morio, Pierre Del Moral, and Juan Carlos Dolado Pérez

Abstract Collision between satellites and space debris seldom happens, but the loss of a satellite by collision may have catastrophic consequences both for the satellite mission and for the space environment. To support the decision to trigger off a collision avoidance manoeuver, an adapted tool is the determination of the collision probability between debris and satellite. This probability estimation can be performed with rare event simulation techniques when Monte Carlo techniques are not enough accurate. In this chapter, we focus on analyzing the influence of different simulation parameters (such as the drag coefficient) that are set for to simplify the simulation, on the collision probability estimation. A bad estimation of these simulation parameters can strongly modify rare event probability estimations. We design here a new island particle Markov chain Monte Carlo algorithm to determine the parameters that, in case of bad estimation, tend to increase the collision probability value. This algorithm also gives an estimate of the collision probability maximum taking into account the likelihood of the parameters. The principles of this statistical technique are described throughout this chapter.

Keywords Rare event • Sequential Monte Carlo • Island particle models • Debris • Satellite • Collision • Adaptive splitting technique

C. Vergé
ONERA - The French Aerospace Lab, F-91761 Palaiseau, France

CNES, 18 avenue Edouard Belin, 31401 Toulouse, France

Centre de Mathématiques Appliquées, Route de Saclay, 91128 Palaiseau, France

J. Morio (✉)
DCPS/UFTMIP/ONERA, ONERA - The French Aerospace Lab, F-31055, Toulouse, France
e-mail: jerome.morio@onera.fr

P. Del Moral
School of Mathematics and Statistics, University of New South Wales, Kensington, Sydney, NSW 2052, Australia

J.C.D. Pérez
CNES, 18 avenue Edouard Belin, 31401 Toulouse, France

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization and Its Applications 114, DOI 10.1007/978-3-319-41508-6_17

1 Introduction

On February 10th 2009, active commercial satellite Iridium-33 and out of order Russian satellite Cosmos-2251 collided [6]. The impact produced more than 2000 trackable debris. Most of them may destroy any satellite, whether in use or not, they might encounter. The safest practice for satellites that encounter space debris is to avoid collision. Avoidance maneuvers are an efficient mean to reduce the collision probability between two orbiting objects, nevertheless they consume fuel reducing the operational lifetime of the satellite and they perturb the operational mission of the satellite. Consequently, satellite safety responsible teams have to take into account the operational mission prior to the definition of a collision avoidance maneuver and try to combine, whenever possible, planned station keeping maneuvers with collision avoidance maneuvers. Avoidance maneuvers are decided, among other parameters, based on the estimated collision probability.

The orbital motion of the space objects is simulated using a simplified deterministic dynamical model that may be considered as an input-output function where the random inputs are, for instance, the position and the speed of the debris and of the satellite as well as other dynamic parameters, and the output is the minimum distance between the debris and the satellite. The collision probability is then estimated on this output. This input-output function can be seen as a "black-box" with random inputs. Some parameters, denoted by a vector Θ, in black-box functions are implicitly set, such as parameters of the model (the drag coefficient for instance) or of the input parametric model density, and their value influences the collision probability estimation. These hypotheses are often assumed for simplification and computational reasons. From a risk analysis point of view, it is interesting to determine the variability of the collision probability with respect to (w.r.t.) the uncertainty on theses input parameters Θ, and to quantify the impact of such tuning in the realization of a collision. Of course, different values of Θ can strongly modify rare event probability estimation and sometimes miss a risk situation. The issue of concern in safety would be to underestimate a risk because of a bad tuning of model parameters Θ. That is why in this paper we propose to estimate the law of the parameters Θ conditionally on a collision between the debris and the satellite. We develop in this chapter the SMC^2 (Sequential Monte Carlo Square) algorithm to estimate this kind of targeted laws introduced [5] to do filtering on hidden Markov models. We apply this island particle algorithm to debris satellite collision use case and analyse its results for the system safety.

2 Debris Satellite Collision Simulation

We consider two space objects (a debris and a satellite) orbiting around an Earth centered inertial reference frame. Their geometry is assumed spherical (i.e. the objects have a high tumbling motion when compared with their orbital period) and we assume that we perfectly know the radius of such sphere and the mass of the objects. We wonder about the relative position of the two satellites and ask whether the distance between the two objects could be smaller than a conflict distance T during the given time span I. To model the orbital motion of both space objects, we consider a general perturbation approach where the original equations of motion are replaced with an analytical approximation that captures the essential character of the motion over some limited time interval, which also enables analytical integration of the equations. SGP4 model [9] is used to propagate the trajectories of debris and satellite according to the time. At time t, the space objects will be represented by their 6-dimensional state vectors $\mathbf{s}_1(t)$ and $\mathbf{s}_2(t)$, i.e. their 3-dimensional position vectors $\mathbf{r}_1(t)$ and $\mathbf{r}_2(t)$ and their 3-dimensional speed vectors $\mathbf{v}_1(t)$ and $\mathbf{v}_2(t)$ such that $\mathbf{s}_i = (\mathbf{r}_i, \mathbf{v}_i)$. The initial conditions in the proposed example are defined in terms of two line elements (TLE), similar to those provided by NORAD (North American Aerospace Defense Command), as the SGP4 model is used for the orbital propagation of the considered objects. The initial condition value is denoted $\mathbf{s}_i^m$ at a given time t_i^m. SGP4 enables us to propagate the orbit of both space objects through time, denoted by a scalar continuous function ν such that

$$\forall i \in \{1, 2\}, \forall t \in I, \mathbf{s}_i(t) = \nu(\mathbf{s}_i^m, t_i^m, t),$$
$$\delta = \min_{t \in I}\{\|\mathbf{r}_2 - \mathbf{r}_1\|(t)\}.$$

The function of time $t \in I \mapsto \|\mathbf{r}_2 - \mathbf{r}_1\|(t)$ makes δ available through numerical optimisation in a deterministic approach. In fact, the position and velocity of space objects are estimated from more or less imprecise measurements. While the measurement means used for satellites (e.g. GPS, laser) result in a reasonable orbital accuracy (e.g. several tens of meters) the measurement means used for debris and uncooperative space-objects (e.g. mainly radar and telescopes) could result in quite imprecise orbits (e.g. several hundred of meters or few kilometers). This lack of accuracy will depend on a great number of factors. TLEs sum up this information and feed the models with the couple $(\mathbf{s}_i^m, t_i^m)$ for $i = 1, 2$, but to cope with their uncertainty, we have added independent and identically distributed Gaussian noises to the model inputs $\mathbf{s}_i^m$.

Debris satellite conflict may be modelled as an input-output function where:

- the input X represents the position and the speed of the debris space (the position and the speed of the satellite are assumed to be known). X is a 6-dimensional multivariate normal random vector of mean $(\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5, \Theta_6)^t$ and covariance matrix is equal to the identity matrix defined on a measurable space $(\mathsf{X}, \mathcal{X})$. The means corresponds to the debris measurement errors on its position and speed;
- the input-output function ϕ enables to propagate the debris and satellite trajectories with the SGP4 model during I. The input-output code includes the transformation that allows to switch from the standard space of the input to

the physical space in which evolve the satellite and debris position and speed. The function ϕ is a continuous positive scalar function $\phi : \mathbb{R}^6 \to \mathbb{R}$ and is static;

- the error on the drag coefficient which is considered inside the function ϕ is also random and follows a normal distribution with mean Θ_7 and variance 1;
- the output Y is the minimum distance between the debris and the satellite during I. We assume that it is a positive random variable.

The complete set of model parameters is summed up in the vector $\Theta = (\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5, \Theta_6, \Theta_7)^t$. The quantity of interest on the output Y is the probability

$$\mathbb{P}(Y < T) = \mathbb{P}(\phi(X) < T) \ .$$

When the event $\{\phi(X) < T\}$ is rare relatively to the available simulation budget (which is often the case in safety and reliability issues), different algorithms described in [1–4, 14, 15, 18] have notably been proposed to estimate accurately its probability.

3 Basics of Safety Analysis

In the present chapter, one focuses on the case where the law of X is uncertain and depends upon unknown parameters. We assume that X is distributed according to a well known parametric model and its parameters, denoted by a random vector Θ, have a probability density ν. We also suppose that Θ has a density f_Θ w.r.t. a dominating measure of reference λ, that is

$$\nu(d\theta) = f_\Theta(\theta) \ \lambda(d\theta) \ .$$

In the application considered here, X is a random vector with a multivariate normal distribution, and Θ describe the mean of X. It corresponds notably to realistic applications where it is not always possible to evaluate accurately the density of input parameters. This formalism enables thus to consider a large range of input probability density function.

The probability of interest $\mathbb{P}(Y < T)$ depends of course on Θ and thus on the distribution ν. In safety applications, it is important to estimate a superior bound of the rare event probability $\mathbb{P}(Y < T)$ taking also into account the prior on Θ. The prior on Θ is important since unrealistic bad tuning values of Θ which lead to high probabilities $\mathbb{P}(Y < T)$ are not relevant. The idea of this chapter is thus to determine the distribution of Θ conditionally on the fact that Y does not exceed the threshold T. This distribution, denoted by π, will be referred to in the sequel as the *target law*.

In the further development, when there is no confusion, we sometimes write $\mathbb{P}(Y < T|\theta)$ instead of $\mathbb{P}(Y < T|\Theta = \theta)$.

Note that using the Bayes' formula, the target law can be written

$$\pi(d\theta) = \frac{1}{\mathbb{P}(Y < T)} \mathbb{P}(Y < T|\theta)\nu(d\theta) \ . \tag{1}$$

We propose in this paper a SMC (Sequential Monte Carlo) algorithm which evolves according to iterative selection and mutation steps, and which approximates π when the number of particles gets large. This algorithm requires the estimation of $\mathbb{P}(\phi(X) < T|\Theta = \theta)$ for different settings of parameter θ. For that purpose, we describe the splitting algorithm that enables us to estimate this probability with accuracy.

4 The SMC^2 Algorithm

4.1 Principle

The SMC^2 algorithm is based on the use of two sets of particles to iteratively approach π. The first set of particles is defined on the parameter Θ and the second set of particles is useful to estimate the probabilities $\mathbb{P}(Y < T|\theta)$. The complete demonstration of interacting particles systems (IPS) convergence and the link with Feynman-Kac framework is given in [10].

Define $T_1, T_2, \ldots, T_n = T$ a series of decreasing thresholds and denote for all $i \in [\![0, n]\!]$

$$\pi_i = \frac{1}{\mathbb{P}(Y < T_i)} \mathbb{P}(Y < T_i|\theta)\nu(d\theta) \ .$$

The target law is of course $\pi = \pi_n$. The probability law π_n is thus proportional to

$$\pi_n \propto \mathbb{P}(Y < T_n|\theta) \ \nu(d\theta)$$

$$\pi_n \propto H_n(\theta) \ \nu(d\theta) \ ,$$

where $H_n(\theta) = \mathbb{P}(Y < T_n|\theta)$. The term $H_n(\theta)$ can be expressed as a product of conditional probabilities

$$H_n(\theta) = \left[\prod_{p=1}^{n-1} \mathbb{P}(Y < T_{p+1}|Y < T_p, \theta) \right] \times \mathbb{P}(Y < T_1|\theta) = \prod_{p=0}^{n-1} h_p(\theta) \ , \tag{2}$$

with

$$\begin{cases} h_p(\theta) = \mathbb{P}(Y < T_{p+1} | Y < T_p, \theta) \\ h_0(\theta) = \qquad \mathbb{P}(Y < T_1 | \theta) \ . \end{cases}$$

In this notation, we have

$$\pi_n \propto \prod_{p=0}^{n-1} h_p(\theta) \ \nu(d\theta) \ . \tag{3}$$

One can also remark that $H_p = H_{p-1} \times h_{p-1}$ and consequently the link between π_{p+1} and π_p can be written on the following way

$$\pi_{p+1} = \psi_{h_p}(\pi_p), \tag{4}$$

where ψ_{h_p} is the so-called Boltzmann-Gibbs transformation. Let $\mathcal{P}(\mathbb{E})$ be the set of probability measures on $\mathbb{E}$. For all positive bounded function G, the Boltzmann-Gibbs transformation $\Psi_G : \mathcal{P}(\mathbb{E}) \to \mathcal{P}(\mathbb{E})$ is defined for all $\mu \in \mathcal{P}(\mathbb{E})$ such that $\mu(G) = \int G(x) \, \mu(dx) > 0$ by

$$\Psi_G(\mu)(dx) := \frac{1}{\mu(G)} G(x) \, \mu(dx).$$

If one assumes that it is possible to determine a Markovian kernel M_p that let π_p invariant (which is not restrictive using, for example, a stage of the acceptance/rejection of Metropolis-Hastings algorithm) we have

$$\pi_p = (\pi_p M_p)(d\theta) = \int \pi_p(d\theta') M_p(\theta', d\theta). \tag{5}$$

This yields the evolution equation

$$\pi_{p+1} = \psi_{h_p}(\pi_p) M_{p+1} \ . \tag{6}$$

Equation 6 may be cast in the Feynman-Kac framework and then, each measure π_p can be approximated by an IPS which evolves with selection steps related to the so called potential functions h_p and mutation steps related to the Markov kernel M_p. Denote by $\{(\theta_p^1, \ldots, \theta_p^{N_1})\}_{n \geq 0}$ a system of N_1 particles.

$$\begin{bmatrix} \theta_0^1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \theta_0^{N_1} \end{bmatrix} \xrightarrow{\text{Selection}} \begin{bmatrix} \hat{\theta}_0^1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \hat{\theta}_0^{N_1} \end{bmatrix} \xrightarrow{\text{Mutation}} \begin{bmatrix} \theta_1^1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \theta_1^{N_1} \end{bmatrix} \xrightarrow{\text{Selection}} \begin{bmatrix} \hat{\theta}_1^1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \hat{\theta}_1^{N_1} \end{bmatrix} \xrightarrow{\text{Mutation}}$$

The selection stage consists in sampling $\{\hat{\theta}_p^i\}_{i=1}^{N_1}$ independently according to the probability measure ψ_{h_p}, i.e. selecting the particles $\{\theta_p^i\}_{i=1}^{N_1}$ with probabilities proportional to their weights $\{h_p(\theta_p^i)\}_{i=1}^{N_1}$. The mutation stage consists in updating the selected particles conditionally independently using the Markov kernel M_{p+1} that let π_{p+1} invariant. This step enables to increase the diversity of $\hat{\theta}_p$ without changing its probability law, that is already close to π_{p+1}. The Feynman-Kac theory [10] ensures that at each transition stage p :

$$\frac{1}{N_1}\sum_{i=1}^{N_1} \delta_{\theta_p^i} \xrightarrow[N_1 \to +\infty]{} \pi_p \ .$$

Thus, at the end of the nth transition stage, the system of particles converges to the target law π_n so that

$$\frac{1}{N_1}\sum_{i=1}^{N_1} \delta_{\theta_n^i} \xrightarrow[N_1 \to +\infty]{} \pi_n \ .$$

Nevertheless, the knowledge of h_p is required to apply the different selection/mutation stages. In practice, $h_p(\theta_p^i)$ is not analytically computable but can be estimated by defining a new set of particles $\{\xi_p^{i,j}\}_{j=1}^{N_2}$ on the random variable X conditionally to the different thresholds T_p and associated to each θ_p^i.

