A.C. Morelli, A. Morselli, C. Giordano, F. Topputo
Convex Trajectory Optimization Using Thrust Regularization
Journal of Guidance Control and Dynamics, published online 29/11/2023
doi:10.2514/1.G007646

Convex Trajectory Optimization using Thrust Regularization*

Andrea C. Morelli†, Alessandro Morselli‡, Carmine Giordano§, and Francesco Topputo¶
Politecnico di Milano, Milan, Italy, 20156

I. Introduction

Following the advances of autonomous terrestrial vehicles, spacecraft with autonomous Guidance, Navigation, and Control capabilities are also becoming appealing, and this motivates the current momentum of research on computational guidance [1] of deep-space satellites [2–4]. Real-time low-thrust guidance implies solving an optimal control problem onboard. This poses significant challenges, as autonomous trajectory optimization introduces exceptionally strict requirements on algorithmic convergence and speed, as a feasible solution must be found without human intervention. Classical direct and indirect methods [5] are in general not suitable for real-time optimization as they are not fast and robust enough, respectively [3]. Therefore, machine learning [6] and convex optimization [7] have recently emerged as most promising techniques for onboard applications. However, machine learning-based methods lack of a complete theoretical framework [8]. Instead, convex optimization represents an interesting direct approach to solve the problem because it provides high levels of robustness with reduced computational effort [9]. Moreover, it relies on sound theoretical guarantees [10]. It has recently been applied to several aerospace-related problems [11]. Still, one major disadvantage of convex optimization is that it only provides the solution at the discretization points. Therefore, its output should be carefully used onboard as is, because interpolation between nodes must be performed to obtain the solution at the desired time instants. Therefore, the physical constraints may be violated, so making the required thrust commands in-between collocation points infeasible or affected by errors.

To overcome the aforementioned issues some works proposed to exploit the covector mapping theorem [12] associated with either direct methods and mesh refinement techniques [3, 13] or a combination of direct and indirect methods [14]. The approach in [13] for nonlinear programs solves the *whole* optimization problem multiple times using denser mesh grids with a number of nodes that can depend on the discretization error. Moreover, the technique in [3] for convex optimization consists of solving the problem twice using modified mesh grids where the nodes are placed at the expected switch on and off times. Finally, recent work developed a convex optimization-based strategy able to provide bang-bang thrust profiles [15]. However, none of the existing strategies is able to compute a feasible and perfectly bang-bang thrust profile with analytical expressions of the thrust components in a quick and robust fashion. In particular, analytical expressions of the thrust angles are desirable because they intrinsically satisfy the constraints at all

*Part of this work was presented as paper AAS 23-186 at the 33rd AAS/AIAA Space Flight Mechanics Meeting
†Ph.D. Candidate, Dept. of Aerospace Science and Technology. Email: andreacarlo.morelli@polimi.it.
‡Assistant Professor, Dept. of Aerospace Science and Technology. Email: alessandro.morselli@polimi.it. Corresponding author.
§PostDoc Fellow, Dept. of Aerospace Science and Technology. Email: carmine.giordano@polimi.it. AIAA Member.
¶Full Professor, Dept. of Aerospace Science and Technology. Email: francesco.topputo@polimi.it. AIAA Senior Member.

time instants. Moreover, they can be used for estimations and computations by other subsystems (e.g., the autonomous navigation algorithm). Finally, computing their derivatives may be relevant as well [16].

This work addresses the need for an exact definition of the thrust profile in terms of switch on and off times and thrust angles by proposing a double-layer algorithm that relies on convex optimization and also exploits a direct shooting method [17]. The utmost advantage of the proposed approach is that it offers a fast and reliable way to obtain near-optimal solutions for the low-thrust spacecraft trajectory optimization problem with analytical expressions for the thrust variables with no further iterative processes other than SCP and no a priori imposition of the thrust profile structure. Another key feature of our approach is the regularization of each thrust arc separately through a single shooting method, therefore simplifying the optimization procedure and enhancing the convergence properties, contrary to previous works in literature. Finally, we develop an analytical procedure based on simple physical considerations to generate the initial guess for the switch on and off times of the thrust arcs. Overall, our approach develops a guidance [18] process that outputs standardized control commands. The approach can consist both of a *closed-loop* one in case of onboard use and of a method to design trajectories on ground. The fact that each thrust arc is regularized separately is in fact tailored to closed-loop guidance, where only a portion of the computed commands are executed. We test the proposed algorithm in extensive simulations using the widely-used and effective Hermite–Simpson (HS) discretization [4] for the convex optimization algorithm.

The remainder of the article is organized as follows. Section II describes the problem that will be solved throughout the article. Section III explains the proposed strategy. Section IV assesses the performance of the strategy. Finally, section V concludes the work.

II. Problem Formulation

A. The Convex Low-Thrust Trajectory Optimization Problem

We consider the problem of finding the minimum-fuel trajectory of a spacecraft in motion around a primary body and equipped with a low-thrust engine. The equations of motion of such spacecraft can be written as [9]

$$\begin{bmatrix} \dot{\mathbf{r}}(t) \\ \dot{\mathbf{v}}(t) \\ \dot{m}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{v}(t) \\ -\mu \frac{\mathbf{r}(t)}{\|\mathbf{r}(t)\|_2^3} + \frac{\mathbf{T}(t)}{m(t)} \\ -\frac{\|\mathbf{T}(t)\|_2}{I_{\text{sp}} g_0} \end{bmatrix} \tag{1}$$

where $\mathbf{r} = [x, y, z]^\top$, $\mathbf{v} = [v_x, v_y, v_z]^\top$, and m are the position, velocity, and mass variables, respectively. The gravitational parameter of the primary body is indicated as μ, $\mathbf{T} = [T_x, T_y, T_z]^\top$ is the thrust vector, I_{sp} is the specific impulse, and g_0 is the gravitational acceleration of the Earth at sea level. As the dynamics in Eq. (1) are nonconvex, they have to be modified for the problem to be expressed in a convex form. The dimensionless convexified low-thrust trajectory optimization problem therefore reads [9, 19]

