72nd International Astronautical Congress (IAC) – 25-29 October 2021.

IAC-21.C1.3.6.x65881

A Homotopic Approach for Robust Low-Thrust Trajectory Design through Convex Optimization

Andrea Carlo Morelli[1], **Christian Hofmann**[2], **Francesco Topputo**[3]

The space industry has recently witnessed a significant decrease of the overall costs of space missions, thanks to the miniaturization of satellites and their components. CubeSats have granted institutions and small companies access to space. However, space operations are still entirely performed from ground, limiting the potentiality of such spacecraft. Enhancing the autonomy of satellites, for example enabling on-board guidance, represents thus an interesting research challenge. The low control authority and little on-board resources of CubeSats require a new trajectory design paradigm. Optimization methods can be compared in terms of computational effort, optimality (the quality of the solution found), and feasibility (the capability of converging to a feasible solution). State-of-the-art approaches lack of computational efficiency, as powerful computers can be used to design the fuel-optimal trajectory of a spacecraft offline. Convex optimization represents instead a sustainable approach when real-time applications are considered due to the limited resources required to solve convex programs. This technique has been recently applied to different space-related problems, including powered descent and landing, entry and low-thrust trajectory optimization. This paper presents a sequential convex programming algorithm based on a Legendre–Gauss–Lobatto discretization scheme with nonlinear control interpolation to solve the minimum-fuel space trajectory optimization problem. Moreover, an adaptive second-order trust region radius change mechanism is developed to reduce the overall computational time. Finally, the sequential convex programming is combined with the homotopic approach to increase robustness of the method. The effectiveness of the approach is shown by means of numerical simulations with poor initial guesses.

1 Introduction

In recent years, CubeSats have allowed a significant reduction of space missions development costs [1], resulting in an increasing number of launches. On the other hand, operating a miniaturized satellite is as expensive as operating a conventional spacecraft [2]. Increasing the level of autonomy and shifting flight-related tasks such as the guidance design on-board is therefore a desirable goal for future missions. These developments require new trajectory design approaches.

The solution of the minimum-fuel optimal control problem is obtained by means of algorithms that can be compared in terms of computational effort (how many computational resources are needed), feasibility (capability of converging even when a poor initial guess is provided) and optimality (minimization of some cost function) [3]. The current paradigm of how space missions are operated allows engineers to repeatedly recompute trajectories during an interplanetary transfer using high-power computers: the focus is thus on optimality. On the other hand, scenarios like autonomous guidance require computationally fast and robust techniques that are able to find (near) optimal solutions in little time and with the available on-board resources.

The most common state-of-the-art methods used to solve the low-thrust space trajectory optimization problem can be divided in direct and indirect approaches. Classical direct methods that solve the full nonlinear optimization problem are in general not suitable for on-board applications due to their computational complexity and poor robustness [4, 5]. Likewise, indirect methods that make use of the calculus of variations are often not a viable option as they have poor convergence properties [6]. Convex optimization has been recently used to solve several space-related optimization problems, including powered descent and landing, entry, and low-thrust trajectory optimization [7, 8, 9]. This is because convex problems can be solved by means of polynomial-time algorithms and limited computational resources [10, 11, 12], and thus are perfectly suitable for real-time optimization. As the majority of the engineering problems are nonconvex, they are transformed into convex programs with two specific techniques: lossless convexification [13, 14] (also called exact convex relaxation [15]) and successive convexification [16]; the original nonconvex problem is then solved by means of an

[1]PhD Candidate, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milano, Italy, andreacarlo.morelli@polimi.it
[2]PhD Candidate, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milano, Italy, christian.hofmann@polimi.it
[3]Full Professor, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milano, Italy, francesco.topputo@polimi.it

iterative technique called Sequential Convex Programming (SCP) [17].

The control associated with the solution of the minimum-fuel low-thrust trajectory optimization problem is discontinuous. For this reason, current methods often lack strong convergence properties [5]. In indirect optimization, smoothing techniques and homotopic approaches are used to overcome this issue; a sequence of simpler, smooth problems is solved first and a continuation is performed until the original, discontinuous problem is eventually solved. A logarithmic smoothing function and a hyperbolic tangent function were considered in [18] and [19], respectively. In [20, 21], an energy-to-fuel continuation was used to generate minimum-fuel low-thrust trajectories. In [22], a cubic function of the thrust was used as homotopic path , while in [23, 24] the performance of additional smoothing functions was compared. In [25], a double-homotopy technique was developed to find the minimum-fuel transfer in the circular restricted three-body problem.

This work combines a homotopic approach with the SCP algorithm. The low-thrust trajectory optimization problem is discretized by means of an arbitrary-order Legendre–Gauss–Lobatto (LGL) quadrature scheme, and a nonlinear control interpolation is exploited to enhance the accuracy of the solution. Moreover, an adaptive second-order trust-region mechanism is developed to speed up the convergence process.

The paper is structured as follows. Section 2 states the convex low-thrust space trajectory optimization problem. Section 3 describes the arbitrary-order LGL quadrature and the nonlinear control interpolation. Section 4 details the combination of the homotopic approach and convex programming. Section 5 shows the efficacy of our strategy through several numerical simulations. Finally, Section 6 concludes the work.

2 Problem Formulation

The minimum-fuel space trajectory optimization (STO) problem aims to find the trajectory between two given points with minimum propellant expenditure. Considering two-body dynamics only, the equations of motion in Cartesian coordinates are given by

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ -\mu\mathbf{r}/r^3 + \tilde{\mathbf{u}}_\alpha/m \\ -T/(I_{\mathrm{sp}} g_0) \end{bmatrix} \tag{1}$$

where $\mathbf{r} = [r_x, r_y, r_z]^\top$, $\mathbf{v} = [v_x, v_y, v_z]^\top$, and m are the position vector, the velocity vector and the mass of the spacecraft, respectively. μ is the standard gravitational parameter of the primary body in the two-body problem, $\tilde{\mathbf{u}}_\alpha = [T_x, T_y, T_z]^\top$, $T = \|\tilde{\mathbf{u}}_\alpha\|_2$, and $\tilde{\mathbf{u}} = [\tilde{\mathbf{u}}_\alpha, T]^\top$ is the control vector. I_{sp} and g_0 are the specific impulse and gravitational acceleration, respectively. Eq. (1) is nonlinear and nonconvex; moreover, the state and control variables are coupled through the term $\tilde{\mathbf{u}}_\alpha/m$. Therefore, we transform the original minimum-fuel STO problem into a convex one [3, 5]:

$$\min_{\mathbf{u}} J_0 = \int_{t_0}^{t_f} \tau(t)\mathrm{d}t \tag{2}$$

subject to:

$$\dot{\mathbf{x}} = \mathbf{f}_f(\mathbf{x}^*) + \mathbf{A}(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + \mathbf{B}\mathbf{u} \tag{3a}$$

