C. Hofmann, F. Topputo
Convex Low-Thrust Trajectory Optimization with No-Thrust Constraints and Moving Target
Journal of Spacecraft and Rockets, published online 30/03/2024
doi:10.2514/1.a35877

Convex Low-Thrust Trajectory Optimization With No-Thrust Constraints and Moving Target*

Christian Hofmann† and Francesco Topputo‡
Polytechnic University of Milan, 20156 Milan, Italy

I. Introduction

Numerous new space missions attempt to satisfy the increasing need for exploration and exploitation of space. The space economy is rapidly increasing, and the interest in new technologies and applications has been growing tremendously in the past few years. Especially minor and major bodies in the deep space are important targets for space agencies and other institutions [1, 2]. CubeSats are now a viable low-cost alternative to conventional spacecraft. The recent success of NASA's MarCO CubeSats [3] and the Italian Space Agency's LICIACube [4] has shown that interplanetary CubeSats are becoming reality.

In contrast to conventional chemical propulsion systems that produce high thrust, CubeSats are often equipped with low-thrust engines that result in a small thrust-to-mass ratio. As the state of the spacecraft changes slowly due to the small control actions, transfer times increase because the thruster has to operate over a significantly larger portion of the flight time. Yet, operational constraints often require periods where the thruster has to remain off. Such coast arcs are necessary to perform mission relevant tasks, for example, communication and orbit determination. Therefore, mission-compliant trajectories are sought where thrust and coast arcs alternate. These so-called *duty cycles* are a sequence of thrust and coast arcs; the latter are included to accommodate spacecraft operations. In preliminary mission design, duty cycles are often modeled by taking margins in the maximum thrust magnitude [5, 6]. Even though this approach was successfully applied in several low-thrust missions like Dawn [7], such margins may be overconservative. Moreover, high-fidelity mission design tools like NASA's Evolutionary Mission Trajectory Generator (EMTG) often require an accurate initial guess where duty cycles are explicitly modeled [8]. Given the current trend towards autonomous guidance, navigation, and control [9, 10], including no-thrust periods in the optimization process may become even more important in the future if the reference trajectory is to be recomputed on board. Therefore, a computationally efficient method is desirable for the simultaneous optimization of low-thrust trajectories while incorporating realistic duty cycle constraints.

In general, indirect and direct methods are mainly used to solve the low-thrust trajectory optimization problem [11]. Indirect methods make use of the calculus of variations to derive the first-order necessary conditions, and the resulting two-point boundary value problem is then solved numerically. Direct methods transcribe the infinite-dimensional optimal

*Part of this work was presented as paper AAS 22-750 at the 2022 AAS/AIAA Astrodynamics Specialist Conference, Charlotte, NC, August 7 – 11, 2022.

†Ph.D. Graduate, Department of Aerospace Science and Technology, Via La Masa 34; currently Postdoctoral Associate, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139; hofmannc@mit.edu.

‡Professor, Department of Aerospace Science and Technology, Via La Masa 34; francesco.topputo@polimi.it. AIAA Senior Member.

control problem into a finite-dimensional parameter optimization problem before solving it with some gradient-based algorithm. Two important optimization methods are nonlinear programming (NLP) and sequential convex programming (SCP).

Various low-thrust trajectory optimization problems have been solved with indirect [12, 13] and direct methods [14, 15]. Regardless of the method, discontinuous constraints where the engine parameters change are often a major challenge due to the required first- and second-order derivatives. In particular, indirect methods suffer from a poor convergence domain when discontinuous constraints are included [16, 17]. Often, continuation techniques are used to enhance convergence [18, 19]. In [20], the discontinuous behavior is addressed using interior-point constraints in combination with a homotopic approach. Only few works consider constraints where the thruster has to remain off during some periods. For example, the paper in [21] uses a homotopic approach to solve the nonlinear program and to successively obtain a solution with no-thrust constraints. The work in [22] requires a mesh refinement process to determine trajectories with shutdown constraints in the two-body environment using SCP. In [23], no-thrust periods are included in a model predictive control approach for small-body proximity operations using a zero-order-hold (ZOH) discretization method within SCP. A composite smooth control technique is used in [24] where the switching times are determined to enforce coast periods within an indirect optimization framework. Even though such duty cycles are of utmost importance for real space missions, most of the methods found in literature allow continuous thrust during the whole transfer. The resulting trajectories are therefore not mission-compliant.

At the same time, many works consider fixed state vectors for the final boundary conditions for rendezvous problems [15, 25]. Yet, space missions typically require time-varying final boundary conditions where planetary ephemeris are incorporated. The reason is that the target body (e.g., an asteroid) is in fact moving and following some known trajectory. Therefore, the final time is free, and it is part of the optimization. As a consequence, the final state is not fixed anymore, but becomes a nonlinear function of time. Depending on the application, it may be sufficient to determine the future state of the target body beforehand using ephemeris data and then solve the corresponding fixed-final-state problem. This, however, requires a feasible initial guess for the transfer time. Directly solving the moving target problem is considered more flexible, as the solver is free to choose the optimal arrival time. Such problems with nonlinear dynamic endpoint constraints are often solved with indirect [26] and direct, NLP-based methods [27], and recently also with convex optimization approaches. In [28], simple near-field rendezvous and docking problems are solved using SCP, assuming that the target follows a Keplerian orbit. It has been demonstrated that a successive approximation approach performs well for problems characterized by a short time of flight, less than one revolution, and a linear gravity model. Additionally, SCP is utilized in [29] for solving the orbital transfer problem to a coplanar orbit. Here, nonlinear, time-independent final boundary conditions are linearized, with the application of a second-order correction. In the domain of powered descent guidance, [30] addresses the six-degree-of-freedom problem using SCP. The optimal engine ignition time is free, and intermediate nonlinear equality constraints are imposed. These constraints ensure that the

position and velocity at ignition time adhere to trajectories represented by first- and second-order polynomials. The time-independent nonlinear terminal constraints in [31] are also linearized within the ascent trajectory optimization problem. Furthermore, the work in [32] employs a convex optimization approach to address the stationkeeping problem for periodic and quasi-periodic orbits. The position on the target orbit is a nonlinear function of the states, and it is determined in advance through a semi-analytical approach. Even though SCP has been used to solve problems with (time-varying) nonlinear boundary conditions before, it is yet to be investigated how it performs for high-fidelity low-thrust trajectory optimization with free final time and a moving target. These features are crucial not only for rapid preliminary trajectory design but also for potential onboard applications.

The contribution of this Note is twofold. First, a method is proposed to directly embed no-thrust periods into the first-order-hold discretization method within SCP. Second, time-varying endpoint constraints are considered using planetary ephemeris, and the free-final-time problem is transformed into an equivalent fixed-final-time problem. It is shown that linearizing around a reference arrival time is an easy yet effective method to convexify the target state if ephemeris data is available. A high-fidelity model with n-body dynamics, solar radiation pressure, and a real thruster model is taken into account. The effectiveness of the approach is demonstrated in transfers to the asteroids 2000 SG344 and Dionysus.

The Note is structured as follows. Section II states the optimal control problem and describes the first-order-hold method. No-thrust constraints are addressed in Section III, and the moving target problem is considered in Section IV. Section V presents the numerical simulations, and Section VI concludes this Note.

II. Optimal Control Problem and First-Order-Hold Discretization

Considering the solar radiation pressure $\mathbf{a}_{\text{SRP}}$ and the perturbing acceleration of other bodies $\mathbf{a}_{\text{nbody}}$, the dynamics of a spacecraft orbiting the Sun are of the form

$$\mathbf{f}(\mathbf{x},\mathbf{u}) = \begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ -\mu\,\mathbf{r}/r^3 \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{a}_{\text{nbody}} \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{a}_{\text{SRP}} \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau} \\ -\Gamma/(g_0\, I_{\text{sp}}(r)) \end{bmatrix} \tag{1}$$

$\mathbf{r} \in \mathbb{R}^3$, $\mathbf{v} \in \mathbb{R}^3$, and $z \in \mathbb{R}$ are the position, velocity, and modified mass of the spacecraft, respectively, and $r \coloneqq \|\mathbf{r}\|_2$. μ denotes the gravitational parameter of the primary body. We consider a real thruster model where the maximum thrust $T_{\max}(r)$ and the variable specific impulse $I_{\text{sp}}(r)$ depend on the distance to the Sun r. g_0 is the gravitational acceleration at sea level, and $\Gamma \coloneqq \|\mathbf{T}\|_2 / m$, $\boldsymbol{\tau} \coloneqq \mathbf{T}/m$, and $z \coloneqq \ln m$ are introduced to decouple states and controls, $\mathbf{T}$ being the thrust component vector and m the mass of the spacecraft [33].

A cannonball model with a constant projected area A_{SC} of the spacecraft is used to compute $\mathbf{a}_{\mathrm{SRP}}$ [34]:

$$\mathbf{a}_{\mathrm{SRP}} = \frac{S_{\mathrm{Sun}}\, C_R\, A_{\mathrm{SC}}}{m}\, \frac{\mathbf{r}}{r^3} \tag{2}$$

where

$$S_{\mathrm{Sun}} = \frac{L_{\mathrm{Sun}}}{4\,\pi\, c} \tag{3}$$

$$L_{\mathrm{Sun}} = 4\,\pi\, C_{\mathrm{Sun}}\, \mathrm{AU}^2 \tag{4}$$

C_R, S_{Sun}, L_{Sun}, and C_{Sun} denote the reflectivity coefficient of the spacecraft, the solar pressure constant, the luminosity of the Sun, and the solar constant, respectively. c is the speed of light, and AU the astronomical unit.

The perturbing acceleration of other bodies is calculated using

$$\mathbf{a}_{\mathrm{nbody}} = \sum_{i}^{n} \mu_i \left(\frac{\mathbf{r}_{\mathrm{sat},i}}{r_{\mathrm{sat},i}^3} - \frac{\mathbf{r}_i}{r_i^3} \right) \tag{5}$$

where μ_i denotes the gravitational constant of the ith body, and $\mathbf{r}_{\mathrm{sat},i} = \mathbf{r}_i - \mathbf{r}$, $\mathbf{r}_i$ being the position of the ith body with respect to the Sun. Note that the time dependency is omitted for better readability.

The general convexified fuel-optimal optimization problem with fixed final time t_f is then given by Problem 1.

