SAPIENZA
UNIVERSITÀ DI ROMA

Robust trajectory optimization with chance-constrained formulation

Facoltà di Ingegneria Civile e Industriale
Dipartimento di Ingegneria Meccanica e Aerospaziale

Dottorato di Ricerca in Ingegneria Aeronautica e Spaziale – XXXVI Ciclo

Candidate

Nicola Marmo
ID number 1473728

Thesis Advisor

Dr. Alessandro Zavoli

2023/2024

Thesis defended on 26 September 2024
in front of a Board of Examiners composed by:

Prof. Sergio Pirozzoli (chairman)

Prof. Giulio Avanzini

Prof. Christian Paravan

Prof. Marco Zannoni

Robust trajectory optimization with chance-constrained formulation
Ph.D. thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Version: 6 December 2024

Author's email: nicola.marmo@uniroma1.it

"Les questions les plus importantes de la vie ne sont en effet, pour la plupart, que des problèmes de probabilité."
Pierre-Simon Laplace

Abstract

This thesis investigates the chance-constraint formulation of robust trajectory optimization for space applications. The importance of robustness in space trajectory optimization is growing significantly as the modern mission design approach prioritizes lightweight satellite architectures and cost reduction. However, this emphasis also raises the risk of deviating from a nominal path due to errors in navigation or incorrect maneuvers. Consequently, it is crucial to compute control laws that directly incorporate quantitative information about uncertainty in system dynamics and stochastic navigation errors during the optimization process. This works aims to formulate a stochastic framework for trajectory optimization with the goal of increasing intrinsic robustness of space flight, in both impulsive and low-thrust transfers. Two different approaches are considered in this regard: open-loop control, assessing the possibility of reducing the state dispersion by optimizing a sequence of deterministic maneuvers, and closed-loop control, defining a linear feedback control law that computes the corrective maneuver during the flight, limiting in this way the state dispersion. In order to deal with the stochastic formulation of the state, chance-constraint theory is employed when expressing the cost function and the constraints on control magnitude. As assessments of the validity and performance capability of the proposed robust trajectory optimization methodologies, test cases involving potential space applications are employed, involving interplanetary transfers and station-keeping control strategies.

Aknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Alessandro Zavoli, for his invaluable help and contribution to this Ph.D. thesis. His expert guidance and insightful feedback have been crucial in shaping the direction and quality of this work. I am also sincerely thankful for his patience and mentorship, as this accomplishment would not have been possible without his exceptional supervision and commitment.

I would also like to extend my heartfelt thanks to Dr. Naoya Ozaki for hosting me at JAXA and providing an enriching and vibrant experience during my stay in Japan. His hospitality and support were essential in making my time there both productive and enjoyable. Dr. Ozaki provided me numerous opportunities to meet many interesting people and students, significantly enhancing my professional and personal growth. I am deeply grateful for his kindness and generosity.

Nomenclature

The list below explains the symbols, acronyms, and abbreviations used throughout this document. In instances where the meaning differs from that stated in this list, the specific definition will be clarified in the text.

Acronyms and abbreviations

ACO	Ant colony optimization
AI	Artificial intelligence
BC	Behavioural cloning
BVP	Boundary value problem
CL	Closed loop
CR3BP	Circular restricted three-body problem
CUT	Conjugate unscented transform
DA	Differential algebra
DE	Differential evolution
DL	Deep learning
DRL	Deep reinforcement learning
EA	Evolutionary algorithm
EoM	Equations of motion
ESA	European space agency
FPE	Fokker-Plank equation
GA	Genetic algorithm
GMM	Gaussian mixture model
GSF	Generic stochastic formulation
HML	Human landing system
IPT	Interplanetary transfer
IRK	Implicit Runge-Kutta

JAXA Japanese aerospace exploration agency

JPL Jet propulsion laboratory

LG Lunar gateway

MC Monte Carlo

MDP Markov decision process

MFB Moon flyby

ML Machine learning

MPBVP Multi-point boundary value problem

MPC Model predictive control

MTE Missed thrust event

NASA National aeronautics and space administration

NE Navigation error

NLP Non-linear programming

NN Neural network

NRHO Near-rectilinear halo orbit

OCP Optimal control problem

OD Orbit determination

ODE Ordinary differential equation

OL Open loop

PCE Polynomial chaos expansion

PDF Probability density function

PE Propulsion error

PMP Pontryagin maximum principle

POI Planetary orbit injection

PSO Particle swarm optimization

RF Reference frame

RNN Recurrent neural network

ROCP Robust optimal control problem

RPOD Rendezvous, proximity operations, and docking

RSO Resident space object

SDE	Stochastic differential equation
SM	Stochastic manipulation
SOCP	Stochastic optimal control problem
SOR	Spiral orbit-raising
SP	Sigma Point
SQP	Sequential quadratic programming
SSA	Space situational awareness
STM	State transition matrix
ToF	Time-of-flight
UP	Uncertainty propagation
UT	Unscented transform

Subcripts

$+, -$	Right after or before the impulsive maneuver, respectively
0	First value
$\oplus$	Related to the Earth
$\leftmoon$	Related to the Moon
$\odot$	Related to the Sun
f	Final value
i	Value of the i-th leg
k	Value at the k-th node
w	Related to the white noise perturbation
max	Maximum value
MC	Related to Monte Carlo analysis
tot	Total value

Superscripts

(i)	Value of the i-th segment
$*$	Optimal value
0	Related to the first value
$\dot{}$	Derivation
$\mathcal{T}$	Related to thrust vector

$\bar{\ }$	Mean value
$'$	Vector with only position and velocity values
$\tilde{\ }$	User-defined parameter
r, v, m	Related to position, velocity, or mass, respectively
T	Transpose
u	Related to control
conv	Reference value

Values and symbols

α, β	Constraint violation probabilities
$\mathbf{0}_n$	Vector of zeros with dimension n
$\mathbf{0}_{n\times m}$	Matrix of zeros with dimension $n \times m$
$\mathbf{1}_n$	Vector of ones with dimension n
$\boldsymbol{\eta}$	Observables noise vector
$\boldsymbol{\mu}$	Mean state
$\boldsymbol{\nu}$	Deterministic component of $\boldsymbol{u}$
$\boldsymbol{\Theta}$	Optimization variables vector
$\boldsymbol{B}$	Control vector mapping matrix
$\boldsymbol{G}$	Stochastic perturbation matrix
$\boldsymbol{H}$	Observables vector mapping matrix
$\boldsymbol{I}_n$	Identity matrix of dimension n
$\boldsymbol{K}$	Gain matrix of stochastic component of $\boldsymbol{u}$
$\boldsymbol{P}$	State or control covariance matrix
$\boldsymbol{Q}$	Instantaneous additive noise matrix
$\boldsymbol{R}$	Observables noise covariance matrix
$\boldsymbol{r}$	Position vector
$\boldsymbol{u}$	Control vector
$\boldsymbol{v}$	Velocity vector
$\boldsymbol{w}$	Noise vector
$\boldsymbol{x}$	State vector
$\boldsymbol{y}$	Observables vector

$\Delta \boldsymbol{v}$	Impulsive velocity variation
ΔT	Time of flight
Δt	Time interval
$\mathbb{E}[x]$	Expected value of x
μ	Gravitational constant
$\mathrm{Cov}[x]$	Covariance of x
$\Pr\{X\}$	Probability of event X
σ	Standard deviation
J	Merit index
L	Number of legs
N	Number of arcs
n	Problem dimension
t	Time

List of Figures

List of Tables

Contents

Chapter 1

Introduction

SPACE trajectory optimization is a complex task that demands a careful balance of engineering, mathematics, and physics. In the vast expanse of outer space, guiding a spacecraft from one point to another requires meticulous consideration of numerous factors, including gravitational forces, propulsion system capabilities, and mission objectives. As space exploration endeavors become increasingly ambitious, from interplanetary missions to lunar landings and beyond, innovative trajectory optimization techniques are essential. Meeting these challenges necessitates sophisticated mathematical models, advanced optimization algorithms, and a profound understanding of both the physical principles governing spaceflight and the practical constraints of spacecraft engineering.

Commonly adopted methods for designing space trajectories typically center around the computation of a control law aimed at optimizing a specified performance index, such as minimizing propellant consumption, all while ensuring adherence to a predefined set of deterministic constraints. However, the challenges inherent in space exploration extend beyond deterministic constraints: uncertainties and disturbances play a pivotal role in actual trajectory execution. Throughout the intricate journey of space exploration, probes encounter a plethora of dynamic perturbations and uncertainties that can significantly deviate their trajectory from the intended path, thereby potentially jeopardizing mission success. These perturbations arise from diverse sources, including gravitational effects from celestial bodies, solar radiation pressure, atmospheric drag, and uncertainties in the spacecraft's dynamics and environmental conditions. Furthermore, uncertainties in navigation measurements compound the complexity of the situation. Collectively, these factors contribute to the formation of deviations, ranging from minor detours to substantial displacements from the nominal trajectory. Such deviations pose significant risks to mission success, as they may compromise critical mission objectives, scientific observations, or the safety of the spacecraft itself. Therefore, effectively mitigating the impact of these perturbations and uncertainties on trajectory deviation stands as a paramount endeavor for ensuring the effectiveness and longevity of space missions.

To mitigate trajectory deviations and minimize the risks of mission failure, spacecraft designers typically allocate propellant margins for correction maneuvers, thereby ensuring a certain level of robustness throughout the mission. Traditionally, the determination of these propellant margins occurs through a retrospective process known as *a posteriori* estimation. This method unfolds after establishing a nominal trajectory and relies on an iterative approach using Monte Carlo (MC) simulations. However, aside from its substantial computational demands, this approach may not comprehensively capture all critical circumstances that could potentially lead to mission failures. Consequently, this limitation poses additional challenges for

mission designers, who must navigate uncertainties while striving to guarantee mission success. This underscores the urgent need for more efficient and nuanced methodologies capable of accurately anticipating and addressing potential deviations, thereby fortifying the resilience and efficacy of space missions. The latest trend in research within this field is the direct integration of position and velocity uncertainties into the optimization process, precisely tailoring in this way a control law for the specific mission scenario, ensuring robust trajectory design. Adopting this modern perspective, the thesis introduces a methodology designed to formulate a robust control law applicable across various mission scenarios.

1.1 Background

In the dawn of space exploration, the strategy for handling crucial issues associated with trajectory and navigation was significantly different from modern practices. Initially, space programs operated with considerably larger budgets [1], which facilitated the use of extra fuel reserves and the incorporation of multiple high-grade components to ensure mission success. As previously highlighted, the approach traditionally adopted by mission analysis offices, originating from the earliest space missions and continuing into contemporary practices, involves reserving an additional margin beyond the nominal fuel expenditure, determined through somewhat empirical means or on the basis of prior experience. More in detail, this process begins with an optimization procedure aimed at identifying the optimal trajectory solution that aligns with the mission's goals and technical limitations. Subsequently, an MC analysis is typically employed to simulate potential control challenges the mission might encounter, such as missed thrust events, navigation inaccuracies, or errors in command execution. The determination of the supplementary fuel reserve is then based on ensuring coverage for the worst-case scenario identified in the simulations or securing nearly all scenarios with a high level of confidence.

Numerous studies have been undertaken in the past with the aim of systematically quantifying the required propellant margin allocation, particularly for missions employing low-thrust propulsion systems, whose stringent power limitations significantly complicate the estimation of potential failure magnitudes and associated risks, as encountered during mission design of Dawn [2], by the National Aeronautics and Space Administration (NASA). Such systems are characterized by potential extended periods of missed or nearly-zero thrust, which could lead to trajectory deviations that are difficult, if not impossible, to correct, in contrast with the dynamics of impulsive propulsion, where, excluding irreparable engine failures, corrective maneuvers can be more readily executed to realign with the nominal trajectory. Oh et al. [3] described a methodology for integrating missed-thrust analysis within the trajectory selection process. In their approach, coasting arcs are enforced at some sensitive points along the trajectory, and the nominal one is optimized to account for the presence of these points. Should the spacecraft proceed without experiencing any missed thrust events, the trajectory undergoes reoptimization, during which the forced coast arc is delayed. Nominally, the spacecraft never reaches a forced coast point, as the trajectory is reoptimized for each new forced coast time, but a nearby coasting solution is always available in case the spacecraft lose thrust. Alizadeh and Villac [4] introduced a technique for calculating a margin for position and velocity within the context of the three-body problem, which holds potential for addressing the missed-thrust issue through strategic placement of coast arcs in low-thrust trajectories. An automated method was proposed by Laipert and Longuski [5] to quantify the effect of missed

thrust on a trajectory by measuring the propellant margin required to recover from a missed thrust and the resulting delay in arrival.

The development of a novel method aimed at enhancing the robustness of a trajectory by adjusting its specific characteristics represents a largely unexplored strategy in mission analysis, with few notable exceptions such as the free return orbit employed by the Apollo mission [6]. In this mission, the spacecraft had to be capable of returning to the Earth without the need for propulsion, significantly increasing the mission's resilience to a variety of failures. In different instances, either fortuitous conditions or the swift and resourceful actions of satellite operators have been able to resolve issues, thereby fully or partially salvaging the mission in alignment with its ultimate goals. A notable example of such resilience is observed in the case of the aforementioned Dawn mission, which experienced an engine shutdown due to a high-energy particle impacting an electrical component of the spacecraft — a situation that threatened the entire mission's viability before a successful recovery was executed [7]. Similarly, the Hiten probe, operated by the Japanese Aerospace eXploration Agency (JAXA), benefited from an ingenious solution devised by Belbruno and Miller from the Jet Propulsion Laboratory (JPL). They introduced a strategy based on ballistic capture, which was subsequently implemented, enabling the probe to achieve lunar orbit injection, its primary objective [8].

The management of critical issues related to trajectory deviations is increasingly moving towards more autonomous control systems, reducing the reliance on human intervention. This shift is primarily driven by the evolving landscape of space exploration, which is increasingly favoring smaller platforms like SmallSats or CubeSats [9, 10]. These compact vehicles are characterized by several constraints that inherently limit the traditional approach to mission correction and control:

- limited control authority, which stems from lower thrust levels and a reduced propellant budget, which constrain the scope of potential corrective actions;
- large uncertainties in state knowledge that arise from the adoption of novel navigation techniques or restricted access to ground-based tracking and control facilities;
- significant errors in command actuation, as a consequence of employing low-maturity components.

In such probes, the limited control authority presents maneuvering challenges, as correcting even minor deviations necessitates prolonged burn times. Consequently, Orbit Determination (OD) and the execution of corrective maneuvers emerge as significant concerns, necessitating their consideration in the initial trajectory planning phase. Therefore, the technological restrictions make it challenging to implement alternative strategies and creative solutions. This aspect underscores the inadequacy of traditional trajectory design methods, which may result in operative plans that account for a number of correction maneuvers and demand an excessive amount of propellant for corrections.

The anticipated reduction in human intervention in future space missions is supported by the expanding body of research focused on enabling autonomous navigation for space probes, as discussed by Turan et al. [11]. The capability of a spacecraft to autonomously and accurately estimate their position and velocity is a critical advancement. This development significantly diminishes the reliance on ground station commands, thereby mitigating the potential for errors that such dependency entails. This progress in autonomous navigation is set to revolutionize space exploration by enhancing mission efficiency and reliability through reduced

human oversight. An example of these pioneering advancements is provided by the CubeSat M-ARGO, developed by the European Space Agency (ESA). This mission is set to utilize optical navigation for an autonomous rendezvous with an asteroid. Once reached the asteroid, M-ARGO aims to conduct a thorough examination of the asteroid's physical properties and evaluate its potential for resource exploitation [12].

The advancements in space exploration and navigation highlighted so far delineate the contemporary trends that are currently at the forefront of spacecraft manufacturers' and researchers' endeavors: miniaturization, autonomy, and cost reduction. Within this emerging paradigm, an additional pivotal enhancement that heralds the beginning of a new era in mission design is robust trajectory optimization. This evolution in mission planning marks a departure from traditional reliance on design margins and redundancy, steering towards more intelligent, alternative solutions. Enhancing the probability of mission success through a robust approach to trajectory planning is a logical and anticipated outcome of this shift. It reflects an understanding that, in an era prioritizing efficiency, adaptability, and affordability, the strategic planning of spacecraft trajectories becomes paramount.

1.2 Robust trajectory optimization

Robust trajectory optimization stands distinct from traditional trajectory optimization primarily due to the integration of stochastic variables within the constraints, optimization variables, state definitions, and/or the cost function. In conventional trajectory optimization, where every parameter is deterministic, it is not possible to asses the robustness of the obtained control law. Indeed, the overall robustness of a solution inherently relies on stochastic effects, which are not accounted for in deterministic models. However, by incorporating stochastic representations into the optimization problem, the level of robustness becomes directly tied to the optimized control law. This inclusion shifts the challenge into the domain of robust trajectory optimization, enabling a more nuanced assessment of a mission's resilience to uncertainties.

The introduction of stochastic variables fundamentally transforms the essence of trajectory optimization problems, enriching them with a more nuanced understanding of the mission's dynamics and uncertainties. By incorporating elements such as the first two stochastic moments — the mean and covariance — of the state, the problem gains direct insights into the dispersion of the state along the flight. This provides a clear metric to evaluate robustness during the iterative solution-finding process, regardless of the specific algorithm employed. When these stochastic metrics are integrated into the formulation of constraints or the cost function, the solution that was optimal under deterministic conditions may no longer hold as the best choice in a robust framework. Instead, a solution that embraces the stochastic nature of the problem, accounting for uncertainty in its optimization process, becomes necessary. This shift underscores a more sophisticated approach to mission design, where the inclusion of stochastic elements necessitates a re-evaluation of what constitutes an "optimal" solution. Furthermore, the specific objectives and constraints of each mission can significantly influence how stochastic variables are incorporated into the optimization problem. For instance, a mission could prioritize minimizing the dispersion of the final state while allowing for greater control magnitudes if the control effort is not a concern. Alternatively, a design might focus on maximizing a chosen robustness index, placing strict limits on maximum allowed control effort to adhere, for example, to fuel constraints. Indeed, the strategic application of robust optimization principles enables the simultaneous minimization of various

factors, whether they are associated with stochastic uncertainties or control dynamics. This approach facilitates the acceptance of solutions that constitute a Pareto front, representing an optimal trade-off among competing objectives. Amidst the various considerations, a critical aspect of robust trajectory optimization is the inherent trade-off that robustness improvements often entail an increase in control effort, hence in propellant consumption. This relationship holds true across a wide range of problems and formulations. Enhancing a mission's robustness, that is, its ability to withstand uncertainties and deviations, typically requires additional control actions, which in turn, consume more resources. This fundamental principle underlines the necessity of carefully balancing the desire for increased mission resilience with the practical limitations of resource availability (i.e., mass constraints).

Another fundamental aspect to consider for a general understanding of robust optimization involves the estimation and/or propagation of state uncertainty. As highlighted, stochastic parameters, that are typically represented by the stochastic moments of the spacecraft state, play a crucial role in robust problems, since they serve as robustness index of the trajectory. Thus, an accurate estimate of these quantities is necessary for robust trajectory optimization. However, such a precise knowledge is not always easy to obtain, particularly during extended ballistic phases of the mission or within complex dynamical environments like the Circular Restricted Three-Body Problem (CR3BP) [13]. To overcome this challenge, researchers can employ a variety of Uncertainty Propagation (UP) techniques to estimate state dispersion along the trajectory [14], a topic that will be further explored in a dedicated chapter of this thesis.

Before delving into the complexities of robust optimization, it is worth noting that several straightforward strategies can enhance the robustness of space missions while remaining within the deterministic realm. These approaches are relevant in addressing challenges associated with critical mission phases like Planetary Orbit Injection (POI). POI represents a crucial juncture of the mission as failing to adequately decelerate the spacecraft can lead to inadvertently exiting the planetary system, powered by an unintentional swing-by, consequently prolonging the time to re-encounter the planet. Certain studies have focused on developing methodologies to shorten the recovery flight time, enabling a spacecraft that has failed its POI and escaped the planet's gravity to re-encounter and attempt orbit injection again [15, 16, 17]. In the work of Ozaki et al. [18], the approach is extended to handle probabilistic partial failures of POIs. Other works concentrate on crafting low-thrust trajectories that can withstand Missed Thrust Events (MTEs). These trajectories incorporate expected thrust fractions—continuous-time random variables that represent the probability of a spacecraft encountering an MTE [19, 20].

The initial efforts to address the complexities of robust trajectory design through more advanced methodologies can be traced back to works that explored solutions of the Lambert problem under uncertainties [21, 22, 23, 24]. These studies introduced the concept of computing corrective control measures in the vicinity of an optimal solution by leveraging the principles of Taylor differential algebra. Other studies have explored a different methodology to quantify stochastic errors, specifically through the application of first-order variational equations [25, 26]. Building on this methodology, a novel derivative-free numerical technique that exploits recent advancements in uncertainty quantification has been introduced in [27].

The studies previously mentioned introduce creative and efficient approaches to managing uncertainties; however, they do not fall squarely within the realm of robust optimization methodologies. Their primary focus is on the development of algorithms for robust design, which, crucially, does not incorporate an optimization process. This distinction is important, as it highlights the difference between designing systems

that can withstand uncertainties and the more complex task of optimizing these systems within uncertain environments to achieve the best possible outcomes. The foundational works that initially integrated optimization strategies primarily focused on solving the rendezvous problem, also known as chaser-target problem. For the linear rendezvous problem, a multi-objective optimization technique was formulated that incorporates a robust performance index based on end-state uncertainties, taking into account errors both in navigation and control [28]. This method established a connection between the performance index, rendezvous duration, and propellant expenditure for brief missions. Subsequent work expanded this approach to include nonlinear rendezvous scenarios and the management of extended-duration phases [29]. Deaconu et al. [30] and Louembet et al. [31] introduced two distinct robust planning methods for spacecraft rendezvous, aimed at addressing navigation and maneuver execution errors. They applied these approaches to a series of linear rendezvous scenarios to demonstrate the effectiveness of their methods.

While the aforementioned studies present innovative and effective strategies for addressing uncertainties, their applicability is confined to the specific scenarios for which they were designed. In contrast, the essence of robust trajectory optimization lies in creating control strategies that are tailored to the unique requirements of a particular mission, yet the underlying methodology for their computation should benefit of a versatility applicable across a broad spectrum of conditions. This approach represents a more holistic and adaptable framework for managing uncertainties in space mission planning and execution. The comprehensive state of research within this domain, encompassing both theoretical advancements and practical applications, will be thoroughly explored in a dedicated chapter, highlighting its significance and potential in advancing the field of space exploration.

1.3 Objectives and contribution

The primary objective of this dissertation is to formalize, design, and evaluate a robust trajectory optimization algorithm tailored for space applications. Specifically, the intent is to present a comprehensive methodology capable of optimizing control laws for interplanetary probe transfers. This methodology seeks to ensure broad adaptability across a variety of mission scenarios, encompassing diverse propulsive systems and a range of uncertainties.

The choice of control strategy for a spacecraft significantly influences its robustness and adaptability to uncertainties, and can be categorized into two main types: Open-Loop (OL) control and Closed-Loop (CL) control. In the latter, maneuvers are adjusted in real-time based on the current state of the spacecraft, making the control adaptive to the flown trajectory or any changes in a general set of parameters. This dynamism enables the spacecraft to respond immediately to deviations, enhancing the robustness of the trajectory. Conversely, OL control maintains a predetermined set of maneuvers that remain unchanged throughout the flight, eliminating the possibility for mid-flight corrections. While CL control is often favored in robust optimization for its ability to directly address state dispersion, OL control has its own merits. Indeed, it provides a foundational understanding of the mission's dynamical environment and its inherent challenges. Through robust OL control, a trajectory is designed to inherently avoid conditions that could exacerbate state dispersion, such as close encounters with massive bodies or prolonged ballistic phases. While less flexible, this approach offers valuable insights into the mission's dynamics and potential vulnerabilities. In this study, the focus is primarily placed on CL control strategies, due to their dynamic adaptability and direct responsiveness to real-time

mission variables and uncertainties; yet, an exploration of OL control is also included, providing a comprehensive analysis of its role and effectiveness.

This thesis aims to develop a methodology valid for both impulsive maneuvers, represented by Δvs, and continuous thrust trajectories. The latter approach will account for the spacecraft mass and its variation over time, thus considering the distinct dynamics introduced by different propulsive systems. By adopting this approach, a comprehensive coverage of the entire spectrum of propulsive scenarios can be attained, ensuring that the methodology developed could be used regardless of the propulsion system adopted in the specific mission.

A critical goal is to provide the algorithm with diverse and precise formulations to account for the typical sources of uncertainty encountered throughout a mission. Foremost among these are the state uncertainties, where a pivotal choice lies in selecting the appropriate model to characterize the dispersion of position and velocity. While, for example, a Gaussian distribution might initially seem like a reasonable choice for modeling state uncertainties, the reality of a mission could lead to significantly different distributions. Depending on the specific mission parameters, the actual distribution of uncertainties could deviate markedly from a Gaussian model, varying dynamically over time and possibly assuming unique configurations at different instants. An inaccurate assessment of the state distribution can significantly compromise the efficacy of the optimized control strategy. This is because optimization based on an erroneous understanding of state dispersion operates under false premises, rendering the control actions ineffective or even counterproductive for the intended objectives. In addition to state uncertainties, modeling Navigation Errors (NEs) and Propulsion Errors (PEs) is important. NEs, in particular, can exhibit a wide range of variability due to the diversity of measurements available for OD during a mission (e.g., range, range rate, elevation angles, etc.). Therefore, in this work a decision has been made to incorporate a simplified model of NEs. PEs are more straightforward to characterize. Each maneuver within the mission is defined by a specific thrust vector, making propulsive errors essentially deviations in the magnitude and/or direction of this vector. By selecting an appropriate distribution model for these errors and the corresponding stochastic moments, the cumulative uncertainties can be calculated with relative ease. This more conventional approach to model propulsive errors allows for a precise quantification of uncertainties associated with propulsion, facilitating the optimization process by providing clear, manageable parameters for error analysis.

Once the foundational aspects are defined, the envisioned application of the algorithm could significantly influence its mathematical framework. The primary goal is to develop an optimized robust control law that transfers an interplanetary probe from its initial state to a desired target state, essentially addressing a rendezvous problem. This is contingent upon a defined initial state distribution and the specifics of the propulsion system. A critical constraint involves ensuring that the dispersion of the final state respects a set of predefined requirements, which are based on the unique characteristics of the mission. Furthermore, it is imperative to incorporate constraints related to the control magnitude, acknowledging the inherent limitations of propulsion system. The cost function primarily aims at minimizing propellant usage, aligning with the conventional priorities that guide the preliminary design phase of most space missions. It is important to note that while this manuscript may propose multiple methodologies, variations in the formulation of the cost function and constraints are expected, though the overarching goal remains consistent.

The primary contribution of this Ph.D. dissertation lies in the theoretical framework that underpins the algorithm's formulation. Upon a comprehensive review of the existing literature, detailed in Chapter 2 and 3, this work aims to mix the

most advantageous elements from the current state of the art, with the aspiration of advancing this research domain. The algorithm's development is oriented around three pivotal objectives:

1. **efficiency:** the primary objective is to develop a robust control law for interplanetary transfers, aimed at minimizing trajectory deviations (state dispersion) while adhering to mission parameters, without requiring excessive control efforts or propellant usage;

2. **computational affordability:** the algorithm is designed to generate a control law that necessitates minimal computational resources, especially for execution on an onboard computer. This consideration is pivotal, given the ongoing trend towards hardware miniaturization and autonomous navigation in satellite technology.

3. **versatility:** adaptability to a broad spectrum of mission scenarios is a desired attribute of the proposed methodologies, ranging from routine operations to the complexities entailed in advanced space exploration.

These objectives delineate the dissertation's intent to refine strategic planning in space missions, ensuring that they are not only more efficient and feasible within the constraints of current and forthcoming technology but also broadly applicable across various space exploration initiatives.

1.4 Thesis outline

The thesis is organized as follows. Chapter 2 serves as a comprehensive exploration of the theoretical framework underpinning this dissertation. Initial sections delve into the deterministic Optimal Control Problem (OCP) and survey the leading approaches for addressing it. The following section then shifts to detail the transition necessary to move from a deterministic to a stochastic framework, thereby enhancing understanding of the Robust Optimization Control Problem (ROCP). The chapter concludes with an overview of the latest advancements in robust optimization methods, succinctly presenting the state of the art in this domain.

Chapter 3 concerns an examination of the prevalent methods for UP. This includes a critical assessment of the primary benefits and limitations inherent to each technique, culminating in a rationale for the selection of a specific method within the context of this research.

Chapter 4 formulates and tests a simple OL ROCP on a rendezvous case study. Although the field predominantly focuses on CL control for its dynamic feedback capabilities, exploring the potential of OL control to enhance trajectory robustness is considered valuable. This chapter delves into the merits of an OL approach, suggesting that non-feedback control strategies can also significantly contribute to a mission's robustness.

Chapter 5 introduces and thoroughly explains a CL formulation for impulsive interplanetary transfers that incorporates magnitude-bounded maneuvers. Within a designated section, NEs and PEs are modeled employing a set of simplifying assumptions to streamline the complexity of these uncertainties. To evaluate the effectiveness and adaptability of the developed algorithm, two realistic scenarios are subsequently examined as potential applications. The first scenario analyzes a phase of a JAXA mission involving a rendezvous with an asteroid, where the mission trajectory starts from and returns to the Earth. The second scenario focuses on the

station keeping of a spacecraft in lunar orbit. By treating the initial and final state as identical, rather than as two separate spatial positions and velocities, this test case demonstrates the algorithm's versatility, even in the unique context of periodic orbit control.

In Chapter 6, the previously proposed algorithm for impulsive control is extended to low-thrust trajectories, largely modifying the mathematical formulation and including the spacecraft's mass as additional stochastic variable. The refined algorithm is then applied to a canonical Earth–Mars transfer scenario, demonstrating its capability and effectiveness in managing the complexities of continuous thrust space missions.

A conclusive chapters summarizes the main achievements of this work and outlines strategies for future developments.

Chapter 2

Overview of robust trajectory optimization

Robust trajectory optimization is a specialized branch of mathematical optimization that addresses optimization problems in presence of uncertainties or disturbances. The uncertainties are modeled as stochastic variables that are function the parameters defining the problem. This discipline focuses on formulating solutions that prioritize not only optimal performance under nominal conditions, but also maintain effectiveness in off-nominal cases, ensuring solutions are resilient and reliable in diverse circumstances.

The formulation of an ROCP (hereafter intended as synonym of robust trajectory optimization) extends the theoretical framework of a deterministic OCP, while maintaining the common aim of computing a control law. This control law is determined to optimize a predefined cost index while ensuring compliance with both dynamical and control-related constraints. Consequently, this chapter begins with a detailed exposition of the principles of deterministic trajectory optimization, laying the essential groundwork for understanding the subject. Furthermore, it encompasses a comprehensive review of the predominant methodologies documented in the literature for the calculation of optimal control.

2.1 The optimal control problem

An OCP represents a noteworthy application and development of the Calculus of Variations (CV), which focuses on identifying OL control functions or feedback gains, for CL control laws, that minimize a performance index for a system subject to both algebraic and differential equation constraints. The genesis of CV can be attributed to Pierre de Fermat (1601-1665), who, in 1662, articulated the principle that light traverses paths of minimum time when passing through various media. Galileo Galilei (1564-1642), in 1638, proposed two emblematic challenges subsequently addressed by CV, that is, the "brachistochrone" problem, which seeks the curve along which a body slides from one point to another in the least time, and the determination of the curve formed by a "heavy chain" suspended between two points. Leonard Euler (1707-1783), drawing inspiration from John Bernoulli (1667-1748), authored a pivotal treatise in 1744, *The method of finding curves that show some property of maximum or minimum*, addressing several specific problems and establishing the foundational elements of CV theory [32]. Jean Louis Lagrange (1736-1813) engaged in correspondence with Euler and developed the method of "variation," a concept praised by Euler and from which the field derives its name. Additionally, Lagrange

introduced the method of *multipliers*, a concept that Euler also embraced, leading to the formulation of the first-order necessary conditions for a stationary solution. These conditions are now known as the Euler-Lagrange equations, foundational to the calculus of variations and optimal control theory.

Before delving into a more rigorous formulation of the mathematical cencepts, it is important to clarify the notation conventions used throughout this thesis. Scalar quantities and functions are represented by lowercase (e.g., a) or capital (e.g., N) Roman letters, whereas vector variables and functions are indicated by lowercase bold letters (e.g., $\boldsymbol{x}$). The Euclidean norm of a vector is signified by the same symbol in non-bold (e.g., $x = |\boldsymbol{x}|$), unless specified otherwise. Additionally, matrices are denoted by uppercase bold Roman letters (e.g., $\boldsymbol{M}$).

2.1.1 Dynamical system and constraints

At any given moment $t \in \mathbb{R}$, the state of the dynamical system is defined by the vector $\boldsymbol{x}(t) \in \mathbb{R}^n$, $n \in \mathbb{N}$, as the dimension of the state vector. This vector collects all the variables required to fully describe the system being studied, such as position, velocity, and mass of a spacecraft. The trajectory of the system through a designated time interval $[t_0, t_f]$ is shaped by the temporal progression of the state vector subject to the control vector $\boldsymbol{u}(t) \in \mathbb{R}^m$, with $m \in \mathbb{N}$ as the dimension of the control vector. This progression is achieved by solving a collection of first-order Ordinary Differential Equations (ODEs), as

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\mathrm{d}t \tag{2.1}$$

where the vector function $\boldsymbol{f}\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \mapsto \mathbb{R}^n$ denotes the system dynamics.

The system trajectory is required to satisfy multiple specifications, expressed as equality or inequality constraints. Constraints solely dependent on the state and control variables at the start and/or at the end of the time domain are classified as *boundary constraints*. These can be collected within a vector $\boldsymbol{\chi} \in \mathbb{R}^p$ composed of homogeneous algebraic equations, as

$$\boldsymbol{\chi}\left(\boldsymbol{x}(t_0), \boldsymbol{x}(t_f), \boldsymbol{u}(t_0), \boldsymbol{u}(t_f), t_0, t_f\right) = \boldsymbol{0} \tag{2.2}$$

with p as the number of boundary constraints. General constraints that are meant to be enforced through all the considered time interval are instead called *path constraints* and are usually expressed with a vector $\boldsymbol{\psi} \in \mathbb{R}^q$ of inequalities as

$$\boldsymbol{\psi}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), t\right) \leqslant \boldsymbol{0} \tag{2.3}$$

with q as the number of path constraints. An example of boundary constraint is given by the departure from an assigned state $\boldsymbol{x}_0$ or the arrival to a target state $\boldsymbol{x}_f$ as

$$\boldsymbol{x}(t_0) - \boldsymbol{x}_0 = \boldsymbol{0} \tag{2.4}$$

$$\boldsymbol{x}(t_f) - \boldsymbol{x}_f = \boldsymbol{0} \tag{2.5}$$

while a path constraint could limit instead the maximum magnitude of the control vector as

$$\|\boldsymbol{u}(t)\| \leqslant \widetilde{u}_{\max} \qquad t \in [t_0, t_f] \tag{2.6}$$

with $\widetilde{u}_{\max}$ as maximum allowed control magnitude.

2.1.2 Cost function

The aim of the OCP is to find the continuous-time control function $\boldsymbol{u}(t)$ that extremize a scalar objective function J, indicative of the system's performance. In this work, the minimization of the cost function is always considered, although it is not an unusual practice to recast the formulation into an equivalent maximization problem by changing the sign of the objective function as needed. This adjustment may be necessary to accommodate the specifics of the chosen numerical solver.

The general expression of an objective function for the OCP is

$$J = \phi\left(\boldsymbol{x}(t_0),\, t_0,\, \boldsymbol{x}(t_f),\, t_f\right) + \int_{t_0}^{t_f} \boldsymbol{\Phi}\left(\boldsymbol{x}(t),\, \boldsymbol{u}(t),\, t\right) \mathrm{d}t \tag{2.7}$$

The fist component is a function of epochs, and the initial and final states only, while the second term depends on the evolution in time of the state and control variables through the function Φ. If J is written with both these two components, the resulting OCP is known a *problem of Bolza*. While for $\phi = 0$ and $\boldsymbol{\Phi} = \mathbf{0}$ the OCP is respectively referred to as *problem of Lagrange* and *problem of Mayer* [33]. It should be highlighted that the same problem can be formulated in any of the three variations by employing appropriate auxiliary variables.

Examples of typical cost functions for space missions are: minimization of the fuel consumption (*fuel-optimal problem*), that leads to the cost function to maximize

$$J = m(t_f) \tag{2.8}$$

where $m(t) \in \mathbb{R}_+$ is the mass of the spacecraft over time; the minimum ToF (*time-optimal problem*) is achieved by minimizing the immediate cost function

$$J = t_f - t_0 = \Delta t \tag{2.9}$$

while the minimization of the imparted control energy (*energy-optimal problem*) is performed adopting the cumulative control magnitude

$$J = -\frac{1}{2}\int_{t_0}^{t_f} \|\boldsymbol{u}\|_2 \mathrm{d}t \tag{2.10}$$

Even though OCPs traditionally deal with the minimization of a scalar cost function for optimization, it is possible to optimize multiple objectives simultaneously using multi-objective (or Pareto) optimization [34, 35], which incorporates several objective functions. In multi-objective optimization, finding a singular solution that optimizes every cost function is unlikely due to potential conflicts between objectives, thus one aims at finding a set of Pareto-optimal solutions. A solution is deemed Pareto optimal (or noninferior) if improving one objective function's value would lead to the deterioration of others. In the absence of further subjective preferences, a multitude of Pareto optimal solutions may exist, all considered equally viable. Therefore, selecting a single solution from the Pareto optimal set (defined as *Pareto front*) requires additional criteria and considerations beyond numerical performance indices. In this dissertation, however, only single objective cost functions are considered for the optimization process.

2.1.3 Multi-arc formulation

The elements introduced in the previous sections are sufficient to formulate simple OCPs. Yet, it is important to note that an excessively compact formulation may lead

to numerical challenges under certain conditions, particularly in scenarios where state and/or control variables experience rapid or abrupt changes. As a result, the optimal control function $\boldsymbol{u}(t)$ might exhibit significant and periodic discontinuities, which can make the associated computational tasks markedly challenging and often impractical in many scenarios. To avoid this issue, a widely used approach in the literature involves breaking down the time horizon of the state and control functions into a series of finite *segments* (or *arcs*), while ensuring the continuity of these functions across the internal boundaries of a trajectory to prevent dynamical inconsistencies. Typically, the dynamical system described by Eq. (2.1) remains unchanged across all segments, although it may be convenient for certain problems to consider a distinct set of ODEs for each arc. The lengths of the time intervals into which a trajectory is conventionally divided may be considered as variables subject to optimization. This perspective allows for a more dynamic and flexible approach to trajectory planning, enabling the optimization framework to tailor these durations for enhanced mission efficiency and performance.

Considering N distinct time intervals, the additional boundary conditions imposed on the state to maintain dynamical consistency are formulated as

$$\boldsymbol{x}(t_f^{(i)}) = \boldsymbol{x}(t_0^{(i+1)}) \qquad \forall i = 1, \dots, N-1 \tag{2.11}$$

where the superscript (i) indicates that a variable belongs to the i-th segment. The initial and final boundary epochs of all segments are grouped in the following sets

$$\mathcal{T}_0 = \left\{ t_0^{(i)},\, \forall i = 1, \dots, N \right\} \tag{2.12}$$

$$\mathcal{T}_f = \left\{ t_f^{(i)},\, \forall i = 1, \dots, N \right\} \tag{2.13}$$

Similarly, the state and controls at the boundaries are grouped as

$$\mathcal{X}_0 = \left\{ \boldsymbol{x}\left(t_0^{(i)}\right),\, \forall i = 1, \dots, N \right\} \tag{2.14}$$

$$\mathcal{U}_0 = \left\{ \boldsymbol{u}\left(t_0^{(i)}\right),\, \forall i = 1, \dots, N \right\} \tag{2.15}$$

$$\mathcal{X}_f = \left\{ \boldsymbol{x}\left(t_f^{(i)}\right),\, \forall i = 1, \dots, N \right\} \tag{2.16}$$

$$\mathcal{U}_f = \left\{ \boldsymbol{u}\left(t_f^{(i)}\right),\, \forall i = 1, \dots, N \right\} \tag{2.17}$$

Incorporating the previous notation, the cost function is written as

$$J = \phi\left(\mathcal{X}_0,\, \mathcal{T}_0,\, \mathcal{X}_f,\, \mathcal{T}_f\right) + \sum_{i=1}^{N} \int_{t_0^{(i)}}^{t_f^{(i)}} \boldsymbol{\Phi}^{(i)}\left(\boldsymbol{x}(t),\, \boldsymbol{u}(t),\, t\right) \mathrm{d}t \tag{2.18}$$

while the boundary constraints are now generalized as

$$\boldsymbol{\chi}\left(\mathcal{X}_0,\, \mathcal{X}_f,\, \mathcal{U}_0,\, \mathcal{U}_f,\, \mathcal{T}_0,\, \mathcal{T}_f\right) = \mathbf{0} \tag{2.19}$$

The ODEs and path constraints in a multi-segment formulation are instead extended as

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}(t),\, \boldsymbol{u}(t),\, t)\mathrm{d}t \tag{2.20}$$

$$\boldsymbol{\psi}\left(\boldsymbol{x}(t),\, \boldsymbol{u}(t),\, t\right) \leqslant \mathbf{0} \qquad t \in \left[t_0^{(i)},\, t_f^{(i)}\right] \qquad \forall i = 1, \dots, N \tag{2.21}$$

The OCP with a multi-segment formulation is thus stated as

$$\min_{\boldsymbol{u}(t)} \text{Eq. (2.18)} \tag{2.22a}$$

$$\text{s.t.} \quad \text{Eqs. (2.19)} - \text{(2.21)} \tag{2.22b}$$

In the following section, the principle methods to address this deterministic trajectory optimization problem are briefly explained together with their advantages and disadvantages.

2.2 Numerical methods for solving optimal control problems

Solving the problem outlined in Eqs. (2.22) involves calculating the optimal time-continuous control function $\boldsymbol{u}(t)$ and the associated state trajectory $\boldsymbol{x}(t)$. A challenge arises as standard optimization algorithms are typically engineered to handle a finite number of optimization variables, whereas optimizing a function leads to an infinite-dimensional problem. Consequently, both the control function and, by extension, the state function necessitate some kind of discretization. The discretization approach and the way the conditions for optimality are handled within the computational framework distinguish the various *optimization methods* used to solve these complex problems. This section offers a comprehensive overview of the principal optimization methods, shedding light on their unique features and how they transform continuous optimization problems into discrete counterparts for effective resolution.

The two main categorizes under which the optimization methods are categorized are the *indirect* and *direct* methods, covered in this section, while other alternative methods are briefly surveyed in conclusion to this section. Indirect methods are based on the principles of the calculus of variations. In particular, the first-order optimality conditions (i.e., the aforementioned Euler-Lagrange equations) and algebraic boundary conditions of the control and state functions are employed to formulate the problem in the form of a Boundary Value Problem (BVP), or a Multi-Point BVP (MPBVP) in the sense of multi-arc problems, whose solution is normally achieved by means of shooting techniques in the literature. A distinctive aspect of these methods is the use of adjoint variables, and Lagrange multipliers, which are employed to ensure compliance with differential and algebraic constraints. While these variables may not directly embody any physical meaning, they are an indispensable component of the optimization process, as they are required in conjunction with the state and control functions to obtain a well-posed problem. Given that the size of the multiplier set precisely matches the state dimension by definition, this set of variables is often termed as *costate*. These methods are termed "indirect" because they determine the optimal solution through the indirect resolution of a BVP involving both the state and costate, complying with differential constraints. Adherence to these conditions ensures the attainment of optimality. A prominent benefit of indirect methods is their ability to swiftly converge to a local optimal solution with remarkable accuracy, utilizing minimal computational resources. However, this efficiency comes with the prerequisite of an initial estimate closely aligned with the optimal solution to ensure convergence. The abstract nature of the costate variables further complicates this process, making the fine-tuning of these parameters towards an optimal solution a challenging endeavor. In addition, managing path constraints within the optimization process often presents challenges,

as it necessitates prior knowledge of the sequence of constrained and unconstrained arcs.

Direct methods, as the name suggests, undertake the optimization of control variables directly using NonLinear Programming (NLP) techniques, bypassing the need for predefined optimality conditions. This approach involves discretizing the state and/or control function and considering their values at each grid time as optimization variables. The optimality conditions are then applied to the NLP problem, rather than the time continuous one. A key benefit of direct methods is their broad convergence radius, which make them particularly suitable for problems where the optimal solution is mostly unknown. Dynamical equations are posed as algebraic constraints, eliminating the necessity for adjoint variables found in indirect methods. However, a notable drawback of direct methods lies in the significantly increased problem size due to the introduction of numerous variables through discretization, leading to a large, yet sparse, NLP problem, which is much harder to solve than an MPBVP from a computational point of view. Another drawback of direct methods is related to the optimality of the achieved solution, which is typically uncertain or challenging to evaluate.

2.2.1 Indirect methods

As previously discussed, indirect methods draw upon CV theory, with the first-order optimality conditions leading to the ODEs for the costate in the resulting BVP. Following the formulation adopted by Bryson and Ho in their pioneering book on optimal control [36], the cost function of Eq. 2.7 is substituted with an updated J^* as

$$J = \phi + \boldsymbol{\mu}^T \boldsymbol{\chi} + \int_{t_0}^{t_f} \left(\boldsymbol{\Phi} + \boldsymbol{\lambda}^T \left(\boldsymbol{f} - \dot{\boldsymbol{x}} \right) \right) \mathrm{dt} \tag{2.23}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ denote the Lagrange multipliers related to boundary and dynamical constraints, respectively, with the latter specifically signifying the costate. For the sake of simplicity, the function arguments are not written explicitly. The "dot notation" indicates a derivative with respect to time. For simplicity, this discussion omits path constraints; however, their inclusion is addressed in [36], where a detailed explanation is provided. It is also noteworthy to remark that, when the boundary conditions and dynamic constraints are both met, the merit indices J and J^*, as well as their respective maximum values, are equivalent.

By applying integration by parts to Eq. (2.23), it is possible to remove the dependency on the state variables' derivatives from the functional J^*, resulting in

$$J = \phi + \boldsymbol{\mu}^T \boldsymbol{\chi} + \boldsymbol{\lambda}_0^T \boldsymbol{x}_0 - \boldsymbol{\lambda}_f^T \boldsymbol{x}_f + \int_{t_0}^{t_f} \left(\boldsymbol{\Phi} + \boldsymbol{\lambda}^T \boldsymbol{f} + \dot{\boldsymbol{\lambda}}^T \boldsymbol{x} \right) \mathrm{dt} \tag{2.24}$$

After introducing the Hamiltonian H as

$$H = \boldsymbol{\Phi} + \boldsymbol{\lambda}^T \boldsymbol{f} \tag{2.25}$$

the first variation of J^* is obtained by differentiating Eq. (2.24)

$$\delta J^* = \left(-H_0 + \frac{\partial \phi}{\partial t_0} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\chi}}{\partial t_0}\right) \delta t_0 + \left(H_f + \frac{\partial \phi}{\partial t_f} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\chi}}{\partial t_f}\right) \delta t_f + \tag{2.26}$$

$$+ \left(\boldsymbol{\lambda}_0^T + \frac{\partial \phi}{\partial \boldsymbol{x}_0} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{x}_0}\right) \delta \boldsymbol{x}_0 + \left(-\boldsymbol{\lambda}_f^T + \frac{\partial \phi}{\partial \boldsymbol{x}_f} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{x}_f}\right) \delta \boldsymbol{x}_f + \tag{2.27}$$

$$+ \int_{t_0}^{t_f} \left(\left(\frac{\partial H}{\partial \boldsymbol{x}} + \dot{\boldsymbol{\lambda}}^T\right) \delta \boldsymbol{x} - \frac{\partial H}{\partial \boldsymbol{u}} \delta \boldsymbol{u}\right) \mathrm{dt} \tag{2.28}$$

The first-order necessary condition for an optimal solution requires that the first variation δJ^* vanishes for all possible variations δt_0, δt_f, $\delta \boldsymbol{x}_0$, $\delta \boldsymbol{x}_f$, $\delta \boldsymbol{x}$, and $\delta \boldsymbol{u}$. By setting the coefficients of $\delta \boldsymbol{x}$ and $\delta \boldsymbol{u}$ within the integral of Eq. (2.28) to zero at any point on the trajectory, the Euler-Lagrange differential equations can be derived, which govern the behavior of the adjoint variables as

$$\dot{\boldsymbol{\lambda}} = -\left(\frac{\partial H}{\partial \boldsymbol{x}}\right)^T \tag{2.29}$$

and the algebraic equations of the optimal control as

$$\left(\frac{\partial H}{\partial \boldsymbol{u}}\right)^T = \mathbf{0} \tag{2.30}$$

Particular attention must be paid if one of the controls is subject to a constraint, meaning it must belong to a given domain of admissibility (e.g., the magnitude of the thrust must be between the minimum value 0 and the maximum value $T_{\max}$). Cases where the constraint depends on time or state variables are not considered, but only those where it is explicit and constant. In the presence of such a constraint, the optimal control value at every point on the trajectory is the one that, while belonging to the domain of admissibility, maximizes (if maxima of J are sought) or minimizes (if minima are sought) the Hamiltonian (2.25) at that point, as established by the Pontryagin Maximum Principle (PMP) [37]. Thus, two scenarios are possible:

- The optimal control value is the given by solving Eq. (2.30) if it falls within the domain of admissibility and thus the constraint does not intervene at that point;
- The optimal value is at the extremes of the domain, i.e., the control assumes the maximum or minimum value if the one given by Eq. (2.30) does not fall within the domain of admissibility.

A particular case occurs if the Hamiltonian is linear with respect to one of the constrained controls because in the corresponding equation (2.30) the control does not appear explicitly and therefore cannot be determined. Therefore, if in Eq. (2.25) the coefficient of the control in question is non-zero, then H is maximized for the maximum value of the control if the coefficient is positive and minimized if it is negative (*bang-bang* control), in accordance with PMP. However, if in Eq. (2.25) the coefficient of the control is identically zero over a finite time interval (*singular arc*), then it is necessary to set all higher-order time derivatives of the coefficient to zero, until the control appears explicitly in one of them: the optimal control is then determined by setting this last derivative equal to zero.

Nullifying the coefficients of the variation terms δt_0, δt_f, $\delta \boldsymbol{x}_0$, and $\delta \boldsymbol{x}_f$ in Eq. (2.28) leads to the definition of the transversality conditions:

$$\boldsymbol{\lambda}_0^T + \frac{\partial \phi}{\partial \boldsymbol{x}_0} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{x}_0} = \mathbf{0} \tag{2.31}$$

$$-\boldsymbol{\lambda}_f^T + \frac{\partial \phi}{\partial \boldsymbol{x}_f} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\chi}}{\partial \boldsymbol{x}_f} = \mathbf{0} \tag{2.32}$$

$$-H_0 + \mu^T \frac{\partial \boldsymbol{\phi}}{\partial t_0} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\chi}}{\partial t_0} = \mathbf{0} \tag{2.33}$$

$$H_f + \mu^T \frac{\partial \boldsymbol{\phi}}{\partial t_f} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\chi}}{\partial t_f} = \mathbf{0} \tag{2.34}$$

The boundary conditions for optimality are inferred by excluding the constant Lagrange multipliers μ from Eqs. (2.31)-(2.34), culminating in a formulation such as:

$$\boldsymbol{\sigma}\left(\boldsymbol{x}_0, \boldsymbol{x}_f, \boldsymbol{\lambda}_0, \boldsymbol{\lambda}_f, t_0, t_f\right) = \mathbf{0}. \tag{2.35}$$

Equations (2.31)-(2.34) specify certain optimality conditions for the adjoint variables $\boldsymbol{\lambda}$ and the Hamiltonian H. The j-th component of a vector is henceforth denoted with the subscript j. When a specific state variable $x_j \in \boldsymbol{x}$ is fixed at the initial and/or final times, leading to $\delta x_{j,0} = 0$ and/or $\delta x_{j,f} = 0$, there are no requirements for the initial and/or final values of the corresponding adjoint variable λ_j, rendering it "free." Conversely, if the initial and/or final value of the state variable $x_j \in \boldsymbol{x}$ is not explicitly stated in the boundary conditions $\boldsymbol{\chi}$ or in the function ϕ, the initial and/or final values of the adjoint variable λ_j must be zero. In a similar vein, if the initial and/or final times are not explicitly included in $\boldsymbol{\chi}$ or ϕ, then the Hamiltonian takes the value of zero at the initial and/or final times.