$$\tau_x^2 + \tau_y^2 + \tau_z^2 \leq \tau^2 \tag{3b}$$

$$0 \leq \tau \leq e^{-w^*}[1 - (w - w^*)] \tag{3c}$$

$$\mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u \tag{3d}$$

$$\mathbf{u}_{\alpha,l} \leq \mathbf{u}_\alpha \leq \mathbf{u}_{\alpha,u} \tag{3e}$$

$$\|\mathbf{x} - \mathbf{x}^*\|_1 \leq R \tag{3f}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{r}(t_f) = \mathbf{r}_f, \quad \mathbf{v}(t_f) = \mathbf{v}_f \tag{3g}$$

with states $\mathbf{x} = [\mathbf{r}, \mathbf{v}, w]^\top$ and controls $\mathbf{u} = [\tau_x, \tau_y, \tau_z, \tau]^\top$. Note that a change of variables has been exploited to replace the mass m and the thrust T such that $w = \ln m$ and $\tau = T/m$ (τ_x, τ_y and τ_z are defined accordingly). Eqs. (2) and (3) represent the convex STO problem, which will be referred to as CXP (Convex Problem) throughout this paper. The dynamics in Eq. (1) have been linearized to obtain (3a) where the superscript $(\cdot)^*$ denotes the reference trajectory. In (3a),

$$\mathbf{f}_f = [v_x, v_y, v_z, -r_x/r^3, -r_y/r^3, -r_z/r^3, 0]^\top \tag{4}$$

denotes the vector of the natural two-body dynamics, $\mathbf{A} = \partial\mathbf{f}_f/\partial\mathbf{x}$ the Jacobian matrix and $\mathbf{B}$ is such that

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{3\times4} \\ \mathbf{I}_{3\times3} \quad \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} \quad b \end{bmatrix} \tag{5}$$

where $b = 1/(I_{\mathrm{sp}} g_0)$. The relationship between the components of the control vector has been convexified in Eq. (3b); the inequality constraint on τ in Eq. (3c) has been linearized due to the change of variables [3]. The lower (subscript l) and upper bounds (subscript u) of states and controls are given in Eq. (3d) and Eq. (3e), respectively. The trust-region constraint in Eq. (3f) is used to keep the solution close to the reference and hence, the linearization valid. Initial $\mathbf{x}_0$ and final $(\mathbf{r}_f, \mathbf{v}_f)$ boundary conditions are given in Eq. (3g).

3 Convex Abritrary-Order Legendre–Gauss–Lobatto Quadrature

CXP in Eqs. (2) and (3) represents an infinte-dimensional optimal control problem. There are several discretization schemes to transform it into a finite-dimensional parameter optimization problem [26]. A popular choice are pseudospectral methods because of their spectral convergence rate for smooth problems [5]. Yet, the discretized problem is often less sparse compared to local methods such as the trapezoidal rule (linear interpolation of states and controls) and higher order methods like the Hermite–Simpson scheme (cubic interpolation of states and linear interpolation of controls). A generalization of Hermite–Simpson is the arbitrary-order Gauss–Lobatto collocation method that is widely used to solve nonlinear programs (NLP) [27]. In this paper, we use the Legendre–Gauss–Lobatto points for both the nodes and the collocation points. To the best of the authors' knowledge, this is the first time that an arbitrary-order Legendre–Gauss–Lobatto method is applied to convex programs.

3.1 State Discretization

The arbitrary-order Gauss–Lobatto collocation method approximates the state variables by means of an arbitrary-order polynomial and is therefore an extension of the Hermite–Simpson scheme. The total time of flight (ToF) is divided into I subintervals. Each time interval $[t_i, t_{i+1}]$ is mapped into the interval $[-1, 1]$ through the transformation

$$t \;\rightarrow\; \frac{h}{2}\xi + \frac{t_{i+1}+t_i}{2} \tag{6}$$

where $\xi \in [-1, 1]$ and $h = t_{i+1} - t_i$ is the time step. Inside the i-th subinterval, the state $\mathbf{x}^{(i)}(\xi) \in \mathbb{R}^{n_x \times 1}$ ($n_x = 7$) is approximated as:

$$\mathbf{x}^{(i)}(\xi) \approx \mathbf{a}_0^{(i)} + \mathbf{a}_1^{(i)}\xi + \cdots + \mathbf{a}_n^{(i)}\xi^n, \quad i = 1, \ldots, I \tag{7}$$

where the column vectors of coefficients $\mathbf{a}_m^{(i)} \in \mathbb{R}^{n_x \times 1}$, $m = 0, \ldots, n$ are unknowns that are used to approximate the state at the nodal point θ. The idea is to find the values of the coefficients $\mathbf{a}_m^{(i)}$ using the information of the states and dynamics at the nodes (the so-called *Hermite interpolation*) and eventually express the constraints at the collocation points. In this paper, nodes and collocation points are defined according to the Legendre–Gauss–Lobatto points, which are the roots of the derivative of the $(n-1)$-th degree Legendre polynomial [27]. The n-th degree Legendre polynomial is given by the Rodrigues formula as [28]

$$P_n(\xi) = \frac{(-1)^n}{2^n n!}\frac{\mathrm{d}^n}{\mathrm{d}\xi^n}(1-\xi^2)^n \tag{8}$$

and the LGL points are then

$$\hat{\boldsymbol{\xi}} = [\hat{\xi}_1 = -1, \hat{\xi}_2, \ldots, \hat{\xi}_i, \ldots, \hat{\xi}_{n-1}, \hat{\xi}_n = 1] \tag{9}$$

where $\hat{\xi}_i$ $(i = 2, \ldots, n-1)$ is the i-th root of the polynomial $\mathrm{d}P_{n-1}(\xi)/\mathrm{d}\xi$. The nodes θ_j and collocation points ζ_j are defined as [26]

$$\theta_j = \hat{\xi}_{2j-1}, \quad j = 1, \ldots, (n+1)/2 \tag{10}$$

and

$$\zeta_j = \hat{\xi}_{2j}, \quad j = 1, \ldots, (n-1)/2 \tag{11}$$

Note that only odd orders of polynomials $n \geq 3$ can be considered, since the number of collocation points inside each interval is $(n-1)/2$ [27].
Considering the information at the nodes, the following linear system can be written for the i-th trajectory segment [27**?**]:

$$\underbrace{\begin{bmatrix} \mathbf{I}_{n_x} & \ldots & \theta_1^n \mathbf{I}_{n_x} \\ \vdots & \vdots & \vdots \\ \mathbf{I}_{n_x} & \ldots & \theta_{n_p}^n \mathbf{I}_{n_x} \\ \mathbf{0}_{n_x} & \ldots & n\theta_2^{n-1}\mathbf{I}_{n_x} \\ \vdots & \vdots & \vdots \\ \mathbf{0}_{n_x} & \ldots & n\theta_{n_p}^{n-1}\mathbf{I}_{n_x} \end{bmatrix}}_{\boldsymbol{\theta}} \underbrace{\begin{bmatrix} \mathbf{a}_0^{(i)} \\ \vdots \\ \mathbf{a}_{n_p}^{(i)} \\ \vdots \\ \mathbf{a}_{n-1}^{(i)} \\ \mathbf{a}_n^{(i)} \end{bmatrix}}_{\mathbf{a}^{(i)}} = \underbrace{\begin{bmatrix} \mathbf{x}^{(i)}(\theta_1) \\ \vdots \\ \mathbf{x}^{(i)}(\theta_{n_p}) \\ \frac{h}{2}\mathbf{f}_l^{(i)}(\theta_1) \\ \vdots \\ \frac{h}{2}\mathbf{f}_l^{(i)}(\theta_{n_p}) \end{bmatrix}}_{\mathbf{b}^{(i)}} \tag{12}$$

where θ_j are the positions of the nodal points as in Eq. (10), $n_p = (n+1)/2$ is the number of nodes in each time interval, $\mathbf{I}_{n_x}$ is the $n_x \times n_x$ identity matrix and $\mathbf{0}_{n_x}$ the $n_x \times n_x$ null matrix. Note that $\mathbf{f}_l(\theta_j)$ is the two-body dynamics as given in (3a) and thus

$$\mathbf{f}_l(\theta_j) = \mathbf{f}_f(\mathbf{x}_j^*) + \mathbf{A}(\mathbf{x}_j^*)(\mathbf{x}_j - \mathbf{x}_j^*) + \mathbf{B}\mathbf{u}_j \tag{13}$$

where the subscript $(\cdot)_j$ indicates that the considered quantity is evaluated at the node θ_j and the superscript $(\cdot)^{(i)}$ has been dropped for simplicity. This linearization is required due to our convex approach. The linear system in (12) can be written in compact form as $\boldsymbol{\theta}\mathbf{a}^{(i)} = \mathbf{b}^{(i)}$ and hence $\mathbf{a}^{(i)} = \boldsymbol{\theta}^{-1}\mathbf{b}^{(i)}$, with $\boldsymbol{\theta} \in \mathbb{R}^{2n_x n_p \times 2n_x n_p}$, $\mathbf{a}^{(i)} \in \mathbb{R}^{2n_x n_p \times 1}$ and $\mathbf{b}^{(i)} \in \mathbb{R}^{2n_x n_p \times 1}$. The state at the collocation points is defined as

$$\mathbf{x}^{(i)}(\zeta) = [\mathbf{x}^{(i)}(\zeta_1), \ldots, \mathbf{x}^{(i)}(\zeta_{n_c})]^\top \in \mathbb{R}^{n_c n \times 1} \tag{14}$$

where $n_c = (n-1)/2$ is the number of collocation points in one subinterval. Considering Eq. (7) and recalling

that $\mathbf{a}^{(i)} = \boldsymbol{\theta}^{-1}\mathbf{b}^{(i)}$, the state can be calculated as

$$\begin{aligned}\mathbf{x}^{(i)}(\zeta) &= \underbrace{\begin{bmatrix} \mathbf{I}_{n_x} & \zeta_1\mathbf{I}_{n_x} & \dots & \zeta_1^n\mathbf{I}_{n_x} \\ \mathbf{I}_{n_x} & \zeta_2\mathbf{I}_{n_x} & \dots & \zeta_2^n\mathbf{I}_{n_x} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{I}_{n_x} & \zeta_{n_c}\mathbf{I}_{n_x} & \dots & \zeta_{n_c}^n\mathbf{I}_{n_x} \end{bmatrix}}_{\boldsymbol{\zeta}} \underbrace{\begin{bmatrix} \mathbf{a}_0^{(i)} \\ \mathbf{a}_1^{(i)} \\ \vdots \\ \mathbf{a}_n^{(i)} \end{bmatrix}}_{\mathbf{a}^{(i)}} \\ &= \boldsymbol{\zeta}\boldsymbol{\theta}^{-1}\mathbf{b}^{(i)} = \boldsymbol{\phi}\mathbf{b}^{(i)}\end{aligned} \tag{15}$$

where $\boldsymbol{\zeta} \in \mathbb{R}^{n_c n_x \times n_x n}$ and $\boldsymbol{\phi} \equiv \boldsymbol{\zeta}\boldsymbol{\theta}^{-1}$. The derivative of the state at the collocation points can therefore be written as

$$\begin{aligned}\frac{\mathrm{d}\mathbf{x}^{(i)}(\zeta)}{\mathrm{d}\xi} &= \underbrace{\begin{bmatrix} \mathbf{0}_{n_x} & \mathbf{I}_{n_x} & \dots & n\zeta_1^{n-1}\mathbf{I}_{n_x} \\ \mathbf{0}_{n_x} & \mathbf{I}_{n_x} & \dots & n\zeta_2^{n-1}\mathbf{I}_{n_x} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{n_x} & \mathbf{I}_{n_x} & \dots & n\zeta_{n_c}^{n-1}\mathbf{I}_{n_x} \end{bmatrix}}_{\boldsymbol{\zeta}'} \underbrace{\begin{bmatrix} \mathbf{a}_0^{(i)} \\ \mathbf{a}_1^{(i)} \\ \vdots \\ \mathbf{a}_n^{(i)} \end{bmatrix}}_{\mathbf{a}^{(i)}} \\ &= \boldsymbol{\zeta}'\boldsymbol{\theta}^{-1}\mathbf{b}^{(i)} = \boldsymbol{\phi}'\mathbf{b}^{(i)}\end{aligned} \tag{16}$$

Note that the left-hand side of the overlying equation indicates the derivative of the state with respect to the independent variable ξ at the collocation points ζ.
These expressions can easily be extended to all trajectory segments by defining the vectors of unknown coefficients $\hat{\mathbf{a}}$ and constant terms $\hat{\mathbf{b}}$ as

$$\begin{aligned}\hat{\mathbf{a}} &= [\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(i)}, \dots, \mathbf{a}^{(I)}]^\top \\ \hat{\mathbf{b}} &= [\mathbf{b}^{(1)}, \dots, \mathbf{b}^{(i)}, \dots, \mathbf{b}^{(I)}]^\top\end{aligned} \tag{17}$$