Problem 1. Find the functions Γ and $\boldsymbol{\tau}$ that solve the following second-order cone program:

$$\underset{\Gamma,\,\boldsymbol{\tau},\,\boldsymbol{\nu},\,\eta}{\text{minimize}} \qquad -z(t_f) + \lambda_\nu\, \|\boldsymbol{\nu}\|_1 + \lambda_\eta\, \max(0,\eta) \tag{6a}$$

$$\text{subject to:} \qquad \dot{\mathbf{x}} = \mathbf{A}(\bar{\mathbf{x}})\,\mathbf{x} + \mathbf{B}(\bar{\mathbf{x}})\,\mathbf{u} + \mathbf{q}\,(\bar{\mathbf{x}},\bar{\mathbf{u}}) + \boldsymbol{\nu} \tag{6b}$$

$$\Gamma \leq T_{\max}(\bar{r})\, \mathrm{e}^{-\bar{z}}\, [1 - z + \bar{z}] + \eta \tag{6c}$$

$$\|\boldsymbol{\tau}\|_2 \leq \Gamma \tag{6d}$$

$$\|\mathbf{x} - \bar{\mathbf{x}}\|_1 \leq R \tag{6e}$$

$$\mathbf{r}(t_0) = \mathbf{r}_0,\ \mathbf{v}(t_0) = \mathbf{v}_0,\ z(t_0) = z_0 \tag{6f}$$

$$\mathbf{r}(t_f) = \mathbf{r}_f,\ \mathbf{v}(t_f) = \mathbf{v}_f \tag{6g}$$

$$\mathbf{x}_{\mathrm{lb}} \leq \mathbf{x}(t) \leq \mathbf{x}_{\mathrm{ub}},\ \mathbf{u}_{\mathrm{lb}} \leq \mathbf{u}(t) \leq \mathbf{u}_{\mathrm{ub}} \tag{6h}$$

where

$$\mathbf{A}(\bar{\mathbf{x}}) \coloneqq \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\bigg|_{\bar{\mathbf{x}},\bar{\mathbf{u}}},\ \mathbf{B}(\bar{\mathbf{x}}) \coloneqq \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\bigg|_{\bar{\mathbf{x}},\bar{\mathbf{u}}},\ \mathbf{q}(\bar{\mathbf{x}},\bar{\mathbf{u}}) \coloneqq \mathbf{f}(\bar{\mathbf{x}},\bar{\mathbf{u}}) - \mathbf{A}(\bar{\mathbf{x}})\,\bar{\mathbf{x}} - \mathbf{B}(\bar{\mathbf{x}})\,\bar{\mathbf{u}} \tag{7}$$

Slack variables $\boldsymbol{\nu}$ and η are added in Eqs. (6b) and (6c) to avoid *artificial infeasibility*, and penalized in the objective

function with factors λ_ν and λ_η. Note that we used the approximation $I_{\text{sp}}(r) \approx I_{\text{sp}}(\bar{r})$ ($\bar{r}$ being the radius at the previous iteration) before linearizing the dynamics. Consequently, the linearzed dynamics in Eq. (6b) do not depend on the reference control history $\bar{\mathbf{u}}$. Our previous works indicate that this approach frequently offers benefits in terms of convergence [35]. Moreover, the maximum available thrust $T_{\max}(r)$ on the right-hand side in Eq. (6c) was not linearized about some reference, but approximated by $T_{\max}(r) \approx T_{\max}(\bar{r})$. It was found that employing this method often proves beneficial in achieving convergence [36]. The constraint on the upper bound of the control magnitude was relaxed in Eq. (6d), and the trust-region constraint in Eq. (6e) with radius R is added to restrict the search space to the neighborhood of the reference solution. Initial and final boundary constraints are given by Eqs. (6f) and (6g), respectively. Upper (subscript ub) and lower (subscript lb) bounds on states and controls are imposed in Eq. (6h). A more detailed derivation of Problem 1 can be found in [15].

Remark 1. The solution to the relaxed problem with $\|\boldsymbol{\tau}\|_2 \leq \Gamma$ is also a solution to the original problem if the inequality constraint in Eq. (6d) is active, i.e., $\|\boldsymbol{\tau}\|_2 = \Gamma$. Several works have proved that this condition is satisfied almost everywhere on $[t_0, t_f]$, and, consequently, the solutions of the original and relaxed problems are identical [37, 38]. When no-thrust constraints are added in Section III, the problem formulation slightly changes because interior-point constraints are implicitly included that force the controls to be zero during certain periods. As such constraints solely depend on time and not on any state variables, no discontinuities are induced in the costates [39]. Therefore, we make use of results from the literature where it is shown that the relaxed problem is also an optimal solution to the original problem when additional constraints on the controls are present [40, 41]. This is confirmed in our simulations where we numerically checked that $\|\boldsymbol{\tau}\|_2 = \Gamma$ holds at the solution.

Using the standard first-order-hold discretization method where the controls are interpolated linearly, the discretized dynamics are given by

$$\mathbf{x}_{k+1} = \mathbf{A}_k \, \mathbf{x}_k + \mathbf{B}_k^- \, \mathbf{u}_k + \mathbf{B}_k^+ \, \mathbf{u}_{k+1} + \mathbf{q}_k + \boldsymbol{\nu}_k \tag{8}$$

where $(\cdot)_k \coloneqq (\cdot)(t_k)$, $k = 1, \ldots, N-1$, N being the number of discretization points. Moreover,

$$\mathbf{A}_k = \boldsymbol{\Phi}(t_{k+1}, t_k) \tag{9a}$$

$$\mathbf{B}_k^- = \mathbf{A}_k \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}^{-1}(t, t_k) \, \mathbf{B}(t) \, \lambda_-(t) \, \mathrm{d}t \tag{9b}$$

$$\mathbf{B}_k^+ = \mathbf{A}_k \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}^{-1}(t, t_k) \, \mathbf{B}(t) \, \lambda_+(t) \, \mathrm{d}t \tag{9c}$$

$$\mathbf{q}_k = \mathbf{A}_k \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}^{-1}(t, t_k) \, \mathbf{q}(t) \, \mathrm{d}t \tag{9d}$$

with the state transition matrix $\mathbf{\Phi}$ and

$$\lambda_-(t) \coloneqq \frac{t_{k+1} - t}{t_{k+1} - t_k}, \qquad \lambda_+(t) \coloneqq \frac{t - t_k}{t_{k+1} - t_k}, \qquad t \in [t_k, t_{k+1}] \tag{10}$$

III. No-Thrust Constraints

Although operational constraints are important for real space missions, they are rarely included in the optimization process. The obtained thrust profiles in the literature often require continuous thrust for several weeks or even months. In reality, this is not feasible, for example due to hardware or mission constraints. Duty cycles are often imposed where thrusting and coasting periods alternate, and it is of paramount importance that mission-compliant trajectories are computed. An example is the onboard orbit determination process in the context of autonomous navigation where a navigation cycle is defined to determine the state of the spacecraft [42]. In this Note, we focus on operational constraints where the thruster has to remain off for certain periods. We extend the approach in [23] where the ZOH discretization, linear dynamics, constant specific impulse and maximum thrust, and a single and fixed thrusting period per segment are considered for small-body proximity operations.

Given a trajectory segment $[t_k, t_{k+1}]$, we define n required no-thrust periods in this segment as

$$[t_i, t_i + \Delta t_{\text{off},i}], \quad \forall i = 1, \dots, n \tag{11}$$

where $t_i > t_k$ and $t_i + \Delta t_{\text{off},i} < t_{k+1}$ without loss of generality, i.e., the no-thrust periods $\Delta t_{\text{off},i}$ lie strictly within a trajectory segment. This assumption is reasonable because the location of the nodes (and thus, endpoints of a segment) can be chosen arbitrarily. Moreover, Δt_{off} is in general much smaller than one segment. If $i > 1$, it is assumed that the no-thrust periods are not overlapping, i.e., $t_{i+1} > t_i + \Delta t_{\text{off},i}$. Therefore,

$$\mathbf{u}(t) = \mathbf{0}, \qquad \forall t \in \left[t_i, t_i + \Delta t_{\text{off},i}\right] \tag{12}$$

and

$$\mathbf{u}(t) = \tilde{\lambda}_-(t)\, \mathbf{u}_k + \tilde{\lambda}_+(t)\, \mathbf{u}_{k+1}, \qquad \forall t \notin \left[t_i, t_i + \Delta t_{\text{off},i}\right] \tag{13}$$

where $\tilde{\lambda}_-(t)$ and $\tilde{\lambda}_+(t)$ are the factors that define the control interpolation. Note that these must be chosen differently compared to the standard formulation in Eq. (10). According to Pontryagin's minimum principle, a fuel-optimal solution yields a bang-bang control structure where the magnitude $u = T_{\max}$ or $u = 0$ [13]. As we defined the accelerations as our control variables, this piecewise constant relationship does not hold anymore; instead, the magnitude of the acceleration Γ increases over time due to the decreasing mass when thrusting. Simply interpolating the accelerations linearly would

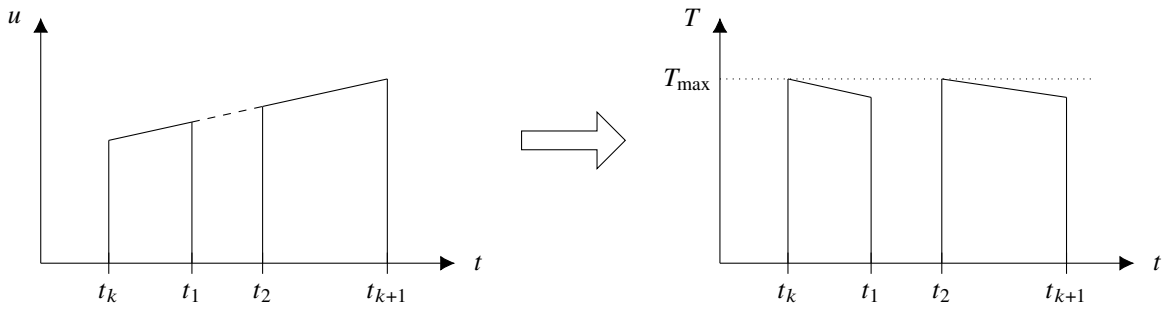


Fig. 1 Linear interpolation of accelerations when no-thrust periods are considered and corresponding thrust magnitude.

therefore yield a thrust magnitude curve that is not piecewise constant if no-thrust constraints are considered (see Fig. 1 where one no-thrust period $[t_1, t_2]$ in a segment $[t_k, t_{k+1}]$ is shown, and $u \equiv \Gamma$). Two different interpolation methods and modifications to the discretized dynamical equations are presented in the following, depending on whether the case considers constant or variable maximum thrust and specific impulse.