4.2 Description

Consider $\{\theta_0^i\}_{i=1}^{N_1}$ generated with probability law ν. At iteration k of the algorithm, with $k \geq 1$, we assume that particles $\{\theta_k^i\}_{i=1}^{N_1}$ are available and then, the interacting island algorithm consists in two iterative type stages:

- **Selection stage** The selection stage consists in choosing randomly and independently N_1 particles amongst $\{\theta_k^i\}_{i=1}^{N_1}$ with probabilities proportional to their weights $\{h_k(\theta_k^i)\}_{i=1}^{N_1}$. Thus, the particles with low weights are killed whereas those with high weights are multiplied. The number of particles is kept constant in this stage and a new set particles $\{\hat{\theta}_k^i\}_{i=1}^{N_1}$ can be defined.

Remind that the potential functions h_k are defined by:

$$\begin{cases} h_k(\theta_k^i) = \mathbb{P}(Y < T_{k+1} | Y < T_k, \theta = \theta_k^i), k \geq 1 \\ h_0(\theta_0^i) = \mathbb{P}(X < T_1 | \theta = \theta_0^i) \end{cases}$$

These quantities have to be computed.

- **Mutation stage** Even if the number of particles is still equal to N_1, some particles have been duplicated, so we apply a Markov kernel to increase the diversity of the particles. Building a π_{k+1}-reversible transition kernel that let π_{k+1} invariant is the objective of mutation stage. For that purpose, the acceptance/rejection step of the Metropolis-Hastings algorithm [17] is useful. This approach results in the exploration of Θ space set without changing the $\left(\hat{\theta}_k^i\right)_{1 \leq i \leq N_1}$ distribution and the increase of the particle diversity. A new particle $\theta_k^{i\prime}$ is proposed with a ν-reversible kernel Q. The acceptation rate of a new proposal is consequently $1 \wedge \frac{H_{k+1}(\hat{\theta}_k^{i\prime})}{H_{k+1}(\hat{\theta}_k^i)}$. If $H_{k+1}(\hat{\theta}_k^{i\prime}) > H_{k+1}(\hat{\theta}_k^i)$, the proposal $\hat{\theta}_k^{i\prime}$ is automatically accepted and replaces $\hat{\theta}_k^i$ in the set of current particles. Otherwise, the proposal $\hat{\theta}_k^{i\prime}$ is accepted with probability $\frac{H_{k+1}(\hat{\theta}_k^{i\prime})}{H_{k+1}(\hat{\theta}_k^i)}$. This acceptance/rejection procedure is repeated N_{app} times to decrease the correlation between the particles. At the end of this stage, a new set of particles $\{\theta_{k+1}^i\}_{1=i}^{N_1}$ can be defined.

Mutation and selection stage are applied n times until reaching the target threshold T_n. At the end of the algorithm, the particles $\{\theta_n^i\}_{1=i}^{N_1}$ provides an estimate of π_n :

$$\hat{\pi}_n^{N_1} = \frac{1}{N_1} \sum_{i=1}^{N_1} \delta_{\theta_n^i} \ .$$

For $i \in [1, N_1]$ and $k \in [0, n]$, the point is to estimate each probability $\{h_l(\theta_k^i)\}_{1 \leq l \leq k}$. It can be done with another interacting particle system (also called, in that case, sequential Monte Carlo, importance splitting, subset simulation or subset sampling). It is a rare event estimation technique which considers the estimation of several conditional probabilities that are easier to evaluate than estimate only one probability through a very tough simulation. Its principle is also based on selection and mutation stages. Let us define $\{\xi_0^{i,j}\}_{j=1}^{N_2}$ with probability density $f_{X|\theta_j^i}$ w.r.t. λ_X, i.e. the density of X knowing $\Theta = \theta_k^i$. At iteration l of the algorithm, we assume that particles $\{\xi_l^{i,j}\}_{j=1}^{N_2}$ are available and then IPS consists in two iterative stages:

- **Selection stage** The selection stage consists in choosing randomly and independently N_2 particles amongst the particles $\{\xi_l^{i,j}\}_{j=1}^{N_2}$ which are above T_l. The particles which have not reached the threshold T_l are thus killed. The number of particles is kept constant, and a new set of particles $\{\hat{\xi}_l^{i,j}\}_{j=1}^{N_2}$ can be defined.
- **Mutation stage** The mutation stage is patterned with acceptance/rejection principle using the Metropolis-Hastings algorithm [17]. A new particle $\hat{\xi}_l^{i,j'}$ is then proposed with a Markov kernel $\tilde{Q}$. If $\phi(\hat{\xi}_l^{i,j'}) < T_l$, then the proposal is accepted with probability $1 \wedge \frac{f_{X|\theta_l^i}(\hat{\xi}_l^{i,j'})\tilde{Q}(\hat{\xi}_l^{i,j'},\hat{\xi}_l^{i,j})}{f_{X|\theta_l^i}(\hat{\xi}_l^{i,j})\tilde{Q}(\hat{\xi}_l^{i,j},\hat{\xi}_l^{i,j'})}$ and $\hat{\xi}_l^{i,j'}$ replaces $\hat{\xi}_l^{i,j}$ in the set of current particles. If $\phi(\hat{\xi}_l^{i,j'}) > T_l$, the proposal is automatically rejected and the particle $\hat{\xi}_l^{i,j}$ is remained. This acceptance/rejection procedure is repeated N_{app2} times to decrease the correlation between the particles. At the end of this stage, a new set of particles $\{\xi_{l+1}^{i,j}\}_{j=1}^{N_2}$ can be defined. An estimate $\hat{h}_l(\theta_k^i)$ of $h_l(\theta_k^i) = \mathbb{P}(Y < T_{l+1} | Y < T_l, \Theta = \theta_k^i)$ is given by the ratio between the number of $\{\xi_l^{i,j}\}_{j=1}^{N_2}$ particles such that $\phi(\xi_l^{i,j}) < T_{l+1}$ and the total number of particles N_2.

Mutation and selection stages are applied k times until reaching the target threshold T_k. At the end of the algorithm, $H_{k+1}(\theta_k^i) = \mathbb{P}(Y < T_{k+1} | \theta_k^i)$ is estimated by

$$\hat{H}_{k+1}(\theta_k^i) = \prod_{l=0}^{k} \hat{h}_l.$$

For a given particle θ_k^i, a complete set of particles $\{\xi_l^{i,j}\}_{1 \le j \le N_2}^{1 \le l \le k}$ is thus generated. An island particle is thus constituted of a particle θ_k^i and its associated $\{\xi_l^{i,j}\}_{1 \le l \le k}^{1 \le j \le N_2}$ particle set.

The SMC^2 algorithm is described more precisely in Algorithm 2. Interacting particle system for probability estimation required in Algorithm 2 is developed in Algorithm 3.

The determination of Q and $\tilde{Q}$, in the general case, implies the use of Metropolis-Hastings algorithm. Nevertheless, if μ is a standard normal distribution, a transition from x to z defined with the following expression

$$x \mapsto z = \sqrt{1-a}\, x + \sqrt{a}\, W, \tag{7}$$

where $W \sim \mathcal{N}(0, 1)$ and a is scalar parameter such as $a \in [0, 1]$, is μ-reversible. In order to use Eq. (7) instead of Metropolis-Hastings algorithm, it is also possible to apply a transformation on the variables X or Θ so that they follow a standard normal PDF. Depending on the available information on the PDF of X, several transformations can be proposed [7, 8, 11–13].

Algorithm 2 The SMC^2 algorithm

```
Setting definition:
Define the thresholds $T_1, \dots, T_n$, the sample sizes $N_1, N_2$ and the number of applications $N_{app}$ of Markov kernel $Q$.
Initialization:
Sample $\left(\theta_0^i\right)_{1\le i\le N_1}$ with probability law $\nu$.
for $i$ from 1 to $N_1$ do
    Sample $\left(\xi_0^{i,j}\right)_{1\le j\le N_2}$ according to the probability density $f_{X|\theta_0^i}$.
end for
Transition:
for $k$ from 0 to $n$ do
    Associate a system of particles $\left(\xi_l^{i,j}\right)_{1\le l\le k}^{1\le j\le N_2}$ to each $\theta_k^i$ in order to estimate $h_k\left(\theta_k^i\right)$ and $H_{k+1}\left(\theta_k^i\right)$ with Algorithm 3.
    Selection of the $\theta$-particles:
    Sample $I_k = \left(I_k^i\right)_{1\le i\le N_1}$ multinomially with probability proportional to $\{h_k\left(\theta_k^i\right)\}_{i=1}^{N_1}$.
    Set $\hat{\theta}_k^i = \theta_k^{I_k^i}$.
    Mutation of the $\theta$-particles:
    for $m$ from 1 to $N_{app}$ do
        for $i$ from 1 to $N_1$ do
            Sample $\hat{\theta}_k^{i\prime}$ with a $\nu$ reversible kernel $Q$.
            Sample $u$ with a uniform random variable.
            if $\left(u < 1 \wedge \frac{H_{k+1}(\hat{\theta}_k^{i\prime})}{H_{k+1}(\hat{\theta}_k^{i})}\right)$ then set $\theta_{k+1}^i = \hat{\theta}_k^{i\prime}$.
            else set $\theta_{k+1}^i = \hat{\theta}_k^i$.
            end if
        end for
        if $m < N_{app}$ then set $\hat{\theta}_k^i = \theta_{k+1}^i$.
        end if
    end for
end for
Estimation:
Estimate $\pi_n$ with $\hat{\pi}_n^{N_1} = \frac{1}{N_1}\sum_{i=1}^{N_1}\delta_{\theta_n^i}$
```

5 Estimation of Collision Probability Between Orbiting Objects

The SMC^2 algorithm has been applied on the debris satellite collision test case in order to estimate π, the conditional law of Θ given $\phi(X) < T$, with the following parameters: $N_1 = 1000$, $N_2 = 50$, $N_{app} = 1$, $N_{app2} = 1$. The intermediate thresholds T_i on the output distance are expressed in meters with $\{200, 100, 66, 50, 40, 33, 28, 25, 22, 20\}$. The estimators of the different marginals of π, obtained with the SMC^2 algorithm, are given in Fig. 1, where the first marginal is related to the first parameter and so on.

The estimated probabilities $\widehat{\mathbb{P}}(Y < T|\Theta = \theta)$, when Θ follows ν and π are represented in Fig. 2. The mean probability $\widehat{\mathbb{P}}(Y < T|\Theta = \theta)$ when Θ follows ν is

Algorithm 3 Interacting particle system for probability estimation

```
For a given value $\theta_k^i$, we build an IPS which allows to estimate both $h_k\left(\theta_k^i\right) = \mathbb{P}\left(Y < T_{k+1} | Y < T_k, \Theta = \theta_k^i\right)$ and $H_{k+1} = \prod_{p=0}^{k} h_p\left(\theta_k^i\right)$.
Setting definition:
Define the number of applications $N_{app2}$ of Markov kernel $\tilde{Q}$ and recall the iteration parameter $k$ and the particle value $\theta_k^i$, the thresholds $T_1, \dots, T_{k+1}$ and the sample size $N_2$, that have been defined or obtained in **Algorithm** 2.
Initialisation:
Sample $\left(\xi_0^{i,j}\right)_{1 \leq j \leq N_2}$ following probability density $f_{X|\theta_k^i}$.
Transition:
**for** $l$ from 0 to $k-1$ **do**
    Selection of the $\xi$ particles:
    **for** $j$ from 1 to $N_2$ **do**
        **if** $\phi\left(\xi_l^{i,j}\right) \leq T_{l+1}$ **then** set $\hat{\xi}_l^{i,j} = \xi_l^{i,j}$.
        **else** Sample $\hat{\xi}_l^{i,j}$ randomly and uniformly among particles which are below the threshold $T_{l+1}$.
        **end if**
    **end for**
    Mutation of the $\xi$ particles:
    **for** $r$ from 1 to $N_{app2}$ **do**
        **for** $j$ from 1 to $N_2$ **do**
            Sample $\hat{\xi}_l^{i,j\prime}$ according to $\tilde{Q}\left(\hat{\xi}_l^{i,j}, .\right)$.
            **if** $\phi\left(\hat{\xi}_l^{i,j\prime}\right) > T_{l+1}$ **then** set $\xi_{l+1}^{i,j} = \hat{\xi}_l^{i,j}$.
            **else** Sample $u$ with a uniform random variable.
                **if** $\left(u < 1 \wedge \frac{f_{|\theta_k^i}(\hat{\xi}_l^{i,j\prime})\tilde{Q}(\hat{\xi}_l^{i,j\prime},\hat{\xi}_l^{i,j})}{f_{|\theta_k^i}(\hat{\xi}_l^{i,j})\tilde{Q}(\hat{\xi}_l^{i,j},\hat{\xi}_l^{i,j\prime})}\right)$ **then** set $\xi_{l+1}^{i,j} = \xi_l^{i,j\prime}$
                **else** set $\xi_{l+1}^{i,j} = \hat{\xi}_l^{i,j}$
                **end if**
            **end if**
        **end for**
        **if** $r < N_{app2}$ **then** set $\hat{\xi}_l^{i,j} = \xi_{l+1}^{i,j}$.
        **end if**
    **end for**
    Set $\hat{h}_l\left(\theta_k^i\right) = \frac{1}{N_2}\sum_{j=1}^{N_2}\mathbb{1}_{\phi\left(\xi_l^{i,j}\right) \leq T_{l+1}}$
**end for**
Set $\hat{h}_k\left(\theta_k^i\right) = \frac{1}{N_2}\sum_{j=1}^{N_2}\mathbb{1}_{\phi\left(\xi_k^{i,j}\right) \leq T_{k+1}}$
Estimation:
Estimate $h_k\left(\theta_k^i\right)$ with $\hat{h}_k\left(\theta_k^i\right)$ and $H_{k+1}\left(\theta_k^i\right)$ with $\prod_{l=0}^{k}\hat{h}_l\left(\theta_k^i\right)$.
```

estimated to $3.9 \cdot 10^{-4}$. When $\theta = \sum_{i=1}^{N_\theta} \theta_m^i / N_\theta$, the probability $\widehat{\mathbb{P}}(Y < 20 | \Theta = \sum_{i=1}^{N_\theta} \theta_m^i / N_\theta)$ is equal to 0.034.