$$\underset{\mathbf{u}(t)}{\text{minimize}} \quad -w(t_f) + \lambda \int_{t_0}^{t_f} \|\boldsymbol{\nu}(t)\|_1 \, \mathrm{d}t + \lambda \int_{t_0}^{t_f} \max\left(0, \eta(t)\right) \mathrm{d}t \tag{2a}$$

$$\text{subject to:} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(\bar{\mathbf{x}}(t), \bar{\mathbf{u}}(t)) + \mathbf{A}(\bar{\mathbf{x}}(t))\left(\mathbf{x}(t) - \bar{\mathbf{x}}(t)\right) + \mathbf{B}(\mathbf{u}(t) - \bar{\mathbf{u}}(t)) + \boldsymbol{\nu}(t) \tag{2b}$$

$$\Gamma(t) \leq T^*_{\text{max}} \mathrm{e}^{-\bar{w}(t)} \left(1 - w(t) + \bar{w}(t)\right) + \eta(t) \tag{2c}$$

$$\|\boldsymbol{\tau}(t)\|_2 \leq \Gamma(t) \tag{2d}$$

$$\|\mathbf{x}(t) - \bar{\mathbf{x}}(t)\|_1 \leq R \tag{2e}$$

$$\mathbf{r}(t_0) = \mathbf{r}_0, \;\; \mathbf{v}(t_0) = \mathbf{v}_0, \;\; w(t_0) = w_0 \tag{2f}$$

$$\mathbf{r}(t_f) = \mathbf{r}_f, \;\; \mathbf{v}(t_f) = \mathbf{v}_f \tag{2g}$$

$$\mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u, \;\; \mathbf{u}_l \leq \mathbf{u} \leq \mathbf{u}_u. \tag{2h}$$

$\mathbf{x} = [\mathbf{r}^\top, \mathbf{v}^\top, w]^\top$ and $\mathbf{u} = [\tau_x, \tau_y, \tau_z, \Gamma]^\top = [\boldsymbol{\tau}^\top, \Gamma]^\top$ are the new state and control variables, respectively; in particular, $w(t) = \ln[m(t)]$ is the modified mass variable, and the relationship between the SCP controls $\mathbf{u}(t)$ and the canonical thrust variables is [19]

$$T_q(t) = \tau_q(t) e^{w(t)}, \; q = x, y, z; \quad \|\mathbf{T}(t)\|_2 = \Gamma(t) e^{w(t)}. \tag{3}$$

$\boldsymbol{\nu}(t)$ and $\eta(t)$ are slack variables introduced to avoid artificial infeasibility, and $\lambda > 0$ is a constant weight. The times t_0 and t_f are the initial and final transfer times, T^*_{max} is the dimensionless maximum allowable thrust, and R is the radius of the trust region assuring that the convexification of the problem is valid. In Eq. (2b), the matrices $\mathbf{A}$ and $\mathbf{B}$ are defined as

$$\mathbf{A}(\bar{\mathbf{x}}(t)) \coloneqq \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\bar{\mathbf{x}}(t)}, \; \mathbf{B} \coloneqq \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\bar{\mathbf{u}}(t)}, \tag{4}$$

where $\mathbf{f} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ are the dynamics

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{v}(t) \\ -\frac{\mathbf{r}(t)}{\|\mathbf{r}(t)\|_2^3} + \boldsymbol{\tau}(t) \\ -\frac{\Gamma(t)}{I^*_{\text{sp}} g^*_0} \end{bmatrix}. \tag{5}$$

The quantities $(\cdot)^*$ indicate dimensionless parameters. The original optimal control problem is solved through the sequential convex programming (SCP) [20], an iterative technique that considers a sequence of convex subproblems of the form (2a)–(2h).

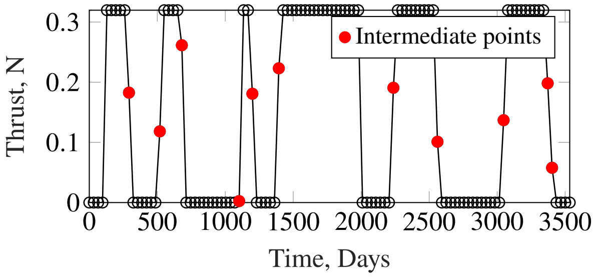


Fig. 1 Typical thrust profile obtained with convex optimization for an Earth–Dionysus transfer.

B. The Thrust Regularization Problem

Figure 1 shows a typical thrust profile obtained with SCP for an Earth to asteroid Dionysus transfer. The profile has (almost) the classical bang-bang structure, indicating that a (sub-)optimal solution to the problem has been found. However, there are some intermediate points (red filled circles) that are neither 0 nor $T_{\max}$. On top of that, Eq. (2d) expresses the convexified version of the real physical constraint on the thrust variables $\sqrt{\tau_x^2(t)+\tau_y^2(t)+\tau_z^2(t)} = \Gamma(t)$. Note that this is a constraint that arises when the problem is written in convex form. Due to discretization, this relationship is only satisfied at the collocation points t_j (i.e., the points where the constraints are actually imposed), with $j = 1, \ldots, M$, independently of the discretization strategy. Outside of these points, the controls must be interpolated. In this work, the Hermite–Simpson discretization is used, and therefore the thrust variables are linearly interpolated in-between two nodes. This causes the constraint on the thrust variables to exceed the convergence tolerance outside of the collocation points [21], as shown in Fig. 2. Let us define the ratio ξ between the maximum violation of the constraint on the SCP thrust variables and the value $T_{\max}$ associated with the selected engine for the considered transfer as

$$\xi = e^{w(t)} \frac{\left\| \sqrt{\tau_x^2(t)+\tau_y^2(t)+\tau_z^2(t)} - \Gamma(t) \right\|}{T_{\max}} \times 100, \quad t \in [t_0, t_f]. \tag{6}$$

The left y-axis of Fig. 3 shows how $\max\limits_{t\in[t_0,t_f]} \xi$ varies as a function of the discretization intervals for the Earth–Dionysus transfer.