Regardless of the interpolation method and assuming one no-thrust period $[t_1, t_2]$ in a segment $[t_k, t_{k+1}]$, the state at t_1 can be computed as:

$$\begin{aligned}\mathbf{x}(t_1) = \mathbf{\Phi}(t_1, t_k)\,\mathbf{x}(t_k) &+ \mathbf{\Phi}(t_1, t_k) \int_{t_k}^{t_1} \mathbf{\Phi}^{-1}(\xi, t_k)\,\tilde{\lambda}_-(\xi)\,\mathbf{B}(\xi)\,\mathrm{d}\xi\,\mathbf{u}(t_k) \\ &+ \mathbf{\Phi}(t_1, t_k) \int_{t_k}^{t_1} \mathbf{\Phi}^{-1}(\xi, t_k)\,\tilde{\lambda}_+(\xi)\,\mathbf{B}(\xi)\,\mathrm{d}\xi\,\mathbf{u}(t_1) + \mathbf{\Phi}(t_1, t_k) \int_{t_k}^{t_1} \mathbf{\Phi}^{-1}(\xi, t_k)\,\mathbf{q}(\xi)\,\mathrm{d}\xi\end{aligned} \tag{14}$$

Next, the final state $\mathbf{x}(t_2)$ of the coasting period is calculated:

$$\mathbf{x}(t_2) = \mathbf{\Phi}(t_2, t_1)\,\mathbf{x}(t_1) + \mathbf{\Phi}(t_2, t_1) \int_{t_1}^{t_2} \mathbf{\Phi}^{-1}(\xi, t_1)\,\mathbf{q}(\xi)\,\mathrm{d}\xi \tag{15}$$

Note that there is no control term. The final step is to calculate the state at the end of the segment:

$$\begin{aligned}\mathbf{x}(t_{k+1}) = \mathbf{\Phi}(t_{k+1}, t_2)\,\mathbf{x}(t_2) &+ \mathbf{\Phi}(t_{k+1}, t_2) \int_{t_2}^{t_{k+1}} \mathbf{\Phi}^{-1}(\xi, t_2)\,\tilde{\lambda}_-(\xi)\,\mathbf{B}(\xi)\,\mathrm{d}\xi\,\mathbf{u}(t_2) \\ &+ \mathbf{\Phi}(t_{k+1}, t_2) \int_{t_2}^{t_{k+1}} \mathbf{\Phi}^{-1}(\xi, t_2)\,\tilde{\lambda}_+(\xi)\,\mathbf{B}(\xi)\,\mathrm{d}\xi\,\mathbf{u}(t_{k+1}) \\ &+ \mathbf{\Phi}(t_{k+1}, t_2) \int_{t_2}^{t_{k+1}} \mathbf{\Phi}^{-1}(\xi, t_k)\,\mathbf{q}(\xi)\,\mathrm{d}\xi\end{aligned} \tag{16}$$

Substituting Eq. (14) into Eq. (15), and Eq. (15) into Eq. (16) yields

$$\begin{aligned}\mathbf{x}(t_{k+1}) = \mathbf{A}(t_2)\,\mathbf{A}(t_1)\,\mathbf{A}(t_k)\,\mathbf{x}(t_k) &+ \mathbf{A}(t_2)\,\mathbf{A}(t_1)\,\mathbf{B}^-(t_k)\,\mathbf{u}(t_k) + \mathbf{A}(t_2)\,\mathbf{A}(t_1)\,\mathbf{B}^+(t_k)\,\mathbf{u}(t_1) \\ &+ \mathbf{A}(t_2)\,\mathbf{A}(t_1)\,\mathbf{q}(t_k) + \mathbf{A}(t_2)\,\mathbf{q}(t_1) + \mathbf{B}^-(t_2)\,\mathbf{u}(t_1) + \mathbf{B}^+(t_2)\,\mathbf{u}(t_2) + \mathbf{q}(t_2)\end{aligned} \tag{17}$$

where

$$\mathbf{A}(t_{k+1}, t_2) = \mathbf{\Phi}(t_{k+1}, t_2),\ \mathbf{A}(t_2, t_1) = \mathbf{\Phi}(t_2, t_1),\ \mathbf{A}(t_2, t_k) = \mathbf{\Phi}(t_2, t_k) \tag{18a}$$

$$\mathbf{B}^-(t_{k+1}, t_2) = \mathbf{\Phi}(t_{k+1}, t_2) \int_{t_2}^{t_{k+1}} \mathbf{\Phi}^{-1}(\xi, t_k)\, \mathbf{B}(\xi)\, \tilde{\lambda}_-(\xi)\, \mathrm{d}\xi \tag{18b}$$

$$\mathbf{B}^+(t_{k+1}, t_2) = \mathbf{\Phi}(t_{k+1}, t_2) \int_{t_2}^{t_{k+1}} \mathbf{\Phi}^{-1}(\xi, t_k)\, \mathbf{B}(\xi)\, \tilde{\lambda}_+(\xi)\, \mathrm{d}\xi \tag{18c}$$

$$\mathbf{B}^-(t_1, t_k) = \mathbf{\Phi}(t_1, t_k) \int_{t_k}^{t_1} \mathbf{\Phi}^{-1}(\xi, t_k)\, \mathbf{B}(\xi)\, \tilde{\lambda}_-(\xi)\, \mathrm{d}\xi \tag{18d}$$

$$\mathbf{B}^+(t_1, t_1) = \mathbf{\Phi}(t_1, t_k) \int_{t_k}^{t_1} \mathbf{\Phi}^{-1}(\xi, t_k)\, \mathbf{B}(\xi)\, \tilde{\lambda}_+(\xi)\, \mathrm{d}\xi \tag{18e}$$

$$\mathbf{q}(t_{k+1}, t_2) = \mathbf{\Phi}(t_{k+1}, t_2) \int_{t_2}^{t_{k+1}} \mathbf{\Phi}^{-1}(\xi, t_k)\, \mathbf{q}(\xi)\, \mathrm{d}\xi \tag{18f}$$

$$\mathbf{q}(t_2, t_1) = \mathbf{\Phi}(t_2, t_1) \int_{t_1}^{t_2} \mathbf{\Phi}^{-1}(\xi, t_1)\, \mathbf{q}(\xi)\, \mathrm{d}\xi \tag{18g}$$

$$\mathbf{q}(t_1, t_k) = \mathbf{\Phi}(t_1, t_k) \int_{t_k}^{t_1} \mathbf{\Phi}^{-1}(\xi, t_k)\, \mathbf{q}(\xi)\, \mathrm{d}\xi \tag{18h}$$

$\mathbf{B}(\xi)$ and $\mathbf{q}(\xi)$ are the Jacobian matrix and constant part of the linearization, respectively, as defined in Eq. (7) in Section II. It is straightforward to extend this approach to more no-thrust periods.

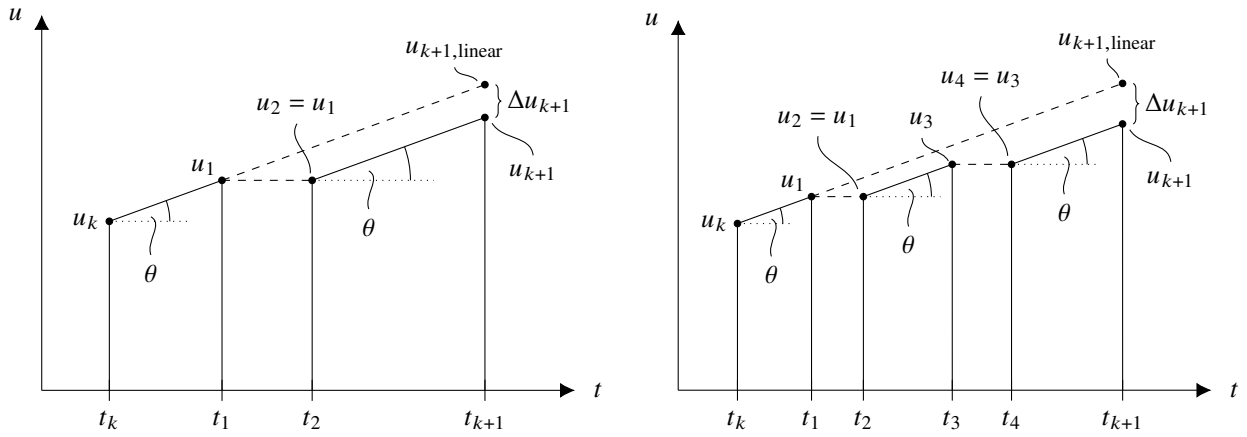
The only unknowns are the interpolated controls $\mathbf{u}(t_1)$ and $\mathbf{u}(t_2)$. Two methods are proposed to show how they can be determined.

A. Constant Maximum Thrust and Specific Impulse

When $T_{\max}$ is constant, we have $T(t) = T_{\max}$ for thrusting periods, $T(t) \coloneqq \|\mathbf{T}(t)\|_2$. Therefore, $\dot{m}(t) = T_{\max}/(g_0\, I_{\mathrm{sp}})$ is also constant $\forall t \notin [t_i, t_i + \Delta t_{\mathrm{off},i}]$ because $I_{\mathrm{sp}} = \text{const.}$, and the mass decreases linearly. As $\Gamma(t) = T(t)/m(t)$, we thus propose a piecewise linear interpolation where the slope of $\Gamma(t)$ (and hence the angle θ in Fig. 2) is constant. This way, the resulting thrust magnitude is piecewise constant in that segment as expected. Figure 2a illustrates the interpolated thrust acceleration in one segment with one no-thrust period. Given the optimization variables $\mathbf{u}_k$ and $\mathbf{u}_{k+1}$ and times $\Delta t_1 = t_1 - t_k$ and $\Delta t_2 = t_{k+1} - t_2$, and using

$$\tan\theta = \frac{\Delta \mathbf{u}_1}{\Delta t_1} = \frac{\mathbf{u}_1 - \mathbf{u}_k}{\Delta t_1} \tag{19}$$

$$\tan\theta = \frac{\Delta \mathbf{u}_2}{\Delta t_2} = \frac{\mathbf{u}_{k+1} - \mathbf{u}_1}{\Delta t_2} \tag{20}$$



(a) One no-thrust period in one segment. **(b) Two no-thrust periods in one segment.**

