Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2025) 416-429

ECCOMAS
Proceedia

UNCECOMP 2025
6th ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Science and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Rhodes Island, Greece, 15-18 June 2025

STOCHASTIC OPTIMIZATION FOR SPACE MISSION ANALYSIS

Thomas Caleb[1], Roberto Armellin[2], Spencer Boone[1], and Stéphanie Lizy-Destrez[1]

[1]DCAS/SaCLaB, ISAE-SUPAERO
Toulouse, 31055, France
e-mail: {thomas.caleb,spencer.boone,stephanie.lizy-destrez}@isae-supaero.fr

[2]Te Pūnaha Ātea – Space Institute, University of Auckland
Auckland, 1142, New Zealand
e-mail: roberto.armellin@auckland.ac.nz

Abstract. *Space missions are increasingly complex, with trajectories sensitive to perturbations that can lead to mission failure or costly maneuvers. Consequently, mission analysis robustness must be carefully evaluated, driving up development costs. This work aims to reduce failure risks while minimizing fuel consumption. Stochastic optimization methods help address these challenges and reduce mission design time. These methods rely on three components: 1. Uncertainty propagation to carry random distributions forward. 2. Transcription that reformulates stochastic constraints into deterministic ones. 3. A deterministic solver for the transcribed problem. Existing stochastic methods often struggle with onboard applications due to computational demands, limited precision, or conservatism. For instance, Stochastic Differential Dynamic Programming (SDDP) [1] uses sequential quadratic optimization and propagates Gaussian uncertainties through linearized dynamics, but its transcription method, based on the spectral radius of constraint covariances, can be overly conservative.*

To address these limitations, we introduce a Taylor polynomial-based SDDP solver, using Taylor polynomials [2] for automatic differentiation, dynamic approximations, and uncertainty propagation. Combining these functions into one framework, we significantly accelerate the code, making it viable for onboard computations. Additionally, we propose a novel transcription method for constraints and cost functions, reducing conservatism. Our tool was tested on an Earth-Mars transfer scenario, yielding a robust control law that meets performance expectations. With a failure risk limit of 5%*, the code runs in under a minute on a standard laptop, with a failure risk of* 2.9%*, compared to* 99.6% *for the non-robust version, with similar fuel consumption. This control law can be implemented onboard, balancing cost control and mission reliability. Thus, this work marks a significant step toward practical onboard applications of stochastic optimization, providing a fast, less conservative approach for reliable control of complex missions.*

Keywords: Space mission analysis, Stochastic optimization, Chance constraints, Differential dynamic programming, High-order Taylor polynomials.

ISSN:2623-3339 © 2025 The Authors. Published by Eccomas Proceedia.
Peer-review under responsibility of the organizing committee of UNCECOMP 2025.
doi: 10.7712/120225.12367.21078

1 INTRODUCTION

Modern space missions increasingly rely on sophisticated dynamical systems, requiring precise trajectory design to ensure success. Interplanetary missions often rely on gravity assists, as in European space agency (ESA) JUICE mission [see 9], or exploit sensitive phenomena like ballistic capture [32], a key factor in the ESA and Japan aerospace exploration agency (JAXA)'s BepiColombo mission [3]. Additionally, advanced dynamical models, such as the circular restricted three-body problem (CR3BP), play a key role in enabling sustainable lunar exploration [30].

However, the non-linear and chaotic nature of these dynamical systems poses significan challenges for trajectory design and mission robustness [27]. Small variations in a spacecraft's position due to navigation uncertainties, measurement errors, or maneuver inaccuracies can lead to substantial deviations from the intended path. Ensuring mission success, therefore, requires trajectory designs that are both efficien and resilient to these uncertainties.

Traditionally, trajectory planning has relied on deterministic optimization, where an optimal path is computed without accounting for stochastic effects. Once determined, the robustness of this trajectory is tested against potential uncertainties, often requiring additional fuel for corrective maneuvers. While effective in many cases, this method can yield suboptimal or even infeasible solutions when dealing with highly sensitive or constrained missions.

To address these limitations, stochastic optimization techniques have gained interest in space mission analysis [6, 24, 2]. These approaches incorporate uncertainty directly into the trajectory design process, allowing for more robust solutions that mitigate risks while maintaining mission feasibility. Recent methodologies have been generally structured into three main blocks:

(i) Uncertainty propagation: models how uncertainties evolve over time, affecting the spacecraft's trajectory.

(ii) Transcription method: transforms stochastic constraints and objective function into a deterministic, computationally tractable form. However, existing approaches often introduce conservatism or are limited to specifi cases [6, 29].

(iii) Deterministic optimization solver: used to solve the transcribed problem. Off-the-shelf solvers are commonly used [14, 2, 21]. Recent research efforts have focused on specialized solvers that perform common computations for all three blocks to improve efficien y and feasibility [24, 23].

To address the limitations of existing methods, we introduce linear stochastic optimization with differential algebra (L-SODA) solver, a constrained stochastic differential dynamic programming (DDP) solver that pools computations using the differential algebra (DA) framework framework to enhance efficiency and accuracy. Moreover, it implements novel transcription methods that strongly reduce conservatism and improve convergence.

Section 2 introduces constrained DDP, stochastic optimization, transcription methods, and the role of DA in scientifi computing. Then, we present novel transcription techniques and introduce the L-SODA solver in Section 3. Section 4 presents validation and application of this solver, respectively to a double integrator theoretical problem, and to a standard fuel-optimal low-thrust Earth-Mars transfer. Finally, conclusions are drawn in Section 5.

2 BACKGROUND

This section briefl presents the theory on constrained DDP, stochastic solvers in astrodynamics and and the DA.

2.1 Constrained Differential Dynamic Programming

This part presents a constrained DDP solver, a deterministic solver for constrained optimization. We firs introduce DDP and then present the augmented Lagrangian (AUL) formulation.