The right y-axis of the same figure reports the CPU time required to solve the problem as a function of the discretization intervals. The error ratio is in the range of $\approx 0.5 - 7\%$, i.e., in the same order of magnitude of the state-of-the-art low-thrust engines execution errors ([22] indicates that these are in the order of 5%). It is worth noting that Fig. 3 only shows the *maximum* point wise error, and hence the cumulative error will be higher. Increasing the number of discretization points does not solve the issue: as Fig. 3 shows, the correspondent CPU time also increases rapidly with the number of intervals. On top of these considerations, the strategy developed in this work generates the

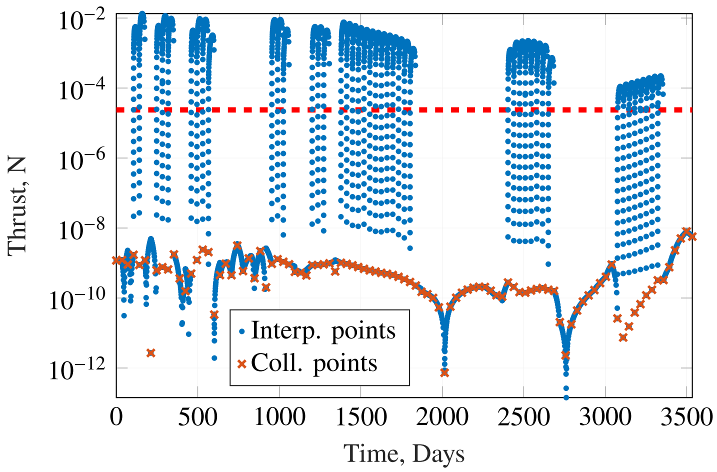


Fig. 2 Thrust constraint violation during the whole duration of a typical Earth–Dionysus transfer, where the red dashed line indicates the SCP algorithm convergence threshold.

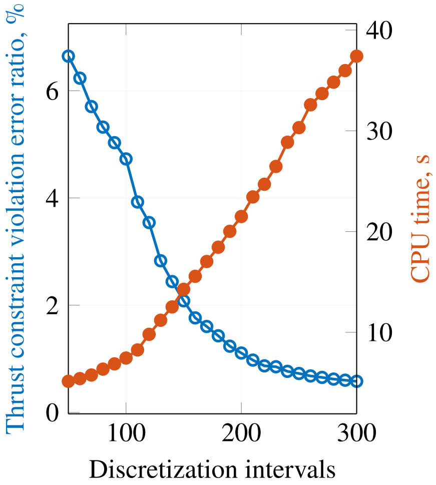


Fig. 3 Max. of ratio ξ (empty circles, left y-axis) and SCP CPU time (filled circles, right y-axis).

thrust profile of a low-thrust spacecraft by formulating the so-called *thrust regularization problem* (TRP):

$$\text{Find} \quad \mathbf{a}^{(i)}, \mathbf{b}^{(i)}, t_{\text{ON}}^{(i)}, t_{\text{OFF}}^{(i)} \tag{7a}$$

$$\text{such that} \quad \mathbf{T}^{(i)}(t) = T_{\max} \begin{bmatrix} \cos[\beta^{(i)}(t)] \cos[\alpha^{(i)}(t)] \\ \cos[\beta^{(i)}(t)] \sin[\alpha^{(i)}(t)] \\ \sin[\beta^{(i)}(t)] \end{bmatrix} \tag{7b}$$

$$\begin{cases} \alpha^{(i)}(t) & = \sum_{j=0}^{p} a_j^{(i)} t^j = \mathbf{a}^{(i)} \cdot [1, t, \ldots, t^p] \\ \beta^{(i)}(t) & = \sum_{j=0}^{p} b_j^{(i)} t^j = \mathbf{b}^{(i)} \cdot [1, t, \ldots, t^p] \end{cases} \tag{7c}$$

for all the thrust segments $i = 1, \ldots, N_{\text{T}}$ that can be identified in the SCP solution of a low-thrust space trajectory optimization problem. We want to find the switch on and off times $t_{\text{ON}}^{(i)}$ and $t_{\text{OFF}}^{(i)}$, respectively, and the coefficients of the polynomials that describe the thrust angles, such that the thrust magnitude is always $T_{\max}$ inside the thrust arcs and 0 elsewhere, i.e., we want to obtain a perfect bang-bang profile. This is justified by the theory, which shows that the solution of the problem in Eq. (2) has such structure [23].

It can be noted that the above statement of the TRP is similar to what is usually done in either indirect or direct shooting methods. The former, although relatively computationally inexpensive, require very good initial guesses for the costates and are therefore not robust enough for onboard applications. On the other hand, the combination of classical nonlinear transcription and direct shooting usually requires long computational times. Our method, instead:

1. builds an algorithm that possesses the three characteristics of robustness, accuracy, and rapidity at the same time;
2. regularizes each thrust arc separately, hence strongly simplifying the optimization process;
3. relies on an analytical initial guess.

Remark 1: in the points above and throughout the rest of the paper, the term accuracy is used differently with respect to how it is used in Ref. [24]. In this work, we refer to the term as the property of the guidance output to not introduce any errors when it is evaluated outside of the collocation points and that, when used to propagate its state, allows the spacecraft to reach the target up to a selected threshold.

III. Strategy

Figure 4 shows the flowchart of the strategy proposed to solve the whole low-thrust trajectory optimization problem. The thin grey boxes indicate the SCP algorithm, whereas the bold blue boxes highlight the methodology developed in this paper, which consists of an initial guess generator and a direct shooting regularization (DSR).