When employing a shooting method to address the BVP, the task simplifies to pinpointing the initial values of the undetermined state and adjoint variables that meet the boundary conditions. This is accomplished by numerically integrating the ensemble of differential equations across the time domain. Typically, an iterative process akin to Newton's method is utilized, generating a succession of initial value problems that iteratively approach convergence. When dealing with multiple-arcs trajectories whose time durations are generally unknown, the resulting MPBVP is solved by considering the boundary conditions

$$\boldsymbol{\chi}(\boldsymbol{s}) = \mathbf{0} \tag{2.36}$$

where in the vector $\boldsymbol{s}$ the constant paramenters, such as the arc durations, and the internal boundary conditions for the state, costate and control are included.

2.2.2 Direct methods

The main challenges with indirect methods are a small convergence radius to an optimal solution and the need to specify an *a priori* control structure, which limits the ability of the methods to explore different solutions. Moreover, unlike state, costate variables lack physical interpretation, can vary greatly in magnitude compared to state variables, and may exhibit discontinuities at the transition points between constrained and unconstrained arcs in the solution. The conventional *shooting* methods, which involve propagating the state and costate via differential equations and assessing errors at the final time, presents significant complexities for resolution of BVPs, especially within the framework of these methods. Hence,

starting from the 1960s, alternative strategies to solve the MPBVP associated with dynamic system optimization began to emerge. A variety of solutions were sought, sharing the overarching strategy of transforming the continuous problem into one of parameter optimization and moving away from the shooting method towards techniques where all free parameters are optimized simultaneously [38]. In particular, the continuous OCP is converted into a finite-dimension NLP problem by means of the discretization of the state and/or control variables. Methods that are based on this transcription lays within the definition of direct methods.

A first category of direct methods, referred as direct shooting methods, involves the discretization of the control $\boldsymbol{u}$ only, using a continuous or piecewise-continuous function, such as a polynomial, which is specified by a limited number of parameters. These parameters become the decision variables or unknowns in the resulting NLP problem. The state variables at any point in time are obtained by numerically integrating the system's differential equations using time-marching algorithms such as the Runge-Kutta family of methods. The optimal values of the parameters that define the control law are then found by solving the related NLP problems. This can be done effectively using a vast plethora of deterministic NLP solvers, like interior-point or Sequential Quadratic Programming (SQP) algorithms [39].

A second group of methods is termed direct collocation and is arguably the most recognized and widely applied direct transcription method. State and control trajectories are approximated through polynomials of a certain degree, while the differential constraints are converted into algebraic constraints. This is done by ensuring the polynomials adhere to the differential equations at selected intermediate collocation points. The initial phase in the collocation transcription process involves segmenting the time domain into discrete intervals by establishing a grid comprised of $N+1$ time points $t_0 < t_1 < \cdots < t_N = t_f$, known as *mesh points* or *nodes*. In each segment within two consecutive nodes the state $\boldsymbol{x}(t)$ is approximated by a polynomial $\boldsymbol{s}$ of degree d in the form

$$\boldsymbol{s}(t) = \sum_{i=0}^{d} c_i t^i \qquad t \in [t_i,\, t_{i+1}] \tag{2.37}$$

with c_i as the constant coefficients of the polynomials. The coefficients are determined by ensuring that the polynomial's value and its first derivative at the mesh points match the state variable $\boldsymbol{x}(t_j)$ and its first derivative $\boldsymbol{f}\left(\boldsymbol{x}(t_j),\, \boldsymbol{u}(t_j), t_j\right)$ at those corresponding points, having

$$\boldsymbol{s}(t_i) = \boldsymbol{x}(t_i) \tag{2.38}$$
$$\dot{\boldsymbol{s}}(t_i) = \boldsymbol{f}\left(\boldsymbol{x}(t_i),\, \boldsymbol{u}(t_i),\, t_i\right) \tag{2.39}$$
$$\boldsymbol{s}(t_{i+1}) = \boldsymbol{x}(t_{i+1}) \tag{2.40}$$
$$\dot{\boldsymbol{s}}(t_{i+1}) = \boldsymbol{f}\left(\boldsymbol{x}(t_{i+1}),\, \boldsymbol{u}(t_{i+1}),\, t_{i+1}\right) \tag{2.41}$$

In case the degree d is too high to evaluate the interpolating polynomial, additional constraints on q internal points $t_{i,k}$, $k = 1, \ldots, q$, between nodes are added. with $q = (d+1) - 4$, that is the $d+1$ constraints needed for the coefficients c_j, minus the 4 given by the pair of nodes. For odd $d > 3$, $q = \left((d+1) - 4\right)/2$, that is the $d+1$ constraints needed for the coefficients c_j, minus the 4 given by the pair of external nodes, while for even $d > 3$, $q = \left((d+1) - 4 - 1\right)/2 + 1$, as the last constraint on the derivative is omitted [40], as

$$\boldsymbol{s}(t_{i,k}) = \boldsymbol{x}(t_{i,k}) \tag{2.42}$$
$$\dot{\boldsymbol{s}}(t_{i,k}) = \boldsymbol{f}\left(\boldsymbol{x}(t_{i,k}),\, \boldsymbol{u}(t_{i,k}),\, t_{i,k}\right) \qquad \forall k = 1, \ldots, q \tag{2.43}$$

To increase the accuracy of the approximating function, additional constraints only on the first derivatives are added between external and internal nodes, constituting the *defects* $\Delta_{i,p}$, $p = 1, \ldots, q+1$, as to have a defect between each pair of internal/external nodes, hence

$$\Delta_{i,p} = \dot{\boldsymbol{s}}(t_{i,p}) - \boldsymbol{f}\left(\boldsymbol{x}(t_{i,p}),\, \boldsymbol{u}(t_{i,p}),\, t_{i,p}\right) \qquad \forall p = 1, \ldots, q+1 \tag{2.44}$$

A similar formulation is given for the interpolating function of control $\boldsymbol{u}(t)$, although simple linear or piecewise-constant function are not rarely chosen.

Enclosing the state and control variables at the internal and external nodes in a vector $\boldsymbol{Z}$, and the nonlinear constraints, including defects, in a vector $\boldsymbol{\Gamma}(\boldsymbol{Z})$, the OCP problem is stated as

$$\min_{\boldsymbol{u}(t)} J\left(\boldsymbol{Z}\right) \tag{2.45a}$$

$$\text{s.t.} \quad \boldsymbol{Z}_L < \boldsymbol{Z} < \boldsymbol{Z}_U \tag{2.45b}$$

$$\boldsymbol{B}_L < \boldsymbol{A}\boldsymbol{Z} < \boldsymbol{B}_U \tag{2.45c}$$

$$\boldsymbol{\Gamma}_L < \boldsymbol{\Gamma}(\boldsymbol{Z}) < \boldsymbol{\Gamma}_U \tag{2.45d}$$

where $\boldsymbol{A}\boldsymbol{Z}$ are the linear constraints and subscripts L and U stands for the lower and upper bounds respectively.

Several reputable commercial solvers, including SNOPT [41] and IPOPT [42], are available for solving large sparse NLPs like the one presented in Eq.(2.45). Additionally, there are various software tools that facilitate the transcription process, such as GPOPS-II [43] and TransWORHP [44].

In recent times, *convex programming*-based direct optimization methods have increasingly become a focal point for solving optimal control problems within the aerospace sector [45, 46, 47]. Unlike NLP solvers, interior-point algorithms tailored to convex problems ensure a polynomial-time convergence to the global optimum, independent of how the problem is initialized. Nonetheless, the formulation of many real-world challenges as a convex optimization problems is often not feasible. As a result, various strategies for transforming the original nonconvex problem into a convex format have been developed, a procedure commonly referred to as *convexification*. Over time, two main techniques for convexification have been established: *lossless convexification* and *successive convexification*. Lossless convexification involves formulating an equivalent convex problem by appropriately altering variables and/or relaxing constraints [48]. If lossless convexification is not sufficient to eliminate all non-convex constraints, successive convexification becomes necessary. This technique involves solving a series of convex subproblems that are defined by linearizing the dynamics and other nonconvex constraints around a previously determined solution. Theoretical assurance that successive convexification also yields a (locally) optimal solution for the original problem is conditional on meeting certain criteria [49, 50].

2.2.3 Alternative methods

Two principal classes of optimization methods recently emerged as alternative approaches for aerospace applications. Evolutionary Algorithms (EAs) form one such category, offering global optimization solutions through heuristic rules. These guidelines are often, but not exclusively, derived from natural phenomena to pinpoint the best solution for optimization challenges. Concurrently, Machine Learning (ML) has surfaced as another influential category, applying its principles to tackle optimization issues [51]. This typically involves training a Neural Network (NN)

through one of two approaches: Behavioural Cloning (BC), which uses a controlled set of solved problems as examples to emulate, or Reinforcement Learning (RL), entailing extensive interaction with numerous instances of the environment. The latter approach, RL, is also recognized for its potential in contributing to robust optimization techniques.

2.2.3.1 Evolutionary algorithms

The foremost benefit of EAs lies in their vast exploration of the optimization domain, thereby amplifying the likelihood of finding a globally optimal solution. This broad survey is achieved through a parallel random search across diverse sets of solutions, wherein the process iteratively eliminates the less optimal solutions in favor of those that exhibit greater performance. The allure of these methods, sometimes referred to as meta-heuristics, is significantly heightened by their capacity to seamlessly converge to an optimal solution, even when the initial solution is considerably distant from the convergence basin. This robust adaptability sets them apart from traditional optimization techniques, especially in scenarios characterized by nonlinear, non-differentiable objectives, or intricate feasibility domains.

A key strategy within EA is represented by Genetic Algorithms (GAs), which initiate by creating a population of potential solutions. These solutions are then subjected to selection, crossover, and mutation processes, drawing inspiration from Darwinian evolutionary theory. This method systematically explores discrete solution spaces in pursuit of optimal outcomes, effectively mimicking the natural selection process to evolve solutions towards higher levels of fitness and performance [52]. Differential Evolution (DE) stands out as a variation of GA, specifically tailored for tackling problems within continuous spaces [53]. Similarly, Particle Swarm Optimization (PSO) [54] and Ant Colony Optimization (ACO) [55] are renowned meta-heuristic techniques, modeled on the collective foraging behaviors observed in bird flocks and ant colonies, respectively. A rather exhaustive list of these bio-inspired optimization techniques is maintained by Campelo et al. [56].

In the aerospace sector, where the design of missions often entails addressing exceedingly complex optimal control challenges, EA methods have gained considerable popularity in recent decades. Genetic algorithms, for example, are frequently utilized to navigate high-dimensional combinatorial problems, such as planning active debris removal missions [57, 58, 59]. Similarly, differential evolution has proven effective in unraveling intricate continuous problems like designing multiple gravity-assist trajectories [60, 61] and rocket ascent paths [62]. While evolutionary algorithms can be directly applied to combinatorial problems, their effectiveness may diminish in overly large search spaces, potentially yielding sub-optimal outcomes. An alternative, particularly suitable when the problem domain permits incremental solution construction from smaller, distinct sub-problems, is the use of *tree searches*. Tree search methods stand out as highly effective for such challenges, modeling decision points as nodes for potential expansion and evaluation. Given the impracticality of exhaustively enumerating all possible expansions, these methods employ selective strategies to explore only the most promising branches. This approach generates more manageable sub-problems, better suited to evolutionary algorithm solutions [63].

Nonetheless, it's important to acknowledge that meta-heuristic algorithms typically exhibit slower convergence rates, significantly elevating the computational demands in comparison to deterministic methods. Furthermore, these algorithms often lack robust convergence guarantees, rendering the iterations required to reach the global optimum or achieve a solution close to optimal within an acceptable tolerance

potentially infinite or indeterminable. Such characteristics render meta-heuristic approaches less viable for scenarios where time is crucial, and where computational efficiency and assured convergence are paramount.

2.2.3.2 Machine Learning

In line with the current interest across various scientific and technical disciplines, Artificial Intelligence (AI) is demonstrating remarkable potential in the realm of trajectory design and optimization. A significant advantage of AI lies in its capability to deliver solutions with minimal computational resources and near-instantaneous response times once the algorithm has been adequately trained for a specific task. ML techniques are used for this particular scope.

Neural Networks (NNs), fundamental tools of ML, stand out as universal function approximators capable of emulating the optimal CL control law, effectively mapping the observed spacecraft state (input vector) to the required thrust magnitude and direction (output vector). This policy evaluation process is exceedingly rapid, surpassing the speed of traditional optimization methods, making it particularly attractive for applications where time is of the essence. However, as previously highlighted, the critical endeavor lies in the training of Neural Networks (NNs). This process involves meticulously optimizing the network's parameters to achieve a reasonable approximation of the optimal control law, ensuring the neural network can accurately model the desired outcomes. Methods that employs artificial neural networks with multiple layers to model complex patterns in data lay within the definition of Deep Learning (DL). Each layer transforms its input data into a slightly more abstract and composite representation, effectively enabling the network to learn from vast amounts of unstructured data. DL demonstrated to be both efficient and swift in resolving control issues across various research domains, particularly in the field of robotics [64].

Three different categories of training are possible to this scope: *supervised learning*, *unsupervised learning*, and RL. Supervised Learning involves training a model on a labeled dataset, where each training example is paired with an output label. The model learns to predict the output from the input data, making it ideal for tasks such as classification and regression. An example is given by BC, which is a widely used training approach that leverages a dataset of optimal trajectories from an "expert" source, such as solutions generated by a deterministic solver, to train the network. The objective is to mimic the expert's actions by reducing the discrepancy between the network's output and the expert's data. BC has been effectively applied in various aerospace scenarios, including interplanetary transfers [65], powered descent landings [66, 67], and hypersonic reentry phases [68, 69]. Unsupervised Learning, on the other hand, deals with datasets without explicit labels. The goal here is to discern underlying patterns or structures within the data. This can involve clustering similar data points together or identifying distinct data distributions. Aerospace applications in literature are principally related to hypersonic vehicles [70, 71]. Reinforcement Learning is a type of ML where an agent learns to make decisions by taking actions in an environment to achieve some objective. The learning process involves the agent interacting with its environment, receiving feedback in the form of rewards or penalties based on the actions taken within the mission scenario, usually formulated as a Markov decision process. The goal of the agent is to maximize the cumulative reward over time. This framework enables the agent to learn from its experiences, adjusting its strategy to improve performance in tasks such as game playing, robotic control, and autonomous navigation. As a derived method, Deep Reinforcement Learning (DRL) combines the representational power of DL

with the decision-making prowess of RL. It's an advanced AI technique where deep neural networks are trained to make sequences of decisions, learning optimal policies directly from high-dimensional sensory input. Despite being a relatively recent area of interest within the aerospace community, RL and DRL have seen a notable surge in research output in recent years. This burgeoning field has been applied to a variety of aerospace challenges, including powered descent landing [72, 73], low-thrust transfer maneuvers [74, 75], cislunar trajectories [76, 77, 78], spacecraft rendezvous [79, 80], proximity operations [81], and terminal guidance [82], as evidenced by a series of publications dedicated to these topics. In part of the cited works, stochastic components are taken into account in the optimizatino problem, leveraging the exploratory behavior of RL algorithms, as it will be further discussed in Section 2.4.2.

Training NNs with comprehensive datasets that span extensive areas of the problem domain, similar to BC, or in variable simulated settings, akin to RL, adds an inherent robustness to the derived policies. This can enhance the methods' performance under atypical conditions. Despite this, the significant computational resources required for training and the lack of formal assurances regarding the policies' effectiveness and robustness continue to restrict their practical use to simpler problems.

2.3 From deterministic to stochastic representation

The optimization techniques for deterministic OCPs, reviewed in the previous section, could be extend to deal with stochastic OCPs where elements like the state, dynamical system, and control may exhibit non-determinstic behaviour. The fundamental components of a deterministic OCP — namely the cost function, ODEs, and path and boundary constraints — are retained in robust formulations. However, transitioning to a Stochastic Optimal Control Problem (SOCP) necessitates accommodating intrinsic and theoretical distinctions. This section provides a comprehensive overview of the requisite modifications that facilitate this transition, delineating how traditional deterministic frameworks are adapted to embrace the uncertainties inherent in stochastic environments.

2.3.1 State modeling

The primary element of distinction from a deterministic OCP is the characterization of the state vector $\boldsymbol{x}(t)$ as a multivariate random variable, characterized by a Gaussian probability distribution. Consequently, the state distribution is described at any time by the *mean* $\boldsymbol{\mu}(t) \in \mathbb{R}^n$ (or *expected value*), defined as

$$\boldsymbol{\mu}(t) = \mathbb{E}\left[\boldsymbol{x}(t)\right] = \left(\mathbb{E}\left[x_1(t)\right], \mathbb{E}\left[x_2(t)\right], \ldots, \mathbb{E}\left[x_n(t)\right]\right)^T \tag{2.46}$$

and the *covariance matrix* $\boldsymbol{P}(t) \in \mathbb{R}^{n\times n}$, that is defined as

$$\boldsymbol{P}(t) = \mathbb{E}\left[\left(\boldsymbol{x}(t) - \boldsymbol{\mu}(t)\right)\left(\boldsymbol{x}(t) - \boldsymbol{\mu}(t)\right)^T\right] \tag{2.47}$$

The stochastic nature of the state vector demands a comprehensive probabilistic framework, so that the behavior of the system can be represented not just by one trajectory, but rather by a spectrum of possible outcomes, each weighted by its likelihood. This essential modification allows to account for uncertainties that are typical of space missions, as an exact precise knowledge of the spacecraft's state during flight is unattainable. The state level of accuracy is higher during phases

where precise navigation is available, or lower during stages where OD becomes more challenging due to factors like sensor limitations, environmental perturbations, or other operational constraints (such as cost reduction).

Considering the state as a random variable, drastically changes the formulation of boundary constraints as assertions on $\boldsymbol{x}(t)$ pertain only to individual realizations of the state probability distribution, rendering them inapplicable and meaningless within a robust framework, where the state is exclusively characterized by its stochastic attributes. Therefore, deterministic initial and terminal constraints as in Eqs. (2.4) and (2.5) in this new context would be substituted for example by

$$\Pr\left\{\boldsymbol{x}(t_0) \in \boldsymbol{\chi}_0\right\} \geqslant \alpha \tag{2.48}$$
$$\Pr\left\{\boldsymbol{x}(t_f) \in \boldsymbol{\chi}_f\right\} \geqslant \alpha \tag{2.49}$$

where $\boldsymbol{\chi}_0$ and $\boldsymbol{\chi}_f$ are n-dimensional generic polytopes to constrain the initial and final state, while $\alpha \in [0,\, 1]$ is the compliance rate of the condition in argument. Assuming a Gaussian distribution for $\boldsymbol{x}(t)$, an alternative stochastic constraint is instead

$$\mathbb{E}\left[\boldsymbol{x}(t_0)\right] = \boldsymbol{\mu}(t_0) = \widetilde{\boldsymbol{\mu}}_0 \tag{2.50}$$
$$\mathbb{E}\left[\boldsymbol{x}(t_f)\right] = \boldsymbol{\mu}(t_f) = \widetilde{\boldsymbol{\mu}}_f \tag{2.51}$$

where $\widetilde{\boldsymbol{\mu}}_0$ and $\widetilde{\boldsymbol{\mu}}_f$ are assigned initial and final mean state. To perfectly match the deterministic conditions however, information on the state dispersion should be also included alongside Eqs. (2.50) and (2.51). Specifically, to replicate the nature of deterministic constraints, which inherently exclude any degree of dispersion, the enforcement of zero dispersion becomes necessary, thus $\boldsymbol{P}(t_0) = \boldsymbol{P}(t_f) = \boldsymbol{0}$. In a robust formulation, additional information on the stochastic moments specific for the mission are added, e.g.

$$\boldsymbol{x}(t_0) \sim \mathcal{N}\left(\widetilde{\boldsymbol{\mu}}_0,\, \widetilde{\boldsymbol{P}}_0\right) \tag{2.52}$$
$$\lambda_{\max}\left(\boldsymbol{P}(t_f)\right) \leqslant \delta \tag{2.53}$$

where $\mathcal{N}\left(\boldsymbol{\mu},\, \boldsymbol{P}\right)$ is a Gaussian distribution of mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{P}$, $\widetilde{\boldsymbol{P}}_0$ is an assigned initial covariance matrix, $\lambda_{\max}\left(\cdot\right)$ denotes the maximum eigenvalue of the matrix in argument, and $\delta \in \mathbb{R}$.

Unlike deterministic OCPs, where constraints are typically applied directly to optimization variables or to quantities that have a straightforward dependency on these variables, the stochastic nature of the state introduces further complexities. In this respect, directly applying constraints to random variables would be inadequate, as this approach would result in a computed control that fails to ensure compliance in every possible execution of the flown trajectory. Therefore, it is the necessary to formulate path constraints on stochastic quantities, which provide the most accurate description of the system. This shift underscores the need for innovative computational strategies to effectively manage and apply path constraints within the inherently uncertain framework of robust optimization. In scenarios where direct evaluation of stochastic quantities is infeasible, uncertainty propagation tools emerge as essential instruments. These tools are designed to estimate stochastic parameters and play a pivotal role in robust optimization frameworks. A detailed exploration of the principal methods for uncertainty propagation, including their methodologies and applications, is provided in Chapter 3.

2.3.2 Stochastic Differential Equations

Where the knowledge of system dynamics is partial or imprecise, the use of deterministic ODEs may inevitably lead to random deviation of the predicted trajectory from the actual one. This discrepancy arises either because the model employed does not adequately capture the intricacies of the actual dynamics or because it fails to account for the system's aleatory components. Thus, the shift towards the SOCP paradigm is driven by the need to account for and manage these uncertainties directly within the problem formulation, in order to obtain a reliable and accurate approach to trajectory optimization under uncertain conditions.

The presence of unmodeled dynamics, as in the case of unknown harmonic elements of a planet, or a complex dynamical environment, as a highly irregular shape of a target celestial body, dedicated analyses are required to bridge these knowledge gaps. However, even when a reasonably accurate dynamical model is accessible, minor random disturbances can persist. These fluctuations may arise from a variety of factors, ranging from structural complexities to navigation inaccuracies. This reasoning highlight the need for a detailed and comprehensive approach to modeling and analyzing the possible perturbations, ensuring the reliability of the mission's trajectory planning is maintained even in the presence of these disturbances.

One of the key factors in this regard is the use of Stochastic Differential Equations (SDEs) in place of the deterministic ODEs. An SDE is made of a deterministic term and stochastic component, leading to a solution that is itself a stochastic process. The most common form in literature is represented by an ODE with the right hand side perturbed by a term dependent on a white noise variable, this latter usually represented by a Wiener process. Two fundamental approaches are possible in literature to treat and solve SDEs: Itô and Stratonovich calculus, respectively named after the mathematicians Kiyosi Itô (1915-2008) and Ruslan Stratonovich (1930-1997). Itô calculus is founded on the principle of causality, a concept particularly relevant in time-based applications. Conversely, Stratonovich calculus aligns more closely with traditional calculus rules and possesses inherent geometric properties, making it better suited for geometric problems like random motion on manifolds [83]. To integrate a perturbation component into a deterministic ODE, thereby adjusting the Equations of Motion (EoM) to reflect a slightly perturbed dynamical model, the standard formulation of SDEs as outlined in Itô calculus, featuring a singular perturbation term, is employed. Equation (2.1) is thus substituted by

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}(t),\, \boldsymbol{u}(t),\, t)\mathrm{d}t + \boldsymbol{g}(\boldsymbol{x}(t),\, \boldsymbol{u}(t),\, t)\mathrm{d}\boldsymbol{w} \tag{2.54}$$

where $\boldsymbol{g}\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \mapsto \mathbb{R}^n \times \mathbb{R}^{n_w}$ is the diffusion term, which serve as a weight for the white noise vector $\boldsymbol{w}(t) \in \mathbb{R}^{n_w}$, modeled as a Wiener process characterized by the following properties [83]:

1. $\boldsymbol{w}(0) = \mathbf{0}_{n_w}$ almost surely[1];
2. For every $t > 0$, the future increments $\boldsymbol{w}(t+\delta t) - \boldsymbol{w}(t)$, $\delta t \in \mathbb{R}_+$, are independent of the past values $\boldsymbol{w}(t')$, with $t' < t$;
3. $\boldsymbol{w}(t + \delta t) - \boldsymbol{w}(t)$ is normally distributed with mean $\mathbf{0}_{n_w}$ and covariance matrix $\delta t \boldsymbol{I}_{n_w}$, where $\boldsymbol{I}_{n_w}$ is the identity matrix with dimension n_w;
4. $\boldsymbol{w}(t)$ is almost surely continuous in t.

[1] In probability theory, an event that *almost surely* happens is equivalent to an event that happens with probability 1.

A number of numerical methods to solve SDEs are available. The most common solution scheme is the Euler-Maruyama method [84], that is an extension of the Euler method for ODEs to SDEs. Extensions of the Runge-Kutta methods [84, 85] and Rosembrock methods [86] to SDEs also exist in literature. Alternatively, the Milsten method involves a more sophisticated formulation based on the derivative of the diffusion term with respect to $\boldsymbol{x}(t)$ [87].

A critical aspect when dealing with SDEs is that numerical integration produces a different outcome for each execution of a same algorithm, attributed to the inherently random nature of certain terms within the equations. Such randomness ensures that outcomes differ with each execution. In MC analysis, this variability is not a problem, as the analysis is designed to leverage a large number of runs, each with independently random outcomes, to statistically infer system behavior. However, within an optimization framework, the presence of random perturbations in a trajectory, which may not be consistent across iterations, thus posing some challenges on convergence of the optimization method. As an example, the performance index varies on unpredictable way, leading to errors when computing gradients.

2.3.3 Cost function and constraints

The stochastic nature of the state variable and of the EoM immediately impacts all facets of the SOCP, mirroring challenges observed with the non-uniqueness in SDEs numerical integration. Optimizing performance indexes tied to specific trajectories or random factors becomes particularly complex, veering towards theoretical irrelevance under traditional deterministic models. This highlights a fundamental shift in approach, necessitating a reevaluation of optimization strategies within SOCP. In the context of aerospace applications, cost functions that rely solely on an OL control, such as cumulative Δv, remain unaffected by stochastic influences because they do not depend on any stochastic variables. When dealing with a CL control law that depends on real-time states with stochastic behavior, the associated cost function becomes inherently variable, contingent on the actual trajectory flown. In such cases, adapting a common cost function like cumulative Δv for use within a robust optimization framework becomes complex. This is because the cost function must now accommodate fluctuations in trajectory due to the probabilistic nature of the state variables, making the straightforward application of deterministic methods less feasible.

Anticipating the key feature of *chance-constraint* methods, a possible approach to formulate a cost function with stochastic components is to express it in probabilistic sense. In other words, since it is not possible to know in advance the actual final value of the cost function, an idea could be to minimize (or maximize) a quantity that with a certain degree of confidence is smaller (or greater) than the actual cost function for a particular realization of the trajectory. Formally, given the vector of the control parameters $\boldsymbol{\Theta}$ and the related cost function $\Phi(\boldsymbol{\Theta})$,

$$\Pr\{\Phi(\boldsymbol{\Theta}) < J\} \geqslant \alpha \tag{2.55}$$

where J is the quantity to minimize. By definition, J represents the p-th percentile of $\Phi(\boldsymbol{\Theta})$, denoted with $Q_p(\Phi(\boldsymbol{\Theta}))$, with $p = 100\alpha$. The aim is thus to optimize the control parameters as to minimize $Q_p(\Phi(\boldsymbol{\Theta}))$. Denoting the optimal control vector with $\boldsymbol{\Theta}^*$, the related cost function $\Phi(\boldsymbol{\Theta}^*) = \Phi^*$ is such that

$$\Pr\{\Phi^* < Q_p(\Phi^*)\} \geqslant \alpha \tag{2.56}$$

The final task is to reformulate the p-th percentile in a manner that is compatible with optimization algorithms. This involves translating the percentile into a quantifiable

metric that can be systematically assessed and manipulated within the framework of the algorithm, ensuring it can be effectively optimized as part of the solution process. This step is explained in Chapter 5 resorting to a mathematical manipulation of Eq. (2.56).

Path constraints that depend on stochastic variables also follow a similar approach to integration within optimization frameworks. These constraints, which define acceptable limits or conditions on the paths taken by system variables, must be reinterpreted when those variables exhibit random behavior. For instance, in case of the control magnitude to be constrained below a given threshold, the probabilistic expression to manipulate is, similarly to Eq. (2.56),

$$\Pr\left\{\|\boldsymbol{u}(t)\| \leqslant \widetilde{u}_{\max}\right\} \geqslant \alpha \qquad t \in [t_0,\, t_f] \tag{2.57}$$

In this case as well, by resorting to a reformulation of the p-th percentile of the control magnitude, the constraint is converted to a treatable form for the optimization algorithm.

Leveraging the aforementioned transformations, a general statement for an SOCP could be given by

$$\min_{\boldsymbol{\Theta}} Q_p\left(\Phi\left(\boldsymbol{\Theta}\right)\right) \tag{2.58a}$$

$$\text{s.t.} \quad \mathrm{d}\boldsymbol{x} = \boldsymbol{f}\left(\boldsymbol{x},\, \boldsymbol{u},\, t\right)\mathrm{d}t + \boldsymbol{g}\left(\boldsymbol{x},\, \boldsymbol{u},\, t\right)\mathrm{d}\boldsymbol{w} \qquad \forall t \in [t_0,\, t_f] \tag{2.58b}$$

$$\boldsymbol{\chi}\left(\boldsymbol{x}_0,\, \boldsymbol{x}_f,\, \boldsymbol{u}_0,\, \boldsymbol{u}_f,\, t_0,\, t_f\right) = \boldsymbol{0} \tag{2.58c}$$

$$\boldsymbol{\psi}\left(\boldsymbol{x},\, \boldsymbol{u},\, t\right) \leqslant \boldsymbol{0} \qquad \forall t \in [t_0,\, t_f] \tag{2.58d}$$

where explicit time dependency of the state and control variable have been omitted, and $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$, $\boldsymbol{x}(t_f) = \boldsymbol{x}_f$, $\boldsymbol{u}(t_0) = \boldsymbol{u}_0$, $\boldsymbol{u}(t_f) = \boldsymbol{u}_f$. Note that in Eqs. (2.58c) and (2.58d), although the same notation of the deterministic OCP is used, the probabilistic expressions discussed in this subsection are included.

2.4 Methods for robust trajectory optimization

This section concerns the diverse methodologies proposed in the literature for addressing an SOCP. While the formulation provided in Eqs. (2.58) offers a broad framework, the field of robust trajectory optimization allows for multiple interpretations and approaches, each contributing to ongoing and vigorous research. As a relatively nascent area of study, robust trajectory optimization is still in its developmental infancy, with no universally accepted methodology established to date, in contrast to the more mature field of traditional OCPs. This dynamic and evolving landscape underscores the potential for significant advancements and the establishment of foundational strategies in the years to come.

2.4.1 Chance-constraint formulation

Chance-constraint optimization is a mathematical approach designed to manage uncertainties within optimization frameworks. This methodology ensures that constraints are adhered to with a specific probability, accommodating uncertainties by integrating them directly into the optimization process. The essence of chance-constrained optimization lies in its ability to balance the trade-off between risk and operational performance, making it particularly suited for applications where parameter values are uncertain and can only be described probabilistically.

The foundational concept of chance-constrained optimization was introduced by Charnes and Cooper in 1959 [88] and Millen and Wagner in 1965 [89], who explored the implications of incorporating probabilistic constraints into linear programming models. This pioneering work has led to widespread adoption and development across various disciplines, including finance, supply chain management, and engineering. Further developments in the field were significantly advanced by Prekopa in 1995 [90], who discussed the broader implications and models of stochastic programming involving chance constraints, providing a deep theoretical foundation and practical examples of applications. Another notable contribution came from Nemirovski and Shapiro [91], who focused on the convex approximations of chance constraints, offering insights into efficient computational strategies and enhancing the feasibility of applying these models in real-world scenarios.

There exist multiple approaches to structuring chance constraints within optimization problems, each with distinct implications for accuracy and feasibility. The various formulations offer specific advantages and potential limitations concerning the precision of solutions and the overall tractability of the problem. Consider an optimization problem under uncertainty:

$$\min_{\boldsymbol{x}} \Phi\left(\boldsymbol{x}, \boldsymbol{\xi}\right) \tag{2.59a}$$

$$\text{s.t.} \quad g\left(\boldsymbol{x}, \boldsymbol{\xi}\right) = 0 \tag{2.59b}$$

$$h\left(\boldsymbol{x}, \boldsymbol{\xi}\right) \geqslant 0 \tag{2.59c}$$

where Φ represents the objective function to be minimized, g encapsulates the set of equality constraints, and h embodies the inequality constraints. The vector $\boldsymbol{x}$ comprises the decision variables, while $\boldsymbol{\xi}$ is the vector of random variables, encompassing all elements of uncertainty within the problem.

- *Individual chance constraints* represent a straightforward transformation of the original constraints $h_1, h_2, \ldots, h_n$. These are expressed as:

$$\Pr\left(h_1\left(\boldsymbol{x}, \boldsymbol{\xi}\right) \geqslant 0\right) \geqslant 1 - \epsilon_1 \tag{2.60}$$

$$\Pr\left(h_1\left(\boldsymbol{x}, \boldsymbol{\xi}\right) \geqslant 0\right) \geqslant 1 - \epsilon_2 \tag{2.61}$$

$$\vdots$$

$$\Pr\left(h_1\left(\boldsymbol{x}, \boldsymbol{\xi}\right) \geqslant 0\right) \geqslant 1 - \epsilon_n \tag{2.62}$$

  where $\epsilon_i \in [0, 1]$, $i = 1, \ldots, n$, are the violation rates. In this configuration, each individual constraint h_i is associated with its distinct tuning parameter ϵ_i. This method's benefit is the possibility to allocate different probabilities to the constraints, reflecting their varying levels of criticality. Moreover, it provides immediate insights into which specific constraint was breached upon a violation. Nonetheless, individual chance constraints only guarantee that each constraint is independently satisfied to its assigned confidence level, rather than ensuring collective compliance.

- *Collective chance constraints* demand the simultaneous satisfaction of all inequality constraints within a combined probability threshold:

$$\Pr\left(h_1\left(\boldsymbol{x}, \boldsymbol{\xi}\right) \geqslant 0 \wedge h_2\left(\boldsymbol{x}, \boldsymbol{\xi}\right) \geqslant 0 \wedge \cdots \wedge h_n\left(\boldsymbol{x}, \boldsymbol{\xi}\right) \geqslant 0\right) \geqslant 1 - \epsilon \tag{2.63}$$

  Assigned a collective parameter $\epsilon \in [0, 1]$, this model ensures that all constraints are met at a given probability level. The logic parallels that of deterministic

constraints, where we anticipate all conditions to be valid simultaneously with absolute certainty. This leads to a more conservative problem setup. However, the primary drawback is the computational complexity these constraints introduce, which limits their application to more straightforward or specialized scenarios.

Chance-constrained optimization often presents computational challenges, predominantly due to the difficulty in calculating the integral for the probability,

$$\Pr\left(h\left(\boldsymbol{x},\,\boldsymbol{\xi}\right)\geqslant 0\right)=\int_{\{\xi|h(\boldsymbol{x},\boldsymbol{\xi})\geqslant 0\}} p_{\xi}(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi} \tag{2.64}$$

where p_ξ is the Probability Density Function (PDF) of $\boldsymbol{\xi}$. This difficulty arises primarily when the PDF is complex or intractable, a situation that is not uncommon especially when the model establishes a nonlinear interplay between uncertainties in inputs and outputs. In these cases, approximation methods become essential. Some of the most prevalent methods include:

- **Deterministic transformation:** This technique simplifies the problem by transforming it into a deterministic format, where the chance constraints are expected to be satisfied with a stipulated violation rate ϵ. Mathematical bounds on ϵ can be derived, with references such as Calafiore and Campi [92] offering insight.
- **Back-mapping:** Bypassing the direct computation of the PDF, this method finds an equivalent representation via a monotonic relationship between uncertainties in input and output, enabling integration within the subspace of input uncertainties. This mapping is then used to facilitate easier integration [93]. While advantageous, the limitation lies in the potential complexity of these monotonic relations.
- **Polynomial Chaos Expansion (PCE):** A more contemporary approach where random variables are represented in terms of a polynomial function of other random variables. The polynomials are chosen to be orthogonal with respect to the joint probability distribution of these random variables. The merit of this representation is the direct accessibility to the moments of the approximated variables, negating the need for further relaxation or discretization. However, such methods may escalate rapidly in complexity with increasing model uncertainties [94, 95]. Refere to the dedicated subsection for mathematical details 3.2.2.

Chance-constraint programming has seen broad application across disciplines, affirming its versatility and effectiveness [96]. Its utility is currently most notable in robotic maneuvering. This method proves especially beneficial in guiding autonomous agents through environments laden with unpredictable obstacles, enhancing navigation and strategic planning within variable movement spaces [97, 98, 99]. Model Predictive Control (MPC) found also applicability for chance-constraint programming [100, 101]. Aerospace applications of this methodology have been instead coupled in the works of Tsiotras et al. with convex optimization [102, 103] and covariance steering [104].

2.4.2 Machine learning approach

In Section 2.2.3, the fundamental concepts behind AI and its potential use for deterministic trajectory optimization were explored, highlighting in the particular the main results obtained with RL and DRL. More recently, the application of these methods to robust control optimization has yielded promising results, showcasing their potential to revolutionize strategies in unpredictable and dynamic settings. This emerging approach couples the adaptive capabilities of AI with the need for resilience against uncertainties, positioning DRL as a key player in crafting robust strategies that can withstand and adapt to unpredictable variables in dynamic environments.

DRL algorithms represent a significant departure from traditional control methods, particularly in their adeptness at managing stochastic control problems. These algorithms are uniquely equipped to process transition probabilities and observation models that are often presented in ambiguous (or even black-box) formats. This flexibility allows DRL to function without the stringent requirement for detailed mathematical formulations typically necessary in conventional approaches. The exploratory nature inherent in RL enables these algorithms to naturally adapt and develop resilience against uncertainties within the models. This characteristic is crucial in environments where precise predictions are challenging, allowing the system to maintain performance despite variable conditions. Furthermore, a major advantage of DRL lies in its computational architecture. The most demanding computational tasks, such as the extensive training of deep neural networks, are conducted during the pre-flight phase. This phase leverages high-performance computing systems capable of handling large datasets and complex learning algorithms efficiently. Once this phase is complete, the operational demands on the spacecraft's onboard systems are relatively minimal. During flight, the execution of deep-RL involves just a single pass through the trained neural network for each guidance step, which is remarkably fast and ensures minimal delay in response times. This efficiency makes DRL particularly suitable for real-time applications in aerospace contexts, where decisions must be made swiftly and reliably to navigate through dynamic and potentially hazardous environments.

When addressing SOCPs within the realm of RL, the challenge is typically reformulated as a Markov Decision Process (MDP). Within an MDP framework, at each discrete timestep $i = 0, \dots, M$, $M \in \mathbb{N}$, the decision-making agent selects an action or control $\boldsymbol{u}_i$, from the set of feasible actions, based on observations $\boldsymbol{y}_i$ of the system's current state $\boldsymbol{x}_i$. This selection is computed by a CL control policy: $\boldsymbol{u}_i = \boldsymbol{\pi}(\boldsymbol{y}_i)$. The consequence of each action transition to a new state $\boldsymbol{x}_{i+1}$, and yields a scalar reward, $R_i = R(\boldsymbol{x}_i, \boldsymbol{u}_i, \boldsymbol{x}_{i+1})$, which reflects the desirability of the state transition. MDPs uniquely satisfy the Markov property, implying that the subsequent state is dependent solely on the present state and the executed action, not on the sequence of preceding states and actions. The overall objective in an MDP is to discover the control policy $\boldsymbol{\pi}^*$ that maximizes the expected return of the discounted rewards along a trajectory $\tau = \{(\boldsymbol{x}_0, \boldsymbol{u}_0), (\boldsymbol{x}_1, \boldsymbol{u}_1), \dots, (\boldsymbol{x}_M, \boldsymbol{u}_M)\}$, formalized as:

$$J(\boldsymbol{\pi}) = \mathbb{E}_{\tau \sim \boldsymbol{\pi}}[G(t)] \tag{2.65}$$

$$G(t) = \sum_{i=0}^{M} \gamma^i R_i \tag{2.66}$$

where $\gamma \in (0, 1)$ is a discount factor that diminishes the value of future rewards. Here, $\mathbb{E}_{\tau \sim \pi}[\cdot]$ represents the expected value over multiple trajectories τ accrued

from interactions with the environment under policy $\boldsymbol{\pi}$.

Several research papers have already dealt with the use of DRL for robust CL guidance of spacecraft during planetary landing maneuvers [105] and proximity operations [106], as well as for cislunar [107] and interplanetary [75] trajectory design.

Integrating DRL algorithms with a recurrent network structure gives rise to a sophisticated model often known as meta-RL, or the concept of "learning to learn." [108] These Recurrent Neural Networks (RNNs), with their feedback loops, have the innate capacity to retain temporal information within their internal states. This feature empowers RNNs to finely tune their outputs to the nuances of the current problem scenario, markedly enhancing their efficacy in environments characterized by non-Markovian properties, partial observability, or multifaceted tasks that require a broader contextual understanding. An RNN-based meta-RL framework has been effectively utilized for the onboard guidance of spacecraft or landers operating within uncertain dynamic environments [105, 109], partially observable settings [82, 110, 111], or scenarios requiring the execution of various tasks [80]. This approach has consistently surpassed the performance of standard RL techniques that rely on a fully-connected policy network, showcasing its superior adaptability and proficiency in complex navigational challenges.

2.4.3 Classifications of methods

The research field of SOCP is currently expanding, standing in stark contrast to the mature field of deterministic control, where methodologies have been thoroughly investigated, and their efficacies extensively documented. In the realm of SOCP, research is actively evolving, with standardizations and benchmarking yet to be solidified. As the field of robust trajectory optimization continues to mature, it becomes increasingly apparent that a classification based on foundational elements is more instructive than trying to delineate distinct methods. Besides methods based on ML that are objectively unique in their formulation, the critical components that define a robust methodology include stochastic approaches, the means of UP, and problem formulation. The literature on robust trajectory optimization reflects a plethora of methodologies, each interlacing the aforementioned fundamental elements. A comprehensive evaluation of how each element affects the overall efficiency of a proposed methodology is yet to be acquired by the scientific community. As the body of SOCP techniques continues to expand, the need for a systematic classification becomes ever more crucial.

A first classification relates to the way state uncertainty, cost function, and constraints are manipulated. This is a crucial factor ingredient for any SOCP solution method and will impact the overall accuracy and computational efficiency of the approach. This classification can be synthesized as follows:

- Stochastic Manipulation (SM): probabilistic expressions of constraints and cost functions are converted in equivalent or nearly-equivalent expressions suitable for numerical solvers (see Sec. 2.4.1);
- PCE: method based on representing a random variable in terms of a orthogonal polynomial functions of other random variables (see Sec. 3.2.2);
- Belief MDP: the problem is formulated as an MDP with stochastic state;
- General Stochastic Formulation (GSF): stochastic elements of the the state are only considered in this approach. This is typically possible in situation

where the limitation of state dispersion is the principal objective for the scopes of the mission.

UP is a pivotal element in robust trajectory optimization, primarily affecting the computational intensity rather than the robustness of the optimized control law itself, assuming the same level of accuracy. Although a more detail and advanced explanation of these techniques is included in the next chapter, UP tools are principally divided into *linear* and *nonlinear* methods. Linear methods typically assume small deviations from a nominal trajectory and rely on linear approximations of the dynamics. These methods often offer the benefits of reduced computational complexity and simpler analytical insight, making them well-suited for quick assessments and problems with modest uncertainty. Nonlinear methods, on the other hand do not make such linear assumptions and can handle larger deviations and more complex uncertainty models. While these methods provide a more accurate depiction of the potential outcomes, they come with increased computational demands.

The choice on how to solve the SOCP depends on the specific application and may play a crucial role on the final solution performance. The main approaches adopted in literature so far are the following:

- Convexification: transformation of a non-convex problem into an equivalent (lossless) or relaxed (successive) convex problem, as covered in Sec. 2.2.2;
- Differential Dynamic Programming (DDP): technique used for solving nonlinear optimal control problems. It operates by locally approximating the dynamics of the system and the cost function using second-order Taylor series expansions. The primary advantage of DDP is its ability to efficiently handle high-dimensional systems with complex dynamics. At each iteration, DDP solves a second-order expansion of the Bellman equation to find the local optimal control, resulting in a locally optimal control law and associated value function. The updates from these subproblems are then used to improve the trajectory in a process similar to Newton's method for optimization [112];
- Numerical algorithms: problem formulated in a OCP fashion without resorting to any of the previous technique to manipulate the general formulation. The OCP-like problem is solved directly via the classical iterative numerical algorithms, e.g., interior point, sequential quadratic programming;

Chance-constraint optimization is the most renowned form of SM, and ensures that constraints are adhered to with a specific probability. This method allows the numerical solver to easily enforce the probabilistic constraints, which would otherwise be challenging to address. Based on stochastic manipulation, several studies have employed convex optimization. The polynomial-time convergence of convex optimization to the global optimum is a significant advantage, particularly for the computationally challenging domain of robust optimization. In the works of Ridderhof et al. [102, 103, 113] and Benedikter et al. [114, 115] covariance steering is performed resorting to convex optimization, while in the works of Oguri et al. [116, 117] and Lew et al. [97] sequential convex programming is used for trajectory optimization and motion planning. In the paper of Xiong et al. [118], polynomial chaos expansion is instead coupled with convex optimization for a landing problem. DDP is employed in the works of Ozaki et al. [119, 120], based on chance-constraint SM.

In the work of Giordano and Topputo [121] polynomial chaos is used to handle the state uncertainty, proposing an integrated approach for preliminary mission analysis,

embedding in the trajectory design and optimization the navigation assessment and the associated stochastic costs. Polynomial chaos has been also applied for robust trajectory design in a three-body system by Feldhacker et al. [122], whose work has been improved by Yang et al. [29] et al. including orbital re-planning.

A belief-based approach is proposed by Greco et al. [123, 124], where an optimal control problem is formulated directly in terms of uncertainty distributions, called beliefs, rather than on realizations of the system state, as in classic deterministic optimal control. This work has been subsequently extended by including a nonlinear navigation analysis under both aleatoric and epistemic uncertainty [125].

The main works that combine the mentioned elements are summarized in Table 2.1, where notations "L" and "NL" stand for respectively linear and nonlinear UP.

Table 2.1. SOCP classification.

	Numerical algorithms	Convexification	DDP
SM	L: [126]	L: [97, 102, 116, 117, 115]	NL: [119, 120]
PCE	NL: [121, 127]	L: [128]	-
Belief MDP	NL: [123]	-	-
GSF	L: [129] NL: [29]	L: [46, 130]	L: [131]

As a minor methodology for the resolution of a SOCP, indirect methods were recently employed in robust optimization. An extension of the primer vector theory to stochastic optimization is proposed by Zimmer et al. [132] for an OL control law, while Oguri et al. [126] extended this approach by including path constraints. An indirect formulation is also employed by Xin et al. in computing a CL control in linear dynamics for proximity operation [133]. A multi-objective optimization is instead discussed by Jenson and Scheeres using cost function with covariance and energy terms [130]. In the work of Heidrich and Holzinger [134], a relaxation approach is adopted to tackle indeterminate control behaviour for singular arcs by means of augmented covariance dynamics, allowing an application to optical control problem with uncertainties. Despite the renowned advantages of an indirect formulation, i.e. fast convergence and reduced computational cost, this class of methods may suffer from limited convenience for advanced trajectory optimization problems. The analytical expressions required in the indirect methods, as state and co-state derivatives, increase the complexity of the problem formulation, reducing the versatility to a confined range of possible mission scenarios.

A further approach that is worth of notation is the formulation of the SOCP in an MPC fashion. MPC is an advanced method of process control that utilizes an explicit dynamic model of the process to predict the future state of the system. At each step, an optimization problem is solved to find the optimal control action by minimizing a cost function over a future time horizon. The optimization considers the current state of the system and the predicted future states. However, only the first control action is implemented, and the process is repeated at the next time step with updated measurements [135]. MPC initially emerged in the 1980s as a control technique for the petroleum refinery industry [136]. Its effectiveness has since been demonstrated across a wide spectrum of sectors, with its applications extending to the aerospace domain among others [137, 138, 139]. MPC has been also extended to robust applications in numerous works [140, 141, 142, 143].

Chapter 3

State uncertainty propagation tools

In the context of space trajectory planning and operations, uncertainty propagation typically involves calculating the PDF or the statistical moments (e.g., mean or covariance matrix) of the spacecraft state

In addition to the context of robust optimization, uncertainty propagation is an indispensable also in the field of Space Situational Awareness (SSA) — a critical domain dedicated to the comprehensive monitoring, characterization, and tracking of space objects. Efficient and precise long-term uncertainty propagation in orbital trajectories is a significant challenge in SSA. This stems from the reality that Resident Space Objects (RSOs) — often defined as those with dimensions of a softball (approximately 10 cm) or larger-vastly outnumber the available tracking sensors. Consequently, tracking any single object is typically based on infrequent observations. Therefore, maintaining a trajectory's uncertainty over several days or multiple orbital periods without new measurements becomes a critical requirement in SSA operations. The previous chapters highlighted the central role of accurately predicting the evolution of state dispersion over the flight time, which is fundamental for the development of a robust control. Indeed, an accurate knowledge of the spacecraft state empowers the algorithm to design a control law tailored to the envisaged perturbations/uncertainties, thereby enhancing the mission efficiency and reliability.

During the latter half of the 20th century, orbital uncertainty propagation was typically tackled using linear models or nonlinear MC simulations. Linear methods, which simplify the problem by linearizing it, offer high efficiency but their accuracy diminishes when applied to systems with significant nonlinearity or over extended durations. Conversely, MC simulations deliver precise outcomes but at a considerable computational cost. To circumvent these issues, a variety of analytical and semi-analytical methods for handling nonlinear orbital uncertainty have been formulated in more recent times.

3.1 Linear methods

Linear methods propagate the initial uncertainty distribution using a linearized model of the dynamics, by leveraging on the so-called State Transition Matrix (STM). These methods rely on the assumption that the linearized model adequately captures the dynamics of trajectories close to a nominal path, and that the uncertainty is fully represented by a Gaussian probability distribution. With these assumptions,

the propagation process reduces to evaluating the mean and covariance matrix at any time.

3.1.1 Local linearization

Assuming a random vector $\boldsymbol{x}(t)$, with PDF $p_x\left(\boldsymbol{x}(t),\, t\right)$, subject to the SDE

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}(t),\, t)\mathrm{d}t + \boldsymbol{g}(\boldsymbol{x}(t),\, t)\mathrm{d}\boldsymbol{w} \tag{3.1}$$

as in Eq. (2.54) without the control dependency, the deterministic component $\boldsymbol{f}\left(\boldsymbol{x}(t),\, t\right)$ of the EoM is approximated with the first-order Taylor expansion, along a reference trajectory $\hat{\boldsymbol{x}}(t)$, as

$$\boldsymbol{f}\left(\boldsymbol{x}(t),\, t\right) \approx \boldsymbol{f}\left(\hat{\boldsymbol{x}}(t),\, t\right) + \boldsymbol{A}\left(\boldsymbol{x}(t),\, t\right)\delta\boldsymbol{x}(t) \tag{3.2}$$

where

$$\delta\boldsymbol{x}(t) = \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t) \tag{3.3}$$

$$\boldsymbol{A}\left(\boldsymbol{x}(t),\, t\right) = \left.\frac{\partial\boldsymbol{f}\left(\boldsymbol{x}(t),\, t\right)}{\partial\boldsymbol{x}(t)}\right|_{\hat{\boldsymbol{x}}(t)} \tag{3.4}$$

The time derivative of the state error $\delta\boldsymbol{x}$ is thus

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta\boldsymbol{x}(t) = \boldsymbol{A}\left(\boldsymbol{x}(t),\, t\right)\delta\boldsymbol{x}(t) + \boldsymbol{g}(\boldsymbol{x}(t),\, t)\dot{\boldsymbol{w}} \tag{3.5}$$

where the superscript dot naturally represents the time derivative. The analytical solution is hence

$$\delta\boldsymbol{x}(t) = \boldsymbol{\Phi}\left(t,\, t_0\right)\delta\boldsymbol{x}(t_0) + \boldsymbol{g}(\boldsymbol{x}(t),\, t)\boldsymbol{w}(t) \tag{3.6}$$

where $\boldsymbol{\Phi}\left(t,\, t_0\right)$ is the STM of $\boldsymbol{f}\left(\boldsymbol{x}(t),\, t\right)$ from t_0 to t. Mean state and covariance matrix are hence expressed as

$$\boldsymbol{\mu}(t) = \boldsymbol{\Phi}\left(t,\, t_0\right)\boldsymbol{\mu}(t_0) \tag{3.7}$$

$$\boldsymbol{P}(t) = \boldsymbol{\Phi}\left(t,\, t_0\right)\boldsymbol{P}(t_0)\boldsymbol{\Phi}\left(t,\, t_0\right)^T + \boldsymbol{g}(\boldsymbol{x}(t),\, t)\boldsymbol{g}(\boldsymbol{x}(t),\, t)^T \tag{3.8}$$

In case the covariance matrix of $\boldsymbol{w}(t)$ is $\boldsymbol{Q}(t) \neq \delta t\boldsymbol{I}_{n_w}$, Eq. (3.8) becomes

$$\boldsymbol{P}(t) = \boldsymbol{\Phi}\left(t,\, t_0\right)\boldsymbol{P}(t_0)\boldsymbol{\Phi}\left(t,\, t_0\right)^T + \boldsymbol{g}(\boldsymbol{x}(t),\, t)\boldsymbol{Q}(t)\boldsymbol{g}(\boldsymbol{x}(t),\, t)^T \tag{3.9}$$

$$\tag{3.10}$$

Among different other aerospace applications, local linearization has been applied in trajectory design for common orbital rendezvous problem [144] and powered lunar descending [145].

