C. Hofmann, F. Topputo
Homotopic Approach for High-Fidelity Convex Low-Thrust Trajectory Optimization
IEEE Transactions on Aerospace and Electronic Systems, Vol. 61, N. 5, p. 11991-12007, 2025 (published online 22/05/2025)
doi:10.1109/taes.2025.3572466

Homotopic Approach for High-Fidelity Convex Low-Thrust Trajectory Optimization

Christian Hofmann and Francesco Topputo

***Abstract*—The low-thrust minimum-fuel trajectory optimization problem is solved in a high-fidelity model using convex optimization. The problem is convexified and discretized with the first-order-hold method. A homotopic approach is developed to improve convergence, accuracy, and computational effort. The homotopy is embedded into the optimization process where the homotopic parameter is dynamically adjusted based on the constraint violation. This method is combined with a high-fidelity model where the complexity of the dynamics and constraints is successively increased. In particular, n-body dynamics, solar radiation pressure, variable specific impulse and maximum thrust are considered. The effectiveness of the proposed method is assessed in several fuel-optimal transfers and compared with related works and a state-of-the-art nonlinear programming solver.**

***Index Terms*—Nonlinear optimal control, low-thrust trajectory optimization, homotopic approach, sequential convex programming.**

I. INTRODUCTION

Highly efficient propulsion systems such as electric propulsion have become an important development during the past years. Even though they provide only low thrust, considerable savings of propellant can be achieved due to their high specific impulse. This enables new space missions, but also requires new trajectory design techniques as the thrusting periods increase and therefore, the assumption of instantaneous impulses does not hold anymore. Although extensive research has been conducted in the field of low-thrust trajectory optimization [1], it was only recently that autonomous guidance and control have attracted more and more attention [2], [3]. Instead of operating spacecraft from ground, flight-related tasks such as guidance, navigation, and control are to be shifted on board. This, however, requires the spacecraft to determine reliably mission compliant trajectories in real time.

Planning the trajectory involves solving a nonlinear optimal control problem (OCP). In contrast to indirect methods that suffer from convergence issues when a poor initial guess is provided [4], direct methods are often preferred, especially when high-fidelity models are used [5]. The infinite-dimensional problem is discretized into a finite-dimensional nonlinear parameter optimization problem. There are several approaches to solve the resulting nonlinear program (NLP), ranging from single and multiple shooting to direct collocation [4]. As solving the NLP directly often requires high computational power and a decent initial guess, sequential convex programming (SCP) techniques have become a popular alternative in the past few years [6]. The problem is convexified and then iteratively solved until the solution satisfies the original nonlinear constraints. Convex programming techniques have been applied successfully in many applications, such as powered descent landing [7], nonlinear model predictive control [8], collision avoidance, and robotics [9]. Their use in low-thrust trajectory optimization has also become more frequent. For example, the minimum-time and minimum-fuel problems have recently been solved with a convex approach [10], [11]. More complex interplanetary trajectories have been computed with an adaptive pseudospectral method [12]. Moreover, the real-time performance of a SCP method on an ARM processor was assessed for interplanetary transfers [13].

Low-thrust trajectory optimization problems that consider higher-fidelity models (for example, third-body perturbations, solar radiation pressure, realistic thruster model) have been solved with indirect [14], [15] and direct, NLP-based methods [16], [17], [18]. With regard to convex programming techniques, the accuracy and convergence can be poor for highly nonlinear applications due to the successive linearization. Therefore, most researchers considered only simple two-body dynamics. The literature on solving more complex models, for example the circular-restricted three-body problem using convex programming, is scarce [19]. In that work, however, the SCP approach had to be combined with a nonlinear programming method to deal with the highly nonlinear dynamics and to achieve convergence. In [20], trajectories to the dwarf planet Ceres are addressed using two-body dynamics and incorporating gravity assists. Notably, an adaptive mesh refinement method is used to mitigate convergence issues. The work in [21] employs SCP to calculate minimum-fuel low-thrust gravity assist trajectories, serving as initial guesses for a nonlinear programming solver. Despite considering n-body dynamics, this method is restricted to scenarios with fixed maximum thrust and short flight times with very few orbital revolutions. Additionally, the spacecraft mass is excluded from the simulations. Similarly, only fixed maximum thrust and few revolutions are considered in [22]. Further, trajectories involving variable specific impulse and maximum thrust have been explored in [23], yet the analysis remains within the two-body dynamics framework, neglecting additional perturbations. A convex optimization-based method is developed in [24] to track a reference trajectory, accounting for J_2 perturbation

Christian Hofmann was with the Department of Aerospace Science and Technology, Politecnico di Milano, 20156 Milan, Italy. He is now with the Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: hofmannc@mit.edu). Francesco Topputo is with the Department of Aerospace Science and Technology, Politecnico di Milano, 20156 Milan, Italy (e-mail: francesco.topputo@polimi.it).

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).

and atmospheric drag, specifically applied to active debris removal missions. This approach, however, limits itself to a subset of orbital elements and relies on highly accurate reference trajectories to enhance convergence. When additional perturbations are considered, convergence and accuracy deteriorate considerably, along with an increase in the number of iterations and computational time. Still, it remains to be investigated how SCP performs when high-fidelity dynamical models are considered for long-duration low-thrust trajectory optimization problems, and how it compares with state-of-the-art NLP solvers.

Indirect methods often make use of a homotopic approach to improve convergence: a series of simpler problems is solved first where each solution is used as an initial guess for the next problem [25], [26]. The process continues until the original problem is solved. This can also be applied to direct methods where an easier problem (e.g., two-body dynamics) is considered first, and then the result is used to determine a solution in a higher-fidelity model (e.g., n-body dynamics). For example, it was shown that using a homotopy from the energy- to the fuel-optimal problem in a convex programming environment can improve convergence [27]. In previous works, however, each subproblem is solved to full optimality, and the step size is fixed and defined by the user [28], [29]. This means that the algorithm requires at least a certain number of homotopic steps to achieve convergence. Depending on the problem structure, however, it may be beneficial to perform larger steps and therefore, achieve faster convergence. Still, most current methods do not adjust the step size dynamically, but rely on a "solve to full optimality and then proceed to the next step" approach. Often, a conservative (i.e., small) value of the step size is chosen to avoid non-convergence. This procedure is not ideal, and it would be desirable to adjust the step size dynamically based on information that can be retrieved from the optimization problem [30]. For example, a continuation method is included into SCP in [31] to model discrete logic constraints. In [22], an adaptive continuation technique is used to change the continuation parameter based on user-defined parameters and the convergence of previous optimizations. Even though such an adaptive approach is preferable to fixed step sizes, some heuristic is needed to define and update the parameters in each iteration. Moreover, each subproblem is still solved to full optimality which results in additional iterations. Other works, such as [32], embed a homotopy within the nonlinear interior-point solver. However, this approach necessitates modifications to an NLP solver and cannot be used as a black box with any arbitrary solver.

The main contribution of this work is the introduction of an embedded homotopic approach, where SCP and numerical continuation steps are executed concurrently. In contrast to conventional methods, where the problem is solved to full optimality after updating the homotopic parameter, we directly integrate the update mechanism into the SCP process. In this context, a trust-region update mechanism tailored to the embedded homotopic approach is presented to expand the feasible region when the homotopic parameter changes. This allows for a gradual increase in model fidelity, incorporating n-body dynamics, solar radiation pressure, and variable specific impulse and maximum thrust. Additionally, we perform an in-depth analysis of the performance properties of the developed approach in several fuel-optimal transfers. This investigation is significant because SCP has demonstrated its potential as a promising method for optimization and autonomous guidance problems. By showcasing its applicability with high-fidelity dynamical models, we further validate its effectiveness. Moreover, we compare the performance of our proposed approach with related works and a state-of-the-art NLP solver.

The paper is structured as follows. Sections II and III describe the optimal control problem and the first-order-hold discretization method, respectively. In Section IV, the homotopic approach to higher-fidelity models is explained, and Section V presents the results of the numerical simulations. Section VI concludes this paper.

II. PROBLEM FORMULATION

Considering a two-body model with the Sun as the primary, the equations of motion of a spacecraft in Cartesian coordinates are given by

$$\dot{\mathbf{r}}(t) = \mathbf{v}(t) \tag{1}$$

$$\dot{\mathbf{v}}(t) = -\frac{\mu\,\mathbf{r}(t)}{r(t)^3} + \frac{\mathbf{T}(t)}{m(t)} \tag{2}$$

$$\dot{m}(t) = -\frac{\|\mathbf{T}(t)\|_2}{g_0\, I_{\text{sp}}} \tag{3}$$

where $\mathbf{r}(t) \in \mathbb{R}^3$ ($r = \|\mathbf{r}(t)\|_2$), $\mathbf{v}(t) \in \mathbb{R}^3$, and $m(t) \in \mathbb{R}$ denote the position, velocity, and mass of the spacecraft, respectively. μ is the gravitational constant of the primary body, $\mathbf{T}(t) \in \mathbb{R}^3$ the thrust vector, g_0 the gravitational acceleration at sea level, and $I_{\text{sp}} \in \mathbb{R}$ the specific impulse.

The fuel-optimal control problem in space flight seeks to minimize fuel consumption, or equivalently to maximize the mass at the final time t_f:

$$\min_{\mathbf{T}(t)} \quad -m(t_f) \tag{4}$$

The rendezvous problem is considered in this work, i.e., the prescribed boundary conditions at the initial t_0 and final time are

$$\mathbf{r}(t_0) = \mathbf{r}_0, \quad \mathbf{v}(t_0) = \mathbf{v}_0, \quad m(t_0) = m_0 \tag{5}$$

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

where the subscripts $(\cdot)_0$ and $(\cdot)_f$ refer to the initial and final values, respectively. Note that the final mass is free. The following lower and upper bounds on the thrust magnitude are imposed:

$$0 \leq \|\mathbf{T}(t)\|_2 \leq T_{\max} \tag{7}$$

with the maximum available thrust $T_{\max}$. Introducing the following change of variables [7]

$$\Gamma(t) \coloneqq \frac{\|\mathbf{T}(t)\|_2}{m(t)}, \quad \boldsymbol{\tau}(t) \coloneqq \frac{\mathbf{T}(t)}{m(t)}, \quad z(t) \coloneqq \ln m(t) \tag{8}$$

and defining the states and controls as $\mathbf{x} = \left[\mathbf{r}^\top, \mathbf{v}^\top, z\right]^\top$ and $\mathbf{u} = \left[\boldsymbol{\tau}^\top, \Gamma\right]^\top$, respectively, allows us to decouple states and

controls in (2). The new dynamics $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$ then read

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \underbrace{\begin{bmatrix} \mathbf{v}(t) \\ -\mu\,\mathbf{r}(t)/r(t)^3 \\ 0 \end{bmatrix}}_{\mathbf{p}(\mathbf{x})} + \underbrace{\begin{bmatrix} \mathbf{0}_{3\times4} \\ \mathbf{1}_{3\times3} \quad \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} \quad -1/(g_0\,I_{\text{sp}}) \end{bmatrix}}_{\mathbf{B}} \begin{bmatrix} \boldsymbol{\tau}(t) \\ \Gamma(t) \end{bmatrix} \\ &= \mathbf{p}(\mathbf{x}(t)) + \mathbf{B}\,\mathbf{u}(t)\end{aligned} \tag{9}$$

As the change of variables results in the nonlinear constraint $0 \leq \Gamma(t) \leq T_{\max}\,\mathrm{e}^{-z(t)}$ of the upper bound of the thrust magnitude, (7) is linearized about the reference $\bar{z}(t)$:

$$0 \leq \Gamma(t) \leq T_{\max}\,\mathrm{e}^{-\bar{z}(t)}\,[1 - z(t) + \bar{z}(t)] \tag{10}$$

The nonconvexity of the equality constraint $\|\boldsymbol{\tau}(t)\|_2 = \Gamma(t)$ is eliminated by relaxing it to the second-order cone constraint $\|\boldsymbol{\tau}(t)\|_2 \leq \Gamma(t)$. This convexification is lossless as an optimal solution of the relaxed problem is also an optimal solution of the original problem [7]. The convexified optimization problem is then given by

$$\min_{\mathbf{u}(t)} \quad -z(t_f) + \lambda_\nu\,\|\boldsymbol{\nu}(t)\|_1 + \lambda_\eta\,\max(0, \eta(t)) \tag{11a}$$

$$\text{s. t.:} \quad \dot{\mathbf{x}}(t) = \mathbf{A}(\bar{\mathbf{x}}(t))\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t) + \mathbf{q}(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t)) + \boldsymbol{\nu}(t) \tag{11b}$$

$$\Gamma(t) \leq T_{\max}\,\mathrm{e}^{-\bar{z}(t)}\,[1 - z(t) + \bar{z}(t)] + \eta(t) \tag{11c}$$

$$\|\boldsymbol{\tau}(t)\|_2 \leq \Gamma(t) \tag{11d}$$

$$\|\mathbf{x}(t) - \bar{\mathbf{x}}(t)\|_1 \leq R \tag{11e}$$

$$\mathbf{r}(t_0) = \mathbf{r}_0,\ \mathbf{v}(t_0) = \mathbf{v}_0,\ z(t_0) = z_0 \tag{11f}$$

$$|\mathbf{r}(t_f) - \mathbf{r}_f| \leq \Delta\mathbf{r},\ |\mathbf{v}(t_f) - \mathbf{v}_f| \leq \Delta\mathbf{v} \tag{11g}$$

with

$$\begin{aligned}\mathbf{A}(\bar{\mathbf{x}}(t)) &\coloneqq \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big|_{\bar{\mathbf{x}}(t)},\ \mathbf{B} \coloneqq \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\Big|_{\bar{\mathbf{u}}(t)}, \\ \mathbf{q}(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t)) &\coloneqq \mathbf{f}(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t)) - \mathbf{A}(\bar{\mathbf{x}}(t))\,\bar{\mathbf{x}}(t) - \mathbf{B}\,\bar{\mathbf{u}}(t)\end{aligned} \tag{12}$$

The linearized dynamics in (11b) and the linearized upper bound on the thrust magnitude in (11c) are augmented with slack variables $\boldsymbol{\nu}(t)$ and $\eta(t) \geq 0$ to prevent artificial infeasibility. As $\boldsymbol{\nu}(t)$ and $\eta(t)$ would result in additional constraint violations when active, they are penalized in the cost function in (11a) with sufficiently large parameters $\lambda_\nu > 0$ and $\lambda_\eta > 0$. The trust-region constraint in (11e) with radius R is needed to keep the linearization accurate enough and close to the reference. The final boundary conditions are relaxed to inequality constraints in (11g) where $\Delta\mathbf{r}$ and $\Delta\mathbf{v}$ depend on the application and the desired accuracy.