Fig. 2 Modified linear interpolation when no-thrust periods are considered with constant $T_{\max}$ and I_{sp}.

the interpolated controls $\mathbf{u}_1 = \mathbf{u}(t_1) = \mathbf{u}_2 = \mathbf{u}(t_2)$ can be computed as follows:

$$\mathbf{u}_1 = \mathbf{u}_2 = \underbrace{\frac{\Delta t_2}{\Delta t_1 + \Delta t_2}}_{\lambda_-(t)} \mathbf{u}_k + \underbrace{\frac{\Delta t_1}{\Delta t_1 + \Delta t_2}}_{\lambda_+(t)} \mathbf{u}_{k+1} \tag{21}$$

It is straightforward to determine the interpolated controls for $n > 1$ off periods. For example, for $n = 2$ (see Fig. 2b) we obtain

$$\mathbf{u}_1 = \mathbf{u}_2 = \frac{\Delta t_2 + \Delta t_3}{\Delta t_1 + \Delta t_2 + \Delta t_3} \mathbf{u}_k + \frac{\Delta t_1}{\Delta t_1 + \Delta t_2 + \Delta t_3} \mathbf{u}_{k+1} \tag{22}$$

$$\mathbf{u}_3 = \mathbf{u}_4 = \frac{\Delta t_3}{\Delta t_1 + \Delta t_2 + \Delta t_3} \mathbf{u}_k + \frac{\Delta t_1 + \Delta t_2}{\Delta t_1 + \Delta t_2 + \Delta t_3} \mathbf{u}_{k+1} \tag{23}$$

B. Variable Maximum Thrust and Specific Impulse

The variable maximum thrust and specific impulse case requires a different approach as $\dot{m}$ cannot be considered constant anymore. Instead of linearly interpolating the accelerations, we approximate the thrust vector $\tilde{\mathbf{T}}(t) \coloneqq [\mathbf{T}(t)^\top, T(t)]^\top$ with affine functions:

$$\tilde{\mathbf{T}}(t) = \underbrace{\frac{t_{k+1} - t}{t_{k+1} - t_k}}_{=:\lambda_-(t)} \tilde{\mathbf{T}}_k + \underbrace{\frac{t - t_k}{t_{k+1} - t_k}}_{=:\lambda_+(t)} \tilde{\mathbf{T}}_{k+1} = \lambda_-(t)\, \tilde{\mathbf{T}}_k + \lambda_+(t)\, \tilde{\mathbf{T}}_{k+1}, \qquad t \in [t_k, t_{k+1}] \tag{24}$$

Substituting $\tilde{\mathbf{T}}(t) = \mathbf{u}(t)\, m(t)$ into Eq. (24) and solving for $\mathbf{u}(t)$ yields

$$\begin{aligned}\mathbf{u}(t) &= \frac{\lambda_-(t)\, m(t_k)}{m(t)}\mathbf{u}_k + \frac{\lambda_+(t)\, m(t_{k+1})}{m(t)}\,\mathbf{u}_{k+1} \\ &\approx \frac{\lambda_-(t)\, \bar{m}(t_k)}{\bar{m}(t)}\,\mathbf{u}_k + \frac{\lambda_+(t)\, \bar{m}(t_{k+1})}{\bar{m}(t)}\,\mathbf{u}_{k+1}\end{aligned} \tag{25}$$

where the mass is approximated using the value of the reference trajectory. Evaluating Eq. (25) at t_i ($i = 1, 2$ for one no-thrust period) allows us to determine the interpolated controls with respect to the accelerations. The only unknown is $\bar{m}(t_1) = \bar{m}(t_2)$ which can be calculated by integrating the differential equation of $\dot{m}$:

$$\bar{m}(t_1) = \bar{m}(t_k) + \int_{t_k}^{t_1} \dot{\bar{m}}\,\mathrm{d}t = \bar{m}(t_k) - \frac{1}{g_0}\int_{t_k}^{t_1} \frac{\bar{T}(t)}{\bar{I}_{\mathrm{sp}}(t)}\,\mathrm{d}t \tag{26}$$

As in Eq. (24), $\bar{T}(t)$ and $\bar{I}_{\mathrm{sp}}(t)$ are assumed to be piecewise affine functions. Recalling that $m = \mathrm{e}^z$, $\bar{m}(t_1)$ can be computed analytically. It follows that

$$\mathbf{u}(t_1) = \underbrace{\frac{\lambda_-(t_1)\, \bar{m}(t_k)}{\bar{m}(t_1)}}_{=:\,\tilde{\lambda}_-(t_1)}\,\mathbf{u}_k + \underbrace{\frac{\lambda_+(t_1)\, \bar{m}(t_{k+1})}{\bar{m}(t_1)}}_{=:\,\tilde{\lambda}_+(t_1)}\,\mathbf{u}_{k+1} = \tilde{\lambda}_-(t_1)\,\mathbf{u}_k + \tilde{\lambda}_+(t_1)\,\mathbf{u}_{k+1} \tag{27}$$

$$\mathbf{u}(t_2) = \underbrace{\frac{\lambda_-(t_2)\, \bar{m}(t_k)}{\bar{m}(t_2)}}_{=:\,\tilde{\lambda}_-(t_2)}\,\mathbf{u}_k + \underbrace{\frac{\lambda_+(t_2)\, \bar{m}(t_{k+1})}{\bar{m}(t_2)}}_{=:\,\tilde{\lambda}_+(t_2)}\,\mathbf{u}_{k+1} = \tilde{\lambda}_-(t_2)\,\mathbf{u}_k + \tilde{\lambda}_+(t_2)\,\mathbf{u}_{k+1} \tag{28}$$

Our simulations suggest that this successive approximation approach yields a good compromise in terms of convergence, accuracy, and computational effort.

IV. Moving Target

In this section, the convex low-thrust trajectory optimization approach is extended to include planetary ephemeris for dynamic endpoint constraints. This increases the flexibility and capabilities of the method because many (preliminary) studies require the computation of fuel-optimal trajectories for targets that follow some trajectory (see for example the asteroid selection process for ESA's Miniaturised Asteroid Remote Geophysical Observer in [27]).

The position $\mathbf{r}_t(t)$ and velocity $\mathbf{v}_t(t)$ of the target become functions of time, and we relax the final boundary conditions to obtain

$$|\mathbf{r}(t_f) - \mathbf{r}_t(t_f)| \leq \Delta\mathbf{r} \tag{29a}$$

$$|\mathbf{v}(t_f) - \mathbf{v}_t(t_f)| \leq \Delta\mathbf{v} \tag{29b}$$

where $\Delta\mathbf{r}$ and $\Delta\mathbf{v}$ are given and define the desired accuracy. If $\Delta\mathbf{r} = \Delta\mathbf{v} = \mathbf{0}$, the spacecraft must precisely reach the target state within the specified tolerances of the solver. Even though the ephemerides of the target body are assumed to be known at all times, determining $\mathbf{r}_t(t)$ and $\mathbf{v}_t(t)$ (e.g., retrieving the data from a lookup table or integrating the dynamics of the target body) is in general a nonlinear and nonconvex operation. As a consequence, the boundary conditions in Eq. (29) become nonconvex. Therefore, we approximate $\mathbf{r}_t(t)$ and $\mathbf{v}_t(t)$ using a first-order Taylor series about a reference final time $\bar{t}_f$ [43]:

$$\mathbf{r}_t(t_f) \approx \mathbf{r}_t(\bar{t}_f) + \left.\frac{\mathrm{d}\mathbf{r}_t(t)}{\mathrm{d}t}\right|_{\bar{t}_f} \left(t_f - \bar{t}_f\right) \tag{30a}$$

$$\mathbf{v}_t(t_f) \approx \mathbf{v}_t(\bar{t}_f) + \left.\frac{\mathrm{d}\mathbf{v}_t(t)}{\mathrm{d}t}\right|_{\bar{t}_f} \left(t_f - \bar{t}_f\right) \tag{30b}$$

As $\mathrm{d}\mathbf{r}_t(t)/\mathrm{d}t = \mathbf{v}_t(t)$ and $\mathrm{d}\mathbf{v}_t(t)/\mathrm{d}t = \mathbf{a}_t(t)$, Eqs. (30a) and (30b) can be rewritten as

$$\mathbf{r}_t(t_f) \approx \mathbf{r}_t(\bar{t}_f) + \mathbf{v}_t(\bar{t}_f) \left(t_f - \bar{t}_f\right) \tag{31a}$$

$$\mathbf{v}_t(t_f) \approx \mathbf{v}_t(\bar{t}_f) + \mathbf{a}_t(\bar{t}_f) \left(t_f - \bar{t}_f\right) \tag{31b}$$

Considering a n-body model, the acceleration $\mathbf{a}_t(\bar{t}_f)$ of the target body can be calculated using

$$\mathbf{a}_t(\bar{t}_f) = -\frac{\mu\,\mathbf{r}_t(\bar{t}_f)}{r_t(\bar{t}_f)^3} + \sum_{i=1}^{n} \mu_i \left(\frac{\mathbf{r}_{\mathrm{t},i}(\bar{t}_f)}{r_{\mathrm{t},i}^3(\bar{t}_f)} - \frac{\mathbf{r}_i(\bar{t}_f)}{r_i^3(\bar{t}_f)} \right) \tag{32}$$

where $\mathbf{r}_{\mathrm{t},i}(\bar{t}_f) = \mathbf{r}_i(\bar{t}_f) - \mathbf{r}_{\mathrm{t}}(\bar{t}_f)$. The relaxed final boundary conditions are thus

$$\left|\mathbf{r}(t_f) - \mathbf{r}_t(\bar{t}_f) - \mathbf{v}_t(\bar{t}_f) \left(t_f - \bar{t}_f\right)\right| \leq \Delta\mathbf{r} + \boldsymbol{\zeta}_r \tag{33a}$$

$$\left|\mathbf{v}(t_f) - \mathbf{v}_t(\bar{t}_f) - \mathbf{a}_t(\bar{t}_f) \left(t_f - \bar{t}_f\right)\right| \leq \Delta\mathbf{v} + \boldsymbol{\zeta}_v \tag{33b}$$

Note that slack variables $\boldsymbol{\zeta}_r$ and $\boldsymbol{\zeta}_v$ are included to avoid artificial infeasibility due to the linearized constraints. These are again penalized by adding

$$\lambda_\zeta \sum_{i=1}^{m} \max(0, \zeta_i) \tag{34}$$

to the cost function, where ζ_i denotes the ith component of $\boldsymbol{\zeta} \coloneqq [\boldsymbol{\zeta}_r^\top, \boldsymbol{\zeta}_v^\top]^\top \in \mathbb{R}^m$.

