Journal of the Franklin Institute, Volume 361, Issue 8, May 2024, Article number 106849
DOI: 10.1016/j.jfranklin.2024.106849

Convex-concave Optimization for a Launch Vehicle Ascent Trajectory with Chance Constraints

Xin Sun[a], Senchun Chai[b,*], Runqi Chai[b], Baihai Zhang[b], Leonard Felicetti[c], Antonios Tsourdos[c]

[a] *College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, PR China*
[b] *School of Automation, Beijing Institute of Technology, Beijing 100081, PR China*
[c] *School of Aerospace, Transport and Manufacturing, Cranfield University, Bedfordshire, MK43 0AL, United Kingdom*

Abstract

The objective of this paper is to present a convex-concave optimization approach for solving the problem of a multistage launch vehicle ascent trajectory. The proposed method combines convex-concave decomposition and successive linearization techniques to generate a new sequence of convex subproblems to replace the original non-convex problem. Bernstein approximation is used to transform the chance constraints into convex ones. A *hp*-adaptive pseudospectral scheme is employed to discretize the optimal control problem into a nonlinear programming problem with less computation cost. The performance of the proposed strategy is compared against other typical techniques in a selection of test case scenarios. Numerical results demonstrate the viability of the method and show pros and cons of the proposed technique.

Keywords: Trajectory Optimization, Convex-concave Decomposition, Chance Constraints.

*Corresponding author
Email addresses: x_sun@cauc.edu.cn (Xin Sun), chaisc97@163.com (Senchun Chai), r.chai@cranfield.ac.uk (Runqi Chai), smczhang@bit.edu.cn (Baihai Zhang), Leonard.Felicetti@cranfield.ac.uk (Leonard Felicetti), a.tsourdos@cranfield.ac.uk (Antonios Tsourdos)

Published by Elsevier. This is the Author Accepted Manuscript issued with: Creative Commons Attribution Non-Commercial No Derivatives License (CC:BY:NC:ND 4.0).
The final published version (version of record) is available online at DOI:10.1016/j.jfranklin.2024.106849. Please refer to any applicable publisher terms of use.

1. Introduction

The research on trajectory optimization of launch vehicles has always been of interest to researchers in the field of aerospace industry. The design of optimal ascent trajectories strongly affects the overall performance and capabilities of space missions: an optimized trajectory often translates into a reduction of the propellant consumption and, therefore into an increase of the payload mass of the launched space system[1]. On the other hand, the optimization problem is also subjected to several constraints which strongly limit the search space where the optimal solution can be found[2].

The trajectory optimization problem can be considered as an optimal control problem with mission requirements and complex constraints that need to be satisfied. Traditional optimal control methods to achieve the above goals use numerical methods [3, 4], such as direct methods [5] and indirect methods [6] or, evolutionary methods[7, 8], such as particle swarm optimization methods [9] and genetic algorithms [10].

Indirect methods solve the boundary value problem by using classical finite differences approaches, but the derivation of first-order necessary conditions might be difficult [11]. Indeed, because of the nature of the constraints, the ascent trajectory optimization problem is generally formulated as a Non-Linear Programming (NLP) problem which is categorized as a direct method [3]. In literature, the direct method is generally used to solve optimization problems in several different scenarios involving launch vehicles, as this does not require calculating the first-order necessary conditions of optimality. However, to solve the NLP problem, a certain amount of computational time is required, because of the non-deterministic nature of the polynomial formulation which transcends from prior determinable constraints [12]. In some cases, this leads to convergence problems that compromise the possibility of obtaining fast and reliable solutions in the first instance [11].

In recent years, convex programming techniques have been introduced and widely used to solve optimization problems formulated via direct method. These

optimization techniques allow for a fast and easy calculation of optimal solutions by using predetermined polynomial time-dependent functions without any particular requirement on initial guesses used for triggering the numerical optimization [13, 14]. However, such strategies require that non-convex constraints can be turned into convex ones. This is done mainly through two different approaches. A lossless approach consists of relaxing the constraints to obtain a convex set of feasible solutions that contain the optimal one [15]. Another approach is to get a convex subset of feasible solutions through successive linearization of the constraints.

Traditional lossless convexification transforms non-convex problems into convex equivalents using auxiliary variables, while convex-concave decomposition decomposes problems into convex and concave subproblems for iterative optimization. The convex-concave decomposition method efficiently addresses certain non-convex problems by iteratively solving convex and concave subproblems, often leading to faster convergence and feasible local solutions, while handling specific structural constraints. Both methods are widely used in literature. In the review paper [16], the author reviews the trajectory optimization problem of different vehicle platforms using convex optimization methods, where both approaches to handle non-convex constraints are used. In [1], a convex method was proven to be lossless for solving the control constraints and a primal-dual interior-point method was employed to solve the optimization problem. Ref [14] compares different discretization and trust-region methods for low-thrust trajectory optimization, assessing their performance and suitability for onboard guidance through interplanetary transfers. Ref [17] proposes a novel obstacle decomposition approach and a safe convex corridor construction method to address the nonconvex collision avoidance constraints in optimization-based motion planning for nonholonomic vehicles, demonstrating their feasibility and superiority over existing methods through simulation experiments. Ref [18] proposes an online convex optimization method to transform the exoatmospheric ascent trajectory optimization problem into a sequence of convex subproblems.

However, such an approach could not convexify the highly non-linear differ-

ential equation constraints that characterized the dynamic model of the landing scenario. For this reason, the lossless methods are often used in conjunction with the successive linearization methods. For instance, Benedikter proposed a convex optimization method which combined a lossless approach and successive linearization to solve the ascent of the launch vehicle trajectory under a set of complex constraints.

The control constraints were handled via a lossless method, and then the rest of the nonconvex constraints were transformed into convex ones by successive linearization[19]. In order to provide a better and more efficient computation, a three-step continuation procedure was also proposed [19]. Li and Pang also used the lossless approach and successive linearization method to replan a reference solution such that different terminal constraints could be considered due to eventual power system fault conditions [20]. Apart the references already cited above, where the successive linearization method was used to solve the optimal ascent trajectory problem, the same method was applied to a wider variety of scenarios such as hypersonic boost-glide vehicle ascent trajectory design problem [21], aircraft trajectory optimization [22], aeroassisted vehicles trajectory design [23].

In view of the successful application of convex optimization methodologies in the above fields, this paper investigates a new convex method for solving a multistage launch vehicle ascending problem. The aim is to maximize the final payload mass, and therefore minimize the propellant mass by planning for an optimal trajectory from a specific initial launch position on the Earth. Considering the effect of the atmosphere, the optimization algorithm is supposed to find specific ascent trajectories for each of the phases and stages of the launch leading to a multi-phase optimal control problem formulation. The model considers a full three-dimensional scenario with complex nonlinear constraints and chance constraints that are then transformed via a thoughtful convexification algorithm. The convex-concave decomposition method and successive linearization techniques are employed alongside an *hp*-adaptive pseudospectral process, to obtain a discrete convex formulation of the optimal problem.

1.1. Paper's contributions

This manuscript's primary contribution stands on the application of the convex-concave decomposition method to the multistage launch vehicle ascent problem. Specifically,

1. the work establishes a noise-perturbed multi-stage launch vehicle trajectory optimization model by including multiple ascent phases and uncertainty in the control process, throughout the introduction of chance constraints.
2. the work proves the effectiveness of the convexification method based on the decomposition of constraints in convex and concave sub-constraints.
3. the work shows that a reduction of the computational cost can be achieved by combining the convex-concave decomposition method and successive linearization method to turn non-linear constraints into convex ones.
4. The proposed method has been extended to handle chance constraints through a Bernstein approximation approach.

1.2. Paper's outline

In the remainder of this work, Section II defines the optimization problem and the dynamic model of the launch vehicle, including uncertainties on the thrust throttle through chance constraints. The convexification of the optimization problem is carried on in Section III, where the convex-concave methodology is described alongside the Successive Linearization method. The same section also describes the method for handling chance constraints via the Bernstein approximation method. Section IV details the discretization method which transforms the continuous problem into a NLP problem. Section V assesses the viability of the proposed methodology and compares it with other two standard optimization techniques. In addition, the proof of the viability of Bernstein approximation to handle constraints with uncertainties is carried on throughout the numerical results in a selected launch scenario. Finally, Section VI summarizes the main achievements and conclusions obtained throughout this investigation.

2. Problem Formulation

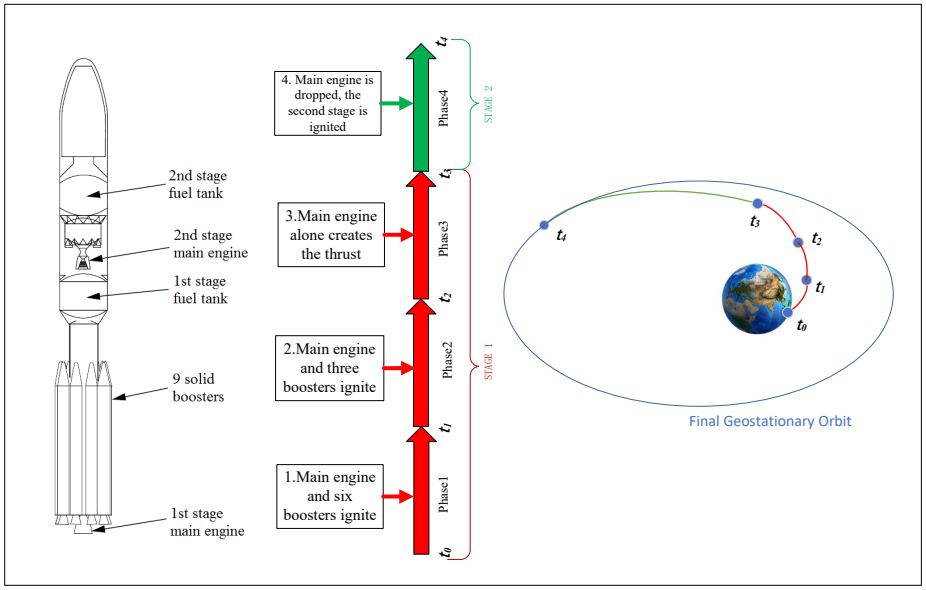


Figure 1: Multi-stage ascent process of a launch vehicle.