III. FIRST-ORDER-HOLD DISCRETIZATION

In the first-order-hold (FOH) discretization method considered in this work, the time domain is divided into $N-1$ equally spaced segments

$$t_0 = t_1 < t_2 < \cdots < t_N = t_f \tag{13}$$

where N denotes the number of discretization points. Note that nodes and collocation points are the same in FOH. Given the controls $\mathbf{u}_k$ and $\mathbf{u}_{k+1}$ at nodes k and $k+1$, respectively, $k = 1, \ldots, N-1$, the continuous control profile $\mathbf{u}(t)$ is approximated as a piecewise affine function as follows:

$$\begin{aligned}\mathbf{u}(t) &= \underbrace{\frac{t_{k+1} - t}{t_{k+1} - t_k}}_{=:\,\lambda_-(t)}\,\mathbf{u}_k + \underbrace{\frac{t - t_k}{t_{k+1} - t_k}}_{=:\,\lambda_+(t)}\,\mathbf{u}_{k+1} \\ &= \lambda_-(t)\,\mathbf{u}_k + \lambda_+(t)\,\mathbf{u}_{k+1}, \qquad t \in [t_k, t_{k+1}]\end{aligned} \tag{14}$$

The linearized dynamics are given by

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\,\mathbf{x}(t) + \mathbf{B}(t)\,\lambda_-(t)\,\mathbf{u}_k + \mathbf{B}(t)\,\lambda_+(t)\,\mathbf{u}_{k+1} + \mathbf{q}(t) \tag{15}$$

which can be rewritten in discretized form as [33]

$$\mathbf{x}_{k+1} = \mathbf{A}_k\,\mathbf{x}_k + \mathbf{B}_k^-\,\mathbf{u}_k + \mathbf{B}_k^+\,\mathbf{u}_{k+1} + \mathbf{q}_k + \boldsymbol{\nu}_k \tag{16}$$

$\boldsymbol{\Phi}$ is the state transition matrix that satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\Phi}(t, t_0) = \mathbf{A}(t)\,\boldsymbol{\Phi}(t, t_0), \qquad \boldsymbol{\Phi}(t_0, t_0) = \mathbf{1} \tag{17}$$

where $\mathbf{1}$ denotes the identity matrix, and

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

$$\mathbf{B}_k^- = \mathbf{A}_k \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}^{-1}(t, t_k)\,\mathbf{B}(t)\,\lambda_-(t)\,\mathrm{d}t \tag{18b}$$

$$\mathbf{B}_k^+ = \mathbf{A}_k \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}^{-1}(t, t_k)\,\mathbf{B}(t)\,\lambda_+(t)\,\mathrm{d}t \tag{18c}$$

$$\mathbf{q}_k = \mathbf{A}_k \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}^{-1}(t, t_k)\,\mathbf{q}(t)\,\mathrm{d}t \tag{18d}$$

At each node, the matrices $\mathbf{A}_k \in \mathbb{R}^{7\times7}$, $\mathbf{B}_k^- \in \mathbb{R}^{7\times4}$, $\mathbf{B}_k^+ \in \mathbb{R}^{7\times4}$, and $\mathbf{q}_k \in \mathbb{R}^{7\times1}$ are calculated by integrating the state transition matrix in (17), the nonlinear dynamics in (9), and the integrands of (18b)–(18d) simultaneously with an explicit fixed-step 8th-order Runge-Kutta method. The reference state at $t \in [t_k, t_{k+1}]$ can be obtained using

$$\bar{\mathbf{x}}(t) = \bar{\mathbf{x}}_k + \int_{t_k}^{t} \mathbf{f}(\mathbf{x}(\xi), \mathbf{u}(\xi))\,\mathrm{d}\xi \tag{19}$$

IV. EMBEDDED HOMOTOPIC APPROACH FOR HIGH-FIDELITY MODELS

As adding high-fidelity models often increases the nonlinearity, a standard SCP algorithm may have difficulties to converge for complex problems. Therefore, homotopy-like procedures are often applied in indirect methods where the complexity of the problem is increased step by step. This section introduces a practical embedded homotopic approach, and demonstrates how n-body dynamics, solar radiation pressure, and variable specific impulse and maximum thrust are incorporated.

A. Methodology

In a homotopic approach, a parameter $\varepsilon \in [1, 0]$ is defined that is gradually reduced from $\varepsilon = 1$ (corresponds to a simpler problem) to $\varepsilon = 0$ (corresponds to the original, more complex problem that is eventually to be solved). Usually, each subproblem is solved to full optimality, and the step size n is fixed and defined by the user. This means that

the algorithm requires at least n homotopic steps to achieve convergence. Often, a conservative (i.e., small) value of the step size is chosen to avoid non-convergence. This procedure is not ideal as it may result in a large number of homotopic steps and thus, iterations and CPU time increase. Even if a larger step size is chosen and then reduced by some heuristic when the solver fails, this would require additional (and unwanted) iterations. Thus, instead of keeping the step size fixed, it is adjusted dynamically in this work. We propose a homotopy path based on the maximum constraint violation c_{max}. ε is updated at each SCP iteration depending on the constraint violation, which is a measure of the progress of the algorithm. Figure 1 shows three examples h_1, h_2, and h_3 that define different homotopy paths. A large c_{max} indicates that we are far from a feasible solution. Therefore, ε is also large, and we want to solve a simpler problem. Once the algorithm makes progress towards an optimal solution, it will select smaller ε values until the desired optimal control problem is solved. For instance, given a relatively simple optimization problem where the initial guess is good and hence, the initial constraint violation is small, our method can exploit this knowledge and select larger step sizes. This results in faster convergence. On the other hand, if we intend to solve a difficult problem where only a poor initial guess is provided, the algorithm will select smaller step sizes as c_{max} will decrease only slowly during the iterations.

Selecting a homotopy path may depend on the problem. For example, h_1 and h_2 in Fig. 1 are more conservative towards the end, whereas h_3 decreases ε more slowly at the beginning of the optimization process. We found in our simulations that the choice is not critical, and a linear path like h_1 or h_2 is often a good compromise. The functions are defined at the beginning of the algorithm based on the maximum constraint violation of the initial guess $c_{\text{max,guess}}$. Defining

$$w \coloneqq \frac{\varepsilon_{\text{max}} - \varepsilon_{\text{min}}}{\log_{10}\left(c_{\text{max,guess}}\right) - \log_{10}\left(c_{\text{max,tol}}\right) - \delta} \tag{20}$$

the homotopy path h_1 that is used throughout this paper can be written as follows:

$$h_1(\log_{10} c_{\text{max}}) = w \log_{10}\left(c_{\text{max}}\right) - w\left[\log_{10}\left(c_{\text{max,tol}}\right) + \delta\right] \tag{21}$$

where $c_{\text{max,guess}}$ and $c_{\text{max,tol}}$ denote the maximum constraint violation of the initial guess and the desired feasibility tolerance, respectively. $\delta > 0$ is a constant to further adjust the path.

n-Body Dynamics

The fidelity of the model is increased by considering the perturbing accelerations $\mathbf{a}_{\text{nbody}}$ of other bodies in the solar system and the solar radiation pressure $\mathbf{a}_{\text{SRP}}$. Omitting the time dependency, the augmented dynamics then read

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ -\mu\,\mathbf{r}/r^3 \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{a}_{\text{nbody}} \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{a}_{\text{SRP}} \\ 0 \end{bmatrix} + \mathbf{B}\,\mathbf{u} \tag{22}$$

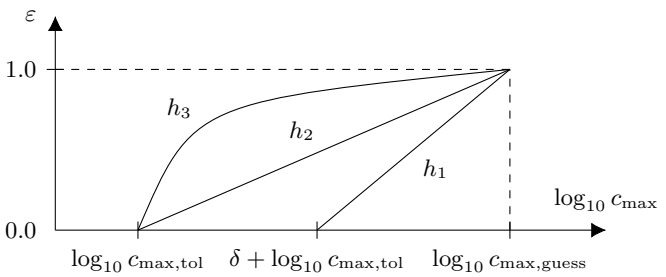


Fig. 1: Three examples of smooth homotopy paths h_1, h_2, and h_3 that define the relationship between the homotopic parameter ε and the maximum constraint violation c_{max}.

With regard to the solar radiation pressure (SRP), a simple cannonball model is used where the projected area A_{SC} of the spacecraft is assumed constant [34], and we have

$$\mathbf{a}_{\text{SRP}} = \frac{S_{\text{Sun}}\, C_R\, A_{\text{SC}}}{m}\,\frac{\mathbf{r}}{r^3} \tag{23}$$

with

$$S_{\text{Sun}} = \frac{L_{\text{Sun}}}{4\,\pi\,c} \tag{24}$$

$$L_{\text{Sun}} = 4\,\pi\,C_{\text{Sun}}\,\text{AU}^2 \tag{25}$$

C_R is the reflectivity coefficient of the spacecraft, S_{Sun} the solar pressure constant, L_{Sun} the luminosity of the Sun, and C_{Sun} the solar constant. c and AU denote the speed of light and astronomical unit, respectively. Note that the mass m in the denominator of (23) is replaced by e^z in the implementation due to the change of variables (see Section II).

Regarding the n-body dynamics, the $n = 10$ perturbations of the barycenters of Mercury, Venus, Earth, Moon, Mars, Jupiter, Saturn, Uranus, Neptune, and Pluto are considered. The acceleration is calculated as follows:

$$\mathbf{a}_{\text{nbody}} = \sum_{i}^{n} \mu_i \left(\frac{\mathbf{r}_{\text{sat},i}}{r_{\text{sat},i}^3} - \frac{\mathbf{r}_i}{r_i^3} \right) \tag{26}$$

μ_i is the gravitational constant of the ith body, $\mathbf{r}_i$ is the position of the ith body with respect to the Sun, and $\mathbf{r}_{\text{sat},i} = \mathbf{r}_i - \mathbf{r}$ denotes the position of the ith body with respect to the spacecraft. The time-dependent positions $\mathbf{r}_i$ of the perturbing bodies can be obtained using the software SPICE [35].

Due to the additional perturbing accelerations, the new dynamical system is highly nonlinear, and the dynamics are linearized as per Section II. When including n-body dynamics, however, special treatment is required when the spacecraft flies in close proximity of another perturbing body (e.g., in case of flybys). This may become a problem as $r_{\text{sat},i}$ is close to zero and hence, the corresponding denominator in (26) is close to zero. When integrating the nonlinear dynamics and state transition matrix during the discretization, the position and velocity of the spacecraft can change rapidly between two integration steps. The reason is the large gravitational force in proximity of a planet. If a fixed-step integrator is used, the resulting perturbing accelerations can vary considerably at each step. As a consequence, the integration becomes inaccurate, and the SCP method eventually fails to converge.

To overcome this issue, a constraint $r_{\text{sat},i} \geq r_{\min}$ can be added to require a minimum distance $r_{\min}$ between spacecraft and planet; flybys are thus avoided. In this work, however, we use a variable-step integrator for a segment when the condition $r_{\text{sat},i} \leq r_{\min}$ is satisfied, i.e., when the spacecraft is close to a planet. This ensures high accuracy without rapid changes and does not unnecessarily constrain the solution space. As it is only used for very few segments, the computational load does not increase considerably.

Instead of solving the problem with n-body dynamics and SRP directly, we introduce the homotopic parameters $\varepsilon_{\text{nbody}}$ and ε_{SRP} and augment (22) as follows:

$$\begin{aligned}\mathbf{f}(\mathbf{x},\mathbf{u}) = &\begin{bmatrix}\mathbf{v}\\ -\mu\,\mathbf{r}/r^3\\ 0\end{bmatrix} + (1-\varepsilon_{\text{nbody}})\begin{bmatrix}\mathbf{0}\\ \mathbf{a}_{\text{nbody}}\\ 0\end{bmatrix}\\ &+ (1-\varepsilon_{\text{SRP}})\begin{bmatrix}\mathbf{0}\\ \mathbf{a}_{\text{SRP}}\\ 0\end{bmatrix} + \mathbf{B}\,\mathbf{u}\end{aligned} \tag{27}$$

During the first iterations, the constraint violations are usually large, and $\varepsilon_{\text{nbody}}$ and ε_{SRP} are usually close to one. This means that the perturbing accelerations are small. When the maximum constraint violation decreases, ε is gradually decreased until $\varepsilon = 0$, i.e., the original problem is solved. Our rationale is that the solver is more likely to find a feasible solution when increasing the nonlinearity (and hence, the linearized term) only step by step compared to solving the full problem directly. Note that such an approach allows us to define different homotopy paths and ε for each perturbing force so that each term can be controlled separately. This may be useful if one perturbation is considered more critical and therefore requires a more conservative step size at the beginning (see also (21) and Fig. 1).