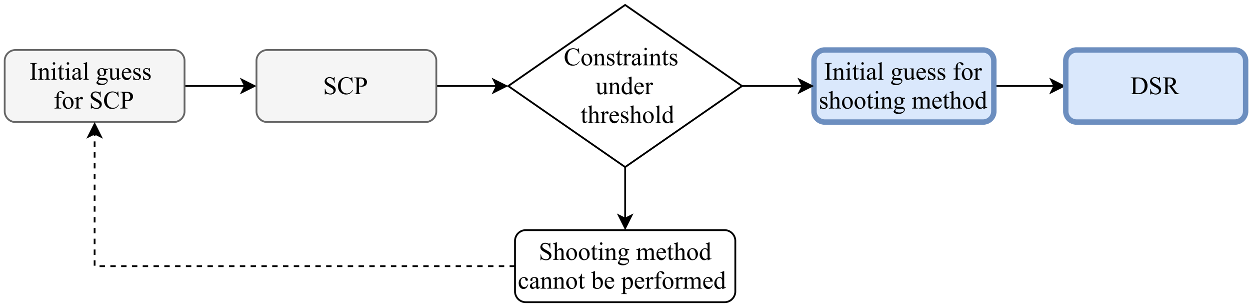

Initial guess for SCP
SCP
Constraints under threshold
Initial guess for shooting method
DSR
Shooting method cannot be performed

Fig. 4 High-level description of the thrust regularization procedure.

The rhomboid block establishes whether the thrust regularization can be performed or not. If the state and control constraints violation of the SCP procedure is below a certain threshold, it is possible to proceed. Otherwise, no further operation can be executed.

A. Thrust Regularization Initial Guess

The DSR initial guess generation consists of three steps. To distinguish among the thrust-regularized, the initial guess for the thrust regularization procedure, and the SCP quantities, we use the notations $[\cdot]$, $[\hat{\cdot}]$, and $[\tilde{\cdot}]$, respectively.

Step 1 First, the thrust arcs must be identified. We consider a threshold ε_{T} under which the SCP thrust magnitude is considered to be 0. Consequently, the SCP thrust on and off times $\tilde{t}_{\mathrm{ON}}^{(i)}$ and $\tilde{t}_{\mathrm{OFF}}^{(i)}$ are defined as the instants when the thrust magnitude becomes higher or lower than the selected threshold, respectively. The actual value of ε_{T} is found through preliminary analysis. The thrust arcs identification process can be formalized as in Algorithm 1. In order to more precisely capture the thrust arcs, the thrust and state variables are interpolated at a higher number of points N_{INT} with respect to the collocation points.

Algorithm 1 Algorithm used for the definition of the thrust arcs.

```
$i = 1$, $j_{\text{old}} = 1$, $N_{\text{T}} = 0$;
while $j_{\text{old}} \leq N_{\text{INT}}$ do
    for $j = j_{\text{old}}, \ldots, N_{\text{INT}}$ do
        if $\left\|\mathbf{T}(t_j)\right\|_2 \geq \varepsilon_{\text{T}}$ then
            $\tilde{t}_{\text{ON}}^{(i)} = t_j$;
            $i = i + 1$;
            for $k = j + 1, \ldots, N_{\text{INT}}$ do
                if $\|\mathbf{T}(t_k)\|_2 \leq \varepsilon_{\text{T}}$ then
                    $\tilde{t}_{\text{OFF}}^{(i)} = t_k$;
                    $j_{\text{old}} = k + 1$;
                    break;
                end if
            end for
            break;
        end if
    end for
end while
$N_{\text{T}} = i - 1$;
```

Remark 2: any arbitrary collocation method can be used associated with the proposed strategy. In the case of methods that parameterize the controls (such as pseudsospectral [3] or higher-order Hermite interpolation-based [4] discretization), however, it is required that the interpolating polynomials of the controls do not exceed the imposed thrust bounds; otherwise, the thrust arcs identification procedure described in Algorithm 1 may fail. A possible approach would be to use Bernstein polynomials to interpolate the controls. In this case, in fact, it has been proven that they remain within the feasible set for all times in case of convex feasible sets [16].

Step 2 We generate the initial guess for the times t_{ON} and t_{OFF} considering the following hypotheses. For each thrust arc i,

1) the total change of spacecraft velocity $\widetilde{\Delta V}^{(i)}$ of the SCP solution is considered equal to the total change of spacecraft velocity $\widehat{\Delta V}^{(i)}$ associated with the thrust regularization initial guess, where

$$\widetilde{\Delta V}^{(i)} \equiv \int_{\tilde{t}_{\text{ON}}^{(i)}}^{\tilde{t}_{\text{OFF}}^{(i)}} \Gamma(t)\,\mathrm{d}t. \tag{8}$$

2) the barycentric time $\tilde{t}_c^{(i)}$ of the SCP solution is considered equal to the barycentric time $\hat{t}_{\text{c}}^{(i)}$ associated with the initial guess thrust profile, where

$$\tilde{t}_{\text{c}}^{(i)} \equiv \frac{\displaystyle\int_{\tilde{t}_{\text{ON}}^{(i)}}^{\tilde{t}_{\text{OFF}}^{(i)}} t\Gamma(t)\,\mathrm{d}t}{\displaystyle\int_{\tilde{t}_{\text{ON}}^{(i)}}^{\tilde{t}_{\text{OFF}}^{(i)}} \Gamma(t)\,\mathrm{d}t}, \tag{9}$$

The integrals in Eqs. (8) and (9) are computed numerically.

3) The thrust magnitude is always equal to the maximum value $T_{\max}$, i.e., $\|\mathbf{T}(t)\|_2 \equiv T_{\max}$.