The scenario considered in this work is the multi-stage ascent of a launch vehicle. A trajectory optimization problem is formulated to minimize the fuel consumption to reach a desired target orbit. Figure 1 shows the multi-stage ascent phases that characterize the launch process. The whole process includes two stages, which are divided into four distinct phases, as shown in Figure 1. At time t_0, the main engine and six of the nine solid boosters ignite. The boosters are depleted at time t_1 and their remaining mass is discarded. The final three boosters are then ignited at the same time t_1, helping the main engine produce the required thrust for phase 2. At time t_2, the fuel of the remaining three boosters is depleted and they can consequently be jettisoned. Thus, only the main engine provides the thrust for the third phase, from t_2 to t_3. The second stage is ignited when the main engine fuel is exhausted and the dry mass of the main engine is jettisoned at time t_3. The payload is dropped off from the launch

vehicle when the target orbit has been reached, at the final time t_4.

2.1. Dynamic Model

We assume that the launch vehicle is modelled as a point mass flying over a spherical rotating Earth. For short-duration and low-velocity flights, the Coriolis force is neglected for simplicity. Therefore, the equations of motions can be written in a Cartesian Earth-Centered Inertial (ECI) reference frame as:

$$\begin{aligned} \dot{\boldsymbol{r}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= -\frac{\mu}{\|\boldsymbol{r}\|^3}\boldsymbol{r} + \frac{T}{m}\hat{\boldsymbol{u}} + \frac{\boldsymbol{D}}{m} \\ \dot{m} &= -\frac{T}{g_0 I_{sp}} \end{aligned} \tag{1}$$

where $\boldsymbol{r} = [x(t), y(t), z(t)]^T$ is the position and $\boldsymbol{v} = [v_x(t), v_y(t), v_z(t)]^T$ is the velocity of the launcher with respect to the ECI reference frame. The control vector is denoted as $\hat{\boldsymbol{u}} = [u_x, u_y, u_z]^T$, while μ and T represent the gravitational parameter of the Earth and the thrust magnitude due to the launcher's propulsion system, respectively. Specifically, the major changes in the thrust magnitude are happening during the transition between four phases of the ascent process. The aerodynamic drag $\boldsymbol{D}$ is defined as $\boldsymbol{D} = -\frac{1}{2}C_D A_r \rho \|\boldsymbol{v}_r\| \boldsymbol{v}_r$, where C_D and A_r are the drag coefficient and the reference area of the launcher, respectively. The model for the atmosphere density ρ, assumes a exponential decay with the altitude h as follows: $\rho = \rho_0 \exp[-h/h_0]$, where ρ_0 and h_0 are the density and the density scale height at the sea level, respectively. The altitude above Earth can be calculated as: $h = \|\boldsymbol{r}\| - R_e$, where R_e is the equatorial radius. The model assumes that the atmosphere is rigidly rotating in the same fashion as the Earth's rotation and consequently, the relative velocity of the launcher concerning the atmosphere $\boldsymbol{v}_r$ is given as: $\boldsymbol{v}_r = \boldsymbol{v} - \boldsymbol{\omega} \times \boldsymbol{r}$, where $\boldsymbol{\omega}$ is the angular velocity (rotation rate) of the Earth. In Eq. 1 the terms g_0 and I_{sp} stand for the acceleration due to gravity at sea level and the specific impulse of the engine, respectively.

The goal of the present work is to find an optimal ascent trajectory that maximizes the final mass injected in the desired orbit. In other words, we aim

to determine the optimal control $\hat{\boldsymbol{u}}^*$ such that the following objective function is minimized:

$$J = -m(t_f) \tag{2}$$

The control vector $\hat{\boldsymbol{u}}$ and the state vehicle's altitude $\boldsymbol{r}$ should also satisfy the following path constraints:

$$\|\hat{\boldsymbol{u}}\|^2 = u_x^2 + u_y^2 + u_z^2 = 1 \tag{3a}$$

$$\|\boldsymbol{r}\| = \sqrt{x^2 + y^2 + z^2} \geq R_e \tag{3b}$$

where both the control and state vectors are expressed in the ECI frame. The constraint (3a) indicates that the thrust direction is a unit vector, thus the squares of components of the control action need to sum up to 1. The state path constraint given by (3b) makes sure that the vehicle will keep the altitude above the surface of the Earth.

Besides the equality and inequality constraints, boundary conditions should also be considered. This includes the initial and terminal conditions as well as the intermediate conditions at different connection points between phases. To simplify, the state vector can be expressed as $\boldsymbol{x} = [x, y, z, v_x, v_y, v_z, m]^T$. According to this notation, we use $\boldsymbol{x}_{t_0}$ to represent the initial conditions in the Cartesian space. On the other hand, terminal conditions are generally given in terms of orbital elements $\mathbf{o}_f = [a_f, e_f, i_f, \Omega_f, \omega_f]^T$, where a_f is the final semi-major axis, e_f is the final eccentricity, i_f is the final inclination, Ω_f is the terminal right ascension of the ascending node, and ω_f is the terminal argument of perigee. Thus, to assess that the trajectory reaches the final desired orbit, we need to transform the final state $\boldsymbol{x}(t_f)$ obtained from the optimizer into orbital elements $\boldsymbol{o}(t_f)$ and then compare them with desired ones $\mathbf{o}_f$. The procedure for converting the final position and velocity of the launcher in orbital elements can be found in [24]. Thus, the following final state constraints can be added to the model:

$$\mathbf{o}(t_f) - \mathbf{o}_f = \mathbf{0} \tag{4}$$

Moreover, each of the four phases in the trajectory should be linked to the

adjoining phases by a set of linkage conditions. The following constraints guarantee that the four phases in the launch are connected:

$$m_p(t_f^p) - m_p^{drop} - m_{p+1}(t_0^p) = 0 \tag{5a}$$

$$\boldsymbol{v}_p(t_f^p) - \boldsymbol{v}_{p+1}(t_0^p) = \mathbf{0} \qquad p = 1, ...3 \tag{5b}$$

$$\boldsymbol{r}_p(t_f^p) - \boldsymbol{r}_{p+1}(t_0^p) = \mathbf{0} \tag{5c}$$

where m_p^{drop} is the dry mass drop (excluding fuel) at the ends of each phase.

2.2. Problem formulation without uncertainty on the thrust magnitude

According to what was defined in the previous part of the section, launch vehicle dynamics, path constraints, final state constraints, and the objective function, the overall launch vehicle trajectory optimization problem (Problem 1) can be formulated as:

$$\min \quad J = -m(t_f) \tag{6a}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{t}) \tag{6b}$$

$$\|\hat{\boldsymbol{u}}\|^2 = u_x^2 + u_y^2 + u_z^2 = 1, \tag{6c}$$

$$\|\boldsymbol{r}\| = \sqrt{x^2 + y^2 + z^2} \geq R_r, \tag{6d}$$

$$m_p(t_f^p) - m_p^{drop} - m_{p+1}(t_0^p) = 0, \qquad p = 1, 2, 3 \tag{6e}$$

$$\boldsymbol{v}_p(t_f^p) - \boldsymbol{v}_{p+1}(t_0^p) = \mathbf{0}, \qquad p = 1, 2, 3 \tag{6f}$$

$$\boldsymbol{r}_p(t_f^p) - \boldsymbol{r}_{p+1}(t_0^p) = \mathbf{0}, \qquad p = 1, 2, 3 \tag{6g}$$

$$\boldsymbol{o}(t_f) - \boldsymbol{o}_f = \mathbf{0} \tag{6h}$$

where Eqs. (6b) represents the original differential equality constraints in Eq. 1. Please note that in this first formulation of the Problem, we consider that the thrust magnitude $T_p(t)$ follows a predefined envelope for each of the phases $p = 1, \ldots, 4$. In other words, to validate only the convex-concave method we have preferred assuming no uncertainties on the thrust magnitude. This condition is then removed in the subsection below when we introduce and assess the viability of using chance constraints for handling the eventual uncertainties on the thrust magnitude.

2.3. Problem formulation with uncertainty on the thrust magnitude

The inclusion of chance constraints allows us to account for uncertainties in the dynamic model of the launch vehicle ascent trajectory. By considering these uncertainties, such as measurement noise or model inaccuracies, we aim to design a trajectory that remains feasible and safe under varying conditions. The term 'noise-perturbed' refers to the introduction of these uncertainties in the control inputs. We model these perturbations as random variables or describe them using a probability distribution. The use of chance constraints enables us to strike a balance between achieving optimal performance and ensuring the reliability of the system in the presence of uncertainties. Chance constraints account for the eventual uncertainties in the formulation and relax the constraint in such a way that the optimization problem can still have feasible solutions. The chance constraint ensures that the probability of meeting specific constraints is above a certain threshold:

$$P\{\boldsymbol{g}_{min} \leq \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, t; \boldsymbol{\xi}) \leq \boldsymbol{g}_{max}\} \geq 1 - \alpha \tag{7}$$

where $P(\cdot)$ denotes the probability that the set of joint constraints $\boldsymbol{g}_{min} \leq \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, t; \boldsymbol{\xi}) \leq \boldsymbol{g}_{max}$ are verified and α is a parameter that defines a "reliability level" of the occurrence. $\boldsymbol{\xi}$ is the random variable which is assumed to be within $\mathbb{R}^g$ where g is the dimension of $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, t; \boldsymbol{\xi})$ and its probability density function (PDF) is also known. Here, $\boldsymbol{g} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \times \mathbb{R}^g \to \mathbb{R}^{n_g}$, where the state is $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ and the control vector is $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$.

In the specific scenario of trajectory optimization for the above-defined launcher, the magnitude of the thrust, produced by the different engines during the ascent phases, is only known within a certain degree of accuracy. Indeed, the thrust might vary due to different environmental conditions as well as the different performances of the single engine units[25]. Such uncertainty can be modeled as a set of chance constraints as follows:

$$P\{T_p^{min} \leq T_p(t) + \boldsymbol{\xi} \leq T_p^{max}\} \geq 1 - \alpha \qquad p = 1, ...3. \tag{8}$$

in which α is the permissible risk factor. The $\boldsymbol{\xi}$ represents the uncertainties on the thrust magnitude and it has a specific known PDF.