This method, with its analytical nature, offers straightforwardness and swift computational performance, making it a suitable tool for autonomous onboard mission planning. Its efficiency, however, diminishes when applied to systems with pronounced nonlinearity, during extended periods of uncertainty propagation, or when the initial uncertainty is substantial. Additionally, local linearization is not tailored for propagating uncertainties that excessively deviate from Gaussian distributions.

3.1.2 Quasi-linearization

This method revolves around the approximation of $\boldsymbol{f}(\boldsymbol{x}(t),\, t)$ with a semi-linear function [146], as

$$\boldsymbol{f}\left(\boldsymbol{x}(t),\, t\right) \approx \hat{\boldsymbol{f}}\left(\boldsymbol{x}(t),\, t\right) + \boldsymbol{N}\boldsymbol{x}(t) \tag{3.11}$$

resulting in

$$\delta\boldsymbol{x}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t),\, t\right) - \hat{\boldsymbol{f}}\left(\boldsymbol{x}(t),\, t\right) - \boldsymbol{N}\boldsymbol{x}(t) \tag{3.12}$$

where the function $\hat{\boldsymbol{f}}\left(\boldsymbol{x}(t),\, t\right)$ and the gain matrix $\boldsymbol{N} \in \mathbb{R}^{n\times n}$, named *describing functions*, are chosen as to minimize the mean square approximation error $J = \mathbb{E}\left[\delta\boldsymbol{x}\boldsymbol{S}\delta\boldsymbol{x}^T\right]$, where $\boldsymbol{S}$ is a symmetric positive semi-definite matrix. By setting the partial derivatives of J with respect to the elements of $\hat{\boldsymbol{f}}$ and $\boldsymbol{N}$ equal to zero, these latter result as

$$\hat{\boldsymbol{f}}\left(\boldsymbol{x}(t),\, t\right) = \mathbb{E}\left[\boldsymbol{f}\left(\boldsymbol{x}(t),\, t\right)\right] \tag{3.13}$$

$$\boldsymbol{N} = \mathbb{E}\left[\boldsymbol{f}\left(\boldsymbol{x}(t),\, t\right)\boldsymbol{x}^T\right]\boldsymbol{P}(t)^{-1} \tag{3.14}$$

where $\boldsymbol{P}(t)$ is the covariance matrix of $\boldsymbol{x}$. Due to the expectation operator in the previous expressions, solution of the describing functions requires the PDF of $\boldsymbol{x}$ to be known, thus it is most frequently assumed that $\boldsymbol{x}$ is Gaussian. Having that

$$\dot{\boldsymbol{\mu}} = \mathbb{E}\left[\dot{\boldsymbol{x}}\right] \tag{3.15}$$

$$\dot{\boldsymbol{P}} = \mathbb{E}\left[\dot{\boldsymbol{x}}\boldsymbol{x}^T + \boldsymbol{x}\dot{\boldsymbol{x}}^T\right] - \dot{\boldsymbol{\mu}}\boldsymbol{\mu}^T - \boldsymbol{\mu}\dot{\boldsymbol{\mu}}^T \tag{3.16}$$

where time dependency was omitted, the statistical differential equations of mean and covariance matrix are obtained as

$$\dot{\boldsymbol{\mu}} = \hat{\boldsymbol{f}}\left(\boldsymbol{x},\, t\right) \tag{3.17}$$

$$\dot{\boldsymbol{P}} = \boldsymbol{N}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{N}^T + \boldsymbol{g}(\boldsymbol{x},\, t)\boldsymbol{Q}\boldsymbol{g}(\boldsymbol{x},\, t)^T \tag{3.18}$$

where, as in previous section, $\boldsymbol{Q}$ is the covariance matrix of $\boldsymbol{w}(t)$ if different from $\delta t\boldsymbol{I}_{n_w}$.

Equation (3.18) is commonly recognized as the Covariance Analysis DEscribing function Technique (CADET). This method was introduced by Gelb [146, 147] in the 1970s to evaluate the accuracy of nonlinear guidance systems [148].

A significant benefit of quasi-linear approximation, evident from the last equations, is its independence from the necessity of derivatives of $\boldsymbol{f}\left(\boldsymbol{x},\, t\right)$. Therefore, it can accommodate various nonlinear behaviors—like relay actions, saturation, and thresholds—without the need for smoothing over discontinuities or abrupt changes in $\boldsymbol{f}\left(\boldsymbol{x},\, t\right)$.

3.2 Nonlinear methods

The main limitation of linear methods for uncertainty propagation is a loss of accuracy when applied to highly nonlinear dynamical systems or situations with substantial initial uncertainties. This is due to the potential deviation of the true trajectory from the assumed linear behavior, particularly in statistical terms [149]. Junkins et al. [150, 151] were among the pioneers to explore the nonlinear, non-Gaussian aspects of orbit uncertainty propagation. Building upon their findings,

Scheeres et al. observed fundamentals constraint on the propagation of orbital uncertainty [152]. Furthermore, they introduced an innovative approach to derive constraints on covariance matrices based on topological aspects [153], enriching the theoretical framework of orbit uncertainty propagation. Following these studies, the past twenty years have seen a surge in the development of analytical and semi-analytical tools for nonlinear uncertainty propagation.

3.2.1 State transition tensors

An extension to linearization techniques, Park and Scheeres introduced a semi-analytic approach for orbit uncertainty propagation, referred to as State Transition Tensors (STTs) [154]. This method propagates the state uncertainty by computing higher-order Taylor series terms that capture the local nonlinearities of the EoM, coupled with an analytical procedure for mapping initial uncertainties.

The nonlinear relative motion in the vicinity of a nominal trajectory can be estimated by implementing a Taylor series expansion of the solution function in terms of the initial state variables. Given a multi-variate random variable $\boldsymbol{x}(t_0) = \sum_{i=1}^{n} \boldsymbol{x}_i(t_0)$, at epoch t_0, with $\boldsymbol{x}_i(t_0) \in \mathbb{R}^n$ as the component along the i-th dimension, and $\boldsymbol{\mu}(t_0)$ and $\boldsymbol{P}(t_0)$ as the related mean and covariance matrix, the generic state deviation $\delta\boldsymbol{x}(t)$, at epoch t, as a function of the initial displacement $\delta\boldsymbol{x}(t_0) = \delta\boldsymbol{x}^0$ from the nominal trajectory is given by

$$\delta\boldsymbol{x}(t) = \boldsymbol{\phi}\left(t;\, \boldsymbol{x}(t_0) + \delta\boldsymbol{x}^0,\, t_0\right) - \boldsymbol{\phi}\left(t;\, \boldsymbol{x}(t_0),\, t_0\right) \tag{3.19}$$

where the operator $\boldsymbol{\phi}\left(t;\, \boldsymbol{x}(t_0),\, t_0\right)$ denotes the dynamical evolution of the state $\boldsymbol{x}(t_0)$, from t_0 to t, through $\boldsymbol{f}\left(\boldsymbol{x}(t),\, t\right)$. Applying a Taylor expansion with respect to the nominal trajectory, the state deviation of the i-th component is obtained as

$$\delta x_i(t) = \sum_{p=1}^{M} \left(\sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \dots \sum_{k_p=1}^{n} \frac{1}{p!} \Phi_{i,\, k_1 k_2 \dots k_p} \delta x_{k_1}^0 \delta x_{k_2}^0 \dots \delta x_{k_p}^0 \right) \tag{3.20}$$

where M is the expansion order, the subscript $k_j = 1, \dots, n$ denotes the k_j-th component of the state, and $\Phi_{i,\, k_1 k_2 \dots k_p} \in \mathbb{R}$ is the STT defined as

$$\Phi_{i,\, k_1 k_2 \dots k_p} = \frac{\partial^p x_i^0}{\partial x_{k_1}^0\, \partial x_{k_2}^0 \dots \partial x_{k_p}^0} \tag{3.21}$$

which can be calculated by numerical integration along the nominal trajectory[154, 149]. In other words, Eq. (3.20) provides the effect on the i-th component of the state given by the initial displacements δx_i^0, $i = 1, \dots, n$.

Substituting Eq. (3.20) into the definition of mean state and covariance, the nonlinear propagation of the components of these moments is be obtained as [154]

$$\mu_i(t) = \mathbb{E}\left[\delta x_i(t)\right] = \sum_{p=1}^{M} \left(\sum_{k_1=1}^{n} \dots \sum_{k_p=1}^{n} \frac{1}{p!} \Phi_{i,\, k_1 \dots k_p} \mathbb{E}\left[\Lambda_{k,\, p}^0\right] \right) \tag{3.22}$$

$$P_{ij}(t) = \mathbb{E}\left[\delta x_i(t) \delta x_j(t)\right] - \mu_i(t)\mu_j(t)^T = \tag{3.23}$$

$$= \sum_{p=1}^{M} \sum_{q=1}^{M} \left(\sum_{k_1=1}^{n} \dots \sum_{k_p=1}^{n} \sum_{l_1=1}^{n} \dots \sum_{l_p=1}^{n} \frac{1}{p!q!} \Phi_{i,\, k_1 \dots k_p} \Phi_{i,\, l_1 \dots l_p} \mathbb{E}\left[\Lambda_{k,\, p}^0 \Lambda_{l,\, p}^0\right] \right) + \tag{3.24}$$

$$- \mu_i(t)\mu_j(t) \qquad \forall i,\, j = 1, \dots, n \tag{3.25}$$

where

$$\Lambda^0_{k,\,p} = \delta x^0_{k_1} \ldots \delta x^0_{k_p} \tag{3.26}$$

$$\Lambda^0_{l,\,p} = \delta x^0_{l_1} \ldots \delta x^0_{l_p} \tag{3.27}$$

For $M = 1$, the latter expressions are equivalent to the first-order linear covariance propagation in Eq. (3.9), without the perturbation term. Given the expressions for the components, the computation of mean and covariance matrix in a future time is just an algebraic operation.

Utilizing the concept of STT, Fujimoto et al. constructed an analytical nonlinear uncertainty propagator for both conservative [155] and non-conservative dynamical systems (such as those affected by atmospheric drag [156]). Furthermore, Park et al. introduced a simplified dynamic model to examine the interplay between the accuracy of dynamic models and the precision of uncertainty propagation [157]. Their findings revealed that long-term, or secular, changes play a pivotal role in mapping uncertainty.

The method based on STTs eliminates the need for random sampling in uncertainty propagation, and its results have shown favorable comparison with MC simulations, offering an effective semi-analytical approach for uncertainty mapping in nonlinear dynamic contexts. Despite its merits, the STT methodology is dependent on the continuity and differentiability of the governing dynamics. The complexity of deriving higher-order STTs makes the method less practical, particularly for systems requiring high-fidelity dynamics due to the computational intensity. To address these challenges, various techniques have been introduced to calculate STTs more efficiently, such as automatic differentiation in the work of Barrio et al. [158] and the modified Picard integrator by Nakhjiri and Villac [159], which simplified the computation process.

3.2.2 Polynomial chaos expansion

Methods based on PCE offer insights into higher-order moments. These methods approximate the inputs and outputs of a given system using a series expansion with standardized random variables, enabling a more comprehensive understanding of the system's behavior under uncertainty. This methodology employs the same set of random variables to characterize both input and output uncertainties. Consequently, the output model is articulated as a series expansion composed of orthogonal polynomials. This series is as

$$\boldsymbol{x}\,(t,\,\boldsymbol{\xi}) = \sum_{k=0}^{\infty} \boldsymbol{c}_k(t)\boldsymbol{\Psi}_k\,(\boldsymbol{\xi}) \tag{3.28}$$

where $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_d]$ is a random vector of dimension $d \in \mathbb{N}$, where each component is independent and identically distributed, $\boldsymbol{\Psi}_k\,(\boldsymbol{\xi})$ denotes a generic element in the set of multidimensional orthogonal polynomials of order $k \in \mathbb{N}_0$, and $\boldsymbol{c}_k(t) \in \mathbb{R}^n$ is the corresponding coefficient vector at the output time t. The orthogonal polynomials have the following property

$$\left\langle \boldsymbol{\Psi}_i\,(\boldsymbol{\xi})\,,\ \boldsymbol{\Psi}_j\,(\boldsymbol{\xi}) \right\rangle = \left\langle \boldsymbol{\Psi}_i\,(\boldsymbol{\xi})^2 \right\rangle \delta_{ij} = \int \boldsymbol{\rho}\,(\boldsymbol{\xi})\,\boldsymbol{\Psi}_i\,(\boldsymbol{\xi})\,\boldsymbol{\Psi}_j\,(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} \tag{3.29}$$

where $\langle \cdot,\, \cdot \rangle$ denotes the inner product, δ_{ij} is the Kronecker delta function, and $\boldsymbol{\rho}\,(\boldsymbol{\xi})$ is the weighting function. The polynomials that comprise these orthogonal bases are

part of the Askey scheme, which is a hierarchy of orthogonal polynomials [160]. Each subset of the orthogonal polynomials in the Askey scheme has a different weighting function in its orthogonality relationship. For istance, Hermite polynomials are the basis for Gaussian distributed $\boldsymbol{\xi}$, while Legendre polynomials are used for uniform distribution [161].

The number of terms for $\{\boldsymbol{\Psi}_i(\boldsymbol{\xi})\}$ in the PCE model in Eq. (3.28) is infinite; thus, truncation at some order p is required for computational purposes. The approximated state $\hat{\boldsymbol{x}}$ is hence modeled with PCE as

$$\hat{\boldsymbol{x}}(t, \boldsymbol{\xi}) = \sum_{k=0}^{P} \boldsymbol{c}_k(t)\boldsymbol{\Psi}_k(\boldsymbol{\xi}) \tag{3.30}$$

where P satisfies [161]

$$P + 1 = \frac{(p+d)!}{p!d!} \tag{3.31}$$

As implied by this last expression, the number of terms P varies exponentially with p and d, which leads to the notorious curse of dimensionality.

The development of PCE involves estimating the coefficients $\boldsymbol{c}_k$. This process may be accomplished in two fashions: *intrusive* and *non-intrusive* method. The intrusive method solves for $\boldsymbol{c}_k$ via the Galerkin projection applied to the stochastic dynamic system's subspace, which results in modifying the original governing equations. Substituting the state expressed as in Eq. (3.30) in the deterministic component of the EoM, the outcome is the following ODE

$$\frac{\mathrm{d}}{\mathrm{dt}}\left(\sum_{k=0}^{P} \boldsymbol{c}_k(t)\boldsymbol{\Psi}_k(\boldsymbol{\xi})\right) = \boldsymbol{f}\left(\sum_{k=0}^{P} \boldsymbol{c}_k(t)\boldsymbol{\Psi}_k(\boldsymbol{\xi}), t\right) \tag{3.32}$$

When applying the Galerkin projection on each orthogonal polynomial $\boldsymbol{\Psi}_k(\boldsymbol{\xi})$ in sequence, a series of deterministic ODEs emerge, encompassing $(P+1) \times n$ interconnected equations

$$\frac{\mathrm{d}c_{ik}}{\mathrm{dt}} = \frac{1}{\langle \boldsymbol{\Psi}_k(\boldsymbol{\xi}), \boldsymbol{\Psi}_k(\boldsymbol{\xi})\rangle}\langle \boldsymbol{f}\left(\sum_{k=0}^{P} \boldsymbol{c}_k(t)\boldsymbol{\Psi}_k(\boldsymbol{\xi}), t\right), \boldsymbol{\Psi}_k(\boldsymbol{\xi})\rangle \qquad \begin{matrix} i = 1, \ldots, n \\ k = 0, \ldots, P \end{matrix} \tag{3.33}$$

where c_{ik} is the k-th coefficient of the i-th component of $\boldsymbol{x}$. The EoM are thus transformed to the deterministic ODEs in Eq. (3.34) with dimension $(P+1) \times n$, which can be solved by any standard ODE solver. Once the distribution of the initial state is given, the initial values of the ODEs in Eq. (3.34) can be calculated using the method presented in [118].

On the other hand, the non-intrusive methods rely on black box solvers to compute c_{ik} via least-squares regression or pseudospectral integration on tensor/sparse grids. In [162], a process based on the least-squares regression is illustrated.

Once the PCE coefficients are computed by means of either intrusive or non-intrusive methods, the mean and covariance of the output uncertainties are obtained as

$$\boldsymbol{\mu}(t) = \boldsymbol{c}_0(t) \tag{3.34}$$

$$\boldsymbol{P}(t) = \boldsymbol{C}(t)\boldsymbol{C}(t)^T \tag{3.35}$$

where $\boldsymbol{C}(t) = [\boldsymbol{c}_1(t), \boldsymbol{c}_2(t), \ldots, \boldsymbol{c}_P(t)]$.

This method of approximation was first proposed by Wiener [163].

To mitigate the curse of dimensionality (see Eq. (3.31)), Wan and Karniadakis introduced the multi-element generalized PC method [164]. This approach partitions the space of stochastic inputs into subdomains when the relative variance error surpasses a predefined threshold. Subsequently, it implements a PC expansion within each of these delineated random elements, effectively managing the complexity by localizing the computational effort to areas of significant uncertainty.

PCE method offers a powerful tool for characterizing finite-variance solutions that may deviate from a Gaussian distribution. In comparison to MC simulations, computation is faster, which is exponentially related to the degree of the polynomial basis, even under non-Gaussian uncertainty. Additionally, it furnishes a clear, functional relationship between output uncertainties and stochastic inputs, facilitating sensitive analytical endeavors and crafting measurement protocols. However, the method's complexity scales exponentially with the increase in the dimensions of input uncertainty.

3.2.3 Gaussian mixture model

A major challenge in UP is that the state distribution often tend to evolve into distributions that significantly depart from the Gaussian one. In OD analysis, Gaussian assumption is commonly adopted, especially when the state dispersion remains relatively confined. However, the contour set that corresponds to a given confidence level (e.g., 3σ) of dispersion does not generally keep, over time, the ellipsoidal shape typical of the Gaussian distribution; instead, it tends to distort and deform, resulting in a geometrical evidence that the state distribution can no longer be assumed as Gaussian distribution.

To address this limitation, the Gaussian Mixture Model (GMM) was introduced as a nonlinear approach that accurately describes the propagated PDF while retaining some of the benefits of the Gaussian distribution model. The core concept of this approach is to represent any arbitrary PDF as a finite sum of weighted Gaussian densities. The weights of the different Gaussian components are computed through numerical optimization techniques. The approximated PDF $\hat{p}$ of $\boldsymbol{x}$ is formulated as

$$\hat{p}\left(\boldsymbol{x}(t),\, t\right) = \sum_{i=1}^{N} \omega_i p_G\left(\boldsymbol{x}(t);\, \boldsymbol{\mu}_i(t),\, \boldsymbol{P}_i(t)\right) \tag{3.36}$$

where N is the number of Gaussian kernels, $\boldsymbol{\mu}_i$ and $\boldsymbol{P}_i$ are the mean and covariance matrix of the i-th Gaussian density function $p_G\left(\boldsymbol{x};\, \boldsymbol{\mu}_i,\, \boldsymbol{P}_i\right)$ respectively, and ω_i is the weight of the i-th Gaussian kernel. Constraint on normalization and sign of the mixture PDF lead to the following constraints on the weights

$$\sum_{i=1}^{N} \omega_i = 1 \qquad \omega_i \geqslant 0 \qquad \forall i = 1, \ldots, N \tag{3.37}$$

The larger is N. the better is the approximation of the original PDF.

The problem is thus reduced to propagate the sub-components of the state distribution, which can be accomplished with other UP tools, as the ones described in this chapter. Once the Gaussian kernels and the related weights are determined,

the propagated mean and covariance are given as

$$\boldsymbol{\mu}(t) = \sum_{i=1}^{N} \omega_i \boldsymbol{\mu}_i(t) \tag{3.38}$$

$$\boldsymbol{P}(t) = \sum_{i=1}^{N} \omega_i \left(\boldsymbol{P}_i(t) + \boldsymbol{\mu}_i(t)\boldsymbol{\mu}_i(t)^T \right) - \boldsymbol{\mu}(t)\boldsymbol{\mu}(t)^T \tag{3.39}$$

where $\boldsymbol{\mu}_i(t)$ and $\boldsymbol{P}_i(t)$ are the i-th propagated mean and covariance respectively. For an accurate propagation, a precise subdivision of the PDF into kernels and computation of the weights is hence the most important aspect of this method, and different solutions are proposed in literature to accomplish this task.

A straightforward method involves propagating an initial Gaussian mixture into a final Gaussian mixture while maintaining constant weights. Various techniques are employed to calculate these initial weights, typically by dividing an initial Gaussian distribution into several weighted Gaussian components. Horwood et al. [165, 166] introduced a method that splits the initial Gaussian distribution by solving a constrained optimization problem. In contrast, DeMars et al. [167, 168] explored an entropy-based splitting technique where the GMM components are constrained to be homoscedastic, meaning that they all share the same variance. Psiaki et al. [169] introduced an innovative algorithm designed to approximate one Gaussian mixture with another, expanding on the traditional particle filter concept. This algorithm aims to closely replicate the original mixture using a constrained number of Gaussian distributions. In parallel, the adaptability of the GMM has been enhanced by Terejanu et al. [170, 171] and Vishwajeet et al. [172], who innovated methods to adjust the weights of GMM components dynamically during the propagation of the PDF. These adaptations involve locating the optimal weights for the GMM components by minimizing, across the entire area of interest, the error from the Fokker-Plank Equation (FPE), which is a partial differential equation that satisfies the propagation of a PDF (hence, the solution of the FPE provides a complete statistical description of the propagation) [173].

The GMM approach effectively breaks down a large uncertainty propagation problem into smaller, manageable segments. This strategy not only enables accurate representation of non-Gaussian distributions, but also mitigates the impacts of dynamic nonlinearity. A significant benefit of using GMM is its ability to approximate the PDF of a non-Gaussian distribution by merely propagating the first two moments of the Gaussian mixtures. However, this approach is not without its limitations. First, achieving a desired level of accuracy requires the propagation of a large numbers of Gaussian components, leading to the curse of dimensionality problem. Also, the method requires solving optimization problems to determine and adjust the weights of the Gaussian mixtures throughout the propagation process, which significantly increases the computational time and resource demand with respect to other approaches.

3.2.4 Unscented transform

The Unscented Transform (UT) is a technique to estimate the mean value and the covariance of a probability distribution related to a certain variable that undergoes a nonlinear transformation. This is achieved by propagating a representative set of points of the distribution, named as Sigma Points (SPs), and estimate the new mean value and covariance by means of these propagated points.

Given a multivariate random state variable $\boldsymbol{x}(t_0) \in \mathbb{R}^n$, subject to a deterministic dynamics $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), t)$, with mean $\boldsymbol{\mu}(t_0)$ and covariance matrix $\boldsymbol{P}(t_0)$ at epoch t_0, UT can be used to estimate of the mean and covariance matrix after a time propagation of Δt. The steps of UT are:

1. Evaluate the set of $2n+1$ SP $\boldsymbol{\mathcal{X}}_i$ and the related weights c_i, $i = 0, \ldots, 2n$, where $\boldsymbol{\mathcal{X}}_0$ is termed central (or pivotal) SP as

$$\boldsymbol{\mathcal{X}}_0 = \boldsymbol{\mu}(t_0) \tag{3.40}$$

$$\boldsymbol{\mathcal{X}}_j = \boldsymbol{\mu}(t_0) + \left(\sqrt{(n+\kappa)\boldsymbol{P}(t_0)}\right)_j \tag{3.41}$$

$$\boldsymbol{\mathcal{X}}_{n+j} = \boldsymbol{\mu}(t_0) - \left(\sqrt{(n+\kappa)\boldsymbol{P}(t_0)}\right)_j \qquad j = 1, \ldots, n \tag{3.42}$$

$$c_0 = \kappa/n + \kappa \tag{3.43}$$

$$c_i = \frac{1}{2(n+\kappa)} \qquad i = 1, \ldots, 2n \tag{3.44}$$

   where $\kappa \in (0, \infty) \subset \mathbb{R}$ is a scaling parameter and the operator $(\cdot)_j$ is used to represents the j-th column vector of the matrix in argument.

2. Propagate the SPs $\boldsymbol{\mathcal{X}}_i$ to obtain the new set $\boldsymbol{\mathcal{Y}}_i$ as

$$\boldsymbol{\mathcal{Y}}_i = \boldsymbol{\phi}(t; \boldsymbol{\mathcal{X}}_i, t_0) \qquad i = 0, \ldots, 2n \tag{3.45}$$

   where the operator $\boldsymbol{\phi}(t; \boldsymbol{x}, t_0)$ denotes the solution flow through $\boldsymbol{f}(\boldsymbol{x}(t), t)$;

3. The propagated mean value $\boldsymbol{\mu}(t)$ and covariance matrix $\boldsymbol{P}(t)$ are evaluated as

$$\boldsymbol{\mu}(t) = \sum_{i=0}^{2n} c_i \boldsymbol{\mathcal{Y}}_i \tag{3.46}$$

$$\boldsymbol{P}(t) = \sum_{i=0}^{2n} c_i \left(\boldsymbol{\mathcal{Y}}_i - \boldsymbol{\mu}(t)\right)\left(\boldsymbol{\mathcal{Y}}_i - \boldsymbol{\mu}(t)\right)^T \tag{3.47}$$

In Fig. 3.1, a schematic visualization of UT is presented, having the initial and final covariance matrices represented with 2D ellipses. The evolution through the nonlinear dynamics of the covariance matrices and the mean state is depicted with dashed lines. Note that the initial mean $\boldsymbol{\mu}(t_0)$ and the pivotal SP $\boldsymbol{\mathcal{X}}_0$ are coincident, according to Eq. (3.40).

Julier et al. [174] introduced the UT to overcome the inherent limitations of linearization techniques. This method suggests that it might be simpler to directly approximate a system's probability distribution than to apply approximations to its nonlinear functions [175, 176, 177]. UT offers a second-order accurate approximation for the first two statistical moments of a mapped distribution. While this feature is advantageous, the method is limited to the propagation of these two moments alone. In scenarios where a more accurate description of the uncertainty is demanded and higher order statistical moments play a critical role, the effectiveness of UT becomes questionable. Extensions of the UT to higher order moments have been proposed by Adurthi et al. [178] under the definition of Conjugate UT (CUT), followed by the order up to which the moment constraints are satisfied (e.g., CUT8).

Given the optimal balance between accuracy and computational cost provided by this technique, the UT is the preferred UP method chosen for the applications discussed in the following chapters.

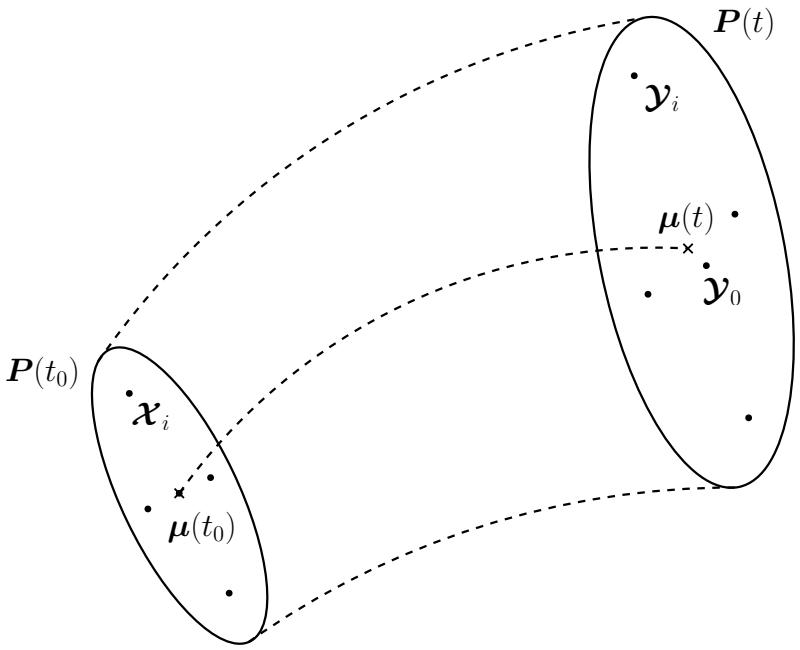


Figure 3.1. Schematic representation of UT.

3.2.5 Other nonlinear methods

Less common nonlinear methods are also proposed in the literature. In order to avoid the derivation of increasingly complex partial derivatives or numerical approximations for the partial derivatives of the STT method, a technique based on Differential Algebra (DA) was developed by Armellin and Di Lizia [179, 180]. DA was first proposed by Berz [181] to compute the derivatives of functions within resorting to numerical methods. The main idea behind the DA method is to compute an arbitrary order expansion for the solution flow of a general ODE with respect to the initial condition. The DA-based approach presents a significant advantage as it eliminates the necessity to integrate variational equations for deriving high-order expansions of the dynamical flow. This DA framework automatically generates these expansions to any desired order, allowing for the nonlinear propagation of uncertainty through these high-order terms. However, like the STT method, the DA approach requires the governing dynamics to be continuous and differentiable, which limits its applicability in scenarios involving discontinuities. For instance, solar radiation pressure perturbation acts as a step function during a satellite's transitions into and out of the Earth's shadow, posing challenges for the DA method at these critical points. Therefore, while the DA method offers robust capabilities for smooth dynamic systems, its utility is constrained in many practical scenarios.

Alternatively, another way to propagate the state uncertainty involves directly solving the Fokker-Planck equation, which governs the time evolution of a PDF. However, solving the FPE for high-dimensional dynamical systems poses significant challenges and may not be a valid option in many cases [182]. As an example, traditional discretization methods, such as the finite element and finite difference methods, require an exponential increase in computational resources as the number of nodes for each dimension increases. If m nodes are used per dimension, the total number of unknowns for an n-dimensional problem scales as n^m. Consequently, for an orbital problem with $n = 6$, the exponential growth in the number of variables

renders the computational complexity intractable, even for advanced supercomputing platforms [183, 184]. In recent years, significant progress has been made in developing efficient numerical solvers for high-dimensional FPE. Kumar et al. [185] introduced a homotopic iterative method for refining solutions of the stationary FPE, offering a systematic approach to define the solution domain for nonlinear systems. Building on this, the same authors developed a semi-analytic partition of unity finite element method aimed at solving the transient FPE for complex, high-dimensional dynamical systems [183]. Further advancements were made by using a numerical strategy for the stationary FPE that integrates tensor decomposition with Chebyshev spectral differentiation [186]. This approach significantly reduces the computational load by decreasing the number of degrees of freedom required to achieve high accuracy in systems with increasing dimensionality.

3.3 Other methods

Beside the linear and nonlinear uncertainty propagation methods previously discussed, there exist various other approaches that do not neatly fit into a single category of propagation technique.

Several hybrid methods have been developed to leverage the strengths of various uncertainty propagation techniques. Fujimoto and Scheeres [187] introduced a method that combines the GMM and STTs for enhanced UP and conjunction assessment. Vittaldev et al. [188] developed an approach that merges GMM with PCE, aiming to mitigate the dimensionality challenges inherent in PC by using GMM to reduce the number of polynomial terms needed for precise accuracy. Aristoff et al. utilized Implicit Runge-Kutta (IRK) based orbit propagators combined with the UT [189] and classical particle filters [190] for rapid UP. This integration is particularly effective because some uncertainty quantification methods, such as particle filtering, require propagating a set of particles or states through nonlinear dynamics. The IRK methods capitalize on the proximity of initial conditions to efficiently propagate uncertainties. Moreover, Horwood et al. introduced a novel approach for orbital uncertainty propagation using the Gauss von Mises distribution [191]. They also conducted a comparative study examining the performance differences between linear method, UT, and their Gauss von Mises method [192], further enriching the field of uncertainty propagation methodologies.

A further class of methods consists in performing a coordinate transformation, with the aim of using efficiently a specific UP method that would not be applicable or accurate enough in the original set of coordinates. The influence of coordinate systems on uncertainty propagation statistics was initially explored by Junkins et al. [150] and later by Junkins and Singla [151]. They introduced coordinate transformations as an innovative method for orbit UP, discovering that orbital elements could enhance predictive accuracy. Nevertheless, their approach remained rooted in linear assumptions, omitting the nonlinear dynamics from their analysis. In contrast, Sabol et al. [193] and Woodburn and Coppola [194] demonstrated that equinoctial elements maintain normality more effectively than Cartesian elements. Building on this, Aristoff et al. [195] devised a method to convert a Gaussian or Gaussian mixture from orbital element space to Cartesian space while preserving accuracy. Furthermore, Majji et al. [196] investigated the transformations of probability density functions across different coordinate systems, highlighting the utility of the change of variables formula for evaluating estimation errors in varying coordinate systems, providing valuable insights for analytical comparisons. Although equinoctial elements effectively represent the probability distribution of perturbed trajectories, the

combined formulation of position and velocity complicates the accurate depiction of position uncertainty. To address this, Hill et al. [197, 198] showed that state uncertainty, when expressed in elliptical curvilinear coordinates, maintains Gaussian characteristics over longer propagation periods and larger in-track distances compared to Cartesian coordinates. Further, Vallado and Alfano [199] crafted a precise transformation from Cartesian to Hill's curvilinear frame aligned with the actual satellite orbit. This transformation proved crucial for correctly positioning a covariance ellipsoid in the Earth-centered inertial frame, particularly when the covariance is large, requiring a bend to accurately represent the ellipsoid shape. Enhancing this approach, Tanygin [200] and Coppola and Tanygin [201] developed explicit equations for these curvilinear coordinates.

In summary, coordinate transformations may effectively mitigate the most dominant nonlinear terms in the equations of motion, thereby maintaining the validity of Gaussian and linear approximations. Moreover, using curvilinear or orbital element spaces can more effectively reduce the impacts of nonlinearity compared to Cartesian coordinates.

Chapter 4

Robust open-loop optimization of impulsive trajectories

This chapter presents an initial step to transition from a deterministic OCP to an ROCP, whose main features have been described in Sec. 2.3. A straightforward methodology to obtain robust OL control law is developed. This approach employs a fixed sequence of control maneuvers as the sole means to enhance the robustness of the trajectory, regardless of the actual flown trajectory. Although an OL control cannot be as effective as a feedback control, which computes corrective maneuvers most suitable for real-time deviations, robust trajectory design with fixed pre-computed maneuvers may still suffice in simpler scenarios. Additionally, it offers valuable insights into how state dispersion during missions is influenced by the dynamical environment.

4.1 General formulation

Aim of the ROCP is to optimize a sequence of impulsive maneuvers to transfer in a limited amount of time a spacecraft from a given departure state, described in terms of position and velocity, to a prescribed target state, described by position and velocity. The time at which each maneuver occurs is free to be optimized. The aim of the optimization is to minimize a measure of the dispersion of the spacecraft state at the arrival, while ensuring that the magnitude of the impulsive maneuvers is bounded and the cumulative change of velocity Δv_{tot} is below a given threshold.

The spacecraft is modeled as a point-mass object and its state $\boldsymbol{x}(t) \in \mathbb{R}^n$ is defined in terms of his Cartesian position as $\boldsymbol{x}(t) = \left[\boldsymbol{r}(t)^T \, \boldsymbol{v}(t)^T\right]^T$, being $\boldsymbol{r}(t)$ and $\boldsymbol{v}(t)$ the position and velocity of the spacecraft, respectively. A three-dimensional space is considered, thus $n = 6$. The trajectory is assumed decomposed into N ballistic arcs, whose duration is subject to optimization, joined by impulsive velocity changes at each node, which represent the spacecraft maneuvers. Additional impulsive maneuvers are assumed at the departure and arrival time. Starting from a fixed departure epoch $\widetilde{t}_i$, where the tilde denotes a constant (user-defined) quantity, the time-grid for the $N + 1$ maneuvers is thus defined as

$$\widetilde{t}_i = t_0 \leqslant t_1 \leqslant \cdots \leqslant t_N = t_f \tag{4.1}$$

having the time intervals for the ballistic arcs as $\Delta t_k = t_k - t_{k-1}$, $\forall k = 1, \ldots, N$. The values of the variables immediately before and after each k-th node are referred to with the subscripts $-$ and $+$, respectively. Position continuity must be ensured at

each node, thus $\boldsymbol{r}_{k-} = \boldsymbol{r}_{k+}$, while velocity changes occur instantaneously according to the impulsive model, as $\boldsymbol{v}_{k+} = \boldsymbol{v}_{k-} + \boldsymbol{u}_k$, where the subscript k is used for quantities at epoch t_k. In this model, the available controls $\boldsymbol{u}_k \in \mathbb{R}^3$ are therefore given by the velocity impulse, that is $\boldsymbol{u}_k = \Delta\boldsymbol{v}_k$, $\forall k = 0, \dots, N$. The update law for the state is thus simply given by

$$\boldsymbol{x}_{k+} = \boldsymbol{x}_{k-} + \boldsymbol{B}\boldsymbol{u}_k \qquad \forall k = 0, \dots, N \tag{4.2}$$

with $\boldsymbol{B} = [\boldsymbol{0}_{3\times 3}\, \boldsymbol{I}_3]^T$. The control magnitude $\|\boldsymbol{u}_k\|$ is constrained below an user-defined threshold $\widetilde{u}_{\max}$.

4.1.1 Boundary constraints

The initial and final constraints for position and velocity are defined as in canonical OCPs, that is

$$\boldsymbol{x}_{0-} = \widetilde{\boldsymbol{x}}_i \tag{4.3}$$
$$\boldsymbol{x}_{N+} = \widetilde{\boldsymbol{x}}_f \tag{4.4}$$

where $\widetilde{\boldsymbol{x}}_i$ and $\widetilde{\boldsymbol{x}}_f$ are respectively the fixed departure state before the first impulsive maneuver $\boldsymbol{u}_0$ and the target state after the last maneuver $\boldsymbol{u}_N$. In other words, starting from the assigned state $\widetilde{\boldsymbol{x}}_i$, the spacecraft is constrained to reach a target state $\widetilde{\boldsymbol{x}}_f$ through $N+1$ impulsive maneuvers.

As additional parameter to be defined, a variable that quantifies the state dispersion along the trajectory is necessary, in order to provide a robustness index in the algorithm. In this regard, the initial covariance matrix $\boldsymbol{P}_0 = \mathrm{Cov}\,[\boldsymbol{x}_{0-}]$ is assigned as

$$\boldsymbol{P}_0 = \widetilde{\boldsymbol{P}}_0 \tag{4.5}$$

The OL outlined methodology revolves around the computation of a sequence of impulsive maneuvers that minimize the final dispersion of the state, in compliance with constraints on single and cumulative control magnitude.

4.1.2 Perturbed state propagation

Since information on the state dispersion upon arrival is incorporated into the cost function to be minimized, it is necessary for the algorithm to include the UP of the initial covariance.

Starting from the initial dispersion $\boldsymbol{P}_0$, the final covariance matrix $\boldsymbol{P}_N$ is computed by means of the UT. A cost function that is involves the elements of $\boldsymbol{P}_N$ is then defined to optimize the sequence of $\boldsymbol{u}_k$, $\forall k = 0, \dots, N$.

Starting from the initial state $\boldsymbol{x}_{0-}$, the state after the first maneuver is

$$\boldsymbol{x}_{0+} = \boldsymbol{x}_{0-} + \boldsymbol{B}\boldsymbol{u}_0 \tag{4.6}$$

This updated state is then used, together with the covariance matrix $\boldsymbol{P}_0$, as mean state and propagated for the ToF Δt_1. The pivotal SP after propagating from time t_0 to time t_1 is $\boldsymbol{\mathcal{Y}}_0 = \boldsymbol{\phi}\,(\Delta t_1;\, \boldsymbol{\mathcal{X}}_0,\, 0)$, as in Eq. (3.45), with $\boldsymbol{\mathcal{X}}_0 = \boldsymbol{x}_{0+}$.

This propagated SP is then used as the final state $\boldsymbol{x}_{1-}$ of the ballistic arc before the second Δv. Then, the propagated covariance matrix $\boldsymbol{P}_1$ is used together with the updated state $\boldsymbol{x}_{1+} = \boldsymbol{x}_{1-} + \boldsymbol{B}\boldsymbol{u}_1$ as now pair of statistical moments to be propagated

through UT. This process is repeated up to the final time, while the state after the final Δv is the last propagated pivotal SP to which $\boldsymbol{u}_N$ is added.

An important feature of the robust optimization to be included during the propagation is however a stochastic perturbation as in the SDE of Eq. (2.54). An issue is posed by the characteristics of UT propagation when considering SDE for the ballistic arcs. The usage of an SDE, as the Itô form in Eq. (2.54), to propagate the SPs during the optimization process would not be possible as the final mean and covariance matrix would differ from iteration to iteration, resulting in convergence issues related to the variability of the computed results. For this reason, the stochastic term of the SDE is neglected when propagating the SPs, that are thus propagated using deterministic EoM

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{f}\left(\boldsymbol{x}(t),\, t\right) \qquad \forall t \in (t_{k-1},\, t_k) \qquad \forall k = 1, \ldots, N \tag{4.7}$$

The random contribution is instead modeled as an instantaneous, additive perturbation $\boldsymbol{Q}$ that takes place at the end of each arc, representing the overall effects of the stochastic perturbations over a finite-time arc. Let $\boldsymbol{P}'_{k-}$ be the covariance matrix estimated by UT, the covariance matrix after the inclusion of the (stocastic) diffusion term is $\boldsymbol{P}_{k-} = \boldsymbol{P}'_{k-} + \boldsymbol{Q}$. Rather than an empirically chosen $\boldsymbol{Q}$, a more sophisticated method to estimate this matrix will be introduced in the impulsive CL chapter (Chapter 5).

At each k-th node, the propagation is thus summarized as

$$\{\boldsymbol{\mathcal{Y}}_0,\, \boldsymbol{P}_k\} = \mathcal{U}\left(\boldsymbol{x}_{(k-1)+},\, \boldsymbol{P}_{k-1},\, \Delta t_k;\, \boldsymbol{Q}\right) \tag{4.8}$$

$$\boldsymbol{x}_{k-} = \boldsymbol{\mathcal{Y}}_0 \qquad k = 1, \ldots, N \tag{4.9}$$

where $\mathcal{U}(\cdot)$ stands for a function based on the UT tool returning the covariance at the end of the k-th ballistic arc and the propagated pivotal SP $\boldsymbol{\mathcal{Y}}_0 = \boldsymbol{\phi}\left(t_k;\, \boldsymbol{\mathcal{X}}_0,\, t_{k-1}\right)$, with $\boldsymbol{\mathcal{X}}_0 = \boldsymbol{x}_{(k-1)+}$. By tracking the state dispersion along the trajectory at each iteration of the optimization algorithm, the cost function can be accurately computed using the elements of the final covariance matrix. This approach ensures that the optimization process considers the impact of state dispersion throughout the mission, as detailed in the following section.

4.1.3 Cost function and control constraints

Once the covariance matrix $\boldsymbol{P}_N$ is evaluated, the cost function can be computed. Different options are available; to quantify the "magnitude" of state dispersion, a straightforward approach is to use the trace of $\boldsymbol{P}_N$; indeed, the trace, which is equal to the sum of the eigenvalues, provides a direct and comprehensive estimate of the overall state dispersion. The cost function J thus simply reads

$$J = \mathrm{tr}\left(\boldsymbol{P}_N\right) \tag{4.10}$$

As an alternative, the maximum eigenvalue of $\boldsymbol{P}_N$, or the maximum element along the main diagonal can be used in place of the trace.

In order to comply with engine limitations and limit the overall fuel expenditure of the mission, constraints on the magnitude of each impulsive control $\boldsymbol{u}_k$ and the cumulative $\Delta v_{\mathrm{tot}} = \sum_{k=0}^{N} \|\boldsymbol{u}_k\|$ are enforced. An additional limitation on the total ToF, $\Delta t_{\mathrm{tot}} = \sum_{k=1}^{N} \Delta t_k$, is considered. These constraints are enforced as

$$\|\boldsymbol{u}_k\| \leqslant \widetilde{u}_{\max} \tag{4.11}$$

$$\Delta v_{\mathrm{tot}} \leqslant \Delta\widetilde{v}_{\max} \tag{4.12}$$

$$\Delta t_{\mathrm{tot}} \leqslant \Delta\widetilde{t}_{\max} \tag{4.13}$$

where $\widetilde{u}_{\max}$ and $\Delta\widetilde{t}_{\max}$ are user-defined parameters that comes from the engine technological limitation and mission planning, while $\Delta\widetilde{v}_{\max}$ comes from the propellant mass budget of the mission.

A possible criterion to select $\Delta\widetilde{v}_{\max}$ is to consider a pre-computed optimal Δv and then add a supplemental margin (e.g., 10%) to it. The optimal Δv is obtained with the deterministic formulation of the outlined OL ROCP where minimization of Δv_{tot} is sought. Having only the unperturbed EoM, this deterministic OCP is simply defined as

$$\Delta v^* = \min_{\boldsymbol{u}_k} \text{Eq. } \Delta v_{\text{tot}} \tag{4.14a}$$

$$\text{s.t.} \qquad \text{Eqs. } (4.2) - (4.4), (4.7), (4.11), (4.13) \tag{4.14b}$$

Once the OCP in Eqs. (4.14) is solved, the resulting Δv^* is increased by multiplying it by a parameter $\eta > 1$, thus Eq. (4.12) reads as

$$\Delta v_{\text{tot}} \leqslant \eta \Delta v^* = \Delta\widetilde{v}_{\max} \tag{4.15}$$

In this way, it possible to really map the increase in Δv required to obtain a reduction in the final state dispersion, embodied with the cost function J. Note that this formulation is based on the assumption that by increasing Δv_{tot} it is possible to reduce the trace of $\boldsymbol{P}_N$ (or any other chosen quantity) of the optimal solution derived from Eqs. (4.14).The covariance matrix $\boldsymbol{P}_N$ is indeed computable also in the deterministic OCP via Eq. (4.8), considering however $\boldsymbol{Q} = \boldsymbol{0}_{6\times 6}$.

The OL ROCP is hence summirized as

$$\min_{\boldsymbol{\Theta}} \text{Eq. } (4.10) \tag{4.16a}$$

$$\text{s.t.} \quad \text{Eqs. } (4.2) - (4.4), (4.8), (4.9), (4.11), (4.13), (4.15) \tag{4.16b}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{N_\Theta}$ is the optimization variables vector of

$$N_\Theta = \frac{n(N+1)}{2} + N \tag{4.17}$$

constituted by the control vector $\boldsymbol{u}_k$, $k = 0, \ldots, N$, and the N ToF Δt_k of each ballistic arc, with $k = 1, \ldots, N$.

Given the OL RCOP in Eq. (4.16), a parametric analysis ranging η from values nearly above 1 is useful to study how an OL control changes the trajectory from the optimal deterministic case to a trajectory that progressively increases in robustness.

4.2 Case study: impulsive rendezvous mission

To demonstrate the effectiveness of the proposed approach, a familiar benchmark scenario is considered, that is, a Keplerian multi-impulsive planar rendezvous problem. A chaser spacecraft departing from a body is tasked to reach a (passive) target spacecraft that flies on a larger circular orbit to a second body orbiting in a larger circular orbit. The two spacecraft are phase 180° apart at the initial epoch.

4.2.1 Problem data

As mentioned, a two-dimensional transfer is considered, and the spacecraft state vector is given by position and inertial velocity in an RF centered on a generic main

body. The dynamical model used for this analysis is a Keplerian motion, whose EoM are simply

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \begin{bmatrix} \boldsymbol{v}(t) \\ -\mu \boldsymbol{r}(t)/r^3 \end{bmatrix} \qquad \forall t \in (t_{k-1},\, t_k) \qquad \forall k = 1, \ldots, N \tag{4.18}$$

The state covariance at departure is defined as

$$\widetilde{\boldsymbol{P}}_0 = \mathrm{diag}\left[(\sigma_0^r)^2\, \boldsymbol{I}_2,\, 0,\, (\sigma_0^v)^2\, \boldsymbol{I}_2^2,\, 0 \right] \tag{4.19}$$

where σ_0^r and σ_0^v are the user-defined parameters for initial position, velocity and mass standard deviation respectively. The initial state, naturally coincident with the state of the first body, is defined by means of the initial angle θ_1 along the departure orbit, starting in a conventional way from the x axis in a counterclockwise direction. Being r_1 the radius of the departure orbit, The initial state $\widetilde{\boldsymbol{x}}_i$ is defined as

$$\widetilde{\boldsymbol{x}}_i = \begin{bmatrix} r_1 \cos\theta_1 \\ r_1 \sin\theta_1 \\ 0 \\ -\sqrt{\mu/r_1} \sin\theta_1 \\ \sqrt{\mu/r_1} \cos\theta_1 \\ 0 \end{bmatrix} \tag{4.20}$$

where $\sqrt{\mu/r_1}$ is the circular velocity of the departure orbit. The arrival state $\widetilde{\boldsymbol{x}}_i$ is not a fixed vector but is instead dependant on the ToT of the trajectory Δt_{tot}. Having θ_2 as the initial angle of the second body along its orbit, with radius r_2, the final angle $\theta_{2,\,f}$ is defined as

$$\theta_{2,\,f} = \theta_2 + 2\pi \frac{\Delta t_{\mathrm{tot}}}{T_2} \tag{4.21}$$

where $T_2 = 2\pi\sqrt{r_2^3/\mu}$ is the period of the second body. The target state at arrival $\widetilde{\boldsymbol{x}}_f$ is thus defined as

$$\widetilde{\boldsymbol{x}}_f = \begin{bmatrix} r_2 \cos\theta_{2,\,f} \\ r_2 \sin\theta_{2,\,f} \\ 0 \\ -\sqrt{\mu/r_2} \sin\theta_{2,\,f} \\ \sqrt{\mu/r_2} \cos\theta_{2,\,f} \\ 0 \end{bmatrix} \tag{4.22}$$

The perturbation matrix $\boldsymbol{Q}$ is set to zero in this preliminary study. A parametric analysis for several values of $\boldsymbol{Q}$ is then presented to assess its impact. The number of arcs N, the radius of the departure and arrival orbit, and other essential parameters are detailed in Table 4.1. The problem data are given in non-dimensional units for the sake of simplicity. In summary, the objective function is expressed as

$$J = \sum_{i=1}^{6} \sigma_i^2 \tag{4.23}$$

where σ_i^2 is the i-the element of the diagonal of $\boldsymbol{P}_N$, having the terminal constraints as in Eqs. (4.3)-(4.5) and the path constraints formulated in Eqs. (4.11), (4.13), and (4.15).

The open-source large-scale sparse nonlinear optimizer WORHP [202] was used for solving both the deterministic and the robust optimization problem.

Table 4.1. Problem data.