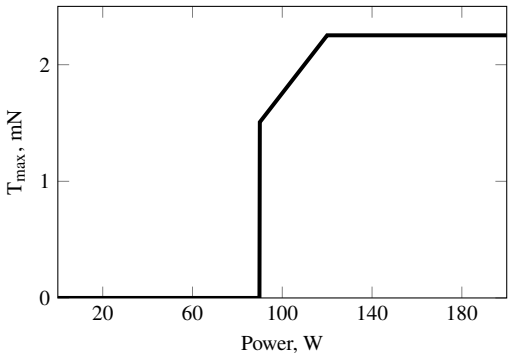
Remark 1. *If a good (e.g., close to feasible) initial guess is provided, it is straightforward to select an initial homotopic parameter ε_0 that is closer to zero, i.e., $0 < \varepsilon_0 < 1$, to shorten the homotopy path, and reduce the number of iterations.*

Variable Specific Impulse and Maximum Thrust

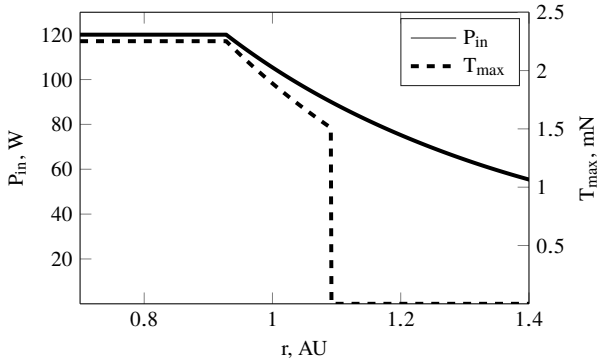
In a real thruster model, the maximum thrust $T_{\max}(P_{\text{in}}(r))$ and specific impulse $I_{\text{sp}}(P_{\text{in}}(r))$ depend on the input power $P_{\text{in}}(r)$, which in turn is a function of the distance r to the Sun. Without loss of generality, the saturation logic of a real thruster is given by [18]

$$T_{\max} = \begin{cases} T_{\max}(P_{\text{in,max}}) & \text{if } P_{\text{in}}(r) > P_{\text{in,max}}\\ T_{\max}(P_{\text{in}}(r)) & \text{if } P_{\text{in,min}} \leq P_{\text{in}}(r) \leq P_{\text{in,max}}\\ 0 & \text{if } P_{\text{in}}(r) < P_{\text{in,min}}\end{cases} \tag{28}$$

where $P_{\text{in,min}}$ and $P_{\text{in,max}}$ denote the minimum and maximum input power, respectively. Whenever the input power is lower than some threshold $P_{\text{in,min}}$, no thrust is available. Moreover, $T_{\max}$ cannot exceed the threshold $T_{\max}(P_{\text{in,max}})$. These functions are often described by nth order polynomials. Typical $P_{\text{in}}(r)$ and $T_{\max}(P_{\text{in}}(r))$ curves are shown in Fig. 2. Such a non-smooth representation increases complexity and results in additional nonlinear constraints. Recalling the dynamics in



(a) $P_{\text{in}}(r)$ vs. $T_{\max}(P_{\text{in}}(r))$ curve.



(b) $P_{\text{in}}(r)$ and $T_{\max}(r)$ curves.

Fig. 2: Typical input power and maximum thrust curves.

(9), it becomes evident that the previously constant Jacobian matrix $\mathbf{B}$ now depends on r, i.e., we have $\mathbf{B}(\mathbf{x})$ for the variable I_{sp} case. As a consequence, linearizing the dynamics would yield an additional term corresponding to the previous control history $\bar{\mathbf{u}}$. This dependence on $\bar{\mathbf{u}}$ is undesirable as it may deteriorate convergence [36]. Therefore, we propose the approximation $\mathbf{f}(\mathbf{x},\mathbf{u}) \approx \mathbf{p}(\mathbf{x}) + \mathbf{B}(\bar{\mathbf{x}})\mathbf{u}$ which results in a partial linearization of the dynamics at $\bar{\mathbf{x}}$:

$$\mathbf{f}(\mathbf{x},\mathbf{u}) \approx \mathbf{A}(\bar{\mathbf{x}})\mathbf{x} + \mathbf{B}(\bar{\mathbf{x}})\mathbf{u} + \mathbf{q}(\bar{\mathbf{x}}) \tag{29}$$

A similar approach is used for the upper bound of the thrust magnitude in (10) where $T_{\max}(r)$ becomes a function of r. Instead of linearizing the constraint, $T_{\max}(r)$ is approximated by

$$T_{\max}(r) \approx T_{\max}(\bar{r}) \tag{30}$$

where $\bar{r}$ denotes the radius of the previous iteration. As the algorithm progresses, $\|r - r^*\| \to 0$ and $\|r - \bar{r}\| \to 0$ due to the shrinking trust-region size, r^* being the radius at the optimal solution. Therefore, $r^* \approx \bar{r}$ when the algorithm determines an optimal solution. Our simulations suggest that this approach is often advantageous in terms of convergence.

Solving the optimal control problem with a real thruster model directly can be challenging, especially for large $P_{\text{in,min}}$ or when the spacecraft is far away from the Sun. An example is the transfer to asteroid Dionysus whose distance to the Sun can be larger than 3 AU, hence resulting in a low $T_{\max}$. A standard way to improve convergence is to solve the problem with fixed

I_{sp} and $T_{\max}$, and then use the solution as the new initial guess to solve another optimization problem with the real thruster model. This is not ideal due to two reasons: First, the solver might not be able to find a solution even when the solution of the constant I_{sp} and $T_{\max}$ case is provided as the initial guess for the real model. This is especially true if the constant and real thruster models differ a lot (e.g., in the Dionysus transfer), or when the input power range is small. Secondly, such an approach would require solving another optimization problem with a potentially significantly d ifferent t hruster c urve, and therefore more computational effort and time. In this paper, two improvements are proposed:

1) A smooth representation of the input power P_{in} over $T_{\max}$ curve.
2) A homotopy from higher thrust levels to the real thruster model to enhance convergence.

Using the general hyperbolic tangent function [37], we introduce the following smooth $T_{\max}$ function of the original relationship in (28):

$$T_{\max} = \begin{cases} T_{\max}(P_{\text{in,max}}) & \text{if } P_{\text{in}}(r) > P_{\text{in,max}} + \rho \\ g(P_{\text{in}}(r)) & \text{if } P_{\text{in,max}} - \rho \leq P_{\text{in}}(r) \leq P_{\text{in,max}} + \rho \\ \xi\, T_{\max}(P_{\text{in}}(r)) & \text{if } P_{\text{in}}(r) < P_{\text{in,max}} - \rho \end{cases} \tag{31}$$

with some function $g(P_{\text{in}}(r))$ that depends on the order of the polynomial of the P_{in}-$T_{\max}$ curve, a smoothing parameter $\rho > 0$, and

$$\xi = \frac{1}{2}\left[\tanh\left(\frac{P_{\text{in}}(r) - P_{\text{in,min}}}{\rho}\right) + 1\right] \tag{32}$$

If $\rho = 0$, this reduces to

$$T_{\max} = T_{\max}(P_{\text{in,max}}) \tag{33}$$

For example, given the following thruster model

$$T_{\max}(P_{\text{in}}) = a_0 + a_1 P_{\text{in}} \tag{34a}$$
$$I_{\text{sp}}(P_{\text{in}}) = b_0 + b_1 P_{\text{in}} \tag{34b}$$
$$P_{\text{in}}(r) = \frac{1}{r^2} \tag{34c}$$

the piecewise defined functions in (28) are smoothly connected using a second-order polynomial:

$$g(P_{\text{in}}(r)) = c_0 + c_1\, P_{\text{in}} + c_2\, P_{\text{in}}^2 \tag{35}$$

where

$$c_0 = \frac{a_1\, P_{\text{in,max}}^2 + 2\, a_1\, P_{\text{in,max}}\, \rho - a_1\, \rho^2 + 4\, a_0 \rho}{4\, \rho} \tag{36a}$$
$$c_1 = \frac{a_1\, (P_{\text{in,max}} + \rho)}{2\, \rho} \tag{36b}$$
$$c_2 = -\frac{a_1}{4\, \rho} \tag{36c}$$

Higher orders can be included in a similar way by increasing the degree of the polynomial $g(P_{\text{in}}(r))$.

The homotopy from higher to lower thrust levels is achieved by increasing $P_{\text{in,max}}$ (see Fig. 3a) and by shifting the P_{in}-$T_{\max}$ curve along the x-axis (see Fig. 3c), hence increasing or decreasing $T_{\max}$ for a given input power which is a function of the distance to the Sun. The effect of ρ is illustrated in Figs. 3b and 3c where the $\tanh$ function smoothes the abrupt changes. Figure 3c shows the combination of shifting the P_{in}-$T_{\max}$ curve and changing the smoothing parameter. Using the mapping in Eqs. (37) and (38), the different values of ε in Fig. 3 refer to the transition from a simpler ($\varepsilon = 1$) to the original problem ($\varepsilon = 0$). Changing $P_{\text{in,max}}$, adjusting the smoothing and shifting the P_{in}-$T_{\max}$ curve simultaneously yields the full homotopic approach for the variable $T_{\max}$ case: For larger values of ε (i.e., larger constraint violations), the nominal curve (solid line corresponding to $\varepsilon = 0$ in Fig. 3c) is shifted to the left. Moreover, the maximum input power is increased, therefore increasing the available maximum thrust. At the same time, the curve is made smoother to have no abrupt changes in the thrust values, which is beneficial for convergence. As the algorithm progresses, the constraint violation reduces and consequently, the Power-$T_{\max}$ curve is shifted back towards the original one and the smoothness is reduced. This way, the complexity is gradually increased until the original problem is solved. We define mappings $\mathcal{M}_{\Delta P}$, $\mathcal{M}_{\rho}$, and $\mathcal{M}_{P_{\text{in,max}}}$ that map the shifting of the power curve ΔP, the smoothing ρ, and the change of $P_{\text{in,max}}$ to the homotopic parameter $\varepsilon \in [0, 1]$ and vice versa:

$$\mathcal{M}_{\Delta P} : \varepsilon \longmapsto \Delta P \tag{37a}$$
$$\mathcal{M}_{\rho} : \varepsilon \longmapsto \rho \tag{37b}$$
$$\mathcal{M}_{P_{\text{in,max}}} : \varepsilon \longmapsto P_{\text{in,max}} \tag{37c}$$

We use the following linear relationship:

$$\mathcal{M}_{\Delta P} : \varepsilon \longmapsto \Delta P_{\min} + \frac{\Delta P_{\max} - \Delta P_{\min}}{\varepsilon_{\max} - \varepsilon_{\min}}\, (\varepsilon - \varepsilon_{\min}) \tag{38a}$$
$$\mathcal{M}_{\rho} : \varepsilon \longmapsto \rho_{\min} + \frac{\rho_{\max} - \rho_{\min}}{\varepsilon_{\max} - \varepsilon_{\min}}\, (\varepsilon - \varepsilon_{\min}) \tag{38b}$$
$$\mathcal{M}_{P_{\text{in,max}}} : \varepsilon \longmapsto P_{\text{in,max}} + \frac{P_{\max} - P_{\text{in,max}}}{\varepsilon_{\max} - \varepsilon_{\min}}\, (\varepsilon - \varepsilon_{\min}) \tag{38c}$$

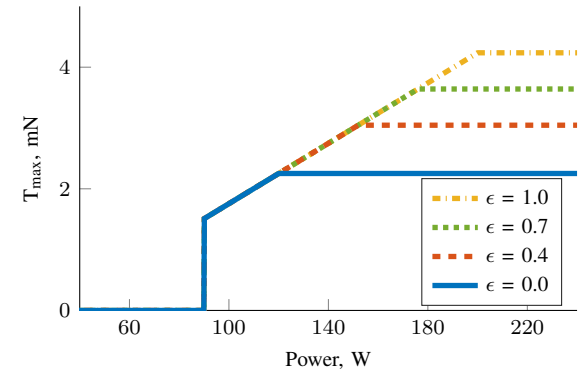
with problem dependent constants $\Delta P_{\min}$, $\Delta P_{\max}$, $\rho_{\min}$, $\rho_{\max}$, and $P_{\max}$. $\varepsilon_{\min}$ and $\varepsilon_{\max}$ are the lower and upper bound of the homotopic parameter, in general 0 and 1. Often, $\Delta P_{\min} = 0$, and $\Delta P_{\max}$ can be chosen such that $T_{\max} = T_{\max}(P_{\text{in,max}})$ over a sufficiently large input power range. $\rho_{\min}$ and $\rho_{\max}$ define the minimum and maximum smoothing parameter, respectively. Generally, values of the order 10^0 for $\rho_{\max}$ and values close to zero for $\rho_{\min}$ ensure a sufficient smoothing. $P_{\max}$ denotes the maximum allowable input power that is to be chosen such that $P_{\max} \geq P_{\text{in,max}}$. The inverse mappings $\mathcal{M}_{\Delta P}^{-1}$, $\mathcal{M}_{\rho}^{-1}$, and $\mathcal{M}_{P_{\text{in,max}}}^{-1}$ are defined accordingly.