2.1.1 Differential Dynamic Programming

DDP tackles unconstrained optimization problems divided into N stages with dynamics of the type: $\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{x}_{k+1}$ [22, 16], where $k \in \mathbb{N}_{N-1} = \{0, 1, \ldots, N-1\}$, $\boldsymbol{x}_k$ is the state vector of size N_x, $\boldsymbol{u}_k$ is the control vector of size N_u, and $\boldsymbol{f}$ represents the dynamics. The initial and target state vectors $\boldsymbol{x}_0$ and $\boldsymbol{x}_t$ of size N_x are given, and the objective function to minimize is:

$$J(\boldsymbol{U}, \boldsymbol{x}_0, \boldsymbol{x}_t) = \sum_{k=0}^{N-1} l(\boldsymbol{x}_k, \boldsymbol{u}_k) + \phi(\boldsymbol{x}_N, \boldsymbol{x}_t) \tag{1}$$

where $\boldsymbol{U} = (\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{N-1})$ is the vector of controls, $\boldsymbol{x}_N$ is the fina propagated state vector, l is the stage cost, and ϕ is the terminal cost. DDP consists of two phases looping until convergence:

(i) **Backward sweep**: Given a control $\boldsymbol{U}$, a list of states $\boldsymbol{X} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$, and costs $l_k = l(\boldsymbol{x}_k, \boldsymbol{u}_k)$ and $\phi_N = \phi(\boldsymbol{x}_N, \boldsymbol{x}_t)$, determine the list of corrections $\boldsymbol{a}_k$ and $\boldsymbol{K}_k$ to perform on $\boldsymbol{u}_k$ from $k = N-1$ to $k = 0$ to reduce the overall cost. These corrections are computed using Bellman's principle and by successive quadratic approximation of the cost-to-go function: $V_k(\boldsymbol{x}_k) = \min_{\boldsymbol{U}_k} \left[\sum_{i=k}^{N-1} l(\boldsymbol{x}_k, \boldsymbol{u}_k) + \phi(\boldsymbol{x}_N, \boldsymbol{x}_t)\right]$. This phase requires the firs and second-order derivatives of $\boldsymbol{f}_k = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)$, l_k, and ϕ_N with respect to $\boldsymbol{x}$ and $\boldsymbol{u}$.

(ii) **Forward pass**: Given the corrections $\boldsymbol{A} = (\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{N-1})$ and $\boldsymbol{K} = (\boldsymbol{K}_0, \boldsymbol{K}_1, \ldots, \boldsymbol{K}_{N-1})$, and starting from $k = 0$ to $k = N-1$, compute the new controls $\boldsymbol{u}_k^*$ depending on the new state $\boldsymbol{x}_k^*$: $\boldsymbol{u}_k^* = \boldsymbol{u}_k + \boldsymbol{a}_k + \boldsymbol{K}_k \cdot (\boldsymbol{x}_k^* - \boldsymbol{x}_k)$, with $\boldsymbol{x}_0^* = \boldsymbol{x}_0$. This phase requires the evaluation of the dynamics, which is the most computation-intensive part [7].

2.1.2 Augmented Lagrangian Formulation

The DDP solver does not implement constraints by design, so they must be added to solve typical space trajectory design problems. We present the method of the ALTRO solver from Howell et al. [15], which uses the AUL formulation. In this approach, constraints are added to the objective function using a dual state and a penalty factor. The constrained problem can now be solved using an unconstrained solver, such as DDP. The constrained problem is formulated as follows:

$$\begin{aligned} &\min_{\boldsymbol{U}} J\left(\boldsymbol{U}, \boldsymbol{x}_0, \boldsymbol{x}_t\right) = \sum_{k=0}^{N-1} l\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) + \phi\left(\boldsymbol{x}_N, \boldsymbol{x}_t\right), \\ &\text{subject to:} \\ &\boldsymbol{x}_{k+1} = \boldsymbol{f}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right), \ \forall k \in \mathbb{N}_{N-1}, \\ &\boldsymbol{g}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \preceq \boldsymbol{0}_{N_g}, \ \forall k \in \mathbb{N}_{N-1}, \\ &\boldsymbol{g}_t\left(\boldsymbol{x}_N, \boldsymbol{x}_t\right) \preceq \boldsymbol{0}_{N_{gt}}, \end{aligned} \tag{2}$$

where $\boldsymbol{g}_k = \boldsymbol{g}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)$ are the path constraints of size N_g, $\boldsymbol{g}_N = \boldsymbol{g}_t\left(\boldsymbol{x}_N, \boldsymbol{x}_t\right)$ are the terminal constraints of size N_{gt}, and $\forall \left(\boldsymbol{y}, \boldsymbol{z}\right) \in \left(\mathbb{R}^d\right)^2$: $\boldsymbol{y} \preceq \boldsymbol{z} \Leftrightarrow \forall i \in \mathbb{N}_d, \ y^i \leq z^i$.

This problem is solved by augmenting the stage costs with the path constraints to obtain $\tilde{l}$ and the terminal cost with the terminal constraints to obtain $\tilde{\phi}$:

$$\begin{cases} \tilde{l}\left(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\lambda}_k, \boldsymbol{\mu}_k\right) & = l_k + \left[\boldsymbol{\lambda}_k + \frac{\mathcal{I}_{\boldsymbol{\mu},k}}{2} \cdot \boldsymbol{g}_k\right]^T \cdot \boldsymbol{g}_k, \ k \in \mathbb{N}_{N-1} \\ \tilde{\phi}\left(\boldsymbol{x}_N, \boldsymbol{x}_t, \boldsymbol{\lambda}_N, \boldsymbol{\mu}_N\right) & = \phi_N + \left[\boldsymbol{\lambda}_N + \frac{\mathcal{I}_{\boldsymbol{\mu},N}}{2} \cdot \boldsymbol{g}_N\right]^T \cdot \boldsymbol{g}_N \end{cases} \tag{3}$$

where the tilde denotes augmented cost functions, $\boldsymbol{\lambda}_k$ is the dual state of size N_g if $k \in \mathbb{N}_{N-1}$ or of size N_{gt} if $k = N$, and $\boldsymbol{\mu}_k$ is the penalty vector of the same size as $\boldsymbol{\lambda}_k$. Please refer to Howell et al. [15] for more details on dual state and penalty updates. The augmented problem is solved using DDP until the maximum constraint violation $g_{\max}$ is below the target constraint satisfaction $\varepsilon_{\text{AUL}} > 0$.

2.2 Stochastic Optimization Problem and Existing Chance-Constraints Transcription Methods

Several assumptions are made: uncertainties follow a multivariate Gaussian distribution [31], functions are sufficiently regular, and covariance norms are small. These ensure all random variables can be modeled as Gaussian distributions.