Variable	Value	Unit
$\Delta\widetilde{t}_{\max}$	7	-
$\widetilde{u}_{\max}$	0.11	-
N	3	-
μ	1	-
r_1	1	-
r_2	1.2	-
θ_1	0	rad
θ_2	π	rad
η	1.05	-
σ_0^r	1.000×10^{-5}	-
σ_0^v	1.000×10^{-5}	-
$\boldsymbol{Q}$	$\mathbf{0}_{6\times6}$	-

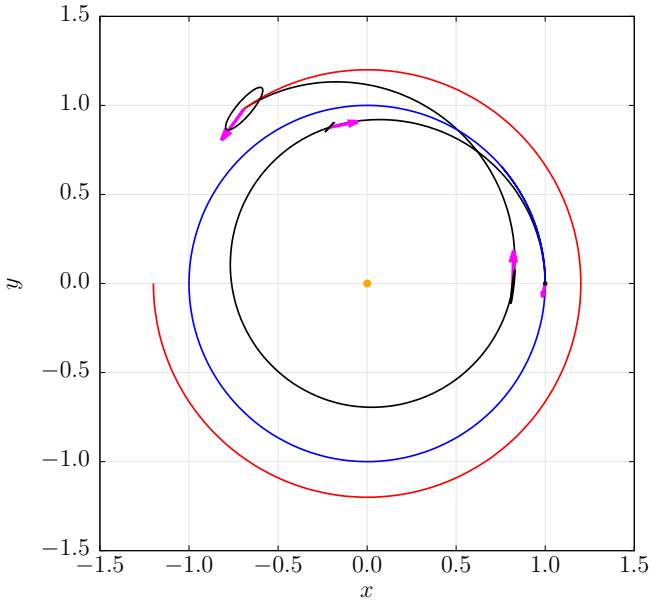
4.2.2 Numerical results

Figure 4.1 visually depicts the trajectory and covariance ellipsoids on position for both the deterministic solution (Fig. 4.1a) and the roboust OL solution (Fig. 4.1b) in the x-y plane. The blue and red lines depicts the trajectories flown by the departure and arrival body respectively during the transfer trajectory. The ROCP solution manages to shrink the state dispersion at the arrival state by properly adjusting the maneuvers and the duration of the ballistic arcs, in the same total ToF of the deterministic solution, which is both cases coincident with the maximum value of 7. The numerical results for both solutions are summarized in Table 4.2. In both the deterministic and robust solution, the maximum allowed ToF is reached as generally beneficial in terms of fuel consumption. The ROCP solution with an extra 5% of fuel consumption ($\eta = 1.05$) is able to roughly halve the trace of the final covariance matrix $\boldsymbol{P}_N$, demonstrating the capability of an OL robust design to mitigate final state dispersion. Although this is a promising enhancement, final state dispersion might not be sufficiently reduced for actual missions, suggesting the need for a more effective robust control strategy.

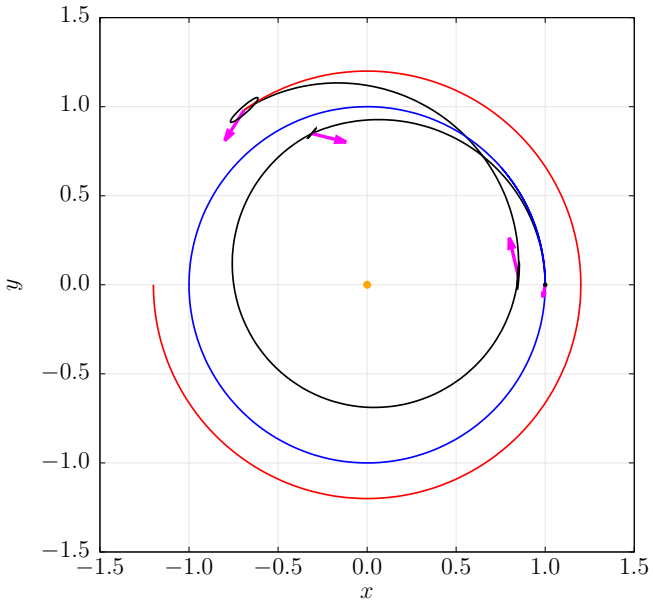
To validate the accuracy of the estimated probability distribution at each step for both the CL and OL cases, an MC analysis comprising 10 000 independent runs was conducted. Starting with a random initial state $\boldsymbol{x}_0 \sim \mathcal{N}(\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{P}}_0)$, each trajectory was propagated towards the final state using the optimized impulsive maneuvers $\boldsymbol{u}_k$ and the ToF Δt_k. Numerical results are reported in Table 4.2, facilitating a comparison between the MC and nominal solutions in both the deterministic and robust cases. The MC analysis confirms an efficient reduction of the cost function, although the MC values slightly differ from those estimated with the UT. Similar discrepancies are also observed in the final standard deviation for in-plane position (σ_N^x and σ_N^y) and velocity ($\sigma_N^{v_x}$ and $\sigma_N^{v_y}$). This deviation is illustrated in the MC plots of Fig. 4.2, where at arrival the final MC samples are spread in a roughly linear shape, as evidenced by the MC position ellipsoid in a red dashed line, whereas the predicted UT ellipsoid exhibits a larger transversal dimension.

The magnitudes of the impulsive maneuvers for the OCP and the ROCP are reported in Table 4.3, together with the angle ϕ measured counterclockwise from the positive x axis.

A parametric study is performed varying the value of η starting from the unitary



(a) OCP solution.



(b) ROCP solution.

Figure 4.1. Nominal spacecraft trajectories with 95%-confidence position ellipses.

value, thus coincident with the OCP solution, to the value of 1.15, hence allowing an increment of 15% from the OCP solution. In Table 4.4, the values of the cumulative imparted Δv, the maximum allowed Δv, the ToF, and $\mathrm{tr}\,(\boldsymbol{P}_N)$ are reported with the respective value of η. For every value, the ToF is exactly equal to $\Delta\widetilde{t}_{\max}$ as expected (as said, longer ToF generally corresponds to less fuel expenditure), as well as Δv_{tot} which is in every case, except for $\eta = 1.15$, is equal to the maximum allowed value of $\eta\Delta v^*$. The cost function strictly monotonically decreases with η, proving again effectiveness of the OL control to reduce the final dispersion. In Fig. 4.3, the MC

Table 4.2. Comparison between the OCP and ROCP solutions with the related MC analysis.

Quantity	OCP	OCP (MC)	ROCP-OL	ROCP-OL (MC)
Δv_{tot}	0.331	0.331	0.347	0.347
$\Delta v_{\max}$	-	-	0.347	0.347
Δt_1	1.733	1.733	1.828	1.828
Δt_2	2.913	2.913	2.859	2.859
Δt_3	2.358	2.358	2.313	2.313
Δt_{tot}	7.000	7.000	7.000	7.000
μ^x_{N+}	-	2.092×10^{-3}	-	1.113×10^{-4}
μ^y_{N+}	-	6.460×10^{-3}	-	1.482×10^{-3}
$\mu^{v_x}_{N+}$	-	1.252×10^{-3}	-	9.601×10^{-4}
$\mu^{v_y}_{N+}$	-	6.800×10^{-3}	-	5.898×10^{-4}
$\operatorname{tr}(\boldsymbol{P}_N)$	6.848×10^{-3}	5.591×10^{-3}	2.820×10^{-3}	3.349×10^{-3}
σ^x_N	4.270×10^{-2}	4.009×10^{-2}	3.163×10^{-2}	3.182×10^{-2}
σ^y_N	4.882×10^{-2}	4.184×10^{-2}	2.843×10^{-2}	3.215×10^{-2}
$\sigma^{v_x}_N$	1.866×10^{-2}	2.090×10^{-2}	1.318×10^{-2}	1.553×10^{-2}
$\sigma^{v_y}_N$	4.789×10^{-2}	4.239×10^{-2}	2.894×10^{-2}	3.258×10^{-2}

Table 4.3. Comparison between the OCP and ROCP solutions for the magnitudes and directions of the impulsive maneuvers.

Maneuver	OCP		ROCP	
	$\|\boldsymbol{u}\|$	ϕ	$\|\boldsymbol{u}\|$	ϕ
$\boldsymbol{u}_1$	3.915×10^{-2}	258.169	3.445×10^{-2}	257.382
$\boldsymbol{u}_2$	8.096×10^{-2}	10.792	1.009×10^{-1}	345.120
$\boldsymbol{u}_3$	1.017×10^{-1}	88.873	1.100×10^{-1}	103.691
$\boldsymbol{u}_4$	1.088×10^{-1}	234.194	1.019×10^{-1}	237.316

trajectories for the two highest values of η in the parametric study are plotted. Note the tendency to extend the first ballistic arc to reduce the final state dispersion, as confirmed in Table 4.4.

In Fig. 4.4, $\operatorname{tr}(\boldsymbol{P}_N)$ together with the standard deviations $\sigma_{\Delta r}$ and $\sigma_{\Delta v}$ of the position and velocity magnitude deviation from the mean value are plotted. In particular, these two quantities are obtained as

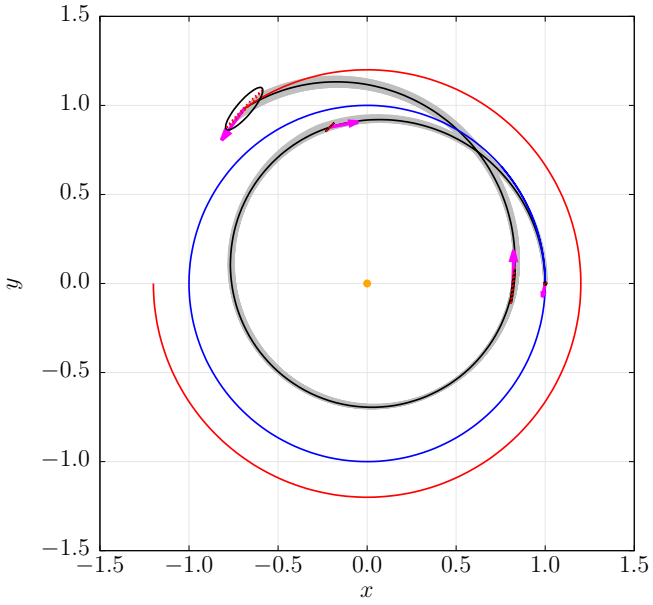
$$\sigma_{\Delta r} = \sqrt{\frac{1}{M}\sum_{k=1}^{M}\left(r_k - \mathbb{E}\left[r_k\right]\right)^2} \tag{4.24}$$

$$\sigma_{\Delta v} = \sqrt{\frac{1}{M}\sum_{k=1}^{M}\left(v_k - \mathbb{E}\left[v_k\right]\right)^2} \tag{4.25}$$

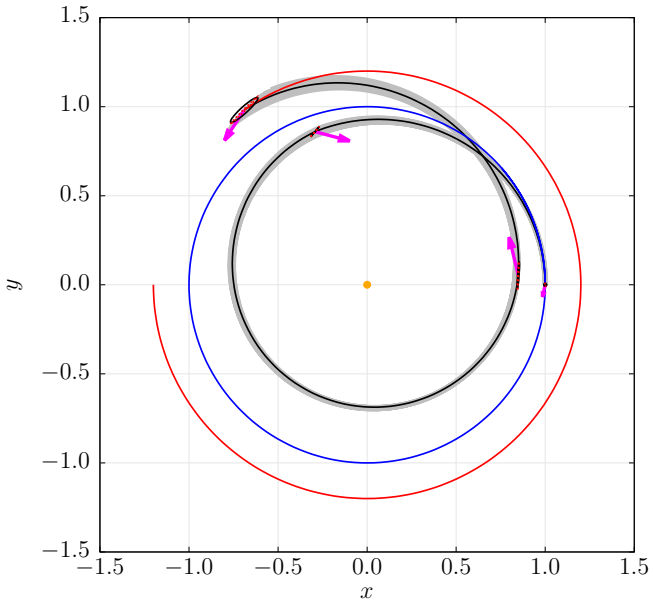
where $M = 10^4$ is the number of MC runs, and

$$r_k = \|\boldsymbol{r}_k - \mathbb{E}\left[\boldsymbol{r}_k\right]\| \tag{4.26}$$

$$v_k = \|\boldsymbol{v}_k - \mathbb{E}\left[\boldsymbol{v}_k\right]\| \tag{4.27}$$



(a) OCP solution.

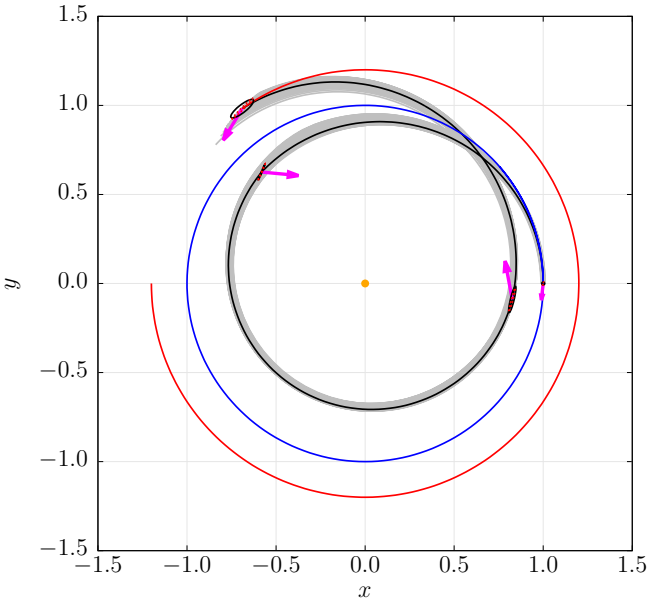


(b) ROCP solution.

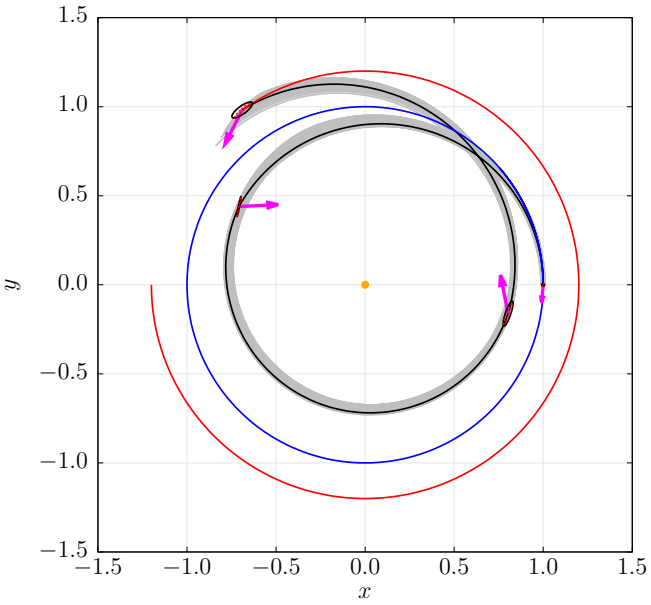
Figure 4.2. MC trajectories (in gray) with corresponding position ellipses (dashed red line) and UT position ellipses (solid black line).

where $\boldsymbol{r}_k$ and $\boldsymbol{v}_k$ are the position and velocity of the final state $\boldsymbol{x}_{N+}$ of the k-th MC run. A rapid decrease in the cost function is observed in the initial part of the η range, with the solution reaching a plateau around the final values. This behavior indicates that further enhancement of robustness beyond those η values is nearly impossible.

A final parametric analysis is performed for different values of the perturbation



(a) $\eta = 1.10$.



(b) $\eta = 1.15$.

Figure 4.3. MC trajectories for the two highest values of η in the parametric analysis.

matrix $\boldsymbol{Q}$, defined similarly to $\widetilde{\boldsymbol{P}}_0$ as

$$\boldsymbol{Q} = \text{diag}\left[(\sigma_w^r)^2\, \boldsymbol{I}_2,\, 0,\, (\sigma_w^v)^2\, \boldsymbol{I}_2^2,\, 0\right] \tag{4.28}$$

The main solution parameters together with the corresponding values of σ_w^r and σ_w^r are reported in Table 4.5, where the other problem data are still the ones in Table 4.1. As expected, a greater perturbation matrix leads to a larger trace of the final covariance matrix. It is noteworthy that the ToF t_{tot} is slightly less than the

Table 4.4. Main solution parameters for different values of η

η	1.00	1.05	1.10	1.15
Δv_{tot}	0.331	0.347	0.364	0.377
Δv_{max}	0.331	0.347	0.364	0.380
Δt_1	1.733	1.828	2.132	2.322
Δt_2	2.913	2.859	2.441	2.218
Δt_3	2.358	2.313	2.427	2.460
Δt_{tot}	7.000	7.000	7.000	7.000
$\text{tr}\,(\boldsymbol{P}_N)$	6.820×10^{-3}	2.819×10^{-3}	1.737×10^{-3}	1.383×10^{-3}

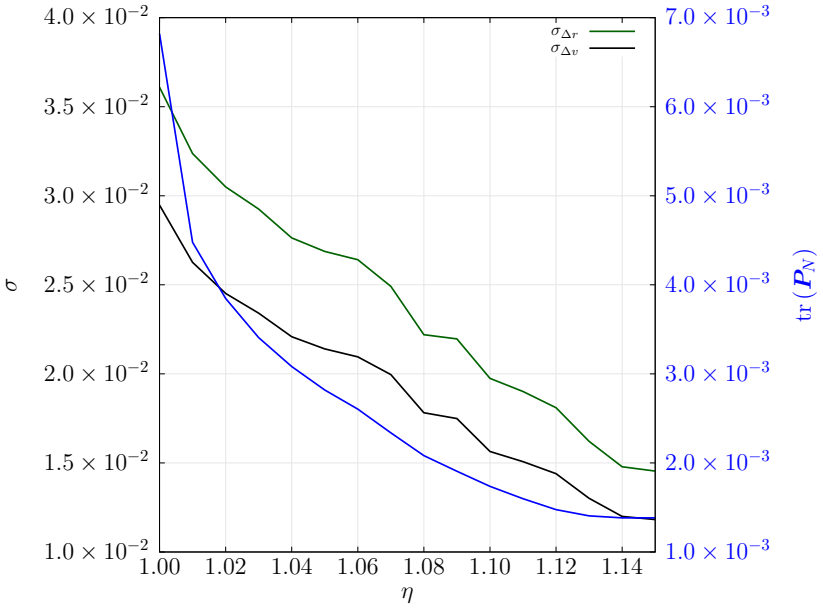


Figure 4.4. Trace and standard deviations of mean Δr and Δv as functions of η.

maximum allowable value of 7 for the largest perturbations considered, suggesting that a shorter flight duration is necessary to comply with all mission constraints.

Table 4.5. Main solution parameters for different values of σ_w^r and σ_w^r.

σ_w^r	0.00	5.000×10^{-6}	1.000×10^{-5}	5.000×10^{-5}
σ_w^v	0.00	5.000×10^{-6}	1.000×10^{-5}	5.000×10^{-5}
Δv_{tot}	0.347	0.347	0.347	0.347
Δv_{max}	0.347	0.347	0.347	0.347
Δt_{tot}	7.000	7.000	6.999	6.812
$\text{tr}\,(\boldsymbol{P}_N)$	2.819×10^{-3}	3.034×10^{-3}	7.359×10^{-3}	2.934×10^{-2}

Chapter 5

Robust closed-loop optimization of impulsive trajectories

The approach used in Chapter 4 to deal with OL robust optimization problem will now be expanded to the case of a feedback (closed-loop) control law. Even though robust OL control laws allow for some limitation on state dispersion, ignoring the actual spacecraft flight path has inherent weaknesses in reducing the state dispersion and may not be sufficient to deliver a robust-enough trajectory.

A natural enhancement is to incorporate a feedback component into the control strategy, enabling corrective maneuvers tailored to the current flight conditions. In this respect, this chapter introduces a methodology for optimizing robust trajectories for impulsive missions, featuring potential space mission applications such as an Earth–asteroid–Earth trajectory and station-keeping within a Near-Rectilinear Halo Orbit (NRHO) around the Moon.

5.1 General formulation

A general description of the proposed robust trajectory optimization problem is provided in this section. The objective of the proposed methodology is to formulate an ROCP for an impulsive transfer, where a feedback control law is used to minimize the cumulative Δv, having fixed initial and final state, described as position and velocity, fixed ToF, and control maneuvers bounded in magnitude. The state dispersion is fixed for the initial state and limited at the arrival.

As in Chapter 4, the spacecraft is modeled as a point-mass object and its state $\boldsymbol{x}(t) = \left[\boldsymbol{r}(t)^T\, \boldsymbol{v}(t)^T\right]^T$ is defined in terms of his Cartesian position $\boldsymbol{r}(t)$ and velocity $\boldsymbol{v}(t)$. The initial and final transfer epochs are fixed and denoted as $\widetilde{t}_f$ and $\widetilde{t}_i$, respectively. As a major difference with the problem stated in Chapter 4, the trajectory is divided into N ballistic arcs of equal duration, joined by impulsive velocity changes, which represent the spacecraft maneuvers. Additional impulsive maneuvers are performed at the departure and the arrival. A uniform time-grid with $N+1$ maneuvering points, or nodes, is thus defined as

$$\widetilde{t}_i = t_0 \leqslant t_1 \leqslant \cdots \leqslant t_N = \widetilde{t}_f \tag{5.1}$$

where duration of the k-th arc is $\Delta t_k = t_k - t_{k-1} = (\widetilde{t}_f - \widetilde{t}_i)/N = \Delta T/N$, with ΔT as the total ToF.

Adopting the same notation as used for the OL case, the values of the variables immediately before and after the k-th impulse are indicated by the subscripts $-$

and +, respectively. Position is considered equal before and after each node, as $\boldsymbol{r}_{k-} = \boldsymbol{r}_{k+}$, while instantaneous changes in velocity occur due to the impulsive maneuvers, as $\boldsymbol{v}_{k+} = \boldsymbol{v}_{k-} + \boldsymbol{u}_k$, where the subscript k denotes quantities at epoch t_k, and the control is again given by Δv_k. The update law for the state after a maneuver is

$$\boldsymbol{x}_{k+} = \boldsymbol{x}_{k-} + \boldsymbol{B}\boldsymbol{u}_k \qquad \forall k = 0, \dots, N \tag{5.2}$$

with $\boldsymbol{B} = [\boldsymbol{0}_{3\times 3}\, \boldsymbol{I}_3]^T$. The control magnitude $\|\boldsymbol{u}_k\|$ is constrained below an user-defined threshold $\widetilde{u}_{\max}$.

5.1.1 Linear control law

The main idea behind the proposed methodology is the definition of a linear-state feedback control law for the impulsive control $\boldsymbol{u}_k$, which is defined as

$$\boldsymbol{u}_k = \boldsymbol{\nu}_k + \boldsymbol{K}_k \left(\boldsymbol{x}_{k-} - \mathbb{E}\left[\boldsymbol{x}_{k-}\right]\right) = \boldsymbol{\nu}_k + \boldsymbol{K}_k \left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right) \qquad \forall k = 0, \dots, N \tag{5.3}$$

where $\boldsymbol{\nu}_k \in \mathbb{R}^3$ represents the deterministic (or feedforward) component of the control, while the feedback term $\boldsymbol{K}_k \left(\boldsymbol{x}_{k-} - \mathbb{E}\left[\boldsymbol{x}_{k-}\right]\right)$ accounts for the difference between the measured and expected states. Here, $\boldsymbol{K}_k \in \mathbb{R}^{3\times 6}$ is a feedback gain matrix, whose value has to be determined with the optimization process. The feedback term is a stochastic component of the control, directly proportional to the deviation of the actual state $\boldsymbol{x}_{k-} = \mathbb{E}\left[\boldsymbol{x}_{k-}\right]$ before the k-th maneuver and the mean state at the same epoch $\boldsymbol{\mu}_{k-}$. Thus, the larger the deviation, the larger the corrective maneuver imparted on the spacecraft.

Given the CL control law of Eq. (5.3), the on-board computer requires both the knowledge of the current state provided by the navigation subsystem and the control parameters given by the deterministic maneuver $\boldsymbol{\nu}_k$, the gain matrix $\boldsymbol{K}_k$, and the mean state right before the maneuver $\boldsymbol{\mu}_{k-}$.

The expected value of the k-th control maneuver $\boldsymbol{u}_k$ is

$$\mathbb{E}\left[\boldsymbol{u}_k\right] = \boldsymbol{\nu}_k \tag{5.4}$$

Having $\boldsymbol{P}_{k-}$ as the covariance matrix of $\boldsymbol{x}_{k-}$, the covariance matrix is analytically derived through the following lemma.

Lemma 5.1.1. *Considering the closed-loop control expression of Eq. (5.3)*

$$\boldsymbol{u}_k = \boldsymbol{\nu}_k + \boldsymbol{K}_k \left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right) \qquad \forall k = 0, \dots, N \tag{5.5}$$

The covariance of $\boldsymbol{u}_k$ can be formulate by definition as

$$\mathrm{Cov}\left[\boldsymbol{u}_k\right] = \boldsymbol{P}_k^u = \boldsymbol{K}_k \boldsymbol{P}_{k-} \boldsymbol{K}_k^T \qquad \forall k = 0, \dots, N \tag{5.6}$$

Proof.

$$\boldsymbol{P}_k^{\boldsymbol{u}} = \mathbb{E}\left[\left(\boldsymbol{u}_k - \mathbb{E}\left[\boldsymbol{u}_k\right]\right)\left(\boldsymbol{u}_k - \mathbb{E}\left[\boldsymbol{u}_k\right]\right)^T\right] = \tag{5.7a}$$

$$= \mathbb{E}\left[\left(\boldsymbol{u}_k - \boldsymbol{\nu}_k\right)\left(\boldsymbol{u}_k - \boldsymbol{\nu}_k\right)^T\right] = \tag{5.7b}$$

$$= \mathbb{E}\left[\left[\boldsymbol{K}_k\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)\right]\left[\boldsymbol{K}_k\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)\right]^T\right] = \tag{5.7c}$$

$$= \mathbb{E}\left[\boldsymbol{K}_k\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)^T\boldsymbol{K}_k^T\right] = \tag{5.7d}$$

$$= \boldsymbol{K}_k\mathbb{E}\left[\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)\right]\boldsymbol{K}_k^T = \tag{5.7e}$$

$$= \boldsymbol{K}_k\boldsymbol{P}_{k-}\boldsymbol{K}_k^T \tag{5.7f}$$

□

The previous expression will be used in the formulation of the cost function and the control constraints, as described in the following sections.

5.1.2 Perturbed state propagation

As highlighted in the previous chapters, an accurate UP is of paramount importance in stochastic optimization, as the optimized control is designed on the knowledge of the state distribution. Thus, an incorrect evaluation of the state uncertainty would lead to an ineffective control action when applied to the actual mission. If the analysis of the state distribution is limited to its first two statistic moments, that is, mean and covariance, the state probability distribution could be easily propagated through a ballistic arc, using one of the UP techniques available in literature (see Chapter 3). As already mentioned, the tool used in this regard in the test cases presented in the last sections of this chapter is the UT, given its reasonable balance between accuracy and computational effort.

Similarly to the OL case in Chapter 4, the perturbation term of the SDE in Eq. (2.54) is neglected when propagating the SPs, that are thus propagated using deterministic EoM. The random contribution is instead again modeled as an instantaneous, additive, Gaussian perturbation $\overline{\boldsymbol{Q}}$ that takes place at the end of each arc, representing the overall effects of the stochastic perturbations over a finite-time arc. Hence, if $\boldsymbol{P}'_{k-}$ is the covariance matrix estimated by the UT, the covariance matrix that is actually considered after propagation is $\boldsymbol{P}_{k-} = \boldsymbol{P}'_{k-} + \overline{\boldsymbol{Q}}$. The matrix $\overline{\boldsymbol{Q}}$ is estimated along a reference trajectory according to the methodology in Appendix A. In this respect, the function $\boldsymbol{g}\left(\boldsymbol{x},\, \boldsymbol{u},\, t\right)$ is approximated as a constant matrix $\boldsymbol{G} \in \mathbb{R}^{6\times 3}$, as

$$\boldsymbol{g}\left(\boldsymbol{x},\, \boldsymbol{u},\, t\right) = \boldsymbol{G} = \begin{bmatrix}\left(\sigma_w^r\right)^2\boldsymbol{I}_3 \\ \left(\sigma_w^v\right)^2\boldsymbol{I}_3\end{bmatrix} \tag{5.8}$$

where $\sigma_w^r \in \mathbb{R}$ and $\sigma_w^v \in \mathbb{R}$, considered as parameters of the problem, model the variance of the position and velocity random disturbance, respectively.

The propagation of the mean state and the covariance matrix along the k-th ballistic arc, i.e. from the $(k-1)$-th and the k-th maneuver, is thus given by the expression

$$\left\{\boldsymbol{\mu}_{k-},\, \boldsymbol{P}_{k-}\right\} = \mathcal{F}^*\left(\boldsymbol{\mu}_{(k-1)+},\, \boldsymbol{P}_{(k-1)+},\, \Delta t_k;\, \overline{\boldsymbol{Q}}\right) \qquad \forall k = 1, \ldots, N \tag{5.9}$$

where $\mathcal{F}^*(\cdot)$ stands for the UP tool function returning the mean state and covariance obtained by propagating the state $\boldsymbol{x}_{(k-1)+} \sim \mathcal{N}\left(\boldsymbol{\mu}_{(k-1)+},\, \boldsymbol{P}_{(k-1)+}\right)$ from time $t_{(k-1)}$

to time t_k, while accounting for the (cumulative) perturbation of covariance $\overline{\boldsymbol{Q}}$. In this particular work, as the UT is chosen for UP; thus, the generic expression $\mathcal{F}^*(\cdot)$ in Eq. (5.9) is substituted with $\mathcal{F}^{\mathrm{UT}}(\cdot)$.

While propagating the state distribution over time, jumps occur due to the presence of impulsive maneuvers, as in Eq. (5.2). The effect of the (impulsive) linear feedback control of Eq. (5.3) on the state probability distribution after the impulse is evaluated by taking the expected value and covariance of $\boldsymbol{x}_{k+}$. The spected values results in

$$\boldsymbol{\mu}_{k+} = \mathbb{E}\left[\boldsymbol{x}_{k+}\right] = \boldsymbol{\mu}_{k-} + \boldsymbol{B}\boldsymbol{\nu}_k \qquad \forall k = 0, \dots, N \tag{5.10}$$

while the covariance matrix is obtained through the following lemma.

Lemma 5.1.2. *Considering the state update in Eq. (5.2), the covariance matrix of $\boldsymbol{x}_{k+}$ results as*

$$\boldsymbol{P}_{k+} = \mathrm{Cov}\left[\boldsymbol{x}_{k+}\right] = (\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k)\boldsymbol{P}_{k-}(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k)^T \qquad \forall k = 0, \dots, N \tag{5.11}$$

Proof.

$$\begin{aligned}
\boldsymbol{P}_{k+} &= \mathbb{E}\left[\left(\boldsymbol{x}_{k+} - \mathbb{E}\left[\boldsymbol{x}_{k+}\right]\right)\left(\boldsymbol{x}_{k+} - \mathbb{E}\left[\boldsymbol{x}_{k+}\right]\right)^T\right] = && \text{(5.12a)}\\
&= \mathbb{E}\left[\left(\boldsymbol{x}_{k+} - \boldsymbol{\mu}_{k+}\right)\left(\dots\right)^T\right] = && \text{(5.12b)}\\
&= \mathbb{E}\left[\left(\boldsymbol{x}_{k-} + \boldsymbol{B}\boldsymbol{u}_k - \boldsymbol{\mu}_{k-} - \boldsymbol{B}\boldsymbol{\nu}_k\right)\left(\dots\right)^T\right] = && \text{(5.12c)}\\
&= \mathbb{E}\left[\left[\boldsymbol{x}_{k-} + \boldsymbol{B}\boldsymbol{K}_k\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right) - \boldsymbol{\mu}_{k-}\right]\left[\dots\right]^T\right] = && \text{(5.12d)}\\
&= (\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k)\,\mathbb{E}\left[\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)\left(\dots\right)^T\right](\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k)^T = && \text{(5.12e)}\\
&= (\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k)\,\boldsymbol{P}_{k-}\,(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k)^T && \text{(5.12f)}
\end{aligned}$$

where the dots are used in place of the identical expression in the previous brackets pair. □

Within the optimization process, the state mean and covariance are estimated along the trajectory by alternatively recurring to Eqs. (5.10) and (5.11) to update the state uncertainty after the impulsive maneuver, and an UP tool as in Eq. (5.9) to propagate along the ballistic arc up to a moment before the following impulsive maneuver, starting from the assigned covariance $\boldsymbol{P}_{0-} = \widetilde{\boldsymbol{P}}_{0-}$. The value of the estimated mean state is used to enforce boundary constraints, while the estimated covariance matrix is used in both path constraints and cost function, as will be explained in the next sections.

5.1.3 Boundary constraints

In a deterministic context, boundary constraints enforce (without any uncertainty) the state to be coincident at a given epoch to an assigned state, e.g. the final position and velocity of a celestial body at arrival time. However, since uncertainty on the state is now allowed, also boundary conditions must be treated in stochastic terms, having in my mind however the ultimate goal of the trajectory, that is, to deliver the spacecraft to a precise target at a precise time.

At the initial epoch, the departure state and the related uncertainty is generally well known; thus, the initial boundary constraints are intuitively formulated to assign

the departure mean state and covariance matrix to the fixed values deriving from the mission scenario. For instance, the initial state uncertainty could be given by the launcher performance statistics when delivering the payload after the final burnout of the last stage. The initial boundary conditions are thus expressed simply as

$$\boldsymbol{\mu}_{0-} = \widetilde{\boldsymbol{\mu}}_{0-} = \left[\widetilde{\boldsymbol{r}}_{0-}^T \; \widetilde{\boldsymbol{v}}_{0-}^T\right]^T \tag{5.13}$$

$$\boldsymbol{P}_{0-} = \widetilde{\boldsymbol{P}}_{0-} \tag{5.14}$$

The final boundary constraints are handled in a different way. If the final mean state is straightforwardly assigned to the final target state, the actual rendezvous to the assigned state is not exactly guaranteed. Indeed, the final approach to the target is enforced on average, but in case the final state uncertainty is relatively large, the spacecraft may completely miss the target, although not far from the expected average final state. It is hence fundamental to enforce also conditions on the final dispersion, as to minimize the chance of being too far from the state goal. A possible idea to limit the "size" of the final dispersion is to impose a threshold to the eigenvalues of the last covariance matrix $\boldsymbol{P}_{N+}$, that is, after the final impulsive control $\boldsymbol{u}_{N+}$. In this work, the threshold on the eigenvalues is imposed in particular to the two sub-matrices of position and velocity of $\boldsymbol{P}_{N+}$, namely, the top left 3×3 square matrix for position, $\boldsymbol{P}_{N+}^r$, and the bottom right 3×3 square matrix for velocity, $\boldsymbol{P}_{N+}^v$. Denoting with $\boldsymbol{\lambda}(\boldsymbol{P})$ the vector of eigenvalues of the matrix $\boldsymbol{P}$ in argument, the final constraints are specified as

$$\boldsymbol{\mu}_{N+} = \widetilde{\boldsymbol{\mu}}_{N+} \tag{5.15}$$

$$\boldsymbol{\lambda}\left(\boldsymbol{P}_{N+}^r\right) - \left(\sigma_{N+}^r\right)^2 \mathbf{1}_3 \leqslant \mathbf{0}_3 \tag{5.16}$$

$$\boldsymbol{\lambda}\left(\boldsymbol{P}_{N+}^v\right) - \left(\sigma_{N+}^v\right)^2 \mathbf{1}_3 \leqslant \mathbf{0}_3 \tag{5.17}$$

where σ_{N+}^r and σ_{N+}^v are maximum allowed values of the standard deviation of position and velocity at the arrival, respectively.

Alternatively, Eqs. (5.16) and (5.17) may be simplified as

$$\sqrt{\lambda_{\max}\left(\boldsymbol{P}_{N+}^r\right)} \leqslant \sigma_{N+}^r \tag{5.18}$$

$$\sqrt{\lambda_{\max}\left(\boldsymbol{P}_{N+}^v\right)} \leqslant \sigma_{N+}^v \tag{5.19}$$

The latter formulation focuses on limiting the magnitude of state dispersion, neglecting the shape pf the dispersion ellipsoid. Note that other ways to constrain the final state dispersion could be considered, without substantially changing the problem solution. For instance, a threshold could be imposed on the trace the final covariance matrix.

5.1.4 Chance-constrained cost function and control constraints

As discussed in Sec. 2.3.3, a stochastic variable requires a stochastic formulation of the cost function and any related constraints, as a traditional deterministic expression is not applicable in the context of an SOCP. In a CL control setup, the control is modeled as a function of a stochastic variable, as stated in Eq. (5.3), rendering the control itself stochastic. Consequently, when the objective involves minimizing the cumulative control magnitude, it necessitates a formulation that statistically confines the control within a specified confidence level. The aim of this

section is to develop a deterministic expression for the control in a chance-constrained framework. This formulation will be instrumental in defining both the cost function and the constraints on control magnitude.

The chance-constraint formulation is based on the assumption of a Gaussian distribution for the state $\boldsymbol{x}_{k-}$ before the impulsive maneuver. As demonstrated in the test cases presented later, this assumption remains valid for scenarios characterized by limited state dispersion, a condition achievable through the implementation of a robust control strategy. Considering the feedback control law in the form of Eq. (5.3), the control vector $\boldsymbol{u}_k$ at time t_k is consequently a random Gaussian vector, that is $\boldsymbol{u}_k \sim \mathcal{N}(\boldsymbol{\nu}_k, \boldsymbol{P}_k^u)$. The following theorem from Ridderhof et al. [102] provides a probabilistic bound for $\|\boldsymbol{u}\|$.

Theorem 5.1.3. *Let $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{P}_z)$ be an w-dimensional random vector with $w \geqslant 1$. Let also $\sigma = \sqrt{\lambda_{\max}(\boldsymbol{P}_z)}$ be the square-root of the maximum eigenvalue of the matrix $\boldsymbol{P}_z$, $\rho > 0$ and $\beta \in (0,1)$. Then*

$$\|\boldsymbol{\mu}_z\| + \gamma(\beta)\sigma \leqslant \rho \Rightarrow \Pr(\|\boldsymbol{z}\| \leqslant \rho) \geqslant 1 - \beta \tag{5.20}$$

with $\gamma(\beta) = \sqrt{2\log\frac{1}{\beta}} + \sqrt{w}$.

Proof of this theorem is included in the same cited work.

By applying this theorem with $\boldsymbol{z} = \boldsymbol{u}_k$, the probabilistic constraint on the control magnitude bound Eq. (2.57) can be rewritten as

$$\|\boldsymbol{\nu}_k\| + \gamma(\beta)\sqrt{\lambda_{\max}\left(\boldsymbol{P}_k^u\right)} \leqslant \widetilde{u}_{\max} \qquad \forall k = 0, \ldots, N \tag{5.21}$$

where $\|\boldsymbol{\nu}_k\| = \Delta v_{\mathrm{d},k}$ is the norm of the deterministic component of control $\boldsymbol{u}_k$ and $\gamma(\beta)\sqrt{\lambda_{\max}(\boldsymbol{P}_k^u)} = \Delta v_{\mathrm{s},k}$ is a measure (up to a confidence level $1-\beta$) of the control magnitude due to the CL term. The bound in Eq. (5.21) has been chosen for the sake of simplicity, although other formulations are possible [203]. Other expressions could be easily selected depending on the researcher's need, as the presented method do not depends on specific manipulation to obtain a convex reformulation of Eq. (5.21) (as is the case of [115]).

The theorem also allows to express a cost function on the cumulative control magnitude in a straightforward form. Assuming the same violation parameter β, the cost function reads

$$J = \sum_{k=0}^{N}\left(\|\boldsymbol{\nu}_k\| + \gamma(\beta)\sqrt{\lambda_{\max}(\boldsymbol{P}_k^u)}\right) = \Delta v_{\mathrm{d}} + \Delta v_{\mathrm{s}} = \Delta v_{\mathrm{tot}} \tag{5.22}$$

where the first term of Eq. (5.22) accounts for the cumulative deterministic Δv cost $\Delta v_{\mathrm{d}} = \sum_{k=0}^{N} \Delta v_{\mathrm{d},k}$, while the second term measures the CL control effort $\Delta v_{\mathrm{s}} = \sum_{k=0}^{N} \Delta v_{\mathrm{s},k}$, which depends on the dispersion of the control vector through the matrix $\boldsymbol{P}_k^u$ and β. Once indeed the control parameters of a given k-th maneuver are selected, Theorem 5.1.3 states that the quantity on the left side of Eq. (5.21) is below the magnitude of the actual executed control, with a confidence level of $1-\beta$. Thus, lowering the value of $\|\boldsymbol{\nu}_k\| + \gamma(\beta)\sqrt{\lambda_{\max}(\boldsymbol{P}_k^u)}$ is equivalent to lowering the stochastic upper bound of $\|\boldsymbol{u}_k\|$, that is, the maximum allowed value of $\|\boldsymbol{u}_k\|$ with $1-\beta$ confidence level. Naturally, the same applies for the sum of all the control maneuvers sequence. Alternative expressions that would account for the mean value

and the uncertainty level of $\|\boldsymbol{u}_k\|$ would prove equivalently efficient as stochastic merit index, as it would serve nonetheless as a stochastic upper bound with a given confidence level. However, for the sake of consistency and to retain a measure on the confidence level the same expression is employed.

Note that this formulation allows to increase the overall deterministic cost Δv_d, when it is beneficial for reducing the CL control term Δv_s, so as to reduce the overall total velocity increment $\Delta v_\mathrm{tot} = \Delta v_\mathrm{d} + \Delta v_\mathrm{s} = J$. The overall ROCP is thus formally formulated as

$$\min_{\boldsymbol{\Theta}} \text{Eq. (5.22)} \tag{5.23a}$$

$$\text{s.t.} \quad \text{Eqs. (5.6), (5.9)} - \text{(5.17), (5.21)} \tag{5.23b}$$

$$\tag{5.23c}$$

where in the optimization variables vector $\boldsymbol{\Theta}$ the reference mean states, the deterministic control, and the gain matrices are included as

$$\boldsymbol{\Theta} = \{\boldsymbol{\mu}_{k-},\, \boldsymbol{\nu}_k,\, \boldsymbol{K}_k,\, \forall k = 0, \dots, N\} \tag{5.24}$$

which results as a vector of dimension $(n^2 + 3n)(N+1)/2$, having however the initial mean state $\boldsymbol{\mu}_{0-}$ fixed to an assigned vector as in Eq. (5.13).

The primary advantage of the approach outlined here, over traditional methods that derive a nominal trajectory before identifying a stabilizing control, lies in its heightened responsiveness to system dynamics that are particularly sensitive to external disturbances, such as strong nonlinearities or rapidly changing conditions. Consequently, the efficacy of this method is expected to be greater in environments where these dynamic complexities are more pronounced, thereby directly addressing the challenges posed by such systems.

5.2 Multi-legs missions

The trajectory considered so far represents a single leg, that is, an alternative sequence of impulsive maneuvers and ballistic arcs that link two assigned states defined as position and velocity. Extension to the case of a multiple-legs mission is straightforward. A multiple-leg trajectory is defined as a sequence of single legs where the assigned final state of a single leg coincides with the assigned initial state of the next leg. Hence, to define a multiple-leg trajectory, it is now necessary to provide the number L of legs, the number N_i of ballistic arcs of each i-th leg, and the sequence of initial and final state of each i-th leg $\boldsymbol{\mu}_{i,\,0-}$ and $\boldsymbol{\mu}_{i,\,N_i+}$, thus requiring the following linkage conditions to ensure the continuity of the trajectory

$$\boldsymbol{\mu}_{i,\,N_i+} = \boldsymbol{\mu}_{i+1,\,0-} \qquad \forall i = 1, \dots, L-1 \tag{5.25}$$

to which the final conditions of all the legs are added as

$$\boldsymbol{\mu}_{i,\,N_i+} = \widetilde{\boldsymbol{\mu}}_{i,\,N_i+} \qquad \forall i = 1, \dots, L \tag{5.26}$$

The initial conditions are thus written as

$$\boldsymbol{\mu}_{1,\,0-} = \widetilde{\boldsymbol{\mu}}_{1,\,0-} = \left[\widetilde{\boldsymbol{r}}^T_{1,\,0-}\ \widetilde{\boldsymbol{v}}^T_{1,\,0-}\right]^T \tag{5.27}$$

$$\boldsymbol{P}_{1,\,0-} = \widetilde{\boldsymbol{P}}_{1,\,0-} \tag{5.28}$$

while the terminal constraints are instead written as

$$\boldsymbol{\lambda}\left(\boldsymbol{P}^{r}_{i,\,N_i+}\right) - \left(\sigma^{r}_{i,\,N_i+}\right)^2 \mathbf{1}_3 \leqslant \mathbf{0}_3 \tag{5.29}$$

$$\boldsymbol{\lambda}\left(\boldsymbol{P}^{v}_{i,\,N_i+}\right) - \left(\sigma^{v}_{i,\,N_i+}\right)^2 \mathbf{1}_3 \leqslant \mathbf{0}_3 \tag{5.30}$$

$$\boldsymbol{\mu}_{i,\,N_i+} = \widetilde{\boldsymbol{\mu}}_{i,\,N_i+} = \left[\widetilde{\boldsymbol{r}}^{T}_{i,\,N_i+}\ \widetilde{\boldsymbol{v}}^{T}_{i,\,N_i+}\right]^{T} \qquad \forall i = 1,\ldots,L \tag{5.31}$$

To avoid inconsistencies (e.g., duplicity of the Δvs), no control can be imparted at the initial point of any leg, but the first one. The following constraints thus hold

$$\boldsymbol{\nu}_{i,\,0} = \mathbf{0}_3 \qquad \boldsymbol{K}_{i,\,0} = \mathbf{0}_{3\times 6} \qquad \forall i = 2,\ldots,L \tag{5.32}$$

which, in turn, leads to

$$\boldsymbol{P}_{i,\,0+} = \boldsymbol{P}_{i,\,0-} = \boldsymbol{P}_{i-1,\,N_{i-1}+} \tag{5.33}$$

$$\boldsymbol{P}^{u}_{i,\,0} = \mathbf{0}_{3\times 3} \qquad \forall i = 2,\ldots,L \tag{5.34}$$

Adding the notation on the leg number, the equations related to the feedback control law are now written as

$$\boldsymbol{u}_{i,\,k} = \boldsymbol{\nu}_{i,\,k} + \boldsymbol{K}_{i,\,k}\left(\boldsymbol{x}_{i,\,k-} - \boldsymbol{\mu}_{i,\,k-}\right) \tag{5.35}$$

$$\boldsymbol{P}^{u}_{i,\,k} = \boldsymbol{K}_{i,\,k}\boldsymbol{P}_{i,\,k-}\boldsymbol{K}^{T}_{i,\,k} \qquad \forall k = 0,\ldots,N_i \qquad \forall i = 1,\ldots,L \tag{5.36}$$

and the equations related to the state propagation as

$$\begin{aligned}&\{\boldsymbol{\mu}_{i,\,k-},\ \boldsymbol{P}_{i,\,k-}\} = \\ &= \mathcal{F}^{*}\left(\boldsymbol{\mu}_{i,\,(k-1)+},\ \boldsymbol{P}_{i,\,(k-1)+},\ \Delta t_{i,\,k};\ \overline{\boldsymbol{Q}}\right) \quad \forall k = 1,\ldots,N_i \qquad \forall i = 1, ldots, L\end{aligned} \tag{5.37}$$

$$\boldsymbol{\mu}_{i,\,k+} = \boldsymbol{\mu}_{i,\,k-} + \boldsymbol{B}\boldsymbol{\nu}_{i,\,k} \tag{5.38}$$

$$\boldsymbol{P}_{i,\,k+} = \boldsymbol{S}_{i,\,k}\boldsymbol{P}_{i,\,k-}\boldsymbol{S}^{T}_{i,\,k} \qquad \forall k = 0,\ldots,N_i \qquad \forall i = 1,\ldots,L \tag{5.39}$$

with $\boldsymbol{S}_{i,k} = \boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_{i,k}$.

The path constraint on control magnitude is now

$$\|\boldsymbol{\nu}_{i,\,k}\| + \gamma(\beta)\sqrt{\lambda_{\max}\left(\boldsymbol{P}^{u}_{i,\,k}\right)} \leqslant \widetilde{u}_{\max} \qquad \forall k = 0,\ldots,N_i \qquad \forall i = 1,\ldots,L \tag{5.40}$$

and eventually, the cost function reads as

$$J = \sum_{i=1}^{L}\sum_{k=0}^{N_i}\left(\|\boldsymbol{\nu}_{i,\,k}\| + \gamma(\beta)\sqrt{\lambda_{\max}(\boldsymbol{P}^{u}_{i,\,k})}\right) \tag{5.41}$$

The multiple-legs ROCP is thus formulated as

$$\min_{\boldsymbol{\Theta}} \text{Eq. (5.41)} \tag{5.42a}$$

$$\text{s.t.}\quad \text{Eqs. (5.25)} - \text{(5.32)},\ \text{(5.36)} - \text{(5.40)} \tag{5.42b}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{N_\Theta}$ is the vector of problem unknowns of

$$N_\Theta = \sum_{i=1}^{L}(n^2 + 3n)(N_i + 1)/2 \tag{5.43}$$

elements, which encompasses the components of the deterministic control $\boldsymbol{\nu}_{i,\,k}$, the gain matrix $\boldsymbol{K}_{i,\,k}$ and mean value of the state $\boldsymbol{\mu}_{i,\,k-}$ before the burn at any of the maneuvering points, as

$$\boldsymbol{\Theta} = \{\boldsymbol{\mu}_{i,\,k-},\ \boldsymbol{\nu}_{i,\,k},\ \boldsymbol{K}_{i,\,k},\ \forall k = 0,\ldots,N_i,\ \forall i = 1,\ldots,L\} \tag{5.44}$$

having however the initial conditions of each i-th leg $\boldsymbol{\mu}_{i,\,0-}$ fixed to an assigned vector as in Eq. (5.27) and indirectly with Eqs. (5.25) and (5.26).

5.3 Modeling the uncertainty sources

The ROC described by Eq. (5.42) can be adapted to account for additional error sources that may arise during a trajectory, such as navigation and PEs being the primary sources of dynamical perturbation in space missions. This section introduces a simplified model for both types of errors, that would allow their inclusion in the robust design of the trajectory. It is important to acknowledge that achieving precise modeling of these errors involves a complex, mission-specific analysis. Therefore, the models presented here serve as a preliminary framework, providing a basis for more detailed investigations tailored to specific mission scenarios.

5.3.1 Navigation errors

Space navigation is critical for the planning, execution, and safety of space missions. It utilizes sophisticated techniques such as radiometric tracking, optical navigation, and onboard autonomous systems to determine the spacecraft's position and velocity [204]. These techniques calculate position and velocity as a function of measurements coming from on-board sensors or Earth-base ground station. Inter-satellite measurements could also be viable in specific missions [205]. Transforming measurements into state estimates is a complex task, which requires possibly nonlinear algorithms, such as Extended Kalman Filter (EKF) [206] or batch sequential filter [207]. Moreover, the dimensionality of the observable vectors might be less than that required for the problem, limited by the capabilities of the navigation sensors. For instance, a ground station might only provide range and range rate data, making a direct transformation of these measurements into the full state vector challenging.

When considering the inclusion of navigation errors in the design of a robust trajectory, two options might be pursued. The first idea consists in simulating a campaign of OD along a reference trajectory using the same observables expected during the actual mission, i.e. estimating the navigation measures obtained from selected ground stations with the respective measurement errors. In this way, the mean state $\boldsymbol{\mu}_{k-}$ and covariance $\boldsymbol{P}_{k-}$ before the impulsive maneuver in Eqs. (5.10) and (5.11) are substituted by covariance and mean state obtained from the OD simulation performed *a priori*. Thus, propagation of the stochastic moments as in Eq. (5.9) would not be needed anymore. With a more accurate approach, the OD campaign could be simulated within the optimization process and updated at every iteration, once all the navigation features and setup are determined as input for an implemented navigation filter in the algorithm.

As an alternative option, a linear relationship is assumed between some observables vector $\boldsymbol{y}_{k-} \in \mathbb{R}^m$ of generic dimension $m \in \mathbb{N}$ and the corresponding state $\boldsymbol{x}_{k-}$, that is

$$\boldsymbol{y}_{k-} = \boldsymbol{H}\boldsymbol{x}_{k-} + \boldsymbol{\eta} \qquad \forall k = 0, \dots, N \tag{5.45}$$

where $\boldsymbol{H} \in \mathbb{R}^{n \times m}$ is the linear mapping matrix, and $\boldsymbol{\eta} \in \mathbb{R}^m$ is a zero-mean measurement random error, which could be assumed, for instance, as a Gaussian multivariate random variable $\boldsymbol{\eta} \sim \left(\mathbf{0}_m, \tilde{\boldsymbol{R}}\right)$. Note that this approach is suitable to represent the behaviour of a system whose state estimate $\boldsymbol{y}_{k-}$ is provided by some Kalman filter with covariance of the state $\tilde{\boldsymbol{R}}$.