In case the same homotopy path h is used for n-body dynamics, SRP, and variable maximum thrust, it suffices to define only a single parameter ε that controls every homotopy in the optimization simultaneously. If different paths are used, one can define

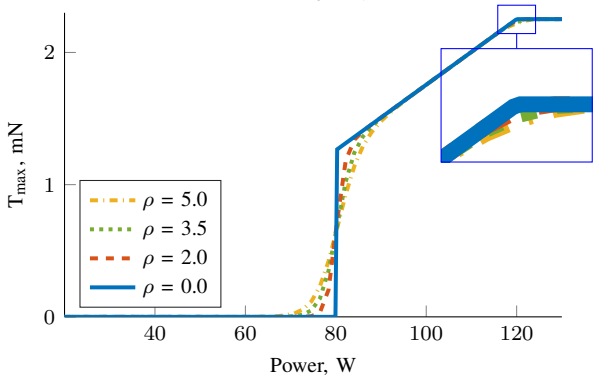
$$\varepsilon := \max\left(\varepsilon_{\text{nbody}}, \varepsilon_{\text{SRP}}, \varepsilon_{T_{\max}}\right) \tag{39}$$

to keep the number of parameters and complexity of the algorithm at a minimum.

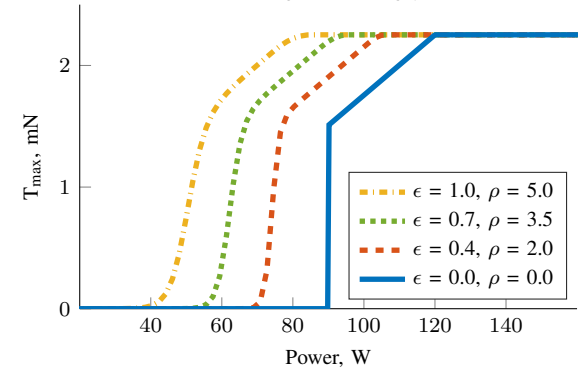
Remark 2. *As in any direct method, the control constraints are only satisfied at the nodes. Due to the linear interpolation*



(a) Increasing $P_{\text{in,max}}$.



(b) Increasing smoothing ρ.



(c) Shifting P_{in}-T_{max} curve.

Fig. 3: Overview of homotopy techniques for variable T_{max}.

of the controls in FOH between the nodes, there will be a minor mismatch between the interpolated controls and the available maximum thrust if the $T_{max}(t)$ curve is nonlinear. This error, however, can be made arbitrarily small if more nodes are added in the corresponding segments.

Adaptive Trust-Region Method

The trust-region method in this work uses the ratio $\varrho^{(k)}$ of the actual and predicted cost reductions at iteration k to measure the progress and decide whether to shrink or increase the trust-region size R. The actual and predicted cost are calculated using the nonlinear and convexified constraints, respectively [13]. If $\varrho^{(k)} \geq \varrho_0$, the current step is accepted

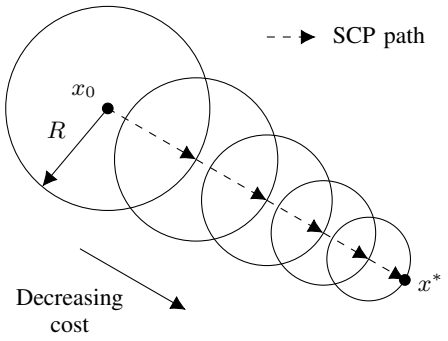


Fig. 4: Change of the trust-region radius and typical SCP path without a homotopic approach.

and R is updated [38], [39]:

$$R^{(k+1)} = \begin{cases} R^{(k)}/\alpha & \text{if} \quad \varrho_0 \leq \varrho^{(k)} < \varrho_1 \\ R^{(k)} & \text{if} \quad \varrho_1 \leq \varrho^{(k)} < \varrho_2 \\ \beta\, R^{(k)} & \text{if} \quad \varrho^{(k)} \geq \varrho_2 \end{cases} \tag{40}$$

where $0 < \varrho_0 < \varrho_1 < \varrho_2 < 1$ are user-defined parameters. $\alpha > 1$ and $\beta > 1$ define the shrinking and growing rates of R, respectively; they can be constant or dynamically adjusted at each iteration (see e.g. [27]).

This procedure often works well in practice. However, when a homotopic approach is included, such methods can deteriorate convergence if not modified appropriately. If no homotopy is considered, the algorithm updates the trust region such that the trajectory is driven towards the optimal solution x^* as illustrated in Fig. 4. If a homotopy is included, the algorithm progresses in a similar way as long as ε does not change. We recall that the previous solution is used as the new initial guess. When ε_j changes to ε_{j+1}, the constraint violation increases because the constraints also (slightly) change. Therefore, the previously optimal trajectory will not satisfy all constraints anymore. In addition, the new optimal solution $x^*(\varepsilon_{j+1})$ will also be different. As a consequence, it might happen that the new optimal path associated with ε_{j+1} lies outside of the trust region defined by R. Even though the algorithm will still often be able to reduce the constraint violation in the following iterations by reducing the trust-region size, the followed path is different from the optimal one. This can eventually result in a non-feasible (and non-optimal) solution as shown in Fig. 5. This is especially true for difficult problems. We propose an adaptive trust-region update to overcome this issue. Instead of using the same trust-region size when ε_j changes to ε_{j+1}, R is increased to expand the feasible region. Thus, the algorithm is able to reach the new optimal solution $x^*(\varepsilon_{j+1})$ (see Fig. 6). Even though expanding R to some large, fixed value might often be sufficient to reach the new optimal path, this approach would require a considerable amount of additional iterations to reduce the trust-region radius again. Moreover, finding an appropriate value for $R(\varepsilon_{j+1})$ is not intuitive. As the constraint violations c_{viol} change when the homotopic parameter is updated, it is reasonable to define an update mechanism based on the difference c_{viol} of the constraint violations. Figure 7 shows the ratio $\sigma = R(\varepsilon_{j+1})/R(\varepsilon_j) = R_{\text{new}}/R_{\text{old}}$ over $\Delta c_{\text{viol}} = \log_{10}(\|\mathbf{c}_{\text{viol}}(\varepsilon_{j+1})\|_1) - \log_{10}(\|\mathbf{c}_{\text{viol}}(\varepsilon_j)\|_1)$ for

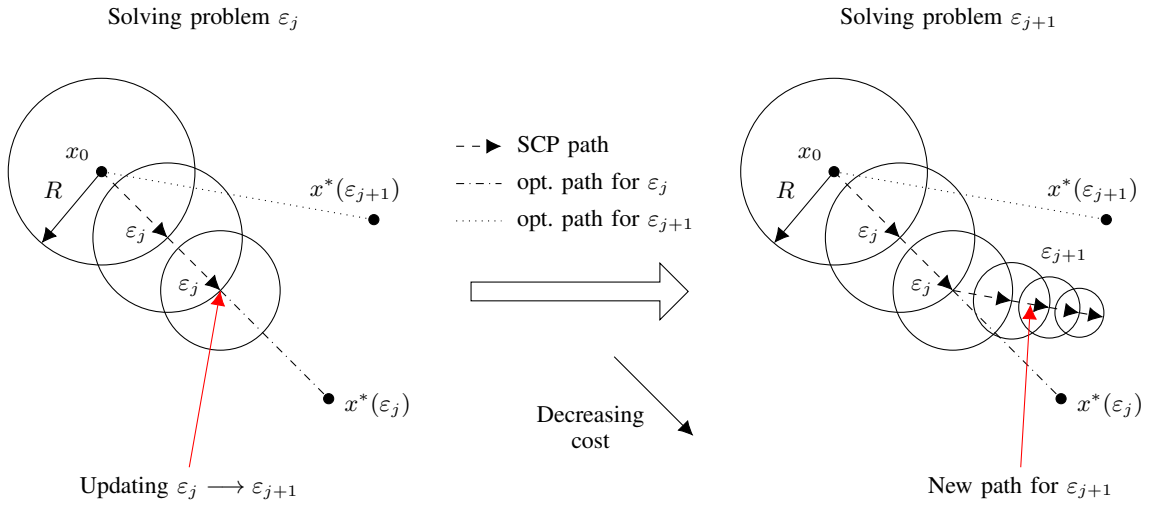


Fig. 5: Change of the trust-region radius and typical SCP path when the trust-region radius is not updated appropriately within a homotopic approach.

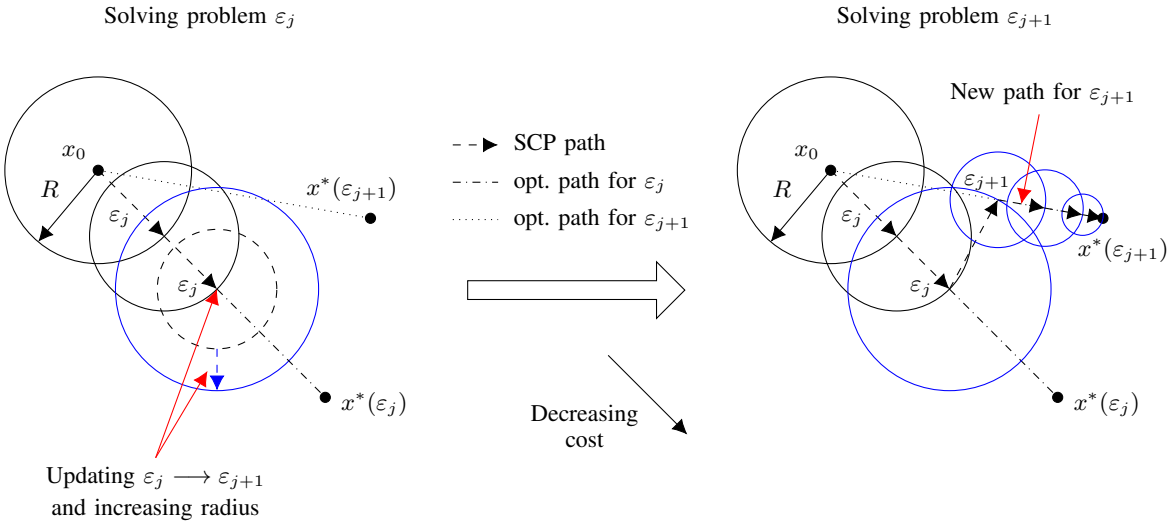


Fig. 6: Change of the trust-region radius and typical SCP path when the trust-region radius is updated appropriately within a homotopic approach.

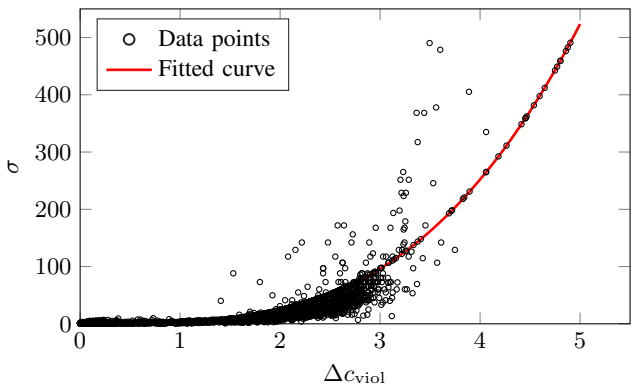


Fig. 7: Relationship between the constraint violation and ratio of new and old trust-region radii.

several hundreds of simulations with different targets and initial guesses, $\mathbf{c}_{\text{viol}}$ being a vector that contains the violations of all constraints. Note that we seek the maximum trust-region radius that is accepted after the homotopic parameter is updated to ensure the largest feasible region possible; this value is denoted by $R(\varepsilon_{j+1})$. We propose to fit the data points using the following equation and a nonlinear least squares approach:

$$\sigma = \frac{R(\varepsilon_{j+1})}{R(\varepsilon_j)} = C_0 + C_1 \left(\Delta c_{\text{viol}} - C_2\right)^n \tag{41}$$

where C_i $(i = 0, 1, 2)$ and n are the coefficients and exponent, respectively. In this work, we use $C_0 = 1.8186$, $C_1 = 5.3518$, $C_1 = 0.3974$, and $n = 3$. Naturally, different approaches can be used to define the relationship between the constraint violation and trust-region update. For instance, a more conservative fitting method would result in larger trust-region sizes, and

consequently, a larger feasible region, but it would also lead to more iterations. Nevertheless, we chose to use a nonlinear least squares approach to fit the data, as we found it to be a good compromise between convergence and the number of iterations. The new trust-region radius $R(\varepsilon_{j+1})$ is then calculated by the relationship

$$R(\varepsilon_{j+1}) = \gamma\,\sigma\,R(\varepsilon_j) \tag{42}$$

with some factor $\gamma > 0$ to increase the flexibility of the update mechanism. Although one could simply set $R(\varepsilon_{j+1})$ equal to the initial trust-region size or some other large value, the proposed approach requires significantly fewer iterations as the factor σ is not fixed, but rather adjusted dynamically based on the current information at each iteration. Even though trust-region based SCP algorithms may suffer from oscillations in the trust-region radius under certain conditions [40], we did not experience this phenomenon in our simulations.