The question is how to analyze the estimated density of π for the tuning of Θ. A possible approach is to consider the Kullback-Leibler distance between the estimated marginal density of π for the parameter Θ_i and the initial marginal density of ν for parameter Θ_i. If the Kullback-Leibler distance is significant for Θ_i, then one

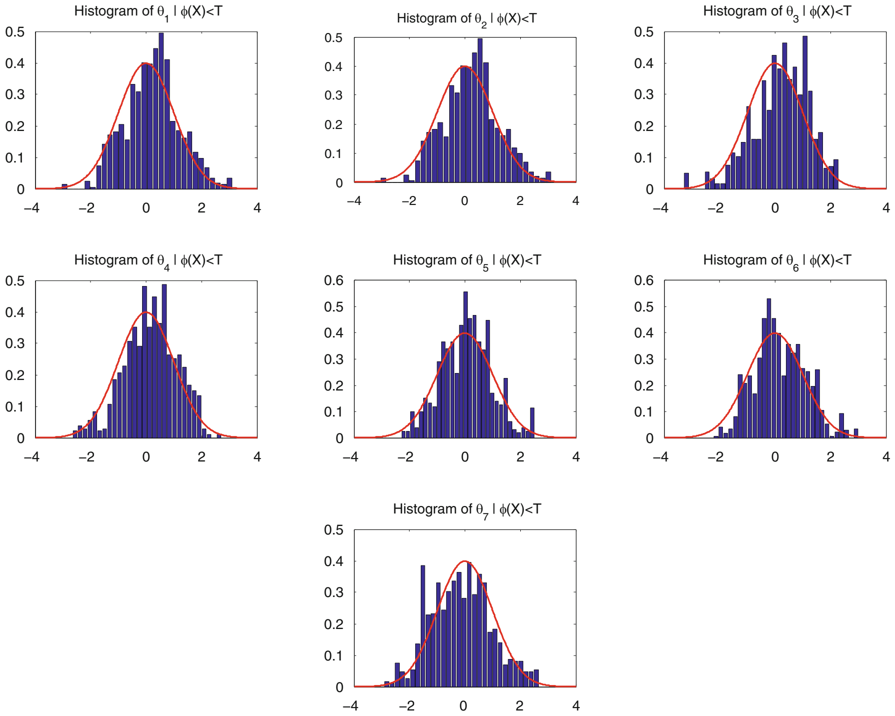


Fig. 1 Estimations of the marginals of π using the SMC^2 algorithm. The *red curve* corresponds to the standard normal density that is the initial marginal of the different parameters

can assume that Θ_i has to be finely tuned and conversely. In that case, a misestimation of Θ_i will indeed tend to increase the failure probability. Table 1 summaries the different Kullback-Leibler distances obtained for the different components of Θ. The first error component Θ_1 of the position vector seems to be the most influent parameter on $\mathbb{P}(\phi(X) < T)$. On the contrary, the second error component of position and speed vector, that are Θ_2 and Θ_5 require a lower accuracy since the considered values for these parameters lead the maximum of the collision probability; an error on these parameters will thus tend to decrease the failure probability. In the same way, the density parameter Θ_7 of the drag coefficient does not require also a too fine tuning in the proposed example.

It may be also interesting in practice to transform the six first components of Θ into usual orbital parameters (the semi-major axis a, eccentricity e, inclination i, argument of perigee ω, longitude of the ascending node Ω, the mean anomaly m) and then to evaluate π in that case. The estimation of the marginals of π for the different orbital parameters is proposed in Fig. 3. The corresponding Kullback-Leibler analysis is given in Table 2. The mean anomaly is on this use case the orbital parameter that has to be most finely tuned. There is indeed a higher chance that the collision probability increases if the mean anomaly is not correctly set.

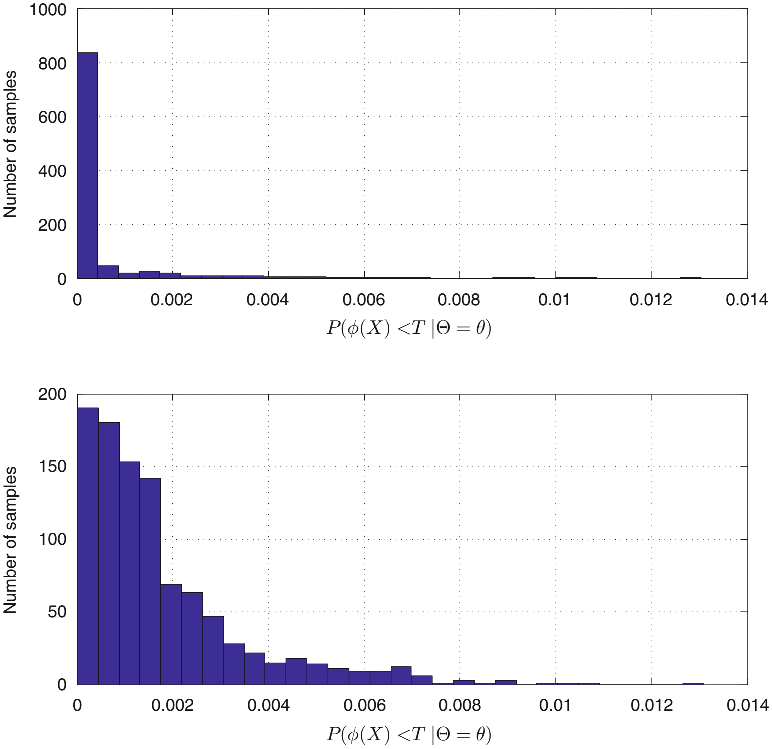


Fig. 2 Estimates of $\mathbb{P}(Y < T|\theta)$ with Θ following ν and π

Table 1 Kullback-Leibler distance between marginal density π and ν for parameters Θ_i

Component of Θ	Kullback-Leibler distance with the marginal of π
Θ_1	0.46
Θ_2	0.13
Θ_3	0.30
Θ_4	0.24
Θ_5	0.11
Θ_6	0.25
Θ_7	0.10

6 Conclusion

In this chapter, we have proposed an original methodology to analyze the influence of parameter model that are set for the sake of simplicity, on a rare failure probability. The proposed SMC^2 algorithm has been described in the case of a general problem where the model is a black-box system with random inputs.

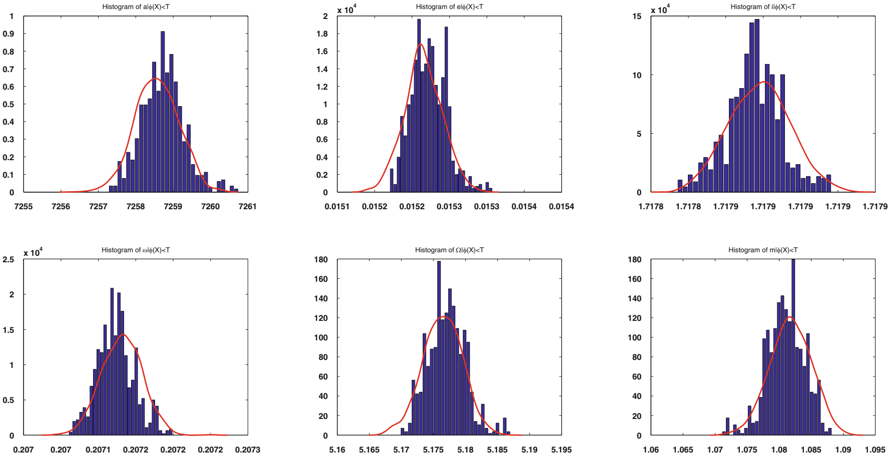


Fig. 3 Estimations of the marginals of π using the SMC^2 algorithm on the orbital parameters. The *red curve* corresponds to the initial density of the orbital parameters

Table 2 Kullback-Leibler distance between marginal density π and ν for the orbital parameters

Orbital parameters	Kullback-Leibler distance with the marginal of π
a	0.24
e	0.20
i	0.16
ω	0.23
Ω	0.15
m	0.34

This algorithm has been applied with success for the analysis of collision probability between space debris and satellite. The set model parameters influence strongly the value of the collision probability and their value has to be carefully investigated to avoid collision probability underestimation.

The complete interpretation of target law π remains complicated and has to be continued. The analysis of the particles obtained by the SMC^2 algorithm with Sobol indices [16] is a potential perspective to this work.

Acknowledgement The work presented in this chapter is part of a CNES/ONERA PhD thesis.

References

1. Bjerager, R.: Methods for Structural Reliability Computation, pp. 89–136. Springer, New York (1991)
2. Botev, Z.I., Kroese, D.P.: Efficient Monte-Carlo simulation via the generalized splitting method. Stat. Comput. **22**(1), 1–16 (2012)

3. Bucklew, J.A.: Introduction to Rare Event Simulation. Springer, Berlin (2004)
4. Cérou, F., Del Moral, P., Furon, T., Guyader, A.: Sequential Monte Carlo for rare event estimation. Stat. Comput. **22**(3), 795–808 (2012)
5. Chopin, N., Jacob, P.E., Papaspiliopoulos, O.: SMC^2: an efficient algorithm for sequential analysis of state space models. J. R. Stat. Soc. Ser. B Stat Methodol. **75**(3), 397–426 (2013)
6. Kelso, T.: Analysis of the Iridium 33-Cosmos 2251 collision. In: Ryan, S., The Maui Economic Development Board (eds.) Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference, Wailea, p. E3 (2009)
7. Lebrun, R., Dutfoy, A.: A generalization of the Nataf transformation to distributions with elliptical copula. Probab. Eng. Mech **24**(2), 172–178 (2009)
8. Lebrun, R., Dutfoy, A.: An innovating analysis of the Nataf transformation from the copula viewpoint. Probab. Eng. Mech. **24**(3), 312–320 (2009)
9. Miura, N.Z.: Comparison and design of Simplified General Perturbation Models (SGP4) and code for NASA Johnson Space Center, Orbital Debris Program Office. Ph.D. Thesis, Faculty of California Polytechnic State University (2009)
10. Del Moral, P., Hu P., Wu, L.: On the concentration properties of interacting particle processes. Foundations and Trends in Machine Learning, vol. 3. Now Publishers, Hanover (2012)
11. Nataf, A.: Distribution des distributions dont les marges sont données. C. R. Acad. Sci. **225**, 42–43 (1962)
12. Pei-Ling, L., Der Kiureghian, A.: Optimization algorithms for structural reliability. Struct. Saf. **9**(3), 161–177 (1991)
13. Rosenblatt, M.: Remarks on a multivariate transformation. Ann. Math. Stat. **23**, 470–472 (1952)
14. Rubinstein, R.Y., Kroese, D.P.: The Cross-Entropy Method: a unified approach to combinatorial optimization, Monte Carlo Simulation and Machine Learning. Springer, Berlin (2004)
15. Sobol, I.M.: A Primer for the Monte Carlo Method. CRC Press, Boca Raton, FL (1994)
16. Sobol, I.M., Kuchereko, S.: Sensitivity estimates for non linear mathematical models. Math. Model. Comput. Exp. **1**, 407–414 (1993)
17. Tierney, L.: Markov chains for exploring posterior distributions. Ann. Stat. **22**, 1701–1762 (1994)
18. Zhang, P.: Nonparametric importance sampling. J. Am. Stat. Assoc. **91**(434), 1245–1253 (1996)

Flatness-Based Low-Thrust Trajectory Optimization for Spacecraft Proximity Operations

Le-ping Yang, Wei-wei Cai, and Yan-wei Zhu

Abstract This chapter presents a novel computational framework integrating the differential flatness theory and the analytic homotopic technique for the low-thrust trajectory optimization for spacecraft proximity operations. Based on the flatness property of relative motion equations, the trajectory optimization problem is transformed into the flat output space with all the differential constraints eliminated and the number of decision variables reduced. Then the mapped Chebyshev pseudospectral method is applied to parameterizing the profiles of flat outputs, whose high-order derivatives are enhanced by improving the differential matrix's ill- conditioning. Furthermore, the analytic homotopic technique is introduced to improve the applicability to non-smooth trajectory optimization problems. Numerical simulation results show that the proposed framework scheme is feasible and effective for spacecraft proximity maneuvers.

Keywords Differential flatness • Homotopic technique • Mapped pseudospectral method • On-orbit operation • Non-smooth trajectory

1 Introduction

Spacecraft proximity operations are becoming increasingly important because they enable critical capabilities such as autonomous on-orbit assembly and inspection, servicing of disabled spacecraft, or debris deorbiting. Of particular interest in this field is the optimization of proximity maneuvering trajectories facilitated by miniaturized and high-efficiency propulsion technologies. The low-thrust trajectory optimization for proximity relative motion could be formulated as a constrained nonlinear optimal control problem, which is usually computationally intractable due

L.-p. Yang (✉) • Y.-w. Zhu
College of Aerospace Science and Engineering, National University of Defense Technology, Changsha, Hunan, P.R. China 410073
e-mail: ylp_1964@163.com

W.-w. Cai
College of Basic Education, National University of Defense Technology, Changsha, Hunan 410073, P.R. China

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization and Its Applications 114, DOI 10.1007/978-3-319-41508-6_18

to the differential dynamic equations and the large dimensionality of state space. In this chapter, we would propose a differential flatness-based hybrid computational framework to facilitate the optimization process.