Please note that Problem 1 considers that the thrust magnitude $T_p(t)$ is deterministically known. On the other hand, if we consider that the thrust magnitude is affected by uncertainties, the following Problem 2 should be considered:

$$\min \quad J = -m(t_f) \tag{9a}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{t}) \tag{9b}$$

$$\|\hat{\boldsymbol{u}}\|^2 = u_x^2 + u_y^2 + u_z^2 = 1, \tag{9c}$$

$$\|\boldsymbol{r}\| = \sqrt{x^2 + y^2 + z^2} \geq R_r, \tag{9d}$$

$$m_p(t_f) - m_p^{drop} - m_{p+1}(t_0) = 0, \qquad p = 1, 2, 3. \tag{9e}$$

$$\boldsymbol{v}_p(t_f) - \boldsymbol{v}_{p+1}(t_0) = \boldsymbol{0}, \qquad p = 1, 2, 3. \tag{9f}$$

$$\boldsymbol{r}_p(t_f) - \boldsymbol{r}_{p+1}(t_0) = \boldsymbol{0}, \qquad p = 1, 2, 3. \tag{9g}$$

$$P\{T_p^{min} \leq T_p(t) + \boldsymbol{\xi} \leq T_p^{max}\} \geq 1 - \alpha \qquad p = 1, \ldots, 4. \tag{9h}$$

where Eq. 9h is defined as the chance constraint.

3. Convexification of the problem

To solve the problems in Eqs. (6) and (9), three convexification algorithms are used in this paper and described in the following subsections. In the first subsection, the nonlinear equality constraints in Eq. (9c) or (6c) are transformed into three inequality constraints: a convex one, a concave one and a slack variable greater than zero. In subsection two, a successive linearization algorithm is applied to deal with the remaining non-convexities of the two problems, such as the differential equations and inequality non-convex constraints. Finally, the chance constraints in Eq. 9h are treated by using the Bernstein approximation to get the determined ones [26].

3.1. Convexification of the Nonlinear Equality Constraints

Non-linear equality constraints (NEC) represent one of the main challenges in processing non-convex problems. In the specific case of the problems in Section 2.1, such kinds of constraints appear in Eq. 9c and Eq. 6c. A convex-concave decomposition method transforms the nonlinear equality constraint into a set of convex ones. This decomposition has been demonstrated to be effective for turning an NEC into a set of convex and concave constraints that convex optimizers can easily handle. Specifically, we adopt the procedure shown in [27], where a demonstration of the convergence of the method is also provided. Following the procedure in [27], we can transform the original NECs in Eq. 9c and Eq. 6c as follows:

$$u_x^2 + u_y^2 + u_z^2 - 1 - \epsilon \leq 0, \tag{10a}$$

$$-(u_x^2 + u_y^2 + u_z^2 - 1) \leq 0, \tag{10b}$$

$$\epsilon \geq 0 \tag{10c}$$

where a slack variable $\epsilon \geq 0$ is introduced to relax the constraints.

Traditional convex techniques such as linearization-based algorithms have some inherent limitations which may hardly guarantee problem convergence, those issues include solution chattering [27] or crawling phenomenon [28]. The approach taken in this paper relaxes the nonlinear equality constraints by transforming them into three inequality constraints with a slack variable. Based on the argument of the exact penalty function theory, the transformed problem has the same solution with a sufficiently large but finite penalty coefficient as the original problem. The main advantage of the convex-concave decomposition algorithm is that it can effectively save computing time with the help of inequality constraints and can reduce the solution chattering.

3.2. Successive Linearization

The traditional successive linearization approach is a technique employed to tackle the remaining nonconvexities of the optimization problem. In this section,

a first-order Taylor series expansion is used to replace the original nonconvex parts with a set of equations linearized around a reference solution. Here, the reference solution is chosen from the solution solving the nonlinear problem via the direct method. In general, the solution obtained with the non-linear problem might be sub-optimal, as it is not sure that the direct method could find the global optimum: different initial guesses might lead to different solutions. On the other hand, by transforming the problem into a convex problem, we are always able to obtain the same optimal solution for any initial guess used for triggering the optimization. This leads to a faster solution to the optimization problem.

3.2.1. Equations of Motion

The core concept revolves around harnessing the advantageous simplicity inherent in linear approximations to enhance the tractability of optimization problems while preserving accuracy within localized domains. A pivotal consideration entails the judicious selection of suitable linearization points, coupled with the imperative of ensuring that the linear approximations faithfully mirror the behaviour of the underlying non-linear constraints in these defined regions. The equations of the nonlinear dynamics in Eq. 9b and Eq. 6b are still nonconvex. To linearize such a system, we adopt the successive linearization approach so that we can reduce the equations into a convex set as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{t}) \approx \boldsymbol{A}(\boldsymbol{t})\Delta\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{t})\Delta\boldsymbol{u} + \boldsymbol{C}. \tag{11}$$

where $\boldsymbol{A}(\boldsymbol{t})$ and $\boldsymbol{B}(\boldsymbol{t})$ represent the derivatives of $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t)$ got from the reference solution at the point $(\boldsymbol{x}^{(n)}, \boldsymbol{u}^{(n)}, t^{(n)})$ with respect to the components of $\boldsymbol{x}$ and $\boldsymbol{u}$. $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}^{(n)}$, $\Delta\boldsymbol{u} = \boldsymbol{u} - \boldsymbol{u}^{(n)}$, and $\boldsymbol{C} = \boldsymbol{f}(\boldsymbol{x}^{(n)}, \boldsymbol{u}^{(n)}, \boldsymbol{t}^{(n)})$, respectively.

In addition, the use of successive linearization requires that the constraints be affine in the solution variable. Essentially, the trust region plays a key role during the convergence of the convex algorithm which could also induce the undesirable crawling phenomenon. The form of the trust region can be expressed

as follows:

$$\begin{aligned} \| \boldsymbol{u} - \boldsymbol{u}^{(n)} \| \leq \delta_{\boldsymbol{u}} \\ \| \boldsymbol{x} - \boldsymbol{x}^{(n)} \| \leq \delta_{\boldsymbol{x}} \end{aligned} \tag{12}$$

where the inequality sign applies componentwise, $\delta_{\boldsymbol{u}}$ and $\delta_{\boldsymbol{u}}$ are all constant vectors with appropriate dimensions. Note that the trust-region constraints are second-order cone constraints denoted with the symbol "$||\cdot||$".

3.2.2. Convexification of Path Constraints

The convexification process should also be extended to encompass both path constraints outlined in Eq. 6d and Eq. 9d, in addition to the transformed constraint presented in Eq. 10b. Furthermore, it should be noted that the terminal condition specified in Eq. 4 should undergo the necessary convexification in future iterations. The convexification of such equations can be done via successive linearization. By defining the entire set of inequality constraints as:

$$-H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{t}) \leq 0. \tag{13}$$

We can convert it into a set of linear constraints as follows:

$$-(H(\boldsymbol{x}^{(n)}, \boldsymbol{u}^{(n)}, \boldsymbol{t}^{(n)}) + H_{\boldsymbol{x}}(\tau)(\boldsymbol{x} - \boldsymbol{x}^{(n)}) + H_{\boldsymbol{u}}(\tau)(\boldsymbol{u} - \boldsymbol{u}^{(n)})) \leq 0. \tag{14}$$

where $H(\boldsymbol{x}^{(n)}, \boldsymbol{u}^{(n)}, \boldsymbol{t}^{(n)})$ is the set of constraints calculated at the reference solution at point $(\boldsymbol{x}^{(n)}, \boldsymbol{u}^{(n)}, t^{(n)})$. $H_{\boldsymbol{x}}(\tau)$ and $H_{\boldsymbol{u}}(\tau)$ are the derivatives of $H(\boldsymbol{x}^{(n)}, \boldsymbol{u}^{(n)}, \boldsymbol{t}^{(n)})$ with respect to $\boldsymbol{x}^{(n)}$ and $\boldsymbol{u}^{(n)}$, respectively.

3.3. Convexification of the Chance Constraints

The set of equations in Eq. 9h is a set of stochastic joint chance constraints. To convexify such kinds of constraints, these need to be decomposed. This can be achieved, for example by considering Boole's inequality, detailed in [26] but specified for the intersection of the independent conditions $g_{j,min} \leq g_i \leq g_{j,max}, j = 1, 2, ..., n_g$, that constitute the joint constraints:

$$P\{\cap_{j=1}^{n_g}(g_j(\xi) \geq g_{j,min}) \cap (g_j(\xi) \leq g_{j,max})\} \geq 1 - \alpha \tag{15}$$

By taking into account the complement to the Eq. 15, it is possible to calculate the probability of failure, i.e. the probability of non-respecting one of the conditions, as follows:

$$P\{\cup_{j=1}^{n_g}(g_j(\xi) < g_{j,min}) \cup (g_j(\xi) > g_{j,max})\} < \alpha \tag{16}$$

Therefore, by applying Boole's inequality theorem [26], it is possible to obtain the following sufficient condition:

$$\sum_{j=1}^{n_g}(P\{g_j(\xi) < g_{j,min}\} + P\{g_j(\xi) > g_{j,max}\}) < \alpha \tag{17}$$

Furthermore, Eq. 17 can be split into independent constraints:

$$P\{g_j(\xi) < g_{j,min}\} \leq \alpha_{j,1}, \quad P\{g_j(\xi) > g_{j,max}\} \leq \alpha_{j,2} \tag{18}$$

with

$$\sum_{j=1}^{n_g}(\alpha_{j,1} + \alpha_{j,2}) \leq \alpha \tag{19}$$

where $\alpha_{j,1}, \alpha_{j,2}$ are called allocated risk parameters. The full set of independent chance constraints obtained from Eq. 18 and Eq. 19 can be represented by:

$$P(G(\boldsymbol{\xi})) > 0. \tag{20}$$

At this point, the Bernstein approximation can be applied to the conditions in Eq. 20, so that it can be substituted by [29]:

$$P(G(\boldsymbol{\xi})) = \mathbb{E}[1_{G(\boldsymbol{\xi})}] \tag{21}$$

where $\mathbb{E}$ is the expectation of verifying the event $G(\xi)$ and the term $1_{(\cdot)}$ is the indicator function, which reads as:

$$1_{(\cdot)} = \begin{cases} 1 & G(\boldsymbol{\xi}) > 0 \\ 0 & otherwise \end{cases} \tag{22}$$

Therefore, the original chance constraint can be transformed as:

$$P(G(\boldsymbol{\xi}) > 0) = \mathbb{E}_{\boldsymbol{\xi}}[1_{(0,+\infty)}(G(\boldsymbol{\xi}))] = \mathbb{E}_{\boldsymbol{\xi}}[1_{(0,+\infty)}(\zeta G(\boldsymbol{\xi}))] \tag{23}$$

where ζ is a positive weight coefficient that does not alter the equality. To transform the condition in Eq. 23 into a convex one, we choose an exponential function to upper-bound the indicator function:

$$P(G(\boldsymbol{\xi}) > 0) = \mathbb{E}_{\boldsymbol{\xi}}[1_{(0,+\infty)}(\zeta G(\boldsymbol{\xi}))] \leq \mathbb{E}_{\boldsymbol{\xi}}[1/exp(1 - \zeta G(\boldsymbol{\xi}))], \quad \zeta > 0. \tag{24}$$

Indeed, the exponential function $1/exp(1 - \zeta G(\boldsymbol{\xi}))$ is always greater than the indicator function $1_{(0,+\infty)}(\zeta G(\boldsymbol{\xi}))$ for any value of $G(\boldsymbol{\xi})$ and has the following properties:

- $1/exp(1 - \zeta G(\boldsymbol{\xi})) > 1 = 1_{(0,+\infty)}(\zeta G(\boldsymbol{\xi}))$ for $G(\boldsymbol{\xi}) > 0$
- $1/exp(1 - \zeta G(\boldsymbol{\xi})) = 1 = 1_{(-\infty,0)}(\zeta G(\boldsymbol{\xi}))$ for $G(\boldsymbol{\xi}) = 0$
- $0 = 1_{(-\infty,0)}(\zeta G(\boldsymbol{\xi})) < exp(\zeta G(\boldsymbol{\xi})) < 1$ for $G(\boldsymbol{\xi}) < 0$

for any values of $\zeta > 0$. Such a substitution has been proven to be convex in [30].