2.2.1 Constrained Stochastic Trajectory Optimization Problem

We now formulate the constrained stochastic trajectory optimization problem within the DDP framework. The states and controls are modeled as Gaussian distributions with respective means and covariances $\bar{\boldsymbol{x}}_k$, $\bar{\boldsymbol{u}}_k$, $\boldsymbol{\Sigma}_{\boldsymbol{x},k}$, and $\boldsymbol{\Sigma}_{\boldsymbol{u},k}$. The nominal controls are augmented with a linear feedback gain $\boldsymbol{K} = \left(\boldsymbol{K}_0, \boldsymbol{K}_1, \ldots, \boldsymbol{K}_{N-1}\right)$. The goal is to fin a control $\left(\bar{\boldsymbol{U}}, \boldsymbol{K}\right)$ of failure risk below a given $\beta \in]0, 1[$, the target failure risk. These controls must minimize $\tilde{J}$, the $(1-\beta)$–quantile of $J\left(\bar{\boldsymbol{U}}, \boldsymbol{K}, \boldsymbol{x}_0, \boldsymbol{x}_t\right)$ (or $(1-\beta)$–cost). In other words, there is a probability $1-\beta$ that the cost function will be below the optimized objective function $\tilde{J}$. The problem is formulated as follows:

$$
\begin{aligned}
&\min_{\bar{\boldsymbol{U}},\boldsymbol{K}} \tilde{J}\left(\bar{\boldsymbol{U}},\boldsymbol{K},\boldsymbol{x}_0,\boldsymbol{x}_t,\beta\right),\\
&\text{subject to:}\\
&J\left(\bar{\boldsymbol{U}},\boldsymbol{K},\boldsymbol{x}_0,\boldsymbol{x}_t\right)=\sum_{k=0}^{N-1} l\left(\boldsymbol{x}_k,\boldsymbol{u}_k\right)+\phi\left(\boldsymbol{x}_N,\boldsymbol{x}_t\right),\\
&\boldsymbol{x}_{k+1}=\boldsymbol{f}\left(\boldsymbol{x}_k,\boldsymbol{u}_k\right)+\boldsymbol{w}_{x,k+1},\ \forall k\in\mathbb{N}_{N-1},\\
&\boldsymbol{u}_k=\bar{\boldsymbol{u}}_k+\boldsymbol{K}_k\cdot(\boldsymbol{x}_k-\bar{\boldsymbol{x}}_k),\ \forall k\in\mathbb{N}_{N-1},\\
&\mathbb{P}\left[J\left(\bar{\boldsymbol{U}},\boldsymbol{K},\boldsymbol{x}_0,\boldsymbol{x}_t\right)\leq\tilde{J}\left(\bar{\boldsymbol{U}},\boldsymbol{K},\boldsymbol{x}_0,\boldsymbol{x}_t,\beta\right)\right]\geq 1-\beta,\\
&\mathbb{P}\left[\boldsymbol{g}\left(\boldsymbol{x}_k,\boldsymbol{u}_k\right)\preceq\boldsymbol{0}_{N_g}\right]\geq 1-\beta,\ \forall k\in\mathbb{N}_{N-1},\\
&\mathbb{P}\left[\boldsymbol{g}_t\left(\boldsymbol{x}_N,\boldsymbol{x}_t\right)\preceq\boldsymbol{0}_{N_{gt}}\right]\geq 1-\beta,
\end{aligned}
\tag{4}
$$

where $\boldsymbol{x}_0 \sim \mathcal{N}(\bar{\boldsymbol{x}}_0, \boldsymbol{\Sigma}_{\boldsymbol{x},0})$ is the initial distribution, $\boldsymbol{x}_t \sim \mathcal{N}(\bar{\boldsymbol{x}}_t, \boldsymbol{\Sigma}_{\boldsymbol{x},t})$ is the target distribution, and $\boldsymbol{w}_{\boldsymbol{x},k} \sim \mathcal{N}(\boldsymbol{0}_{Nx}, \boldsymbol{\mathcal{Q}}_{\boldsymbol{x},k})$ is the navigation error. Robust solvers' transcription methods are designed to enforce hard constraints with a failure risk lower than a given β. First of all, recall that $\boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ and $\boldsymbol{g}_t(\boldsymbol{x}_N, \boldsymbol{x}_t)$ are Gaussian variables of respective size N_g and N_{gt}. Therefore, findin a transcription method for constraints of the type $\mathbb{P}(\boldsymbol{y} \preceq \boldsymbol{0}_d) \geq 1-\beta$ for any $\boldsymbol{y} \sim \mathcal{N}(\bar{\boldsymbol{y}}, \boldsymbol{\Sigma})$ of size $d \in \mathbb{N}^*$ will allow us to transcribe the constraints of Eq. (4). These methods essentially consist of findin a tractable sufficien condition $\mathcal{A}(\bar{\boldsymbol{y}}, \boldsymbol{\Sigma}, \beta)$: $\mathcal{A}(\bar{\boldsymbol{y}}, \boldsymbol{\Sigma}, \beta) \implies \mathbb{P}(\boldsymbol{y} \preceq \boldsymbol{0}_d) \geq 1-\beta$.

2.2.2 Transcription methods

Certain methods were designed to tackle specifi types of chance-constraints. For instance, Ridderhof et al. [29] transcribed control norm constraints used in Ridderhof et al. [29], Benedikter et al. [2], Marmo and Zavoli [20]. As shown in Caleb et al. [12], this method rests on an spectral radius domination and are not easily transposed to other constraint types. Generic transcriptions exist for any univariate Gaussian distribution y. Blackmore et al. [6] start from a multivariate Gaussian random variable $\boldsymbol{\lambda} \sim \mathcal{N}\left(\bar{\boldsymbol{\lambda}}, \Sigma_\lambda\right)$ of size v and tackle the following chance-constraint:

$$\mathbb{P}\left(\boldsymbol{h}^T \cdot \boldsymbol{\lambda} \leq a\right) \geq 1-\beta \tag{5}$$

where $\boldsymbol{h} \in \mathbb{R}^v$ and $a \in \mathbb{R}$. They provide the following necessary and sufficien condition:

$$\boldsymbol{h}^T \cdot \bar{\boldsymbol{\lambda}} + \Phi_G^{-1}(1-\beta)\sqrt{\boldsymbol{h}^T \boldsymbol{\Sigma}_\lambda \boldsymbol{h}} \leq a \;\Leftrightarrow\; \mathbb{P}\left(\boldsymbol{h}^T \cdot \boldsymbol{\lambda} \leq a\right) \geq 1-\beta \tag{6}$$

where Φ_G^{-1} is the inverse cumulative distribution function (CDF) of a standard Gaussian distribution.

Setting $y = \boldsymbol{h}^T \cdot \boldsymbol{\lambda} - a$ allows us to generalize:

$$\bar{y} + \Phi_G^{-1}(1-\beta)\sqrt{\Sigma} \leq 0 \;\Leftrightarrow\; \mathbb{P}(y \leq 0) \geq 1-\beta \tag{7}$$

This method is widely used in the literature [see 28, 18, 25], though it cannot tackle multidimensional constraints.