Remark 3. *Note that other existing trust-region algorithms, such as state-dependent methods [41] or higher-order methods [42], [43], can potentially be used within the presented embedded homotopic approach if modified accordingly to account for the required expansion when the homotopic parameter changes.*

B. Flowchart of the Algorithm

The flowchart of the homotopic approach is illustrated in Fig. 8. After solving the convexified problem, the trust-region radius is updated as in the standard SCP method (see (40)). The homotopic parameter is then adjusted based on the predefined homotopic path. Additional parameters $\Delta\varepsilon_{\text{min}}$ and $\Delta\varepsilon_{\text{max}}$ are introduced that denote the minimum and maximum allowable changes of ε, respectively. This ensures that the step size is neither too small (which would require more iterations) nor too large (which might result in non-convergence). Furthermore, the ε domain is divided into two regions with different values of $\Delta\varepsilon_{\text{min}}$ and $\Delta\varepsilon_{\text{max}}$. In particular, we have

1) $\Delta\varepsilon_{\text{min,1}}$, $\Delta\varepsilon_{\text{max,1}}$ if $\varepsilon \in [y_1, 1]$
2) $\Delta\varepsilon_{\text{min,2}}$, $\Delta\varepsilon_{\text{max,2}}$ if $\varepsilon \in [0, y_1)$

where $y_1 \in (0, 1)$ defines the switching between the different minimum and maximum step sizes. In this work, we use $y_1 = 0.7$ and $\Delta\varepsilon_{\text{min,1}} = 0.0$, $\Delta\varepsilon_{\text{max,1}} = 0.025$, and $\Delta\varepsilon_{\text{min,2}} = 0.05$, $\Delta\varepsilon_{\text{max,2}} = 0.10$. Based on our simulations, the most critical part of the algorithm is during the first iterations when $\varepsilon = 1$ and the linear constraint violations are large. This is often the case if the initial guess is poor. Therefore, it is convenient to use small values for $\Delta\varepsilon_{\text{min}}$ and ε_{max} at the beginning. This means that the optimal solution of the problem with the updated ε changes only slightly, and hence, the solver is more likely to find a solution. Once the algorithm makes progress, $\Delta\varepsilon_{\text{min}}$ and $\Delta\varepsilon_{\text{max}}$ are increased to avoid an excessive number of iterations. If the homotopic parameter changes, the constraint violations are updated and the trust-region size is expanded based on (42). The process continues until an optimal solution is found. Although an in-depth theoretical analysis is not the focus of this paper, our numerical simulations in Section V demonstrate the efficacy of the approach and support our rationale for selecting the parameters.

V. NUMERICAL SIMULATIONS

We compare our method with a standard homotopic approach to demonstrate the benefits. Moreover, the performance of the developed homotopic approach in terms of success rate, CPU time, and obtained final mass is compared with a standard SCP method without any homotopy [12] and the state-of-the-art optimal control software GPOPS-II [44] in combination with the Sparse Nonlinear Optimizer (SNOPT) [45]. All simulations are performed in MATLAB on an Intel Core i7-8565 1.80 GHz Laptop with four cores and 16 GB of RAM. The numerical integration of (9) and (18b)–(18d) within SCP is performed using a mex function. The second-order cone program in (11) is solved using the open-source Embedded Conic Solver (ECOS) [46]. In contrast, GPOPS-II solves the full nonlinear program directly. We compute fuel-optimal trajectories from the Sun-Earth Lagrange point L_2 (SEL_2) to the asteroids 2000 SG344 and Dionysus. The algorithms stop if the maximum constraint violation and change in the objective function are smaller than the feasibility ε_c and optimality ε_J tolerances, respectively. Relevant parameters for the transfers and algorithms are given in Tables I and II. Physical constants are given in Table III, and parameters related to the homotopic approach in Table IV.

As described in Section IV, n-body dynamics, solar radiation pressure, and a real thruster model with variable specific impulse and maximum thrust are considered. In particular, the following thruster model is used for the transfer to 2000 SG344 [18]:

$$T_{\text{max}}(P_{\text{in}}) = a_0 + a_1 P_{\text{in}} + a_2 P_{\text{in}}^2 + a_3 P_{\text{in}}^3 + a_4 P_{\text{in}}^4 \tag{43a}$$
$$I_{\text{sp}}(P_{\text{in}}) = b_0 + b_1 P_{\text{in}} + b_2 P_{\text{in}}^2 + b_3 P_{\text{in}}^3 + b_4 P_{\text{in}}^4 \tag{43b}$$
$$P_{\text{in}}(r) = c_0 + c_1 r + c_2 r^2 + c_3 r^3 + c_4 r^4 \tag{43c}$$

with $a_0 = -0.7253\,\text{mN}$, $a_1 = 0.024\,81\,\text{mN}\,\text{W}^{-1}$, $a_2 = a_3 = a_4 = 0$, $b_0 = 2652$ s, $b_1 = -18.123\,\text{s}\,\text{W}^{-1}$, $b_2 = 0.3887\,\text{s}\,\text{W}^{-2}$, $b_3 = -0.001\,74\,\text{s}\,\text{W}^{-3}$, $b_4 = 0$, and $c_0 = 840.11\,\text{W}$, $c_1 = -1754.3\,\text{W}\,\text{AU}^{-1}$, $c_2 = 1625.01\,\text{W}\,\text{AU}^{-2}$, $c_3 = -739.87\,\text{W}\,\text{AU}^{-3}$, $c_4 = 134.45\,\text{W}\,\text{AU}^{-4}$, and r in AU. For Dionysus, we use a modified version of the model presented in [15]:

$$T_{\text{max}}(P_{\text{in}}) = \tilde{a}_0 + \tilde{a}_1 P_{\text{in}} \tag{44a}$$
$$I_{\text{sp}}(P_{\text{in}}) = I_{\text{sp,max}} \tag{44b}$$
$$P_{\text{in}}(r) = \frac{1}{r^2} \tag{44c}$$

where $\tilde{a}_0 = 0.1069\,\text{N}$, $\tilde{a}_1 = 3.9307 \times 10^{-4}\,\text{N}\,\text{W}^{-1}$, and r in AU.

A. Comparison With Conventional and Adaptive Homotopic Approaches

In this subsection, the proposed homotopic approach is compared with a conventional homotopy method and a recently developed adaptive homotopy from the literature [22].

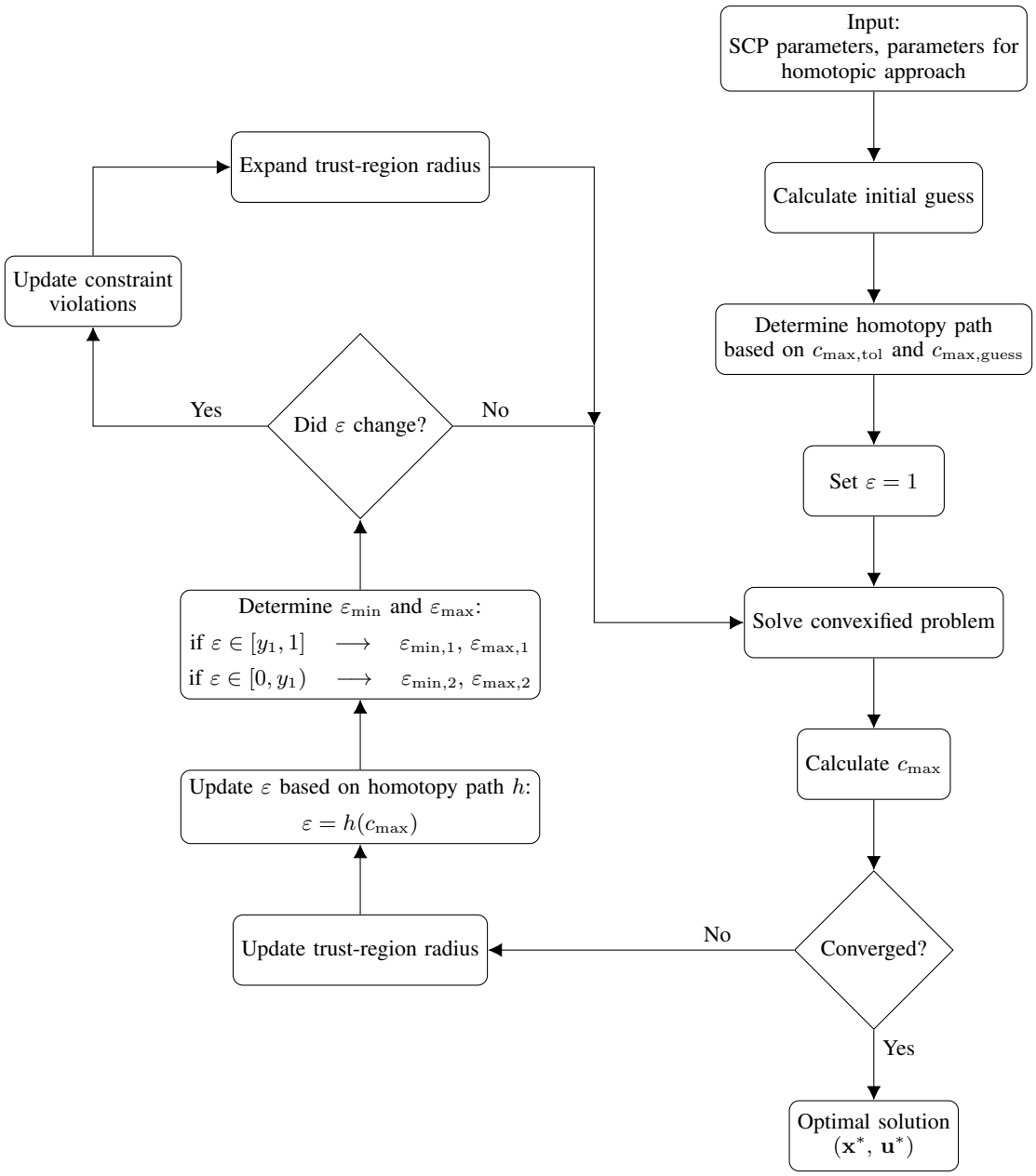


Fig. 8: Flowchart of the homotopic approach.

In the conventional method, a fixed step size of the homotopic parameter is employed. In the adaptive method, the step size is updated based on the progress of the algorithm and previous iterations. Details can be found in [22]. Moreover, different trust-region methods are discussed and assessed for use in the homotopic approach. Existing trust-region methods in the literature often fail when applied without modifications in the embedded approach due to the lack of expansion of the feasible region. Therefore, we only consider different trust-region methods in the adaptive homotopy that is compared with the proposed approach. In particular, we consider an adaptive hard-trust region method that has proven effective for low-thrust trajectory optimization [39]. In hard trust-region methods, the trust-region constraint is explicitly considered as a constraint, and the size is updated based on some heuristic. In contrast, the trust-region radius becomes an optimization parameter in soft trust-region strategies and is penalized in the objective function [48]. This makes such methods potentially appealing for a homotopic approach because the solver can choose the size, avoiding manual heuristic expansion of the trust-region radius. For the comparison, we consider the recently developed hybrid-order soft trust-region method in [43]. Preliminary simulations were performed to find the best parameters for each method.

Figure 9a compares the proposed homotopic approach with the conventional homotopy using a hard trust region (TR), the adaptive approach using a hard TR, and the adaptive approach combined with a second-order soft TR for the asteroid 2000 SG344 transfer. In the conventional and adaptive homotopy strategies with hard TRs, approximately twice as many itera-

TABLE I: Simulation values for the transfers from SEL_2 to the asteroids 2000 SG344 and Dionysus [18], [47].

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 position $\mathbf{r}_0$, AU	$[-0.701\,860\,65,\ 0.706\,232\,44,\ -3.511\,15\times10^{-5}]^\top$	$[-0.023\,941\,014,\ 0.993\,253\,72,\ -3.027\,63\times10^{-5}]^\top$
Initial velocity $\mathbf{v}_0$, VU	$[-0.732\,969\,49,\ -0.715\,904\,85,\ 4.402\,45\times10^{-5}]^\top$	$[-1.026\,373\,47,\ -0.028\,097\,21,\ 1.985\,38\times10^{-6}]^\top$
Initial mass m_0, kg	22.6	4000
Final position $\mathbf{r}_f$, AU	$[0.418\,067\,95,\ 0.828\,971\,14,\ -0.001\,433\,82]^\top$	$[-2.040\,617\,82,\ 2.051\,791\,30,\ 0.554\,288\,95]^\top$
Final velocity $\mathbf{v}_f$, VU	$[-0.969\,903\,32,\ 0.436\,302\,20,\ -0.001\,233\,81]^\top$	$[-0.142\,319\,32,\ -0.451\,088\,00,\ 0.018\,946\,90]^\top$
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
Time of flight t_f, days	700	3534

TABLE II: Parameters of the algorithms.

Parameter	Value
Feasibility tol. ε_c	10^{-6}
Optimality tol. ε_J	10^{-4}, 10^{-5}
Max. iterations	1500, 3000
λ_ν, λ_η	10.0, 10.0
$r_{\min}$, AU	0.03
ϱ_0, ϱ_1, ϱ_2	0.01, 0.25, 0.85
α, β	1.5, 1.5

TABLE III: Physical constants in all simulations.

Parameter	Value
Gravitational const. μ	$1.327\,124\,4\times10^{11}\,\text{km}^3\,\text{s}^{-2}$
Gravitational accel. g_0	$9.806\,65\times10^{-3}\,\text{km}\,\text{s}^{-2}$
Length unit AU	$1.495\,978\,707\times10^{8}\,\text{km}$
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 IV: Homotopy parameters for the transfers from SEL_2 to the asteroids 2000 SG344 and Dionysus.