The optimal control problem formulated in the convex optimal problem

(Problem 3), is rewritten as:

$$min \quad J = -m(t_f) + p\epsilon \tag{25a}$$

$$s.t. \quad \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{t}) \approx \boldsymbol{A}(\boldsymbol{t})\Delta\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{t})\Delta\boldsymbol{u} + \boldsymbol{C} \tag{25b}$$

$$\| \boldsymbol{u} - \boldsymbol{u}^{(n)} \| \leq \delta_{\boldsymbol{u}} \tag{25c}$$

$$\| \boldsymbol{x} - \boldsymbol{x}^{(n)} \| \leq \delta_{\boldsymbol{x}} \tag{25d}$$

$$- (H(\boldsymbol{x}^{(n)}, \boldsymbol{u}^{(n)}, \boldsymbol{t}^{(n)}) + H_{\boldsymbol{x}}(\tau)(\boldsymbol{x} - \boldsymbol{x}^{(n)}) + H_{\boldsymbol{u}}(\tau)(\boldsymbol{u} - \boldsymbol{u}^{(n)})) \leq 0 \tag{25e}$$

$$u_x^2 + u_y^2 + u_z^2 - 1 - \epsilon \leq 0, \tag{25f}$$

$$\epsilon \geq 0 \tag{25g}$$

$$m_p(t_f) - m_p^{drop} - m_{p+1}(t_0) = 0, \qquad p = 1, 2, 3 \tag{25h}$$

$$\boldsymbol{v}_p(t_f) - \boldsymbol{v}_{p+1}(t_0) = \boldsymbol{0}, \qquad p = 1, 2, 3 \tag{25i}$$

$$\boldsymbol{r}_p(t_f) - \boldsymbol{r}_{p+1}(t_0) = \boldsymbol{0}, \qquad p = 1, 2, 3 \tag{25j}$$

$$\boldsymbol{o}(t_f) - \boldsymbol{o}_f = \boldsymbol{0} \tag{25k}$$

$$P(G(\boldsymbol{\xi}) > 0) = \mathbb{E}_{\boldsymbol{\xi}}[1_{(0,+\infty)}(\zeta G(\boldsymbol{\xi}))] \leq \mathbb{E}_{\boldsymbol{\xi}}[1/exp(1 - \zeta G(\boldsymbol{\xi}))], \quad \zeta > 0 \tag{25l}$$

4. Discretization Method

Direct methods are required to transform the continuous-time problem into an NLP problem. Note that the discretization is a core step in the formulation of the second order cone programming problem, as it determines how accurately the problem captures the continuous-time dynamics. Proper discretization is mandatory to ensure that the converged solution of the linearized problem adheres to the original nonlinear dynamics. For example, a typical discretization method is the global pseudospectral method (p-methods), whose performance, in terms of convergence to a feasible solution is strictly dependent on the number and location of the collocation points [31]. However, those methods require very large-degree global polynomials to approximate the original problem better. Moreover, p-methods may also be computationally intractable or inefficient,

as the convergence rate may be extremely slow [32]. On the other hand, direct collocation methods (h-methods) can be used, where fixed low-degree polynomials are used to approximate the state. H-methods may require a higher number of mesh intervals but the convergence rate is faster, if compared to p-methods [33]. Another class of discretization methods is obtained by combining adaptively the characteristics of both the p-methods and h-methods: these kinds of methods are called hp-adaptive pseudospectral methods [34]. The hp-methods change the number of segments and polynomial degrees to reach a more accurate approximation of the state and to increase the utility of the pseudospectral method. In the present application, we adopt an hp-adaptive pseudospectral method to discretize the problems in Eq. 6 and Eq. 9. Reference [35] presented a novel variable-order adaptive pseudospectral method for the solution of optimal control problems. The proposed method dynamically adjusts the mesh spacing and polynomial degree within each interval to meet a predefined error tolerance. It achieves convergence by refining the mesh in regions characterized by high-curvature dynamics, while it enhances the approximation accuracy by increasing the polynomial degree in regions with low-curvature dynamics. Furthermore, this article acknowledges that due to limitations in space, the comprehensive details of the method are not reported herein. Such a method allows for obtaining a final discretized approximation of the control action as follows:

$$\boldsymbol{u}^k(\phi) = \sum_{i=1}^{N_k} \boldsymbol{u}_i^k L_i^k(\phi), \tag{26}$$

where N_k represents the degree of Legendre-Gauss-Radu polynomial, $k \in [1, ..., K]$ is the interval index, and $\phi \in [-1, 1]$ is the transformed form of $t \in [t_{k-1}, t_k]$ in each segment. $L_i(\phi)$, $(i = 1, ..., N_k)$ represents the basis of Lagrange polynomials:

$$L_i^k(\phi) = \prod_{j=1, j\neq i}^{N_k} \frac{\phi - \phi_j^k}{\phi_i - \phi_j^k}. \tag{27}$$

and the coefficients $\boldsymbol{u}_i^k$ are the unknown coefficients to be found through the optimization process. Defining a set of D points $(\bar{\phi}_1^k, ..., \bar{\phi}_D^k) \in [-1, 1]$, the accuracy of dynamics and path equations in a segment is determined by the

number of D sample points. Specifically, the accuracy tolerances $\boldsymbol{a}_d^{(k)}$ and $\boldsymbol{b}_d^{(k)}$ can be calculated at the points $(\bar{\phi}_1^k, ..., \bar{\phi}_D^k)$ as following:

$$\begin{aligned} |\dot{\boldsymbol{x}}^{(k)}(\bar{\phi}_d^k) - \tfrac{t_k - t_{k-1}}{2} f(\boldsymbol{x}_d^{(k)}, \boldsymbol{u}_d^{(k)}, \phi_d^{(k)}; t_{k-1}; t_k)| = \boldsymbol{a}_d^{(k)} \\ G(\boldsymbol{x}_d^{(k)}, \boldsymbol{u}_d^{(k)}, \phi_d^{(k)}; t_{k-1}, t_k) = \boldsymbol{b}_d^{(k)} \end{aligned} \tag{28}$$

where $d = 1, ..., D$. The optimal control problem formulated in the discrete convex optimal problem (Problem 4), is rewritten as:

$$\begin{aligned} min \quad & J = J_\Phi + J_L \approx \Phi(X_1^{(1)}, t_0, X_{N_{K+1}}^K, t_K) + \\ & \sum_{k=1}^{K} \sum_{i=1}^{N_k} \frac{t_k - t_{k-1}}{2} \omega_j^{(k)} L(X_i^{(k)}, U_i^{(k)}, \iota_i^{(k)}; t_{k-1}, t_k), && \text{(29a)} \\ s.t. \quad & \sum_{j=1}^{N_k+1} X_i^{(k)} D_{ij}^{(k)} - \frac{t_k - t_{k-1}}{2} f(X_i^{(k)}, U_i^{(k)}, \iota_i^{(k)}; t_{k-1}, t_k) = 0 && \text{(29b)} \\ & H^{(k)}(X_i^{(k)}, U_i^{(k)}, \iota_i; t_{k-1}, t_k) \leq 0, \quad (i = 1, \ldots, N_k) && \text{(29c)} \\ & \| \boldsymbol{U}_i^{(k)} - \boldsymbol{u}^{(n)} \| \leq \delta_{\boldsymbol{u}} && \text{(29d)} \\ & \| \boldsymbol{U}_i^{(k)} - \boldsymbol{x}^{(n)} \| \leq \delta_{\boldsymbol{x}} && \text{(29e)} \\ & \frac{1}{M} \sum_{m \in I_N} \Phi(\tau, g_i(u_j, \xi^m)) \leq 1 - \epsilon_i && \text{(29f)} \end{aligned}$$

where $X_1^{(1)}$ and $X_{N_{K+1}}^K$ represent the approximation of initial $(\boldsymbol{x}(t_0))$ and final $(\boldsymbol{x}(t_f))$ state variables. Where the problem has been divided into K segments,the time interval segment $k \in [1, ..., K]$ is $[t_{k-1}, t_k]$, N_k is the number of collocation points in segment k, and $t_K \equiv t_f$. L is the running cost and $\omega_i^{(k)}$ are the weight parameters. $\boldsymbol{U}_i^{(k)}$ and $\boldsymbol{X}_i^{(k)}$ are the approximations of the control and states at the N_k Legendre-Gauss-Radau (LGR) points in segment k. $D_{ij}^{(k)}$ is a basis of Lagrange polynomials, $(\iota_1^{(k)}, ..., \iota_i^{(k)})$ are the LGR collocation points in segment k.

5. Numerical Results

In this section, we elucidate the outcomes derived from a simulation campaign designed to evaluate the feasibility and efficacy of the proposed optimization algorithm based on convex-concave decomposition. The study specifically

addresses the ascent of a multiple-stage launch vehicle. As explicated in Section II, the simulation scenario delves into the dynamics of a two-stage launcher, encompassing three fixed-time phases and a concluding free-time phase. The characteristics and performance of each of the stages and boosters are shown in Table 1.