2.3 High-Order Taylor Polynomial Algebra

DA is a framework for manipulating high-order Taylor expansions [5]. It represents a differentiable function $\boldsymbol{h}$ of v variables with its Taylor expansion $\mathcal{P}_h$, providing a description of $\boldsymbol{h}$ over its domain. Algebraic and functional operations, including polynomial inversion, are well-define on Taylor polynomials [4]. It has many practical applications such as:

- **Function Approximation**: Uses Taylor-Lagrange's theorem for evaluating functions at points near a given value. It provides a polynomial approximation with bounded error. For instance, evaluating $\sin$ at a point y near x can be done by evaluating the polynomial $\sum_{k=0}^{p} \frac{\sin^{(k)}(x)}{k!}(y-x)^k$ at $\delta x = y - x$, with an absolute error bounded by $\frac{\sin^{(p+1)}(\xi)}{(p+1)!}\delta x^{p+1}$. The run time of this method is faster than standard evaluation for large sampling [1]. This method is extensively used for sampling, Monte Carlo validation, or faster optimization [10, 19].
- **Automatic Differentiation**: Differentiates functions without using finit differences, which are computationally intensive and imprecise, and without computing and implementing derivatives by hand. The derivatives are accurate up to machine precision. The derivatives are simply retrieved from the coefficien of the Taylor polynomials. This property is useful for parameter-dependent functions and quick testing, allowing easy changes in dynamics, constraints, etc. The differential algebra-based differential dynamic programming (DADDy)[1] solver implements this feature, along with function approximation implemented in Caleb et al. [11].
- **Uncertainty Propagation**: Uses Taylor polynomials to approximate how uncertain states propagate over time. Given an initial uncertain state $\mathcal{P}_{\boldsymbol{x},0} = \boldsymbol{x}_0 + \delta\boldsymbol{x}_0$ and a dynamical system $\boldsymbol{f}$ such that $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k)$, the mapping $\mathcal{P}_{\boldsymbol{x},t}$ of the state $\boldsymbol{x}$ at any time t can be computed as a function of $\delta\boldsymbol{x}_0$. This method simulates the fina state and distribution with arbitrary precision using either high-order polynomials or automatic domain splitting (ADS) [33]. Work has been done to specificall represent Gaussian distributions using DA [26, 8, 19].

3 METHODOLOGY

3.1 Chance-Constraints Transcription

This part present two general d–dimensional transcription methods. Let $\boldsymbol{y} \sim \mathcal{N}(\bar{\boldsymbol{y}}, \boldsymbol{\Sigma})$, $\beta \in]0,1[$ the target failure risk, and $\rho(\boldsymbol{\Sigma})$ the positive square root of the spectral radius of $\boldsymbol{\Sigma}$, i. e., its largest eigenvalue. A firs transcription results inspired by the spectral radius domination of Ridderhof et al. [29] is given:

Theorem 1 (SPECTRAL RADIUS TRANSCRIPTION.).

$$\bar{\boldsymbol{y}} + \Psi_d^{-1}(\beta)\,\rho(\boldsymbol{\Sigma}) \preceq \mathbf{0}_d \implies \mathbb{P}(\boldsymbol{y} \preceq \mathbf{0}_d) \geq 1 - \beta \tag{8}$$

where $\Psi_d^{-1}(\beta) = \sqrt{\Phi_d^{-1}(1-\beta)}$, and Φ_d^{-1} is the inverse CDF of a standard χ^2 distribution with d degrees of freedom.

Proof. See Caleb et al. [12]. □

[1] Library available at: `https://github.com/ThomasClb/DADDy.git` [last accessed Feb 20, 2025].

We also provide a second less conservative result:

Theorem 2 (FIRST ORDER TRANSCRIPTION.). *Let $\boldsymbol{\sigma}$ be the vector of the positive square roots of the diagonal coefficients of $\boldsymbol{\Sigma}$, then:*

$$\bar{\boldsymbol{y}} + \Psi_d^{-1}(\beta)\,\boldsymbol{\sigma} \preceq \boldsymbol{0}_d \implies \mathbb{P}(\boldsymbol{y} \preceq \boldsymbol{0}_d) \geq 1-\beta \tag{9}$$

Proof. See Caleb et al. [12]. □

These two new results can be used for constraints and cost function transcription provided that the functions are sufficientl regular and that the covariances are small enough so that the functions are locally linear.

Indeed, if the chance constraint is formulated as $\mathbb{P}(\boldsymbol{h}(\boldsymbol{x},\boldsymbol{u}) \preceq \boldsymbol{0}) \geq 1-\beta$, where $\boldsymbol{h}$ is a linear function of the state and the control that outputs in $\mathbb{R}^d$. Moreover, we defin $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x},\boldsymbol{u})$. If $\delta\boldsymbol{x} = \boldsymbol{x} - \bar{\boldsymbol{x}}$, then:

$$\boldsymbol{y} = \boldsymbol{h}(\bar{\boldsymbol{x}},\bar{\boldsymbol{u}}) + [\boldsymbol{h}_{\boldsymbol{x}} + \boldsymbol{h}_{\boldsymbol{u}} \cdot \boldsymbol{K}] \cdot \delta\boldsymbol{x} \tag{10}$$

where $\boldsymbol{h}_{\boldsymbol{x}}$ and $\boldsymbol{h}_{\boldsymbol{u}}$ are respectively the gradient of $\boldsymbol{h}$ with respect to $\boldsymbol{x}$ and $\boldsymbol{u}$. Note that $\delta\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{x}})$, therefore, since $\boldsymbol{y}$ is a linear combination of Gaussian variables, $\boldsymbol{y} \sim \mathcal{N}(\bar{\boldsymbol{y}}, \boldsymbol{\Sigma})$ with:

$$\bar{\boldsymbol{y}} = \boldsymbol{h}(\bar{\boldsymbol{x}},\bar{\boldsymbol{u}}),\ \boldsymbol{\Sigma} = (\boldsymbol{h}_{\boldsymbol{x}} + \boldsymbol{h}_{\boldsymbol{u}} \cdot \boldsymbol{K}) \cdot \Sigma_x \cdot (\boldsymbol{h}_{\boldsymbol{x}} + \boldsymbol{h}_{\boldsymbol{u}} \cdot \boldsymbol{K})^T \tag{11}$$

Thus, if a given function $\boldsymbol{h}$ is linear, chance constraints of type: $\mathbb{P}(\boldsymbol{h}(\boldsymbol{x},\boldsymbol{u}) \preceq \boldsymbol{0}) \geq 1-\beta$ can now be transcribed.

3.2 Linear Stochastic Optimization solver

In this part we introduce the L-SODA solver that implement our novel transcription methods, and our DADDy deterministic solver [11].