Parameter	SEL_2 - 2000 SG344	SEL_2 - Dionysus
δ	2.0	
n, C_0, C_1, C_2	3, 1.8186, 5.3518, 0.3974	
γ	1.0	
y_1	0.7	
$\Delta\varepsilon_{\text{min},1}$, $\Delta\varepsilon_{\text{max},1}$	0.0, 0.025	
$\Delta\varepsilon_{\text{min},2}$, $\Delta\varepsilon_{\text{max},2}$	0.05, 0.10	
$\Delta P_{\min}$, W	0	0
$\Delta P_{\max}$, W	180	1500
$\rho_{\min}$	10^{-4}	10^{-4}
$\rho_{\max}$	5.0	5.0
$P_{\max}$, W	250	2000

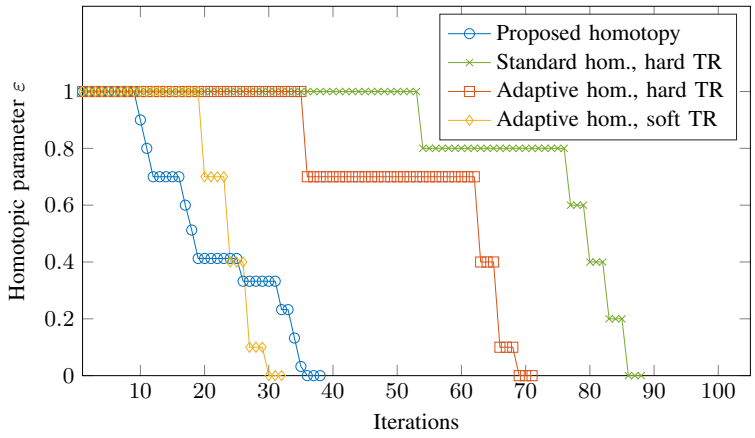
tions are typically needed to find an optimal solution compared to our proposed approach. As the algorithm progresses, the constraint violation decreases, enabling the homotopic parameter to be reduced more rapidly in our approach compared to other methods. With regard to the soft TR, the number of iterations is often similar to that obtained with the proposed approach for the simple 2000 SG344 transfer. The main advantage of the soft TR is that the solver can choose the most appropriate value of the trust region, resulting in fewer iterations than in the hard TR case. The parameters of our approach are selected conservatively to enhance convergence, which sometimes requires more iterations than the soft TR approach. However, as a consequence, the success rate of the proposed method is slightly higher.

Regarding the more complex transfer to asteroid Dionysus, the soft TR often struggles to converge when the homotopic parameter changes. It is therefore omitted in Fig. 9b. Although the conventional and adaptive methods frequently find optimal solutions, they require significantly more iterations than the proposed approach (see Fig. 9b). This confirms the efficacy of the embedded homotopy for complex low-thrust transfers.

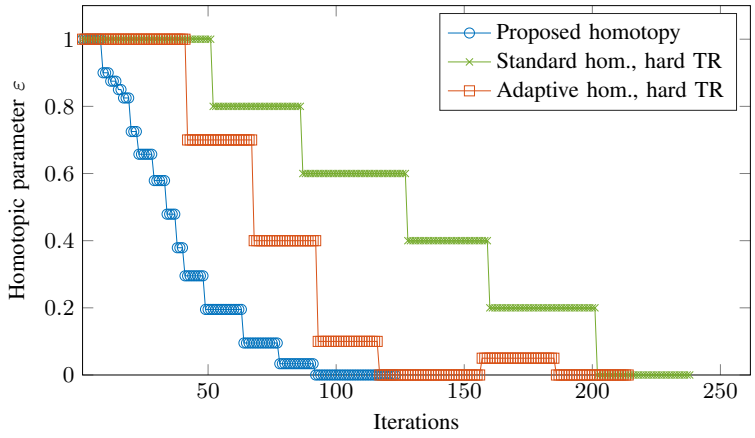
B. Assessment of Homotopic Approach

Different numbers of nodes and optimality tolerances are considered for GPOPS-II to account for potential discrepancies in the performance if an inappropriate value is selected. The problem is discretized using 5 and 10 nodes per segment for each transfer. 30 and 15 segments are used for 2000 SG344, respectively, and 50 and 25 segments for Dionysus, respectively. Hence, the total number of nodes is 150 (2000 SG344) and 250 (Dionysus), respectively. The same number of points is chosen for the first-order-hold discretization method. Two different methods are used to generate the initial guesses:

1) A simple perturbed cubic interpolation that results in infeasible trajectories that neither satisfy the dynamical nor the endpoint constraints [39]. The number of revolutions of the initial guesses ranges from 1.6 to 2.6 for 2000 SG344, and 4 to 7 for Dionysus. Given a step



(a) Evolution of ε for 2000 SG344.



(b) Evolution of ε for Dionysus.

Fig. 9: Typical evolution of the homotopic parameter ε for the proposed, conventional, and adaptive homotopic approaches in transfers to 2000 SG344 and Dionysus.

size of 0.01, the total number of simulations is therefore 101 (2000 SG344) and 301 (Dionysus), respectively. The controls are set to zero.

2) Propagation of the two-body dynamics without SRP and with constant tangential thrust. The thrust magnitude for each initial guess is varied from 0 to $T_{\max}$ (2000 SG344) and 0 to $0.5\,T_{\max}$ (Dionysus), respectively. This results in dynamically feasible trajectories, but the endpoints are far from the target positions.

The selected transfers to the asteroids 2000 SG344 (short time of flight, few revolutions, small changes of orbital elements) and Dionysus (long time of flight, several revolutions, large changes of orbital elements), along with the the different initial guesses and numbers of revolutions, cover a wide range of cases. Therefore, we believe that our examples serve as a good indicator of the performance of the proposed homotopic approach for other transfers with similar characteristics. An overview of the number of simulations and initial guesses is given in Table V. The results are shown in Figs. 10 and 11. SCP is the standard SCP method without homotopy, SCP_H the homotopic approach. The notation GPOPS_i^k refers to the number of nodes $i \in \{5, 10\}$ per segment, and $k \in \{10^{-4}, 10^{-5}\}$ to the optimality tolerance for GPOPS-II. Even though we use 10^{-4} for SCP, selecting a smaller value would not change the results as feasibility is the main factor for SCP.

TABLE V: Overview of simulations for each target and initial guess.

Initial guess	2000 SG344	Dionysus
Cubic interpolation	101	301
Propagation	101	51

TABLE VI: Number of nodes and segments for GPOPS-II.

Target	Nodes per segment	Segments
2000 SG344	5 and 10	30 and 15
Dionysus	5 and 10	50 and 25

With regard to the 2000 SG344 transfer, the benefit of a homotopic approach becomes clear in Fig. 10a. When using a cubic interpolation to generate the initial guess, the success rate is only $39\,\%$ if the nonlinear problem is solved directly with a SCP method. GPOPS-II achieves a slightly lower rate of approximately $31 - 36\,\%$ regardless of the number of nodes

TABLE VII: Number of iterations (median) for SCP and SCP_H.

Method \ Target	SEL_2 - 2000 SG344 cubic	SEL_2 - 2000 SG344 propagation	SEL_2 - Dionysus cubic	SEL_2 - Dionysus propagation
SCP	19	24	47	47
SCP_H	33	28	105	102

and optimality tolerance. In contrast, SCP_H converges in more than $81\,\%$ of the cases, thus doubling the success rate. As the perturbing accelerations due to the n bodies and SRP are small, the thrust continuation is often the most critical part for convergence. However, there are cases where a continuation of the dynamics is also required to achieve convergence. This is especially true for poor initial guesses where the initial constraint violation is large. Judging from Fig. 10a, GPOPS-II benefits from an initial guess that satisfies the dynamics. Remarkably, considerably more simulations converge successfully for SCP ($67\,\%$) and GPOPS-II (50 – $56\,\%$) if the propagation guess is used. In this case, the convergence of SCP_H increases only slightly to $70\,\%$. The reason is that the initial guesses become very poor for larger thrust magnitudes, and even a homotopic approach is not able to find feasible solutions anymore.

According to Fig. 10b, the CPU times are lowest for the propagation guess for all methods. The standard SCP algorithm requires the least CPU time (median of approximately $9\,\text{s}$), followed by SCP_H that requires a few more iterations (see Table VII) with 15 – $26\,\text{s}$, and GPOPS-II (18 – $54\,\text{s}$). As expected, increasing the number of nodes and decreasing the optimality tolerance result in higher CPU times for GPOPS-II. Remarkably, regardless of the parameters, the error bars and thus variation of the results are substantially larger for GPOPS-II. For example, there are cases where the NLP solver requires more than $120\,\text{s}$ to find an optimal solution. Although the final masses are similar for all methods, the solutions obtained with both SCP algorithms are more consistent and closer to the optimal value compared to GPOPS-II. Furthermore, all methods yield an excellent accuracy as the propagation error (i.e., the difference between the optimized final state and the state obtained when integrating the dynamics with the obtained controls) is of the order $10^0\,\text{km}$ (position) and $10^{-7}\,\text{km}\,\text{s}^{-1}$ (velocity) only.

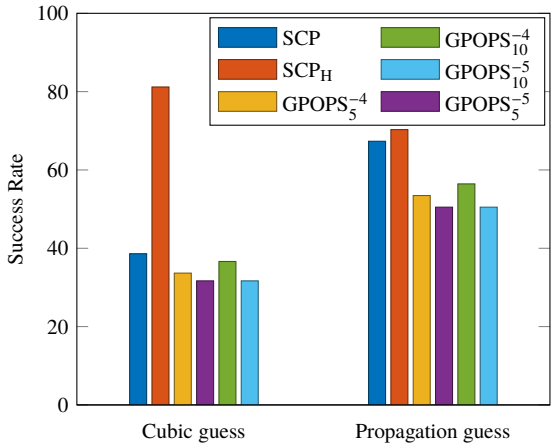
With regard to the Dionysus transfer, SCP converges in only $33\,\%$ (propagation guess) and $46\,\%$ (cubic interpolation guess) of the simulations, therefore performing worst among all methods. Depending on the optimality threshold, GPOPS-II achieves a success rate of 53 – $65\,\%$ ($\varepsilon_J = 10^{-4}$) and 35 – $55\,\%$ ($\varepsilon_J = 10^{-5}$). As opposed to the 2000 SG344 transfer, the cubic guess yields more converged simulations than the propagation guess for GPOPS-II for the Dionysus transfer. Again, the success rate can approximately be doubled if the embedded homotopic approach is used, resulting in more than $95\,\%$ converged simulations. Remarkably, it is evident from Fig. 11b that both SCP methods require up to one order of magnitude fewer seconds to converge than GPOPS-II. Even though the CPU time per iteration is similar for SCP and SCP_H, the latter requires twice as many iterations as shown in Table VII, and therefore, also twice as much CPU time. The CPU time of GPOPS-II can in general be reduced significantly (sometimes up tp $50\,\%$) if the propagation guess is used. The discrepancy in the final mass becomes more significant now as shown in Fig. 11c. Especially if a larger optimality threshold is chosen, the final masses obtained with GPOPS-II are considerably lower compared to both SCP methods. Interestingly, SCP_H does not only improve convergence, but also finds more optimal solutions than the standard SCP. The propagation error is of the order 10^1 to $10^2\,\text{km}$ (position) and 10^{-6} to $10^{-5}\,\text{km}\,\text{s}^{-1}$ (velocity), therefore again being very small.

Typical transfer trajectories and control profiles are shown in Figs. 12 and 13. These plots confirm that SCP and GPOPS-II find different solutions. Moreover, the controls obtained with GPOPS-II are often jittery (see also, e.g., [49]). A much smaller optimality threshold would be required to yield more accurate bang-bang structures. With regard to the Dionysus transfer, it is evident from Fig. 13b that the available maximum thrust decreases considerably over time as the spacecraft gets farther away from the Sun.

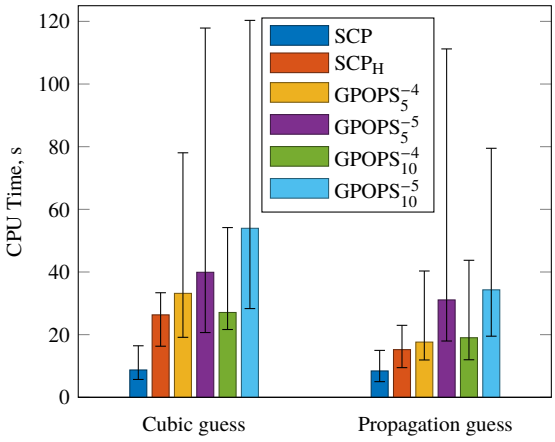
Even though a homotopic approach requires in general more iterations, the simulations confirm that convergence can be improved significantly by gradually increasing the complexity of the dynamical model. Often, only a continuation from higher thrust levels to the real thruster model is sufficient to achieve convergence. The reason is that the additional accelerations due to the perturbing bodies and SRP are small, and therefore, a continuation of the dynamics may not be needed. Yet, for large initial constraint violations (e.g., if the final target state of the initial guess deviates considerably from the desired target state), we observed that a continuation of the dynamics is also required so that the algorithm converges successfully. Due to the rapid speed of SCP, the additional iterations may not be as relevant when compared to methods that usually require more computational effort, such as nonlinear programming solvers. Instead of using SCP only to generate a decent initial guess for an indirect method that makes use of a continuation technique [50], our approach embeds the homotopy directly into SCP. Remarkably, the number of iterations can be reduced considerably with an embedded approach compared to the method in [28] where each optimization problem is solved to full optimality. Our simulations show that methods based on convex optimization can yield a high accuracy even if high-fidelity models are considered.