Note that the first node of the i-th subsegment and the last node of the $(i-1)$-th subsegment are equal. Consequently, the vectors $\mathbf{x}^{(i)}(\theta_1)$ and $\mathbf{f}_l^{(i)}(\theta_1)$ in $\mathbf{b}^{(i)}$ are the same as the vectors $\mathbf{x}^{(i-1)}(\theta_{n_p})$ and $\mathbf{f}_l^{(i-1)}(\theta_{n_p})$ in $\mathbf{b}^{(i-1)}$ for $i = 2, \dots, I-1$. Extending the linear system in Eq. (12) to all trajectory segments yields

$$\boldsymbol{\Theta}\hat{\mathbf{a}} = \hat{\mathbf{b}} \iff \hat{\mathbf{a}} = \boldsymbol{\Theta}^{-1}\hat{\mathbf{b}} \tag{18}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{2In_x n_p \times 2In_x n_p}$ is a diagonal matrix with $\boldsymbol{\theta}$ on the main diagonal. Defining $\hat{\mathbf{x}}(\zeta) = [\mathbf{x}^{(1)}(\zeta), \mathbf{x}^{(2)}(\zeta), \dots \mathbf{x}^{(i)}(\zeta), \dots, \mathbf{x}^{(I)}(\zeta)]^\top$ as the column vector of concatenated states at the collocation points, we obtain

$$\hat{\mathbf{x}}(\zeta) = \underbrace{\mathbf{Z}\boldsymbol{\Theta}^{-1}}_{\boldsymbol{\Phi}}\hat{\mathbf{b}} \equiv \boldsymbol{\Phi}\hat{\mathbf{b}} \tag{19}$$

$$\frac{\mathrm{d}\hat{\mathbf{x}}(\zeta)}{\mathrm{d}\xi} = \underbrace{\mathbf{Z}'\boldsymbol{\Theta}^{-1}}_{\boldsymbol{\Phi}'}\hat{\mathbf{b}} \equiv \boldsymbol{\Phi}'\hat{\mathbf{b}} \tag{20}$$

$\mathbf{Z}$ and $\mathbf{Z}'$ are diagonal matrices with $\boldsymbol{\zeta}$ and $\boldsymbol{\zeta}'$ on the main diagonal, respectively. These expressions are now used to write the defect constraints $\boldsymbol{\Delta}$ in a compact way:

$$\boldsymbol{\Delta} = \boldsymbol{\Phi}'\hat{\mathbf{b}} - \frac{h}{2}[\hat{\mathbf{f}}_f + \hat{\mathbf{A}}(\boldsymbol{\Phi}\hat{\mathbf{b}} - \boldsymbol{\Phi}\hat{\mathbf{b}}^*)] + \hat{\mathbf{B}}\hat{\mathbf{u}}(\zeta) = \mathbf{0} \tag{21}$$

with

$$\hat{\mathbf{f}}_f = [\mathbf{f}_{f,1}^{(1)}, \dots, \mathbf{f}_{f,n_c}^{(1)}, \dots, \mathbf{f}_{f,1}^{(I)}, \dots \mathbf{f}_{f,n_c}^{(I)}]^\top \tag{22}$$

$$\begin{aligned}\hat{\mathbf{A}} &= \begin{bmatrix} \mathbf{A}(\zeta_1) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}(\zeta_2) & & \vdots \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \dots & \dots & \mathbf{A}(\zeta_p) \end{bmatrix} \\ \hat{\mathbf{B}} &= \begin{bmatrix} \mathbf{B} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{B} & & \vdots \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \dots & \dots & \mathbf{B} \end{bmatrix}\end{aligned} \tag{23}$$

$p = In_c$ is the total number of collocation points and $\mathbf{A}(\zeta_j) = \mathbf{A}(\mathbf{x}(\zeta_j))$. The concatenated control vector $\hat{\mathbf{u}}(\zeta)$ will be described in the next section.

Remark 1: Different from the conventional LGL scheme, we include the initial and final points in the optimization to easily account for the initial and final boundary conditions.

3.2 Nonlinear Control Interpolation

Unlike the state, the control variables are often linearly interpolated in direct collocation methods [26]. As this may result in a poor approximation, we approximate the controls $\mathbf{u}^{(i)}(\theta) \in \mathbb{R}^{4\times 1}$ in the i-th subinterval as a polynomial of degree $n_p - 1$:

$$\mathbf{u}^{(i)}(\theta) \approx \mathbf{a}_0^{(i)} + \mathbf{a}_1^{(i)}\theta + \dots + \mathbf{a}_{n_p-1}^{(i)}\theta^{n_p-1} \tag{24}$$

Similar to (15), the controls at the collocation points are

$$\begin{aligned}\mathbf{u}^{(i)}(\zeta) &= \underbrace{\begin{bmatrix} \mathbf{I}_{n_u} & \zeta_1\mathbf{I}_{n_u} & \dots & \zeta_1^{n_p-1}\mathbf{I}_{n_u} \\ \mathbf{I}_{n_u} & \zeta_2\mathbf{I}_{n_u} & \dots & \zeta_2^{n_p-1}\mathbf{I}_{n_u} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{I}_{n_u} & \zeta_{n_c}\mathbf{I}_{n_u} & \dots & \zeta_{n_c}^{n_p-1}\mathbf{I}_{n_u} \end{bmatrix}}_{\boldsymbol{\zeta}_u} \underbrace{\begin{bmatrix} \mathbf{a}_0^{(i)} \\ \mathbf{a}_1^{(i)} \\ \vdots \\ \mathbf{a}_{n_p-1}^{(i)} \end{bmatrix}}_{\mathbf{a}_u^{(i)}} \\ &= \boldsymbol{\zeta}_u\boldsymbol{\theta}_u^{-1}\mathbf{b}_u^{(i)} = \boldsymbol{\phi}_u\mathbf{b}_u^{(i)}\end{aligned} \tag{25}$$

where $n_u = 4$ is the number of control variables and the subscript $(\cdot)_u$ refers to expressions related to controls instead of states. Combining the controls at all collocation points into one vector $\hat{\mathbf{u}}(\zeta) = [\mathbf{u}^{(1)}(\zeta), \mathbf{u}^{(2)}(\zeta), \dots \mathbf{u}^{(i)}(\zeta), \dots, \mathbf{u}^{(I)}(\zeta)]^\top$, we can write

$$\hat{\mathbf{u}}(\zeta) = \mathbf{Z}_u\boldsymbol{\Theta}_u^{-1}\hat{\mathbf{b}}_u = \boldsymbol{\Phi}_u\hat{\mathbf{b}}_u \tag{26}$$

with similar notation as in Section 3.1. Eq. (21) can now be rewritten to obtain

$$\boldsymbol{\Delta} = \boldsymbol{\Phi}'\hat{\mathbf{b}} - \frac{h}{2}[\hat{\mathbf{f}}_f + \hat{\mathbf{A}}(\boldsymbol{\Phi}\hat{\mathbf{b}} - \boldsymbol{\Phi}\hat{\mathbf{b}}^*)] + \hat{\mathbf{B}}\boldsymbol{\Phi}_u\hat{\mathbf{b}}_u = \mathbf{0} \quad (27)$$

Remark 2: Our simulations show that due to numerical reasons, only LGL orders for which the interpolating polynomial is of odd degree (i.e. when $n_p - 1$ is an odd number) can be considered.