Table 1: Characteristics of the launcher

	Solid Motors	Stage 1	Stage 2
Total Mass (kg)	19290	104380	19300
Propellant Mass(kg)	17010	95550	16820
Engine Thrust (N)	628500	1083100	110094
Isp (s)	284	301.7	462.4
Number of Engines	9	1	1
Burn Time (s)	75.2	261	693

Table 2: Aerodynamic Parameters in Launch Vehicle

Constant	Value
Payload Mass (kg)	4146
$S(\mathrm{m}^2)$	4π
C_d	0.5
$\rho_0(\mathrm{kg/m}^3)$	1.225
$H(\mathrm{m})$	7200
t_1 (s)	75.2
t_2 (s)	150.4
t_3 (s)	261
R_e (m)	6378145
Ω (rad/s)	$7.29211585{*}10^{-5}$

Table 2 gives the aerodynamic parameters in the launch vehicle. We assume that the launch site is located at 28.5 *deg* longitude and corresponds to the geocentric latitude of Cape Canaveral, Florida, and it is arbitrarily assumed that

the inertially fixed axes are such that the initial inertial longitude is zero. 0 deg latitude, which translates into an initial position $\boldsymbol{r}_0 = [5605.2, 0, 3043.4]$ km and velocity $\boldsymbol{v}_0 = [0, 408.8, 0]$ m/s relative to the ECI frame, and the initial mass is $301454kg$, as derived from the characteristics of the launcher in Table 1. The simulations in this paper are considering the problem of insertion of a satellite to a parking orbit: this orbit is indeed inserted as a terminal constraint into the optimization problem. The final orbital elements that need to be achieved are $O_f = [a_f, e_f, i_f, \Omega_f, \omega_f]^T = [24361.14km, 0.7308, 10°, 269.8°, 130.5°]^T$, which we refer as ref [36], These orbital elements can be transformed into ECI coordinates via the transformation, T_{o2c} as given in [37].

The simulations were conducted within the Matlab environment, utilizing GPOPS-II in conjunction with SNOPT to address the optimization problems under investigation[38].

5.1. Comparative Analysis

To assess the performance of the proposed convex-concave decomposition with successive linearization (CCD+SL) method, a comparative test campaign was conducted. Two established methods, namely lossless convexification (LC) [39] and the successive linearization method (SL), were employed as benchmarks to gauge the computational demands of the newly proposed approach. The comparison was conducted while maintaining consistent mesh accuracy tolerance in the *hp*-adaptive pseudospectral discretization method and adhering to the same optimization problem definition strategy. An initial estimate for the three methods was derived from a reference trajectory, initially computed using the NLP method, which may represent a locally optimal solution. Despite the free final time, the optimization problem is framed around minimizing the overall propellant consumption (comprising both fuel and oxidizer) during the ascent phase. This objective corresponds to maximizing the launch vehicle's efficiency by accomplishing mission objectives with minimal fuel consumption. Our goal is to optimize the payload mass delivered to the designated orbit.

All simulations and optimization algorithms were executed on a laptop equipped

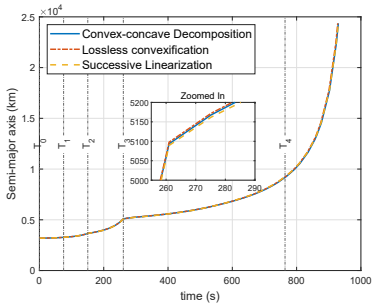
with an Intel(R) Core(TM) i7-12700H processor clocked at 2.30 GHz and 32.0 GB of RAM. The computational results were averaged over five repetitions to provide a representative characterization of the CPU times.

Table 3 displays the computational time required for running these algorithms, along with the corresponding optimal values of the cost function, as defined in Eq. 2. It is noteworthy to highlight that the CCD+SL method exhibits a reduced optimization time compared to the LC and SL methods, indicating its efficiency in solving the problem at hand. The proposed method confers a substantial advantage in terms of attaining the ultimate objective. The findings of this study provide compelling evidence that the application of the convex-concave decomposition technique yields notable enhancements in convergence, particularly in relation to the speed of solution generation. Through the decomposition of the original optimization problem into convex and concave subproblems, we effectively leverage the favorable characteristics of each subproblem and capitalize on specialized optimization algorithms and techniques tailored to convex and concave optimization, respectively.

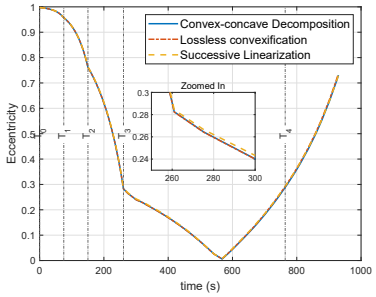
Table 3: Transitions selected for thermometry

Performance	CCD+SL method	LC method	SL method
Computational Time (s)	1.47	1.49	1.59
$Mass_f(kg)$	7427.03	7396.04	7394.43

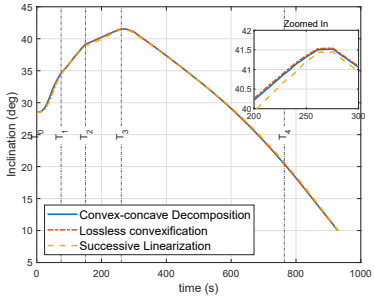
This advantage primarily arises from the iterative structure of the CCD method and the characteristics of the problems it is designed to address. In many trajectory optimization problems, the variables can be separated into two groups: one set that can be treated as convex and another set that can be treated as concave. This separation enables the CCD method to address each set of variables efficiently using specialized solvers tailored to convex and concave optimization. Using the successive linearization method to convexify the concave part within CCD can lead to faster convergence, improved accuracy within local regions, and more efficient optimization steps.



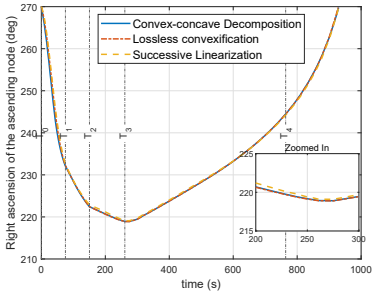
(a) Semi-major axis



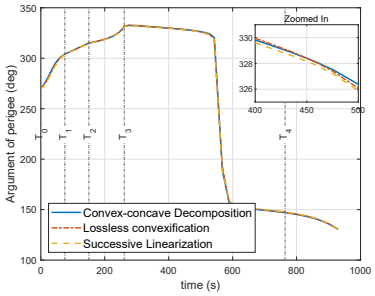
(b) Eccentrictiy



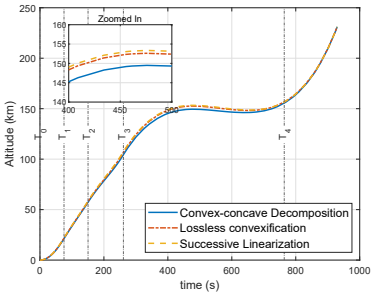
(c) Inclination



(d) Right ascension of the ascending node



(e) Argument of perigee



(f) Altitude

Figure 2: The comparison of orbit elements with three convex methods

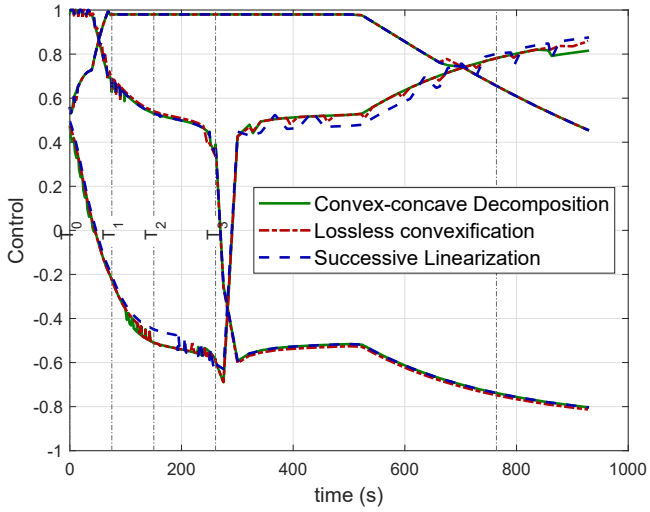


Figure 3: The comparative results of control variables using three convex methods

Fig. 2 shows the evolution of all the orbital elements during the simulated launch. Specifically, the semi-major axis, the eccentricity, the inclination, the right ascension of the ascending node, and the argument of perigee of the whole ascent process are represented from Fig. 2(a) to Fig. 2(e), respectively. The altitude of the trajectory with respect to the Earth's surface is illustrated in Fig. 2(f), where the four phases are also represented. Considering that the three convex optimization methods utilize identical reference trajectories and demonstrate similar levels of accuracy, the discrepancies observed in the resultant curves are negligible. The main distinctions primarily revolve around the altitude depicted in Fig. 2(f) and the control characteristics illustrated in Fig. 3. When comparing the control volume curves depicted in Fig. 3 with those generated by the other two convexification methods, it becomes evident that the SL method exhibits a higher degree of oscillation. Conversely, the control volume curves produced by the CCD+SL and LC methods demonstrate comparatively smoother behaviour.

A Local-Vertical-Local-Horizontal (LVLH) reference frame is adopted to represent in an intuitive way the thrust direction. Such a reference frame has the

centre of mass of the launcher as its origin and its axes are defined as follows:

- $\hat{\boldsymbol{x}}_{lvlh}$ is aligned along the radial direction and it can be calculated as follows:

$$\hat{\boldsymbol{x}}_{lvlh} = \frac{\boldsymbol{r}}{|\boldsymbol{r}|} \tag{2.8c}$$

- $\hat{\boldsymbol{z}}_{lvlh}$ is aligned along the instantaneous orbit normal, which is defined as follows:

$$\hat{\boldsymbol{z}}_{lvlh} = \frac{\boldsymbol{r}\times\boldsymbol{V}}{|\boldsymbol{r}\times\boldsymbol{V}|} \tag{30}$$

- $\hat{\boldsymbol{y}}_{lvlh}$ is built to form a right-handed set as follows:

$$\hat{\boldsymbol{y}}_{lvlh} = \boldsymbol{x}_{lvlh} \times \boldsymbol{z}_{lvlh} \tag{31}$$

In such a way, it is possible to define the rotation matrix:

$$\boldsymbol{R} = [\hat{\boldsymbol{x}}_{lvlh}, \hat{\boldsymbol{y}}_{lvlh}, \hat{\boldsymbol{z}}_{lvlh}], \tag{32}$$

that can be used to transform the components of the thrust vector $\hat{\boldsymbol{u}}_{eci}$ from the ECI reference frame into the LVLH reference frame as follows:

$$\boldsymbol{u}_{lvlh} = \boldsymbol{R}' \cdot \boldsymbol{u}_{eci} \tag{33}$$

where the $\boldsymbol{R}'$ is the transpose of the rotation matrix $\boldsymbol{R}$. Based on this transformation, the thrust direction can be represented by two angles that are defined concerning the LVLH reference. These are:

- the in-plane thrust angle, defined as follows:

$$\tan(\epsilon_u) = \frac{u^x_{lvlh}}{u^y_{lvlh}} \tag{34}$$

- the out-of-plane thrust angle, defined as follows:

$$\tan(\theta_u) = \frac{u^z_{lvlh}}{\sqrt{(u^x_{lvlh})^2 + (u^y_{lvlh})^2}} \tag{35}$$

where u^x_{lvlh}, u^y_{lvlh}, and u^z_{lvlh} are the components of the thrust direction unit vector $\hat{u}$ along the three axes of the LVLH reference frame, respectively. The

behavior of these two thrust angles, ϵ_u and θ_u are illustrated in Fig.4(a) and in Fig. 4(b). These show that all three methods produce a similar optimal thrust direction, with the in-plane thrust angle that starts from 70 *deg* and rapidly decreases down to 20 *deg* during the first phase of the first stage (time T_1 in Fig.4(a)) and then continues decreasing with a slower rate during the next phases, down to -20 *deg* at the end of the insertion manoeuvre. Fig. 4(b) shows the behavior of the out-of-plane thrust angle. Despite such differences the inclination of the target orbit is reached in all the cases, as shown in Fig. 2(c).