In the light of these considerations, the following system of equations can be written to define the initial guess values for the times $t_{\mathrm{ON}}^{(i)}$ and $t_{\mathrm{OFF}}^{(i)}$:

$$\begin{cases} \displaystyle\int_{\hat{t}_{\mathrm{ON}}}^{\hat{t}_{\mathrm{OFF}}} \frac{T_{\max}}{m}\,\mathrm{d}t = \widetilde{\Delta V} \\[2ex] \dfrac{\displaystyle\int_{\hat{t}_{\mathrm{ON}}}^{\hat{t}_{\mathrm{OFF}}} t\frac{T_{\max}}{m}\,\mathrm{d}t}{\displaystyle\int_{\hat{t}_{\mathrm{ON}}}^{\hat{t}_{\mathrm{OFF}}} \frac{T_{\max}}{m}\,\mathrm{d}t} = \tilde{t}_c \\[2ex] \dot{m} = -\dfrac{T_{\max}}{I_{\mathrm{sp}}g_0} \end{cases} \tag{10}$$

where the only unknowns are $\hat{t}_{\mathrm{ON}}$ and $\hat{t}_{\mathrm{OFF}}$, i.e. the initial guesses for the times t_{ON} and t_{OFF}. In Eqs. (10), the apex i has been dropped for simplicity. Considering the third of Eqs. (10), we have

$$m_f = m_i - \frac{T_{\max}}{I_{\mathrm{sp}}g_0}(\hat{t}_{\mathrm{OFF}} - \hat{t}_{\mathrm{ON}}), \tag{11}$$

where $m_i = m(\hat{t}_{\mathrm{ON}})$ and $m_f = m(\hat{t}_{\mathrm{OFF}})$. According to the Tsiolkovsky equation and under the hypothesis that the first of Eqs. (10) holds, m_f can also be found as

$$m_f = m_i \exp\left(-\frac{\widetilde{\Delta V}}{I_{\mathrm{sp}}g_0}\right). \tag{12}$$

Therefore,

$$\begin{aligned} m_i - \frac{T_{\max}}{I_{\mathrm{sp}}g_0}(\hat{t}_{\mathrm{OFF}} - \hat{t}_{\mathrm{ON}}) &= m_i \exp\left(-\frac{\widetilde{\Delta V}}{I_{\mathrm{sp}}g_0}\right) \\ \frac{T_{\max}}{I_{\mathrm{sp}}g_0}(\hat{t}_{\mathrm{OFF}} - \hat{t}_{\mathrm{ON}}) &= m_i\left[1 - \exp\left(-\frac{\widetilde{\Delta V}}{I_{\mathrm{sp}}g_0}\right)\right], \end{aligned}$$

and, finally,

$$\hat{t}_{\mathrm{OFF}} = \hat{t}_{\mathrm{ON}} + \frac{m_i I_{\mathrm{sp}}g_0}{T_{\max}}\left[1 - \exp\left(-\frac{\widetilde{\Delta V}}{I_{\mathrm{sp}}g_0}\right)\right]. \tag{13}$$

Let $k_1 \equiv \frac{m_i I_{\text{sp}} g_0}{T_{\text{max}}}$ and $k_2 \equiv \exp\left(-\frac{\widetilde{\Delta V}}{I_{\text{sp}} g_0}\right)$. If we now substitute Eq. (11) at a general time instant t inside the second of Eqs. (10), we obtain

$$\begin{aligned}
&\tilde{t}_c \int_{\hat{t}_{\text{ON}}}^{\hat{t}_{\text{OFF}}} \frac{T_{\text{max}}}{m_i - \frac{T_{\text{max}}}{I_{\text{sp}} g_0}(t - \hat{t}_{\text{ON}})} \, \mathrm{d}t = \int_{\hat{t}_{\text{ON}}}^{\hat{t}_{\text{OFF}}} t \frac{T_{\text{max}}}{m_i - \frac{T_{\text{max}}}{I_{\text{sp}} g_0}(t - \hat{t}_{\text{ON}})} \, \mathrm{d}t \\
&\tilde{t}_c \int_{\hat{t}_{\text{ON}}}^{\hat{t}_{\text{OFF}}} \frac{1}{t - (k_1 + \hat{t}_{\text{ON}})} \, \mathrm{d}t = \int_{\hat{t}_{\text{ON}}}^{\hat{t}_{\text{OFF}}} \frac{t}{t - (k_1 + \hat{t}_{\text{ON}})} \, \mathrm{d}t \\
&\tilde{t}_c \left[\ln|t - k_1 - \hat{t}_{\text{ON}}| \right]_{\hat{t}_{\text{ON}}}^{\hat{t}_{\text{OFF}}} = \left[(k_1 + \hat{t}_{\text{ON}}) \ln|t - k_1 - \hat{t}_{\text{ON}}| + t \right]_{\hat{t}_{\text{ON}}}^{\hat{t}_{\text{OFF}}} \\
&\tilde{t}_c \ln\left(\frac{-\hat{t}_{\text{OFF}} + k_1 + \hat{t}_{\text{ON}}}{k_1}\right) = (k_1 + \hat{t}_{\text{ON}}) \ln\left(\frac{-\hat{t}_{\text{OFF}} + k_1 + \hat{t}_{\text{ON}}}{k_1}\right) + (\hat{t}_{\text{OFF}} - \hat{t}_{\text{ON}})
\end{aligned} \tag{14}$$

By substituting the expression of $\hat{t}_{\text{OFF}}$ obtained in Eq. (13), we get

$$\begin{aligned}
\tilde{t}_c \ln k_2 &= (k_1 + \hat{t}_{\text{ON}}) \ln k_2 + k_1 - k_1 k_2 \\
\hat{t}_{\text{ON}} &= \frac{\tilde{t}_c \ln k_2 + k_1(k_2 - \ln k_2 - 1)}{\ln k_2}.
\end{aligned} \tag{15}$$

Finally, by back-substituting the constants k_1 and k_2, it is obtained

$$\hat{t}_{\text{ON}} = \tilde{t}_c - \frac{m_i I_{\text{sp}} g_0}{T_{\text{max}}} \left[1 + \frac{I_{\text{sp}} g_0}{\widetilde{\Delta V}} \left(\exp\left(-\frac{\widetilde{\Delta V}}{I_{\text{sp}} g_0} \right) - 1 \right) \right] \tag{16}$$

Figure 5 shows the scheme of the thrust profile as obtained by the SCP algorithm (together with the associated switch on and off times $\tilde{t}_{\text{ON}}^{(i)}$ and $\tilde{t}_{\text{OFF}}^{(i)}$) and the initial guess for the switch on and off times of the regularized thrust profile $\hat{t}_{\text{ON}}^{(i)}$ and $\hat{t}_{\text{OFF}}^{(i)}$, respectively. The red filled circles represent the thrust magnitude at the interpolation points.