The firs uncertainty source is the error on the initial state $\boldsymbol{x}_0 \sim \mathcal{N}(\bar{\boldsymbol{x}}_0, \boldsymbol{\Sigma}_{\boldsymbol{x},0})$. Then, we consider navigation errors at each time step: $\widehat{\boldsymbol{x}}_{k+1} = \widehat{\boldsymbol{f}}_k + \boldsymbol{w}_{x,k+1}$, where $\boldsymbol{w}_{\boldsymbol{x},k} \sim \mathcal{N}(\boldsymbol{0}_{Nx}, \boldsymbol{\mathcal{Q}}_{\boldsymbol{x},k})$. Therefore, the state $\boldsymbol{x}_k \sim \mathcal{N}(\bar{\boldsymbol{x}}_k, \boldsymbol{\Sigma}_{\boldsymbol{x},k})$ can be updated as follows [2]:

$$\begin{cases} \boldsymbol{\Sigma}_{\boldsymbol{u},k} &= \boldsymbol{K}_k \boldsymbol{\Sigma}_{\boldsymbol{x},k} \boldsymbol{K}_k^T \\ \bar{\boldsymbol{x}}_{k+1} &= \bar{\boldsymbol{f}}_k \\ \boldsymbol{\Sigma}_{\boldsymbol{x},k+1} &= \left(\bar{\boldsymbol{f}}_{k,\boldsymbol{x}} + \boldsymbol{K}_k \bar{\boldsymbol{f}}_{k,\boldsymbol{u}}\right) \cdot \boldsymbol{\Sigma}_{\boldsymbol{x},k} \cdot \left(\bar{\boldsymbol{f}}_{k,\boldsymbol{x}} + \boldsymbol{K}_k \bar{\boldsymbol{f}}_{k,\boldsymbol{u}}\right)^T + \boldsymbol{\mathcal{Q}}_{\boldsymbol{x},k+1} \end{cases} \tag{12}$$

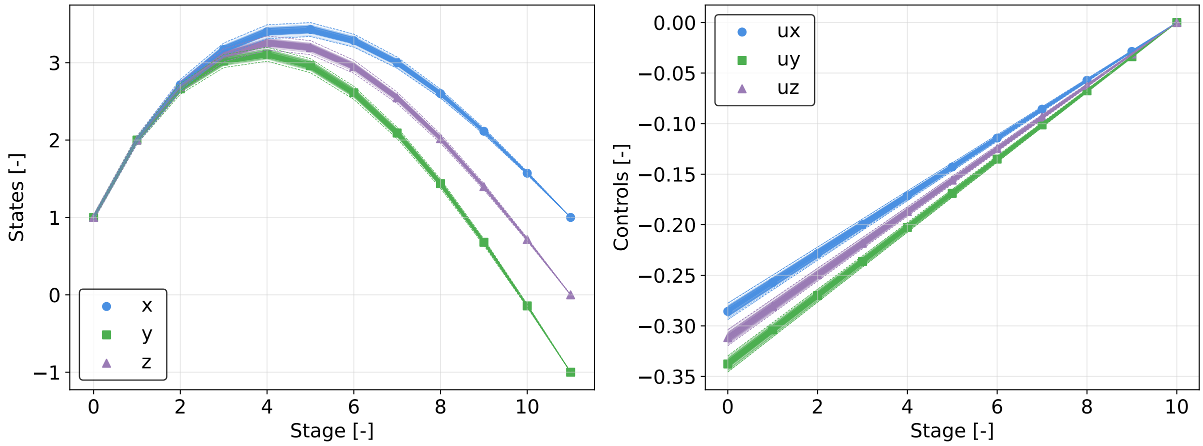
These computations can be done effortlessly using the gradients already computed during the DDP round.

From this, the cost functions l, ϕ, and the constraints $\boldsymbol{g}$ and $\boldsymbol{g}_t$ can be linearized and computed. They are then modifie using the transcription method. For instance, l and $\boldsymbol{g}$ are concatenated together: $\boldsymbol{y} = \left[l(\boldsymbol{x},\boldsymbol{u}), \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u})^T\right]^T \sim \mathcal{N}(\bar{\boldsymbol{y}}, \boldsymbol{\Sigma})$. Let us define

$$\tilde{\boldsymbol{y}} = \bar{\boldsymbol{y}} + \Psi_d^{-1}(\beta)\,\boldsymbol{\sigma} \tag{13}$$

where $d = N_g + 1$. We also set $\tilde{l}$ as the firs component of $\tilde{\boldsymbol{y}}$ and $\tilde{\boldsymbol{g}}$ as the N_g remaining components. Then:

$$\mathbb{P}\left[\left\{l(\boldsymbol{x},\boldsymbol{u}) \leq \tilde{l}(\boldsymbol{x},\boldsymbol{u})\right\} \cap \{\boldsymbol{g}(\boldsymbol{x},\boldsymbol{u}) \leq \tilde{\boldsymbol{g}}(\boldsymbol{x},\boldsymbol{u})\}\right] \geq 1-\beta \tag{14}$$



(a) State vectors.

(b) Controls.

Figure 1: Solution to the double integrator problem.

Thus, replacing l and $\boldsymbol{g}$ by $\tilde{l}$ and $\tilde{\boldsymbol{g}}$, and similarly for ϕ and $\boldsymbol{g}_t$, allows us to solve the stochastic problem of Eq. (4).

Finally, the linearity hypothesis is checked via the non-linearity index (NLI) from Losacco et al. [19]. This index corresponds to the norm of the Hessian divided by the norm of the gradient. A high NLI means the Hessian has similar magnitude as the gradient, i. e., the dynamics is not linear. Having a small NLI shows the second-order terms of the Taylor expansion can be neglected. In other words, the linear hypothesis is valid. This criteria is computed efficiently thanks to previous polynomial computations of the DA framework.