As a result, the launcher follows an ascending trajectory by changing the velocity as shown in Fig. 5, where the three components of the velocity are represented for the LVLH reference frame. Fig. 5(a) shows the behaviour of the launcher velocity along the radial direction, which starts from 0 km/s at T_0 and rapidly reaches almost 0.5 km/s at the end of the first stage (at time T_3) for then reaching its value to 0.83 km/s during the second stage. On the other hand, the tangential component of the velocity, shown in Fig. 5(b), shows a completely different behaviour, starting with a non-zero value due to the Earth's rotation and then raising its value up to the target orbital velocity of 10.3 km/s. Finally, the out-of-plane component of the velocity does not sensibly change during the manoeuvre, as shown in Fig. 5(c) for all three methods under investigation.

The direction of the velocity can also be represented by two angles defined to the LVLH reference:

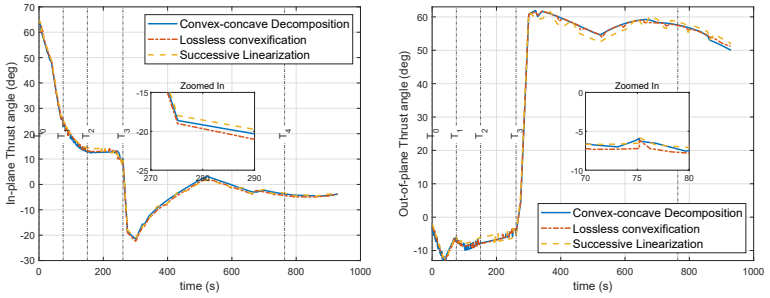
- the climbing angle, defined as follows:

$$\tan(\epsilon_v) = \frac{v^x_{lvlh}}{v^y_{lvlh}} \tag{36}$$

- the turning angle, defined as follows:

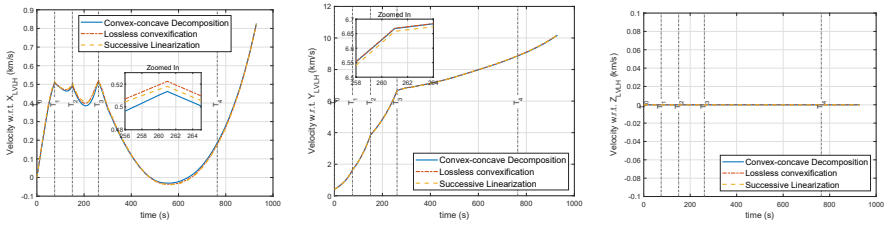
$$\tan(\theta_v) = \frac{v^z_{lvlh}}{\sqrt{(v^x_{lvlh})^2 + (v^y_{lvlh})^2}} \tag{37}$$

where v^x_{lvlh}, v^y_{lvlh},and v^z_{lvlh} are the components of the velocity $\boldsymbol{v}_{lvlh}$ along the three axes of the LVLH reference frame, respectively. The behaviors of ϵ_v and



(a) In plane thrust angle (b) Out plane thrust angle

Figure 4: The comparison of in-plane thrust angle and out-of-plane thrust angle results under LVLH frame



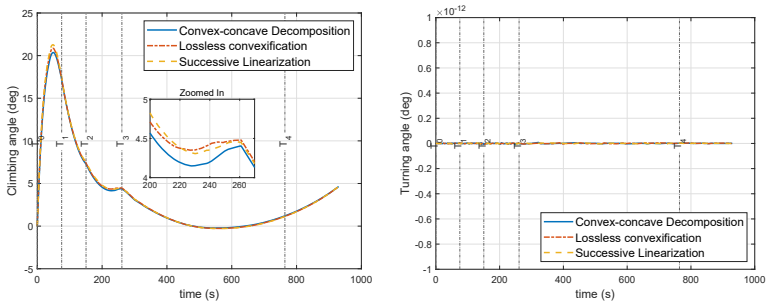
(a) Velocity along LVLH X axis (b) Velocity along LVLH Y axis (c) Velocity along LVLH Z axis

Figure 5: The velocity comparing results under LVLH frame

θ_v over the time of the maneuver, are plotted in Fig. 9. Fig. 10(a) shows that the climbing angle rises from 0 *deg* to nearly 22 *deg* in the first phase, producing a rapid increase of the launcher altitude in the first phase of the launch and, then gradually decreases to 0 *deg* from time T_1 to the final time T_4, with the velocity getting aligned along the tangential velocity. The limited changes in the out-of-plane thrust direction are reflected in Fig. 10(b), where the turning angle does not show any sensible change and remains constant at zero for the duration of the launch.

5.2. Verification of the Effectiveness of Chance Constraints

An additional test campaign was carried out to verify the performance of the proposed convex-concave decomposition method when also chance constraints



(a) Climbing angle (b) Turning angle

Figure 6: The comparison of Climbing angle and Turning results under LVLH frame

are considered for modelling the uncertainty on the thrust magnitude. In this case, we consider the problem formulation reported in Problem 2 in Section 2.3.

$$T \leq 0.8 * randn(m, 1) + T \tag{38}$$

This equation represents adding the uncertainty (calculated using the expression involving rand) to the thrust magnitude while ensuring that the result falls within a predefined upper bound.

The first key result to be reported relates to the computational time necessary to run the modified algorithm, including the chance constraint. With the same computational setup as before, we noticed that the inclusion of the algorithms handling the chance constraint significantly increased the computational time from $1.47s$ to $23.2s$, for the case without chance constraints and for the case with chance constraints, respectively.

The uncertainty on the thrust magnitude on the other side, plays a vital role in the behaviours represented in Fig. 7. Such a figure shows the trends of the violation rate, intended as the probability of violating the constraint in Eq. 9h during the different phases of the ascent. The reported values are calculated through the Eq. 24 and definitively provide evidence that if not handled opportunely, the uncertainty on the thrust magnitude brings the violation rate above the threshold α set to 0.1 Indeed the behaviour of the solution without

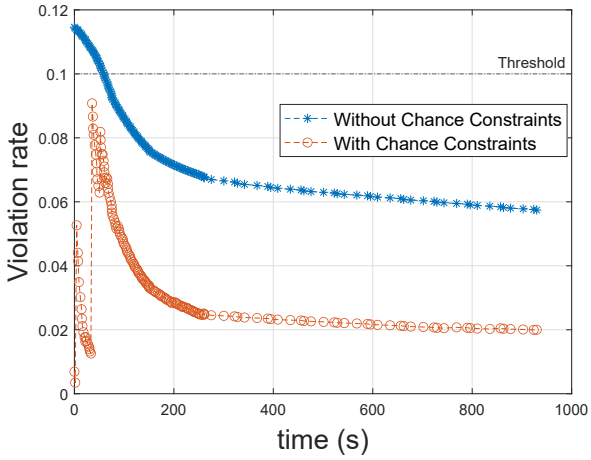


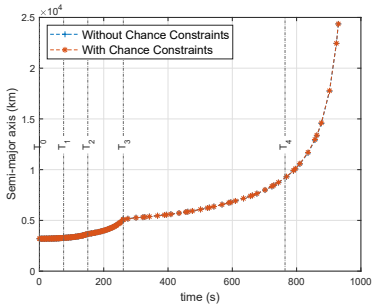
Figure 7: The comparing results with/without Chance constraints

chance constraint is fair above such a threshold during the first two phases of the launch. On the other hand, if correctly handled with the algorithm shown in Section 3.3, the violation rate of the condition in Eq. 24 remains below the predefined threshold.

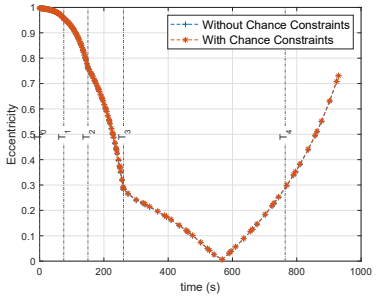
Fig. 8 describes the behaviour of all the orbital elements in the two cases with or without the chance constraints, respectively. For almost all the parameters, there is no evidence of any particular difference between the two cases, with trends of the orbital parameters matching each other. The only registered difference is noted in the behaviour of the right ascension of the ascending node in Fig. 8(d), where the solution with the handling of the chance constrain shows a less accentuated drop just in the middle of the first phase, compared to the case without chance constraint. In any case, both the solutions successfully achieve the target final orbit.

Fig. 10 illustrates the behaviour of the in-plane and out-of-plane angles. Fig. 11 illustrates the comparative outcomes of control variables when considering the presence or absence of chance constraints. The presence of chance constraints induces slight fluctuations, or jitter, in the control variables.