4 Homotopic Approach and Convex Programming

In this section, we present the sequential convex programming algorithm combined with a homotopic approach that considers continuation from the minimum-energy to the minimum-fuel problem to enhance the overall robustness. This method is similar to the one employed in literature to improve numerical convergence of indirect methods (see for example [23]).

4.1 Sequential Convex Programming

The original nonconvex problem is solved by iteratively solving a sequence of convex subproblems. As the dynamical and thrust constraints have been linearized, the resulting problem may become infeasible even though the original problem is feasible. To avoid this *artificial infeasibility*, we add unconstrained slack variables $\boldsymbol{\nu}$ and η in Eqs. (3a) and (3c) to obtain [5]

$$\dot{\mathbf{x}} = \mathbf{f}_f(\mathbf{x}^*) + \mathbf{A}(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + \mathbf{B}\mathbf{u} + \boldsymbol{\nu} \quad (28)$$

$$0 \leq \tau \leq e^{-w^*}[1 - (w - w^*)] + \eta \quad (29)$$

Note that $\boldsymbol{\nu}$ and η must be zero at the end of the optimization process so that the constraints are satisfied. Therefore, the objective function J_0 in (2) is augmented with two terms [5]

$$J = J_0 + \sum_{i \in I_{eq}} \mu_i ||\boldsymbol{\nu}_i||_1 + \sum_{i \in I_{ineq}} \lambda_i \max(0, \eta_i) \quad (30)$$

where μ_i and λ_i are two sufficiently large parameters. I_{eq} and I_{ineq} are the set of equality and inequality constraints, respectively.

One key feature of the SCP algorithm is the trust-region mechanism to keep the solution close to the reference and thus the linearization valid. The ratio of the actual cost reduction $\Delta\phi_k = \phi_{k-1} - \phi_k$ to the predicted cost reduction $\Delta\hat{\phi}_k = \hat{\phi}_{k-1} - \hat{\phi}_k$ (with k denoting the current iteration) serves as a measure to decide whether the step is to be accepted or not. At each iteration, the merit functions ϕ_k and $\hat{\phi}_k$ are calculated as [29]

$$\phi_k = J_0 + \sum_{j \in I_{eq}} \mu_j ||\mathbf{h}_j||_1 + \sum_{j \in I_{ineq}} \lambda_j \max(0, g_j) \quad (31)$$

$$\hat{\phi}_k = J_0 + \sum_{j \in I_{eq}} \mu_j ||\boldsymbol{\nu}_j||_1 + \sum_{j \in I_{ineq}} \lambda_j \max(0, \eta_j) + \sum_{j \in I_{ineq}} \lambda_j \max(0, \sigma_j) \quad (32)$$

$\mathbf{h}_j$ and g_j are the constraint violations of the equality and inequality constraints, respectively, of the original, nonconvex problem. In contrast, (32) refers to the violations of the convex problem with $\sigma_j = \tau_{x,j}^2 + \tau_{y,j}^2 + \tau_{z,j}^2 - \tau_j^2$. The interested reader is referred to [5, 17] for a more detailed description of the standard SCP algorithm.

4.2 Homotopic Sequential Convex Programming

Instead of solving the minimum-fuel problem directly, a sequence of simpler and smoother problems is solved first in a homotopic approach. These solutions serve as initial guesses for the next subproblem where the complexity gradually increases until the original problem is eventually solved. One of the most common continuations is the energy-to-fuel homotopy which uses the objective function [20]

$$J_\gamma \equiv J(\gamma) = \int_{t_0}^{t_f} [(1 - \gamma)\tau + \gamma\tau^2] \, \mathrm{d}t \quad (33)$$

with some parameter $\gamma \in [0, 1]$ that defines the homotopic path from the minimum-energy ($\gamma = 1$) to the minimum-fuel ($\gamma = 0$) problem. τ is the acceleration magnitude defined in Section 2. As it will be pointed out in Section 5, we used a convex solver that requires the problem to be a Conic Linear Program (CLP) [11]; consequently, it was necessary to reformulate the objective function to make it a linear function of the optimization variables. The interested reader is invited to consult [30] for more information.

In this work, we start solving the minimum-energy problem; then, γ is gradually decreased until feasibility with respect to the original nonlinear constraints is reached. Note that we change the homotopic parameter γ only when a step is accepted; however, to speed up convergence, if the maximum nonlinear constraint violation

$c_{\max}$ reaches the value of $10^3\varepsilon_c$ (where ε_c is the convergence threshold) before the maximum number of homotopic steps is reached, we set $\gamma = 0$ anyway.

The definition of the quantities ϕ_k and $\hat{\phi}_k$ in Eqs. (31) and (32) includes the objective function J_0; however, the values of J_0 cannot be compared while γ is still changing, simply because they refer to different problems. To avoid that the algorithm gets stuck for this reason, we substitute the term related to the objective function inside Eqs. (31) and (32) with a term associated with the final spacecraft mass. The aforementioned quantities are thus redefined as

$$\phi_k^M = \mu_{\mathrm{m}}(e^{w_0} - e^{w_f}) + \sum_{j\in I_{eq}} \mu_j ||\mathbf{h}_j||_1 + \sum_{j\in I_{ineq}} \lambda_j \max(0, g_j) \tag{34}$$

$$\hat{\phi}_k^M = \mu_{\mathrm{m}}(e^{w_0} - e^{w_f}) + \sum_{j\in I_{eq}} \mu_j ||\boldsymbol{\nu}_j||_1 + \sum_{j\in I_{ineq}} \lambda_j \max(0, \eta_j) + \sum_{j\in I_{ineq}} \lambda_j \max(0, \sigma_j) \tag{35}$$

where μ_{m} is a parameter such that $1 < \mu_{\mathrm{m}} \ll \mu_i, \lambda_i$.

4.3 Adaptive Second-Order Trust-Region Radius Change

In standard trust-region methods such as in [3, 17], a step is rejected if $\rho_k \equiv \Delta\phi_k/\Delta\hat{\phi}_k < \rho_0$ because this indicates that there is no (sufficiently large) progress. When a solution is accepted, the trust region is updated as follows:

$$R_{k+1} = \begin{cases} R_k/\alpha & \text{if} \quad \rho_0 \le \rho_k < \rho_1 \\ R_k & \text{if} \quad \rho_1 \le \rho_k < \rho_2 \\ \beta R_k & \text{if} \quad \rho_k \ge \rho_2 \end{cases} \tag{36}$$

where α and β are two constants greater than 1 and $0 < \rho_0 < \rho_1 < \rho_2 < 1$. Updating the trust-region radius like this often works well, but it is not particularly flexible for two reasons. First, the constants α and β must be selected by the user and are fixed during the optimization process. Secondly, the update only depends on the value of the parameter ρ at the current iteration k, without considering some potentially useful information from previous iterations.