In case of a moving target, the final time t_f cannot be considered fixed anymore. Therefore, the problem becomes a free-final-time problem with t_f being an optimization parameter. The normalized time $\xi \in [0, 1]$ is introduced to obtain an equivalent fixed-final-time problem [44]. It is defined such that

$$0 = \xi_1 < \xi_2 < \cdots < \xi_N = 1 \tag{35}$$

Using $t_f = \mathrm{d}t/\mathrm{d}\xi$, the nonlinear dynamics become

$$\mathbf{F}(t_f, \mathbf{x}(\xi), \mathbf{u}(\xi)) \coloneqq \mathbf{x}'(\xi) \coloneqq \frac{\mathrm{d}}{\mathrm{d}\xi}\mathbf{x}(\xi) = \frac{\mathrm{d}t}{\mathrm{d}\xi}\,\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(\xi) = t_f\,\dot{\mathbf{x}}(\xi) \tag{36}$$

As a consequence, t_f becomes an additional optimization variable, and the discretized optimal control problem is now formulated in the ξ domain. Therefore, the linearized dynamics change to

$$\mathbf{x}'(\xi) = t_f\,\mathbf{f}(\mathbf{x}(\xi), \mathbf{u}(\xi)) \tag{37}$$

and we obtain

$$\mathbf{x}'(\xi) = \mathbf{A}(\xi)\,\mathbf{x}(\xi) + \mathbf{B}\,\lambda_-(\xi)\,\mathbf{u}_k + \mathbf{B}\,\lambda_+(\xi)\,\mathbf{u}_{k+1} + \mathbf{S}(\xi)\,t_f + \mathbf{q}(\xi) \tag{38}$$

where the Jacobian matrices are now determined with respect to the modified dynamics $\mathbf{x}'(\xi)$. The discretized dynamics then read

$$\mathbf{x}_{k+1} = \mathbf{A}_k\,\mathbf{x}_k + \mathbf{B}_k^-\,\mathbf{u}_k + \mathbf{B}_k^+\,\mathbf{u}_{k+1} + \mathbf{S}_k\,t_f + \mathbf{q}_k + \boldsymbol{\nu}_k \tag{39}$$

The matrices and vectors $\mathbf{A}_k$, $\mathbf{B}_k^-$, $\mathbf{B}_k^+$, and $\mathbf{q}_k$ are similar to Eq. (9), but with respect to $\mathbf{F}$ and in the ξ domain. $\mathbf{S}_k$ is given by

$$\mathbf{S}_k = \mathbf{A}_k \int_{\xi_k}^{\xi_{k+1}} \boldsymbol{\Phi}^{-1}(\xi, \xi_k)\,\mathbf{S}(\xi)\,\mathrm{d}\xi \tag{40}$$

where $\mathbf{S} \coloneqq \partial\mathbf{F}(t_f, \mathbf{x}(\xi), \mathbf{u}(\xi))/\partial t_f$. The reference state at $\xi \in [\xi_k, \xi_{k+1}]$ and the nonlinear constraint violations are computed in the same way as in the formulation with a fixed final time.

Given N discretization points, the resulting convex optimization problem is then given by Problem 2.

Problem 2. Find the vectors $\mathbf{x}$, $\mathbf{u}$, $\boldsymbol{\nu}$, $\boldsymbol{\eta}$, $\boldsymbol{\zeta}$, and t_f that solve the following second-order cone program:

$$\min_{t_f, \mathbf{x}, \mathbf{u}, \boldsymbol{\nu}, \boldsymbol{\eta}, \boldsymbol{\zeta}} \quad -z_N + \lambda_\nu \sum_{i=1}^{N-1} \|\boldsymbol{\nu}_i\|_1 + \lambda_\eta \sum_{i=1}^{N} \max(0, \eta_i) + \lambda_\zeta \sum_{i=1}^{m} \max(0, \zeta_i) \tag{41a}$$

$$\text{subject to:} \quad \mathbf{x}_{k+1} = \mathbf{A}_k\,\mathbf{x}_k + \mathbf{B}_k^-\,\mathbf{u}_k + \mathbf{B}_k^+\,\mathbf{u}_{k+1} + \mathbf{S}_k\,t_f + \mathbf{q}_k + \boldsymbol{\nu}_k, \quad k = 1, \ldots, N-1 \tag{41b}$$

$$\Gamma_k \leq T_{\max}(\bar{r}_k)\,\mathrm{e}^{-\bar{z}_k}\,(1 - z_k + \bar{z}_k) + \eta_k, \quad k = 1, \ldots, N \tag{41c}$$

$$\|\boldsymbol{\tau}_k\|_2 \leq \Gamma_k, \quad k = 1, \ldots, N \tag{41d}$$

$$\|\mathbf{y} - \bar{\mathbf{y}}\|_1 \leq R \tag{41e}$$

$$t_{f,\mathrm{lb}} \leq t_f \leq t_{f,\mathrm{ub}} \tag{41f}$$

$$\mathbf{r}_1 = \mathbf{r}_0,\ \mathbf{v}_1 = \mathbf{v}_0,\ z_1 = z_0 \tag{41g}$$

$$\left\| \begin{bmatrix} \mathbf{r}_N \\ \mathbf{v}_N \end{bmatrix} - \begin{bmatrix} \mathbf{r}_t(\bar{t}_f) + \mathbf{v}_t(\bar{t}_f)\ \left(t_f - \bar{t}_f\right) \\ \mathbf{v}_t(\bar{t}_f) + \mathbf{a}_t(\bar{t}_f)\ \left(t_f - \bar{t}_f\right) \end{bmatrix} \right\| \leq \begin{bmatrix} \Delta\mathbf{r} \\ \Delta\mathbf{v} \end{bmatrix} + \boldsymbol{\zeta} \tag{41h}$$

where $\mathbf{y} \coloneqq [\mathbf{x}^\top, t_f]^\top$ in Eq. (41e), i.e., we include states and the final time in the trust-region constraint. Our simulations suggest that neglecting the controls does not affect the results, but reduces the problem size. The constraint in Eq. (41f) refers to the lower and upper bound of t_f, respectively.

V. Numerical Simulations

The effectiveness of the proposed methods is demonstrated in several numerical simulations. In particular, we compute fuel-optimal trajectories with no-thrust constraints from the Sun-Earth Lagrange point L_2 (SEL_2) to the asteroids 2000 SG344 and Dionysus. 2000 SG344 is a potential target of ESA's Miniaturised Asteroid Remote Geophysical Observer (M-ARGO) mission [27], and the transfer to Dionysus is often considered in literature because it requires several revolutions and large changes in semi-major axis, eccentricity, and inclination [24, 45]. Moreover, we briefly compare the performance of the moving target approach with the state-of-the-art optimal control software GPOPS-II [46] in combination with the Interior Point Optimizer (IPOPT) [47]. The simulations are performed in MATLAB on an Intel Core i7-8565 1.80 GHz Laptop with four cores and 16 GB of RAM. A *mex* function is used to numerically integrate Eqs. (9b)–(9d), and we use the open-source Embedded Conic Solver (ECOS) [48] to solve the second-order cone program in Problem 1. The initial guesses are generated with a shape-based method where the initial and target states are connected by cubic polynomials [49].

The transfer data is given in Table 1, and tolerances and parameters relevant for the SCP algorithm in Table 2. Details about the algorithm and the trust-region parameters ρ_0, ρ_1, ρ_2, and α and β can be found in [50, 51]. We use 150 (2000 SG344) and 250 (Dionysus) discretization points, respectively. GPOPS-II employs 5 collocation points per segment as this was found to be favorable for accuracy and computational efficiency. Note that the feasibility and optimality threshold are the same for both algorithms. Table 3 provides the values of the physical constants.

The thruster model for the 2000 SG344 transfer is given by [27]:

$$T_{\max}(P_{\text{in}}) = a_0 + a_1\, P_{\text{in}} + a_2\, P_{\text{in}}^2 + a_3\, P_{\text{in}}^3 + a_4\, P_{\text{in}}^4 \tag{42a}$$

$$I_{\text{sp}}(P_{\text{in}}) = b_0 + b_1\, P_{\text{in}} + b_2\, P_{\text{in}}^2 + b_3\, P_{\text{in}}^3 + b_4\, P_{\text{in}}^4 \tag{42b}$$

$$P_{\text{in}}(r) = c_0 + c_1\, r + c_2\, r^2 + c_3\, r^3 + c_4 r^4 \tag{42c}$$

Table 1 Simulation values for the transfers from SEL_2 to the asteroids 2000 SG344 and Dionysus [27, 45].

Parameter	SEL_2 - 2000 SG344	SEL_2 - Dionysus
Initial epoch	04-Feb-2024 12:00:00 UTC	23-Dec-2012 00:00:00 UTC
Initial mass m_0, kg	22.6	4000
Final mass $m(t_f)$, kg	free	free
Min. input power $P_{\text{in,min}}$, W	90	62.5
Max. input power $P_{\text{in,max}}$, W	120	1000
Max. thrust $T_{\max}$, N	2.2519×10^{-3}	0.5
Max. specific impulse $I_{\text{sp,max}}$, s	3067	3000
Spacecraft area A_{SC}, m^2	0.05	100
Reflectivity coefficient C_R	1.3	1.3

Table 2 Parameters of the algorithm.

Parameter	Value
Feasibility tolerance	10^{-6}
Optimality tolerance	10^{-4}
λ_ν, λ_η	10.0, 10.0
ρ_0, ρ_1, ρ_2	0.01, 0.25, 0.85
α, β	1.5, 1.5

Table 3 Physical constants in all simulations.

Parameter	Value
Gravitational const. μ	$1.327\,124\,4 \times 10^{11}\ \text{km}^3\,\text{s}^{-2}$
Gravitational accel. g_0	$9.806\,65 \times 10^{-3}\ \text{km}\,\text{s}^{-2}$
Length unit LU = AU	$1.495\,978\,707 \times 10^{8}$ km
Velocity unit VU	$\sqrt{\mu\ \text{AU}^{-1}}$
Time unit TU	$\text{AU}\,\text{VU}^{-1}$
Acceleration unit ACU	$\text{VU}\,\text{TU}^{-1}$
Mass unit MU	m_0

Table 4 Coefficients for the real thruster model for the 2000 SG344 transfer.