C. Convergence Analysis

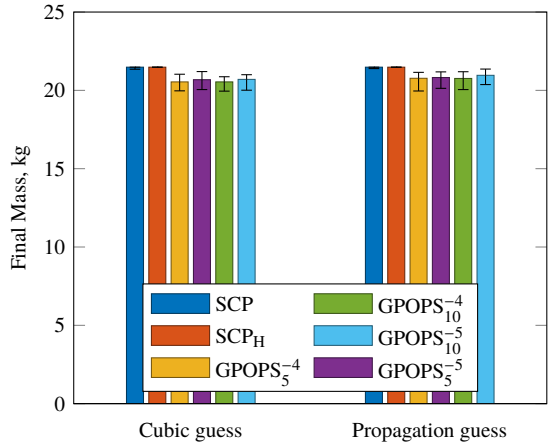
Clearly, a homotopic approach is beneficial when solving more complex problems with convex optimization. Ideally, it would be desirable to have some criterion that defines whether a homotopy is needed, or if a standard SCP method suffices. This way, the additional iterations required by SCP_H could be avoided. The complexity of the problem depends on the problem itself (e.g., number of revolutions, dynamical system), but also on the quality of the initial guess. For example, a highly nonlinear problem can still be solved if a decent initial guess is provided. Our simulations suggest that non-perturbed



(a) Comparison of success rate.

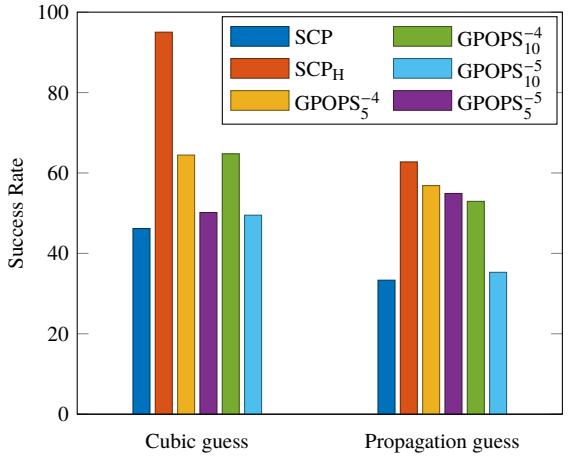


(b) Comparison of CPU time.

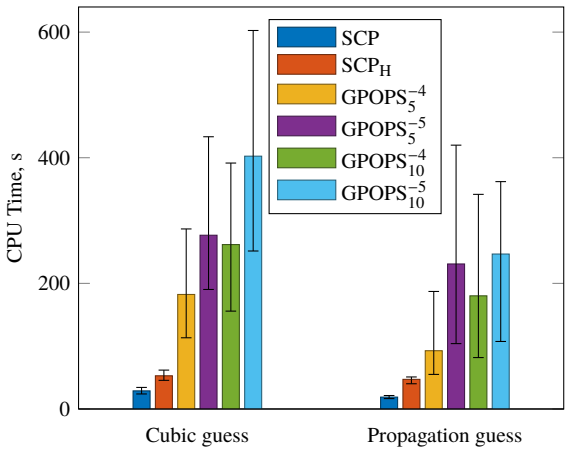


(c) Comparison of final mass.

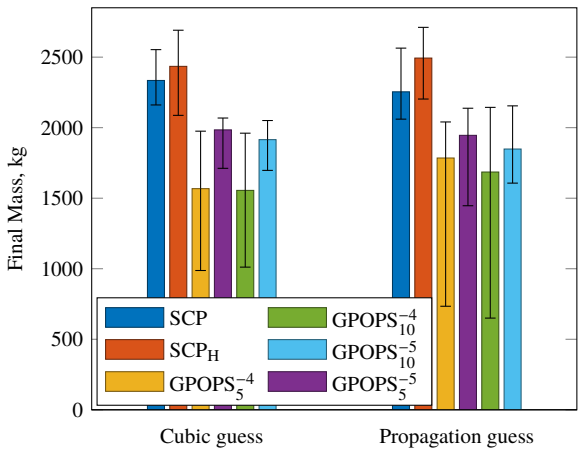
Fig. 10: Comparison of SCP, SCP$_H$ and GPOPS-II in terms of success rate, CPU time, and final mass for the 2000 SG344 transfer. Median values are shown, and the error bars refer to the 80th and 20th percentile of the corresponding quantity.



(a) Comparison of success rate.

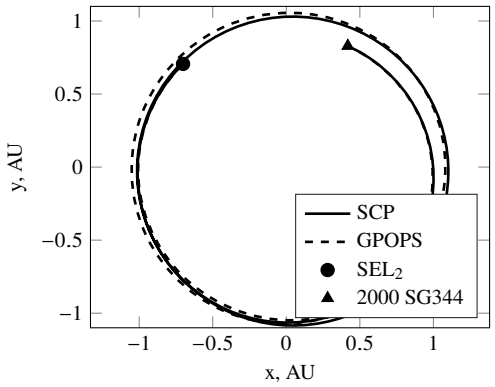


(b) Comparison of CPU time.

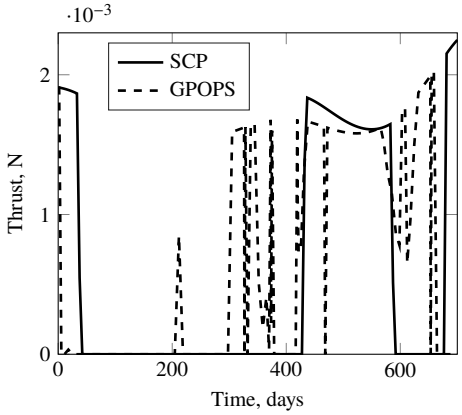


(c) Comparison of final mass.

Fig. 11: Comparison of SCP, SCP$_H$ and GPOPS-II in terms of success rate, CPU time, and final mass for the Dionysus transfer. Median values are shown, and the error bars refer to the 80th and 20th percentile of the corresponding quantity.

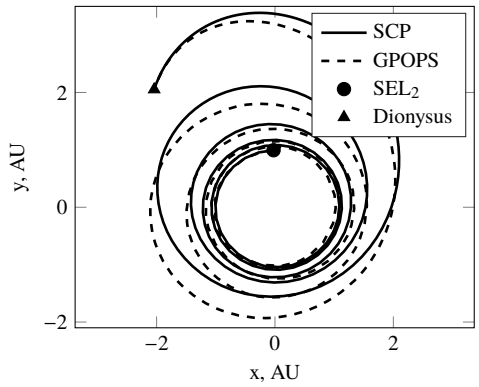


(a) Transfer trajectory.

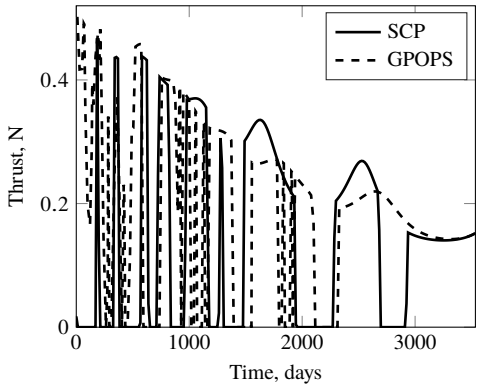


(b) Thrust magnitude.

Fig. 12: Typical transfer and control trajectories for 2000 SG344 obtained with SCP and GPOPS-II.



(a) Transfer trajectory.



(b) Thrust magnitude.

Fig. 13: Typical transfer and control trajectories for Dionysus obtained with SCP and GPOPS-II.

initial guesses where the final boundary conditions are satisfied and the constraint violations of the dynamics are not too large, SCP is in general able to converge successfully even if high-fidelity models are considered. Therefore, we define the normalized maximum constraint violation $\bar{c}_{\text{max}}$ and normalized constraint violation $\bar{c}_{\text{viol}}$ as follows:

$$\bar{c}_{\text{max}} = \frac{\|\mathbf{c}_{\text{viol}}\|_\infty}{\|\mathbf{x}\|_\infty} \tag{45}$$

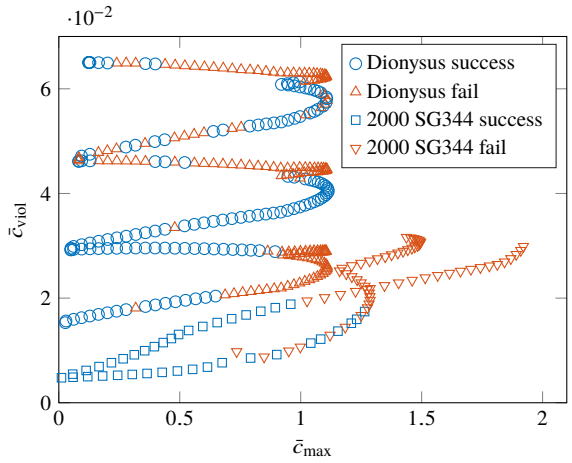
$$\bar{c}_{\text{viol}} = \frac{\|\mathbf{c}_{\text{viol}}\|_1}{\|\mathbf{x}\|_1} \tag{46}$$

where $\mathbf{c}_{\text{viol}}$ is a vector that contains all constraint violations, and $\mathbf{x}$ is the solution vector. The normalization accounts for the strong dependence on the initial and final boundary conditions (e.g., larger values of the coordinates if the target is farther away from the primary body). Figure 14 illustrates $\bar{c}_{\text{max}}$ and $\bar{c}_{\text{viol}}$ for each cubic interpolation guess for SCP and SCP_H. It is apparent that the closer both values are to zero, the more likely it is that SCP converges successfully (see Fig. 14a). Yet, there is no unique pair of values that clearly defines success or failure of the standard algorithm. For example, SCP fails if $(\bar{c}_{\text{max}}, \bar{c}_{\text{viol}}) = (0.9, 0.02)$, but converges again if $\bar{c}_{\text{viol}}$ becomes larger (e.g., 0.045 and 0.055). Still, $\bar{c}_{\text{max}} \geq 1.1$ seems to be the threshold where the standard SCP cannot find optimal solutions anymore regardless of the normalized constraint violation. As shown in Fig. 14b, this value increases to approximately 1.4 if a homotopic approach is used. Moreover, a much higher constraint violation is required so that SCP_H fails. Interestingly, the failure and success cases follow a more regular pattern than in the non-homotopic case. In general, a homotopic approach seems to be useful for a perturbed cubic guess whenever $\bar{c}_{\text{max}} \geq 0.7 \wedge \bar{c}_{\text{viol}} \geq 0.02$. SCP_H seems to fail if $\bar{c}_{\text{max}} \geq 1.4$ or $\bar{c}_{\text{max}} \geq 0.8 \wedge \bar{c}_{\text{viol}} \geq 0.06$.

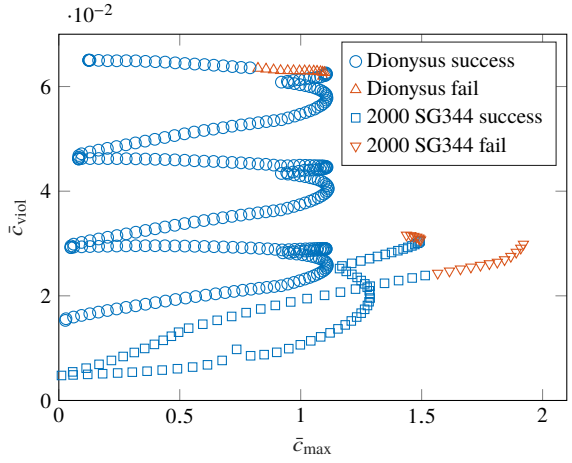
With regard to the propagation guess in Fig. 15, only the transfer to 2000 SG344 shows a pattern. If $\bar{c}_{\text{max}} \geq 1.5 \wedge \bar{c}_{\text{viol}} \geq 0.015$, both algorithms seem to fail (even though there very few other cases where SCP does not converge, see Fig. 15a). For Dionysus, however, there does not seem to be a visible correlation. It therefore may be beneficial to make use of a homotopic approach whenever $\bar{c}_{\text{max}} \geq 0.5$, even though SCP_H will fail at some point due to the large initial error on the final boundary condition.

VI. CONCLUSION

This work combines the sequential convex programming method with an embedded homotopic approach to solve the low-thrust trajectory optimization problem for high-fidelity models. The simulations show that increasing the complexity of the dynamical model only step by step increases convergence significantly. Moreover, considering the progress of the

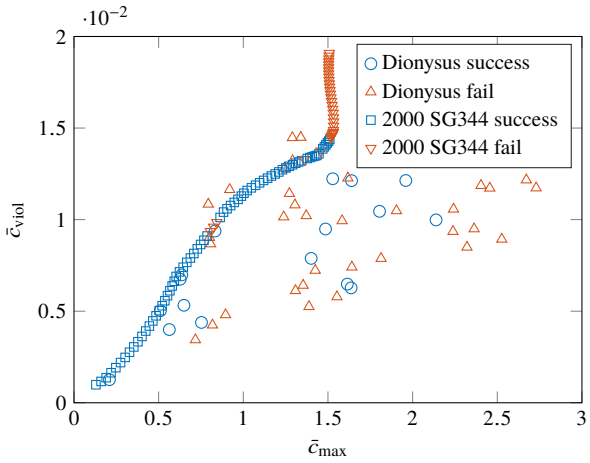


(a) Standard SCP.