Due to the flatness property of the proximity relative motion model, the system states and inputs could be formulated as functions of flat outputs and their derivatives. Thus the original trajectory optimization problem could be transformed to optimizing the flat output profiles with the elimination of differential constraints and reduction of design variables [1, 2]. After obtaining the flat output solutions, they are mapped back into the original domain, generating the nominal trajectory and the corresponding inputs. For the flat output optimization problem, it is ultimately converted into a nonlinear programming (NLP) problem by parameterizing the flat outputs with Chebyshev pseudospectral method (CPM), where the high-order derivatives of flat outputs could be easily computed via the pseudospectral differentiation matrices. However, the ill-conditioning of the standard pseudospectral differentiation matrices would evidently influence the numerical accuracy of the high-order derivatives. Thus the conformal map and barycentric rational interpolation techniques are utilized to improve the ill-conditioning of pseudospectral differentiation matrices.

Note that the aforementioned approach is quite efficient and accurate for smooth trajectories, but its application to bang-bang type optimal control problems, such as time-optimal low-thrust proximity maneuvering, would induce some numerical difficulties. Specifically, due to the discontinuities in the derivatives of flat outputs, the obtained solutions would exhibit the well-known Gibbs phenomenon which is resulted from approximating a non-smooth function with a finite number of smooth functions [3]. Though adding more discretization nodes for mesh refinement may be working, it would result in inefficiencies. In this chapter, the non-smooth difficulty is eliminated by using the analytic homotopic approach. For example, given the time-optimal low-thrust trajectory planning problem, the related but smoother energy-optimal trajectory planning problem is firstly introduced and solved utilizing the aforementioned numerical approach. Based on the obtained energy-optimal trajectory, the analytic homotopic approach constructs an auxiliary optimal control problem whose costates are simply zero, avoiding the difficulties of initial costates estimation in the traditional homotopic approach. Clearly, the proposed hybrid framework successfully addresses the issues of pseudospectral method and homotopic approach when they are applied separately. In the end, numerical simulations are presented to validate the performance of the hybrid computational framework.

2 Relative Translational Dynamic Equations

The mathematical model for the relative motion between two spacecraft, i.e. the leader *Sat*A and the follower *Sat*B, is briefly developed in this section. Note that only two spacecraft are considered here, since a multi-spacecraft scenario can be

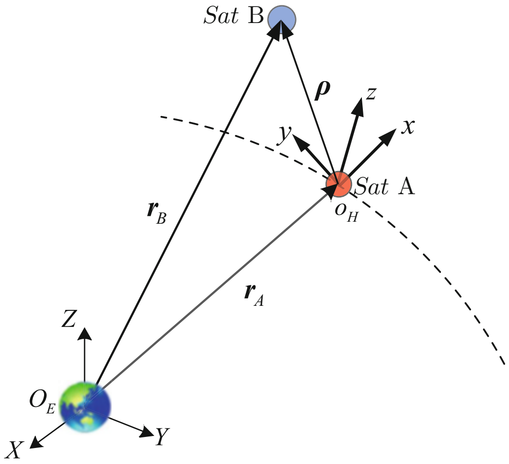


Fig. 1 Geometry of the coordinate frames

similarly analyzed and addressed. For the purposes of our formulation, we shall assume that the follower is forced while the leader is not. In addition, all the non-Keplerian acceleration (for example, caused by J2 or atmospheric drag) is neglected here.

The relative motion is described in the rotating Hill orbit frame $\boldsymbol{O}_{\mathrm{H}}$-$xyz$ whose origin is chosen to be at the mass center of *Sat*A as shown in Fig. 1. The Cartesian coordinates of x, y and z are aligned with the directions of the orbit radial (outward), orbital velocity vector and normal vector with respect to the orbital plane. Moreover, to describe the inertial motions of both satellites, the ECI frame $\boldsymbol{O}_{\mathrm{E}}$-$XYZ$ is defined with its origin located at the center of the Earth, the X and Z axis aligned with the equinox and the Earth's self-spin axis, while the Y axis completes the right-hand triad.

The inertial equations of motion of both spacecraft (assuming the Earth is spherical) are respectively given by

$$\frac{d^2\mathbf{r}_A}{dt^2} = -\frac{\mu\mathbf{r}_A}{r_A^3} \tag{1}$$

$$\frac{d^2\mathbf{r}_A}{dt^2} = -\frac{\mu\mathbf{r}_B}{r_B^3} + \mathbf{f}_B \tag{2}$$

where $\mathbf{r}_A$ and $\mathbf{r}_B$ denote the leader and follower positions relative to the center of the Earth; $d(\,)/dt$ represents the time derivative in frame $\boldsymbol{O}_{\mathrm{E}}$-$XYZ$; $\mathbf{f}_B$ is the acceleration caused by the thrusters and $\mu = 3.986 \times 10^{14}\ \mathrm{m}^3/\mathrm{s}^2$ is the gravitational constant of Earth.

Let $\boldsymbol{\rho} = \mathbf{r}_B - \mathbf{r}_A$ denote the position vector of *Sat*B relative to *Sat*A. Subtracting Eq. (1) from Eq. (2) yields

$$\begin{aligned}\frac{d^2\boldsymbol{\rho}}{dt^2} &= \frac{d^2\mathbf{r}_B}{dt^2} - \frac{d^2\mathbf{r}_A}{dt^2} = -\left(\frac{\mu}{r_B^3}\mathbf{r}_B - \frac{\mu}{r_A^3}\mathbf{r}_A\right) + \mathbf{f}_B \\ &= \frac{\mu}{r_A^3}\left[\mathbf{r}_A - \left(\frac{r_A}{r_B}\right)^3 \mathbf{r}_B\right] + \mathbf{f}_B\end{aligned} \tag{3}$$

In order to express the relative motion in frame $\boldsymbol{O}_{\mathrm{H}}$-$xyz$, we recall that

$$\frac{d^2\boldsymbol{\rho}}{dt^2} = \frac{\delta^2\boldsymbol{\rho}}{\delta t^2} + \frac{\delta\boldsymbol{\omega}_A}{\delta t} \times \boldsymbol{\rho} + 2\boldsymbol{\omega}_A \times \frac{\delta\boldsymbol{\rho}}{\delta t} + \boldsymbol{\omega}_A \times (\boldsymbol{\omega}_A \times \boldsymbol{\rho}) \tag{4}$$

where $\boldsymbol{\omega}_A$ is the angular vector of frame $\boldsymbol{O}_{\mathrm{H}}$-$xyz$ relative to frame $\boldsymbol{O}_{\mathrm{E}}$-$XYZ$, and $\delta(\ )/\delta t$ represents the time derivative in frame $\boldsymbol{O}_{\mathrm{H}}$-$xyz$.

Let a, e, n, θ denote the semi-major axis, eccentricity, mean angular velocity and true anomaly of *Sat*A, then we can obtain the following auxiliary relations:

$$n = \sqrt{\frac{\mu}{a^3}},\ r_A = \frac{a\left(1-e^2\right)}{1+e\cos\theta},\ \dot{\theta} = \frac{n\,(1+e\cos\theta)}{(1-e^2)^{3/2}},\ \ddot{\theta} = \frac{-2n^2 e\sin\theta(1+e\cos\theta)^3}{(1-e^2)^3} \tag{5}$$

As $\boldsymbol{\omega}_A$ is normal to the orbital plane, we may write

$$\boldsymbol{\omega}_A = \left[0\ 0\ \dot{\theta}\right]^{\mathrm{T}},\ \ \delta\boldsymbol{\omega}_A/\delta t = \left[0\ 0\ \ddot{\theta}\right]^{\mathrm{T}} \tag{6}$$

Also, let

$$\boldsymbol{\rho} = \left[x_{\mathrm{H}}\ y_{\mathrm{H}}\ z_{\mathrm{H}}\right]^{\mathrm{T}},\ \delta\boldsymbol{\rho}/\delta t = \left[\dot{x}_{\mathrm{H}}\ \dot{y}_{\mathrm{H}}\ \dot{z}_{\mathrm{H}}\right]^{\mathrm{T}},\ \mathbf{r}_A = \left[r_A\ 0\ 0\right]^{\mathrm{T}} \tag{7}$$

Substituting Eqs. (3), (6) and (7) into Eq. (4) yields the following component-wise equations for relative motion:

$$\begin{cases}\ddot{x}_{\mathrm{H}} = \dot{\theta}^2 x_{\mathrm{H}} + \ddot{\theta} y_{\mathrm{H}} + 2\dot{\theta}\dot{y}_{\mathrm{H}} + \mu/r_A^2 - \mu\,(r_A + x_{\mathrm{H}})\,/r_B^3 + f_{Bx} \\ \ddot{y}_{\mathrm{H}} = -\ddot{\theta} x_{\mathrm{H}} + \dot{\theta}^2 y_{\mathrm{H}} - 2\dot{\theta}\dot{x}_{\mathrm{H}} - \mu y_{\mathrm{H}}/r_B^3 + f_{By} \\ \ddot{z}_{\mathrm{H}} = -\mu z_{\mathrm{H}}/r_B^3 + f_{Bz}\end{cases} \tag{8}$$

where $r_B = \sqrt{(r_A + x_{\mathrm{H}})^2 + y_{\mathrm{H}}^2 + z_{\mathrm{H}}^2}$.

Assuming that the relative distance is much less than the orbital radius of *Sat*A, a more compact form can be obtained by applying the first-order linearization technique [4]

$$\begin{cases}\ddot{x}_{\mathrm{H}} - 2\dot{\theta}\dot{y}_{\mathrm{H}} - \dot{\theta}^2 x_{\mathrm{H}} - \ddot{\theta} y_{\mathrm{H}} - 2\mu x_{\mathrm{H}}/r_B^3 = f_{Bx} \\ \ddot{y}_{\mathrm{H}} + 2\dot{\theta}\dot{x}_{\mathrm{H}} - \dot{\theta}^2 y_{\mathrm{H}} + \ddot{\theta} x_{\mathrm{H}} + \mu y_{\mathrm{H}}/r_B^3 = f_{By} \\ \ddot{z}_{\mathrm{H}} + \mu z_{\mathrm{H}}/r_B^3 = f_{Bz}\end{cases} \tag{9}$$

where $\mu/r_B^3 = n^2(1 + e\cos\theta)^3/\left(1 - e^2\right)^3$.

Considering the capability of the low-thrust actuators on *Sat*B, the acceleration $\mathbf{f}_B$ may be written as

$$\mathbf{f}_B = A_{\max}\mathbf{u} \tag{10}$$

where $A_{\max}$ denotes the magnitude of the maximum acceleration.

When the leader is moving along a circular orbit, i.e. the eccentricity $e = 0$, the time derivatives of the true anomaly are

$$\dot{\theta} = n, \quad \ddot{\theta} = 0 \tag{11}$$

Substituting Eqs. (10) and (11) into Eq. (9) yields

$$\begin{cases} \ddot{x}_{\mathrm{H}} - 2n\dot{y}_{\mathrm{H}} - 3n^2 x_{\mathrm{H}} = A_{\max}u_x \\ \ddot{y}_{\mathrm{H}} + 2n\dot{x}_{\mathrm{H}} = A_{\max}u_y \\ \ddot{z}_{\mathrm{H}} + n^2 z_{\mathrm{H}} = A_{\max}u_z \end{cases} \tag{12}$$

The linearized equations of motion are called the Clohessy-Wiltshire (C-W) equations or the Hill equations. Note that the in-plane relative motion and the out-of-plane motion are obviously decoupled. Thus when given a proximity relative motion planning problem, we could address the in-plane and out-of-plane trajectories separately.

3 Problem Formulation

3.1 Formulation in State Space

For a typical on-orbit proximity operations mission, the system needs to observe some constraints for safety or actuator's capability considerations. Among the various kinds of constraints, the aforementioned dynamic model is a typical kind which ensures the physical feasibility of the maneuver trajectory. Moreover, the limitations on allowable control inputs contribute another important kind:

$$\mathbf{u}_{\min} \le \mathbf{u} \le \mathbf{u}_{\max} \tag{13}$$

The performance criteria denote the designer's requirement for the proximity operations. Without loss of generality, the performance criterion could be written in Bolza form as

$$J = \Phi\left[\mathbf{x}\left(t_0\right), \mathbf{x}\left(t_f\right), t_0, t_f\right] + \int_{t_0}^{t_f} \Lambda\left[\mathbf{x}(t), \mathbf{u}(t), t\right] dt \tag{14}$$

where $\Phi(\cdot)$ denotes the Mayer cost and $\Lambda(\cdot)$ the Lagrangian cost. Note that $\mathbf{x} \in \mathbb{R}^n$ here is defined as $\left[x_{\mathrm{H}}, y_{\mathrm{H}}, \dot{x}_{\mathrm{H}}, \dot{y}_{\mathrm{H}}\right]^{\mathrm{T}}$ for the in-plane motion and $\left[z_{\mathrm{H}}, \dot{z}_{\mathrm{H}}\right]^{\mathrm{T}}$ for the out-of-plane motion. The symbol $\mathbf{u} \in \mathbb{R}^m$ $(m \leq n)$ can be set similarly.

Till now, we can formulate the low-thrust trajectory optimization problem as the following optimal control problem.

Problem A Find the state-control profile $\{\mathbf{x}(t), \mathbf{u}(t)\}, t \in \left[t_0, t_f\right]$, possibly also the endpoint time t_f, that minimize the performance criterion in Eq. (14), and subject to the constraints on controls in Eq. (13), the system equations

$$\dot{\mathbf{x}}(t) = \mathbf{g}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \tag{15}$$

and the boundary conditions:

$$\mathbf{x}\left(t_0\right) = \mathbf{x}_0, \ \mathbf{x}\left(t_f\right) = \mathbf{x}_f \tag{16}$$

where $\mathbf{g}(\cdot)$ denotes the aforementioned governing equations.