$T_{\max}$	I_{sp}	P_{in}
$a_0 = -0.7253\,\text{mN}$	$b_0 = 2652\,\text{s}$	$c_0 = 840.11\,\text{W}$
$a_1 = 0.024\,81\,\text{mN}\,\text{W}^{-1}$	$b_1 = -18.123\,\text{s}\,\text{W}^{-1}$	$c_1 = -1754.3\,\text{W}\,\text{AU}^{-1}$
$a_2 = 0$	$b_2 = 0.3887\,\text{s}\,\text{W}^{-2}$	$c_2 = 1625.01\,\text{W}\,\text{AU}^{-2}$
$a_3 = 0$	$b_3 = -0.001\,74\,\text{s}\,\text{W}^{-3}$	$c_3 = -739.87\,\text{W}\,\text{AU}^{-3}$
$a_4 = 0$	$b_4 = 0$	$c_4 = 134.45\,\text{W}\,\text{AU}^{-4}$

Table 5 Coefficients for the real thruster model for the Dionysus transfer.

$T_{\max}$	I_{sp}
$\tilde{a}_0 = 0.1069\,\text{N}$	$\tilde{b}_0 = 3000\,\text{s}$
$\tilde{a}_1 = 3.9307 \times 10^{-4}\,\text{N}\,\text{W}^{-1}$	$\tilde{b}_1 = 0$

with r in AU. A similar model as in [18] is used for the Dionysus transfer:

$$T_{\max}(P_{\text{in}}) = \tilde{a}_0 + \tilde{a}_1\, P_{\text{in}} \tag{43a}$$

$$I_{\text{sp}}(P_{\text{in}}) = \tilde{b}_0 + \tilde{b}_1\, P_{\text{in}} \tag{43b}$$

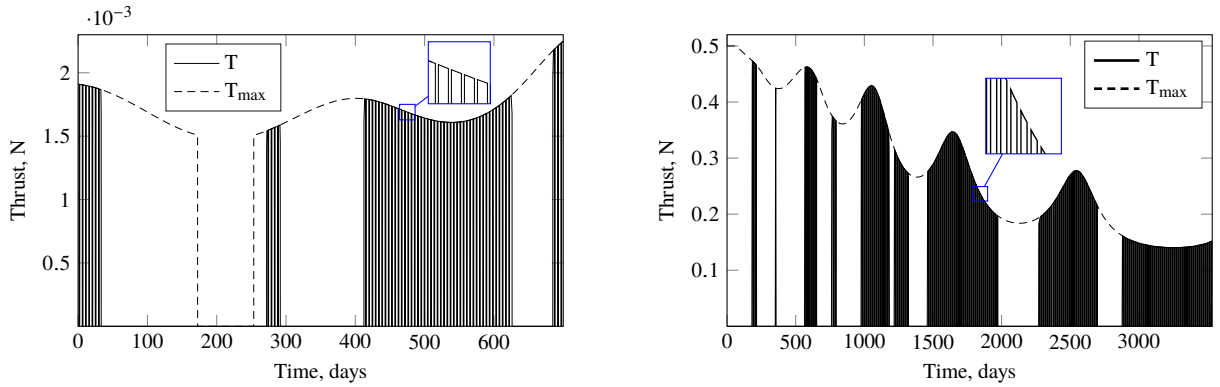
$$P_{\text{in}}(r) = \frac{1}{r^2} \tag{43c}$$

The coefficients are given in Tables 4 and 5, respectively.

A. No-Thrust Constraints

We first assess the performance of SCP when no-thrust periods are included. Without loss of generality, periods of one day (2000 SG344) and two days (Dionysus) per segment are considered where no thrust is available. This corresponds to duty cycles of 3.7 days (thrust) and 1 day (no thrust) for the 2000 SG344 transfer, and 12.2 days (thrust) and 2 days (no thrust) for Dionysus. Typical profiles of the thrust magnitude are shown in Figs. 3a and 3b for the variable I_{sp} and $T_{\max}$ case. The dashed black line refers to the maximum available thrust, and the vertical lines correspond to the on and off switches. Clearly, the thrust magnitude follows the available maximum thrust, being either zero or taking the maximum value that depends on the distance to the Sun. Note that the available thrust is zero between 175 and 250 days in Fig. 3a because the input power drops below the required minimum value of 90 W. Due to the long transfer time to Dionysus, there are many no-thrust periods, making this problem difficult to solve. The corresponding trajectories are depicted in Figs. 4a and 4b. The thrust arcs in red are discontinuous due to the no-thrust periods.

The profiles for the case when continuous thrust is considered look similar (see Fig. 5). The optimal solutions do therefore not alter significantly, and the no-thrust periods are often simply added around the nominal thrust arcs. As a



(a) Thrust profile for transfer to 2000 SG344.

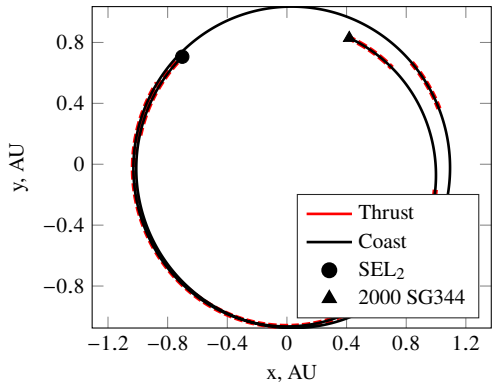
(b) Thrust profile for transfer to Dionysus.

Fig. 3 Typical thrust profiles obtained with SCP for the transfers to 2000 SG344 and Dionysus when no-thrust periods are considered (variable I_{sp} and T_{max}).

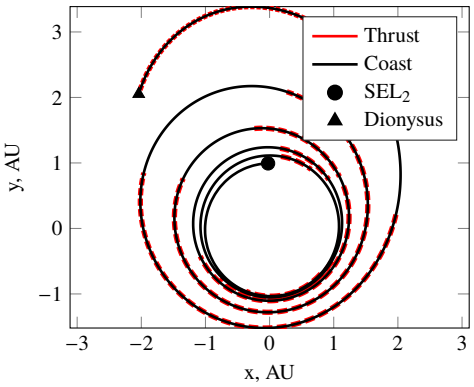
consequence, the final masses are often nearly the same. For example, we obtain $m_f = 2562\,\text{kg}$ (with no-thrust periods) and $m_f = 2564\,\text{kg}$ (continuous thrust) for the Dionysus transfer. The fuel consumption for 2000 SG344 with no-thrust periods is slightly higher ($m_f = 21.39\,\text{kg}$ compared to $m_f = 21.49\,\text{kg}$) due to the additional control actions around 290 days.

In case of constant I_{sp} and T_{max}, the procedure described in Section III.A can be used. Again, SCP finds accurate control profile as shown in Fig. 6.

The comparison of the number of iterations and CPU time with the standard SCP method (i.e., continuous thrust) is shown in Fig. 7. For each transfer, we compute ten optimal trajectories with a perturbed cubic interpolation guess where the number of revolutions ranges from 5.0 to 7.0 (Dionysus) and 1.8 to 2.1 (2000 SG344). Each case is computed several times to measure the CPU time. Median values are given, and the error bars refer to the 80th and 20th percentile of the corresponding quantity. SCP_{NT} denotes the case with no-thrust periods. With regard to the transfer to 2000 SG344, the number of iterations increases from 19 to 31 when no-thrust periods are enforced. The reason is that the trust-region size grows rapidly first, but then has to shrink again which requires additional iterations. This can be avoided if different trust-region parameters are chosen. Interestingly, SCP_{NT} requires slightly fewer iterations to find a solution for the Dionysus transfer. The CPU time behaves accordingly (see Fig. 7b), even though the discrepancy is larger for 2000 SG344 due the greater difference in the number of iterations. Most of the CPU time is required outside of the convex solver. The CPU time of the convex solver is approximately the same regardless of whether no-thrust periods are considered or not. The reason is that the convex program itself does not significantly change. However, the integration of the integrands in Eq. (18) requires more time compared to Eq. (9) due to the additional terms. As each segment is divided into subsegments in the no-thrust case, the number of integration intervals increases, while the integration periods become smaller. Therefore, if the same integrator is chosen, both the accuracy and computational effort increase.

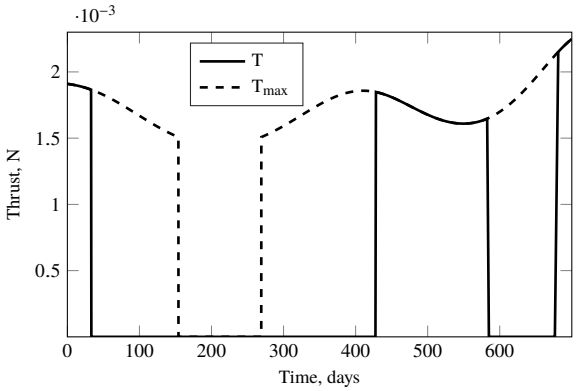


(a) Trajectory for transfer to 2000 SG344.

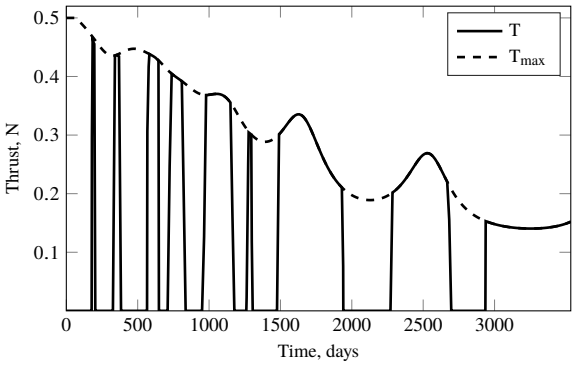


(b) Trajectory for transfer to Dionysus.

Fig. 4 Typical trajectories obtained with SCP for the transfers to 2000 SG344 and Dionysus when no-thrust periods are considered (variable I_{sp} and T_{max}).

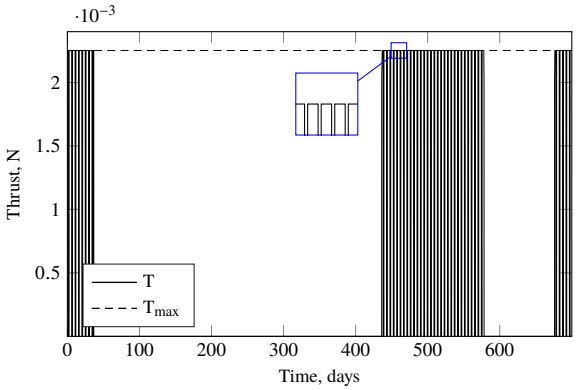


(a) Thrust profile for transfer to 2000 SG344.