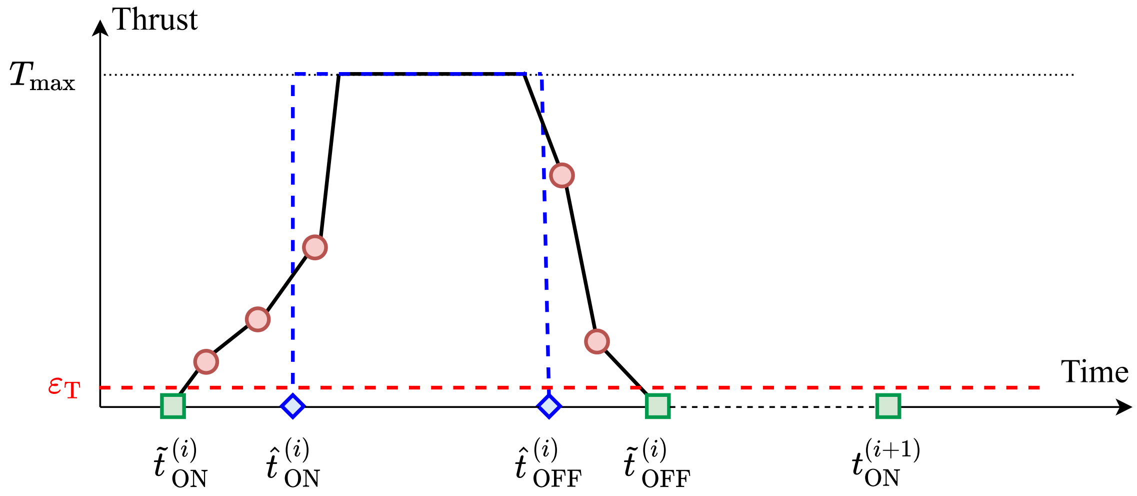

Thrust
T_{max}
ε_{T}
Time
$\tilde{t}_{\text{ON}}^{(i)}$
$\hat{t}_{\text{ON}}^{(i)}$
$\hat{t}_{\text{OFF}}^{(i)}$
$\tilde{t}_{\text{OFF}}^{(i)}$
$t_{\text{ON}}^{(i+1)}$

Fig. 5 Comparison of SCP (black solid line) and initial-guess identified (blue dashed line) thrust profiles.

Step 3 The thrust angles generation procedure consists of three sub-steps:

1. the thrust variables of the SCP algorithm $\{\tau_x, \tau_y, \tau_z, \Gamma\}$ are transformed into the canonical thrust variables $\left\{T_x, T_y.T_z, \|\mathbf{T}\|_2\right\}$ [19].
2. After that, a change of coordinates is performed to obtain the values of the thrust angles in Eqs. (7c) at the SCP collocation points.
3. Finally, a least-square polynomial interpolation is executed to finally obtain the coefficients of the polynomials in Eqs. (7c).

Remark 3: Note that the initial guess times $\hat{t}_{\mathrm{ON}}$ and $\hat{t}_{\mathrm{OFF}}$ represent significantly better estimates for the regularized times t_{ON} and t_{OFF} if compared with the SCP switch on and off times $\tilde{t}_{\mathrm{ON}}$ and $\tilde{t}_{\mathrm{OFF}}$: it has been verified that, for the Earth–Dionysus transfer,

$$|\tilde{t}_{\mathrm{ON}} - t_{\mathrm{ON}}| \approx |\tilde{t}_{\mathrm{OFF}} - t_{\mathrm{OFF}}| \approx 10^1 \text{ days}$$

$$|\hat{t}_{\mathrm{ON}} - t_{\mathrm{ON}}| \approx |\hat{t}_{\mathrm{OFF}} - t_{\mathrm{OFF}}| \approx 10^{-1} - 10^0 \text{ days},$$

i.e., the proposed initial guesses are more precise than using the simple SCP results by at least one order of magnitude.

B. Direct Shooting Optimization

A direct shooting method is used to solve the TRP described in Eqs. (7). Algorithm 2 summarizes the fundamental steps of the approach. The whole strategy consists of imposing that the spacecraft state at the beginning and at the end of each thrust arc is the same before and after the thrust regularization procedure. First, the states $\mathbf{x}_1^{(1)}$ and $\mathbf{x}_2^{(1)}$ associated with the SCP switch on and off times of the first thrust segment are fed to the algorithm (line 1). Then, for each thrust arc i, the shooting method iterations k are initialized, and the initial guess for $t_{\mathrm{ON}}^{(i)}$, $t_{\mathrm{OFF}}^{(i)}$, and for the coefficients of the thrust angles polynomials $\mathbf{a}^{(i)}$ and $\mathbf{b}^{(i)}$ are provided (lines 2–4). Referring to Fig. 5, the previously-defined instants $\tilde{t}_{\mathrm{ON}}^{(i)}$, $\hat{t}_{\mathrm{ON}}^{(i)}$, $\tilde{t}_{\mathrm{OFF}}^{(i)}$, and $\hat{t}_{\mathrm{OFF}}^{(i)}$ identify three time intervals. Starting at $\mathbf{x}_1^{(i)}$, the spacecraft free dynamics is propagated from $\tilde{t}_{\mathrm{ON}}^{(i)}$ to $t_{\mathrm{ON},k}^{(i)}$, i.e. the switch on time of the regularized thrust profile associated with the shooting method iteration k (line 6). Then, the low-thrust dynamics in Eq. (1) is propagated from $t_{\mathrm{ON},k}^{(i)}$ to $t_{\mathrm{OFF},k}^{(i)}$ (line 7), where the latter is the switch off time of the regularized thrust profile associated with the shooting method iteration k. In Eq. (1), the thrust vector $\mathbf{T}(t)$ is such that its components are defined as in Eq. (7b). The free dynamics is further propagated from $t_{\mathrm{OFF},k}^{(i)}$ to $\tilde{t}_{\mathrm{OFF}}^{(i)}$ (line 8). The ∞-norm of the error between the spacecraft state after the thrust regularization procedure, $\mathbf{x}_{c,k}^{(i)}$, and the target spacecraft state $\mathbf{x}_2^{(i)}$ is computed (line 9). The algorithm stops when the error is below a certain threshold or if either the elapsed optimization time Δs or the shooting iterations k overcome some thresholds $s_{\max}$ and $k_{\max}$, respectively (line 5). The overlying procedure is executed separately for each thrust arc; this means that the user can choose whether