4 RESULTS

4.1 Validation: Double Integrator Problem

The solver developed for this work was validated against the unconstrained double integrator optimization problem from [17]. The optimization problem is define by:

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) &= \left[\boldsymbol{v}^T, \boldsymbol{u}^T\right]^T \\ l(\boldsymbol{x}, \boldsymbol{u}) &= \boldsymbol{u}^T \cdot \boldsymbol{u} \\ \phi(\boldsymbol{x}, \boldsymbol{x}_t) &= (\boldsymbol{r} - \boldsymbol{r}_t)^T \cdot (\boldsymbol{r} - \boldsymbol{r}_t) \end{aligned} \tag{15}$$

where $N_x = 6$, $N_u = 3$, $\boldsymbol{x} = [x, y, z, v^x, v^y, v^z]^T = \left[\boldsymbol{r}^T, \boldsymbol{v}^T\right]^T$, with $\boldsymbol{r}$ the position of size $\frac{N_x}{2}$, and $\boldsymbol{v} = \dot{\boldsymbol{r}}$ the velocity. The number of stages is $N = 11$, the initial conditions and target of the problem are $\boldsymbol{x}_0 = [1, 1, 1, 1, 1, 1]^T$ and $\boldsymbol{x}_t = [1, -1, 0, 0, 0, 0]^T$, the initial covariance is $\boldsymbol{\Sigma}_{\boldsymbol{x},0} = \text{diag}\left(10^{-2}, 10^{-2}, 10^{-2}, 10^{-2}, 10^{-2}, 10^{-2}\right)^2$, and the navigation errors have a covariance $\boldsymbol{\mathcal{Q}}_{\boldsymbol{x},k} = \frac{1}{100}\boldsymbol{\Sigma}_{\boldsymbol{x},0}$. This problem was solved with as firs guess $\boldsymbol{U}_0$: a $N \cdot N_u$ vector with all components equal to 10^{-5}. The result, shown in Fig. 1, is found in a single iteration. Fig. 1a shows the states and Fig. 1b the controls. The dots correspond to the nominal system state or control and the lines correspond to sample trajectories. The results match the ones of [17] and Table 1 provides additional information on the total cost and run time of the method. The two solvers converged towards a similar robust solution, since the problem is convex. However, the run time is higher for the robust method due to significan overhead.

Table 1: Double integrator convergence data.

Solver	Run time [s]	Nominal cost	$(1-\beta)$–quartile cost
Deterministic	5.32×10^{-3}	1.1273	1.3616
Stochastic $\beta = 5\%$	9.73×10^{-3}	1.1273	1.3616

Table 2: Earth-Mars two-body problem transfer initial and target states.

Type	x [km]	y [km]	z [km]	$\dot{x}$ [km/s]	$\dot{y}$ [km/s]	$\dot{z}$ [km/s]	m [kg]
Departure	$-140\,699\,693$	$-51\,614\,428$	980	9.774 596	$-28.078\,28$	$4.337\,725 \times 10^{-4}$	1000
Target	$-172\,682\,023$	176 959 469	7 948 912	$-16.427\,384$	$-14.860\,506$	$9.214\,86 \times 10^{-2}$	-

4.2 Stochastic Earth-Mars Fuel-Optimal Transfer

The solver was then tested against a fuel-optimal low-thrust Earth-Mars transfer optimization problem adapted from [17]:

$$
\begin{aligned}
\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) &= \boldsymbol{x} + \int_0^{\Delta t} \left[\boldsymbol{v}^T, \dot{\boldsymbol{v}}^T, \dot{m}\right]^T dt \\
\dot{\boldsymbol{v}} &= -\frac{\mu}{\|\boldsymbol{r}\|_2^3}\boldsymbol{r} + \frac{\boldsymbol{u}}{m} \\
\dot{m} &= -\frac{\|\boldsymbol{u}\|_2}{g_0 \cdot \text{Isp}} \\
l(\boldsymbol{x}, \boldsymbol{u}) &= \|\boldsymbol{u}\|_2 \\
\phi(\boldsymbol{x}, \boldsymbol{x}_t) &= (\boldsymbol{r} - \boldsymbol{r}_t)^T \cdot (\boldsymbol{r} - \boldsymbol{r}_t) + (\boldsymbol{v} - \boldsymbol{v}_t)^T \cdot (\boldsymbol{v} - \boldsymbol{v}_t) \\
\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) &= \left[\boldsymbol{u}^T \cdot \boldsymbol{u} - u_{\max}^2, m_{\text{dry}} - m\right]^T
\end{aligned}
\tag{16}
$$

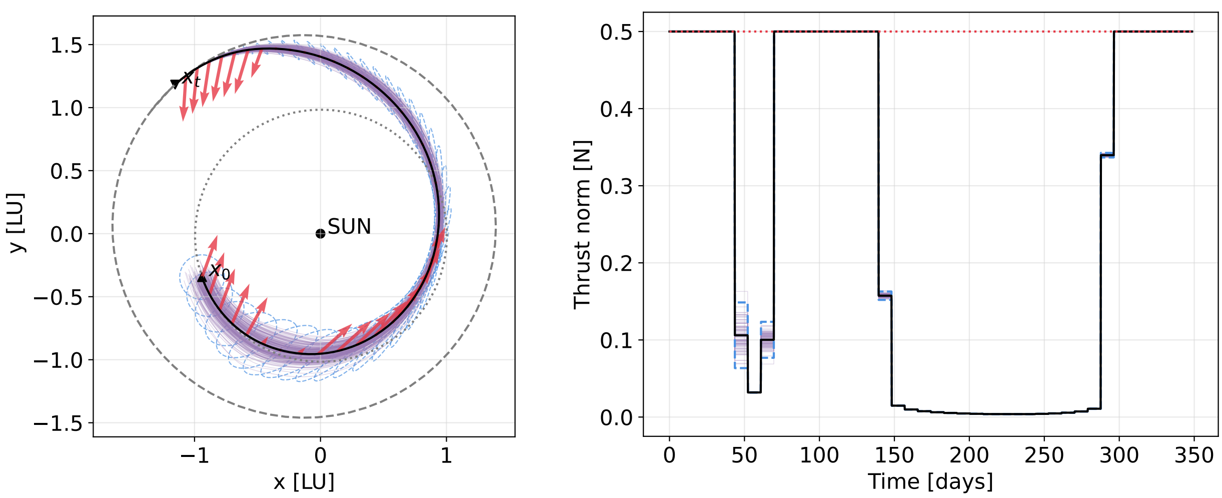
where $N_x = 7$, $N_u = 3$, the state vector $\boldsymbol{x}$ can be written: $[x, y, z, \dot{x}, \dot{y}, \dot{z}, m]$ or $\left[\boldsymbol{r}^T, \boldsymbol{v}^T, m\right]^T$, and m is the spacecraft mass. The number of stages is $N = 40$, the time-of-fligh (ToF) is $348.79\,\text{d}$, $\Delta t = \frac{\text{ToF}}{N}$, the initial conditions and target are given in Table 2. The initial covariance is $\boldsymbol{\Sigma}_{\boldsymbol{x},0} = \text{diag}\left(10^{-6}, 10^{-6}, 10^{-6}, 5 \times 10^{-6}, 5 \times 10^{-6}, 5 \times 10^{-6}\right)^2$, and the navigation errors have a covariance $\boldsymbol{\mathcal{Q}}_{\boldsymbol{x},k} = \frac{1}{100}\boldsymbol{\Sigma}_{\boldsymbol{x},0}$. These errors were selected such that the highest value of the NLI is small (around 10^{-4}).