The CL law of Eq. (5.3) in this measurement-feedback scenario modifies as

$$\boldsymbol{u}_k = \boldsymbol{\nu}_k + \boldsymbol{K}_k \left(\boldsymbol{y}_{k-} - \mathbb{E}\left[\boldsymbol{y}_{k-}\right]\right) = \boldsymbol{\nu}_k + \boldsymbol{K}_k \left(\boldsymbol{H}\boldsymbol{x}_{k-} + \boldsymbol{\eta} - \boldsymbol{H}\boldsymbol{\mu}_{k-}\right) \qquad \forall k = 0, \dots, N \tag{5.46}$$

or simply

$$\boldsymbol{u}_k = \boldsymbol{\nu}_k + \boldsymbol{K}_k \boldsymbol{H} \left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right) + \boldsymbol{K}_k \boldsymbol{\eta}_k \qquad \forall k = 0, \dots, N \tag{5.47}$$

where $\boldsymbol{K}_k \in \mathbb{R}^{3\times m}$ is the control gain matrix, whose parameters are to be found by the optimizer.

With the measurement-feedback expression of the control, the update equation of the mean state as in Eq. (5.10) becomes

$$\boldsymbol{\mu}_{k+} = \mathbb{E}\left[\boldsymbol{y}_{k+}\right] = \boldsymbol{\mu}_{k-} + \boldsymbol{B}\boldsymbol{\nu}_k \qquad \forall k = 0, \dots, N \tag{5.48}$$

The covariance matrix of the state as in Eq. (5.11) modifies according to the following lemma.

Lemma 5.3.1. *Considering the CL control with NEs of Eq. (5.47), the covariance of $\boldsymbol{x}_{k+}$ can be formulated as*

$$\boldsymbol{P}_{k+} = \mathrm{Cov}\left[\boldsymbol{x}_{k+}\right] = \left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{H}\right)\boldsymbol{P}_{k-}\left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{H}\right)^T + \boldsymbol{B}\boldsymbol{K}_k\tilde{\boldsymbol{R}}\boldsymbol{K}_k^T\boldsymbol{B}^T \tag{5.49}$$

for all $k = 0, \dots, N$.

Proof.

$$\begin{aligned}
\boldsymbol{P}_{k+} &= \mathbb{E}\left[\left(\boldsymbol{x}_{k+} - \mathbb{E}\left[\boldsymbol{x}_{k+}\right]\right)\left(\boldsymbol{x}_{k+} - \mathbb{E}\left[\boldsymbol{x}_{k+}\right]\right)^T\right] = && \text{(5.50a)}\\
&= \mathbb{E}\left[\left(\boldsymbol{x}_{k+} - \boldsymbol{\mu}_{k+}\right)\left(\dots\right)^T\right] = && \text{(5.50b)}\\
&= \mathbb{E}\left[\left(\boldsymbol{x}_{k-} + \boldsymbol{B}\boldsymbol{u}_k - \boldsymbol{\mu}_{k-} - \boldsymbol{B}\boldsymbol{\nu}_k\right)\left(\dots\right)^T\right] = && \text{(5.50c)}\\
&= \mathbb{E}\left[\left[\boldsymbol{x}_{k-} + \boldsymbol{B}\boldsymbol{K}_k\left(\boldsymbol{y}_{k-} - \boldsymbol{H}\boldsymbol{\mu}_{k-}\right) - \boldsymbol{\mu}_{k-}\right]\left[\dots\right]^T\right] = && \text{(5.50d)}\\
&= \mathbb{E}\left[\left[\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-} + \boldsymbol{B}\boldsymbol{K}_k\left(\boldsymbol{H}\boldsymbol{x}_{k-} + \boldsymbol{\eta} - \boldsymbol{H}\boldsymbol{\mu}_{k-}\right)\right]\left[\dots\right]^T\right] = && \text{(5.50e)}\\
&= \mathbb{E}\left[\left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{H}\right)\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)\left(\dots\right)^T\left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{H}\right)^T\right.\\
&\quad + \left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{H}\right)\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)\left(\boldsymbol{B}\boldsymbol{K}_k\boldsymbol{\eta}\right)^T +\\
&\quad \left(\boldsymbol{B}\boldsymbol{K}_k\boldsymbol{\eta}\right)\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)^T\left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{H}\right)^T +\\
&\quad \left.+\boldsymbol{B}\boldsymbol{K}_k\boldsymbol{\eta}\boldsymbol{\eta}^T\boldsymbol{K}_k^T\boldsymbol{B}^T\right] = && \text{(5.50f)}
\end{aligned}$$

since there is no cross-correlation between $\boldsymbol{\eta}$ and $(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-})$,

$$\begin{aligned}
&= \mathbb{E}\left[\left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{H}\right)\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)\left(\dots\right)^T\left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{H}\right)^T + \right. && \text{(5.50g)}\\
&\left.+\boldsymbol{B}\boldsymbol{K}_k\boldsymbol{\eta}\boldsymbol{\eta}^T\boldsymbol{K}_k^T\right] && \text{(5.50h)}\\
&= \left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{H}\right)\boldsymbol{P}_{k-}\left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{H}\right)^T + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{R}\boldsymbol{K}_k^T\boldsymbol{B}^T. && \text{(5.50i)}
\end{aligned}$$

□

The mean of the control $\boldsymbol{u}_k$ as in Eq. (5.4) becomes

$$\mathbb{E}\left[\boldsymbol{u}_k\right] = \boldsymbol{\nu}_k \qquad \forall k = 0, \dots, N \tag{5.51}$$

The covariance matrix of the control $\boldsymbol{u}_k$ as in Eq. (5.6) modifies according to the following lemma.

Lemma 5.3.2. *Considering the CL control with NEs of Eq. (5.47), the covariance of $\boldsymbol{u}_k$ can be formulated by definition as*

$$\boldsymbol{P}_k^u = \mathrm{Cov}\left[\boldsymbol{u}_k\right] = \boldsymbol{K}_k \boldsymbol{H} \boldsymbol{P}_{k-} \boldsymbol{H}^T \boldsymbol{K}_k^T + \boldsymbol{K}_k \tilde{\boldsymbol{R}} \boldsymbol{K}_k^T \tag{5.52}$$

for all $k = 0, \dots, N$, being $\tilde{\boldsymbol{R}}$ the covariance matrix of $\boldsymbol{\eta}$.

Proof.

$$\begin{aligned}
\boldsymbol{P}_k^{\boldsymbol{u}} &= \mathbb{E}\left[\left(\boldsymbol{u}_k - \mathbb{E}\left[\boldsymbol{u}_k\right]\right)\left(\boldsymbol{u}_k - \mathbb{E}\left[\boldsymbol{u}_k\right]\right)^T\right] = && (5.53a)\\
&= \mathbb{E}\left[\left(\boldsymbol{u}_k - \boldsymbol{\nu}_k\right)\left(\dots\right)^T\right] = && (5.53b)\\
&= \mathbb{E}\left[\left[\boldsymbol{K}_k\left(\boldsymbol{y}_{k-} - \boldsymbol{H}\boldsymbol{\mu}_{k-}\right)\right]\left[\dots\right]^T\right] = && (5.53c)\\
&= \mathbb{E}\left[\left[\boldsymbol{K}_k\left(\boldsymbol{H}\boldsymbol{x}_{k-} + \boldsymbol{\eta} - \boldsymbol{H}\boldsymbol{\mu}_{k-}\right)\right]\left[\dots\right]^T\right] = && (5.53d)\\
&= \mathbb{E}\left[\boldsymbol{K}_k\left[\boldsymbol{H}\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right) + \boldsymbol{\eta}\right]\left[\dots\right]^T \boldsymbol{K}_k^T\right] = && (5.53e)\\
&= \mathbb{E}\Big[\boldsymbol{K}_k\left[\boldsymbol{H}\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)\right]\left[\boldsymbol{H}\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)\right]^T \boldsymbol{K}_k^T + \\
&+ \boldsymbol{K}_k\left[\boldsymbol{H}\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)\right]\boldsymbol{\eta}^T \boldsymbol{K}_k^T + \boldsymbol{K}_k\boldsymbol{\eta}\left[\boldsymbol{H}(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-})\right]^T \boldsymbol{K}_k^T + && (5.53f)\\
&+ \boldsymbol{K}_k\boldsymbol{\eta}\boldsymbol{\eta}^T \boldsymbol{K}_k^T\Big] = \\
&\text{since there is no cross-correlation between } \boldsymbol{\eta} \text{ and } (\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}),\\
&= \mathbb{E}\left[\boldsymbol{K}_k\boldsymbol{H}\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)\left(\dots\right)^T \boldsymbol{H}^T \boldsymbol{K}_k^T + \boldsymbol{K}_k\boldsymbol{\eta}\boldsymbol{\eta}^T \boldsymbol{K}_k^T\right] = && (5.53g)\\
&= \boldsymbol{K}_k\boldsymbol{H}\boldsymbol{P}_{k-}\boldsymbol{H}_k^T \boldsymbol{K}_k^T + \boldsymbol{K}_k\boldsymbol{R}\boldsymbol{K}_k^T && (5.53h)
\end{aligned}$$

□

In case a general nonlinear transformation from the state to the observables vector is instead considered as

$$\boldsymbol{y}_{k-} = \boldsymbol{h}\left(\boldsymbol{x}_{k-}\right) + \boldsymbol{\eta} \qquad \forall k = 0, \dots, N \tag{5.54}$$

with $\boldsymbol{h}\colon \mathbb{R}^n \mapsto \mathbb{R}^m$ as the nonlinear transformation, the estimate of the updated covariance state and control get sensibly complex. The CL control is now formulated as

$$\boldsymbol{u}_k = \boldsymbol{\nu}_k + \boldsymbol{K}_k\left(\boldsymbol{h}\left(\boldsymbol{x}_{k-}\right) - \overline{\boldsymbol{h}}_{k-} + \boldsymbol{\eta}\right) \qquad \forall k = 0, \dots, N \tag{5.55}$$

where $\overline{\boldsymbol{h}}_{k-} = \mathbb{E}\left[\boldsymbol{h}\left(\boldsymbol{x}_{k-}\right)\right]$. The covariance of control is given by the following lemma.

Lemma 5.3.3. *Considering the CL control with NEs of Eq. (5.55), the covariance of $\boldsymbol{u}_k$ can be formulated by definition as*

$$\boldsymbol{P}_k^{\boldsymbol{u}} = \boldsymbol{K}_k \boldsymbol{P}_{k-}^{\boldsymbol{y}} \boldsymbol{K}_k^T + \boldsymbol{K}_k \tilde{\boldsymbol{R}} \boldsymbol{K}_k^T \tag{5.56}$$

for all $k = 0, \dots, N$, being $\boldsymbol{P}^{y}_{k-}$ the covariance matrix of the observables vector $\boldsymbol{h}(\boldsymbol{x}_{k-})$, computed with an UP tool.

Proof.

$$\boldsymbol{P}^{\boldsymbol{u}}_{k} = \mathbb{E}\left[(\boldsymbol{u}_k - \mathbb{E}[\boldsymbol{u}_k])(\boldsymbol{u}_k - \mathbb{E}[\boldsymbol{u}_k])^T\right] = \tag{5.57a}$$

$$= \mathbb{E}\left[(\boldsymbol{u}_k - \boldsymbol{\nu}_k)(\dots)^T\right] = \tag{5.57b}$$

$$= \mathbb{E}\left[\left[\boldsymbol{K}_k\left(\boldsymbol{h}(\boldsymbol{x}_{k-}) - \overline{\boldsymbol{h}}_{k-} + \boldsymbol{\eta}\right)\right][\dots]^T\right] = \tag{5.57c}$$

$$= \boldsymbol{K}_k \boldsymbol{P}^{y}_{k-} \boldsymbol{K}_k^T + \boldsymbol{K}_k \tilde{\boldsymbol{R}} \boldsymbol{K}_k^T \tag{5.57d}$$

□

Note that $\mathbb{E}[\boldsymbol{u}_k] = \boldsymbol{\nu}_k + \boldsymbol{K}_k\left(\overline{\boldsymbol{h}}_{k-} - \overline{\boldsymbol{h}}_{k-}\right) = \boldsymbol{\nu}_k$. The covariance matrix of the state is obtained as follows.

Lemma 5.3.4. *Considering the CL control with NEs of Eq. (5.55), the covariance of the state can be formulated by definition as*

$$\boldsymbol{P}_{k+} = \boldsymbol{P}_{k-} + \boldsymbol{B}\boldsymbol{K}_k \boldsymbol{P}^{y}_{k-} \boldsymbol{K}_k^T \boldsymbol{B}^T + \boldsymbol{B}\boldsymbol{K}_k \tilde{\boldsymbol{R}} \boldsymbol{K}_k^T \boldsymbol{B}^T + \boldsymbol{B}\boldsymbol{K}_k \boldsymbol{M} + \boldsymbol{M}^T \boldsymbol{K}_k^T \boldsymbol{B}^T \tag{5.58}$$

for all $k = 0, \dots, N$, being $\boldsymbol{P}^{y}_{k-}$ the covariance matrix of the observables vector $\boldsymbol{h}(\boldsymbol{x}_{k-})$, computed with an UP tool, and $\boldsymbol{M} = \mathbb{E}\left[\left(\boldsymbol{h}(\boldsymbol{x}_{k-}) - \overline{\boldsymbol{h}}_{k-}\right)(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-})^T\right] \in \mathbb{R}^{m \times n}$.

Proof.

$$\boldsymbol{P}_{k+} = \mathbb{E}\left[(\boldsymbol{x}_{k+} - \mathbb{E}[\boldsymbol{x}_{k+}])(\boldsymbol{x}_{k+} - \mathbb{E}[\boldsymbol{x}_{k+}])^T\right] = \tag{5.59a}$$

$$= \mathbb{E}\left[(\boldsymbol{x}_{k+} - \boldsymbol{\mu}_{k+})(\dots)^T\right] = \tag{5.59b}$$

$$= \mathbb{E}\left[\left[(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}) + \boldsymbol{B}\boldsymbol{K}_k\left(\boldsymbol{h}(\boldsymbol{x}_{k-}) - \overline{\boldsymbol{h}}_{k-}\right) + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{\eta}\right][\dots]^T\right] = \tag{5.59c}$$

$$= \boldsymbol{P}_{k-} + \boldsymbol{B}\boldsymbol{K}_k \boldsymbol{P}^{y}_{k-} \boldsymbol{K}_k^T \boldsymbol{B}^T + \boldsymbol{B}\boldsymbol{K}_k \tilde{\boldsymbol{R}} \boldsymbol{K}_k^T \boldsymbol{B}^T + \tag{5.59d}$$

$$+ \boldsymbol{B}\boldsymbol{K}_k \mathbb{E}\left[\left(\boldsymbol{h}(\boldsymbol{x}_{k-}) - \overline{\boldsymbol{h}}_{k-}\right)(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-})^T\right] +$$

$$+ \mathbb{E}\left[(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-})\left(\boldsymbol{h}(\boldsymbol{x}_{k-}) - \overline{\boldsymbol{h}}_{k-}\right)^T\right] \boldsymbol{K}_k^T \boldsymbol{B}^T = \tag{5.59e}$$

$$= \boldsymbol{P}_{k-} + \boldsymbol{B}\boldsymbol{K}_k \boldsymbol{P}^{y}_{k-} \boldsymbol{K}_k^T \boldsymbol{B}^T + \boldsymbol{B}\boldsymbol{K}_k \tilde{\boldsymbol{R}} \boldsymbol{K}_k^T \boldsymbol{B}^T + \boldsymbol{B}\boldsymbol{K}_k \boldsymbol{M} + \boldsymbol{M}^T \boldsymbol{K}_k^T \boldsymbol{B}^T \tag{5.59f}$$

□

The elements of the matrix $\boldsymbol{M}$ can be computed following the definition of

covariance, as

$$M_{ij} = \mathbb{E}\left[h_i\left(\boldsymbol{x}_{k-}\right) x_{j,\,k-}\right] - \mathbb{E}\left[h_i\left(\boldsymbol{x}_{k-}\right)\right]\mathbb{E}\left[x_{j,\,k-}\right] \quad \forall i = 1, \ldots, m \qquad \forall j = 1, \ldots, n \tag{5.60}$$

where naturally $h_i\left(\boldsymbol{x}_{k-}\right)$ is the i-th element of $\boldsymbol{h}\left(\boldsymbol{x}_{k-}\right)$ and $x_{j,\,k-}$ is the j-th element of $\boldsymbol{x}_{k-}$. It is again possible to obtain these quantities by means of a proper UP tool. Considering the UT for instance, the quantity $\mathbb{E}\left[h_i\left(\boldsymbol{x}_{k-}\right) x_{j,\,k-}\right]$ could be estimated by averaging the sum between the cross products between the two sets of SPs of the propagated state and observables vector, as

$$\mathbb{E}\left[h_i\left(\boldsymbol{x}_{k-}\right) x_{j,\,k-}\right] = \frac{1}{(2n+1)(2m+1)} \sum_{p=0}^{2n} \sum_{q=0}^{2m} \left(h_i\left(\boldsymbol{x}_{k-}\right)\right)_p \left(x_{j,\,k-}\right)_q \tag{5.61}$$

where the subscripts p and q denote the affiliation to the p-th or q-th SP of the propagated state set and observables vector set respectively.

5.3.2 Propulsive errors

In space missions, inaccuracies in both the magnitude and direction of thrust delivered by the spacecraft's propulsion system are a critical source of uncertainty that can significantly affect trajectory design and mission success. Variability in the performance of thrusters, environmental influences, and system malfunctions contribute to these inaccuracies. For instance, chemical propulsion systems may suffer from fluctuations in fuel pressure, while electric propulsion systems can be affected by variations in power levels.

Similarly to the previous section, a simplified model to analytically characterize the uncertainty contribution of PEs is provided, considering exclusively errors on magnitude and direction of the deterministic component $\boldsymbol{\nu}_k$ of the control. Errors related to MTEs or major thrust failures (i.e., Δv less than 1/10 of its nominal magnitude) are not taken into account in the following analysis.

Considering the CL control law in Eq. (5.3), the deterministic component $\boldsymbol{\nu}_k$ is allowed to vary in magnitude and direction. Starting from the nominal vector, a deviation by an angle $\delta\theta$ in every possible direction results in a locus of the vector apex obtained as the intersection between a cone, whose apex is the point of application of $\boldsymbol{\nu}_k$, aperture $2\delta\theta$, and the vertical axis coincides with $\boldsymbol{\nu}_k$, and a sphere centered in the point of application of $\boldsymbol{\nu}_k$ and radius ν_k. When considering also deviation in magnitude of the nominal vector in a $\pm\delta\nu$ range, the locus of the thrust vector apex is obtained as the intersection between the same cone and a spherical shell of thickness $2\delta\nu$ centered in the point of application of $\boldsymbol{\nu}_k$ with maximum and minimum radius $\nu_k + \delta\nu$ and $\nu_k - \delta\nu$, respectively. Figure 5.1 shows the region (in grey) of the possible vertices of $\boldsymbol{\nu}_k$ given the deviations $\delta\theta$ and $\delta\nu$. Note that, when considering an unbounded distribution, the area in grey is to be considered as the contour region corresponding to the 3σ contour plot. In 3D, a further deviation angle $\delta\phi \in [0,\, \pi)$ is needed to identify the direction of the component perpendicular to $\boldsymbol{\nu}_k$. Having thus the deviations $\delta\theta$, $\delta\phi$, and $\delta\nu$, which might be sampled from selected probability distributions, such as

$$\delta\phi \sim \mathcal{U}_{[0,\,\pi)} \tag{5.62}$$

$$\delta\theta \sim \mathcal{N}\left(0,\, \sigma_\theta\right) \tag{5.63}$$

$$\delta\nu \sim \mathcal{N}\left(0,\, \sigma_\nu\right) \tag{5.64}$$

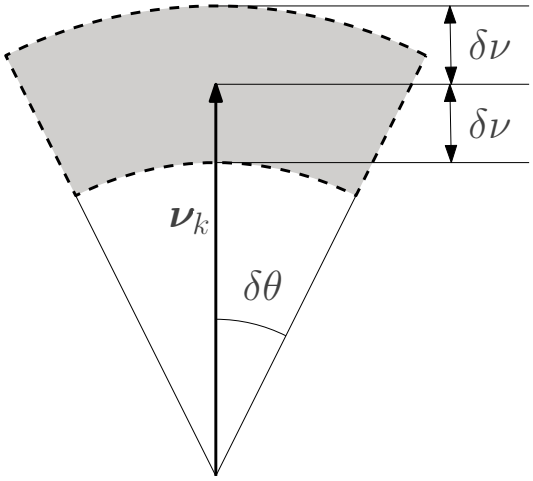


Figure 5.1. PE 2D scheme.

where the deviation standards derives from the engine technical specifications, the actuated control $\boldsymbol{\nu}'_k$ (accounting for PEs) is

$$\boldsymbol{\nu}'_k = (1 + \delta\nu)\, \boldsymbol{\mathcal{R}}\,(\delta\theta,\, \delta\phi)\, \boldsymbol{\nu}_k \qquad \forall k = 0, \dots, N \tag{5.65}$$

where $\boldsymbol{\mathcal{R}}\,(\cdot,\, \cdot)$ is a rotation matrix made by two pure rotation of angles $\delta\theta$ and $\delta\phi$. Considering a generic coordinate system with the x axis perpendicular to $\boldsymbol{\nu}_k$, the y axis coincident with $\boldsymbol{\nu}_k$, and the z axis completing a right-hand system, the rotation matrix $\boldsymbol{\mathcal{R}}\,(\delta\theta,\, \delta\phi)$ is obtained by a first rotation about x by an angle $\delta\theta$ and a second rotation about y by an angle $\delta\phi$, as

$$\boldsymbol{\mathcal{R}}\,(\delta\theta,\, \delta\phi) = \begin{bmatrix} \cos\delta\phi & 0 & \sin\delta\phi \\ 0 & 1 & 0 \\ -\sin\delta\phi & 0 & \cos\delta\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\delta\theta & -\sin\delta\theta \\ 0 & \sin\delta\theta & \cos\delta\theta \end{bmatrix} = \tag{5.66}$$

$$= \begin{bmatrix} \cos\delta\phi & \sin\delta\phi\sin\delta\theta & \sin\delta\phi\cos\delta\theta \\ 0 & \cos\delta\theta & -\sin\delta\theta \\ -\sin\delta\phi & \cos\delta\phi\sin\delta\theta & \cos\delta\phi\cos\delta\theta \end{bmatrix} \tag{5.67}$$

While accurate, this model is impractical for the ROCP, due to its nonlinearity. In particular, the presence of the rotation matrix $\boldsymbol{\mathcal{R}}\,(\delta\theta,\, \delta\phi)$ results in an extremely difficult analytical derivation of the update law for the covariance matrix of the state and control as in Eqs. (5.11) and (5.6), respectively. In addition, note that the deviation is imparted on the deterministic component of the impulsive control only, rather than on the total control $\boldsymbol{u}_k$, for simplification purposes. However, under the small displacements assumption, the errors on magnitude and directions can be considered independently and then superposed, leading to a further simplified mode, that will allow an analytical derivation of the update equation for the covariance matrices.

The magnitude deviation $(1 + \delta u_\mathrm{n})$, $\delta u_\mathrm{n} \sim \mathcal{N}\,(0,\, \sigma_\mathrm{n})$, $\sigma_\mathrm{n} \in \mathbb{R}$, as in Eq. (5.65) is still considered, to which a Gaussian deviation vector $\delta\boldsymbol{u}_\mathrm{d} \sim \mathcal{N}\,(0,\, \sigma_\mathrm{d}\boldsymbol{I}_3)$, $\sigma_\mathrm{d} \in \mathbb{R}$, is simply added, as

$$\boldsymbol{u}'_k = (1 + \delta u_\mathrm{n})\, \boldsymbol{u}_k + \delta\boldsymbol{u}_\mathrm{d} = \tag{5.68}$$

$$= \boldsymbol{u}_k + \delta u_\mathrm{n}\boldsymbol{\nu}_k + \delta u \boldsymbol{K}_k\, (\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}) + \delta\boldsymbol{u}_\mathrm{d} = \tag{5.69}$$

$$= \boldsymbol{\nu}_k + \delta u_\mathrm{n}\boldsymbol{\nu}_k + (1 + \delta u)\, \boldsymbol{K}_k\, (\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}) + \delta\boldsymbol{u}_\mathrm{d} \qquad \forall k = 0, \dots, N \tag{5.70}$$

Figure 5.2 presents the 2D area (in grey) of the possible vertices of $\boldsymbol{u}_k$ given the deviations $\delta\boldsymbol{u}_\mathrm{d}$ and δu_n is plotted. Also in this case, the area is intended to be a reference shape of the distribution.

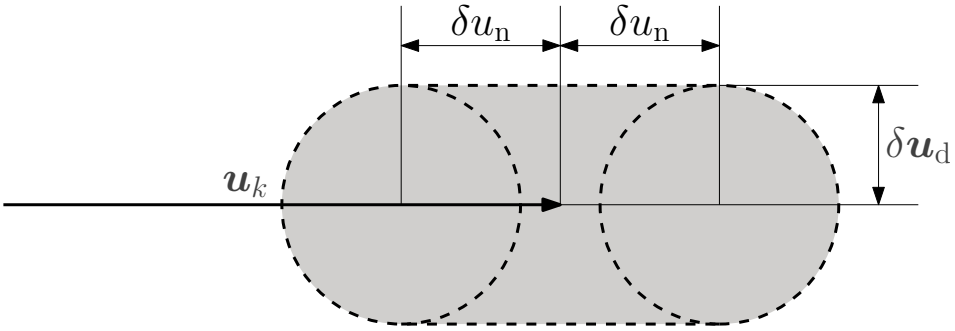


Figure 5.2. Propulsion simplified error 2D scheme.

The updated state is evaluated as

$$\begin{aligned}
\boldsymbol{x}_{k+} &= \boldsymbol{x}_{k-} + \boldsymbol{u}'_k = \boldsymbol{x}_{k-} + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{B}\delta u_\mathrm{n}\boldsymbol{u}_k + \boldsymbol{B}\delta\boldsymbol{u}_\mathrm{d} = && (5.71)\\
&= \boldsymbol{x}_{k-} + (1+\delta u_\mathrm{n})\,\boldsymbol{B}\boldsymbol{\nu}_k + (1+\delta u_\mathrm{n})\,\boldsymbol{B}\boldsymbol{K}_k\,(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}) + && (5.72)\\
&+ \boldsymbol{B}\delta\boldsymbol{u}_\mathrm{d} \qquad \forall k = 0,\dots,N && (5.73)
\end{aligned}$$

The mean value of the control and the state of respectively Eq. (5.70) and (5.73) are

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{u}'_k\right] &= \boldsymbol{\nu}_k && (5.74)\\
\mathbb{E}\left[\boldsymbol{x}_{k+}\right] &= \boldsymbol{\mu}_{k+} = \boldsymbol{\mu}_{k-} + \boldsymbol{B}\boldsymbol{\nu}_k && (5.75)
\end{aligned}$$

having assumed no cross-correlation between δu_n and $(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-})$ and that the deviations δu_n and $\boldsymbol{u}_\mathrm{d}$ are zero-mean random variables.

The update law of the control covariance matrix is thus given by the following lemma

Lemma 5.3.5. *Considering the CL control with PEs of Eq. (5.70), the covariance of the control $\boldsymbol{u}'_k$ can be formulated by definition as*

$$\boldsymbol{P}^{\boldsymbol{u}}_k = \left(1+\sigma^2_\mathrm{n}\right)\boldsymbol{K}_k\boldsymbol{P}_{k-}\boldsymbol{K}^T_k + \sigma^2_\mathrm{n}\boldsymbol{\nu}_k\boldsymbol{\nu}^T_k + \sigma_\mathrm{d}\boldsymbol{I}_3 \qquad (5.76)$$

for all $k = 0,\dots,N$.

Proof.

$$\begin{aligned}
\boldsymbol{P}^{\boldsymbol{u}}_k &= \mathbb{E}\left[\left(\boldsymbol{u}'_k - \mathbb{E}\left[\boldsymbol{u}'_k\right]\right)\left(\boldsymbol{u}'_k - \mathbb{E}\left[\boldsymbol{u}'_k\right]\right)^T\right] = && (5.77a)\\
&= \mathbb{E}\left[\left[(1+\delta u_\mathrm{n})\,\boldsymbol{K}_k\,(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}) + \delta u_\mathrm{n}\boldsymbol{\nu}_k + \delta\boldsymbol{u}_\mathrm{d}\right][\dots]^T\right] = && (5.77b)\\
&= \left(1+\sigma^2_\mathrm{n}\right)\boldsymbol{K}_k\boldsymbol{P}_{k-}\boldsymbol{K}^T_k + \sigma^2_\mathrm{n}\boldsymbol{\nu}_k\boldsymbol{\nu}^T_k + \sigma_\mathrm{d}\boldsymbol{I}_3 && (5.77c)
\end{aligned}$$

as there is no cross-correlation between δu_n, $\delta\boldsymbol{u}_\mathrm{d}$, and $(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-})$. □

The update law of the state covariance matrix is instead obtained as follows.

Lemma 5.3.6. *Considering the CL control with PEs of Eq. (5.70), the covariance of the state can be formulated by definition as*

$$\boldsymbol{P}_{k+} = \left(1+\sigma_{\mathrm{n}}^2\right)\left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{H}\right)\boldsymbol{P}_{k-}\left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{H}\right)^T + \sigma_{\mathrm{n}}^2\boldsymbol{B}\boldsymbol{\nu}_k\boldsymbol{\nu}_k^T\boldsymbol{B}^T + \\ + \sigma_{\mathrm{d}}^2\boldsymbol{B}\boldsymbol{B}^T \tag{5.78}$$

for all $k = 0, \ldots, N$.

Proof.

$$\boldsymbol{P}_{k+} = \mathbb{E}\left[\left(\boldsymbol{x}_{k+} - \boldsymbol{\mu}_{k+}\right)\left(\boldsymbol{x}_{k+} - \boldsymbol{\mu}_{k+}\right)^T\right] = \tag{5.79a}$$

$$= \mathbb{E}\left[\left[\left(1+\delta u_{\mathrm{n}}\right)\left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\right)\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right) + \delta u_{\mathrm{n}}\boldsymbol{B}\boldsymbol{\nu}_k + \boldsymbol{B}\delta\boldsymbol{u}_{\mathrm{d}}\right]\left[\ldots\right]^T\right] = \tag{5.79b}$$

$$= \left(1+\sigma_{\mathrm{n}}^2\right)\left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{H}\right)\boldsymbol{P}_{k-}\left(\boldsymbol{I}_n + \boldsymbol{B}\boldsymbol{K}_k\boldsymbol{H}\right)^T + \sigma_{\mathrm{n}}^2\boldsymbol{B}\boldsymbol{\nu}_k\boldsymbol{\nu}_k^T\boldsymbol{B}^T + \\ + \sigma_{\mathrm{d}}^2\boldsymbol{B}\boldsymbol{B}^T \tag{5.79c}$$

as there is no cross-correlation between δu_{n}, $\delta\boldsymbol{u}_{\mathrm{d}}$, and $\left(\boldsymbol{x}_{k-} - \boldsymbol{\mu}_{k-}\right)$. □

5.4 Case study A: DESTINY+ interplanetary phase

In this section, the proposed methodology is applied to compute the robust trajectory of an actual space mission, that is, a portion of the interplanetary flight of the DESTINY+ probe. DESTINY+ is an upcoming JAXA exploration mission aimed to observe, by means of a close passage, the Geminids meteor shower parent body (3200) Phaethon. In addition to the meaningful scientific objectives, DESTINY+ will serve also as a demonstrator of several unprecedented technologies, including lightweight solar array panels, advanced asteroid flyby observation instruments, and a highly efficient ion engine system. This latter low-thrust engine will be used in particular to escape from Earth's vicinity into deep space after numerous revolutions of the Earth. The mission is in fact divided into three phases: the Spiral Orbit-Raising (SOR) phase, where a low-thrust spiraling trajectory will raise the altitude of the spacecraft up to the Moon's orbit, the Moon Flyby (MFB) phase, in which sequential Moon gravity assists will provide the spacecraft with the required v_∞ for the subsequent final InterPlanetary Transfer (IPT) phase, consisting in flybys of multiple asteroids utilizing low-thrust maneuvers and Earth gravity assist maneuvers. Similarly to this mission, the conceptual mission of sequential rendezvous of asteroids was also recently proposed by JAXA [208].

5.4.1 Problem data

The nominal Earth–Phaethon–Earth transfer of the IPT phase is presented in Fig. 5.3, highlighting the three crucial events: departure from Earth, close passage to Phaethon, arrival to Earth. The first leg Earth–Phaethon is in yellow, while Phaethon–Earth is in green. Considering a departure epoch around December 2026, DESTINY+ is expected to perform the first leg with ToF of approximately 1 year up to the encounter with Phaethon, whose ascending node is located in the

proximity of the Earth's orbit, making it the perfect candidate node for a flyby, since the other descending node is instead placed inside Mercury's orbit. After having intercepted Phaeton's passage in the ecliptic plane, DESTINY+ will then return in Earth proximity in approximately 5 months. After having accomplished this main scientific objective, the mission might continue with additional asteroid observations, like 2005 UD, another asteroid of the Apollo group. The extended trajectory of the mission however is not treated in this work.

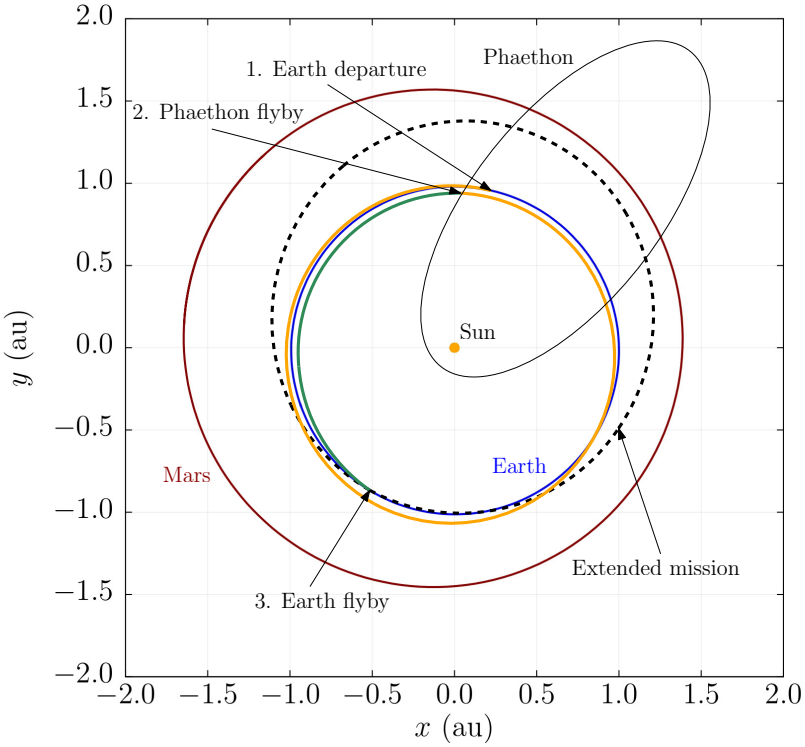


Figure 5.3. Baseline interplanetary trajectory of DESTINY+ in the Sun-centered ECLIPJ2000 inertial frame.

In the current analysis, a three-dimensional transfer is considered, and the spacecraft state vector is given by position and inertial velocity in the Sun-centered inertial RF J2000. The dynamical model used for this analysis is a heliocentric Keplerian motion, whose EoM are simply

$$\mathrm{d}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{v}(t) \\ -\mu \boldsymbol{r}(t)/r^3 \end{bmatrix} \mathrm{d}t \qquad \forall t \in (t_{k-1},\, t_k) \qquad \forall k = 1, \ldots, N \tag{5.80}$$

Given the close encounter with Phaethon during the trajectory, this mission requires the spacecraft to transit at a specific epoch at the ascending node of Phaethon's orbit. This mission phase is thus modeled as a multiple-leg trajectory with $L = 2$. The first leg consists in the flight from the Earth to Phaethon, while the second leg is from Phaethon to the Earth. In addition, to ensure the flyby, a constraint is enforced on the spacecraft position at the end of the first leg, so that $\tilde{\boldsymbol{r}}_{1,\,N_1+} = \boldsymbol{r}_{1,\,N_1+}$ is the position of Phaethon at the flyby epoch $t_{\mathrm{fb}} = \Delta T_1$, having

$$\Delta T_i = \sum_{k=1}^{N_i} \Delta t_k \qquad \forall i = 1, \ldots, L \tag{5.81}$$

A constraint on the mean velocity at the flyby is not considered in this preliminary analysis but could be easily added to the computation.

For this test case, NEs with linear mapping relation as in Eq. (5.45) are considered. Null PEs are instead included. The initial assigned covariance matrix $\widetilde{\boldsymbol{P}}_{1,0-}$ and the covariance matrix $\widetilde{\boldsymbol{R}}$ of the additive Gaussian noise $\boldsymbol{\eta}$ in (5.47) are defined as

$$\widetilde{\boldsymbol{P}}_{1,0-} = \operatorname{diag}\left[\left(\sigma^r_{1,0-}\right)^2 \boldsymbol{I}_3,\ \left(\sigma^v_{1,0-}\right)^2 \boldsymbol{I}_3\right] \tag{5.82}$$

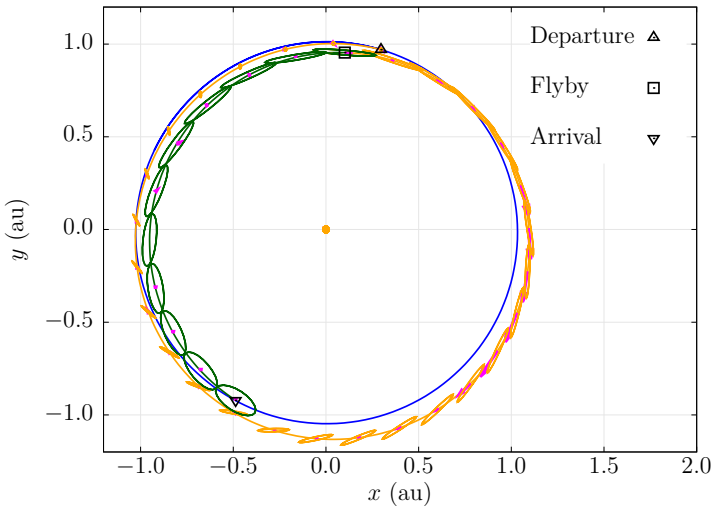
$$\widetilde{\boldsymbol{R}} = \operatorname{diag}\left[\left(\sigma^r_R\right)^2 \boldsymbol{I}_3,\ \left(\sigma^v_R\right)^2 \boldsymbol{I}_3\right] \tag{5.83}$$

where $\sigma^r_{1,0-}$ and $\sigma^v_{1,0-}$ are the assigned initial standard deviations for position and velocity respectively, and σ^r_R and σ^v_R are the fixed standard deviation for the NE on position and velocity respectively. All problem data are summarized in Table 5.1. The numbers of arcs N_1 and N_2 were chosen to be roughly proportional to the ToF ΔT_1 and ΔT_2 and to ensure an adequate number of maneuvers for each leg.

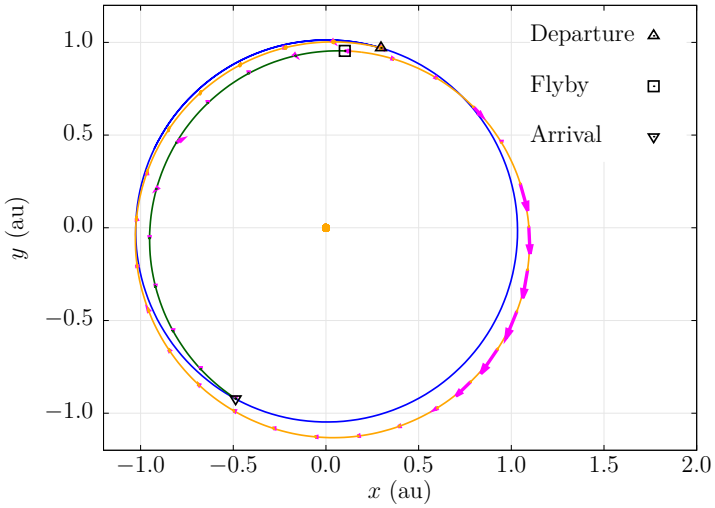
All numerical computations were made non-dimensional by scaling lengths, velocities, and times with respect to $r^{\text{conv}} = \|\widetilde{\boldsymbol{r}}_{1,0-}\|$, $v^{\text{conv}} = \|\widetilde{\boldsymbol{v}}_{1,0-}\|$, and $t^{\text{conv}} = r^{\text{conv}}/v^{\text{conv}}$. The large-scale sparse nonlinear optimizer WORHP [202] was used for solving the optimization problem.

Table 5.1. Problem data.

Variable	Value	Unit
L	2	-
N_1	28	-
N_2	10	-
ΔT_1	394.451	days
ΔT_2	140.191	days
$\widetilde{u}_{\max}$	105	m/s
β	5%	-
$\mu_\odot$	132712440018	km^3/s^2
$\widetilde{\boldsymbol{r}}_{1,0-}$	$[43148032,\ 140976675,\ -8649]^T$	km
$\widetilde{\boldsymbol{v}}_{1,0-}$	$[-29.804,\ 7.537,\ 0.641]^T$	km/s
$\widetilde{\boldsymbol{r}}_{1,N_1+}$	$[14739776,\ 138648434,\ 1279019,]^T$	km
$\widetilde{\boldsymbol{r}}_{2,N_2+}$	$[-70786621,\ -133919742,\ 8765]^T$	km
$\widetilde{\boldsymbol{v}}_{2,N_2+}$	$[24.531,\ -16.081,\ -0.647]^T$	km/s
$\sigma^r_{1,0-}$	4.662×10^5	km
$\sigma^v_{1,0-}$	30.003	m/s
σ^r_{1,N_1+}	3.297×10^4	km
σ^v_{1,N_1+}	21.215	m/s
σ^r_{2,N_2+}	1.043×10^5	km
σ^v_{2,N_2+}	6.709	m/s
σ^r_w	4.662×10^5	km
σ^v_w	30.003	m/s
σ^r_R	8.075×10^4	km
σ^v_R	5.197	m/s



(a) OCP solution.



(b) CL solution.

Figure 5.4. Nominal spacecraft trajectories with 95%-confidence position ellipses.

5.4.2 Numerical results

Figure 5.4 presents the DESTINY+ IPT trajectory. The Δvs and the covariance ellipsoids on position for the OCP solution (Fig. 5.4a) and the CL (Fig. 5.4b) solution in the x-y plane are also presented. The OCP solution is computed starting from a null state dispersion, while forcing the gain matrices of the CL control to zero. The constraints on the final covariance of the state from the optimization problem of Eq. (5.42) are instead removed. Once the OCP solution is obtained, the state uncertainty along the trajectory is estimated applying UT separately to analyze how

the initial state distribution propagates. It is apparent that the CL control is able to reduce the estimated (by UT) dispersion of the state along the trajectory and most prominently at the arrival point. The main feature of both OCP and CL solutions

Table 5.2. Comparison between the solution of the ROCP and the related MC analysis in the OCP and CL cases.

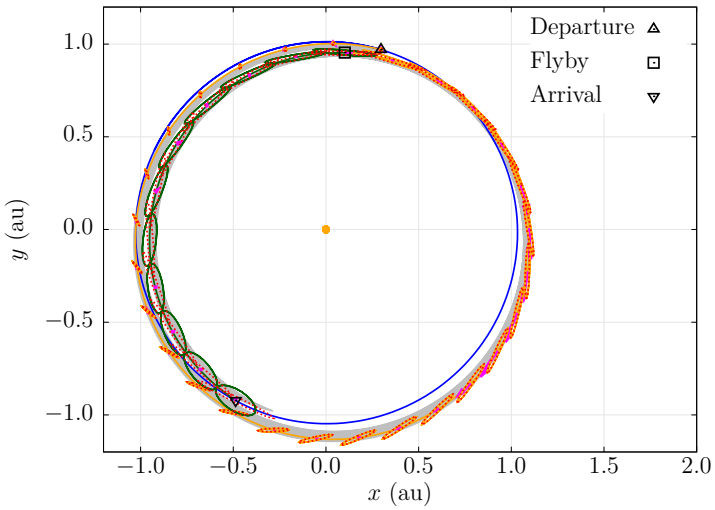
Quantity	OCP	OCP (MC)	ROCP-CL	ROCP-CL (MC)	Unit
Δv_{tot}	0.695	0.695	1.405	1.406	km/s
Δv_{d}	0.695	0.695	0.695	0.695	km/s
Δv_{s}	-	-	0.710	0.711	km/s
$\Delta r^{x}_{2,\,N_2+}$	-	2.488×10^{5}	-	56.136	km
$\Delta r^{y}_{2,\,N_2+}$	-	5.655×10^{5}	-	77.411	km
$\Delta v^{x}_{2,\,N_2+}$	-	0.102	-	6.763×10^{-7}	km/s
$\Delta v^{y}_{2,\,N_2+}$	-	6.213×10^{-2}	-	1.077×10^{-5}	km/s
$\sigma^{x}_{2,\,N_2+}$	6.416×10^{6}	1.203×10^{7}	6.948×10^{4}	6.958×10^{4}	km
$\sigma^{y}_{2,\,N_2+}$	4.884×10^{6}	5.984×10^{6}	7.348×10^{4}	7.344×10^{4}	km
$\sigma^{v_x}_{2,\,N_2+}$	0.826	1.459	6.253×10^{-3}	6.249×10^{-3}	km/s
$\sigma^{v_y}_{2,\,N_2+}$	1.082	2.263	6.096×10^{-3}	6.103×10^{-3}	km/s

are also summarized in Table 5.2, which presents the deterministic (Δv_{d}), stochastic (Δv_{s}) and total (Δv_{tot}) velocity increment, as well as the mean terminal errors on the in-plane components of the final position ($\Delta r^{x}_{2,\,N_2+}$ and $\Delta r^{y}_{2,\,N_2+}$) and velocity ($\Delta v^{x}_{2,\,N_2+}$ and $\Delta v^{y}_{2,\,N_2+}$). Note that the stochastic Δv is a function of the violation rate β. Also, the square root of the diagonal terms of the final covariance matrix ($\sigma^{x}_{2,\,N_2+}$, σ^{y}_{2,N_2+}, $\sigma^{v_x}_{2,\,N_2+}$, and $\sigma^{v_y}_{2,\,N_2+}$) related to the in-plane position and velocity components are reported.

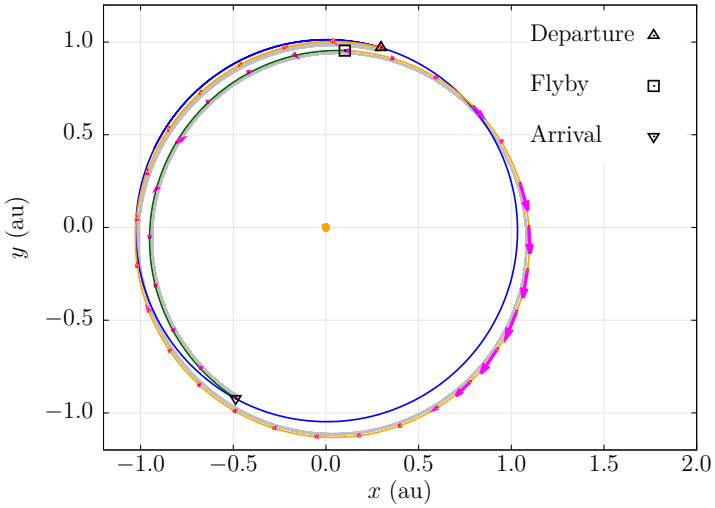
The additional robustness layer provided by the CL control is obtained at the cost of about twice the baseline trajectory fuel, while the total deterministic Δv is roughly the same, considering a 95% confidence level and the perturbation magnitude reported in Table 5.1. Note that the imposed maximum state variances at the asteroid rendezvous and Earth arrival are relatively severe, and lower extra fuel expenditures are possible if larger dispersions are allowed.

An MC analysis of 100 000 independent runs was carried out to verify the accuracy of the estimated probability distribution at each step for both the CL and OCP cases. Starting from a random initial state $\boldsymbol{\mu}_{1,\,0-} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_{1,\,0-},\, \tilde{\boldsymbol{P}}_{1,\,0-})$, each trajectory is propagated towards the final state using the feedback control law in Eq. (5.47). The propagation of the SDEs is accomplished by means of the stochastic fourth-order Runge-Kutta method discussed in the work of N. J. Kasdin [85]. Numerical results are reported in Table 5.2 to allow for a comparison between the MC and ROCP solutions. The stochastic Δv (Δv_{s}) is evaluated through the second term of Eq. (5.22), having the covariance matrix of the control derived from the MC runs. It is possible to observe how state and dynamical errors affect the state uncertainty when the deterministic OCP control is used, and how these uncertainties are conversely mitigated in presence of the CL control law. Note that all MC values for the CL case are in good agreement with the values estimated in the ROCP with the UT. The large errors of the mean terminal state for the OCP MC are due to the inaccuracy of UT in estimating the mean state when propagating wide uncertainties.

The MC campaign of the trajectories in Fig. 5.4 are presented in Fig. 5.5, showing



(a) OCP solution with MC position ellipses (dashed red line) and UT position ellipses (solid line of the leg colour).



(b) CL solution.

Figure 5.5. MC trajectories (in gray).

good agreement, in the CL case, with the covariance ellipsoids found by solving the ROCP in Eqs. (5.42). The position ellipses estimated in the MC are also shown in red dotted line. In Fig. 5.6, the CL MC analysis is plotted together with both the UT and MC ellipses up-scaled by a factor 5.

The covariance matrices $\boldsymbol{P}_{i,\,k+}$ empirically computed in the MC analysis are then compared to those obtained during the ROCP that uses UT for propagating the state uncertainty. The diagonal terms of either matrix are compared in Figs. 5.7– 5.9

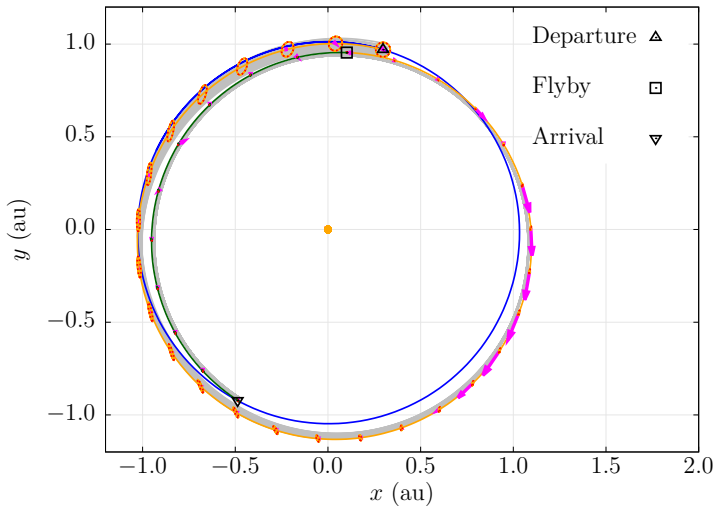
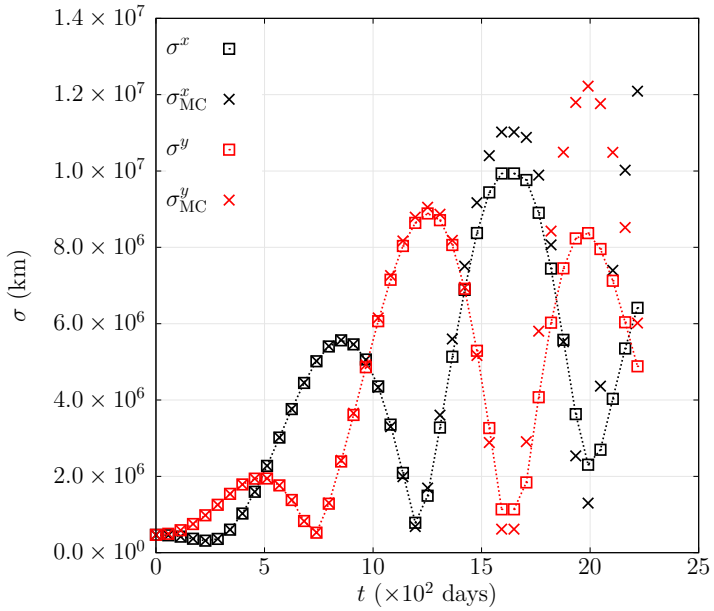


Figure 5.6. MC trajectories (in gray) with corresponding position ellipses (dashed red line) and UT position ellipses (solid line of the leg colour). Ellipses are up-scaled by a factor 5.