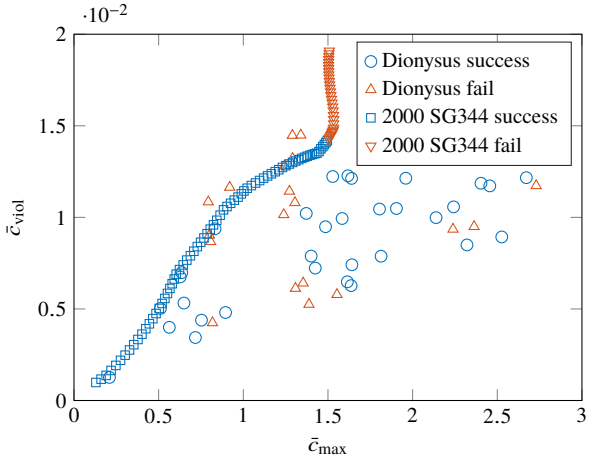


(b) SCP with homotopic approach.

Fig. 14: Overview of the constraint violation of each initial guess generated by cubic interpolation for SCP and SCP_H.



(a) Standard SCP.



(b) SCP with homotopic approach.

Fig. 15: Overview of the constraint violation of each initial guess generated by propagation for SCP and SCP_H.

algorithm to update the homotopic parameter and trust-region radius dynamically can be an effective means to reduce the overall number of iterations. Even though more iterations may be required to reach convergence if a homotopic approach is used, the rapid speed of the proposed method makes it an excellent alternative to nonlinear programming solvers, especially if more complex models are considered.

References

[1] D. Morante, M. Sanjurjo Rivo, and M. Soler, "A survey on low-thrust trajectory optimization approaches," *Aerospace*, vol. 8, no. 3, 2021.

[2] P. Lu, "Introducing computational guidance and control," *J. Guid. Control Dyn.*, vol. 40, no. 2, pp. 193–193, 2017.

[3] M. B. Quadrelli, L. J. Wood, J. E. Riedel, M. C. McHenry, M. Aung, L. A. Cangahuala, R. A. Volpe, P. M. Beauchamp, and J. A. Cutts, "Guidance, navigation, and control technology assessment for future planetary science missions," *J. Guid. Control Dyn.*, vol. 38, no. 7, pp. 1165–1186, 2015.

[4] J. T. Betts, "Survey of numerical methods for trajectory optimization," *J. Guid. Control Dyn.*, vol. 21, no. 2, pp. 193–207, 1998.

[5] B. A. Conway, "A survey of methods available for the numerical optimization of continuous dynamic systems," *J. Optim. Theory Appl.*, vol. 152, no. 2, pp. 271–306, 2012.

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

[7] B. Açıkmeşe and S. R. Ploen, "Convex programming approach to powered descent guidance for mars landing," *J. Guid. Control Dyn.*, vol. 30, no. 5, pp. 1353–1366, 2007.

[8] D. Morgan, S.-J. Chung, and F. Y. Hadaegh, "Model predictive control of swarms of spacecraft using sequential convex programming," *J. Guid. Control Dyn.*, vol. 37, no. 6, pp. 1725–1740, 2014.

[9] Y. Chen, M. Cutler, and J. P. How, "Decoupled multiagent path planning via incremental sequential convex programming," in *Proc. IEEE Conf. on Robot. and Autom.*, 2015, pp. 5954–5961.

[10] Z. Wang and M. J. Grant, "Minimum-fuel low-thrust transfers for spacecraft: A convex approach," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 5, pp. 2274–2290, 2018.

[11] Z. Wang and M. J. Grant, "Optimization of minimum-time low-thrust transfers using convex programming," *J. Spacecr. Rockets*, vol. 55, no. 3, pp. 586–598, 2018.

[12] C. Hofmann and F. Topputo, "Rapid low-thrust trajectory optimization in deep space based on convex programming," *J. Guid. Control Dyn.*, vol. 44, no. 7, pp. 1379–1388, 2021.

[13] C. Hofmann and F. Topputo, "Toward on-board guidance of low-thrust spacecraft in deep space using sequential convex programming," in *Proc. AAS/AIAA Space Flight Mechanics Meeting*, 2021, paper AAS 21-350.

[14] M. Saghamanesh, E. Taheri, and H. Baoyin, "Systematic low-thrust trajectory design to mars based on a full ephemeris modeling," *Adv. Space Res.*, vol. 64, no. 11, pp. 2356–2378, 2019.

[15] E. Taheri, J. L. Junkins, I. Kolmanovsky, and A. Girard, "A novel approach for optimal trajectory design with multiple operation modes of propulsion system, part 1," *Acta Astronaut.*, vol. 172, pp. 151–165,

2020.
[16] G. J. Whiffen, "Mystic: Implementation of the static dynamic optimal control algorithm for high-fidelity, low-thrust trajectory design," in *Proc. AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2006.
[17] J. Englander, J. M. Knittel, K. Williams, D. Stanbridge, and D. H. Ellison, "Validation of a low-thrust mission design tool using operational navigation software," in *Proc. AAS/AIAA Space Flight Mechanics Meeting*, 2017, paper AAS 17-204.
[18] F. Topputo, Y. Wang, G. Giordano, V. Franzese, H. Goldberg, F. Perez-Lissi, and R. Walker, "Envelop of reachable asteroids by M-ARGO cubesat," *Adv. Space Res.*, vol. 67, no. 12, pp. 4193–4221, 2021.
[19] Y. Kayama, K. C. Howell, M. Bando, and S. Hokamoto, "Low-thrust trajectory design with successive convex optimization for libration point orbits," *J. Guid. Control Dyn.*, vol. 45, no. 4, pp. 623–637, 2022.
[20] N. Kumagai and K. Oguri, "Adaptive-mesh sequential convex programming for space trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 0, no. 0, pp. 1–8, 2024.
[21] N. Ozaki, K. Oguri, and R. Funase, "Procyon mission reanalysis: Low-thrust asteroid flyby trajectory design leveraging convex programming," *The Journal of the Astronautical Sciences*, vol. 69, no. 1, pp. 1–27, 2022.
[22] J. Yang, Z. Zhang, F. Jiang, and J. Li, "Low-energy transfer design of heliocentric formation using lunar swingby on the example of lisa," *Aerospace*, vol. 10, no. 1, p. 18, 2023.
[23] F. Jia, D. Qiao, H. Han, and X. Li, "Efficient optimization method for variable-specific-impulse low-thrust trajectories with shutdown constraint," *Science China Technological Sciences*, vol. 65, no. 3, p. 581–594, 2022.
[24] M. C. Wijayatunga, R. Armellin, H. Holt, L. Pirovano, and C. Bombardelli, "Convex-optimization-based model predictive control for space debris removal mission guidance," *Journal of Guidance, Control, and Dynamics*, vol. 0, no. 0, pp. 1–20, 2024.
[25] B. Pan, P. Lu, X. Pan, and Y. Ma, "Double-homotopy method for solving optimal control problems," *J. Guid. Control Dyn.*, vol. 39, no. 8, pp. 1706–1720, 6 2016.
[26] F. Jiang, H. Baoyin, and J. Li, "Practical techniques for low-thrust trajectory optimization with homotopic approach," *J. Guid. Control Dyn.*, vol. 35, no. 1, pp. 245–258, 8 2012.
[27] A. C. Morelli, C. Hofmann, and F. Topputo, "Robust low-thrust trajectory optimization using convex programming and a homotopic approach," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 3, pp. 2103–2116, 2021.
[28] C. Hofmann and F. Topputo, "Pseudospectral convex low-thrust trajectory optimization in a high-fidelity model," in *Proc. AAS/AIAA Astrodynamics Specialist Conference*, 2021, paper AAS 21-678.
[29] C. Zhang, F. Topputo, F. Bernelli-Zazzera, and Y. S. Zhao, "Low-thrust minimum-fuel optimization in the circular restricted three-body problem," *J. Guid. Control Dyn.*, vol. 38, no. 8, pp. 1501–1510, 2015.
[30] C. Hofmann and F. Topputo, "Embedded homotopy for convex low-thrust trajectory optimization with operational constraints," in *AAS/AIAA Astrodynamics Specialist Conference*, 2022, paper AAS 22-750.
[31] D. Malyuta and B. Açıkmeşe, "Fast homotopy for spacecraft rendezvous trajectory optimization with discrete logic," *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 7, pp. 1262–1279, 2023.
[32] K. Chen, D. Zhang, K. Wang, Z. Shao, and L. T. Biegler, "Nonlinear homotopy interior-point algorithm for 6-dof powered landing guidance," *Aerospace Science and Technology*, vol. 127, p. 107707, 2022.
[33] W. J. Rugh, *Linear System Theory*, 2nd ed. Prentice-Hall, 1996.
[34] D. P. Lubey and D. J. Scheeres, "Identifying and estimating mismodeled dynamics via optimal control policies and distance metrics," *J. Guid. Control Dyn.*, vol. 37, no. 5, pp. 1512–1523, 2014.
[35] C. Acton, N. Bachman, B. Semenov, and E. Wright, "A look towards the future in the handling of space science mission geometry," *Planetary and Space Science*, vol. 150, pp. 9–12, 2018.
[36] X. Liu, Z. Shen, and P. Lu, "Exact convex relaxation for optimal flight of aerodynamically controlled missiles," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 4, 2016.
[37] E. Taheri and J. L. Junkins, "Generic smoothing for optimal bang-off-bang spacecraft maneuvers," *J. Guid. Control Dyn.*, vol. 41, no. 11, pp. 2470–2475, 2018.
[38] Y. Mao, M. Szmuk, X. Xu, and B. Açıkmeşe, "Successive convexification: A superlinearly convergent algorithm for non-convex optimal control problems," 2019, preprint, https://arxiv.org/abs/1804.06539.
[39] C. Hofmann, A. C. Morelli, and F. Topputo, "Performance assessment of convex low-thrust trajectory optimization methods," *J. Spacecr. Rockets*, vol. 60, no. 1, pp. 299–314, 2023.
[40] L. Xie, R.-Z. He, H.-B. Zhang, and G.-J. Tang, "Oscillation phenomenon in trust-region-based sequential convex programming for the nonlinear trajectory planning problem," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 4, pp. 3337–3352, 2022.
[41] N. Bernardini, N. Baresi, and R. Armellin, "State-dependent trust region for successive convex programming for autonomous spacecraft," *Astrodynamics*, 2024.
[42] L. Xie, X. Zhou, H.-B. Zhang, and G.-J. Tang, "Higher-order soft-trust-region-based sequential convex programming," *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 11, pp. 2199–2206, 2023.
[43] L. Xie, X. Zhou, H.-B. Zhang, and G.-J. Tang, "Hybrid-order soft trust region-based sequential convex programming for reentry trajectory optimization," *Advances in Space Research*, vol. 73, no. 6, pp. 3195–3208, 2024.
[44] M. A. Patterson and A. V. Rao, "GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Trans. Math. Softw.*, vol. 41, no. 1, 2014.
[45] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Journal on Optimization*, vol. 12, no. 4, pp. 979–1006, 2002.
[46] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *Proc. Eur. Control Conf.*, 2013, pp. 3071–3076.
[47] E. Taheri, I. Kolmanovsky, and E. Atkins, "Enhanced smoothing technique for indirect optimization of minimum-fuel low-thrust trajectories," *J. Guid. Control Dyn.*, vol. 39, no. 11, pp. 2500–2511, 2016.
[48] T. P. Reynolds, M. Szmuk, D. Malyuta, M. Mesbahi, B. Açıkmeşe, and J. M. Carson, "Dual quaternion-based powered descent guidance with state-triggered constraints," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 9, pp. 1584–1599, 2020.
[49] H. Zhou, X. Wang, Y. Bai, and N. Cui, "Ascent phase trajectory optimization for vehicle with multi-combined cycle engine based on improved particle swarm optimization," *Acta Astronaut.*, vol. 140, pp. 156–165, 2017.
[50] G. Tang, F. Jiang, and J. Li, "Fuel-optimal low-thrust trajectory optimization using indirect method and successive convex programming," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 2053–2066, 2018.

Christian Hofmann has recently completed his PhD in Aerospace Engineering in the Deep-Space Astrodynamics Research and Technology (DART) group at Politecnico di Milano, Italy, where he was working on trajectory optimization and autonomous guidance for interplanetary CubeSats. He holds a Master of Science in Aerospace Engineering from the Technical University of Braunschweig and a Bachelor of Science in Aerospace Computer Science from the University of Würzburg in Germany. He worked as a research assistant at the German Aerospace Center (DLR) before he joined the DART group in November 2018. He is now a postdoctoral researcher at the Massachusetts Institute of Technology, USA, where he works on mission analysis, trajectory design and optimization, and analytical methods in astrodynamics. His research interests include astrodynamics, nonlinear optimal control, interplanetary CubeSats, space trajectory optimization and computational guidance and control.

Francesco Topputo is a Full Professor of Aerospace Systems at Politecnico di Milano, Italy, and holds a position as Visiting Researcher at TU Delft, The Netherlands. His core research involves space flight dynamics and control, autonomous navigation, interplanetary CubeSat mission and system design. He has been PI in numerous research projects and he leads the DART research team composed of 25 researchers. He has authored more than 80 journal articles and over 200 works in total.