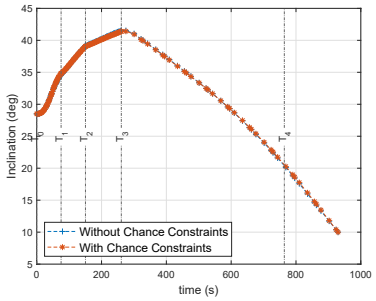
The incorporation of chance constraints in trajectory optimization intro-



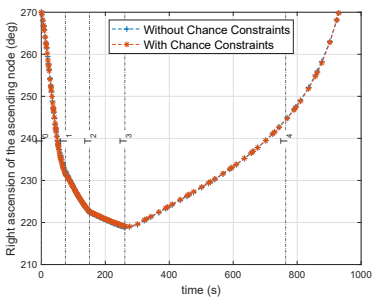
(a) Semi-major axis



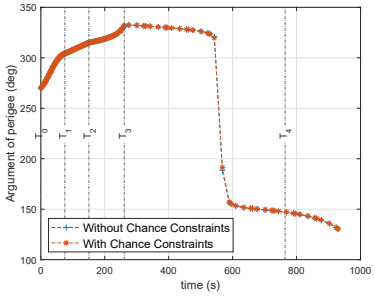
(b) Eccentrictiy



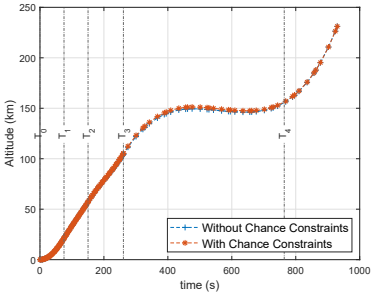
(c) Inclination



(d) Right ascension of the ascending node

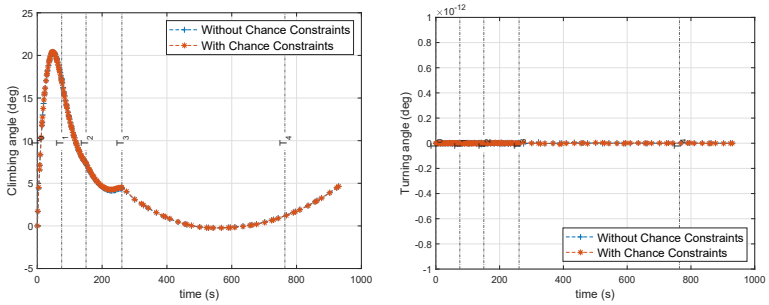


(e) Argument of perigee



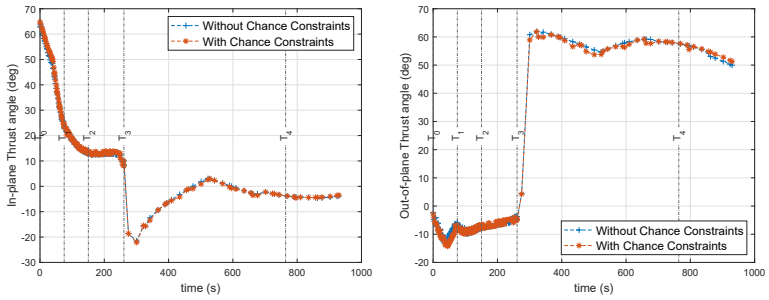
(f) Attitude

Figure 8: The state comparison results with/without Chance constraints under different uncertainty max degree



(a) Climbing angle (b) Turning angle

Figure 9: The comparison of Climbing angle and Turning results under LVLH frame



(a) In-plane angles (b) Out-of-plane angles

Figure 10: The comparison of In-plane angle and Out-of-plane angle results under LVLH frame

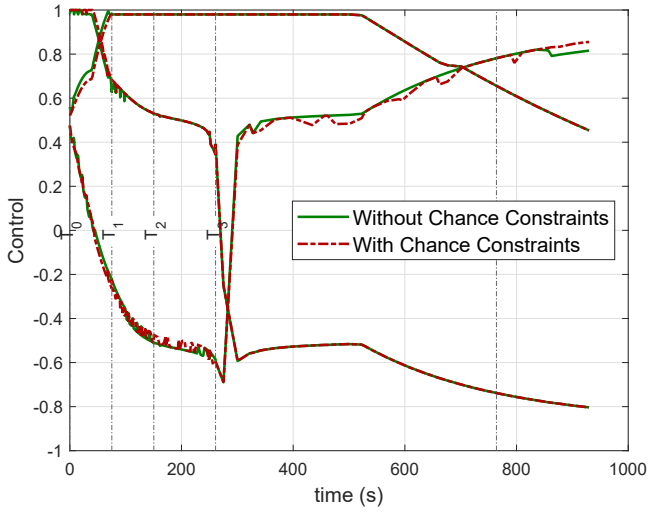


Figure 11: The comparative results of control variables with/without chance constraints

duces trade-offs in terms of solution complexity and computational demands. The inclusion of these constraints requires modeling and quantifying uncertainties, which adds complexity to the problem formulation. Additionally, the transformation of non-convex problems into convex ones through techniques like Bernstein approximation further contributes to the solution complexity. Computationally, the evaluation of probabilistic quantities and the solution of convex optimization problems may impose additional computational overhead. Therefore, careful consideration of available computational resources and problem size is crucial. Despite these trade-offs, the approach offers the advantage of accounting for uncertainties and ensuring system reliability. Ultimately, striking the right balance between the benefits of incorporating chance constraints and the associated computational costs is essential for successful implementation.

6. Conclusion

This paper presents a convex approach to solving the multiple-stage launch vehicle trajectory optimization problem. The existence of the nonlinear equality constraints in the problem has been effectively tackled via a convex-concave

decomposition process. As a result three new convex constraints, two inequality constraints plus another constraint that includes a slack variable, can be substituted for the original ones still guaranteeing the same solution of the original optimal control problem. The slack variable allows for having feasible solutions, e.g. solutions that satisfy the set of new convex constraints. To deal with the remaining non-convex parts of the optimization problem, successive linearization is used to convexify the remaining constraints, such as differential equality constraints, and non-convex inequality constraints. The inclusion of uncertainty in the model is carried on through the definition of chance constraints. These are handled via a specific convexification process based on the Bernstein approximation. In addition, the study shows that the *hp*-adaptive pseudospectral method can discretize the problem accurately, and generate a sparse discrete problem that can be solved efficiently. The effectiveness of the proposed algorithm is demonstrated through numerical simulations evaluating the performance sensitivity to the goal of the optimization problem.

The proposed convex-concave optimization approach for trajectory optimization of multistage launch vehicles presents opportunities for future research to enhance its effectiveness and applicability. Three key areas for exploration include integrating machine learning techniques to improve efficiency and decision-making, conducting research on real-time trajectory adjustments for adaptive optimization, and expanding the comparative evaluation of the proposed approach against alternative methods in diverse scenarios. Additionally, exploring cooperative game penetration guidance, task coupling-based layered cooperative guidance, and developing three-dimensional impact-angle-constrained fixed-time cooperative guidance algorithms will further enhance cooperative guidance capabilities. Addressing these research areas will lead to significant improvements in trajectory optimization strategies, ensuring safer and more successful space missions. Future endeavours in these areas will contribute to the development of advanced techniques for coordinating and guiding multiple vehicles in complex and safety-critical scenarios.

7. Appendix

$\boldsymbol{r} = [x(t), y(t), z(t)]^T$ is the position and $\boldsymbol{v} = [v_x(t), v_y(t), v_z(t)]^T$ is the velocity of the launcher with respect to the ECI reference frame. The control vector is denoted as $\hat{\boldsymbol{u}} = [u_x, u_y, u_z]^T$.

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t) \tag{.1}$$

$\boldsymbol{x} = [x_x, x_y, x_z, v_x, v_y, v_z, m]$,

$$A = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} O_{3*3} & I_{3*3} & O_{3*1} \\ V^r_{3*3} & V^v_{3*3} & V^m_{3*1} \\ O_{1*3} & O_{1*3} & O_{1*1} \end{bmatrix}, \; B = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} = \begin{bmatrix} O_{3*3} & B_{3*3} & O_{3*1} \end{bmatrix},$$

where O is the zero matrix.

$$V^r_{3*3} = \begin{bmatrix} -\frac{\mu}{(\|x^2+\boldsymbol{r}\|)^3} - \frac{3x^2}{\|x^2+\boldsymbol{r}\|^5} & \frac{3\mu xy}{\|y^2+\boldsymbol{r}\|^5} & \frac{3\mu xz}{\|z^2+\boldsymbol{r}\|^5} \\ \frac{3\mu xy}{\|x^2+\boldsymbol{r}\|^5} & -\frac{\mu}{(\|y^2+\boldsymbol{r}\|)^3} - \frac{3y^2}{\|y^2+\boldsymbol{r}\|^5} & \frac{3\mu xz}{\|z^2+\boldsymbol{r}\|^5} \\ \frac{3\mu xz}{\|x^2+\boldsymbol{r}\|^5} & \frac{3\mu yz}{\|y^2+\boldsymbol{r}\|^5} & -\frac{\mu}{(\|z^2+\boldsymbol{r}\|)^3} - \frac{3z^2}{\|z^2+\boldsymbol{r}\|^5} \end{bmatrix} \tag{.2}$$

$$V^v_{3*3} = \begin{bmatrix} \boldsymbol{v}_x(\boldsymbol{v}_x - \boldsymbol{\omega} \times \boldsymbol{r}) & 0 & 0 \\ 0 & \boldsymbol{v}_y(\boldsymbol{v}_y - \boldsymbol{\omega} \times \boldsymbol{r}) & 0 \\ 0 & 0 & \boldsymbol{v}_z(\boldsymbol{v}_z - \boldsymbol{\omega} \times \boldsymbol{r}) \end{bmatrix} \tag{.3}$$

$V^m_{3*1} = [\frac{-Tu_x - D}{m^2} \quad \frac{-Tu_y - D}{m^2} \quad \frac{-Tu_z - D}{m^2}]^T$

$$B_{3*3} == \begin{bmatrix} \frac{T}{m} & 0 & 0 \\ 0 & \frac{T}{m} & 0 \\ 0 & 0 & \frac{T}{m} \end{bmatrix} \tag{.4}$$

In this paper, Two nonlinear inequations are as follows:

$$-H^r(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = -(\sqrt{x^2 + y^2 + z^2} - R_e) \leq 0 \tag{.5a}$$

$$-H^u(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = -(u_x^2 + u_y^2 + u_z^2 - 1) \leq 0 \tag{.5b}$$

Employing the method of successive linearization, we have succeeded in rendering these intricate non-linear expressions as a singular entity denoted by

the symbol "H". The ensuing representation depicts the two aforementioned equations in their transformed state:

$$lH_x^r(\tau) = \frac{-\boldsymbol{x}_n}{\sqrt{\boldsymbol{x}_n^2 + \boldsymbol{y}_n^2 + \boldsymbol{z}_n^2}} \tag{.6a}$$

$$H_y^r(\tau) = \frac{-\boldsymbol{y}_n}{\sqrt{\boldsymbol{x}_n^2 + \boldsymbol{y}_n^2 + \boldsymbol{z}_n^2}} \tag{.6b}$$

$$H_z^r(\tau) = \frac{-\boldsymbol{z}_n}{\sqrt{\boldsymbol{x}_n^2 + \boldsymbol{y}_n^2 + \boldsymbol{z}_n^2}} \tag{.6c}$$

$$H_u^r(\tau) = 0 \tag{.6d}$$

$$H_r^u(\tau) = 0 \tag{.7a}$$

$$H_{u^1}^u(\tau) = 2u_n^1 \tag{.7b}$$

$$H_{u^2}^u(\tau) = 2u_n^2 \tag{.7c}$$

$$H_{u^3}^u(\tau) = 2u_n^3 \tag{.7d}$$

References

[1] Y. Li, C. Wei, Y. He, R. Hu, A convex approach to trajectory optimization for boost back of vertical take-off/vertical landing reusable launch vehicles, Journal of the Franklin Institute 358 (7) (2021) 3403–3423. doi:https://doi.org/10.1016/j.jfranklin.2021.02.017.