Algorithm 2 Thrust regularization algorithm

```
 1: $\mathbf{x}_1^{(1)} = \mathbf{x}(\tilde{t}_{\text{ON}}^{(1)})$, $\mathbf{x}_2^{(1)} = \mathbf{x}(\tilde{t}_{\text{OFF}}^{(1)})$;
 2: for $i = 1, \dots, N_{\text{T}}$ do
 3:     $k = 1$, $s_i = current\ time$, $\Delta = 1$, $t_{\text{ON},0}^{(i)} = \tilde{t}_{\text{ON}}^{(i)}$, $t_{\text{OFF},0}^{(i)} = \tilde{t}_{\text{OFF}}^{(i)}$, $\mathbf{a}_0^{(i)} = \hat{\mathbf{a}}^{(i)}$, $\mathbf{b}_0^{(i)} = \hat{\mathbf{b}}^{(i)}$;
 4:     while $\Delta > \varepsilon_{\text{ST}}$ and $k < k_{\max}$ and $\Delta s < s_{\max}$ do
 5:         Starting at $\mathbf{x}_1^{(i)}$, propagate the free dynamics from $\tilde{t}_{\text{ON}}^{(i)}$ to $t_{\text{ON},k}^{(i)}$ and obtain $\mathbf{x}_{a,k}^{(i)}$;
 6:         Starting at $\mathbf{x}_{a,k}^{(i)}$, propagate the low-thrust dynamics in Eq. (1) from $t_{\text{ON},k}^{(i)}$ to $t_{\text{OFF},k}^{(i)}$ and obtain $\mathbf{x}_{b,k}^{(i)}$;
 7:         Starting at $\mathbf{x}_{b,k}^{(i)}$, propagate the free dynamics from $t_{\text{OFF},k}^{(i)}$ to $\tilde{t}_{\text{OFF}}^{(i)}$ and obtain $\mathbf{x}_{c,k}^{(i)}$;
 8:         $\Delta = \left\|\mathbf{x}_{c,k}^{(i)} - \mathbf{x}_2^{(i)}\right\|_\infty$;        ▷ Note that the constraint does not include the mass.
 9:         Update variables $t_{\text{ON,k}}$, $t_{\text{OFF,k}}$, $\mathbf{a}_k^{(i)}$, $\mathbf{b}_k^{(i)}$;
10:         $k = k + 1$;
11:         $\Delta s = s_i - current\ time$;
12:     end while
13:     $t_{\text{ON}}^{(i)} = t_{\text{ON},k-1}^{(i)}$, $t_{\text{OFF}}^{(i)} = t_{\text{OFF},k-1}^{(i)}$, $\mathbf{a}^{(i)} = \mathbf{a}_{k-1}^{(i)}$, $\mathbf{b}^{(i)} = \mathbf{b}_{k-1}^{(i)}$;
14:     if $1 < i < N_{\text{T}} - 1$ then
15:         Starting at $\mathbf{x}_{\text{c},k}^{(i)}$, propagate the free dynamics from $t_{\text{OFF}}^{(i)}$ to $\tilde{t}_{\text{ON}}^{(i+1)}$ and obtain $\mathbf{x}_{d,k}^{(i)}$;
16:         $\mathbf{x}_1^{(i+1)} = \mathbf{x}_{d,k}^{(i)}$, $\mathbf{x}_2^{(i+1)} = \mathbf{x}(\tilde{t}_{\text{OFF}}^{(i+1)})$;
17:     else if $i \equiv N_{\text{T}} - 1$ then
18:         $\mathbf{x}_1^{(i+1)} = \mathbf{x}_{d,k}^{(i)}$, $\mathbf{x}_2^{(i+1)} = \mathbf{x}_f$;
19:     end if
20: end for
```

to perform the regularization for the whole thrust profile or not. In the first case, the initial state $\mathbf{x}_1^{(i+1)}$ is obtained by further propagating the free dynamics from $t_{\text{OFF}}^{(i)}$ to $\tilde{t}_{\text{ON}}^{(i+1)}$, for $i = 1, \dots, N_{\text{T}} - 2$ (lines 15–16). For the first thrust arc, on the contrary, the state $\mathbf{x}_1^{(1)}$ is directly obtained by interpolating the SCP solution and evaluating it at time $\tilde{t}_{\text{ON}}^{(1)}$ (line 1). Moreover, for the last thrust arc (i.e., $i = N_{\text{T}-1}$), the state $\mathbf{x}_2^{(N_{\text{T}})}$ is considered to be the final boundary condition (line 19). The MATLAB® function *fmincon* with the *active-set* algorithm is used in this article to solve the shooting optimization.