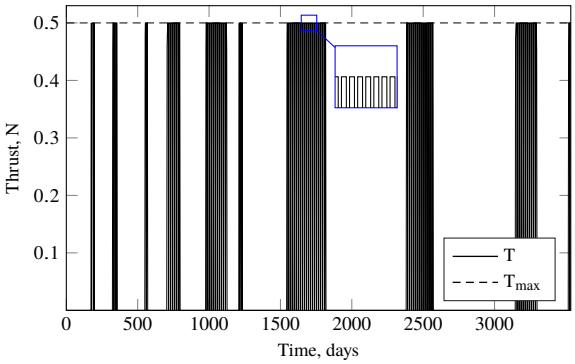


(b) Thrust profile for transfer to Dionysus.

Fig. 5 Typical thrust profiles obtained with SCP for the transfers to 2000 SG344 and Dionysus without considering no-thrust periods (variable I_{sp} and T_{max}).

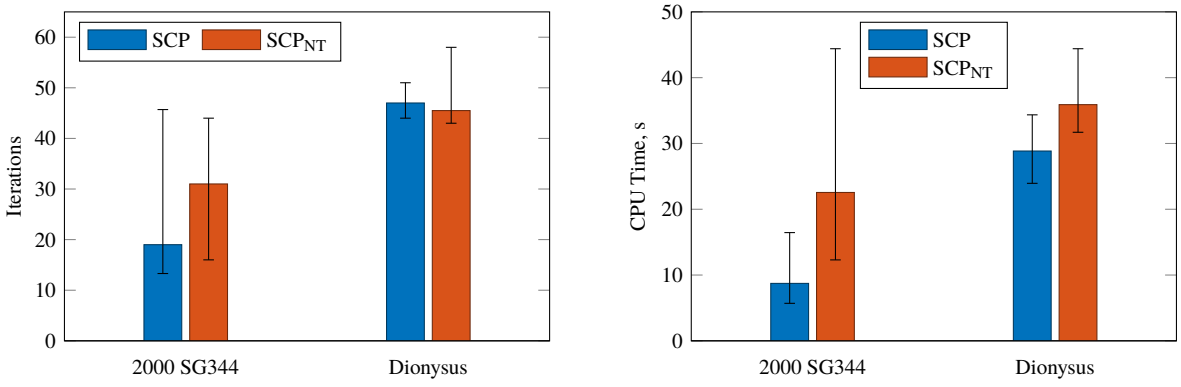


(a) Thrust profile for transfer to 2000 SG344.



(b) Thrust profile for transfer to Dionysus.

Fig. 6 Typical thrust profiles obtained with SCP for the transfers to 2000 SG344 and Dionysus when no-thrust periods are considered (constant I_{sp} and T_{max}).



(a) Number of iterations. **(b) CPU time.**

Fig. 7 Comparison of the number of iterations and CPU time for standard SCP and SCP$_{NT}$ where no-thrust periods are considered.

B. Moving Target

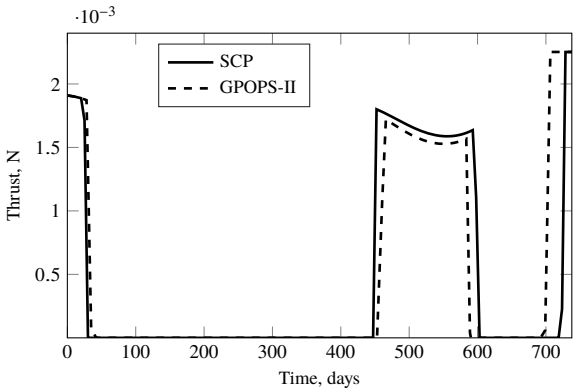
We now assess the performance of the moving target algorithm. Note that no-thrust periods are not considered to ensure a fair comparison with GPOPS-II. In case of the cubic interpolation guess, GPOPS-II is often able to find (slightly) higher final masses than SCP. With regard to the 2000 SG344 transfer, GPOPS-II determines an optimal solution with $t_f = 738$ days and $m_f = 21.59$ kg, whereas SCP finds $t_f = 749$ days and $m_f = 21.56$ kg. The corresponding thrust profiles are shown in Fig. 8a. This discrepancy increases for the Dionysus transfer, where the final mass obtained with GPOPS-II can be a few hundred kilograms larger (e.g., $m_f = 2823$ kg and $t_f = 3534$ days for GPOPS-II vs. $m_f = 2650$ kg and $t_f = 3514$ days for SCP in Fig. 8b). Apparently, GPOPS-II finds a different optimal solution for the provided initial guess, probably due to the trust-region constraint and the successive linearization within SCP. Yet, SCP requires only approximately 20 – 25 s (2000 SG344) and 30 – 40 s (Dionysus) to solve the problem. In contrast, the CPU time for GPOPS-II is often one or even two orders of magnitude higher. The propagation error (i.e., the difference between the optimized states and the propagated states) is similar for SCP and GPOPS-II and of the order $10^2 - 10^3$ km.

The problem size increases only slightly for the moving target problem. For example, t_f and the slack variable $\boldsymbol{\zeta}$ are now part of the solution vector for SCP, and the additional term in Eq. (40) is to be integrated. As a consequence, we found only a small rise of approximately 5 % in CPU time for the moving target problem in our simulations compared to the standard SCP method when solving the fuel-optimal problem (see blue bars in Fig. 7).

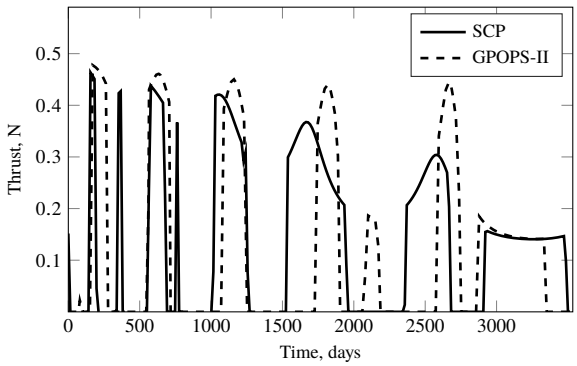
C. Moving Target and No-Thrust Constraints

Naturally, dynamic endpoint and no-thrust constraints can be combined into one optimization. In this case, the CPU time is dominated by the no-thrust periods due to the increased computational effort; letting the final boundary conditions vary with time does not require significantly more CPU time.

Considering the previous case where we obtain a time of flight of 749 days and a final mass of 21.56 kg for the 2000



(a) Thrust profiles for transfer to 2000 SG344.



(b) Thrust profiles for transfer to Dionysus.

Fig. 8 Comparison of thrust profiles obtained with SCP and GPOPS-II for the moving target problem and the transfers to 2000 SG344 and Dionysus.

SG344 transfer, these values change to $t_f = 756$ days and $m_f = 21.50$ kg if no-thrust periods are included. The time of flight for the Dionysus transfer is nearly identical, and the final mass decreases slightly from 2650 kg to 2632 kg. The corresponding plots of the state and control trajectories look similar and are therefore omitted. Naturally, different input parameters (e.g., initial guess, trust-region parameters) may yield (slightly) different results as the algorithm may find different optimal solutions.

VI. Conclusion

In this work, methods are proposed to include no-thrust periods and time-varying endpoint constraints in the sequential convex programming approach. It was shown that planetary ephemeris for dynamic endpoint constraints can be incorporated to increase the flexibility and capability of convex low-thrust trajectory optimization methods. The resulting nonlinear final boundary condition in the moving target problem can be handled by SCP even if a poor initial guess is provided. The rapid calculation speed compared to nonlinear programming solvers makes SCP an excellent choice for preliminary studies. As no-thrust constraints can directly be included in the optimization process without a significant increase in computational effort and decrease in convergence, the proposed approach can be considered another step towards computing more mission-compliant trajectories using convex programming techniques.

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).

References

[1] Slavinskis, A., Janhunen, P., Toivanen, P., Muinonen, K., Penttilä, A., Granvik, M., Kohout, T., Gritsevich, M., Slavinskis, A., Pajusalu, M., Sünter, I., Ehrpais, H., Dalbins, J., Iakubivskyi, I., Eenmäe, T., Pajusalu, M., Ilbis, E., Ehrpais, H., Muinonen, K., Gritsevich, M., Mauro, D., Stupl, J., Rivkin, A. S., and Bottke, W. F., "Nanospacecraft Fleet for Multi-Asteroid Touring With Electric Solar Wind Sails," *2018 IEEE Aerospace Conference*, IEEE, New York, 2018. https://doi.org/10.1109/AERO.2018.8396670.

[2] Hein, A., Saidani, M., and Tollu, H., "Exploring Potential Environmental Benefits of Asteroid Mining," *69th International Astronautical Congress*, International Astronautical Federation, 2018. Paper IAC-18,D4,5,11,x47396.

[3] Klesh, A., and Krajewski, J., "MarCO: Mars Cube One – Lessons Learned from Readying the First Interplanetary Cubesats for Flight," *Lunar and Planetary Science Conference*, Lunar and Planetary Institute, 2018. Paper 2923.

[4] Dotto, E., and Zinzi, A., "Impact Observations of Asteroid Dimorphos via Light Italian CubeSat for Imaging of Asteroids (LICIACube)," *Nature Communications*, Vol. 14, No. 1, 2023, p. 3055. https://doi.org/10.1038/s41467-023-38705-0.

[5] Oh, D., Landau, D., Randolph, T., Timmerman, P., Chase, J., Sims, J., and Kowalkowski, T., "Analysis of System Margins on Deep Space Missions Using Solar Electric Propulsion," *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, 2008. https://doi.org/10.2514/6.2008-5286, Paper AIAA 2008-5286.

[6] Ozaki, N., Campagnola, S., and Funase, R., "Tube Stochastic Optimal Control for Nonlinear Constrained Trajectory Optimization Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 4, 2020, pp. 645–655. https://doi.org/10.2514/1.G004363.

[7] Rayman, M. D., Fraschetti, T. C., Raymond, C. A., and Russell, C. T., "Coupling of system resource margins through the use of electric propulsion: Implications in preparing for the Dawn mission to Ceres and Vesta," *Acta Astronautica*, Vol. 60, No. 10, 2007, pp. 930–938. https://doi.org/10.1016/j.actaastro.2006.11.012.