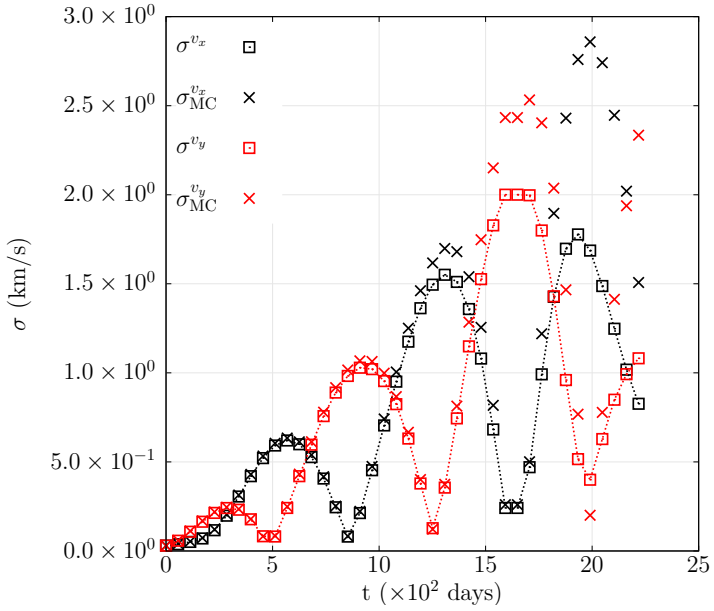
for the OCP and CL solutions. According to the attained results, the UT succeeds in propagating the state distribution very accurately in the CL case, where the displacement from the nominal trajectory is kept very small and thus the state distribution is well approximated by a Gaussian one. On the other hand, in the OCP case, the MC trajectories may significantly deviate from the nominal solution, especially at the end of the transfer, and the estimates of covariance matrix are less accurate. This phenomenon likely occurs because, in the absence of a control term to contain state dispersion at each maneuver, the probability distribution progressively widens and deviates from a Gaussian shape. The UT's reliance on Gaussian assumptions becomes ineffective as the distribution stretches and distorts, underscoring the need for a control strategy that consistently curbs dispersion and maintains the distribution within a manageable range for precise propagation.

In Figure 5.10, the thrust profiles of both the CL and OCP cases are presented alongside the MC runs for the CL solution and the magnitude threshold $u_{\max}$ (the red dashed line). Note that the commanded values are reported, resulting from Eq. (5.35), not the actuated one, where the maximum control magnitude is saturated by $u_{\max}$. In this plot, the rare violation of the control magnitude with a maximum rate of β is visible. A significantly increase in thrust is also visible in the first part of the trajectory for the robust case, unlike the OCP where small to null thrust is instead performed before the midway peak. In this first part, the robust control law impart a small amount of thrust to compensate for the reduced nominal thrust $\boldsymbol{\nu}$ at the peak. In case the nominal thrust would be kept close to the maximum threshold as in the OCP case, there would not be margin for correction maneuvers. Additionally, this initial thrust is effective also to reduce the state dispersion to meet the deviation constraints at the asteroid encounter.

A parametric analysis has been carried out in the previous work to investigate the behavior of the solution for increasing values of the process noise, that is, σ_w^r and σ_w^v. The stochastic component of control becomes progressively larger as the



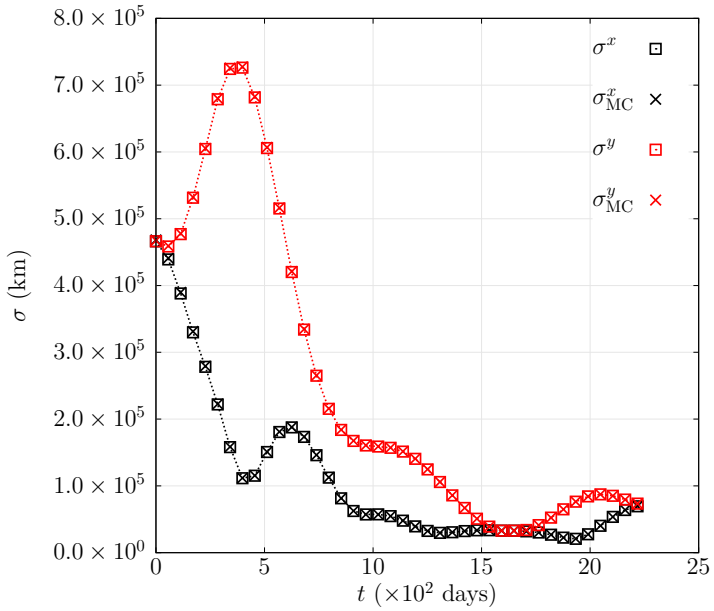
(a) OCP, in-plane position.



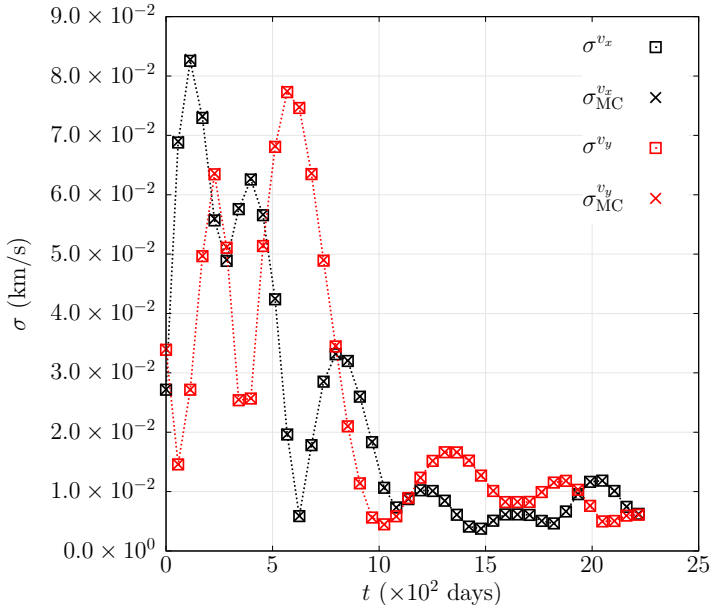
(b) OCP, in-plane velocity.

Figure 5.7. Covariance matrices trace comparison of the OCP case.

magnitude of the perturbation increases. An effect of the disturbance term is also visible on the deterministic component Δv_{d}, which increases with $\overline{\boldsymbol{Q}}$, as already discussed. In this work, a similar analysis is performed for increasing values of NE, that is, σ_R^r and σ_R^v. The results in this case are similar to the previous process noise analysis, as reported in Table 5.3, where the "Medium" case is the test case previously presented (data in Table 5.1), and the other cases are obtained scaling σ_R^r and σ_R^v according to the parameter in brackets. The maximum eigenvalues of the final in-plane position and velocity, that is related the covariance matrix $\boldsymbol{P}_{2,\,N_2+}$,



(a) CL, in-plane position.

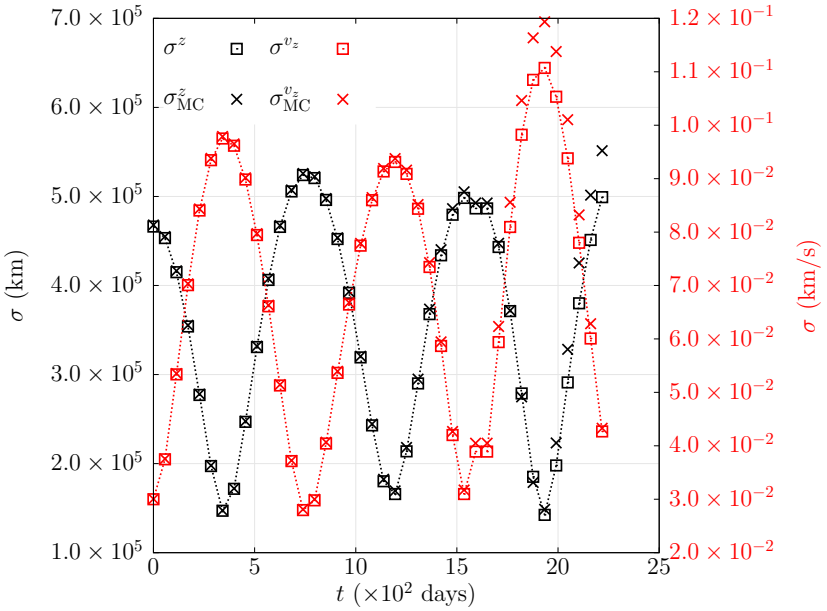


(b) CL, in-plane velocity.

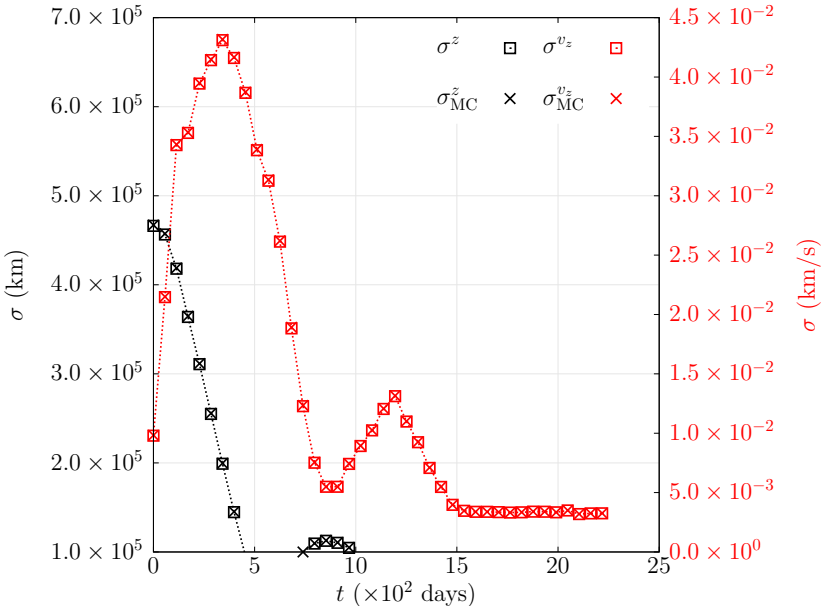
Figure 5.8. Covariance matrices trace comparison of the CL case.

are denoted as $\lambda^{xy}_{\max} = \lambda^{xy}_{\max}(\boldsymbol{P}_{2,N_2+})$ and $\lambda^{v_xv_y}_{\max} = \lambda^{v_xv_y}_{\max}(\boldsymbol{P}_{2,N_2+})$. The cumulative stochastic control Δv_{s} is the most affected component, given a significant raise for higher NE, while only a marginal increment is reported for the deterministic control Δv_{d}. The maximum eigenvalue of the in-plane velocity $\lambda^{v_xv_y}_{\max}$ appears to be a limiting parameter of the solution, as the values in the four cases are close to the maximum allowed one of 6.709 m/s, while the maximum eigenvalue of the in-plane position $\lambda^{xy}_{\max}$ remains largely below the maximum allowed value of 1.043×10^5 km.

In order to asses the enhancement of the robust solution in which NE are included,



(a) OCP, out-of-plane state.



(b) CL, out-of-plane state.

Figure 5.9. Covariance matrices trace comparison of the CL case.

a comparison between the performance of two solutions of the ROCP computed with and without NE are performed. In particular, the first one is the problem formulated with data in Table 5.1, while the second one has the same process noise but null NE. From these two solutions, two MC analyses are then performed considering for both the process noise and the NE of the first solution. The numerical comparison is reported in Table 5.4. It is evident the worse performance of the control law computed without including NE, as the final state dispersion is much larger in the second MC, especially for velocity. The mean terminal errors on mean position and

Table 5.3. Comparison of the solutions for the different values of NE.

Quantity	NEs				Unit
	None (0×)	Low (1×)	Medium (3×)	High (5×)	
Δv_{d}	0.695	0.696	0.695	0.695	km/s
Δv_{s}	0.584	0.622	0.710	0.736	km/s
Δv_{tot}	1.279	1.317	1.405	1.431	km/s
$\sqrt{\lambda_{\max}^{xy}}$	4.510×10^4	5.818×10^4	9.454×10^4	1.041×10^5	km
$\sigma^{x}_{2,\,N_2+}$	4.437×10^4	5.125×10^4	6.948×10^4	8.151×10^4	km
$\sigma^{y}_{2,\,N_2+}$	9.662×10^3	3.554×10^4	7.348×10^4	7.477×10^4	km
$\sqrt{\lambda_{\max}^{v_x v_y}}$	6.709	6.709	6.709	6.710	m/s
$\sigma^{v_x}_{2,\,N_2+}$	1.074×10^{-3}	1.089×10^{-3}	1.142×10^{-3}	1.105×10^{-3}	km/s
$\sigma^{v_y}_{2,\,N_2+}$	6.219×10^{-4}	1.038×10^{-3}	1.113×10^{-3}	9.928×10^{-4}	km/s

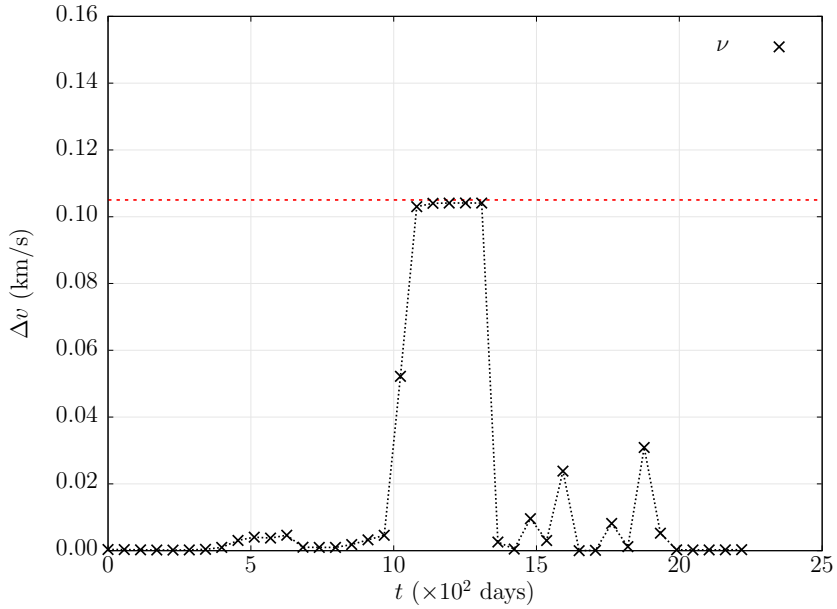
Table 5.4. Comparison between the MC analyses of the CL solution computed with (left column) and without (right column) NE in Table 5.1, considering the same NE in both the MCs.

Quantity	ROCP-CL$_{R \neq 0}$	ROCP-CL$_{R = 0}$	Unit	Ratio
$\Delta r^{x}_{2,\,N_2+}$	56.136	124.763	km	2.223
$\Delta r^{y}_{2,\,N_2+}$	77.411	84.193	km	1.088
$\Delta v^{x}_{2,\,N_2+}$	6.763×10^{-7}	3.236×10^{-5}	km/s	47.844
$\Delta v^{y}_{2,\,N_2+}$	1.077×10^{-5}	4.404×10^{-5}	km/s	4.089
$\sigma^{x}_{2,\,N_2+}$	6.958×10^4	2.493×10^5	km	3.583
$\sigma^{y}_{2,\,N_2+}$	7.344×10^4	9.524×10^4	km	1.297
$\sigma^{v_x}_{2,\,N_2+}$	6.249×10^{-3}	2.856×10^{-2}	km/s	4.571
$\sigma^{v_y}_{2,\,N_2+}$	6.103×10^{-3}	3.312×10^{-2}	km/s	5.426

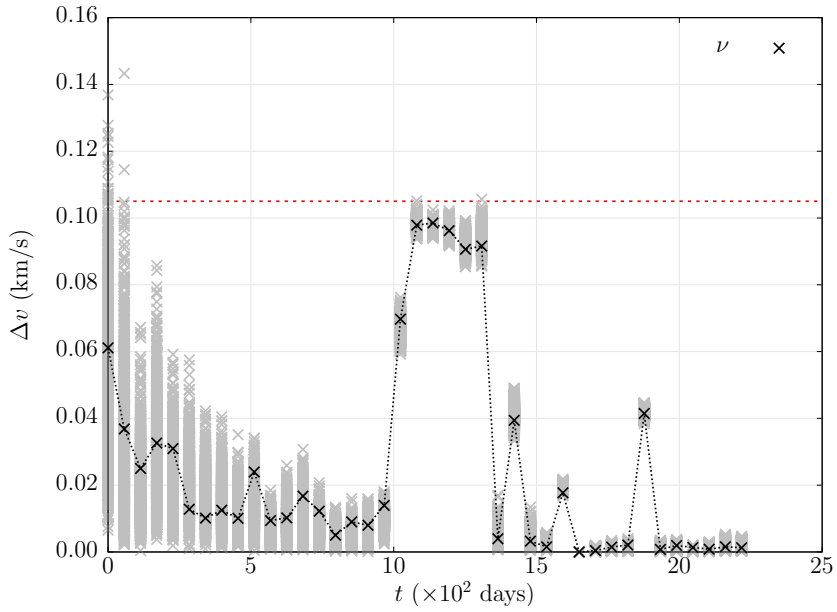
velocity do not appear instead to be particularly influenced by the erroneous control law, since the values are close to each other in both cases.

5.5 Case study B: Station-keeping of NRHO

The Lunar Gateway (LG) is a proposed space station in lunar orbit that will serve as a staging point for future human missions to the Moon and beyond. Unlike previous space stations, the LG will not orbit the Earth but will instead be placed in an NRHO around the Moon. NRHOs are especially convenient due to their cost-effective transfer options from Earth, the ability to facilitate transfers to the lunar surface and other orbits in cislunar space (and beyond), and their favorable eclipsing properties. The usage of lunar NRHOs however is not limited to the LG, but the Orion spacecraft and other vehicles core to the Human Landing System (HLS) program of NASA will rely upon NRHOs to support a variety of future Artemis mission objectives. NRHOs are a subset of the halo family of orbits, which are periodic orbits that lie in the vicinity of the collinear points and are analytically identified in the CR3BP [13]. NRHOs are characterized, as suggested by their name, by a quasi-rectilinear motion, similar to a highly eccentric orbit in Keplerian dynamics.



(a) OCP solution.



(b) CL solution.

Figure 5.10. MC analysis of thrust profiles for the OCP and CL cases.

The CR3BP dynamical model provides a preliminary, yet satisfactory approximation of higher-fidelity dynamical models of binary systems similar to the Earth–Moon system, that can be successfully used to investigate the station-keeping of spacecraft in an NRHO. In the CR3BP, the two main bodies (or primaries) are modeled as point-mass and move around their common barycenter in circular orbits. The motion of the spacecraft, whose mass is assumed negligible in comparison to the primaries, is then described in relation to a synodic coordinate frame, that rotates synchronously with the binary system. In the CR3BP, quantities are non-dimensionalized, and the

distance between the primaries and their mean motion are both set to the unity. The CR3BP admits five relative equilibrium points: the collinear points L_1, L_2, and L_3, located along the primaries line, and two equilateral points, L_4 and L_5, forming equilateral triangles with the two primaries. Since the CR3BP is autonomous, a constant energy integral exists relative to the rotating frame and is defined as the Jacobi constant C_J, such that $C_J = 2U - v^2$, where U is the pseudo-potential function and v is the velocity magnitude relative to the rotating frame. More details on the CR3BP can be found in Appendix B.

5.5.1 Problem data

The considered NRHO is selected among the L_2 family of halo orbits in the Earth–Moon system, resembling the expected NRHO for the LG [129]. The main parameters of this periodic orbit are reported in Table 5.5. This chosen NRHO is among the subset of stable halo orbits [209], which makes it suitable for long-term orbiting platforms such as the LG. The trajectory in the synodic RF is plotted in Fig. 5.11, where the spherical shape approximates the actual Moon size. The trajectory is divided into $L = 8$ legs of equal ToF, and the initial point of each leg is marked with a red square, with a label that specifies the order of encounters. Note that, without any loss of generality, the starting point is arbitrarily chosen as the farthest point from the Moon. Internal nodes, marked by black crosses, further divide each leg into segments of equal duration. Impulsive maneuvers are supposed to be executed at both the starting point of each leg and at the internal nodes.

Table 5.5. Data of the reference NRHO.

Parameter	Value	Unit
Period	6.893	days
x amplitude	1.561×10^4	km
y amplitude	3.774×10^4	km
z amplitude	7.561×10^4	km
C_J	3.041	-

The ROCP outlined in Eqs. (5.42) is now formulated to find out a station-keeping control law for the selected NRHO, having as (mean value of) initial and final state the position and velocity at the point labeled as "1" in Fig. 5.11. The sequence of initial and final state of each i-th leg $\widetilde{\boldsymbol{\mu}}_{i,\,0-}$ and $\widetilde{\boldsymbol{\mu}}_{i,\,N_i+}$ is computed *a priori*, selecting L states along the orbit equally spaced in time. That is, having T as the period of the NRHO, one has

$$\Delta T_i = \Delta T = T/L \qquad \forall i = 1, \ldots, L \tag{5.84}$$

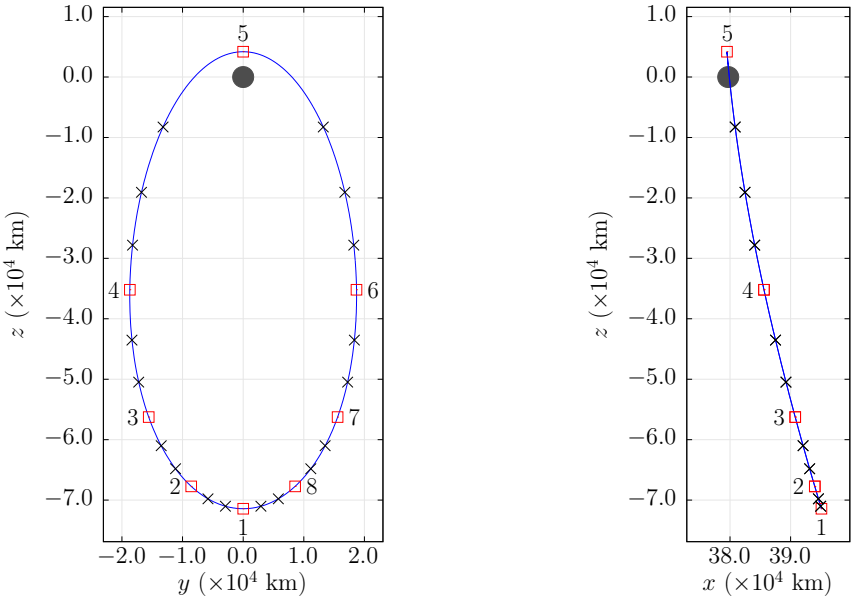
The initial and final points are defined as

$$\widetilde{\boldsymbol{\mu}}_{1,\,0-} = \left[\widetilde{\boldsymbol{r}}^T_{1,\,0-} \;\; \widetilde{\boldsymbol{v}}^T_{1,\,0-}\right]^T = \tag{5.85}$$

$$= \widetilde{\boldsymbol{\mu}}_{L,\,N_L+} = \left[\widetilde{\boldsymbol{r}}^T_{L,\,N_L+} \;\; \widetilde{\boldsymbol{v}}^T_{L,\,N_L+}\right]^T \tag{5.86}$$

At each node, the mean state is constrained to be equal to the value of the (reference) periodic orbit, hence

$$\widetilde{\boldsymbol{\mu}}_{i,\,0-} = \boldsymbol{\phi}\left((i-1)\,\Delta T;\, \boldsymbol{\mu}_{1,\,0-},\, t_{1,\,0}\right) \qquad \forall i = 2, \ldots, L \tag{5.87}$$



(a) yz projection. **(b)** xz projection.

Figure 5.11. Reference NRHO.

The initial dispersion of the spacecraft state at the departure point "1", $\sigma^r_{1,0-} = \sigma^r_0$ and $\sigma^v_{1,0-} = \sigma^v_0$, and the maximum state dispersion $\sigma^r_{i,N_L-} = \sigma^r_{\max}$ and $\sigma^v_{i,N_L-} = \sigma^v_{\max}$ allowed at each node, that is, at each red markers in Fig. 5.11, are selected according to the recent work of Woffinden and Barton [129] and reported in Table 5.6, which provides a solid justification of these figures in terms of correctness and safe execution of Rendezvous, Proximity Operations, and Docking (RPOD). Null NEs and PEs are considered for this test case.

The initial assigned covariance matrix $\widetilde{\boldsymbol{P}}_{1,0-}$ is defined as

$$\widetilde{\boldsymbol{P}}_{1,0-} = \operatorname{diag}\left[\left(\sigma^r_{1,0-}\right)^2 \boldsymbol{I}_3,\ \left(\sigma^v_{1,0-}\right)^2 \boldsymbol{I}_3\right] \tag{5.88}$$

The number of segments that each leg is divided into is selected to provide relatively frequent control maneuvers along the orbit, that is one every approximately 5-7 hours. The control frequency is increased where the velocity is the highest (legs 4 and 5 of the NRHO) to counter the (possibly divergent) growth in state dispersion. Note that, in order to provide a continuous control strategy that can be sequentially applied at each revolution, the terminal maneuver of the last leg is not performed, as the final state of a given revolution coincides with the initial state of the next one, resulting in two distinct maneuvers imparted at the same position. Therefore, the conditions in Eq. 5.32, which avoid duplicity of maneuvers at consecutive legs, are amended by enforcing also

$$\boldsymbol{\nu}_{L,N_L} = \boldsymbol{0}_3 \tag{5.89}$$

$$\boldsymbol{K}_{L,N_L} = \boldsymbol{0}_{3\times 6} \tag{5.90}$$

The maximum magnitude $\widetilde{u}_{\max}$ of each impulse maneuver is selected to have as maximum total Δv expenditure per revolution, with a 3σ confidence level, the one used in the cited work [129], that is 20 m/s. By dividing this amount for the total

number of impulsive maneuvers per revolution $\sum_{i=1}^{L} N_i$, the value of $\widetilde{u}_{\max}$ reported in Table 5.6 is computed. Equations. (5.29) and (5.30) are also applied to the covariance matrix $\boldsymbol{P}_{L,\,N_L-}$ as

$$\boldsymbol{\lambda}\left(\boldsymbol{P}^{r}_{L,\,N_L-}\right) - \left(\sigma^{r}_{\max}\right)^2 \mathbf{1}_3 \leqslant \mathbf{0}_3 \tag{5.91}$$

$$\boldsymbol{\lambda}\left(\boldsymbol{P}^{v}_{L,\,N_L-}\right) - \left(\sigma^{v}_{\max}\right)^2 \mathbf{1}_3 \leqslant \mathbf{0}_3 \tag{5.92}$$

The standard deviations σ^{r}_{w} and σ^{v}_{w} used to compute the perturbation $\overline{\boldsymbol{Q}}$ are to be selected in base of the accuracy level of the dynamical model used for the ROCP. In this instance, only dynamical perturbations on velocity are considered ($\sigma^{r}_{w} = 0$) and the value, rather than derived from relevant literature, is chosen as relatively large, since the CR3BP model, which has strong simplifying assumptions, is adopted for the complex dynamical environment of this specific application. The parameters are summarized in Table 5.6.

Table 5.6. Problem data.

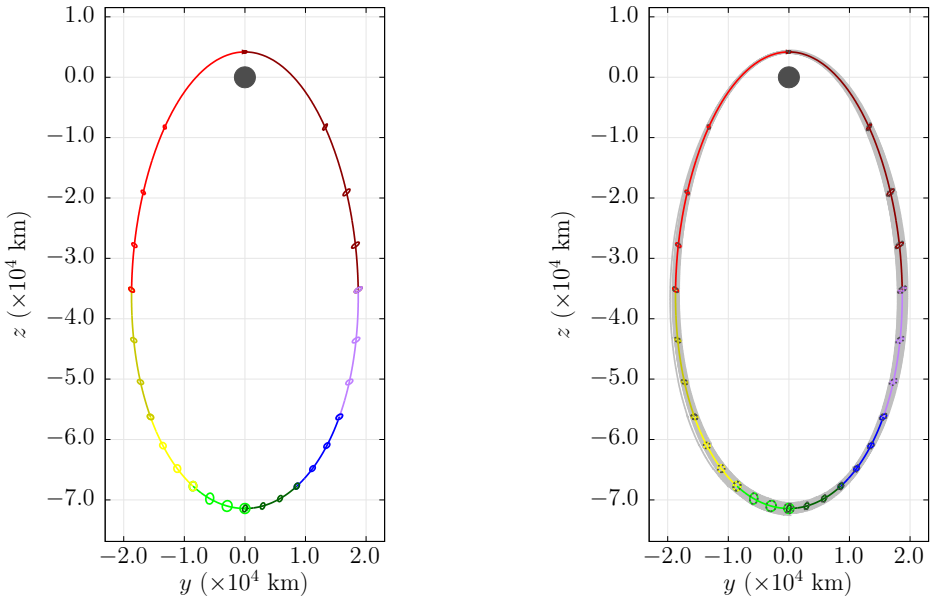
Variable		Value	Unit
L		8	-
N_i	$\forall i = 1, 2, 3, 6, 7, 8$	3	-
N_i	$\forall i = 4, 5$	4	-
ΔT		20.678	hrs
$\widetilde{u}_{\max}$		0.769	m/s
β		5%	-
$\mu = \mu_{\leftmoon} / \left(\mu_{\leftmoon} + \mu_{\oplus}\right)$		0.012	-
$\widetilde{\boldsymbol{r}}_{1,\,0-} = \widetilde{\boldsymbol{r}}_{L,\,N_L+}$		[395 083, 0, −71 423]	km
$\widetilde{\boldsymbol{v}}_{1,\,0-} = \widetilde{\boldsymbol{v}}_{L,\,N_L+}$		[0.000, −0.118, 0.000]	km/s
$3\sigma^{r}_{1,\,0-}$		20	km
$3\sigma^{v}_{1,\,0-}$		1	m/s
$3\sigma^{r}_{\max}$		20	km
$3\sigma^{v}_{\max}$		1	m/s
σ^{r}_{w}		1.216×10^3	km
σ^{v}_{w}		1.025	m/s

The large-scale sparse nonlinear optimizer WORHP [202] was used for solving the optimization problem.

5.5.2 Numerical results

In Fig. 5.12a, the NRHO with the trajectory computed through the formulated ROCP is presented. Each leg is highlighted with a different color in chromatic order. The state dispersion ellipsoids with a 95% of confidence are also plotted with an enlarging factor of 50.

The main features of the CL solutions is summarized in Table 5.7, which presents the deterministic (Δv_{d}), stochastic (Δv_{s}) and total (Δv_{tot}) velocity increment, as well as the mean terminal errors of the final position ($\Delta r^{x}_{8,\,N_8+}$, $\Delta r^{y}_{8,\,N_8+}$, and $\Delta r^{z}_{8,\,N_8+}$) and velocity ($\Delta v^{x}_{8,\,N_8+}$, $\Delta v^{y}_{8,N_8+}$, and $\Delta v^{z}_{8,\,N_8+}$) after one revolution. The final standard deviations for position ($\sigma^{x}_{8,\,N_8+}$, $\sigma^{y}_{8,\,N_8+}$, and $\sigma^{z}_{8,\,N_8+}$) and velocity ($\sigma^{v_x}_{8,\,N_8+}$, $\sigma^{v_y}_{8,\,N_8+}$, and $\sigma^{v_z}_{8,3+}$) are also included as comparison withe MC solution. The total Δv_{d}, although negligible, is not exactly null; this is due to the tolerances of



(a) CL solution with position ellipses up-scaled by a factor 50.

(b) MC trajectories (in gray) with corresponding position ellipses (dashed green line). Ellipses are up-scaled by a factor 50.

Figure 5.12. Trajectory resulting from the ROCP with 95%-confidence position ellipses.

the constraints in the optimization algorithm settings. By implementing stricter tolerances, this minor numerical concern can be prevented.

An MC analysis of 100 000 independent runs was carried out to verify the accuracy of the estimated probability distribution at each step of the ROCP solution. Starting from a random initial state $\boldsymbol{x}_{0-} \sim \mathcal{N}\left(\tilde{\boldsymbol{\mu}}_{0-}, \tilde{\boldsymbol{P}}_{0-}\right)$, each trajectory is propagated for one revolution using the feedback control law in Eq. (5.3). The stochastic fourth-order Runge-Kutta integration method for SDE discussed in the work of N. J. Kasdin [85] was used to propagate each MC trajectory. Numerical results are reported in Table 5.7 to allow for a comparison between the MC and ROCP solution. It is possible to observe how state and dynamical errors are mitigated in presence of the CL control law. It should be noted that all MC values for the CL case are in reasonable agreement with the values estimated in the ROCP with the UT.

The MC campaign of the trajectory in Fig. 5.12a is presented in Fig. 5.12b. The ellipsoids computed from the MC analysis are also plotted in dashed dark gray lines. Good agreement with the covariance ellipsoids found by solving the ROCP in Eqs. (5.42) is shown, as the MC and ROCP ellipses well overlap along the trajectory. This is quantitatively confirmed by the plots in Figs. 5.13 andì 5.14, where the standard deviations of position and velocity before each impulse (that is, the squared diagonal terms of $\boldsymbol{P}_{i,k-}$ matrices) are plotted for each impulsive maneuver. After the perilune, the UT loses accuracy in propagating the uncertainty, although confirming a reasonable alignment with the MC values, with maximum errors around 10%.

In Figs. 5.15a and 5.15b, a different visualization of the MC analysis is shown. The difference vectors for position and velocity, $\Delta\boldsymbol{r}(t)$ and $\Delta\boldsymbol{v}(t)$, between the state of a single MC run and the reference NRHO is computed at every epoch t for one

Table 5.7. Comparison between the solution of the ROCP and the related MC analysis.

Quantity	ROCP	MC	Unit
Δv_{tot}	3.027	3.142	m/s
Δv_{d}	1.693×10^{-2}	1.693×10^{-2}	m/s
Δv_{s}	3.010	3.125	m/s
$\Delta r^{x}_{8,\,N_8+}$	-	0.205	km
$\Delta r^{y}_{8,\,N_8+}$	-	5.713×10^{-2}	km
$\Delta r^{z}_{8,\,N_8+}$	-	0.123	km
$\Delta v^{x}_{8,\,N_8+}$	-	1.105×10^{-3}	m/s
$\Delta v^{y}_{8,\,N_8+}$	-	4.575×10^{-4}	m/s
$\Delta v^{z}_{8,\,N_8+}$	-	1.890×10^{-4}	m/s
$\sigma^{x}_{8,\,N_8+}$	4.636	4.728	km
$\sigma^{y}_{8,\,N_8+}$	3.072	3.308	km
$\sigma^{z}_{8,\,N_8+}$	4.557	5.432	km
$\sigma^{v_x}_{8,\,N_8+}$	1.776×10^{-2}	1.960×10^{-2}	m/s
$\sigma^{v_y}_{8,\,N_8+}$	2.537×10^{-2}	2.304×10^{-2}	m/s
$\sigma^{v_z}_{8,\,N_8+}$	9.824×10^{-3}	1.132×10^{-2}	m/s

revolution, as

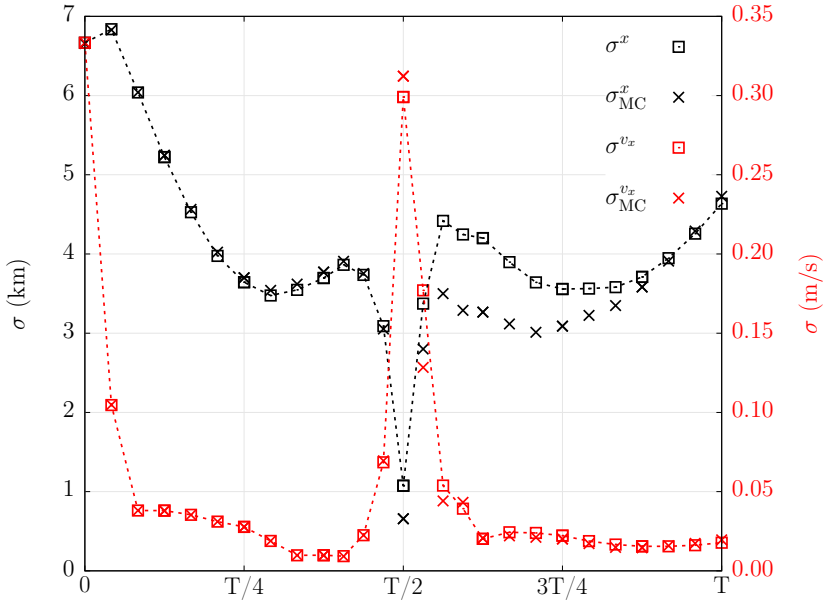
$$\Delta \boldsymbol{r}(t) = \boldsymbol{r}_{\text{MC}}(t) - \boldsymbol{\mu}^{r}(t) \tag{5.93}$$

$$\Delta \boldsymbol{v}(t) = \boldsymbol{v}_{\text{MC}}(t) - \boldsymbol{\mu}^{v}(t) \tag{5.94}$$

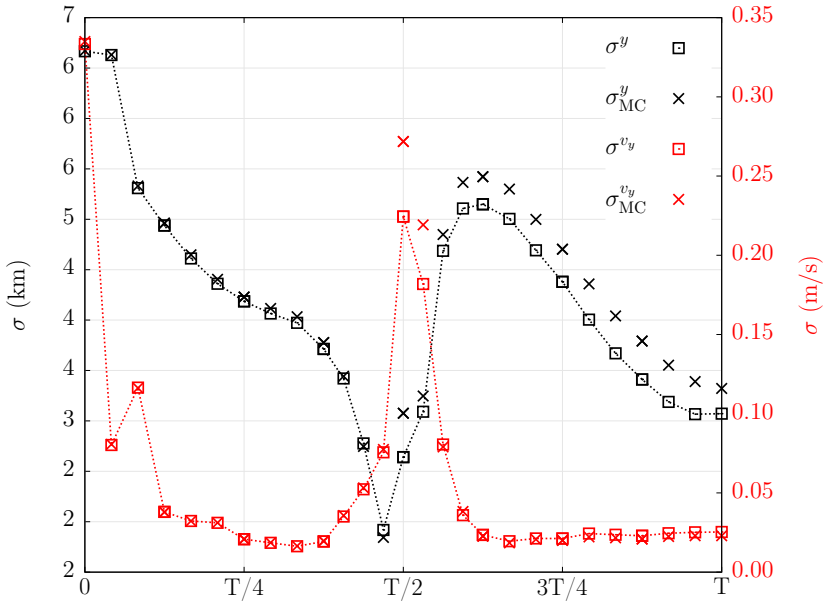
where $\boldsymbol{\mu}^{r}(t)$ and $\boldsymbol{\mu}^{v}(t)$ are respectively the position and velocity of the reference NRHO at time t. The evolution in time of the norm of $\Delta \boldsymbol{r}(t)$ and $\Delta \boldsymbol{v}(t)$ for every MC run is plotted in the figures, where also the maximum eigenvalues of the two related covariance matrices for position and velocity (computed considering the reference NRHO as mean state) are included. In the plots, the values of the maximum allowed σ^{r}_{max} and σ^{v}_{max}, together with the related 3-sigmas values, is traced as threshold. The initial value of the maximum position eigenvalue is close to the red dotted line as expected (initial dispersion is set as the maximum value, as in Table 5.6), while the maximum velocity eigenvalue starts well below its limit as the curve begins after the first impulse, thus with an already shrunk dispersion (effect of the maneuver as in Eq. (5.11)). The most limiting effect of the control law against the spread of state uncertainty takes place in proximity of the perilune, as in this phase the velocity dispersion reaches its peak. At the end of leg 4, the maximum velocity eigenvalue approaches the allowed limit but still satisfies the associated constraint. Consequently, the maximum position eigenvalue at the same moment remains well below the threshold due to the velocity dispersion being the primary limiting factor. In leg 5, the maximum velocity eigenvalue exceeds instead the threshold at the beginning since the constraints are valid only at the end of each leg, confirming however the tendency of the velocity dispersion to spread up at perilune.

5.6 Open-loop and closed-loop performance comparison

The performance capabilities of the CL formulation were discussed and analyzed in the previous test cases (Sec. 5.4 and 5.5), demonstrating significant improvements in robustness over the deterministic OCP solution. However, it is also worthwhile



(a) x, v_x



(b) y, v_y

Figure 5.13. Covariance matrices trace comparison of the in-plane dimension.

to evaluate the enhancements obtained from the OL formulation. Therefore, this section presents a comparative analysis between the proposed impulsive OL and CL ROCP, focusing on final state dispersion and control effort for the same test case.

5.6.1 Comparison formulation

The trajectory considered for this comparison is the test case for the OL ROCP in Sec. 4.2, that is, the non-dimensional transfer from two concentric orbits. The

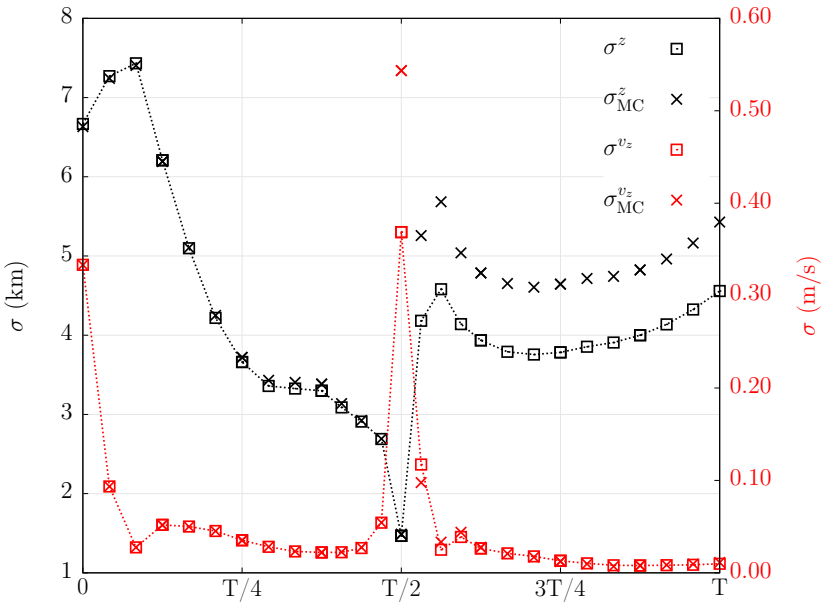


Figure 5.14. Covariance matrices trace comparison of the out-of-plane dimension.

same parameters are used in this comparison, as in Table 4.1, with the exception of the δv incremental parameter η, which is not considered in this comparison, as later explained.

A fundamental adjustment to the CL formulation for a coherent comparison involves including the ballistic arc duration, δt_k, in the optimization variables vector, rather than assuming arcs of equal duration $\Delta T/N$. This is essential because arcs with variable durations are considered for the OL control. Consequently, the single-leg CL ROCP in Eq. 5.23 must be updated to account for the new optimization variables vector.

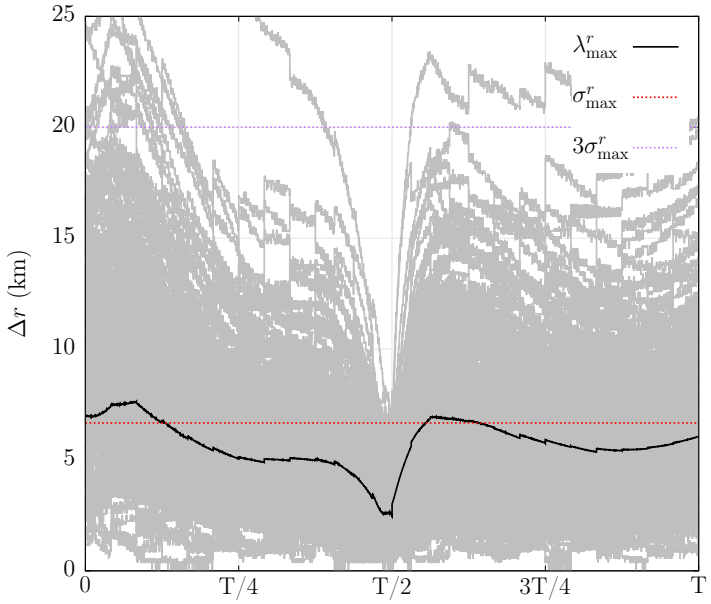
$$\boldsymbol{\Theta} = \{\Delta t_k,\, \boldsymbol{\mu}_{k-},\, \boldsymbol{\nu}_k,\, \boldsymbol{K}_k,\, \forall k = 0, \ldots, N\} \tag{5.95}$$

which results as a vector of dimension $(n^2 + 3n + 2)(N + 1)/2$. A further constraint on the total ToF is consequently added, with the same value of the OL test case.

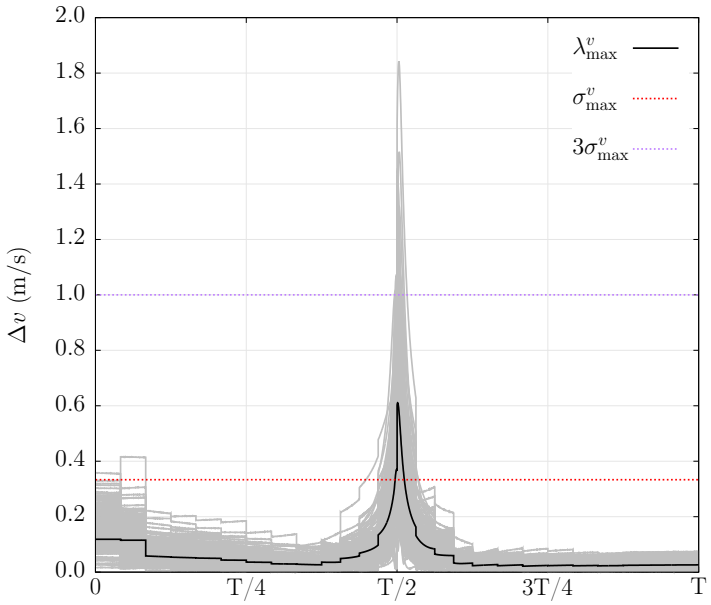
Another fundamental adjustment for the CL problem is that, since variable ToF is now assumed, final mean state $\boldsymbol{\mu}_{N+}$ is no longer fixed, but rather equal to the state $\widetilde{\boldsymbol{x}}_f$ in Eq. (4.22). The initial mean state $\boldsymbol{\mu}_{0-}$ is instead equal to the fixed state $\widetilde{\boldsymbol{x}}_i$ in Eq. (4.20). No random perturbation and navigation errors are included in both the formulations.

The comparison is conducted as follows: the CL ROCP is initially performed selecting arbitrary values for the maximum standard deviations of position and velocity, as to comply with constraints in Eqs. (5.16) and (5.17). Once the CL solution is obtained, the resulting Δv_{tot}, sum of deterministic and stochastic Δv, is selected as maximum threshold for the control effort of the OL ROCP. This approach ensures that the OL solution is constrained to use a control effort that does not exceed the total control effort used in the CL ROCP with a $p = 1 - \beta$ confidence level, evaluating whether the OL approach can achieve comparable robustness and performance employing the same control effort.

The parameters for the comparison are summarized in Table 5.8, having first the parameters that are common in the two formulations, and then the ones specific



(a) MC of the norm in position.



(b) MC of the norm in velocity.

Figure 5.15. Difference from the MC runs and the reference NRHO.

for each of the two. For the CL control, the final standard deviations are selected as equal to the initial ones.

5.6.2 Numerical results

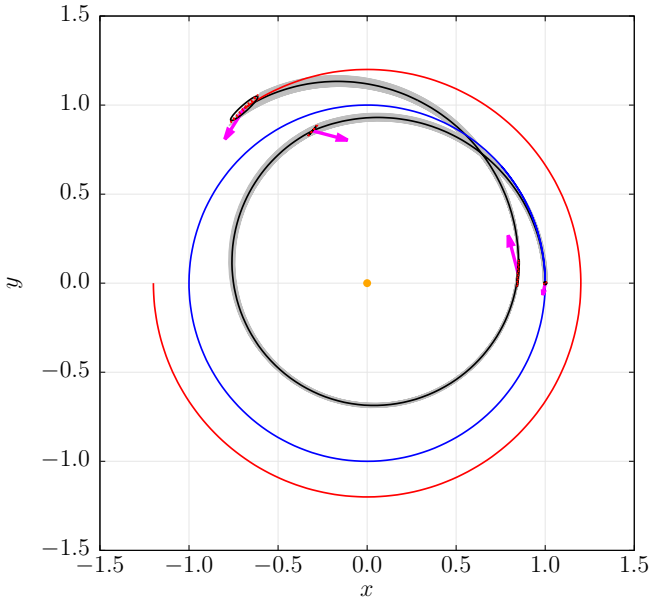
In Fig. 5.16, the trajectories obtained from the two robust methodologies are plotted together with the MC runs (100 000) and the position ellipsoids, with 95% confidence level, derived from the MC analysis and UT. Table 5.9 compares the main

Table 5.8. Problem data.

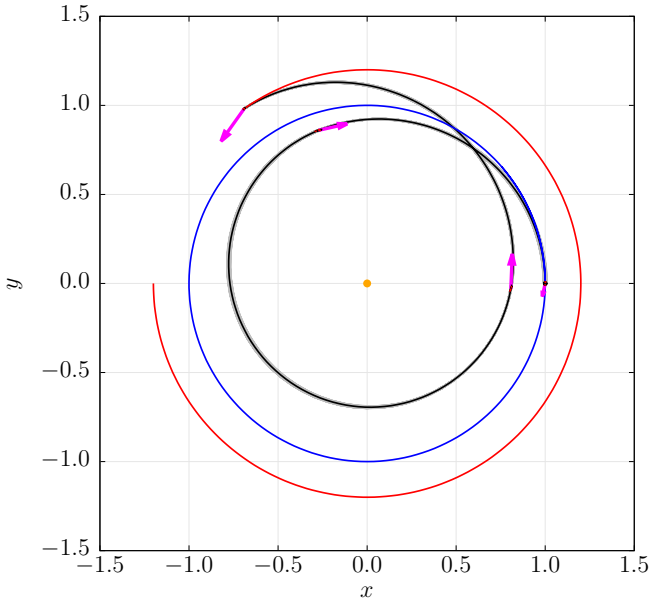
Variable	Value	Unit
$\Delta\widetilde{t}_{\max}$	7	-
$\widetilde{u}_{\max}$	0.11	-
N	3	-
μ	1	-
r_1	1	-
r_2	1.2	-
θ_1	0	rad
θ_2	π	rad
	CL	
β	5%	-
$\widetilde{\boldsymbol{\mu}}_{0-}$	$\widetilde{\boldsymbol{x}}_i$	-
$\widetilde{\boldsymbol{\mu}}_{N+}$	$\widetilde{\boldsymbol{x}}_f$	-
σ^r_{0-}, σ^v_{0-}	1.000×10^{-5}	-
σ^v_{0-}	1.000×10^{-5}	-
σ^r_{N+}	1.000×10^{-5}	-
σ^v_{N+}	1.000×10^{-5}	-
	OL	
$\Delta\widetilde{v}_{\max}$	Δv_{tot} from ROCP-CL	-
σ^r_0	1.000×10^{-5}	-
σ^v_0	1.000×10^{-5}	-

features of the two solutions. The cumulative velocity change Δv_{tot} is intended to be, for the OL case, the sum of the deterministic maneuvers, while, in the CL case, the sum of the deterministic and stochastic Δv as expressed by Eq. (5.22). Similarly, the maximum eigenvalues for position and velocity $\lambda^r_{\max}$ and $\lambda^v_{\max}$ correspond to the final covariance matrix $\boldsymbol{P}_N$ in the OL case, and $\boldsymbol{P}_{N+}$ in the CL case. Both the plots and the table shows a significant improvement in robustness in the CL formulation, as the final position and velocity eigenvalues are 2-3 orders of magnitude smaller than the OL solution. It is important to note that, in both cases, the algorithm selects the maximum ToF, as expected. Additionally, only minor variations in the individual durations of the ballistic arcs are observed between the two cases.

The magnitudes of the impulsive maneuvers, for the OL case, and the deterministic components of the impulsive maneuvers, for the CL case, are reported in Table 5.10, together with the angle ϕ.



(a) OL solution.



(b) CL solution.

Figure 5.16. Nominal solutions of OL and CL formulations, together with MC trajectories (in gray) and corresponding position ellipses (dashed red line) and UT position ellipses (solid line of the leg color).

Table 5.9. Comparison between the solution of the ROCP in OL and CL formulation.