For the proximity relative motion problem, considering the agile orbital maneuver capability required by many space missions, the minimum maneuver time in Eq. (17) is an important performance criterion. On the other hand, the minimum control effort in Eq. (18) constitutes another important index since it indicates the engineering feasibility of space missions, especially for those realized by propellant-based actuators.

$$J_1 = t_f \tag{17}$$

$$J_2 = \int_{t_0}^{t_f} \sum_{i=1}^{m} u_i^2 dt \tag{18}$$

3.2 Reformulation in Flat Output Space

Differential flatness reveals a structural property of general nonlinear systems, denoting that all states and inputs can be expressed in terms of a set of differentially independent variables and their time derivatives. More precisely, consider the nonlinear system in Eq. (15). It is differentially flat if and only if there is a vector $\varsigma \in \mathbb{R}^m$ with differentially independent components, so-called flat outputs, such that the state and control input vectors could be reformulated as

$$\begin{cases} \mathbf{x} = \boldsymbol{\Gamma}_{\mathbf{x}}\left(\varsigma, \dot{\varsigma}, \cdots, \varsigma^{(\eta-1)}\right) \\ \mathbf{u} = \boldsymbol{\Gamma}_{\mathbf{u}}\left(\varsigma, \dot{\varsigma}, \cdots, \varsigma^{(\eta)}\right) \end{cases} \tag{19}$$

where $\mathbf{\Gamma_x}$, $\mathbf{\Gamma_u}$ are smooth function; $\varsigma_i^{(k)}$ denotes the kth order time derivative of the ith component of the vector ς; and $\boldsymbol{\eta} = [\eta_1 \ldots \eta_m]^{\mathrm{T}}$ the relative degree of ς

$$\eta_i = \min\left\{k \in \mathrm{N}^*, \exists j \in \{1, \ldots, m\} \middle| \, \partial \varsigma_i^{(k)} / \partial u_j \neq 0\right\}, (i = 1, \ldots, m) \tag{20}$$

For the proximity relative motion problem studied here, the aforementioned dynamic model is obviously flat. For example, the equations for the in-plane motion could be rewritten as:

$$\begin{cases} \dot{x}_1 = x_3, \quad \dot{x}_2 = x_4 \\ \dot{x}_3 = 3n^2 x_1 + 2n x_4 + A_{\max} u_1 \\ \dot{x}_4 = -2n x_3 + A_{\max} u_2 \end{cases} \tag{21}$$

where $\mathbf{x} = [x_1, x_2, x_3, x_4]^{\mathrm{T}} = \left[x_{\mathrm{H}}, y_{\mathrm{H}}, \dot{x}_{\mathrm{H}}, \dot{y}_{\mathrm{H}}\right]^{\mathrm{T}}$, $\mathbf{u} = [u_1, u_2]^{\mathrm{T}} = \left[u_x, u_y\right]^{\mathrm{T}}$.

Let $\boldsymbol{\varsigma} = [\varsigma_1, \varsigma_2]^{\mathrm{T}} = [x_1, x_2]^{\mathrm{T}}$ be the candidate flat output vector, the in-plane motion states and control inputs can be written as

$$\begin{cases} x_1 = \varsigma_1, \quad x_2 = \varsigma_2 \\ x_3 = \dot{\varsigma}_1, \quad x_4 = \dot{\varsigma}_2 \\ u_1 = \left(\ddot{\varsigma}_1 - 3n^2 \varsigma_1 - 2n \dot{\varsigma}_2\right) / A_{\max} \\ u_2 = \left(\ddot{\varsigma}_2 + 2n \dot{\varsigma}_1\right) / A_{\max} \end{cases} \tag{22}$$

Substituting the reformulated systems states and inputs into the aforementioned constraints and performance criteria, the original low-thrust trajectory optimization problem for proximity relative motion is transformed into the flat output space and formulated in the framework of optimal control problem.

Problem B Determine the composite variable $\tilde{\boldsymbol{\varsigma}}(t) = \left[\boldsymbol{\varsigma}(t), \dot{\boldsymbol{\varsigma}}(t), \ddot{\boldsymbol{\varsigma}}(t)\right]^{\mathrm{T}}, t \in \left[t_0, t_f\right]$, and possibly the endpoint time t_f, that minimize the performance criterion

$$J\left(\tilde{\boldsymbol{\varsigma}}, t_0, t_f\right) = \Phi\left[\tilde{\boldsymbol{\varsigma}}\left(t_0\right), \tilde{\boldsymbol{\varsigma}}\left(t_f\right), t_0, t_f\right] + \int_{t_0}^{t_f} \Lambda\left[\tilde{\boldsymbol{\varsigma}}(t)\right] dt \tag{23}$$

and subject to the boundary constraints

$$\mathbf{E}\left(\tilde{\boldsymbol{\varsigma}}\left(t_0\right), \tilde{\boldsymbol{\varsigma}}\left(t_f\right), t_0, t_f\right) = \mathbf{0} \tag{24}$$

and path constraints on control inputs

$$\mathbf{u}_{\min} \leq \mathbf{\Gamma_u}\left(\tilde{\boldsymbol{\varsigma}}(t), t_0, t_f\right) \leq \mathbf{u}_{\max}, \quad t \in \left[t_0, t_f\right] \tag{25}$$

Since the flat output represents a minimal description of the system's behavior, the number of design variables for the relative motion trajectory optimization problem has been greatly reduced after converting into the flat space. Moreover, the flatness transformation has entirely eliminated the original differential constraints, generating an integration-free geometric programming problem, which is computationally tractable and quickly solvable. Although the feasible region in the flat output space is usually non-convex, which may affect the search for the global optimal solution, the side effects could be improved by approximating the feasible region with polytopes or superquadric surfaces [5, 6].

4 Flat Output Optimization

A general and efficient way to find the optimal flat outputs is to parameterize them and then transform Problem B into a nonlinear programming problem. Traditionally, this transformation is enabled by approximating the flat outputs with piecewise-continuous functions such as Bézier polynomials and B-splines [1, 7]. However, these schemes are proven neither most accurate nor most efficient [8]. In this section, we propose to parameterize the flat outputs by Mapped Chebyshev Pseudospectral Method (MCPM). Actually, some pseudospectral methods have been applied to optimizing the trajectories for flat systems in some earlier researches [9–11]. However, the advantages of MCPM mainly include that the approximation accuracy for the higher-order derivatives of flat outputs has been greatly enhanced by introducing the conformal map and barycentric rational interpolation techniques.

4.1 Mapped Chebyshev Pseudospectral Method

Since the mapped Chebyshev-Gauss-Lobatto (MCGL) nodes lie in the computational interval $[-1,\ 1]$, the affine transformation in Eq. (26) is used to scale the time domain $[t_0, t_f]$:

$$\tau = \left[2t - \left(t_f + t_0\right)\right] / \left(t_f - t_0\right),\ t \in \left[t_0, t_f\right] \tag{26}$$

Using the famous conformal map proposed by Kosloff and Tal-Eaer [12], the MCGL nodes are defined as

$$\lambda_k = g\left(\tau_k, \alpha\right) = \frac{\text{asin}\left(\alpha \tau_k\right)}{\text{asin}\alpha},\ (k = 0, \ldots, N) \tag{27}$$

where $0 \leq \alpha < 1$ is the map parameter that determines the degree to which the nodes are shifted toward equidistant spacing, and τ_k the standard CGL nodes whose closed form are given by

$$\tau_k = \cos\left(\pi k / N\right),\ \ k = 0, \ldots, N \tag{28}$$

Compared with other maps, the one-to-one and sufficiently smooth properties of conformal maps preserve the CPM's spectral accuracy. Another important advantage of the Kosloff-Tal-Ezer map in our research is that it improves the theoretical convergence of the interpolation approximation. In addition, to compensate for the round-off error induced by the conformal map, the parameter α is preferred to be valued as

$$\alpha = \operatorname{sech}\left(\left|\ln \varepsilon\right| / N\right) \tag{29}$$

where ε is the desired round-off error.

For the shifted interpolation nodes, the classical Lagrange interpolation technique in the standard CPM may induce the well-known Runge phenomenon, thus we propose to approximate the desired outputs with the barycentric rational interpolation formulation:

$$\varsigma_i\left(\tau\right) \approx \overline{\varsigma}_i^N\left(\tau\right) = \sum_{k=0}^{N} \frac{\omega_k^{bary} \varsigma_i\left(\lambda_k\right)}{\lambda - \lambda_k} / \sum_{k=0}^{N} \frac{\omega_k^{bary}}{\lambda - \lambda_k} \tag{30}$$

where ω_k^{bary}, $(k = 0, 1, \ldots, N)$ denotes the associated barycentric weight of λ_k:

$$\omega_0^{bary} = \frac{1}{2},\ \omega_N^{bary} = \frac{(-1)^N}{2},\ \omega_k^{bary} = (-1)^k,\ (k = 1, \ldots, N-1) \tag{31}$$

Evaluating the derivatives of flat output $\varsigma_i(\tau)$ at the mapped CGL nodes λ_k gives a matrix multiplication of the following form:

$$\begin{cases} \dot{\varsigma}_i\left(\lambda_k\right) \approx \dot{\overline{\varsigma}}_i^N\left(\lambda_k\right) = \sum\limits_{j=0}^{N} \overline{D}_{kj} \varsigma_i\left(\lambda_j\right) \\ \ddot{\varsigma}_i\left(\lambda_k\right) \approx \ddot{\overline{\varsigma}}_i^N\left(\lambda_k\right) = \sum\limits_{j=0}^{N} \overline{D}_{kj}^{(2)} \varsigma_i\left(\lambda_j\right) \end{cases} \tag{32}$$

where $\overline{D}_{kj}$ are entries of the $(N+1) \times (N+1)$ first order mapped Chebyshev differentiation matrix $\overline{\mathbf{D}}$, while $\overline{D}_{kj}^{(2)}$ the second order. The matrices are given by:

for $k \neq i$,

$$\begin{cases} \overline{D}_{ij} = \frac{\omega_j^{bary} / \omega_i^{bary}}{\lambda_i - \lambda_j} \\ \overline{D}_{ij}^{(2)} = -2 \frac{\omega_j^{bary} / \omega_i^{bary}}{\lambda_i - \lambda_j} \left[\sum\limits_{k \neq i} \frac{\omega_k^{bary} / \omega_i^{bary}}{\lambda_i - \lambda_k} - \frac{1}{\lambda_i - \lambda_j} \right] \end{cases} \tag{33}$$

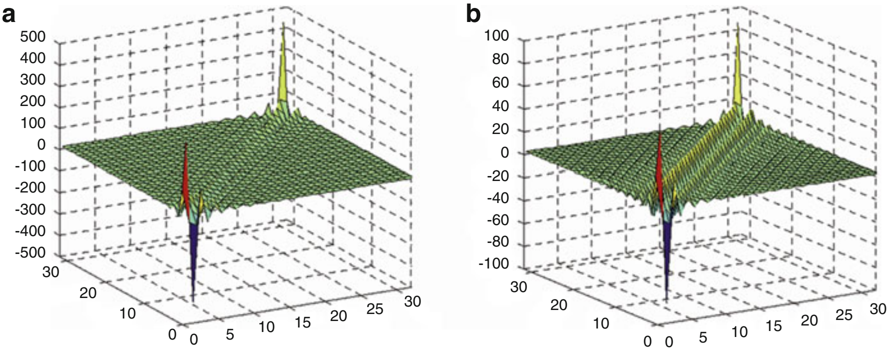


Fig. 2 Elements of Chebyshev differentiation matrix

for $k = i$,

$$\overline{D}_{ii} = -\sum_{k \neq i} \overline{D}_{ik},\ \overline{D}_{ii}^{(2)} = -\sum_{k \neq i} \overline{D}_{ik}^{(2)} \tag{34}$$

To illustrate the improvement of differentiation matrix's ill-conditioning, the distributions of both the standard and mapped differentiation matrices for $N = 32$, $\alpha = 0.99$ are presented in Fig. 2. Obviously, compared with the standard differentiation matrix, the magnitude of the elements near the diagonal endpoints has been greatly reduced, validating the validity of MCPM in enhancing the numerical accuracy for derivatives of desired outputs.

For MCGL nodes, the cost function in Eq. (23) can be reformulated as

$$J = \Phi\left[\tilde{\varsigma}(1), t_f\right] + \int_{-1}^{1} \Lambda\left(g\left(\tau\right)\right) g'\left(\tau\right) d\tau \tag{35}$$

Applying the Clenshaw-Curtis quadrature scheme to discretizing the integral part into a finite sum yields [13]

$$J \approx J^N = \Phi\left[\tilde{\varsigma}(1), t_f\right] + \sum_{k=0}^{N} \omega_k^C g'\left(\tau_k\right) \Lambda\left[g\left(\tau_k\right)\right] \tag{36}$$

where $g'(\tau)$ is the first order derivative of the conformal map in Eq. (27), and ω_k^C $(k = 0, \ldots, N)$ are Clenshaw-Curtis weights given by

for N even,

$$\begin{cases} \omega_0^C = \omega_N^C = \frac{1}{N^2-1} \\ \omega_s^C = \omega_{N-s}^C = \frac{4}{N} {\sum_{i=0}^{(N/2)}}'' \dfrac{1}{1-4i^2} \cos\dfrac{2\pi i s}{N}, \quad s = 1, \ldots, \frac{N}{2} \end{cases} \tag{37}$$

for N odd,

$$\begin{cases} \omega_0^C = \omega_N^C = \frac{1}{N^2} \\ \omega_s^C = \omega_{N-s}^C = \frac{4}{N} \sum\limits_{i=0}^{((N-1)/2)''} \dfrac{1}{1-4i^2} \cos\dfrac{2\pi i s}{N}, \quad s = 1, \ldots, \frac{N-1}{2} \end{cases} \tag{38}$$

where the double prime in the summation operations denotes that the first and last elements have to be halved.

Therefore, given the values of desired outputs at MCGL nodes $\varsigma\left(\lambda_k\right)$, $(k = 0, 1 \ldots, N)$, the system states and inputs can be reformulated as their functions, transforming Problem B into the following NLP problem:

Problem C Determine the variables $\varsigma\left(\lambda_k\right)$, $(k = 0, 1 \ldots, N)$, and possibly the endpoint time t_f, that minimize the cost function in Eq. (36), and subject to the constraints

$$\mathbf{F}\left(\varsigma\left(\lambda_k\right), t_0, t_f\right) \leq \mathbf{0} \tag{39}$$

where $\mathbf{F}\left(\cdot\right)$ represents a generalized form for the boundary and input constraints. In general, this is an NLP which can be solved by suitable algorithms or commercial packages. After obtaining the flat output solutions, the relative motion trajectory and the corresponding control inputs can be easily generated by mapping back into the original state space.

4.2 Non-Smoothness Difficulty

In practice, we found that the aforementioned computational framework presents high performance for flat system's smooth trajectory optimization. However, when directly applied to optimizing the trajectories with bang-bang type control inputs, namely non-smooth trajectories, the situation becomes much tougher. The numerical difficulties mainly include two aspects. Firstly, since the control inputs and time derivatives of states are discontinuous for the bang-bang type trajectories, the application of MCPM would inevitably yield the well-known Gibbs phenomenon, which is resulted from the approximation of a non-smooth function with a finite number of smooth functions. Secondly, the switch points for the bang-bang type inputs can't be captured exactly, fundamentally due to the predetermined distribution of collocation nodes.