We extend the work in [29] and propose an improved method where α and β can vary based on the values of ρ in the current k and previous iteration $k-1$. Introducing the new constant parameter δ with $1 < \delta < \alpha, \beta$, we define the update mechanism of α and β as follows:

1. If $\rho_k \ge \rho_0$ and $\rho_{k-1} \ge \rho_0$ (that is, both the current and the previous steps are accepted), then $\beta_k = \delta\beta_{k-1}$ and $\alpha_k = \alpha_{k-1}/\delta$. Our rationale is that if two subsequent steps are accepted, the algorithm will benefit from a larger increase of the trust region in the next iteration.
2. If $\rho_k \ge \rho_0$ and $\rho_{k-1} < \rho_0$ (that is, the current step is accepted but the previous was not), then $\beta_k = \beta_{k-1}/\delta$ and $\alpha_k = \delta\alpha_{k-1}$. This suggests to stay closer to the current solution to avoid a rejected step in the next iteration.
3. If $\rho_k < \rho_0$ and $\rho_{k-1} \ge \rho_0$ (that is, the current step is rejected whereas the previous one was accepted), then $\alpha_k = \alpha_{k-1}$ and $\beta_k = \beta_{k-1}$. If on one hand the current iteration is rejected, on the other hand the previous step was accepted, it is thus convenient to neither increase nor decrease the parameters.
4. If $\rho_k < \rho_0$ and $\rho_{k-1} < \rho_0$ (that is, both the current and the previous steps are rejected), then $\alpha_k = \delta\alpha_{k-1}$ to stay closer to the reference solution and speed up convergence.

The parameters α and β increase the degrees of freedom of the algorithm and become part of the optimization process. Note that we impose $\alpha_{\min} < \alpha < \alpha_{\max}$ and $\beta_{\min} < \beta < \beta_{\max}$. The effectiveness of this approach is shown in Section 5.

5 Simulations and Results

We have already stated that the contribution of this work is threefold: the introduction of an adaptive trust region radius update; the development of a LGL discretization scheme applied to convex environments; the inclusion of an energy-to-fuel homotopic path. We thus perform three different analyses to show how our strategies are effectively increasing the performance of the standard SCP algorithm.

The minimum-fuel transfer from Earth to Venus is considered in this work. This problem is shown for the sake of comparison with previous works [5]. We formulated the problem in standard form so that it could be solved using the embedded conic solver ECOS [11]. . We consider two-body dynamics without any additional perturbations, assume a constant specific impulse and use the gravitational constant $\mu = 1.32712440018 \times 10^{20}$ m^3/s^2 and gravitational acceleration at sea level $g_0 = 9.80665$ m/s^2. The problem is scaled with the quantities given in Tab. 2. Relevant parameters of the Earth to Venus transfer and the SCP algorithm are given in Tabs. 3 and 1, respectively. In particular, the quantities ε_c and ε_ϕ in Tab. 1 refer to the thresholds that must be overcome by the maximum constraint violation and the quantity

$\Delta\phi$ for the algorithm to converge, respectively. We use a simple cubic interpolation approach to generate the initial guess. This method requires the user to specify the number of revolutions. In this paper, the nominal case with $N_{\text{rev}} = 2$ is considered. In all robustness analyses, we perturb the nominal initial guess by slightly changing the value of N_{rev} (with random perturbations inside the interval $[-10^{-2}, 10^{-2}]$). Figs. 1a and 1b show the typical thrust profile and transfer trajectory obtained with the SCP algorithm. The thrust profile has a bang-bang structure, indicating that the obtained solution is at least suboptimal. The final spacecraft mass in the nominal case is 1035 kg, consistent with the values in literature [5].

Parameter	**Value**
Penalty weights λ, μ	100
Penalty weight μ_{m}	3
Trust region r_0	100
ρ_0, ρ_1, ρ_2	$0.01, 0.25, 0.9$
α_0, β_0	1.4
δ	1.3
$\alpha_{\min}$, $\beta_{\min}$	1.01
$\alpha_{\max}$, $\beta_{\max}$	5.2
ε_c	10^{-5}
ε_ϕ	10^{-3}
Max. iterations	200

Tab. 1: Parameters of the algorithm.

5.1 Adaptive Trust-Region Radius Update Analysis

Throughout this section, SCP refers to the standard algorithm with fixed α and β for the trust-region mechanism whereas SCP_{ATR} uses the adaptive second-order trust-region update described in Section 4.3. We use an Hermite–Simpson discretization method (equivalent to a LGL scheme with $n = 3$) to perform a robustness analysis on both the SCP and SCP_{ATR} algorithms and compare their performances. In particular, Tab. 4 shows that the SCP_{ATR} algorithm was able to converge 34% more times than the simple SCP. The CPU time and iterations required by the SCP_{ATR} algorithm are only $\approx 20\%$ the ones required by the standard SCP. Moreover, the SCP_{ATR} found slightly lower final masses; however, we recall that optimality is less relevant with respect to robustness and computational efficiency considering on-board guidance scenarios.

5.2 Legendre–Gauss–Lobatto Analysis

Section 5.1 pointed out how the adaptive trust-region radius update outperforms the standard SCP method in terms of convergence rate and computational effort; however, it also highlighted that a simple Hermite–Simpson interpolation may be inappropriate when complex trajectories need to be designed, since only 35% of the considered cases converged to a feasible solution. Therefore, a robustness analysis on the arbitrary-order LGL discretization with 100 perturbed cases and using the newly-developed SCP_{ATR} algorithm was performed. Results show that higher-orders polynomial interpolation can effectively increase the convergence rates of the algorithm: remarkably, higher-order LGL interpolation increased convergence by $59\% - 65\%$. This was at the expense of a higher computational time, which may be due to the fact that the matrices defined in Section 3 are denser as n increases. The optimality of the solutions remains approximately constant when changing n. Note that if the order of the interpolating polynomials becomes too high, the obtained thrust profiles become less accurate, i.e. they present shapes that are far from being bang-off-bang; for this reason, we stopped our robustness analysis at $n = 15$.