Normalization units and various dynamics parameters are reported in Table 3. All mass parameters in this work were obtained from JPL DE431 ephemerides [2] [13]. The firs guess $\boldsymbol{U}_0$ is a $N \cdot N_u$ vector with all components equal to 10^{-6} N. Fig. 2 shows the solution to this optimization problem. Fig. 2a shows the trajectory from Earth to Mars with errors magnifie $\times 50\,000$. The nominal trajectory is in black, the red arrows represent the thrust vectors, the theoretical $(1-\beta)$–ellipsoids are shown in blue and the purple lines each represent a different sample. Fig. 2b represents the control norm with errors magnifie $\times 200$. The black line is the nominal control, the dotted red line is the control maximum magnitude $u_{\max}$, the dashed blue line represents the $(1-\beta)$–ellipsoids (two lines in 1D), and the purple lines correspond to the controls of the samples. These plots show the random samples follow a control law similar to the nominal one, yet it is clear the stochastic maneuvers are performed at the transition between

[2] Publicly available at: `https://naif.jpl.nasa.gov/pub/naif/generic_kernels/pck/gm_de431.tpc` [retrieved on Feb 20, 2025].

Table 3: Sun-centered normalization units and dynamics parameters.

Type	Unit	Symbol	Value	Comment
Normalization units	Mass parameter [-]	μ	$1.327\,124\,400\,41 \times 10^{11}$	Sun's mass parameter
	Length [km]	LU	149 597 870.7	Astronomical unit of length
	Time [s]	TU	5 022 642.891	$(\mathrm{LU}^3/\mu)^{1/2}$
	Velocity [km/s]	VU	29.784 691 83	LU/TU
Parameters	Standard gravity [m/s^2]	g_0	9.81	
	Specifi impulse [s]	Isp	2000	
	Spacecraft dry mass [kg]	m_{dry}	500	
	Maximum spacecraft thrust [N]	$u_{\max}$	0.5	



(a) Trajectory in the x–y plane.

(b) Thrust profile

Figure 2: Solution to the stochastic fuel-optimal Earth-Mars transfer.

Table 4: Stochastic fuel-optimal Earth-Mars transfer convergence data for $\beta = 5\%$.

Solver	**Run time** [s]	**Nominal error [-]**	$(1-\beta)$**–cost** [kg]	**Failure rate [%]**	**Conservatism**
Deterministic	45.9	4×10^{-13}	396.9	99.6	–
Stochastic SR	69.2	-6×10^{-6}	397.6	$<3 \times 10^{-3}$	>1700
Stochastic FO	45.9	-1×10^{-10}	396.9	2.93	1.7

the nominal thrust and cruise arcs. The covariance is at its largest at the beginning of the transfer and shrinks when performing these correction maneuvers.

Quantitative results comparing the deterministic solution of this problem and the stochastic solving using the spectral radius transcription (SR) and the firs order transcription (FO) are shown in Table 4. The $(1-\beta)$–cost, the real failure rate, and the conservatism were computed with Monte-Carlo (MC) sampling of size 10^5. It is clear that deterministic solving leads to unsafe solutions when adding uncertainties. Comparing the two transcription methods, it appears that the spectral radius method is highly conservative. Indeed, the nominal is far below 0 while the firs order method is four orders of magnitude closer. This leads to an extra fuel consumption of $0.7\,\text{kg}$ of fuel, which is non-negligible considering the small magnitude of the errors. In addition, due to the repeated computation of eigenvalues of matrices with poor conditioning, the solver struggles to converge more than with the firs order method that appears smoother. It results in a 51% longer runtime. On the other hand, the firs order method has the same runtime and $(1-\beta)$–cost as the deterministic method and a conservatism below 1.7 while having a satisfactory failure rate.

5 CONCLUSION

This work presented the L-SODA solver, a novel stochastic solver that uses synergies between applications of DA. It uses automatic differentiation, dynamics approximation, and covariance propagation to efficientl compute robust control laws while maintaining a reasonable runtime.

Two novel transcription methods were tested: a spectral radius method, adapted from existing approaches, and a novel first-orde method designed for enforcing robustness while minimizing deviations from optimality. The firs order method shows demonstrates strong robustness, achieving a failure rate below the specifie threshold with a runtime of under a minute. Moreover, the fuel consumption is similar to the non-robust solver. These results highlight the potential of L-SODA for robust trajectory design.

FUNDING SOURCES

This work was funded by SaCLaB (grant number 2022-CIF-R-1), a research group of ISAE-SUPAERO.

References

[1] 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*, 107(4):451–470, 6 2010. doi: 10.1007/s10569-010-9283-5.

[2] 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, 45(11):2061–2075, 2022. doi: 10.2514/1.G006806. URL https://doi.org/10.2514/1.G006806.

[3] J. Benkhoff, G. Murakami, W. Baumjohann, S. Besse, E. Bunce, M. Casale, G. Cremosese, K.-H. Glassmeier, H. Hayakawa, D. Heyner, H. Hiesinger, J. Huovelin, H. Hussmann, V. Iafolla, L. Iess, Y. Kasaba, M. Kobayashi, A. Milillo, I. G. Mitrofanov, E. Montagnon, M. Novara, S. Orsini, E. Quemerais, U. Reininghaus, Y. Saito, F. Santoli, D. Stramaccioni, O. Sutherland, N. Thomas, I. Yoshikawa, and J. Zender. Bepicolombo - mission overview and science goals. *Space Science Reviews*, 217(8), Dec. 2021. ISSN 1572-9672. doi: 10.1007/s11214-021-00861-4.

[4] M. Berz. High-order computation and normal form analysis of repetitive systems. In *AIP Conference Proceedings*, pages 1–36, 1992. doi: 10.1063/1.41975.

[5] M. Berz. *Modern Map Methods in Particle Beam Physics*, chapter 2. Elsevier, 1999. doi: 10.1016/s1076-5670(08)70227-1.

[6] L. Blackmore, M. Ono, and B. C. Williams. Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics*, 27(6):1080–1094 , Dec. 2011. ISSN 1941-0468. doi: 10.1109/tro.2011.2161160.

[7] S. Boone and J. Mcmahon. Rapid local trajectory optimization using higher-order state transition tensors and differential dynamic programming. In *2020 AAS/AIAA Astrodynamics Specialist Conference*, number AAS 20-582, Aug. 2020.

[8] S. Boone and J. McMahon. Non-gaussian chance-constrained trajectory control using gaussian mixtures and risk allocation. In *IEEE 61st Conference on Decision and Control (CDC)*, pages 3592–3597, 2022.