[2] X. LIU, F. ZHANG, Z. LI, Y. ZHAO, Approach and landing guidance design for reusable launch vehicle using multiple sliding surfaces technique, Chinese Journal of Aeronautics 30 (4) (2017) 1582–1591. doi:https://doi.org/10.1016/j.cja.2017.06.008.

[3] T. Liu, D. Han, Y. Lin, K. Liu, Distributed multi-uav trajectory optimization over directed networks, Journal of the Franklin Institute 358 (10) (2021) 5470–5487. doi:https://doi.org/10.1016/j.jfranklin.2021.04.044.

[4] W. Jia, S. Qin, An adaptive penalty-like continuous-time algorithm to constrained distributed convex optimization, Journal of the Franklin Institute 359 (8) (2022) 3692–3716. doi:https://doi.org/10.1016/j.jfranklin.2022.03.046.

[5] W. Roh, Y. Kim, Trajectory Optimization for a Multi-Stage Launch Vehicle Using Time Finite Element and Direct Collocation Methods, Engineering Optimization 34 (1) (2002) 15–32. doi:10.1080/03052150210912.

[6] M. Huo, L. Yang, N. Peng, C. Zhao, W. Feng, Z. Yu, N. Qi, Fast costate estimation for indirect trajectory optimization using bezier-curve-based shaping approach, Aerospace Science and Technology 126 (2022) 107582. doi:https://doi.org/10.1016/j.ast.2022.107582.

[7] D. Yazdani, R. Cheng, D. Yazdani, J. Branke, Y. Jin, X. Yao, A survey of evolutionary continuous dynamic optimization over two decades—part b, IEEE Transactions on Evolutionary Computation 25 (4) (2021) 630–650.

[8] T. Osa, Multimodal trajectory optimization for motion planning, The International Journal of Robotics Research 39 (8) (2020) 983–1001.

[9] M. V. Dileep, S. Kamath, V. G. Nair, Optimal trajectory generation of launch vehicle using PSO algorithm, in: 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015, pp. 56–60. doi:10.1109/ABLAZE.2015.7154970.

[10] D. J. Bayley, R. J. Hartfield, J. E. Burkhalter, R. M. Jenkins, Design Optimization of a Space Launch Vehicle Using a Genetic Algorithm, Journal of Spacecraft and Rockets 45 (4) (2008) 733–740. doi:10.2514/1.35318.

[11] D. Morante, M. Sanjurjo Rivo, M. Soler, A Survey on Low-Thrust Trajectory Optimization Approaches, Aerospace 8 (3) (2021) 88. doi:10.3390/aerospace8030088.

[12] H. Hou, W. W. Hager, A. V. Rao, Convergence of a Gauss Pseudospectral Method for Optimal Control (August) (2012) 1–9. doi:10.2514/6.2012-4452.

[13] A. C. Morelli, A. Morselli, C. Giordano, F. Topputo, Convex trajectory optimization using thrust regularization, Journal of Guidance, Control, and Dynamics 47 (2) (2024) 339–346.

[14] C. Hofmann, A. C. Morelli, F. Topputo, Performance assessment of convex low-thrust trajectory optimization methods, Journal of Spacecraft and Rockets 60 (1) (2023) 299–314.

[15] B. Açıkmeşe, J. M. Carson, L. Blackmore, Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem, IEEE Transactions on Control Systems Technology 21 (6) (2013) 2104–2113.

[16] X. Liu, P. Lu, B. Pan, Survey of convex optimization for aerospace applications, Astrodynamics 1 (1) (2017) 23–40. doi:10.1007/s42064-017-0003-8.

[17] Y. Li, S. Liang, J. Gao, Z. Chen, S. Qiao, Z. Yin, Trajectory optimization for the nonholonomic space rover in cluttered environments using safe convex corridors, Aerospace 10 (8) (2023) 705.

[18] Y. Li, R. Sun, W. Chen, Online trajectory optimization and guidance algorithm for space interceptors with nonlinear terminal constraints via convex programming, Aircraft Engineering and Aerospace Technology 95 (1) (2023) 53–61.

[19] B. Benedikter, A. Zavoli, G. Colasurdo, S. Pizzurro, E. Cavallini, Convex approach to three-dimensional launch vehicle ascent trajectory optimization, Journal of Guidance, Control, and Dynamics (2021) 1–16.

[20] Y. Li, B. Pang, C. Wei, N. Cui, Y. Liu, Online trajectory optimization for power system fault of launch vehicles via convex programming 98 (1) (2020) 105682. doi:https://doi.org/10.1016/j.ast.2020.105682.

[21] S. Zhai, J. Yang, Piecewise analytic optimized ascent trajectory design and robust adaptive finite-time tracking control for hypersonic boost-glide vehicle, Journal of the Franklin Institute 357 (9) (2020) 5485–5501. doi:https://doi.org/10.1016/j.jfranklin.2020.03.002.

[22] L. CHENG, P. SHI, S. GONG, Z. WANG, Real-time trajectory optimization for powered planetary landings based on analytical shooting equations, Chinese Journal of Aeronautics 35 (7) (2022) 91–99. doi:https://doi.org/10.1016/j.cja.2021.07.024.

[23] R. CHAI, A. TSOURDOS, A. SAVVARIS, S. CHAI, Y. XIA, High-fidelity trajectory optimization for aeroassisted vehicles using variable order pseudospectral method, Chinese Journal of Aeronautics 34 (1) (2021) 237–251. doi:https://doi.org/10.1016/j.cja.2020.07.032.

[24] D. A. Vallado, Fundamentals of astrodynamics and applications, Vol. 12, Springer Science & Business Media, 2001.

[25] X. Sun, B. Zhang, R. Chai, A. Tsourdos, S. Chai, Uav trajectory optimization using chance-constrained second-order cone programming, Aerospace Science and Technology 121 (2022) 107283. doi:https://doi.org/10.1016/j.ast.2021.107283.

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

[27] P. Lu, Convex–concave decomposition of nonlinear equality constraints in optimal control, Journal of Guidance, Control, and Dynamics 44 (1) (2021) 4–14. doi:10.2514/1.G005443.

[28] T. P. Reynolds, M. Mesbahi, The Crawling Phenomenon in Sequential Convex Programming, in: 2020 American Control Conference (ACC), 2020, pp. 3613–3618. doi:10.23919/ACC45564.2020.9147550.

[29] F. Usta, On new modification of bernstein operators: Theory and applications, Iranian Journal of Science and Technology, Transaction A: Science 44 (4) (2020) 1119–1124. doi:10.1007/s40995-020-00919-y.

[30] Z. Zhao, M. Kumar, A mcmc/bernstein approach to chance constrained programs, in: 2014 American Control Conference, 2014, pp. 4318–4323. doi:10.1109/ACC.2014.6859159.

[31] G. Elnagar, M. A. Kazemi, M. Razzaghi, The Pseudospectral Legendre Method for Discretizing Optimal Control Problems, IEEE Transactions on Automatic Control 40 (10) (1995) 1793–1796. doi:10.1109/9.467672.

[32] D. A. Benson, A gauss Pseudospectral Transcription for Optimal Control, MIT Libraries (2005) 111–123.

[33] S. Kameswaran, L. T. Biegler, Convergence rates for direct transcription of optimal control problems using collocation at Radau points, Computational Optimization and Applications 41 (1) (2008) 81–126. doi:10.1007/s10589-007-9098-9.

[34] C. L. Darby, W. W. Hager, A. V. Rao, An hp-adaptive pseudospectral method for solving optimal control problems, Optimal Control Applications and Methods 32 (4) (2011) 476–502. doi:10.1002/oca.957.

[35] C. L. Darby, W. W. Hager, A. V. Rao, Direct trajectory optimization using a variable low-order adaptive pseudospectral method, Journal of Spacecraft and Rockets 48 (3) (2011) 433–445. doi:10.2514/1.52136.

[36] A. V. Rao, D. A. Benson, C. Darby, M. A. Patterson, C. Francolin, I. Sanders, G. T. Huntington, Algorithm 902: Gpops, a matlab software

for solving multiple-phase optimal control problems using the gauss pseudospectral method, ACM Transactions on Mathematical Software (TOMS) 37 (2) (2010) 1–39.

[37] R. R. Bate, D. D. Mueller, J. E. White, W. W. Saylor, Fundamentals of astrodynamics, Courier Dover Publications, 2020.

[38] M. A. Patterson, 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 Transactions on Mathematical Software 41 (1) (2014) 1–37. doi:10.1145/2558904.

[39] B. Açıkmeşe, L. Blackmore, Lossless convexification of a class of optimal control problems with non-convex control constraints, Automatica 47 (2) (2011) 341–347.



https://dspace.org/

Convex–concave optimization for a launch vehicle ascent trajectory with chance constraints

Sun, Xin; Chai, Senchun; Chai, Runqi; Zhang, Baihai; Felicetti, Leonard ...

2024-05-01
Elsevier
https://doi.org/10.1016/j.jfranklin.2024.106849
Sun X, Chai S, Chai R, et al., (2024) Convex-concave optimization for a launch vehicle ascent trajectory with chance constraints. Journal of the Franklin Institute, Volume 361, Issue 8, May 2024, Article number 106849

Downloaded from a DSpace repository. https://dspace.org/