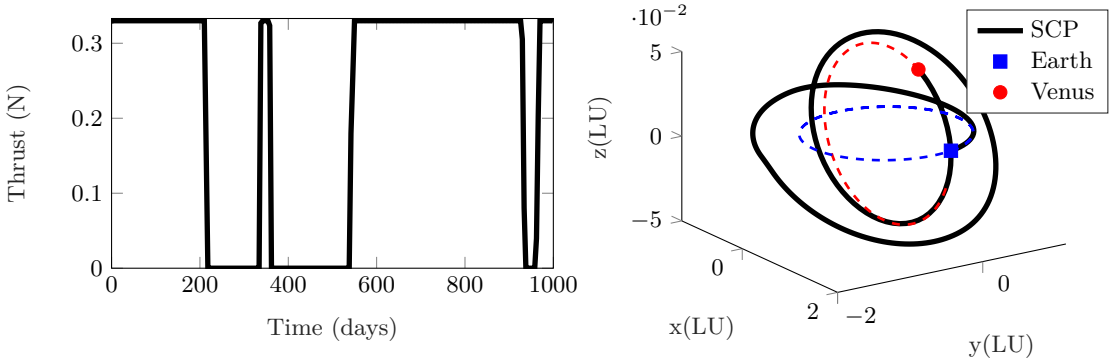
5.3 Homotopic Approach Analysis

We address the efficacy of the proposed homotopic approach performing a robustness analysis with 100 perturbed cases using a 3^{rd}-order LGL discretization. This is because we do not want the LGL order to interfere with the efficacy of the homotopic strategy we developed. We compare the results obtained solving the minimum-energy problem (associated with $\gamma = 1$), the homotopic minimum-fuel problem with a number of homotopic steps $s = 20$ and the pure minimum-fuel problem. Fig. 3 illustrates the thrust profiles associated with different values of the smoothing parameter γ. It can be observed that the minimum-energy thrust profile is smooth, and that the smoothness decreases as the homotopic parameter approaches zero. Note that the figures show the profiles obtained by solving the associated problems without any homotopy.

Results are presented in Fig. 2 and Tab. 6 where the converged cases, average final mass, average iterations and average CPU time are compared. Solving the minimum-energy problem always resulted in a converged solution; moreover, the homotopic approach was able to increase the convergence rate of the minimum-fuel problem by 12%. Iterations and computational time are similar for the minimum-energy and minimum-fuel problems, while they are considerably higher for the homotopic problem. This may be due to the way we defined the transition be-

Physical Quantity	Normalization Factor	Value
Length (m)	LU = 1 AU	$1.49597870 \times 10^{11}$
Velocity (m/s)	VU = $\sqrt{\mu/\text{LU}}$	2.97846918×10^{4}
Time (s)	TU = LU/VU	$5.022642856 \times 10^{6}$
Acceleration (m/s^2)	ACU = VU^2/LU	5.930083×10^{-3}
Mass (kg)	MU = m_0	1500
Force (N)	FU = $T_{\max}$	0.33

Tab. 2: Physical quantities of the problem.



(a) Thrust profile.

(b) Transfer trajectory.

Fig. 1: Thrust profile and transfer trajectory of the Earth to Venus transfer.

Parameter	Value
$\mathbf{r}_0 = [r_x, r_y, r_z]_0^\top$ (LU)	$[0.9708, 0.2376, 0]^\top$
$\mathbf{v}_0 = [v_x, v_y, v_z]_0^\top$ (VU)	$[-0.2545, 0.9687, 0]^\top$
m_0 (MU)	1
$\mathbf{r}_f = [r_x, r_y, r_z]_f^\top$ (LU)	$[-0.3277, 0.6389, 0.0277]^\top$
$\mathbf{v}_f = [v_x, v_y, v_z]_f^\top$ (VU)	$[-1.0509, -0.5436, 0.0532]^\top$
$T_{\max}$ (FU)	1
I_{sp} (TU)	7.5657×10^{-4}
ToF (days)	1000

Tab. 3: Parameters of Earth to Venus transfer.

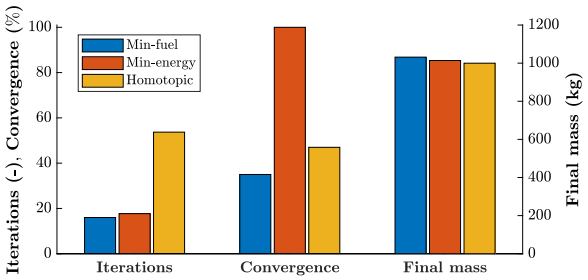


Fig. 2: Performance of SCP$_{\text{ATR}}$ for minimum-fuel, minimum-energy and homotopic minimum-fuel problems with perturbed initial guesses.

tween the minimum-energy and the minimum-fuel problem (see Section 4.2). Still, only few seconds are required to solve the considered problems. The final spacecraft mass obtained by the different problems are very similar, even though the minimum-fuel problem performed best among the others.

6 Conclusions

The objective of this paper was to present strategies to improve the robustness and the computational efficiency of the standard SCP algorithm. Results show that the high-order LGL discretization scheme increases the convergence rate of the simple Hermite–Simpson rule; moreover, the newly-developed adaptive trust-region radius update has shown superiority in terms of computational time with respect to the standard approach; finally, the combination of a homotopic approach with convex programming also increased the convergence properties of the minimum-fuel space trajectory optimization problem.

Overall, the proposed algorithm represents a promising alternative to standard nonlinear programming methods in scenarios like deep-space cruise where robustness and convergence are more important than high accuracy. The rapid speed makes our method an ideal choice for preliminary studies and also real-time applications.

Algorithm	**Iters.**	$\mathbf{m}(t_f)$	**CPU time** (s)	**Convergence** (%)
SCP	84	1037.1	6.94	1
SCP_{ATR}	16	1031.6	1.27	35

Tab. 4: Comparison of SCP and SCP_{ATR} algorithms.

n	**I**	**N**	**Iters.**	$\mathbf{m}(\mathbf{t_f})$ (kg)	**CPU time** (s)	**Convergence** (%)
3	119	120	16	1031.6	1.27	35
7	40	121	19.8	982.3	3.55	94
11	24	121	19.1	1037.4	3.55	100
15	17	120	19.4	1016.7	5.51	96

Tab. 5: Comparison of different LGL orders.

Problem	**Iters.**	$\mathbf{m}(t_f)$	**CPU time** (s)	**Convergence**(%)
Minimum-energy	17.7	1013.8	1.65	100
Homotopic	53.7	999.8	4.76	47
Minimum-fuel	16	1031.6	1.27	35

Tab. 6: Comparison of minimum-energy, homotopic and minimum-fuel algorithms.

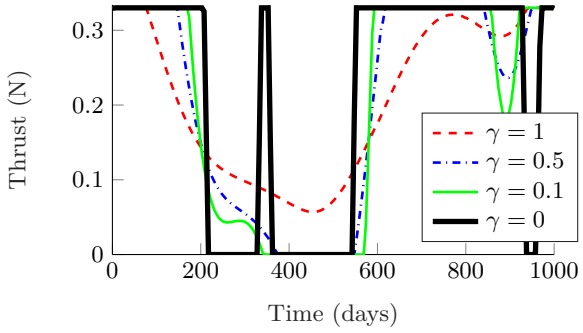


Fig. 3: Thrust profiles obtained with $\gamma = 1$ (dashed line), $\gamma = 0.5$ (dashed-dotted line), $\gamma = 0.1$ (solid line), and $\gamma = 0$ (solid bold line).