Quantity	ROCP-OL	ROCP-CL
Δv_{tot}	0.348	0.348
Δt_1	1.835	1.794
Δt_2	2.856	2.852
Δt_3	2.309	2.354
Δt_{tot}	7.000	7.000
$\lambda^r_{\max}$	1.696×10^{-3}	6.045×10^{-6}
$\lambda^v_{\max}$	9.667×10^{-4}	1.431×10^{-6}

Table 5.10. Comparison between the solution of the ROCP in OL and CL formulation for the magnitudes and directions of the impulsive maneuvers.

Maneuver	ROCP-OL		ROCP-CL	
	$\|\boldsymbol{u}\|$	ϕ	$\|\boldsymbol{\nu}\|$	ϕ
#1	3.384×10^{-2}	258.463	3.738×10^{-2}	257.238
#2	1.022×10^{-1}	345.152	8.287×10^{-2}	12.738
#3	1.100×10^{-1}	104.766	9.825×10^{-2}	87.640
#4	1.025×10^{-1}	237.834	1.124×10^{-1}	234.679

Chapter 6

Closed-loop methodology for low-thrust trajectories

6.1 General formulation

Insofar, only bounded-magnitude impulsive trajectories have been considered for robust optimization. Goal of this chapter is to formulate an ROCP for continuous low-thrust transfer with fixed initial and final state, described as position and velocity, ToF, and control maneuvers bounded in magnitude. Minimization of the propellant consumption is sought, while the state dispersion is fixed for the initial state and limited at the arrival.

As in Chapter 5, the spacecraft is modeled as a point-mass object and its state is defined in terms of the Cartesian position and velocity components, as $\boldsymbol{x}'(t) = \left[\boldsymbol{r}(t)^T\, \boldsymbol{v}(t)^T\right]^T$. However, since a low-thrust model is here adopted, mass $m(t) \in \mathbb{R}_+$ is also included as a state variable. Thus, hereafter the augmented (or complete) state $\boldsymbol{x}(t) = \left[\boldsymbol{x}'(t)\, m(t)\right]^T \in \mathbb{R}^n$, $n = 7$, composed by the Cartesian state and mass, is adopted.

The initial and final transfer epochs are denoted as t_i and t_f, respectively. The trajectory is assumed to be comprised of N arcs of equal duration, where, differently from the impulsive method, the thrust remains constant in both magnitude and direction throughout each arc. A uniform time grid is thus defined as

$$\widetilde{t}_i = t_0 \leqslant t_1 \leqslant \cdots \leqslant t_N = \widetilde{t}_f \tag{6.1}$$

having again $\Delta t_k = t_k - t_{k-1} = (\widetilde{t}_f - \widetilde{t}_i)/N$, $\forall k = 1, \ldots, N$.

For the sake of conciseness, the state at time t_k is denoted as $\boldsymbol{x}(t_k) = \boldsymbol{x}_k$. Analogously, $\boldsymbol{r}_k$, $\boldsymbol{v}_k$, m_k are position, velocity, and mass at time t_k, respectively. Note that in this model a continuous thrust is considered; hence, the state is now continuous at grid nodes. The constraint on the thrust $\|\boldsymbol{T}_k\| \leqslant \widetilde{T}_{\max}$ is applied to ensure that the required thrust complies with the technological limitations of the engine.

6.1.1 Linear control law for piece-wise constant thrust arcs

The thrust $\boldsymbol{T}_k \in \mathbb{R}^3$, imparted by the engines, is assumed constant in direction and magnitude during the arc. Similarly to the impulsive case, the proposed approach is based on the definition of a linear CL control used to evaluate the thrust vector for

each arc as

$$\boldsymbol{T}_k = \boldsymbol{S}_k + \boldsymbol{K}_k \left(\boldsymbol{x}'_{k-1} - \boldsymbol{\mu}'_{k-1}\right) \qquad \forall k = 1, \ldots, N \tag{6.2}$$

where $\boldsymbol{S}_k \in \mathbb{R}^3$ represents the deterministic component of the thrust, while the CL correction is $\boldsymbol{K}_k \left(\boldsymbol{x}'_{k-1} - \boldsymbol{\mu}'_{k-1}\right)$, which depends on the difference between the actual Cartesian state $\boldsymbol{x}'_{k-1} = \boldsymbol{x}'(t_{k-1})$ at the beginning of the k-th arc and its mean value $\boldsymbol{\mu}'_{k-1} = \mathbb{E}\left[\boldsymbol{x}'_{k-1}\right]$, multiplied by a feedback gain matrix $\boldsymbol{K}_k \in \mathbb{R}^{3\times 6}$. Note that the trajectory is guided towards the reference mean by leveraging only on position and velocity as driving factors, rather than the full augmented state $\boldsymbol{x}(t)$, which would also comprise the mass $m(t)$.

6.1.2 Perturbed state propagation

In the case of low-thrust mission, the spacecraft (perturbed) dynamics is represented by a set of SDEs, defined as the deterministic EoM, as in Eq. (2.54), with the additional presence of the spacecraft mass. Hence,

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{T}_k, t)\mathrm{d}t + \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{T}_k, t)\mathrm{d}\boldsymbol{w} \qquad \forall t \in [t_{k-1}, t_k] \qquad \forall k = 1, \ldots, N \tag{6.3}$$

where, $\boldsymbol{f}\colon \mathbb{R}^7 \times \mathbb{R} \mapsto \mathbb{R}^7$ is the drift term, $\boldsymbol{g}\colon \mathbb{R}^7 \times \mathbb{R} \mapsto \mathbb{R}^7 \times \mathbb{R}^{n_w}$ is the diffusion term. For the sake of simplicity, $n_w = n = 7$ is considered hereafter.

As done in Sec. 5.1.2, for the purpose of UT propagation, the effect due to the diffusive term in Eq. (6.3) over each k-th arc of trajectory, that represents the cumulative effects of stochastic perturbations, is modeled as an instantaneous, additive, Gaussian perturbation with zero mean and covariance $\boldsymbol{Q}_k$, applied at the end of the arc. The matrix $\boldsymbol{Q}_k$ is estimated for each arc along a reference trajectory obtained from the optimal deterministic (i.e., non-robust) solution, following the same approach outlined in Sec. 5.1.2 and Appendix A. As a major difference from the impulsive case, where the same average matrix $\bar{\boldsymbol{Q}}$ was added at the end of each arc, a more accurate estimation of the stochastic disturbance is required; hence, a different $\boldsymbol{Q}_k$ is used for each arc. In this respect, the function $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{T}, t)$ is approximated as a constant matrix $\boldsymbol{G} \in \mathbb{R}^{7\times 7}$, as

$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{T}, t) = \boldsymbol{G} = \begin{bmatrix} (\sigma_w^r)^2\, \boldsymbol{I}_3 & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 1} \\ \boldsymbol{0}_{3\times 3} & (\sigma_w^v)^2\, \boldsymbol{I}_3 & \boldsymbol{0}_{3\times 1} \\ \boldsymbol{0}_{1\times 3} & \boldsymbol{0}_{1\times 3} & (\sigma_w^m)^2 \end{bmatrix} \tag{6.4}$$

where σ_w^r, σ_w^v, and $\sigma_w^m \in \mathbb{R}$ model the variance of the position, velocity, and mass random disturbance, respectively.

Given the CL control expression for the thrust vector in Eq. (6.2), uncertainty propagation tools prove inadequate for estimating the final state. This limitation arises as the EoM, formulated by the low-thrust dynamical system in Eq. (6.3) with the control law specified in (6.2), involves a term dependent on the actual initial state, making the differential equations dependant on the initial state. This unique characteristic introduces a level of complexity in the propagation of stochastic moments, deviating from the more straightforward nature of a pure ballistic arc. The presence of a term dependent on the initial state makes many conventional techniques infeasible or exceedingly challenging for this specific scenario. Hence, to effectively propagate the first two stochastic moments of the state and the CL control, essential for solving an SOCP, this work employs a modified version of the UT, here outlined.

Let the initial state be $\boldsymbol{x}_{k-1} \sim \mathcal{N}(\boldsymbol{\mu}_{k-1}, \boldsymbol{P}_{k-1})$, $k = 1, \dots, N$, where $\boldsymbol{\mu}_{k-1} = \mathbb{E}[\boldsymbol{x}_{k-1}]$ and $\boldsymbol{P}_{k-1} = \mathrm{Cov}[\boldsymbol{x}_{k-1}]$. The CL control law of Eq. 6.2 is defined by parameters $\boldsymbol{S}_k$ and $\boldsymbol{K}_k$. Eventually, the mean $\boldsymbol{\mu}_k$ and the covariance $\boldsymbol{P}_k$ of the state $\boldsymbol{x}_k$ at time t_k are evaluated as follows:

1. generate a set of $2n+1$ SPs $\boldsymbol{\mathcal{X}}_i$ and the related weights c_i for $\boldsymbol{x}_{k-1}$

$$\boldsymbol{\mathcal{X}}_0 = \boldsymbol{\mu}_{k-1} \tag{6.5}$$

$$\boldsymbol{\mathcal{X}}_j = \boldsymbol{\mu}_{k-1} + (\sqrt{(n+\kappa)\boldsymbol{P}_{k-1}})_j \qquad \forall j = 1, \dots, n \tag{6.6}$$

$$\boldsymbol{\mathcal{X}}_{n+j} = \boldsymbol{\mu}_{k-1} - (\sqrt{(n+\kappa)\boldsymbol{P}_{k-1}})_j \qquad \forall j = 1, \dots, n \tag{6.7}$$

$$c_0 = \kappa/n + \kappa \tag{6.8}$$

$$c_i = \frac{1}{2(n+\kappa)} \qquad \forall i = 1, \dots, 2n \tag{6.9}$$

2. for each SP the thrust vector is computed as

$$\boldsymbol{T}_0 = \boldsymbol{S}_k \tag{6.10}$$

$$\boldsymbol{T}_i = \boldsymbol{S}_k + \boldsymbol{K}_k \left(\boldsymbol{\mathcal{X}}'_i - \boldsymbol{\mu}'_{k-1}\right) \qquad \forall i = 1, \dots, 2n \tag{6.11}$$

where $\boldsymbol{\mathcal{X}}'_i \in \mathbb{R}^{n-1}$ corresponds to the SP $\boldsymbol{\mathcal{X}}_i$ excluding the mass term, mirroring the structure of $\boldsymbol{x}'_{k-1}$ in relation to $\boldsymbol{x}_{k-1}$. Note that $\boldsymbol{\mathcal{X}}'_{2n-1} = \boldsymbol{\mathcal{X}}'_{2n} = \boldsymbol{\mu}'_{k-1}$, implying that $\boldsymbol{T}_{2n-1} = \boldsymbol{T}_{2n} = \boldsymbol{T}_0$. This equivalence arises because, for these two specific values of i, the corresponding SPs exhibit variations solely in mass. Consequently, their forms without the mass term are precisely identical to the mean position and velocity $\boldsymbol{\mu}'_{k-1}$.

The covariance $\boldsymbol{P}_k^{\mathcal{T}}$ of the thrust vector $\boldsymbol{T}_k$ is evaluated as

$$\boldsymbol{P}_k^{\mathcal{T}} = \sum_{i=1}^{2(n-1)} \frac{1}{2(n-1)} (\boldsymbol{T}_i - \boldsymbol{S}_k)(\boldsymbol{T}_i - \boldsymbol{S}_k)^T \tag{6.12}$$

since $\mathbb{E}[\boldsymbol{T}_i] = \boldsymbol{S}_k$. The pivotal thrust vector $\boldsymbol{T}_0$ is intentionally excluded from the computation of $\boldsymbol{P}_k^{\mathcal{T}}$ to align with the choice of $\kappa = 0$ in the state UT propagation. The thrust vectors $\boldsymbol{T}_{2n-1}$ and $\boldsymbol{T}_{2n}$ are likewise omitted from the computation, given that both are equal to $\boldsymbol{T}_0$.

3. the SPs $\boldsymbol{\mathcal{X}}_i$ are propagated to obtain the new set of samples $\boldsymbol{\mathcal{Y}}_i$, $i = 0, \dots, 2n$. The propagation is performed using the deterministic component of the EoM in Eq. (6.3), as

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{T}_i, t)\mathrm{d}t \qquad \forall t \in [0, \Delta t_k] \qquad \forall i = 0, \dots, 2n \tag{6.13}$$

The propagated SPs are thus

$$\boldsymbol{\mathcal{Y}}_i = \boldsymbol{\phi}(\Delta t_k; \boldsymbol{\mathcal{X}}_i, 0) \qquad \forall i = 0, \dots, 2n \tag{6.14}$$

The final mean state $\boldsymbol{\mu}_k$ and its covariance matrix $\boldsymbol{P}_k$ are thus computed as

$$\boldsymbol{\mu}_k = \sum_{i=0}^{2n} c_j \boldsymbol{\mathcal{Y}}_i \tag{6.15}$$

$$\boldsymbol{P}_k = \sum_{i=0}^{2n} c_j (\boldsymbol{\mathcal{Y}}_i - \boldsymbol{\mu}_k)(\boldsymbol{\mathcal{Y}}_i - \boldsymbol{\mu}_k)^T \tag{6.16}$$

The propagation of the mean state and the covariance matrix of the state and thrust vector is thus summarized by the expression

$$\left\{\boldsymbol{\mu}_k,\, \boldsymbol{P}_k,\, \boldsymbol{P}_k^{\mathcal{T}}\right\} = \mathcal{F}\left(\boldsymbol{\mu}_{k-1},\, \boldsymbol{P}_{k-1},\, \boldsymbol{S}_k,\, \boldsymbol{K}_k,\, \Delta t_k;\, \boldsymbol{Q}_k\right) \qquad \forall k = 1, \ldots, N \tag{6.17}$$

Here, $\mathcal{F}(\cdot)$ embodies the functionality of any uncertainty propagation tool proficient in delivering the control covariance, as well as the propagated mean and covariance of the state based on the control parameters, time interval, and perturbation parameters.

6.1.3 Boundary constraints

For the low-thrust case, the initial stochastic moments are defined similarly to the impulsive ROCP as

$$\boldsymbol{\mu}_0 = \widetilde{\boldsymbol{\mu}}_0 = [\widetilde{\boldsymbol{\mu}}_0^r,\, \widetilde{\boldsymbol{\mu}}_0^v,\, \widetilde{\mu}_0^m] \tag{6.18}$$

$$\boldsymbol{P}_0 = \widetilde{\boldsymbol{P}}_0 \tag{6.19}$$

with $\widetilde{\boldsymbol{\mu}}_0^r$, $\widetilde{\boldsymbol{\mu}}_0^v$ and $\widetilde{\mu}_0^m$ as the assigned mean position, velocity and mass respectively. The final mean state is instead simply enforced with

$$\boldsymbol{\mu}'_N = \widetilde{\boldsymbol{\mu}}'_N = [\widetilde{\boldsymbol{\mu}}_N^r,\, \widetilde{\boldsymbol{\mu}}_N^v] \tag{6.20}$$

as no constraints are imposed on the final mass to ensure that there are no limitations on the trajectory in terms of mass consumption. By selecting σ_N^r and σ_N^v as user-defined parameters for the standard deviation of final position and velocity respectively, the constraints are enforced on the eigenvalues of the final covariance matrix $\boldsymbol{P}'_N = \mathrm{Cov}\left[\boldsymbol{x}'_N\right]$, following the same formulation of the impulsive case as in Eqs. (5.16) and (5.17)

$$\boldsymbol{\lambda}\left(\boldsymbol{P}_N^r\right) - \left(\sigma_N^r\right)^2 \mathbf{1}_3 \leqslant \mathbf{0}_3 \tag{6.21}$$

$$\boldsymbol{\lambda}\left(\boldsymbol{P}_N^v\right) - \left(\sigma_N^v\right)^2 \mathbf{1}_3 \leqslant \mathbf{0}_3 \tag{6.22}$$

where $\boldsymbol{P}_N^r$ and $\boldsymbol{P}_N^v$ are the two 3×3 sub-matrices of $\boldsymbol{P}'_N$ for position and velocity, constituting the diagonal blocks of $\boldsymbol{P}_N$. As discussed in Sec. 5.1.3, this approach prioritizes constraining the magnitude of state dispersion, regardless of the values of its components.

6.1.4 Cost function and control constraints

The aim of the optimization is to minimize the propellant consumption given by $\sum_{k=0}^{N} \|\boldsymbol{T}_k\| \Delta t_k / c$, which, under the assumption of uniform time grid and constant I_{sp}, is proportional to the cumulative thrust magnitude, that is

$$\min_{\boldsymbol{T}_k} J = \sum_{k=0}^{N} \|\boldsymbol{T}_k\| \tag{6.23}$$

Similarly to the CL impulsive case, the thrust control vector is a stochastic variable. Thus, any quantity which is dependant on $\boldsymbol{T}_k$ requires a numerically tractable formulation to be included in the optimization algorithm. In this respect, the cost function of Eq. (6.23) must be read in a probabilistic sense as in Eq. (2.57). Assuming the state $\boldsymbol{x}'_{k-1}$ follows a multivariate Gaussian distribution, the thrust

vector $\boldsymbol{T}_k$, governed by the feedback control law in Eq. (6.2), can, in turn, be considered a random Gaussian vector, i.e., $\boldsymbol{T}_k \sim \mathcal{N}\left(\boldsymbol{S}_k,\, \boldsymbol{P}_k^{\mathcal{T}}\right)$. Consequently, the squared norm of the control, $\|\boldsymbol{T}_k\|^2$, follows a non-general central χ-square distribution [210]. This assumption holds true for degrees of position and velocity dispersion that align with the typical navigation requirements of space missions. The justification for this assumption is based on the efficacy of the proposed method, as the central objective of the designed robust control is to effectively constrain this dispersion below the allowable thresholds.

The theorem from Ridderhof et al. [103], already adopted for the impulsive ROCP in Sec. 5.1.4, provides a strong bound for the p-th percentile of a χ-square distribution. By applying this theorem with $1-\beta$ again as the compliance rate, the chance-constraint on the control magnitude bound,

$$\Pr\left\{\|\boldsymbol{T}_k\| \leqslant \widetilde{T}_{\max}\right\} \geqslant \alpha \qquad \forall k = 1, \ldots, N \tag{6.24}$$

can be rewritten as

$$\|\boldsymbol{S}_k\| + \gamma(\beta)\sqrt{\lambda_{\max}\left(\boldsymbol{P}_k^{\mathcal{T}}\right)} \leqslant T_{\max} \qquad \forall k = 1, \ldots, N \tag{6.25}$$

where $\gamma(\beta) = \sqrt{2\log\frac{1}{\beta}} + \sqrt{w}$, where $w = (n-1)/2$ is the dimension of $\boldsymbol{T}_k$. The term $\|\boldsymbol{S}_k\| = T_{\mathrm{d},\,k}$ is the norm of the deterministic component of thrust vector control $\boldsymbol{T}_k$ and $\gamma(\beta)\sqrt{\lambda_{\max}\left(\boldsymbol{P}_k^{\mathcal{T}}\right)} = T_{\mathrm{s},\,k}$ is a measure (up to a confidence level $1-\beta$) of the control effort due to the CL term.

The theorem also allows to express the merit index Eq. (6.23) in a straightforward form, that is

$$J = \sum_{k=1}^{N}\left(\|\boldsymbol{S}_k\| + \gamma(\beta)\sqrt{\lambda_{\max}(\boldsymbol{P}_k^{\mathcal{T}})}\right) \tag{6.26}$$

where the first term of Eq. (5.22) accounts for the cumulative deterministic cost, while the second term measures the CL control effort, that depends on the dispersion of the control vector through the matrix $\boldsymbol{P}_k^{\mathcal{T}}$.

It is noteworthy to recall that with this formulation the optimization algorithm evaluates the most efficient trade-off between the overall deterministic thrust expenditure, denoted as $T_{\mathrm{d}} = \sum_{k=1}^{N} T_{\mathrm{d},\,k}$, and the CL control term $T_{\mathrm{s}} = \sum_{k=1}^{N} T_{\mathrm{s},\,k}$. This strategic flexibility aims to diminish the overall thrust magnitude, expressed as $T_{\mathrm{tot}} = T_{\mathrm{d}} + T_{\mathrm{s}}$. This distinctive feature stands as the primary advantage of the proposed approach over traditional methods that involve deriving the nominal trajectory first and subsequently searching for a stabilizing control. The efficacy of the proposed approach is anticipated to enhance, particularly in scenarios where the system dynamics exhibit heightened sensitivity to external disturbances, such as strong nonlinearities or rapidly time-varying properties.

The low-thrust ROCP is thus formulated as

$$\min_{\boldsymbol{\Theta}} \text{ Eq. (6.26)} \tag{6.27a}$$

$$\text{s.t. } \text{ Eqs. (6.17)} - \text{(6.22)}, \text{(6.25)} \tag{6.27b}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{N_\Theta}$ is the vectore of problem unknowns, that is composed by

$$N_\Theta = N[n + (n-1)/2 + (n-1)^2/2] = N(n^2 + 3n/2) \tag{6.28}$$

optimization variables, which encompasses the n elements of $\boldsymbol{\mu}_k$, the $(n-1)/2$ components of the deterministic control $\boldsymbol{S}_k$ and the $(n-1)^2/2$ elements of the gain matrix $\boldsymbol{K}_k$. The vector is thus written as

$$\boldsymbol{\Theta} = \{\boldsymbol{\mu}_k,\, \boldsymbol{S}_k,\, \boldsymbol{K}_k,\, \forall k = 1, \ldots, N\} \tag{6.29}$$

having that $\boldsymbol{\mu}_k$ for $k = N$ is fixed to the assigned final mean state as in Eq. (6.20).

6.2 Case study: Earth–Mars transfer

In this section, the application of the proposed low-thrust methodology is explored by considering a time-fixed heliocentric single-leg Earth–Mars transfer in Keplerian motion, which is an established case study employed in analogous works [120, 102]. The choice of this mission serves to evaluate and demonstrate the effectiveness of the proposed approach within a familiar benchmark scenario.

6.2.1 Problem data

Considering the ecliptic J2000 RF centered on the Sun, the initial mean state $\widetilde{\boldsymbol{\mu}}_0$ is defined based on the Earth's position and velocity at the initial time, with the assumption of zero excess hyperbolic velocity. For the continuous thrust heliocentric Keplerian motion, the deterministic EoM are simply

$$\mathrm{d}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{v}(t) \\ -\mu_\odot \boldsymbol{r}(t)/r^3 + \boldsymbol{T}_k/m(t) \\ -\|\boldsymbol{T}_k\|/\left(g_0 I_{\mathrm{sp}}\right) \end{bmatrix} \mathrm{d}t \qquad \forall t \in [t_{k-1},\, t_k] \qquad \forall k = 1, \ldots, N \tag{6.30}$$

where g_0 is the standard acceleration of gravity and I_{sp} is the specific impulse of the engine. The state covariance at departure is defined as

$$\widetilde{\boldsymbol{P}}_0 = \mathrm{diag}\left[(\sigma_0^r)^2\, \boldsymbol{I}_3,\, (\sigma_0^v)^2\, \boldsymbol{I}_3,\, (\sigma_0^m)^2\right] \tag{6.31}$$

where σ_0^r, σ_0^v and σ_0^m are the user-defined parameters for initial position, velocity and mass standard deviation respectively.

The terminal dispersion is constrained in accordance with Eqs. (5.16) and (5.17), contingent upon establishing the maximum standard deviations for position and velocity, denoted as σ_N^r and σ_N^v. Essential problem parameters, including the confidence level β, the number of arcs N, and the maximum thrust magnitude $T_{\max}$, are detailed in Table 6.1. The initial dispersion for position and velocity are intentionally set as relatively large. This deliberate choice aims to demonstrate the efficacy of the method in ensuring the satisfaction of severe final constraints even when faced with significant initial inaccuracies. In contrast, the initial standard deviation of the mass is deliberately chosen to be substantially null. This decision is based on the recognition that the initial mass of a spacecraft is a well-known and accurately determined quantity at launch. The disturbance term for the mass, denoted as σ_w^m, is assumed to be null in this specific application. However, it can be set as different from zero in scenarios where a source of uncertainty in the mass evolution exists, such as variations in the specific impulse of the engine over time.

All numerical computations were made non-dimensional by scaling lengths, velocities, and times with respect to $r^{\mathrm{conv}} = \|\widetilde{\boldsymbol{r}}_0\|$, $v^{\mathrm{conv}} = \|\widetilde{\boldsymbol{v}}_0\|$, and $t^{\mathrm{conv}} = r^{\mathrm{conv}}/v^{\mathrm{conv}}$.

The open-source large-scale sparse nonlinear optimizer WORHP [202] was used for solving the optimization problem.

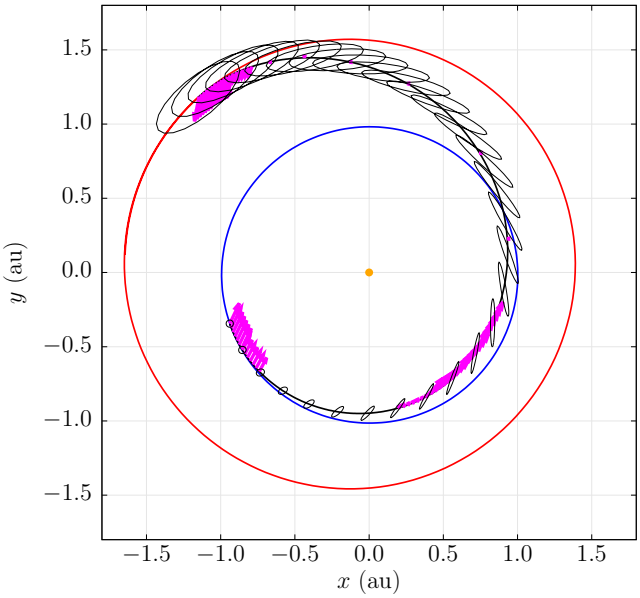
Table 6.1. Problem data.

Variable	Value	Unit
t_f	358.790	days
$\widetilde{T}_{\max}$	1	N
β	5%	-
N	30	-
I_{sp}	2000	s
$\mu_\odot$	132712440018	kg^3/s^2
$\widetilde{\boldsymbol{r}}_0$	$[-140699693, -51614428, 980]^T$	km
$\widetilde{\boldsymbol{v}}_0$	$[9.775, -28.078, 4.338 \times 10^{-4}]^T$	km/s
$\widetilde{m}_0$	1000	kg
$\widetilde{\boldsymbol{r}}_N$	$[-172682023, 176959469, 7948912]^T$	km
$\widetilde{\boldsymbol{v}}_N$	$[-16.427, -14.861, 9.215 \times 10^{-2}]^T$	km/s
σ_0^r	1.499×10^6	km
σ_0^v	94.102	m/s
σ_0^m	1.000×10^{-5}	kg
σ_N^r	1.499×10^5	km
σ_N^v	9.410	m/s
σ_w^r	3.351×10^6	km
σ_w^v	210.419	m/s
σ_w^m	0.000	kg

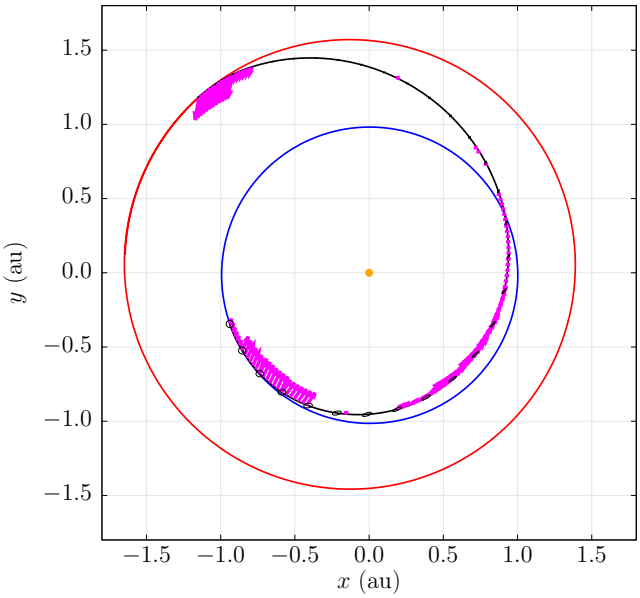
6.2.2 Numerical results

Figure 6.1 visually depicts the trajectory and covariance ellipsoids on position for both the deterministic OCP solution (Fig. 6.1a) and the CL solution (Fig. 6.1b) in the x-y plane. The OCP solution, similarly to test case A (see Sec. 5.4), is obtained forcing the elements of $\boldsymbol{K}_k$ to zero and removing the constraints on the final covariance of the state. Once the OCP solution is computed, the state uncertainty along the trajectory is estimated separately with the modified UT. It is appearent that the CL control effectively reduces the dispersion of the state along the trajectory, especially at the arrival point. The numerical results for both solutions are summarized in Table 5.7, providing insights into control efforts, including the final average mass (μ_N^m), and the $\pm 3\sigma$ final mass ($m_{3\sigma+}$ and $m_{3\sigma-}$). Additionally, the mean terminal errors on mean in-plane position and velocity (Δr_N^x, Δr_N^y, Δv_N^x, and Δv_N^y) and the diagonal terms of the final covariance matrix (σ_N^x, σ_N^y, $\sigma_N^{v_x}$, $\sigma_N^{v_y}$, and σ_N^m) for the in-plane components are reported. The augmented robustness offered by the CL control comes at the expense of approximately 10 kg of extra propellant on average and up to about 80 kg for the worst case. These observations hold true for a 95% confidence level, considering the perturbation magnitude detailed in Table.

To validate the accuracy of the estimated probability distribution at each step for both the CL and OCP cases, an MC analysis comprising 100 000 independent runs was conducted. Starting with a random initial state $\boldsymbol{x}_0 \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}_0, \tilde{\boldsymbol{P}}_0)$, each trajectory was propagated towards the final state using the feedback control law from Eq. (6.2) for the CL case and solely the deterministic component $\boldsymbol{S}_k$ for the OCP case. The propagation of the SDEs in Eq. (6.3) was executed using the Stochastic Fourth-order Runge-Kutta method, as discussed in the work of N. J. Kasdin [85]. The numerical results, summarized in Table 6.2, facilitate a comprehensive comparison between



(a) OCP solution.



(b) CL solution.

Figure 6.1. Nominal spacecraft trajectories with 95%-confidence position ellipses.

the MC solutions and the solution of the ROCP presented in Eqs. (6.27). It is noteworthy that all MC values align well with the ROCP solution, affirming the robustness and reliability of the proposed methodology. In particular, the variances at final time are in good agreement with the $\boldsymbol{P}$ matrices of the OCP solution, and the same consideration applies to the mean error on the terminal state. An inaccuracy for the v_y standard deviation of about 50% is however noticed in the CL case. The trajectories from the MC campaign solutions are illustrated in Fig. 6.2, revealing a commendable alignment with the covariance ellipsoids obtained by solving the

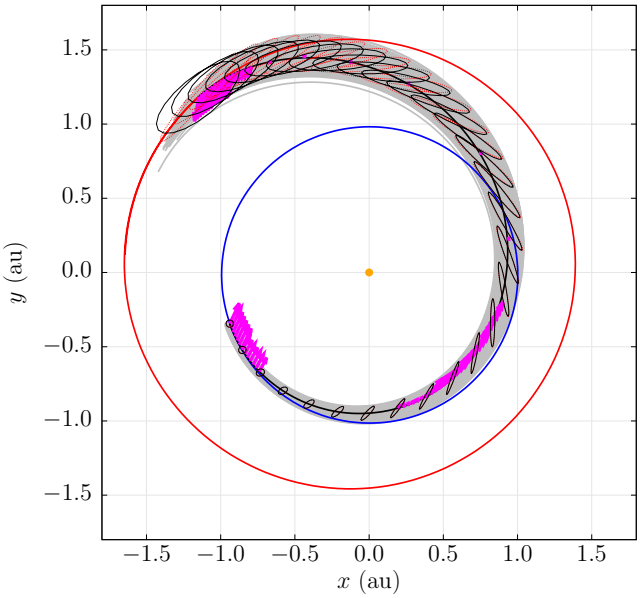
Table 6.2. Comparison between the solution of the ROCP and the related MC analysis in the OCP and CL cases.

Quantity	OCP	OCP (MC)	ROCP-CL	ROCP-CL (MC)	Unit
μ_N^m	628.165	628.163	612.071	611.170	kg
$m_{3\sigma+}$	628.165*	628.193*	640.172*	642.011*	kg
$m_{3\sigma-}$	628.164*	628.133*	583.969*	580.328*	kg
σ_N^m	1.000×10^{-5}	1.002×10^{-5}	9.367	10.280	kg
Δr_N^x	-	6.191×10^5	-	359.112	km
Δr_N^y	-	1.100×10^5	-	887.948	km
Δv_N^x	-	53.618	-	0.503	m/s
Δv_N^y	-	54.801	-	1.030	m/s
σ_N^x	1.720×10^7	1.595×10^7	1.304×10^5	1.332×10^5	km
σ_N^y	1.491×10^7	1.843×10^7	1.440×10^5	1.446×10^5	km
$\sigma_N^{v_x}$	905.970	1.376×10^3	8.292	10.392	m/s
$\sigma_N^{v_y}$	1.411×10^3	1.713×10^3	9.390	12.176	m/s

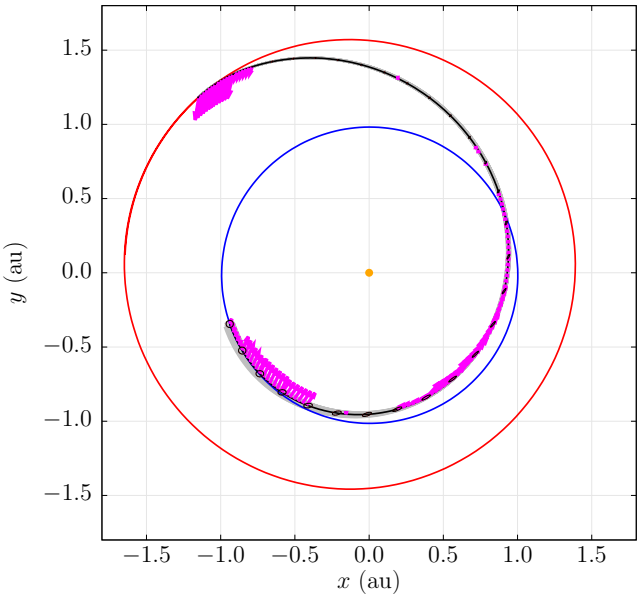
* These values are computed as $\mu_N^m \pm 3\sigma_N^m$.

stochastic optimal control problem in Eqs. (6.27). This visual comparison underscores the consistency and accuracy of the proposed approach in capturing the stochastic behavior of the system. The covariance matrices $\boldsymbol{P}_k$ empirically computed in the MC analysis are compared to those obtained during the robust optimization that uses the modified UT tool for propagating the state uncertainty. The diagonal terms of either matrix are compared in Figs. 6.3 and 6.4 for the OCP and CL solutions. According to the attained results, UT succeeds in propagating the state distribution very accurately in the CL case, where the displacement from the nominal trajectory is kept very small and thus the state distribution is well approximated by a Gaussian one. On the other hand, in the OCP case, the MC trajectories may significantly deviate from the nominal solution, especially at the end of the transfer, and the estimates of covariance matrix are slightly less accurate. This discrepancy is also evident in Fig. 6.2a, where the position ellipses computed in the MC analysis exhibit significant divergence from the estimated ellipses indicated by the black dotted lines. This is due to the fact that, without a control term that limits the state dispersion at each maneuver, the probability distribution progressively spreads up and deviates from a Gaussian distribution, reducing the accuracy in propagating the state distribution. Considerations for the out-of-plane dimension are omitted, given that the trajectory is largely aligned with the ecliptic plane, and the distribution along the z axis can be easily tracked.

The empirically computed covariance matrices $\boldsymbol{P}_k$ utilizing the MC approach are subsequently juxtaposed with those derived during the optimization process employing the modified UT for state uncertainty propagation. The comparison is particularly focused on the diagonal terms of both matrices, as illustrated in Figs. 6.3 and 6.4 for both the OCP and CL solutions. The results highlight the effectiveness of the adopted technique in accurately propagating the state distribution, especially in the CL case where deviations from the nominal trajectory are minimal, resulting in a well-approximated Gaussian state distribution. Conversely, in the OCP case, the MC trajectories exhibit substantial deviation from the nominal solution, particularly towards the end of the transfer. Consequently, the estimates of the covariance matrix are less accurate in this scenario. Figure 6.5 presents the norm of the deterministic component of the control $\boldsymbol{S}_k$, along with the spacecraft mass over



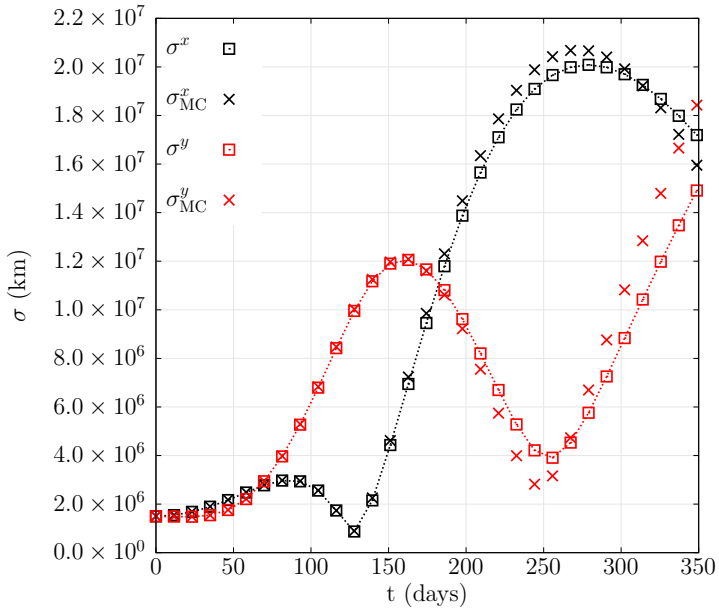
(a) OCP solution.



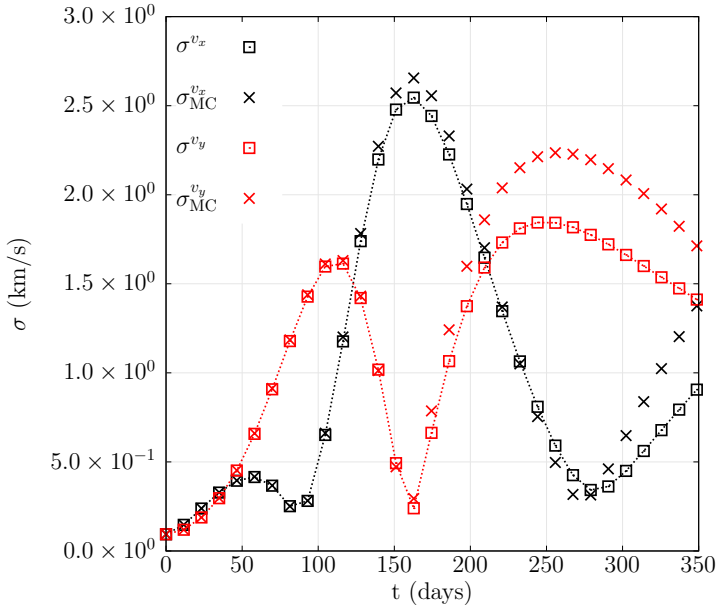
(b) CL solution.

Figure 6.2. MC trajectories (in gray) with corresponding position ellipses (dashed red line) and UT position ellipses (solid black line).

time $m(t)$, normalized with respect to the initial mass $m(t_0)$. The trend in the (normalized) average mass $\mu^m/\tilde{\mu}_0^m$ is also depicted. The grey lines represent the mass evolution for each MC run, plotted alongside the average mass computed from the MC and the related 3σ error bars, revealing a good match with the estimated average mass. These plots provide insights into the behavior of the control system and the evolution of the spacecraft's mass over time. Figure 6.5b highlights a crucial observation concerning the deterministic control in the CL solution. In contrast to the



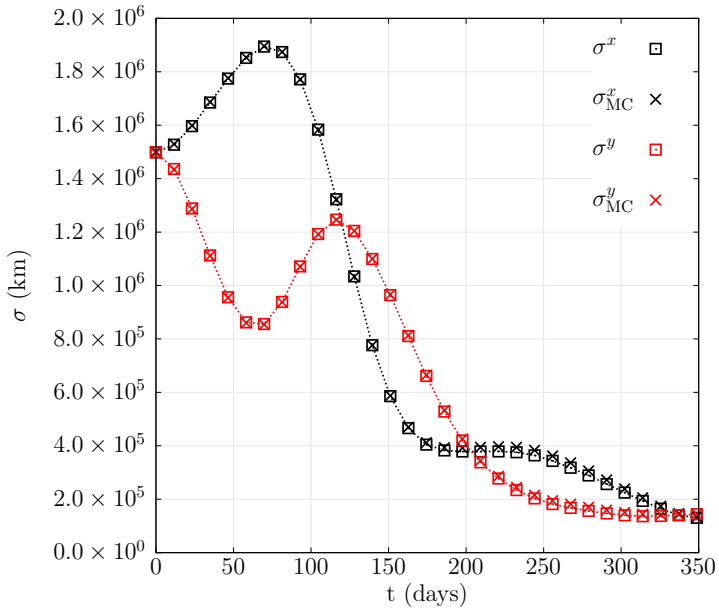
(a) OCP, in-plane position.



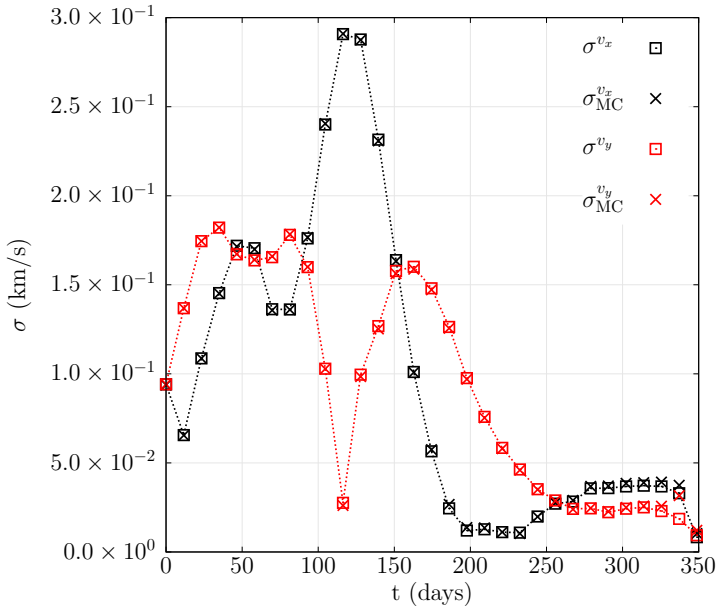
(b) OCP, in-plane velocity.

Figure 6.3. Covariance matrices trace comparison of the OCP case.

OCP case, the deterministic portion of the thrust norm does not reach the maximum admissible value $\widetilde{T}_{\max}$. The nominal control design ensures the preservation of a margin for necessary corrective actions to meet the final covariance constraint. By incorporating this margin, the system is better equipped to handle uncertainties and variations, thereby enhancing the overall robustness of the trajectory. The margin is not uniformly distributed over all burn arcs, as it would be in the case of a sub-optimal, empirical design. Instead, it accommodates expected uncertainties as well as the actual need for corrections.



(a) CL, in-plane position.

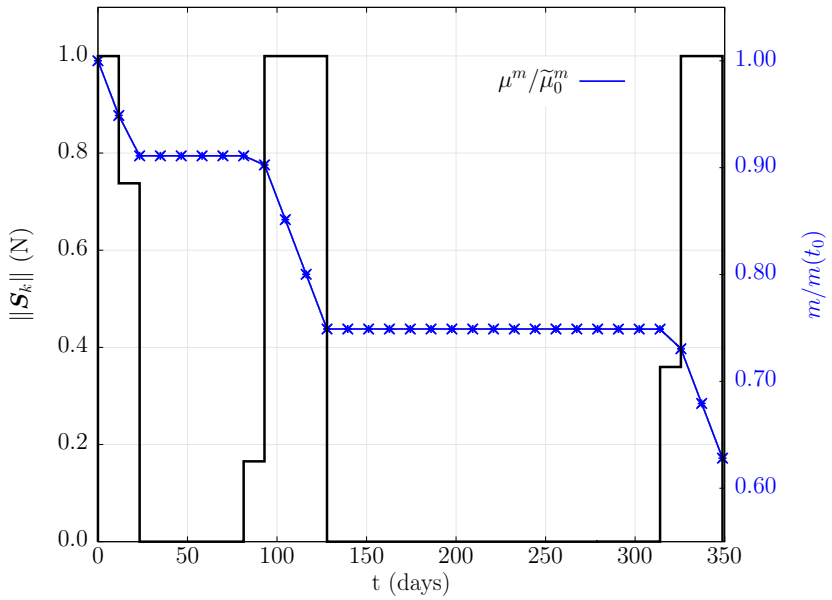


(b) CL, in-plane velocity.

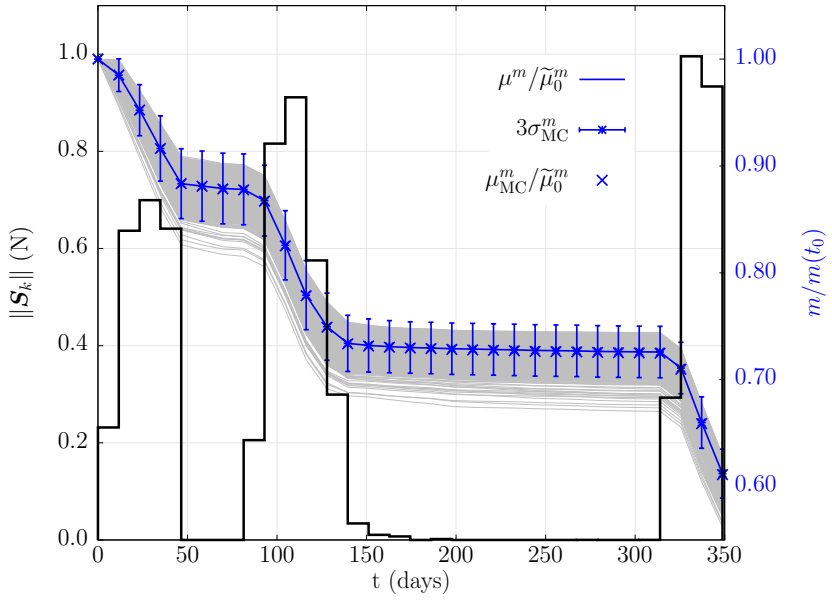
Figure 6.4. Covariance matrices trace comparison of the CL case.

An MC analysis of the thrust magnitude $\|\boldsymbol{T}_k\|$ is visually presented in Figure 6.6, where the grey lines represent the thrust profiles for each MC run, and the dotted red line is the $\widetilde{T}_{\max}$ threshold. Notably, the thrust magnitude is consistently maintained below the maximum allowed limit of 1 N. However, due to the chance-constrained formulation of this bound, there is a possibility of violation at a rate of β. Nevertheless, as observed in the figure, no violation is evident.

Figure 6.7 shows the comparison of the standard deviation of mass σ^m between the MC value and the value estimated within the ROCP. The graph reveals a



(a) OCP solution.



(b) CL solution.

Figure 6.5. Norm of the deterministic thrust and average mass evolution along trajectory.

maximum difference on the order of 1 kg. This discrepancy arises from the final distribution of the mass at the end of each arc, which slightly deviates from a normal Gaussian distribution, which is the probability distribution best-tracked by UT. Note that, the mass distribution exhibits a subtle negative skewness with respect to the mean value, as evident in Fig. 6.5b. The MC values tend to spread wider when below the mean mass rather than above, indicating this negative skewness. This skewness is a consequence of a slight positive skewness of the thrust magnitude with respect to $\|\boldsymbol{S}_k\|$, as the stochastic term $\boldsymbol{K}_k\left(\boldsymbol{x}'_{k-1} - \boldsymbol{\mu}'_{k-1}\right)$ generally increases the

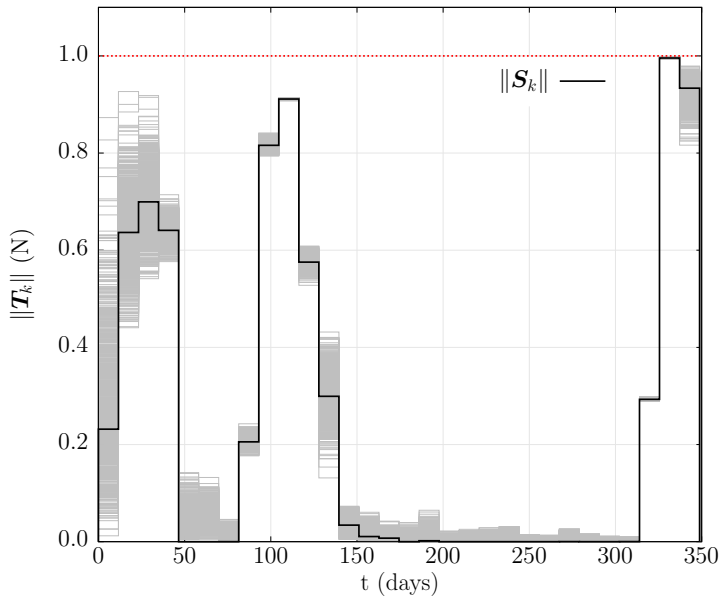


Figure 6.6. MC analysis of the thrust magnitude.

thrust magnitude $\|\boldsymbol{T}_k\|$. Consequently, a greater thrust magnitude corresponds to a greater mass consumption. This deviation from a normal Gaussian distribution is likely to be the root cause of the aforementioned discrepancy in σ_m.

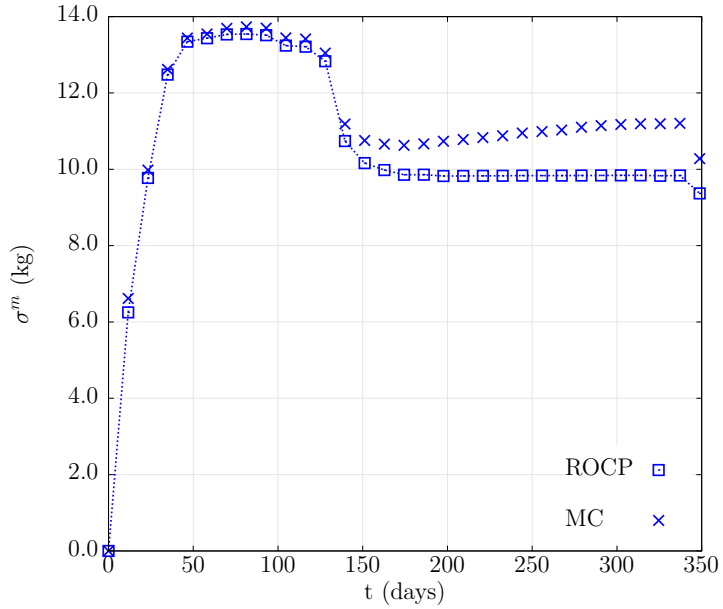


Figure 6.7. Standard deviation of mass comparison for the CL case.

Chapter 7

Conclusions

This thesis investigated methodologies of robust optimal control problems for space trajectory design applications. Two categories of methods were considered: Open-Loop (OL) or feedforward robust control and Closed-Loop (CL) or feedback robust control. The first approach consists in an optimized deterministic control law to be imparted on the spacecraft, without any consideration of the ongoing trajectory. The latter is a control law that depends on the estimated state of the spacecraft during its flight and thus takes into account the actual deviation from the nominal trajectory. In CL robust design, the parameters of the control law required to compute the maneuvers on-board are optimized *a priori*, rather than optimizing directly the maneuvers as in the OL case.

An OL methodology has been proposed for the problem of an impulsive transfer from a given initial state to a target state. Here the optimization variables are the impulsive control vectors and the duration of the ballistic arcs between the maneuvers. Once a Gaussian distribution of the initial state is defined, the goal of the optimization is to minimize the state dispersion at the arrival, quantified with the trace of the final covariance matrix (although other formulations are possible), subject to a maximum threshold for the magnitude of the single control vectors, total magnitude, and total ToF. Specifically, the maximum total control magnitude can be selected by adding an adjustable margin to the control effort derived from an equivalent deterministic OCP; this latter, having as cost function as the total control magnitude and identical constraints on single control magnitudes and total ToF. The unscented transform was adopted to propagate the state uncertainty through the ballistic arcs. As test case, an impulsive rendezvous was considered, showing the ability of the OL robust control to roughly halve the final state dispersion with a marginal increase in fuel consumption.