One straightforward way is to divide the whole interval into segments that are free of interior discontinuities. Then for each segment, that initial trajectory optimization problem is transformed into the NLP problem by MCPM, while the relationships with neighboring segments are guaranteed by the linkage conditions [10, 14]. It should be noted that the successful application of this method greatly relies on the segment division, i.e. detecting the switching points of the bang-bang type inputs. Nevertheless, incorporating the switching detection algorithms into mesh refinement would make the solution procedure more complicated. Moreover, MCPM only enforces the constraints on the mapped CGL nodes, hence no guarantee on constraint satisfaction between neighboring collocation nodes.

In our research, the analytic homotopic approach is proposed to address the aforementioned numerical difficulties. Note that the conventional homotopic approach was first proposed to address the issues of small convergence domain and initial costates sensitivity in the indirect method for optimal control problems [15]. The key concept of homotopic approach is to construct a related, easy-handling auxiliary problem to connect with the original problem through a perturbation parameter, namely hotmotopic parameter. Starting from the auxiliary problem and varying the perturbation parameter, the solution to the original problem can be obtained by continuously calculating the associated two-point boundary value problem (TPBVP) for each perturbation parameter. The fundamentals of homotopic approach involve taking the solution to previous runs as an initial guess for the current run, yielding rapid computation of the sequences of TPBVPs. However, how to start the homotopic procedure remains a challenge, because reasonable initial guess of costates is still absent for the auxiliary problem. This is what the analytic homotopic approach addresses. Taking the proximity relative motion problem herein for example, the time-optimal low-thrust trajectory is non-smooth, while the related energy-optimal one is much smooth. Thus the related and tractable energy-optimal trajectory problem is firstly generated using the flatness and MCPM based computational framework. Then based on the energy-optimal trajectory, an auxiliary optimal control problem whose costates are naturally valued zero is constructed, eliminating the aforementioned starting challenge.

5 Numerical Continuation by Analytic Homotopic Approach

In this section, we would introduce how to construct an auxiliary optimal control problem with zero-valued costates, and how to achieve the solutions to the original optimal control problem through a continuation scheme.

5.1 Auxiliary Optimal Control Problem

Assume that the optimal trajectory for Problem A is non-smooth, and we have generated a related but smooth trajectory $\{\mathbf{x}_A(t), \mathbf{u}_A(t), t_{fA}\}$. The subscript 'A' here means auxiliary trajectory for constructing the auxiliary optimal control problem whose costates are simply zero. Since the auxiliary trajectory exactly observes the boundary conditions of the original optimal control problem, the auxiliary optimal control problem can be formulated as follows.

Problem D Determine the state-input pair $\{\mathbf{x}(t) \in \mathbb{R}^n, \mathbf{u}(t) \in \mathbb{R}^m\}, t \in \left[t_0, t_f\right]$, and the endpoint time t_f that minimize the performance criterion:

$$\begin{aligned} J_A(\mathbf{u}, t) &= \Phi_A\left(\mathbf{x}(t_0), \mathbf{x}\left(t_f\right), t_0, t_f\right) + \int_{t_0}^{t_f} \Lambda_A(\mathbf{x}(t), \mathbf{u}(t), t)\, dt \\ &= \tfrac{1}{2}\left(t_f - t_{fA}\right)^2 + \tfrac{1}{2}\int_{t_0}^{t_f} \left|\mathbf{u}(t) - \mathbf{u}_A\left(t \cdot t_{fA}/t_f\right)\right|^2 dt \end{aligned} \tag{40}$$

and subject to the governing equations

$$\dot{\mathbf{x}}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{41}$$

the boundary conditions

$$\mathbf{x}(t_0) = \mathbf{x}_0,\ \mathbf{x}\left(t_f\right) = \mathbf{x}_f \tag{42}$$

and constraints on controls

$$\mathbf{u}_{\min} \leq \mathbf{u}(t) \leq \mathbf{u}_{\max} \tag{43}$$

It is obvious that the performance criterion gets its minimum value $J_{A\min} = 0$ at the given auxiliary trajectory, i.e. the optimal solution to Problem D is $\{\mathbf{x}_A(t), \mathbf{u}_A(t), t_{fA}\}$. This conclusion can be analyzed as follows.

The Hamiltonian function of Problem D is given by:

$$\begin{aligned} H_A &= \tfrac{1}{2}\left|\mathbf{u}(t) - \mathbf{u}_A\left(t \cdot t_{fA}/t_f\right)\right|^2 + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{g}(\mathbf{x}, \mathbf{u}) \\ &= \sum_{i=1}^{m} \frac{1}{2}\left(u_i(t) - u_{iA}\left(t \cdot t_{fA}/t_f\right)\right)^2 + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{g}(\mathbf{x}, \mathbf{u}) \end{aligned} \tag{44}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^n$ denotes the costate vector. The first-order necessary optimality condition can be derived as

$$\frac{\partial H_A}{\partial \mathbf{u}} = \left(\mathbf{u}(t) - \mathbf{u}_A\left(t \cdot t_{fA}/t_f\right)\right)^{\mathrm{T}}\mathbf{I}_m + \boldsymbol{\lambda}^{\mathrm{T}}\frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{u}} = \mathbf{0} \tag{45}$$

where $\mathbf{I}_m$ is a $m \times m$ unit matrix.

The associated costate equation is

$$\dot{\boldsymbol{\lambda}}^{\mathrm{T}} = -\frac{\partial H_A}{\partial \mathbf{x}} = -\boldsymbol{\lambda}^{\mathrm{T}}\frac{\partial \mathbf{g}}{\partial \mathbf{x}} \tag{46}$$

with the transversality condition

$$\boldsymbol{\lambda}\left(t_f\right) = \left.\frac{\partial \Phi_A}{\partial \mathbf{x}}\right|_{t_f} = \mathbf{0} \tag{47}$$

Considering the free endpoint time, the Hamiltonian function H_A at t_f is written as

$$H_A\left(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}, t\right)\big|_{t_f} = -\left.\frac{\partial \Phi_A}{\partial t}\right|_{t_f} = -\left(t_f - t_{fA}\right) \tag{48}$$

Since the state-control pair $\{\mathbf{x}_A(t), \mathbf{u}_A(t)\}$ is the optimal solution to Problem D, it is easy to find that the costate vector $\boldsymbol{\lambda} = \mathbf{0}$ satisfies these optimality conditions in Eqs. (45)–(47). Moreover, the Hamilton function H_A evaluates zero when the costate vector is valued zero. Thus according to the transversality condition in Eq. (48), it can be concluded that $t_f = t_{fA}$ is the optimal endpoint time.

5.2 Continuation Toward the Original Problem

Note that the governing equations and constraints of Problem D, i.e. auxiliary optimal control problem, are consistent with those in Problem A, thus these two problems can be related by introducing the so-called homotopic parameter ε to reformulate the performance criterion as:

$$J_\varepsilon\left(\mathbf{x}, \mathbf{u}, t\right) = \Phi_\varepsilon\left[\mathbf{x}\left(t_0\right), \mathbf{x}\left(t_f\right), t_0, t_f\right] + \int_{t_0}^{t_f} \Lambda_\varepsilon\left[\mathbf{x}(t), \mathbf{u}(t), t\right] dt \tag{49}$$

where

$$\begin{cases} \Phi_\varepsilon\left[\mathbf{x}\left(t_0\right), \mathbf{x}\left(t_f\right), t_0, t_f\right] = (1-\varepsilon)\, \Phi\left[\mathbf{x}\left(t_0\right), \mathbf{x}\left(t_f\right), t_0, t_f\right] + \varepsilon \Phi_A\left[\mathbf{x}\left(t_0\right), \mathbf{x}\left(t_f\right), t_0, t_f\right] \\ \Lambda_\varepsilon\left[\mathbf{x}(t), \mathbf{u}(t), t\right] = (1-\varepsilon)\, \Lambda\left[\mathbf{x}(t), \mathbf{u}(t), t\right] + \varepsilon \Lambda_A\left[\mathbf{x}(t), \mathbf{u}(t), t\right] \end{cases} \tag{50}$$

Obviously, the parameter $\varepsilon = 1$ corresponds to Problem D, and $\varepsilon = 0$ to Problem A. Thus starting from the solution to Problem D, the non-smooth solutions of Problem A could be obtained by homotopic algorithm.

By varying the homotopic parameter from 1 to 0, a series of optimal control problems associated with each J_ε are defined, and the corresponding TPBVPs can be developed by applying the Pontryagin principle. To solve these TPBVPs with the simple shooting method, the first run of the homotopic procedure, i.e. $\varepsilon = 1$, is initialized with the solutions to the auxiliary optimal control problem, namely Problem D, whereas the subsequent steps use the solution to the previous runs as initialization. Actually, this is the key idea underlying the analytic homotopic algorithm.

It should be noted that as the homotopic parameter ε approaches zero, the optimal controls become less and less smooth, inducing trouble for the integration accuracy of classical integrator with adaptive step size. To address this problem, the fourth-order Runge–Kutta algorithm with fixed step size combined with switching detection is proposed in [16]. This integration algorithm is effective and easy to achieve for systems with small number of inputs, while much more complicated for multi-input systems. Alternatively, Li suggested utilizing the *events* checking function of *ode45*, a MATLAB embedded propagator, to capture the switching time, and then changing the controls according to the corresponding switching function [17].

6 Numerical Simulation

In this section, two numerical examples, an energy-optimal trajectory planning problem with the performance criterion in Eq. (18) and a time-optimal one with the criterion in Eq. (17), are presented to validate the performance of the proposed computational framework. Just as discussed before, the in-plane motion and out-of-plane motion are studied separately here.

6.1 Energy-Optimal Trajectory Planning

The leader is assumed to be moving in a circular orbit at an altitude of 1000 *km*, and the associated mean angular velocity n is 9.9621×10^{-4} *rad/s*. The maximum control acceleration on the follower $A_{\max}$ is set to be 6.0×10^{-5} *m/s*^{2}, and the constraints on control inputs take the form of $-1 \leq u_i \leq 1$. Note that the endpoint time t_f is fixed at 5000 s for the energy-optimal trajectory planning problem. Let $\boldsymbol{\rho} = [x_{\mathrm{H}}, y_{\mathrm{H}}, z_{\mathrm{H}}]^{\mathrm{T}}$ denote the relative position vector, and then the initial and terminal relative states are given by

$$\boldsymbol{\rho}_0 = [160, -300, -90]^{\mathrm{T}} m, \quad \dot{\boldsymbol{\rho}}_0 = [-0.15, -0.3, \ 0.1]^{\mathrm{T}} m/s$$

$$\boldsymbol{\rho}_f = [0, \ 150, \ 10]^{\mathrm{T}} m, \quad \dot{\boldsymbol{\rho}}_f = [0, \ 0, \ 0]^{\mathrm{T}} m/s$$

According to the analysis in Sect. 3.2, the governing equations for the in-plane relative motion are differentially flat. Thus the associated energy-optimal trajectory is readily generated under the framework of flatness and MCPM. The parameters N and β of MCPM are valued as 32 and 1.0×10^{-12} respectively, resulting in a conformal map parameter α of 0.7161. The MATLAB *fmincon* function is applied to the NLP problem with all the parameters 'TolFun', 'TolCon' and 'TolX' set to be

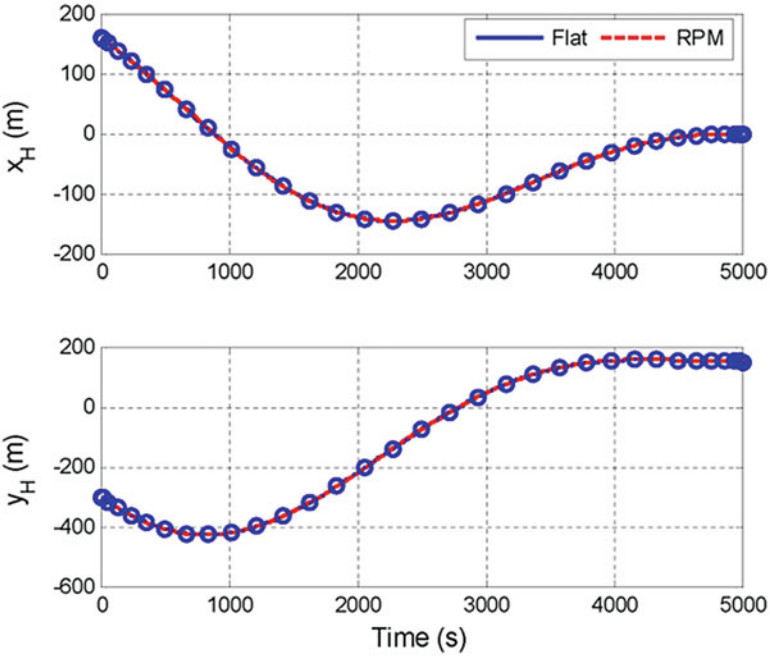


Fig. 3 Time histories of relative position (in-plane)

1.0×10^{-12}. The time histories of system states and control inputs are presented in Figs. 3, 4 and 5, where the *circles* denote the corresponding variables at the mapped CGL nodes. The energy-optimal trajectory for in-plane motion is smooth, and the control inputs observe the saturation constraints. To evaluate the results of our research, Radau pseudospectral method (RPM) is also applied to this problem by utilizing a modified version of the open source GPOPS package [14]. This package has been widely used in various fields due to its excellent performance. The field of 'autoscale' in GPOPS is set 'on' to invoke the automatic scaling routine, while the 'mesh refinement' option is set by default to accurately distribute the collocation nodes. The 'RPM' trajectories are also illustrated in Figs. 3–5 by *dashed lines* for comparison. The differences between the 'Flat' and 'RPM' trajectories are very slight, and both the associated performance indexes are 1237.1 s, validating the feasibility and validity of the flatness based framework for smooth trajectories. All simulations were performed on a standard desktop PC with a 2.80-GHz processor. The CPU time required for the flatness-based framework is about 4.2 s, while 4.7 s for the RPM approach. Note that the CPU time herein only provides a reference, because the computational efficiency mainly depends on the scale and solver of the NLP problem. In GPOPS package, the NLP problem scale is evidently larger, because the discretization mesh is refined repeatedly to generate trajectories in higher accuracy. On the other hand, the commercial solver SNOPT is utilized in