Acknowledgments

This research is part of EXTREMA, a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 864697).

Bibliography

[1] K. Woellert, P. Ehrenfreund, A. J. Ricco, and H. Hertzfeld, "Cubesats: Cost-effective science and technology platforms for emerging and developing nations," *Advances in Aerospace Research*, vol. 47, no. 4, pp. 663–684, Feb. 2011.

[2] R. Walker, D. Binns, C. Bramanti, M. Casasco, P. Concari, D. Izzo, D. Feili, P. Fernandez, J. G. Fernandez, P. Hager, D. Koschny, V. Pesquita, N. Wallace, I. Carnelli, M. Khan, M. Scoubeau, and D. Taubert, "Deep-space cubesats: Thinking inside the box," *Astronomy & Geophysics*, vol. 59, no. 5, pp. 5–24, Oct. 2018.

[3] Z. Wang and M. J. Grant, "Minimum-fuel low-thrust transfers for spacecraft: A convex approach," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2274–2290, Mar. 2018.

[4] B. A. Conway, "A survey of methods available for the numerical optimization of continuous dynamic systems," *Journal of Optimization Theory and Applications*, vol. 152, no. 2, p. 271–306, Sep. 2012.

[5] C. Hofmann and F. Topputo, "Rapid low-thrust trajectory optimization in deep space based on convex programming," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 7, pp. 1379–1388, Apr. 2021.

[6] B. Pan, P. Lu, X. Pan, and Y. Ma, "Double-homotopy method for solving optimal control problems," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 8, pp. 1706–1720, Jun. 2016.

[7] M. Sagliano, "Generalized *hp* pseudospectral-convex programming for powered descent and landing," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 7, pp. 1562–1570, Apr. 2019.

[8] X. Liu, Z. Shen, and P. Lu, "Entry trajectory optimization by second-order cone programming," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 2, pp. 227–241, Aug. 2016.

[9] Y. Song and S. Gong, "Solar-sail deep space trajectory optimization using successive convex programming," *Astrophysics and Space Science*, vol. 364, no. 106, Jul. 2019.

[10] X. Liu, P. Lu, and B. Pan, "Survey of convex optimization for aerospace applications," *Astrodynamics*, vol. 1, no. 1, pp. 23–40, Sep. 2017.

[11] A. Domahidi, E. Chu, and S. Boyd, "Ecos: An socp solver for embedded systems," in *Proceedings of European Control Conference*, Zurich, Switzerland, 2013, pp. 3071–3076.

[12] S. Michael, C. A. Pascucci, D. Dueri, and B. Açıkmeşe, "Convexification and real-time onboard optimization for agile quad-rotor maneuvering and obstacle avoidance," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver, BC, Canada, Sep. 2017, pp. 4862–4868.

[13] M. W. Harris and B. Açıkmeşe, "Lossless convexification for a class of optimal control problems with linear state constraints," in *Proceedings of* 52^{nd} *IEEE Conference on Decision and Control*, Florence, Italy, Dec. 2013, pp. 7113–7118.

[14] B. Açıkmeşe, J. M. Carson, and L. Blackmore, "Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2104–2113, Nov. 2013.

[15] X. Liu, Z. Shen, and P. Lu, "Exact convex relaxation for optimal flight of aerodynamically controlled missiles," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1881–1892, May 2016.

[16] Y. Mao, M. Szmuk, and B. Açıkmeşe, "Successive convexification of non-convex optimal control problems and its convergence properties," in *Proceedings of the* 55^{th} *IEEE Conference on Decision and Control*, Las Vegas, NV, USA, Dec. 2016, pp. 3636–3641.

[17] Y. Mao, M. Szmuk, X. Xu, and B. Açıkmeşe, "Successive convexification: A superlinearly convergent algorithm for non-convex optimal control problems," Preprint, submitted Feb. 2019, https://arxiv.org/abs/1804.06539.

[18] E. Taheri, I. Kolmanovsky, and E. Atkins, "Enhanced smoothing technique for indirect optimization of minimum-fuel low-thrust trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 11, pp. 2500–2511, Sep. 2016.

[19] E. Taheri and J. L. Junkins, "Generic smoothing for optimal bang-off-bang spacecraft maneuvers," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 11, pp. 2467–2472, Sep. 2018.

[20] F. Jiang, H. Baoyin, and J. Li, "Practical techniques for low-thrust trajectory optimization with homotopic approach," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 1, pp. 245–258, Aug. 2012.

[21] C. Zhang, F. Topputo, F. Bernelli-Zazzera, and Y.-S. Zhao, "Low-thrust minimum-fuel optimization in the circular restricted three-body problem," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 8, pp. 1501–1510, Mar. 2015.

[22] J. F. Li, Z. Che, and H. N. Li, "Applications of homotopic approach in variable-specific-impulse low-thrust trajectory optimization," in *Proceedings of IEEE Chinese Guidance, Navigation and Control Conference*, Aug. 2014, pp. 2100–2104.

[23] E. Taheri, E. M. Atkins, and I. Kolmanovsky, "Performance comparison of smoothing functions for indirect optimization of minimum-fuel low-thrust trajectories," in *Proceedings of Space Flight Mechanics Meeting, AIAA SciTech Forum, AIAA Paper AIAA 2018-0214*, Kissimmee, FL, USA, Jan. 2018.

[24] R. Bertrand and R. Epenoy, "New smoothing techniques for solving bang-bang optimal control problems numerical results and statistical interpretation," *Optimal Control Applications and Methods*, vol. 23, no. 4, pp. 171–197, Aug. 2002.

[25] X. Pan and B. Pan, "Practical homotopy methods for finding the best minimum-fuel transfer in the circular restricted three-body problems," *IEEE Access*, vol. 8, pp. 47 845–47 862, Feb. 2020.

[26] F. Topputo and C. Zhang, "Survey of direct transcription for low-thrust space trajectory optimization with applications," *Abstract and Applied Analysis*, vol. 2014, Jun. 2014.

[27] P. Williams, "Hermite–legendre–gauss–lobatto direct transcription in trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 4, pp. 1392–1395, Aug. 2009.

[28] A. Ronveaux and J. Mawhin, "Rediscovering the contributions of rodrigues on the representation of special functions," *Expositiones Mathematicae*, vol. 23, no. 4, pp. 361–369, Dec. 2005.

[29] C. Hofmann and F. Topputo, "Toward on-board guidance of low-thrust spacecraft in deep space using sequential convex programming," in *Proceedings of* 31^{st} *AAS/AIAA Space Flight Mechanics Meeting Paper AAS 21-350*, Feb. 2021, pp. 1–19.

[30] ——, "Pseudospectral convex low-thrust trajectory optimization in a high-fidelity model," in *Proceedings of AAS/AIAA Astrodynamics Specialist Conference Paper AAS 21-678*, Aug. 2021, pp. 1–19.