[9] A. Boutonnet, Y. Langevin, and C. Erd. Designing the juice trajectory. *Space Science Reviews*, 220(6), Sept. 2024. ISSN 1572-9672. doi: 10.1007/s11214-024-01093-y.

[10] T. Caleb, R. Armellin, and S. Lizy-Destrez. Fast gradient evaluation and mapping method for bi-impulsive transfers within periodic orbits. *Journal of Guidance, Control, and Dynamics*, 0(0):1–9, 2024. doi: 10.2514/1.G008038. URL https://arc.aiaa.org/doi/abs/10.2514/1.G008038.

[11] T. Caleb, R. Armellin, S. Boone, and S. Lizy-Destrez. A polynomial-based constrained solver for fuel-optimal low-thrust trajectory optimization. 2025. URL https://doi.org/10.48550/ARXIV.2502.00398.

[12] T. Caleb, R. Armellin, S. Boone, and S. Lizy-Destrez. Chance constraints transcription and failure risk estimation for stochastic trajectory optimization. 2025. URL https://doi.org/10.48550/arXiv.2502.15949.

[13] W. Folkner, J. Williams, D. Boggs, R. Park, and P. Kuchynka. The Planetary and Lunar Ephemerides DE430 and DE431. Technical Report 42-196, Jet Propulsion Laboratory, California Institute of Technology, 2014.

[14] C. Greco, S. Campagnola, and M. Vasile. Robust space trajectory design using belief optimal control. *Journal of Guidance, Control, and Dynamics*, 45(6):1060–1077, June 2022. ISSN 1533-3884. doi: 10.2514/1.g005704.

[15] T. A. Howell, B. E. Jackson, and Z. Manchester. Altro: A fast solver for constrained trajectory optimization. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–6, 2019. doi: 10.1109/IROS40897.2019.8967788. URL `https://doi.org/10.1109/IROS40897.2019.8967788`.

[16] G. Lantoine and R. P. Russell. A hybrid differential dynamic programming algorithm for constrained optimal control problems. part 1: Theory. *Journal of Optimization Theory and Applications*, 154(2):382–417, 4 2012. doi: 10.1007/s10957-012-0039-0. URL `https://doi.org/10.1007/s10957-012-0039-0`.

[17] G. Lantoine and R. P. Russell. A hybrid differential dynamic programming algorithm for constrained optimal control problems. part 2: Application. *Journal of Optimization Theory and Applications*, 154(2):418–442, 8 2012. doi: 10.1007/s10957-012-0038-1. URL `https://doi.org/10.1007/s10957-012-0038-1`.

[18] T. Lew, R. Bonalli, and M. Pavone. Chance-constrained sequential convex programming for robust trajectory optimization. In *2020 European Control Conference (ECC)*, pages 1871–1878. IEEE, May 2020. doi: 10.23919/ecc51009.2020.9143595.

[19] M. Losacco, A. Fossà, and R. Armellin. Low-order automatic domain splitting approach for nonlinear uncertainty mapping. *Journal of Guidance, Control, and Dynamics*, 47(2): 291–310, Feb. 2024. ISSN 1533-3884. doi: 10.2514/1.g007271.

[20] N. Marmo and A. Zavoli. Chance-constraint method for covariance control of low-thrust interplanetary missions. In *AIAA SCITECH 2024 Forum*. American Institute of Aeronautics and Astronautics, Jan. 2024. doi: 10.2514/6.2024-0630.

[21] N. Marmo, A. Zavoli, N. Ozaki, and Y. Kawakatsu. A hybrid multiple-shooting approach for covariance control of interplanetary missions with navigation errors. In *33rd AAS/AIAA Space Flight Mechanics Meeting*, number AAS 23-208, 01 2023.

[22] D. Mayne. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *International Journal of Control*, 3(1):85–95, 1966. doi: 10.1080/00207176608921369. URL `https://doi.org/10.1080/00207176608921369`.

[23] K. Oguri and J. W. McMahon. Stochastic primer vector for robust low-thrust trajectory design under uncertainty. *Journal of Guidance, Control, and Dynamics*, 45(1):84–102, Jan. 2022. ISSN 1533-3884. doi: 10.2514/1.g005970.

[24] 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*, 41(2):377–387, 2018. doi: 10.2514/1.G002367. URL `https://doi.org/10.2514/1.G002367`.

[25] N. Ozaki, S. Campagnola, and R. Funase. Tube stochastic optimal control for nonlinear constrained trajectory optimization problems. *Journal of Guidance, Control, and Dynamics*, 43(4):645–655, 2020. doi: 10.2514/1.G004363. URL `https://doi.org/10.2514/1.G004363`.

[26] R. S. Park and D. J. Scheeres. Nonlinear semi-analytic methods for trajectory estimation.

Journal of Guidance, Control, and Dynamics, 30(6):1668–1676, Nov. 2007. ISSN 1533-3884. doi: 10.2514/1.29106.

[27] H. Poincaré. *Les Méthodes Nouvelles de la Mécanique Céleste*. Gauthier-Villars, Paris, France, 1892.

[28] J. Ridderhof, K. Okamoto, and P. Tsiotras. Nonlinear uncertainty control with iterative covariance steering. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3484–3490. IEEE, Dec. 2019. doi: 10.1109/cdc40024.2019.9029993.

[29] J. Ridderhof, J. Pilipovsky, and P. Tsiotras. Chance-constraints covariance control for low-thrust minimum-fuel trajectory optimization. In *AIAA/AAS Astrodynamics Specialist Conference*, pages 1–20, 2020.

[30] M. Smith, D. Craig, N. Herrmann,E. Mahoney, J. Krezel, N. McIntyre, and K. Goodliff. The Artemis Program: An Overview of NASA's Activities to Return Humans to the Moon. In *2020 IEEE Aerospace Conference*, pages 1–10, 2020. doi: 10.1109/AERO47225.2020.9172323.

[31] Y. L. Tong. *The Multivariate Normal Distribution*, chapter 3. Springer-Verlag New York Inc., 1990. doi: 10.1007/978-1-4613-9655-0. URL `https://doi.org/10.1007/978-1-4613-9655-0`.

[32] F. Topputo and E. Belbruno. Earth–mars transfers with ballistic capture. *Celestial Mechanics and Dynamical Astronomy*, 121(4):329–346, Feb. 2015. ISSN 1572-9478. doi: 10.1007/s10569-015-9605-8.

[33] A. Wittig, P. Di Lizia, R. Armellin, K. Makino, F. Bernelli-Zazzera, and M. Berz. Propagation of large uncertainty sets in orbital dynamics by automatic domain splitting. *Celestial Mechanics and Dynamical Astronomy*, 122(3):239–261, 5 2015. doi: 10.1007/s10569-015-9618-3.