Next, a CL robust optimization algorithm for impulsive transfers was conceived, formulated, and tested. A linear feedback control law was adopted with the aim of countering the state dispersion during the flight, thus having as input the real-time state right before the impulsive maneuver. Analytical expressions were obtained to compute the effect of the CL impulsive control on the covariance matrix of the state, while the unscented transform was resorted to propagate the state distribution through the ballistic arcs. Stochastic perturbations on the EoM, in order to model inaccurate knowledge of the dynamical system, are introduced via an instantaneous covariance matrix that is added at the end of the arc to the estimated state covariance matrix. Chance-constraint formulations are instead used for the cost function, representing a stochastic estimate of the total control magnitude with a given level of confidence, and a constraint on the control magnitude of the single impulse. Further constraints are enforced on the state uncertainty for critical epochs of the mission

(e.g., arrival or close encounter with an asteroid) via a maximum threshold for the position and velocity eigenvalues of the covariance matrix (although alternative constraints on the state dispersion are possible). Two distinct test cases were presented in this regard. The first concerned an Earth–asteroid–Earth trajectory in the context of the future JAXA mission DESTINY+, whose primary goal is the encounter with the Geminids meteor shower parent body (3200) Phaethon. Results show an effective limitation of the state uncertainty of the mission, with reasonable increments on the total control effort with respect to the optimal deterministic non-robust formulation. The second application consisted of a station-keeping strategy along an NRHO around the Moon, with control efforts compliant with alternative station-keeping strategies available in the literature. Monte Carlo simulations were performed in both cases to confirm the accurate state propagation of the UT along the flight.

The impact of CL impulsive control on the covariance matrix of the state, considering navigation and propulsion errors, was thoroughly examined. These errors are among the most critical sources of inaccuracy in space missions, making a direct understanding of their effects essential for robust optimization. For navigation errors, a simplified model was formulated using a linear mapping from the state with a Gaussian random perturbation. The resulting observables vector is then utilized in the CL control law in place of the state vector. Subsequently, navigation errors with a general nonlinear mapping were addressed, and strategies to estimate the resulting strongly nonlinear terms were suggested. For propulsive errors, a simplified model accounting for inaccuracies in the magnitude and direction of the control was developed to determine their impact on state estimation, which is then incorporated into the CL control.

The CL formulation was extended to continuous thrust trajectories, following a similar approach to the impulsive methodology. The critical difference is the inclusion of the spacecraft mass as an additional element of the state vector. A linear feedback control law is used to formulate the thrust vector, which is assumed to be constant in magnitude and direction in each segment of the trajectory. To propagate the state distribution along the trajectory, a modified version of the UT is developed and implemented in the proposed algorithm. A chance-constraint formulation is also adopted in this low-thrust method to express both the cost function, again a stochastic estimate of the total control effort, and the constraint on the maximum thrust vector magnitude for each segment. A canonical Earth-Mars transfer demonstrated the efficacy of a robust CL control law in mitigating state dispersion during the mission and meeting the robustness requirements of the trajectory with only a marginal increase in fuel consumption. Additionally, an MC analysis confirmed the reasonable accuracy of the uncertainty propagation method used in this context.

In light of the discussed methodologies, the objectives of this thesis are reasonably deemed achieved. The robust methodologies developed for both impulsive and low-thrust trajectories have demonstrated their effectiveness in limiting state dispersion along the trajectory, even under significant dynamical perturbations. These methods have shown versatility across various mission scenarios, largely due to the flexible uncertainty propagation tool, in particular for the impulsive method. Furthermore, for the CL control, since the maneuvers are computed onboard through a single algebraic expression, the computational burden is handled during the optimization process, resulting in minimal computational effort required from the onboard system during mission execution.

7.1 Future works

There are several potential enhancements for the proposed methodologies. A crucial aspect that requires more accurate investigation is the inclusion of the perturbation effect of SDE into the robust optimization algorithm, rather than relying on an additive term applied to the covariance matrix at the end of each propagation arc. For relatively simple trajectories, one approach could be to use a semi-empirical method to quantify this random contribution. This would involve propagating a limited number of samples with a set of SDE and estimating the propagated state dispersion to be considered within the optimization process. Although this approach would significantly improve the accuracy of dispersion tracking, the computational effort would increase exponentially, making it infeasible with limited computational resources. Another technique could involve addressing the specific dynamical uncertainties of the gravitational model considered (e.g., selecting ranges of gravitational parameters) and performing an *a priori* analysis of these uncertainties' effects on the trajectory, rather than considering a general white noise perturbation on the dynamics. This approach would more realistically follow the actual design phase of a trajectory, given that most missions have a reasonable understanding of the dynamical environment, except for small margins on specific parameters (e.g., planetary harmonics).

Gravity assist, which is a fundamental tool of space trajectories, could be also investigated using the robust algorithm discussed in this thesis. The key modification that a flyby would require is the estimate of how the state distribution evolves during the close passage with the selected body. Rapid trajectories, relative to the main body, are generally highly sensitive to numerical propagation, making it challenging to estimate the evolution of state dispersion in such scenarios accurately. Therefore, a precise analysis of how to perform this state estimate, balancing both numerical accuracy and computational effort, is essential. An improvement in this sense would allow the usage of the proposed robust optimization methods and similar ones that rely on the knowledge of the state distribution for complex missions that need one or multiple gravity assists.

Another possible development of this work would be the adjustment of the robust formulation proposed for the continuous-thrust case to the use of indirect optimization methods. Crucial advantage in this regard would be a continuous control law for the propulsion system, rather than assuming a constant thrust vector in each segment. Another immediate benefit would be a more rapid convergence to an optimal solution (assumed a close first guess), which is typical of this class of methods. Several challenges however would be posed by this approach. Firstly, state propagation with continuous thrust would not be trivial to be computed, since control is included in the EoM, as already discussed in the low-thrust chapter. Most importantly, extensive analytical derivations would be necessary to obtain the Euler-Lagrange equations in this case, where also the elements of the gain matrix and a chance-constraint formulation of the cost function are expected. In addition to those, also the common challenges of the indirect methods would pose an issue, like a close estimate of the optimal state and co-state.

Appendix

A Dynamical disturbance model

Considering the linear stochastic system obtained by linearizing Eq. 2.1 along a reference trajectory $(\hat{\boldsymbol{x}}(t),\, \hat{\boldsymbol{u}}(t))$

$$\mathrm{d}\boldsymbol{x} = (\boldsymbol{A}(t)\boldsymbol{x} + \boldsymbol{B}(t)\boldsymbol{u} + \boldsymbol{c}(t))\,\mathrm{d}t + \boldsymbol{G}(t)\mathrm{d}\boldsymbol{w} \tag{A.1}$$

where

$$\boldsymbol{A}(t) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\,(\hat{\boldsymbol{x}}(t),\, \hat{\boldsymbol{u}}(t),\, t) \tag{A.2}$$

$$\boldsymbol{B}(t) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\,(\hat{\boldsymbol{x}}(t),\, \hat{\boldsymbol{u}}(t),\, t) \tag{A.3}$$

$$\boldsymbol{G}(t) = \boldsymbol{g}\,(\hat{\boldsymbol{x}}(t),\, \hat{\boldsymbol{u}}(t),\, t) \tag{A.4}$$

$$\boldsymbol{c}(t) = \boldsymbol{f}\,(\hat{\boldsymbol{x}}(t),\, \hat{\boldsymbol{u}}(t),\, t) - \boldsymbol{A}(t)\boldsymbol{x}(t) - \boldsymbol{B}(t)\boldsymbol{u}(t) \tag{A.5}$$

Assume a time grid discretization $t_0 \leqslant t_1 \leqslant \cdots \leqslant t_N = t_f$, and a piece wise constant control law in each sub-interval, that is

$$\boldsymbol{u}(t) = \boldsymbol{u}(t_k) = \boldsymbol{u}_k \qquad t \in [t_k,\, t_{k+1}) \qquad \forall k = 0, \ldots, N-1 \tag{A.6}$$

Equation (A.1) can be written in the discrete form

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k + \boldsymbol{c}_k + \boldsymbol{G}_k\boldsymbol{w}_k \tag{A.7}$$

with

$$\boldsymbol{A}_k = \boldsymbol{\Phi}(t_{k+1},\, t_k) \tag{A.8}$$

$$\boldsymbol{B}_k = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1},\, s)\boldsymbol{B}(s)\mathrm{d}s \tag{A.9}$$

$$\boldsymbol{c}_k = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1}, s)\boldsymbol{c}(s)\mathrm{d}s \tag{A.10}$$

where $\boldsymbol{\Phi}(t,\, s) = \boldsymbol{A}(t)$ is the state transition matrix and $\boldsymbol{G}_k$ is a matrix so that the stochastic process $\boldsymbol{G}_k\boldsymbol{w}_k$ has covariance $\boldsymbol{Q}_k$, evaluated as

$$\boldsymbol{Q}_k = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1},\, s)\boldsymbol{G}(s)\boldsymbol{G}(s)^T\boldsymbol{\Phi}(t_{k+1},\, s)^T\mathrm{d}s \tag{A.11}$$

Recalling that $\boldsymbol{\Phi}(s,\, t_{k+1}) = \boldsymbol{\Phi}(s,\, t_k)\boldsymbol{\Phi}(t_k,\, t_{k+1})$ and $\boldsymbol{\Phi}(t_k,\, s) = \boldsymbol{\Phi}(s,\, t_k)^{-1}$, the previous equation can be rewritten as

$$\boldsymbol{Q}_k = A_k \left(\int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}^{-1}\,(s,\, t_k)\,\boldsymbol{G}(s)\boldsymbol{G}^T(s)\left(\boldsymbol{\Phi}^{-1}(s,\, t_k)\right)^T \mathrm{d}s \right) A_k^T \tag{A.12}$$

In the present manuscript, no time-continuous control is present, so the term $\boldsymbol{B}_k$ and $\boldsymbol{u}_k$ can be neglected. As reference trajectory to evaluate $\boldsymbol{Q}_k$, the OCP solution of the deterministic optimization problem is chosen. For simplicity, we define the average perturbation effect on the covariance matrix $\overline{\boldsymbol{Q}}$, calculated as

$$\overline{\boldsymbol{Q}} = \frac{1}{N} \sum_{k=1}^{N} \boldsymbol{Q}_k \tag{A.13}$$

B CR3BP dynamics

Consider a body P_3 of mass m_3 in the vector field of a larger primary P_1, and a smaller primary P_2, of masses m_1 and m_2, respectively, such that the condition $m_3 \ll m_2 < m_1$ is satisfied. The primaries revolve in planar configuration at constant angular speed. The motion of the third body, or particle, is studied in a rotating synodic RF, whose origin is located at the primaries center of mass, the x axis is always aligned with the P_1–P_2 direction, the z axis is orthogonal to their plane of motion, and the y axis forms a right-hand tern. By selecting proper non-dimensional units [13], the equations of motion depend only on the mass parameter, defined as $\mu = m_2/(m_1 + m_2)$. The non-dimensional values are such that the distance between the primaries, their angular speed, and the sum of their masses are set to an unity value. The equations of motion of P_3 are

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} v_x \\ v_y \\ v_z \\ 2v_y + U_{/x} \\ -2v_x + U_{/y} \\ U_{/z} \end{bmatrix} \tag{B.1}$$

where $\boldsymbol{x} = [x,\, y,\, z,\, v_x,\, v_y,\, v_z]^T$ is the P_3 state vector, dots denote derivatives with respect to non-dimensional time, and slashed subscripts are partial derivatives. The three-body pseudo-potential U is

$$U = \frac{1}{2}\left(x^2 + y^2 + z^2\right) + \frac{1-\mu}{r_1} + \frac{\mu}{r_2} + \frac{\mu(1-\mu)}{2} \tag{B.2}$$

where terms $r_1 = \sqrt{(x+\mu)^2 + y^2 + z^2}$ and $r_2 = \sqrt{(x-1+\mu)^2 + y^2 + z^2}$ are the distances between P_3 and each primary, respectively.

Bibliography

[1] "US and soviet space programs: Comparative size," tech. rep., Central Intelligence Agency, USA, 1966. https://irp.fas.org/cia/product/sovmim66.pdf.

[2] M. D. Rayman, T. C. Fraschetti, C. A. Raymond, and C. T. Russell, "Coupling of system resource margins through the use of electric propulsion: implications in preparing for the dawn mission to ceres and vesta," *Acta Astronautica*, vol. 60, no. 10, pp. 930–938, 2007.

[3] D. Oh, D. Landau, T. Randolph, P. Timmerman, J. Chase, J. Sims, and T. Kowalkowski, "Analysis of system margins on deep space missions using solar electric propulsion," in *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, 2012.

[4] I. Alizadeh and B. F. Villac, "Targeting requirements and stability characterization for a class of ballistic transfers," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 5, pp. 1473–1489, 2010.

[5] F. E. Laipert and J. M. Longuski, "Automated missed-thrust propellant margin analysis for low-thrust trajectories," *Journal of Spacecraft and Rockets*, vol. 52, no. 4, pp. 1135–1143, 2015.

[6] "Trajectories in the Earth-Moon space with symmetrical free return," tech. rep., NASA, USA, 1963. https://babel.hathitrust.org/cgi/pt?id=uiug.30112106596973&view=1up&seq=1.

[7] I. O'Neill, "Safe mode slows Dawn mission's progress to ceres," *Discover News*, Sept. 2014. https://www.space.com/27206-safe-mode-slows-dawn-mission-ceres.html.

[8] E. A. Belbruno and J. K. Miller, "Sun-perturbed earth-to-moon transfers with ballistic capture," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 4, pp. 770–775, 1993.

[9] A. Poghosyan and A. Golkar, "Cubesat evolution: analyzing cubesat capabilities for conducting science missions," *Progress in Aerospace Sciences*, vol. 88, pp. 59–83, 2017.

[10] R. Walker, D. Binns, C. Bramanti, M. Casasco, P. Concari, D. Izzo, D. Feili, P. Fernandez, J. G. Fernandez, P. Hager, D. Koschny, V. Pesquita, N. Wallace, I. Carnelli, M. Khan, M. Scoubeau, and D. Taubert, "Deep-space CubeSats: thinking inside the box," *Astronomy & Geophysics*, vol. 59, pp. 5.24–5.30, Oct. 2018.

[11] E. Turan, S. Speretta, and E. Gill, "Autonomous navigation for deep space small satellites: scientific and technological advances," *Acta Astronautica*, vol. 193, pp. 56–74, 2022.

[12] V. Franzese, F. Topputo, F. Ankersen, and R. Walker, "Deep-space optical navigation for M-ARGO mission," *The Journal of the Astronautical Sciences*, vol. 68, pp. 1034–1055, Dec. 2021.

[13] V. Szebehely, *Theory of orbits: the restricted problem of three bodies*. New York: Academic press, 1967.

[14] Y.-z. Luo and Z. Yang, "A review of uncertainty propagation in orbital mechanics," *Progress in Aerospace Sciences*, vol. 89, pp. 23–39, Feb. 2017.

[15] Y. Kawakatsu, M. Yamamoto, and J. Kawaguchi, "Study on a lunar approach strategy tolerant of a lunar orbit injection failure," *Transactions of the Japan society for aeronautical and space sciences*, vol. 5, pp. 1–7, 2007.

[16] S. Takahashi, N. Ogawa, and Y. Kawakatsu, "General characteristics of free-return ensured orbit insertion and trajectory design with MOI robustness in MMX mission," *Transactions of the Japan society for aeronautical and space sciences*, vol. 17, no. 3, pp. 404–411, 2019.

[17] N. Ogawa, Y. Tsuda, Y. Takei, H. Inoue, S. Takahashi, and Y. Kawakatsu, "Orbit design for the Martian Moons Exploration mission," *Transactions of the Japan society for aeronautical and space sciences*, vol. 17, no. 3, pp. 398–403, 2019.

[18] N. Ozaki, T. Chikazawa, K. Kakihara, A. Ishikawa, and Y. Kawakatsu, "Extended robust planetary orbit insertion method under probabilistic uncertainties," *Journal of Spacecraft and Rockets*, vol. 57, no. 6, pp. 1153–1164, 2020.

[19] A. Rubinsztejn, C. G. Sandel, R. Sood, and F. E. Laipert, "Designing trajectories resilient to missed thrust events using expected thrust fraction," *Aerospace Science and Technology*, vol. 115, p. 106780, 2021.

[20] A. Rubinsztejn, R. Sood, and F. Laipert, "Expected thrust fraction: resilient trajectory design applied to the Earth return orbiter," in *31st AAS/AIAA Space Flight Mechanics Meeting*, Feb. 2021.

[21] P. Di Lizia, R. Armellin, F. Bernelli-Zazzera, and M. Berz, "High order optimal control of space trajectories with uncertain boundary conditions," *Acta Astronautica*, vol. 93, pp. 217–229, 2014.

[22] P. Di Lizia, R. Armellin, E. Finzi, and M. Berz, "High-order robust guidance of interplanetary trajectories based on differential algebra," *Journal of Aerospace Engineering, Sciences and Applications*, vol. 1, pp. 43–57, Jan. 2008.

[23] P. Di Lizia, R. Armellin, A. Morselli, and F. Bernelli-Zazzera, "High order optimal feedback control of space trajectories with bounded control," *Acta Astronautica*, vol. 94, no. 1, pp. 383–394, 2014.

[24] R. Armellin, P. Di Lizia, F. Topputo, M. Lavagna, F. Bernelli-Zazzera, and M. Berz, "Gravity assist space pruning based on differential algebra," *Celestial Mechanics and Dynamical Astronomy*, vol. 106, pp. 1573–1584, 2010.

[25] P. W. Schumacher, C. Sabol, C. C. Higginson, and K. T. Alfriend, "Uncertain Lambert problem," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 9, pp. 1573–1584, 2015.

[26] G. Zhang, D. Zhou, D. Mortari, and M. R. Akella, "Covariance analysis of lambert's problem via lagrange's transfer-time formulation," *Aerospace Science and Technology*, vol. 77, pp. 765–773, 2018.

[27] N. Adurthi and M. Majji, "Uncertain lambert problem: A probabilistic approach," *The Journal of the Astronautical Sciences*, vol. 67, no. 2, pp. 361–386, 2020.

[28] H. yang Li, Y.-Z. Luo, Jin-Zhang, and G.-J. Tang, "Optimal multi-objective linearized impulsive rendezvous under uncertainty," *Acta Astronautica*, vol. 66, no. 3, pp. 439–445, 2010.

[29] Z. Yang, Y.-z. Luo, and J. Zhang, "Robust planning of nonlinear rendezvous with uncertainty," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 8, pp. 1954–1967, 2017.

[30] G. Deaconu, C. Louembet, and A. Théron, "Minimizing the effects of navigation uncertainties on the spacecraft rendezvous precision," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 2, pp. 695–700, 2014.

[31] C. Louembet, D. Arzelier, and G. Deaconu, "Robust rendezvous planning under maneuver execution errors," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 1, pp. 76–93, 2015.

[32] H. H. Goldstine, *A History of the Calculus of Variations from the 17th through the 19th Century*. Springer-Verlag, 1980.

[33] J. M. Longuski, J. Guzmán, and J. E. Prussing, *Optimal control with aerospace applications*. Microcosm press, El Segundo (CA), 2014.

[34] A. Gambier and E. Badreddin, "Multi-objective optimal control: An overview," in *2007 IEEE International Conference on Control Applications*, pp. 170–175, 2007.

[35] J. K. Mandal, S. Mukhopadhyay, and P. Dutta, *Multi-Objective Optimization*. Springer Singapore, 2018.

[36] A. E. B. Jr. and Y. chi Ho, *Applied optimal control*. Hemisphere Publishing, Washington D.C., 1975.

[37] L. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mishchenko, *The mathematical theory of optimal processes*. New York: Wiley, 1962.

[38] R. D. Russell and L. F. Shampine, "A collocation method for boundary value problems," *Numerische Mathematik*, vol. 19, no. 1, pp. 1–28, 1972.

[39] J. T. Betts and W. P. Huffman, "Path-constrained trajectory optimization using sparse sequential quadratic programming," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 1, pp. 59–68, 1993.

[40] C. Hargraves and S. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338–342, 1987.

[41] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Journal on Optimization*, vol. 12, no. 4, pp. 979–1006, 2002.

[42] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[43] M. A. Patterson and A. V. Rao, "GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Trans. Math. Softw.*, vol. 41, Oct. 2014.

[44] M. Knauer and C. Büskens, *Real-Time Optimal Control Using TransWORHP and WORHP Zen*, pp. 211–232. Cham: Springer International Publishing, 2019.

[45] X. Liu, P. Lu, and B. Pan, "Survey of convex optimization for aerospace applications," *Astrodynamics*, vol. 1, no. 1, pp. 23–40, 2017.

[46] L. Blackmore, B. Açikmeşe, and D. P. Scharf, "Minimum-landing-error powered-descent guidance for mars landing using convex optimization," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 4, pp. 1161–1171, 2010.

[47] L. Blackmore, "Autonomous precision landing of space rockets," in *Frontiers of engineering: reports on leading-edge engineering from the 2016 symposium*, vol. 46, (The Bridge Washington, DC), pp. 15–20, Jan. 2016.

[48] B. Açıkmeşe and L. Blackmore, "Lossless convexification of a class of optimal control problems with non-convex control constraints," *Automatica*, vol. 47, no. 2, pp. 341–347, 2011.

[49] Y. Mao, M. Szmuk, and B. Açıkmeşe, "Successive convexification of non-convex optimal control problems and its convergence properties," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 3636–3641, 2016.

[50] R. Bonalli, A. Cauligi, A. Bylard, T. Lew, and M. Pavone, "Trajectory optimization on manifolds: A theoretically-guaranteed embedded sequential convex programming approach," in *Proceedings of Robotics: Science and Systems*, June 2019.

[51] D. Izzo, M. Märtens, and B. Pan, "A survey on artificial intelligence trends in spacecraft guidance dynamics and control," *Astrodynamics*, vol. 3, no. 4, pp. 287–299, 2019.

[52] A. Zavoli, L. Federici, B. Benedikter, and G. Colasurdo, "Comparative analysis of genetic crossover operators for the optimization of impulsive multi-rendezvous trajectories," in *AIDAA 2019 - XXV International Congress*, (Rome, Italy), Sept. 2019.

[53] L. Federici, B. Benedikter, and A. Zavoli, "EOS: a parallel, self-adaptive, multi-population evolutionary algorithm for constrained global optimization," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–10, 2020.

[54] M. Pontani and B. A. Conway, "Particle swarm optimization applied to space trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 5, pp. 1429–1441, 2010.

[55] G. Radice and G. Olmo, "Ant colony algorithms for two impluse interplanetary trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1440–1444, 2006.

[56] F. Campelo and C. Aranha, "EC Bestiary: A bestiary of evolutionary, swarm and other metaphor-based algorithms," June 2018.

[57] D. Zona, L. Federici, and A. Zavoli, "Preliminary design of multi-chaser active debris removal missions with evolutionary algorithms," in *AAS/AIAA Astrodynamics Specialist Conference*, Aug. 2021.

[58] L. Federici, A. Zavoli, and G. Colasurdo, "Evolutionary optimization of multi-rendezvous impulsive trajectories," *International Journal of Aerospace Engineering*, vol. 2021, p. 9921555, 2021.

[59] D. Izzo and M. Märtens, "The Kessler run: on the design of the GTOC9 challenge," *Acta Futura*, Jan. 2018.

[60] L. Federici, A. Zavoli, G. Colasurdo, and L. L. Vahala, "Preliminary capture trajectory design for europa tomography probe," *International Journal of Aerospace Engineering*, vol. 2018, p. 6890173, 2018.

[61] M. Ceriotti and M. Vasile, "Mga trajectory planning with an aco-inspired algorithm," *Acta Astronautica*, vol. 67, no. 9, pp. 1202–1217, 2010.

[62] L. Federici, A. Zavoli, G. Colasurdo, L. Mancini, and A. Neri, "Integrated optimization of first-stage srm and ascent trajectory of multistage launch vehicles," *Journal of Spacecraft and Rockets*, vol. 58, no. 3, pp. 786–797, 2021.

[63] C. Wilt, J. Thayer, and W. Ruml, "A comparison of greedy search algorithms," in *Third annual symposium on combinatorial search*, July 2010.

[64] H. A. Pierson and M. S. Gashler, "Deep learning in robotics: A review of recent research," 2017.

[65] D. Izzo, E. Öztürk, and M. Märtens, "Interplanetary transfers via deep representations of the optimal policy and/or of the value function," GECCO '19, (New York, NY, USA), p. 1971–1979, Association for Computing Machinery, 2019.

[66] C. Sánchez-Sánchez and D. Izzo, "Real-time optimal control via deep neural networks: Study on landing problems," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 5, pp. 1122–1135, 2018.

[67] L. Cheng, Z. Wang, Y. Song, and F. Jiang, "Real-time optimal control for irregular asteroid landings using deep neural networks," *Acta Astronautica*, vol. 170, pp. 66 – 79, 2020.

[68] Y. Shi and Z. Wang, "A deep learning-based approach to real-time trajectory optimization for hypersonic vehicles," in *AIAA Scitech 2020 Forum*.

[69] Y. Shi and Z. Wang, "Onboard generation of optimal trajectories for hypersonic vehicles using deep learning," *Journal of Spacecraft and Rockets*, vol. 58, no. 2, pp. 400–414, 2021.

[70] J. Ortiz, M. J. Grant, K. McCoy, M. Sparapany, R. Ghanem, H. N. Najm, and C. Safta, "Trajectory optimization via unsupervised probabilistic learning on manifolds," 9 2021.

[71] C. Safta, R. G. Ghanem, M. J. Grant, M. Sparapany, and H. N. Najm, "Trajectory design via unsupervised probabilistic learning on optimal manifolds," *Data-Centric Engineering*, vol. 3, p. e26, 2022.

[72] B. Gaudet and R. Furfaro, "Adaptive pinpoint and fuel efficient mars landing using reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 1, no. 4, pp. 397–411, 2014.

[73] R. Furfaro, A. Scorsoglio, R. Linares, and M. Massari, "Adaptive generalized zem-zev feedback guidance for planetary landing via a deep reinforcement learning approach," *Acta Astronautica*, vol. 171, pp. 156–171, 2020.

[74] H. Holt, R. Armellin, A. Scorsoglio, and R. Furfaro, "Low-thrust trajectory design using closed-loop feedback-driven control laws and state-dependent parameters," in *AIAA Scitech 2020 Forum*.

[75] D. Miller, J. A. Englander, and R. Linares, "Interplanetary low-thrust design using proximal policy optimization," in *AAS/AIAA Astrodynamics Specialist Conference*, (Portland, ME, USA), 2019.

[76] L. Federici, A. Scorsoglio, A. Zavoli, and R. Furfaro, "Autonomous guidance for cislunar orbit transfers via reinforcement learning," in *2021 AAS/AIAA Astrodynamics Specialist Conference*, 2021.

[77] N. B. LaFarge, D. Miller, K. C. Howell, and R. Linares, "Guidance for closed-loop transfers using reinforcement learning with application to libration point orbits," in *AIAA Scitech 2020 Forum*.

[78] C. J. Sullivan and N. Bosanac, "Using reinforcement learning to design a low-thrust approach into a periodic orbit in a multi-body system," in *AIAA Scitech 2020 Forum*.

[79] J. Broida and R. Linares, "Spacecraft rendezvous guidance in cluttered environments via reinforcement learning," in *29th AAS/AIAA Space Flight Mechanics Meeting*, (Ka'anapali, HI, USA), 2019.

[80] L. Federici, A. Scorsoglio, A. Zavoli, and R. Furfaro, "Meta-reinforcement learning for adaptive spacecraft guidance during multi-target missions," in *72nd International Astronautical Congress (IAC)*, (Dubai, UAE), 2021.

[81] L. Federici, B. Benedikter, and A. Zavoli, "Deep learning techniques for autonomous spacecraft guidance during proximity operations," *Journal of Spacecraft and Rockets*, pp. 1–12, 2021.

[82] L. Federici, A. Scorsoglio, L. Ghilardi, A. D'Ambrosio, B. Benedikter, A. Zavoli, and R. Furfaro, "Image-based meta-reinforcement learning for autonomous terminal guidance of an impactor in a binary asteroid system," in *AIAA Scitech 2021 Forum*.

[83] B. Øksendal, *Stochastic differential equations: an introduction with applications*. Berlin, Heidelberg: Springer-Verlag, 6 ed., 2010.

[84] P. Kloeden and E. Platen, *Numerical solution of stochastic differential equations*. Berlin, Heidelberg: Springer-Verlag, 1992.

[85] N. J. Kasdin, "Runge-Kutta algorithm for the numerical integration of stochastic differential equations," *Journal of Guidance, Control, and Dynamics*, vol. 18, pp. 114–120, Jan. 1995.

[86] S. S. Artemiev and T. A. Averina, *Numerical Analysis of Systems of Ordinary and Stochastic Differential Equations*. Berlin, New York: De Gruyter, 1997.

[87] G. N. Mil'shtejn, "Approximate integration of stochastic differential equations," *Theory of probability & its applications*, vol. 19, no. 3, pp. 557–562, 1975.

[88] A. Charnes and W. W. Cooper, "Chance-constrained programming," *Management Science*, vol. 6, no. 1, pp. 73–79, 1959.

[89] B. L. Miller and H. M. Wagner, "Chance constrained programming with joint constraints," *Operations Research*, vol. 13, no. 6, pp. 930–945, 1965.

[90] A. Prekopa, *Stochastic Programming*. Kluwer Academic Publishers, 1995.

[91] A. Nemirovski and A. Shapiro, "Convex approximations of chance constrained programs," *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 969–996, 2006.

[92] G. C. Calafiore and M. C. Campi, "Uncertain convex programs: randomized solutions and confidence levels," *Mathematical Programming*, vol. 102, no. 1, pp. 25–46, 2005.

[93] M. Wendt, P. Li, and G. Wozny, "Nonlinear chance-constrained process optimization under uncertainty," *Industrial & Engineering Chemistry Research*, vol. 41, no. 15, pp. 3621–3629, 2002.

[94] T. Mühlpfordt, L. Roald, V. Hagenmeyer, T. Faulwasser, and S. Misra, "Chance-constrained ac optimal power flow: A polynomial chaos approach," *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 4806–4816, 2019.

[95] T. Mühlpfordt, T. Faulwasser, and V. Hagenmeyer, "A generalized framework for chance-constrained optimal power flow," *Sustainable Energy, Grids and Networks*, vol. 16, pp. 231–242, 2018.

[96] S. Uryasev, *Probabilistic constrained optimization: methodology and applications*, vol. 49. Springer Science & Business Media, 2013.

[97] T. Lew, R. Bonalli, and M. Pavone, "Chance-constrained sequential convex programming for robust trajectory optimization," in *2020 European Control Conference (ECC)*, pp. 1871–1878, 2020.

[98] L. Blackmore, M. Ono, and B. C. Williams, "Chance-constrained optimal path planning with obstacles," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1080–1094, 2011.

[99] D. Strawser and B. Williams, "Approximate branch and bound for fast, risk-bound stochastic path planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7047–7054, 2018.

[100] L. Hewing and M. N. Zeilinger, "Stochastic model predictive control for linear systems using probabilistic reachable sets," in *Proceedings of the IEEE Conference on Decision and Control*, 2018.

[101] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *Proceedings of the IEEE Conference on Decision and Control*, 2018.

[102] J. Ridderhof, J. Pilipovsky, and P. Tsiotras, "Chance-constrained covariance control for low-thrust minimum-fuel trajectory optimization," in *AAS/AIAA Astrodynamics Specialist Conference*, (South Lake Tahoe, CA, USA), Aug. 2020.

[103] J. Ridderhof and P. Tsiotras, "Chance-constrained covariance steering in a gaussian random field via successive convex programming," 2022.

[104] K. Okamoto, M. Goldshtein, and P. Tsiotras, "Optimal covariance control for stochastic systems under chance constraints," 2018.

[105] B. Gaudet, R. Linares, and R. Furfaro, "Deep reinforcement learning for six degree-of-freedom planetary landing," *Advances in Space Research*, vol. 65, pp. 1723–1741, Apr. 2020.

[106] L. Federici, B. Benedikter, and A. Zavoli, "Machine learning techniques for autonomous spacecraft guidance during proximity operations," in *AIAA Scitech 2021 Forum*, pp. 1–18, 2021.

[107] N. B. LaFarge, D. Miller, K. C. Howell, and R. Linares, "Autonomous closed-loop guidance using reinforcement learning in a low-thrust, multi-body dynamical environment," *Acta Astronautica*, vol. 186, pp. 1–23, 2021.

[108] S. Hochreiter, A. S. Younger, and P. R. Conwell, "Learning to learn using gradient descent," in *Artificial Neural Networks - ICANN 2001*, pp. 87–94, Springer Berlin Heidelberg, 2001.

[109] B. Gaudet, R. Linares, and R. Furfaro, "Terminal adaptive guidance via reinforcement meta-learning: Applications to autonomous asteroid close-proximity operations," *Acta Astronautica*, vol. 171, pp. 1–13, 2020.

[110] A. Scorsoglio, A. D'Ambrosio, L. Ghilardi, B. Gaudet, F. Curti, and R. Furfaro, "Image-based deep reinforcement meta-learning for autonomous lunar landing," *Journal of Spacecraft and Rockets*, vol. 59, no. 1, pp. 153–165, 2022.

[111] L. Federici, A. Scorsoglio, A. Zavoli, and R. Furfaro, "Optical 6-dof guidance and navigation of an asteroid impactor via meta-reinforcement learning," in *2022 AAS/AIAA Astrodynamics Specialist Conference*, Aug. 2022.

[112] D. H. Jacobson and D. Q. Mayne, *Differential dynamic programming*. American Elsevier Pub. Co New York, 1970.

[113] J. Ridderhof, P. Tsiotras, and B. J. Johnson, "Stochastic entry guidance," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 2, pp. 320–334, 2022.

[114] B. Benedikter, A. Zavoli, G. Colasurdo, S. Pizzurro, and E. Cavallini, "Autonomous upper stage guidance with robust splash-down constraint," in *72nd International Astronautical Congress (IAC)*, (Dubai, UAE), 2021.

[115] B. Benedikter, A. Zavoli, Z. Wang, S. Pizzurro, and E. Cavallini, "Convex approach to covariance control with application to stochastic low-thrust trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 11, pp. 2061–2075, 2022.

[116] K. Oguri and J. Mcmahon, "Risk-aware trajectory design with impulsive maneuvers: Convex optimization approach," in *Proceedings of the 2019 AAS/AIAA Astrodynamics Specialist*, (Portland, ME, USA), Aug. 2019.

[117] K. Oguri and G. Lantoine, "Stochastic sequential convex programming for robust low-thrust trajectory design under uncertainty," in *AAS/AIAA Astrodynamics Specialist Conference*, (Charlotte, NC, USA), Aug. 2022.

[118] F. Xiong, S. Chen, and Y. Xiong, "Dynamic system uncertainty propagation using polynomial chaos," *Chinese Journal of Aeronautics*, vol. 27, no. 5, pp. 1156–1170, 2014.

[119] N. Ozaki, S. Campagnola, R. Funase, and C. H. Yam, "Stochastic differential dynamic programming with unscented transform for low-thrust trajectory design," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 2, pp. 377–387, 2018.

[120] N. Ozaki, S. Campagnola, and R. Funase, "Tube stochastic optimal control for nonlinear constrained trajectory optimization problems," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 4, 2020.

[121] C. Giordano and F. Topputo, "Analysis, design, and optimization of robust trajectories in cislunar environment for limited-capability spacecraft," *The Journal of the Astronautical Sciences*, vol. 70, 11 2023.

[122] J. D. Feldhacker, B. A. Jones, and A. Doostan, "Trajectory optimization under uncertainty for rendezvous in the CRTBP," in *26th AAS/AIAA Space Flight Mechanics Meeting*, (Springfield, VA, USA), Feb. 2016.

[123] C. Greco, S. Campagnola, and M. Vasile, "Robust space trajectory design using belief optimal control," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 6, pp. 1060–1077, 2022.

[124] C. Greco, M. Di Carlo, M. Vasile, and R. Epenoy, "Direct multiple shooting transcription with polynomial algebra for optimal control problems under uncertainty," *Acta Astronautica*, vol. 170, pp. 224–234, 2020.

[125] C. Greco, S. Campagnola, and M. Vasile, "Robust space trajectory design using belief stochastic optimal control," in *AIAA Scitech 2020 Forum*, American Institute of Aeronautics and Astronautics, Jan. 2020.

[126] K. Oguri and J. W. McMahon, "Stochastic primer vector for robust low-thrust trajectory design under uncertainty," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 1, pp. 84–102, 2022.

[127] J. Fisher and R. Bhattacharya, "Optimal Trajectory Generation With Probabilistic System Uncertainty Using Polynomial Chaos," *Journal of Dynamic Systems, Measurement, and Control*, vol. 133, p. 014501, Nov. 2010.

[128] F. Wang, S. Yang, F. Xiong, Q. Lin, and J. Song, "Robust trajectory optimization using polynomial chaos and convex optimization," *Aerospace Science and Technology*, vol. 92, pp. 314–325, 2019.

[129] D. Woffinden and B. Barton, "Optimized trajectory correction burn placement for NRHO orbit maintenance," in *33rd AAS/AIAA Space Flight Mechanics Meeting*, (Austin, TX, USA), Jan. 2023.

[130] E. L. Jenson and D. J. Scheeres, "Multi-objective optimization of covariance and energy for asteroid transfers," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 7, pp. 1253–1265, 2021.

[131] E. Theodorou, Y. Tassa, and E. Todorov, "Stochastic differential dynamic programming," in *Proceedings of the 2010 American Control Conference*, pp. 1125–1132, 2010.

[132] S. Zimmer, C. Ocampo, and R. Bishop, "Reducing orbit covariance for continuous thrust spacecraft transfers," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 2, pp. 771–791, 2010.

[133] M. Xin and H. Pan, "Indirect robust control of spacecraft via optimal control solution," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 2, pp. 1798–1809, 2012.

[134] C. R. Heidrich and M. J. Holzinger, "Auxiliary minimum covariance resolution of infinite-order singular arcs," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 10, pp. 1847–1861, 2022.

[135] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2002.

[136] C. R. Cutler and B. L. Ramaker, "Dynamic matrix control – a computer control algorithm," in *Joint Automatic Control Conference*, vol. 17, p. 72, 1980.

[137] U. Eren, A. Prach, B. B. Kocer, S. V. Rakovic, E. Kayacan, and B. Ackmese, "Model predictive control in aerospace systems: Current state and opportunities," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 7, pp. 1541–1566, 2017.

[138] P. Iannelli, F. Angeletti, and P. Gasbarri, "An attitude/spin control architecture for a spacecraft equipped with a flexible rotating payload based on model predictive control," in *72nd International Astronautical Congress (IAC)*, (Dubai, UAE), 2021.

[139] Z. Wang and M. J. Grant, "Optimization of minimum-time low-thrust transfers using convex programming," *Journal of Spacecraft and Rockets*, vol. 55, no. 3, pp. 586–598, 2018.

[140] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.

[141] Y. Kuwata, A. Richards, and J. How, "Robust receding horizon control using generalized constraint tightening," in *American Control Conference*, (New York, NY), pp. 4482–4487, Institute of Electrical and Electronics Engineers (IEEE), July 2007.

[142] C. Buckner and R. Lampariello, "Tube-based model predictive control for the approach maneuver of a spacecraft to a free-tumbling target satellite," in *2018 Annual American Control Conference (ACC)*, pp. 5690–5697, June 2018.

[143] B. Benedikter, A. Zavoli, Z. Wang, S. Pizzurro, and E. Cavallini, "Convex approach to covariance control with application to stochastic low-thrust trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 11, pp. 2061–2075, 2022.

[144] D. Geller, M. B. Rose, and D. Woffinden, "Event triggers in linear covariance analysis with applications to orbital rendezvous," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 1, pp. 102–111, 2009.

[145] D. Geller and D. Christensen, "Linear covariance analysis for powered lunar descent and landing," *Journal of Spacecraft and Rockets*, vol. 46, no. 6, pp. 1231–1248, 2009.

[146] A. Gelb and R. Warren, "Direct statistical analysis of nonlinear systems: Cadet," *AIAA Journal*, vol. 11, no. 5, pp. 689–694, 1973.

[147] A. Gelb, *Applied Optimal Estimation*. MIT Press, 1974.

[148] J. Taylor, "Handbook for the direct statistical analysis of missile guidance systems via cadet," Tech. Rep. AD-A013397, National Technical Information Service, Reading, MA, 1975.

[149] S. Park, *Nonlinear Trajectory Navigation*. PhD thesis, University of Michigan, Ann Arbor, MI, 2007.

[150] J. L. Junkins, M. R. Akella, and K. T. Alfriend, "Non-gaussian error propagation in orbital mechanics," *Journal of Astronautical Sciences*, vol. 44, no. 4, pp. 541–563, 1996.

[151] J. L. Junkins and P. Singla, "How nonlinear is it? a tutorial on nonlinearity of orbit and attitude dynamics," *Journal of Astronautical Sciences*, vol. 52, no. 1-2, pp. 7–60, 2004.

[152] D. J. Scheeres, F.-Y. Hsiao, R. S. Park, B. F. Villac, and J. M. Maruskin, "Fundamental limits on spacecraft orbit uncertainty and distribution propagation," *Journal of Astronautical Sciences*, vol. 54, no. 3-4, pp. 505–523, 2006.

[153] D. J. Scheeres, M. A. de Gosson, and J. M. Maruskin, "Applications of symplectic topology to orbit uncertainty and spacecraft navigation," *Journal of Astronautical Sciences*, vol. 59, no. 1-2, pp. 63–83, 2012.

[154] R. S. Park and D. J. Scheeres, "Nonlinear mapping of gaussian statistics: theory and applications to spacecraft trajectory design," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 6, pp. 1367–1375, 2006.

[155] K. Fujimoto, D. J. Scheeres, and K. T. Alfriend, "Analytical nonlinear propagation of uncertainty in the two-body problem," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 497–509, 2012.

[156] K. Fujimoto, *New Methods in Optical Track Association and Uncertainty Mapping of Earth-orbiting Objects*. PhD thesis, University of Colorado Boulder, 2013.

[157] R. S. Park, K. Fujimoto, and D. J. Scheeres, "Effect of dynamical accuracy for uncertainty propagation of perturbed keplerian motion," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 12, pp. 2287–2300, 2015.

[158] R. Barrio, M. Rodríguez, A. Abad, and S. Serrano, "Uncertainty propagation or box propagation," *Mathematics and Computer Modelling*, vol. 54, no. 11-12, pp. 2602–2615, 2011.

[159] N. Nakhriri and B. Villac, "Modified picard integrator for spaceflight mechanics," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 5, pp. 1625–1637, 2014.

[160] R. Askey and J. Wilson, "Some basic hypergeometric polynomials that generalize jacobi polynomials (memoirs of the american mathematical society, 319)," *American Mathematical Society, Providence, RI*, 1985.

[161] D. Xiu and G. E. Karniadakis, "The wiener–askey polynomial chaos for stochastic differential equations," *SIAM Journal on Scientific Computing*, vol. 24, no. 2, pp. 619–644, 2002.

[162] B. A. Jones, A. Doostan, and G. H. Born, "Nonlinear propagation of orbit uncertainty using non-intrusive polynomial chaos," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 2, pp. 430–444, 2013.

[163] N. Wiener, "The homogeneous chaos," *American Journal of Mathematics*, vol. 60, no. 4, pp. 897–936, 1938.

[164] X. Wan and G. Karniadakis, "Multi-element generalized polynomial chaos for arbitrary probability measures," *SIAM Journal on Scientific Computing*, vol. 28, pp. 901–928, 01 2006.

[165] J. T. Horwood, N. D. Aragon, and A. B. Poore, "Gaussian sum filters for space surveillance: theory and simulations," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 6, pp. 1839–1851, 2011.

[166] J. T. Horwood and A. B. Poore, "Adaptive gaussian sum filters for space surveillance," *IEEE Transactions on Automatic Control*, vol. 56, no. 8, pp. 1777–1790, 2011.

[167] K. J. DeMars, R. H. Bishop, and M. K. Jah, "Entropy-based approach for uncertainty propagation of nonlinear dynamical systems," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 4, pp. 1047–1057, 2013.

[168] K. J. DeMars, *Nonlinear Orbit Uncertainty Prediction and Rectification for Space Situational Awareness*. PhD thesis, University of Texas at Austin, 2010.

[169] L. M. Psiaki, J. R. Schoenberg, and I. M. Miller, "Gaussian sum reapproximation for use in a nonlinear filter," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 2, pp. 292–303, 2015.

[170] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, "Uncertainty propagation for nonlinear dynamic systems using gaussian mixture models," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 6, pp. 1623–1633, 2008.

[171] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, "Adaptive gaussian sum filter for nonlinear bayesian estimation," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2151–2156, 2011.

[172] K. Vishwajeet, P. Singla, and M. K. Jah, "Nonlinear uncertainty propagation for perturbed two-body orbits," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 5, pp. 1415–1425, 2014.

[173] A. Fuller, "Analysis of nonlinear stochastic systems by means of the fokker–planck equation," *International Journal of Control*, vol. 9, no. 6, pp. 603–655, 1969.

[174] S. Julier and J. K. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," tech. rep., 1996.

[175] J. Uhlmann, S. J. Julier, and H. Durrant-Whyte, "A new method for the non linear transformation of means and covariances in filters and estimations," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, 2000.

[176] S. J. Julier and J. K. Uhlmann, "Reduced sigma point filters for the propagation of means and covariances through nonlinear transformations," in *2002 American Control Conference*, (Anchorage (AL) USA), pp. 887–892, IEEE, May 2002. IEEE Cat. No. CH37301.

[177] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[178] N. Adurthi, P. Singla, and T. Singh, "The conjugate unscented transform — an approach to evaluate multi-dimensional expectation integrals," in *2012 American Control Conference (ACC)*, pp. 5556–5561, 2012.

[179] R. Armellin, P. Di Lizia, F. Bernelli-Zazzera, and M. Berz, "Asteroid close encounters characterization using differential algebra: the case of apophis," *Celestial Mechanics and Dynamical Astronomy*, vol. 107, no. 4, pp. 451–470, 2010.

[180] M. Valli, R. Armellin, P. Di Lizia, and M. R. Lavagna, "Nonlinear mapping of uncertainties in celestial mechanics," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 1, pp. 48–63, 2013.

[181] M. Berz, *Modern Map Methods in Particle Beam Physics*. London: Academic Press, 1999.

[182] Y. Sun and M. Kumar, "Uncertainty propagation in orbital mechanics via tensor decomposition," *Celestial Mechanics and Dynamical Astronomy*, vol. 124, no. 3, pp. 269–294, 2015.

[183] M. Kumar, S. Chakravorty, and J. Junkins, "A semianalytic meshless approach to the transient fokker–planck equation," *Probabilistic Engineering Mechanics*, vol. 25, no. 3, pp. 323–331, 2010.

[184] Y. Sun and M. Kumar, "A markov chain monte carlo particle solution of the initial uncertainty propagation problem," in *AIAA Guidance, Navigation, and Control Conference*, (Minneapolis, MN, USA), 2012.

[185] M. Kumar, S. Chakravorty, and J. Junkins, "A homotopic approach to domain determination and solution refinement for the stationary fokker-planck equation," *Probabilistic Engineering Mechanics*, vol. 24, no. 3, pp. 265–277, 2009.

[186] Y. Sun and M. Kumar, "Numerical solution of high dimensional stationary fokker–planck equations via tensor decomposition and chebyshev spectral differentiation," *Computational Mathematics and Applications*, vol. 67, no. 10, pp. 1960–1977, 2014.

[187] K. Fujimoto and D. J. Scheeres, "Tractable expressions for nonlinearity propagated uncertainties," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 6, pp. 1146–1151, 2015.

[188] V. R. Vittaldev, R. P. Russell, and R. Linares, "Spacecraft uncertainty propagation using gaussian mixture models and polynomials," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 12, pp. 2615–2626, 2016.

[189] J. M. Aristoff, J. T. Horwood, and A. B. Poore, "Orbit and uncertainty propagation: a comparison of gauss–legendre-, dormand–prince-, and chebyshev–picard-based approaches," *Celestial Mechanics and Dynamical Astronomy*, vol. 118, no. 1, pp. 13–28, 2014.

[190] J. M. Aristoff, J. T. Horwood, and A. B. Poore, "Implicit–runge–kutta–based methods for fast, precise, and scalable uncertainty propagation," *Celestial Mechanics and Dynamical Astronomy*, vol. 122, no. 2, pp. 169–182, 2015.

[191] J. T. Horwood and A. B. Poore, "Gauss von mises distribution for improved uncertainty realism in space situational awareness," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 2, pp. 276–304, 2014.

[192] J. T. Horwood, J. M. Aristoff, N. Singh, and A. B. Poore, "A comparative study of new non-linear uncertainty propagation methods for space surveillance," in *SPIE Defense+ Security*, International Society for Optics and Photonics, 2014.

[193] C. Sabol, K. Hill, K. T. Alfriend, and T. Sukut, "Nonlinear effects in the correlation of tracks and covariance propagation," *Acta Astronautica*, vol. 84, pp. 69–80, 2013.

[194] J. W. Woodburn and V. T. Coppola, "Effect of coordinate selection on orbit determination," in *AAS/AIAA Astrodynamics Specialist Conference*, (Hilton Head Island, SC, USA), 2013.

[195] J. M. Aristoff, J. T. Horwood, N. Singh, and A. B. Poore, "Nonlinear uncertainty propagation in orbital elements and transformation to cartesian space without loss of realism," in *AAS/AIAA Astrodynamics Specialist Conference*, (San Diego, CA, USA), AIAA, 2014.

[196] M. Majji, J. L. Junkins, and J. D. Turner, "Measurement model nonlinearity in estimation of dynamical systems," *Journal of Astronautical Sciences*, vol. 59, no. 1-2, pp. 41–62, 2012.

[197] K. Hill, K. T. Alfriend, C. Sabol, and T. Sukut, "Covariance-based uncorrelated track association," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, (Honolulu, HI, USA), AIAA, 2008.

[198] K. Hill, C. Sabol, and K. T. Alfriend, "Comparison of covariance based track association approaches using simulated radar data," *Journal of Astronautical Sciences*, vol. 59, no. 1-2, pp. 287–306, 2012.

[199] D. A. Vallado and S. Alfano, "Curvilinear coordinate transformations for relative motion," *Celestial Mechanics and Dynamical Astronomy*, vol. 118, no. 3, pp. 253–271, 2014.

[200] S. Tanygin, "Efficient covariance interpolation using blending of approximate covariance propagations," *Journal of Astronautical Sciences*, vol. 61, no. 1, pp. 107–132, 2014.

[201] V. T. Coppola and S. Tanygin, "Using bent ellipsoids to represent large position covariance in orbit propagation," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 9, pp. 1775–1784, 2015.

[202] C. Büskens and D. Wassel, "The esa nlp solver worhp," in *Modeling and Optimization in Space Engineering* (G. Fasano and J. D. Pintér, eds.), vol. 73, pp. 85–110, Springer New York, 2013.

[203] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and its Applications*, vol. 284, no. 1, pp. 193–228, 1998. International Linear Algebra Society (ILAS) Symposium on Fast Algorithms for Control, Signals and Image Processing.

[204] O. Montenbruck and E. Gill, *Satellite Orbits: Models, Methods, and Applications*. Springer-Verlag, 2000.

[205] X. He, U. Hugentobler, A. Schlicht, Y. Nie, and B. Duan, "Precise orbit determination for a large leo constellation with inter-satellite links and the measurements from different ground networks: a simulation study," *Satellite Navigation*, vol. 3, no. 1, p. 22, 2022.

[206] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, Mar. 1960.

[207] B. Tapley, B. Schutz, and G. Born, *Chapter 4. Fundamentals of Orbit Determination*, pp. 159–284. 12 2004.

[208] N. Ozaki, R. Hyodo, Y. Takao, D. Z. Seligman, M. E. Brown, S. Hernandez, M. Yoshikawa, and M. Fujimoto, "Rapid response flyby exploration using deep space constellation deployed on asteroid cyclers," in *8th IAA Planetary Defense Conference*, (Vienna, Austria), Apr. 2023.

[209] D. Guzzetti, E. M. Zimovan, K. C. Howell, and D. C. Davis, "Stationkeeping analysis for spacecraft in lunar near rectilinear halo orbits," in *AAS/AIAA Astrodynamics Specialist Conference*, Feb. 2017.

[210] A. Mathai and S. Provost, *Quadratic forms in random variables: theory and applications*, vol. 87. Dec. 1992.