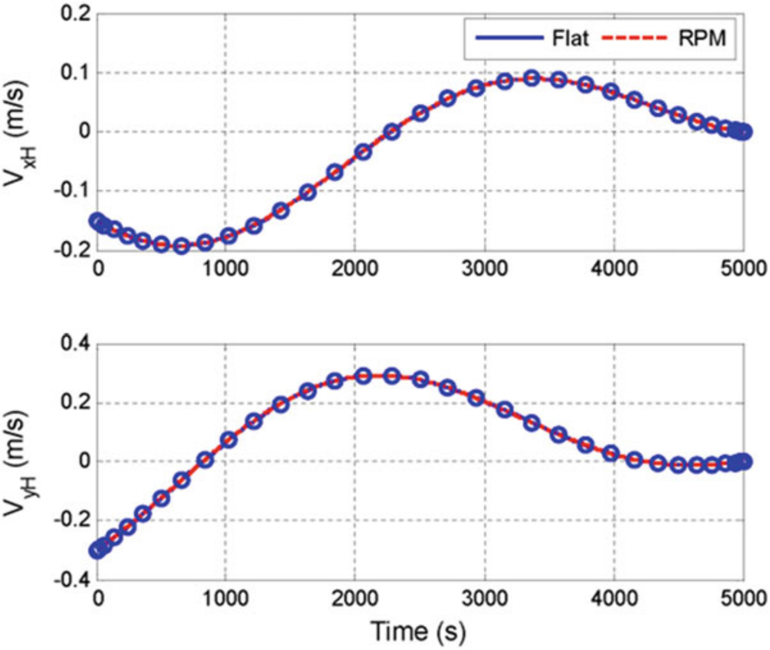


Fig. 4 Time histories of relative velocity (in-plane)

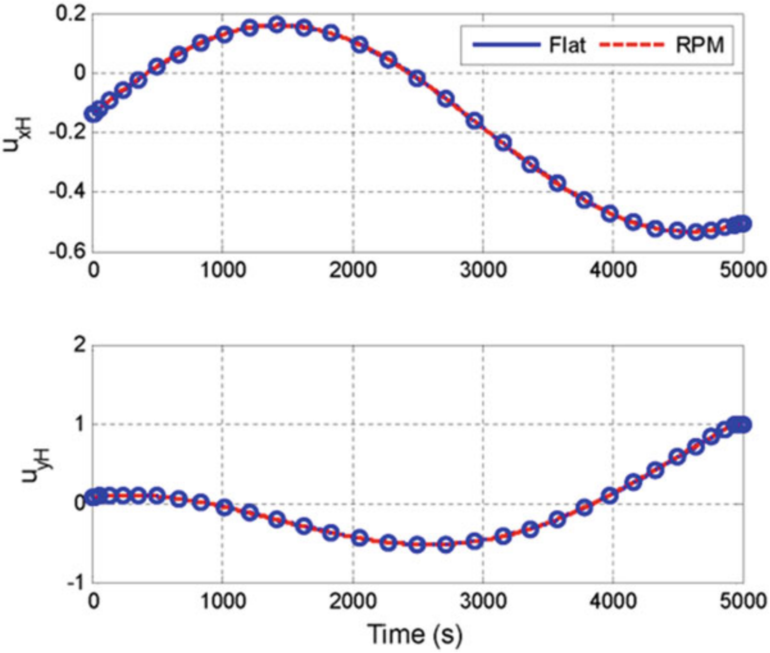


Fig. 5 Time histories of control inputs (in-plane)

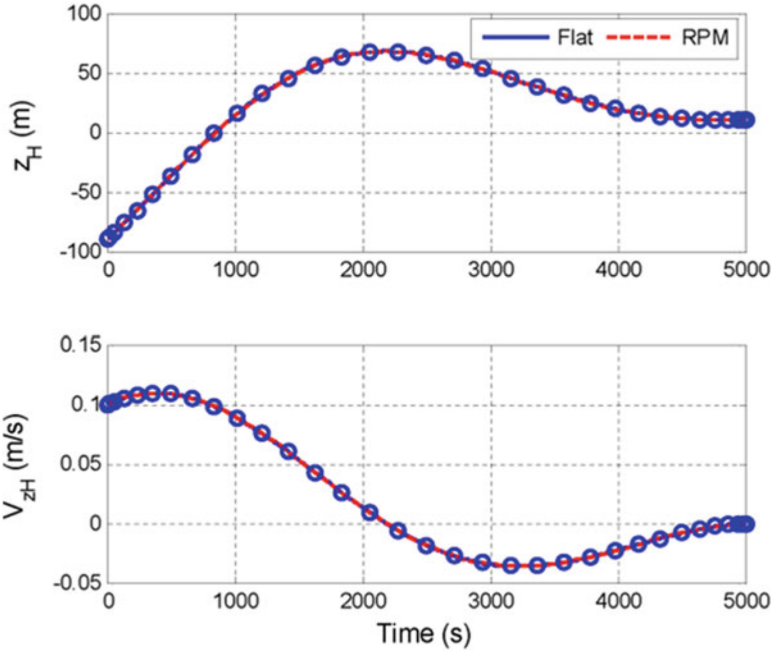


Fig. 6 Time histories of relative states (out-of-plane)

GPOPS, whereas the embedded *fmincon* function for the flatness-based framework. For the out-of-plane relative motion, the same computation procedure is conducted, and the corresponding energy-optimal trajectories are presented in Figs. 6 and 7.

6.2 Time-Optimal Trajectory Planning

Let the configuration parameters of the time-optimal trajectory planning problem keep exactly the same as those for energy-optimal trajectory planning, except for the free endpoint time. For the in-plane relative motion, the time-optimal trajectory is firstly generated utilizing the flatness based approach, and the time histories of control inputs are presented in Figs. 8 and 9. It can be seen that the obvious Gibbs phenomenon exists, and the switching points cannot be identified. Thus the analytic homotopic approach is needed to eliminate these numerical difficulties, and gradually achieving the time-optimal trajectory from the aforementioned energy-optimal solution.

Taking the in-plane motion for example, the energy-optimal trajectory based auxiliary optimal control problem could be readily written in the form of Problem D. The detailed formulations have been omitted here for compactness, and only the associated TPBVP for some specific homotopic parameters is presented here.

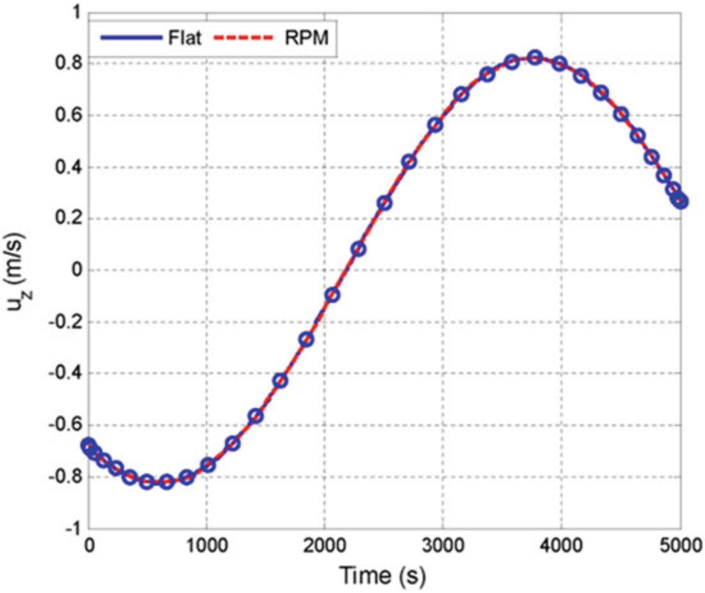


Fig. 7 Time histories of control inputs (out-of-plane)

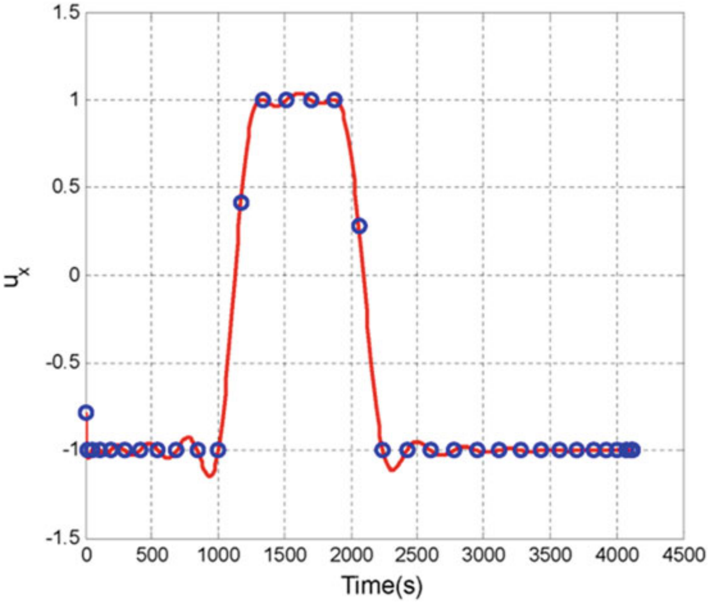


Fig. 8 Time histories of u_x

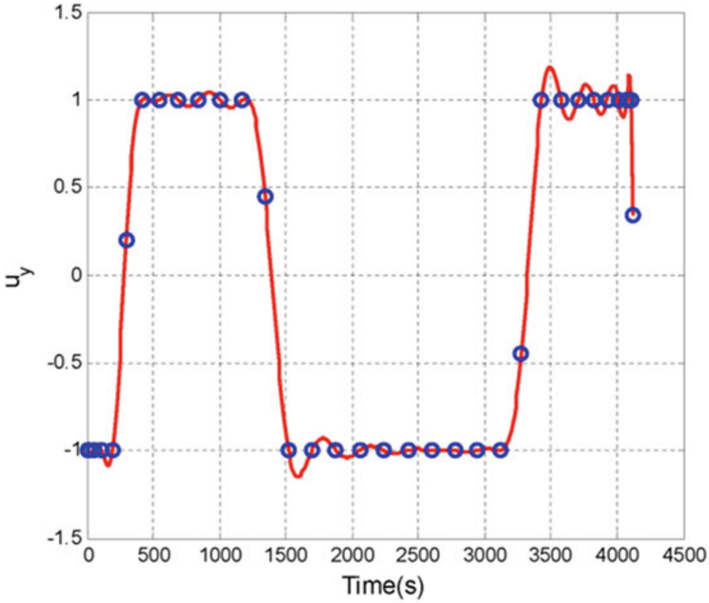


Fig. 9 Time histories of u_y

For a given homotopic parameter ε, the corresponding Hamilton function in the homotopic procedure is written as

$$\begin{aligned} H_\varepsilon = \tfrac{\varepsilon}{2}\left((u_1 - u_{1A})^2 + (u_2 - u_{2A})^2\right) + \lambda_1 x_3 + \lambda_2 x_4 + \ldots \\ \lambda_3\left(3n^2 x_1 + 2nx_4 + A_{\max} u_1\right) + \lambda_4\left(-2nx_3 + A_{\max} u_2\right) \end{aligned} \tag{51}$$

where $x_i, (i = 1, 2 \cdots 4)$ and $u_i, (i = 1, 2)$ are defined as those in Eq. (21), and the subscript ‘A’ denotes the corresponding variables of the energy-optimal trajectory.

Thus the costate equations are given by

$$\begin{cases} \dot{\lambda}_1 = -3n^2\lambda_3 \\ \dot{\lambda}_2 = 0 \\ \dot{\lambda}_3 = -\lambda_1 + 2n\lambda_4 \\ \dot{\lambda}_4 = -\lambda_2 - 2n\lambda_3 \end{cases} \tag{52}$$

The necessary optimality condition $\partial H_\varepsilon / \partial \mathbf{u} = \mathbf{0}$ yields the optimal inputs:

$$u_i = \begin{cases} u_{\min}, & s_i \le u_{\min} \\ u_{\max}, & s_i \ge u_{\max} \\ s_i, & \text{else} \end{cases} \tag{53}$$

Table 1 Results of homotopic procedure (in-plane motion)

ε	Iter.	T_f	Initial value of costates			
1	0	5000	0	0	0	0
0.6	3	4999.93	-1.7464×10^{-4}	1.1657×10^{-5}	−0.0955	−0.0698
0.2	3	4999.50	9.3353×10^{-4}	-1.4464×10^{-4}	0.5600	0.2897
0.1	3	4998.87	0.0020	-3.0085×10^{-4}	1.2142	0.6482
0.01	6	4987.68	0.0023	-3.3988×10^{-4}	1.3770	0.7370
10^{-3}	9	4889.61	0.0025	-3.7491×10^{-4}	1.5103	0.8041
10^{-4}	13	4408.42	0.0024	-3.8437×10^{-4}	1.4686	0.7425
10^{-5}	13	4128.10	0.0021	-4.4967×10^{-4}	1.3324	0.5047
10^{-6}	7	4114.20	0.0020	-5.1285×10^{-4}	1.1480	0.4179
10^{-7}	23	4114.05	0.0019	-5.1062×10^{-4}	1.1025	0.3941
10^{-8}	16	4114.05	0.0019	-5.1063×10^{-4}	1.1003	0.3925

where $s_i = u_{iA} - \lambda_{i+2} A_{\max}/\varepsilon,\ (i = 1,\ 2)$ is the switching function.