[8] Englander, J. A., Vavrina, M. A., and Ghosh, A. R., "Multi-Objective Hybrid Optimal Control for Multiple-Flyby Low-Thrust Mission Design," *AAS/AIAA Space Flight Mechanics Meeting*, 2015. Paper AAS 15-227.

[9] Quadrelli, M. B., Wood, L. J., Riedel, J. E., McHenry, M. C., Aung, M., Cangahuala, L. A., Volpe, R. A., Beauchamp, P. M., and Cutts, J. A., "Guidance, Navigation, and Control Technology Assessment for Future Planetary Science Missions," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 7, 2015, pp. 1165–1186. https://doi.org/10.2514/1.G000525.

[10] Lu, P., "Introducing Computational Guidance and Control," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 193–193. https://doi.org/10.2514/1.G002745.

[11] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193 – 207. https://doi.org/10.2514/2.4231.

[12] Casalino, L., Colasurdo, G., and Pastrone, D., "Optimal Low-Thrust Escape Trajectories Using Gravity Assist," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 5, 1999, pp. 637–642. https://doi.org/10.2514/2.4451.

[13] Zhang, C., Topputo, F., Bernelli-Zazzera, F., and Zhao, Y. S., "Low-Thrust Minimum-Fuel Optimization in the Circular Restricted Three-Body Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 8, 2015, pp. 1501–1510. https://doi.org/10.2514/1.G001080.

[14] Betts, J. T., "Very Low-Thrust Trajectory Optimization Using a Direct SQP Method," *Journal of Computational and Applied Mathematics*, Vol. 120, No. 1, 2000, pp. 27–40. https://doi.org/10.1016/S0377-0427(00)00301-0.

[15] Hofmann, C., and Topputo, F., "Rapid Low-Thrust Trajectory Optimization in Deep Space Based on Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 7, 2021, pp. 1379–1388. https://doi.org/10.2514/1.G005839.

[16] Mengali, G., and Quarta, A. A., "Fuel-Optimal, Power-Limited Rendezvous with Variable Thruster Efficiency," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 6, 2005, pp. 1194–1199. https://doi.org/10.2514/1.12480.

[17] Li, T., Wang, Z., and Zhang, Y., "Double-Homotopy Technique for Fuel Optimization of Power-Limited Interplanetary Trajectories," *Astrophysics and Space Science*, Vol. 364, No. 9, 2019, p. 144. https://doi.org/10.1007/s10509-019-3637-6.

[18] Taheri, E., Junkins, J. L., Kolmanovsky, I., and Girard, A., "A Novel Approach for Optimal Trajectory Design With Multiple Operation Modes of Propulsion System, Part 1," *Acta Astronautica*, Vol. 172, 2020, pp. 151–165. https://doi.org/10.1016/j.actaastro.2020.02.042.

[19] Arya, V., Taheri, E., and Junkins, J. L., "Low-Thrust Gravity-Assist Trajectory Design Using Optimal Multimode Propulsion Models," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 7, 2021, pp. 1280–1294. https://doi.org/10.2514/1.G005750.

[20] Wang, Y., and Topputo, F., "Indirect Optimization of Power-Limited Asteroid Rendezvous Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 5, 2022, pp. 962–971. https://doi.org/10.2514/1.G006179.

[21] Mannocchi, A., Giordano, C., and Topputo, F., "A Homotopic Direct Collocation Approach for Operational-Compliant Trajectory Design," *The Journal of the Astronautical Sciences*, 2022. https://doi.org/10.1007/s40295-022-00351-x.

[22] Jia, F., Qiao, D., Han, H., and Li, X., "Efficient Optimization Method for Variable-Specific-Impulse Low-Thrust Trajectories With Shutdown Constraint," *Science China Technological Sciences*, Vol. 65, No. 3, 2022, p. 581–594. https://doi.org/10.1007/s11431-021-1949-0.

[23] Carson, J. M., and Açıkmeşe, B., "A Model Predictive Control Technique with Guaranteed Resolvability and Required Thruster Silent Times for Small-Body Proximity Operations," *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2006. https://doi.org/10.2514/6.2006-6780, Paper AIAA 2006-6780.

[24] Nurre, N. P., and Taheri, E., "Duty-Cycle-Aware Low-Thrust Trajectory Optimization Using Embedded Homotopy," *Acta Astronautica*, Vol. 212, 2023, pp. 630–642. https://doi.org/10.1016/j.actaastro.2023.08.022.

[25] Wang, Z., and Grant, M. J., "Minimum-Fuel Low-Thrust Transfers for Spacecraft: A Convex Approach," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54, No. 5, 2018, pp. 2274–2290. https://doi.org/10.1109/TAES.2018.2812558.

[26] Chi, Z., Li, H., Jiang, F., and J., L., "Power-Limited Low-Thrust Trajectory Optimization With Operation Point Detection," *Astrophysics and Space Science*, Vol. 363, No. 6, 2018, p. 122. https://doi.org/10.1007/s10509-018-3344-8.

[27] Topputo, F., Wang, Y., Giordano, G., Franzese, V., Goldberg, H., Perez-Lissi, F., and Walker, R., "Envelop of Reachable Asteroids by M-ARGO CubeSat," *Advances in Space Research*, Vol. 67, No. 12, 2021, pp. 4193–4221. https://doi.org/10.1016/j.asr.2021.02.031.

[28] Lu, P., and Liu, X., "Autonomous Trajectory Planning for Rendezvous and Proximity Operations by Conic Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 2, 2013, pp. 375–389. https://doi.org/10.2514/1.58436.

[29] Liu, X., and Lu, P., "Solving Nonconvex Optimal Control Problems by Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765. https://doi.org/10.2514/1.62110.

[30] Szmuk, M., Reynolds, T. P., and Açıkmeşe, B., "Successive Convexification for Real-Time Six-Degree-of-Freedom Powered Descent Guidance with State-Triggered Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 8, 2020, pp. 1399–1413. https://doi.org/10.2514/1.G004549.

[31] Benedikter, B., Zavoli, A., Colasurdo, G., Pizzurro, S., and Cavallini, E., "Convex Approach to Three-Dimensional Launch Vehicle Ascent Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 6, 2021, pp. 1116–1131. https://doi.org/10.2514/1.G005376.

[32] Bernardini, N., Baresi, N., and Armellin, R., "Fuel-Efficient Stationkeeping of Quasi-Satellite Orbits via Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 12, 2023, pp. 2421–2431. https://doi.org/10.2514/1.G007557.

[33] Açıkmeşe, B., and Ploen, S. R., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366. https://doi.org/10.2514/1.27553.

[34] Lubey, D. P., and Scheeres, D. J., "Identifying and Estimating Mismodeled Dynamics via Optimal Control Policies and Distance Metrics," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 5, 2014, pp. 1512–1523. https://doi.org/10.2514/1.G000369.

[35] Hofmann, C., and Topputo, F., "Homotopic Approach for High-Fidelity Convex Low-Thrust Trajectory Optimization," , 2023. *IEEE Transactions on Aerospace and Electronic Systems*, submitted for publication.

[36] Hofmann, C., and Topputo, F., "Embedded Homotopy for Convex Low-Thrust Trajectory Optimization with Operational Constraints," *AAS/AIAA Astrodynamics Specialist Conference*, 2022. Paper AAS 22-750.

[37] Açıkmeşe, B., and Ploen, S. R., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366. https://doi.org/10.2514/1.27553.

[38] Blackmore, L., Açikmeşe, B., and Scharf, D. P., "Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, 2010, pp. 1161–1171. https://doi.org/10.2514/1.47202.

[39] Bryson, A. E., and Ho, Y.-C., *Applied Optimal Control: Optimization, Estimation, and Control*, Blaisdell Publishing Company, 1969, pp. 90–127.

[40] Açikmeşe, B., Carson, J. M., and Blackmore, L., "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, 2013, pp. 2104–2113. https://doi.org/10.1109/TCST.2012.2237346.

[41] Liu, X., "Fuel-Optimal Rocket Landing with Aerodynamic Controls," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 1, 2019, pp. 65–77. https://doi.org/10.2514/1.G003537.

[42] Andreis, E., Franzese, V., and Topputo, F., "Onboard Orbit Determination for Deep-Space CubeSats," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 8, 2022, pp. 1466–1480. https://doi.org/10.2514/1.G006294.

[43] Morelli, A. C., Merisio, G., Hofmann, C., and Topputo, F., "A Convex Guidance Approach to Target Ballistic Capture Corridors at Mars," *AAS Guidance, Navigation and Control Conference*, 2022. Paper AAS 22-083.

[44] Szmuk, M., Reynolds, T. P., and Açıkmeşe, B., "Successive Convexification for Real-Time Six-Degree-of-Freedom Powered Descent Guidance with State-Triggered Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 8, 2020, pp. 1399–1413. https://doi.org/10.2514/1.G004549.

[45] Taheri, E., Kolmanovsky, I., and Atkins, E., "Enhanced Smoothing Technique for Indirect Optimization of Minimum-Fuel Low-Thrust Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 11, 2016, pp. 2500–2511. https://doi.org/10.2514/1.g000379.

[46] Patterson, M. A., and Rao, A. V., "GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems using Hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming," *ACM Trans. Math. Softw.*, Vol. 41, No. 1, 2014. https://doi.org/10.1145/2558904.

[47] Wächter, A., and Biegler, L., "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, Vol. 106, No. 1, 2006, pp. 25–57. https://doi.org/10.1007/s10107-004-0559-y.

[48] Domahidi, A., Chu, E., and Boyd, S., "ECOS: An SOCP Solver for Embedded Systems," *European Control Conference*, 2013, pp. 3071–3076. https://doi.org/10.23919/ECC.2013.6669541.

[49] Taheri, E., and Abdelkhalik, O., "Initial Three-Dimensional Low-Thrust Trajectory Design," *Advances in Space Research*, Vol. 57, No. 3, 2016, pp. 889 – 903. https://doi.org/10.1016/j.asr.2015.11.034.

[50] Hofmann, C., Morelli, A. C., and Topputo, F., "Performance Assessment of Convex Low-Thrust Trajectory Optimization Methods," *Journal of Spacecraft and Rockets*, Vol. 60, No. 1, 2023, pp. 299–314. https://doi.org/10.2514/1.A35461.

[51] Mao, Y., Szmuk, M., Xu, X., and Açıkmeşe, B., "Successive Convexification: A Superlinearly Convergent Algorithm for Non-convex Optimal Control Problems," , 2019. Preprint, https://arxiv.org/abs/1804.06539.