The boundary and trans-versatility conditions are

$$\mathbf{x}(t_0) = \mathbf{x}_0,\ \mathbf{x}\left(t_f\right) = \mathbf{x}_f \tag{54}$$

$$\boldsymbol{\lambda}\left(t_f\right) = \left[\partial\Phi_\varepsilon/\partial\mathbf{x} + \left(\partial N_1^{\mathrm{T}}/\partial\mathbf{x}\right)\boldsymbol{\gamma}\right]_{tf} \tag{55}$$

And the Hamilton function at endpoint time observes

$$H^*\left(t_f^*\right) = -\left[\partial\Phi_\varepsilon/\partial t + \boldsymbol{\gamma}^{\mathrm{T}}\left(\partial N_1/\partial\mathbf{x}\right)\right]_{tf} \tag{56}$$

The TPBVP for homotopic parameter ε is addressed utilizing the simple shooting method, where the resulted nonlinear shooting equations are solved by the MATLAB *fsolve* function with the option parameters ‘TolX’, ‘TolFun’ and ‘TolCon’ set as 1.0×10^{-12}. The results of the associated TPBVPs during the homotopic procedure are listed in Table 1 and the continuous achieving process of controls and states are illustrated in Figs. 10, 11, 12 and 13 by *dotted lines*. Obviously, by initializing the shooting method with the results of the previous run, the solutions can be obtained in a small number of iteration steps. The final result 4114.05 s for $\varepsilon = 10^{-8}$ is accurate enough to be regarded as the minimum time solution, whose corresponding profiles of controls and states are presented in Figs. 10, 11, 12 and 13 by *solid lines*. It can be seen that the Gibbs phenomenon of MCPM solution has been entirely eliminated. Moreover, utilizing the same computational framework, the time-optimal trajectory for the out-of-plane motion could be easily obtained with the achieving process of controls and states illustrated in Figs. 14 and 15. The minimum time required to accomplish the out-of-plane maneuver is 3333.66 s, and the associated switching points of the control inputs are given in Table 2.

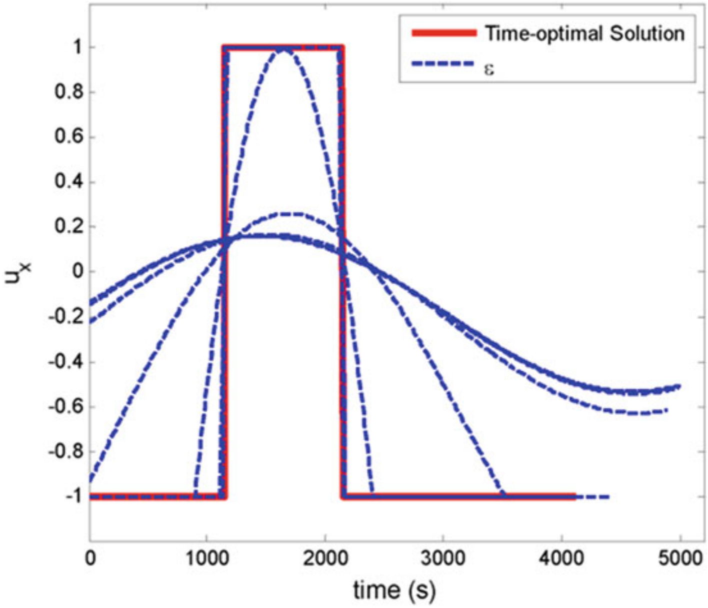


Fig. 10 Homotopy procedure of u_x

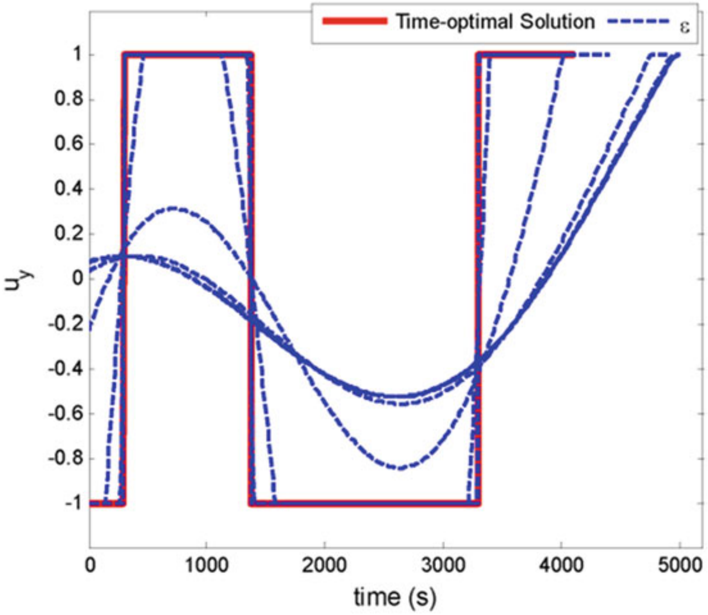


Fig. 11 Homotopy procedure of u_y

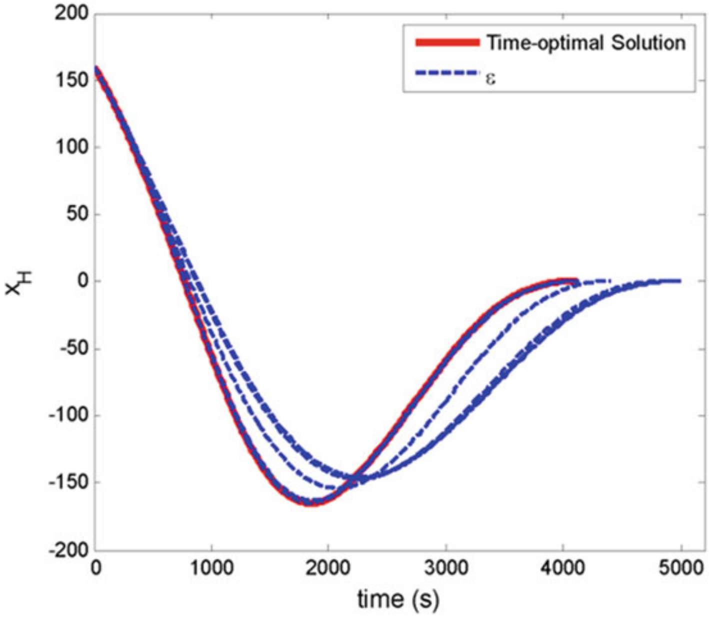


Fig. 12 Homotopy procedure of x_{H}

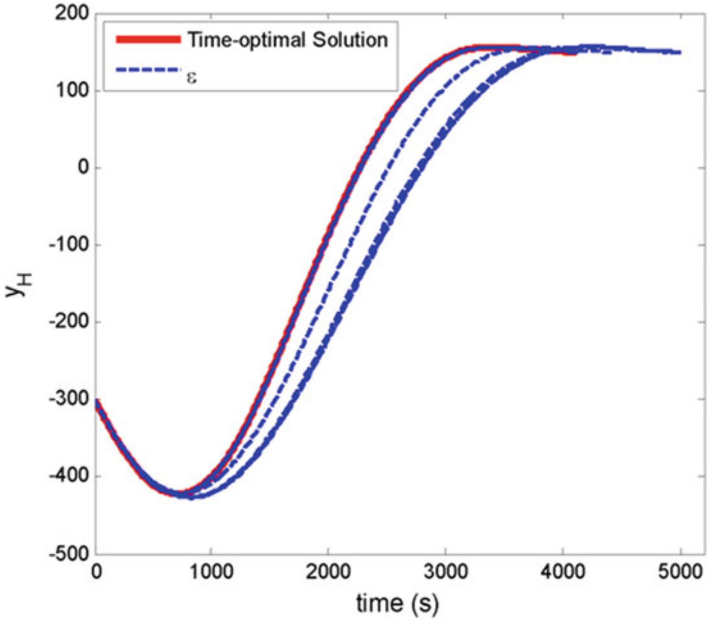


Fig. 13 Homotopy procedure of y_{H}

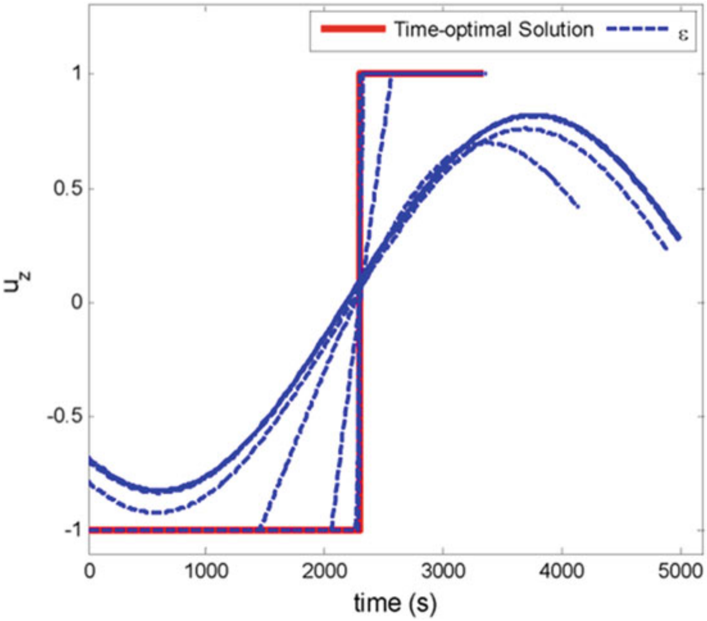


Fig. 14 Homotopy procedure of u_z

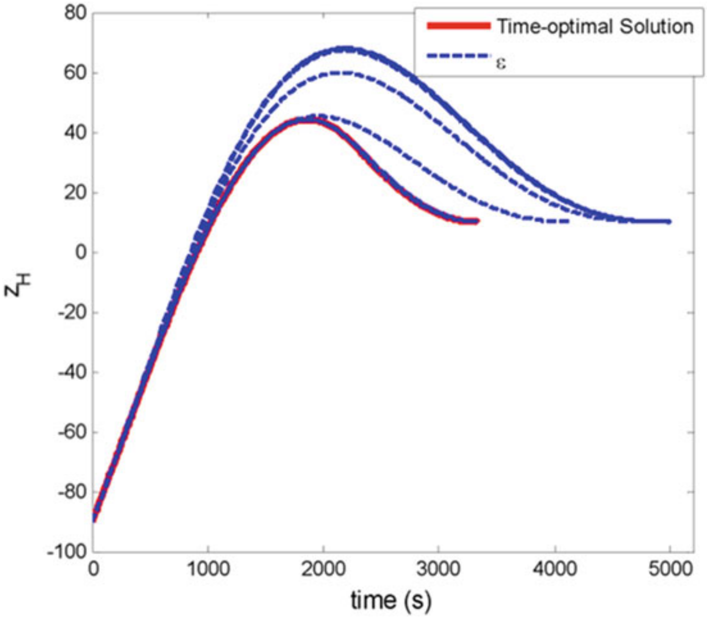


Fig. 15 Homotopy procedure of z_{H}

Table 2 Switching points

Control	Point 1	Point 2	Point 3
u_x (s)	1146.79	2144.45	–
u_y (s)	293.13	1373.06	3296.38
u_z (s)	2296.06	–	–

7 Conclusions

Against the background of on-orbit proximity operations missions, we studied a hybrid computational framework which is based on the flatness theory and analytic homotopic approach to optimize the low-thrust relative motion trajectory. Some useful conclusions are drawn as follows:

1. For differentially flat systems, the associated optimal control problem could be reformulated in the flat output space with the reduction of decision variables and elimination of the differential constraints.
2. For the smooth trajectory optimization problems, the MCPM based flat output discretization approach presents excellent performance, where the approximation accuracy for the derivatives of flat outputs is evidently improved by conformal map and barycentric rational interpolation techniques.
3. For the non-smooth trajectory optimization problems, the numerical difficulties of the MCPM based flat output discretization approach can be eliminated by introducing the analytic homotopic technique. On the other hand, this combined framework also addresses the initialization difficulty of the conventional homotopic approach.

References

1. Chamseddine, A., Zhang, Y., Rabbath, C.A., Join, C., Theilliol, D.: Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle. IEEE Trans. Aerosp. Electron. Syst. **48**, 2832–2848 (2012)
2. Levine, J.: Analysis and Control of Nonlinear Systems. Springer, Berlin and Heidelberg (2009)
3. Fornberg, B.: A Practical Guide to Pseudospectral Methods. Cambridge University Press, New York (1998)
4. de Ruiter, A.H., Damaren, C.J., Forbes, J.R.: Spacecraft Dynamics and Control: An Introduction. Wiley, West Sussex, UK (2013)
5. Faiz, N., Agrawal, S.K., Murray, R.M.: Trajectory planning of differentially flat systems with dynamics and inequalities. J. Guid. Control. Dyn. **24**(2), 219–227 (2001)
6. Morio, V., Cazaurang, F., Zolghadri, A., Vernis, P.: Onboard path planning for reusable launch vehicles application to the shuttle orbiter reentry mission. Int Rev Aerosp Eng **1**(6), 492–503 (2008)
7. Louembet, C., Cazaurang, F., Zolghadri, A.: Motion planning for flat systems using positive B-splines: an LMI approach. Automatica **46**(8), 1305–1309 (2010)
8. Qi, G., Kang, W., Ross, I.M.: A pseudospectral method for the optimal control of constrained feedback linearizable systems. IEEE Trans. Autom. Control **51**(7), 1115–1129 (2006)

9. Desiderio, D., Lovera, M.: Flatness-based guidance for planetary landing. 2010 American control conference, Marriott Waterfront, Baltimore, MD, 2010, pp. 3642–3647.
10. Ross, I.M., Fahroo, F.: Pseudospectral methods for optimal motion planning of differentially flat systems. IEEE Trans. Autom. Control **49**(8), 1410–1413 (2004)
11. Zhuang, Y., Ma, G., Li, C., Huang, H.: Time-optimal trajectory planning for underactuated rigid spacecraft using differential flatness. J. Astronaut. **32**(8), 1753–1761 (2011)
12. Kosloff, D., Tal-Ezer, H.: A modified Chebyshev pseudospectral method with an O(N-1) time step restriction. J. Comput. Phys. **104**(2), 457–469 (1993)
13. Fahroo, F., Ross, I.M.: Direct trajectory optimization by a Chebyshev pseudospectral method. J. Guid. Control. Dyn. **25**, 160–166 (2002)
14. Darby, C.L., Hager, W.W., Rao, A.V.: An hp-adaptive pseudospectral method for solving optimal control problems. Optim. Control Appl. Methods **32**(4), 476–502 (2011)
15. Bertrand, R., Epenoy, R.: New smoothing techniques for solving bang-bang optimal control problems—numerical results and statistical interpretation. Optim. Control Appl. Methods **23**, 171–197 (2002)
16. Jiang, F., Baoyin, H., Li, J.: Practical techniques for low-thrust trajectory optimization with homotopic approach. J. Guid. Control. Dyn. **35**, 245–258 (2012)
17. Li, J., Xi, X.: Fuel-optimal low-thrust reconfiguration of formation-flying satellites via homopotic approach. J. Guid. Control. Dyn. **35**, 1709–1717 (2012)

Index

G. Fasano, J.D. Pintér (eds.), *Space Engineering*, Springer Optimization and Its Applications 114, DOI 10.1007/978-3-319-41508-6