Remark 4: Due to the imperfect bang-bang structure of the thrust profile obtained with SCP, the times $t_{\text{ON}}^{(i)}$ and $t_{\text{OFF}}^{(i)}$ are such that

$$t_{\text{ON}}^{(i)} > \tilde{t}_{\text{ON}}^{(i)}, \quad t_{\text{OFF}}^{(i)} < \tilde{t}_{\text{OFF}}^{(i)}, \tag{18}$$

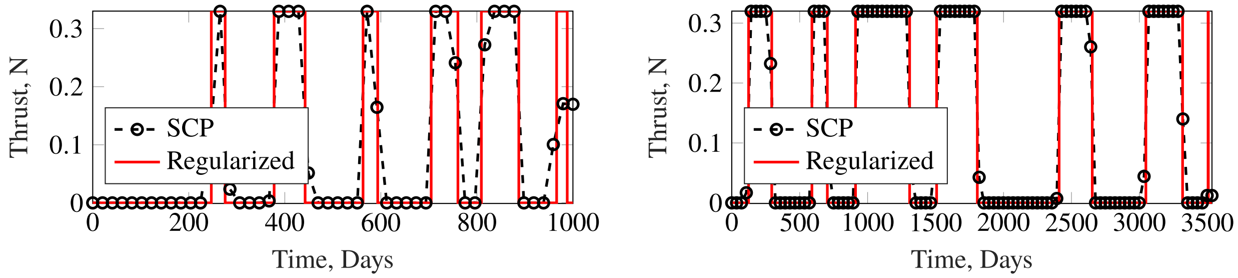
and in particular $t_{\text{ON}}^{(1)} > \tilde{t}_{\text{ON}}^{(1)} \geq t_0$ and $t_{\text{OFF}}^{(N_{\text{T}})} < \tilde{t}_{\text{OFF}}^{(N_{\text{T}})} \leq t_f$, i.e. the thrust regularization problem is feasible even when the thrust profile starts or terminates with the thrust on. If errors from previous thrust arcs regularization procedures are encountered, however, the regularization of the last thrust arc for the case $\tilde{t}_{\text{OFF}}^{(N_{\text{T}})} \approx t_f$ may fail to converge.

IV. Numerical Simulations

The performance of the proposed algorithm is assessed in several simulations where the entire thrust profile is regularized. The Earth–Venus and Earth–Dionysus transfers are considered as test cases. Data on the transfers and the associated spacecraft engines can be found in literature, together with various parameters that are used to solve the SCP layer [9]. Table 1 summarizes the thrust regularization algorithm parameters. To address the quality of the

Table 1 Parameters for the thrust regularization algorithm

Parameter	Value
Shooting convergence threshold $\varepsilon_{\mathrm{ST}}$	1×10^{-11}
Thrust threshold ε_{T}	1×10^{-6}
Elapsed optimization time threshold $s_{\max}$, s	5
Max. shooting iters. $k_{\max}$	10
Max. SCP iters. $k_{\max}^{\mathrm{SCP}}$	70, 100
Thrust regularization flag threshold $\varepsilon_{\mathrm{SCP}}$	1×10^{-4}



(a) Earth–Venus transfer.

(b) Earth–Dionysus

Fig. 6 Comparison of thrust profiles as obtained by the SCP algorithm and after regularization.

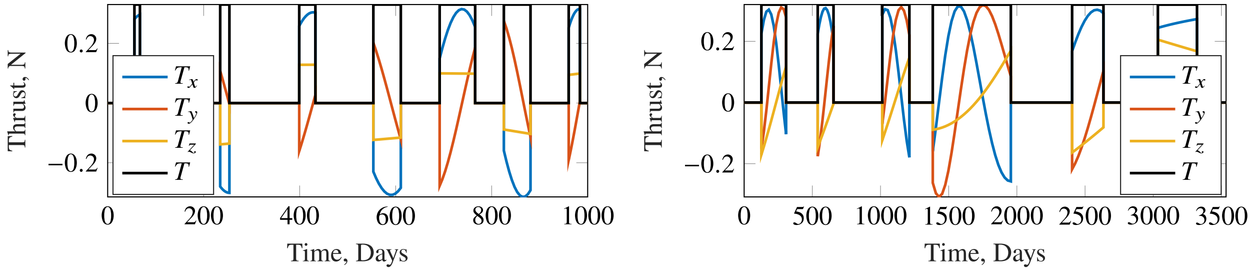
proposed algorithm, we perform the following simulations. We select different number of discretization nodes and we run 100 simulations for each of them, providing the SCP algorithm with poor perturbed cubic interpolation-based initial guesses [9]. We run the thrust regularization strategy even in the cases for which the SCP algorithm is close to convergence after a maximum number of iterations, but still it did not satisfy the convergence criterion. The maximum number of admissible SCP iterations $k_{\max}^{\mathrm{SCP}}$ and the threshold $\varepsilon_{\mathrm{SCP}}$ that is used to decide whether to perform the thrust regularization are also reported in Table 1. Clearly, the value of $\varepsilon_{\mathrm{SCP}}$ is higher than the SCP convergence threshold, which in our case is 1×10^{-6} as in [9]. All simulations are carried out in MATLAB® version R2020b on an Intel Core i7-10700@2.90 GHz desktop computer with 16 GB of RAM. Figure 6 shows examples of regularized thrust profiles for the Earth–Venus and Earth–Dionysus transfers, respectively. Finally, Figs. 7a and 7b highlight the thrust components of the regularized thrust profiles.

A. Convergence Assessment

The following criterion is used when deciding whether the SCP algorithm or the thrust regularization procedure find a feasible solution. First, the equations of motion in Eq. (1) are propagated with the obtained controls and the functions

$$g = \left\|\mathbf{r}_{\mathrm{prop}}(t_f) - \mathbf{r}_f\right\|_2 - 10^3 \quad [\mathrm{km}] \tag{19a}$$

$$h = \left\|\mathbf{v}_{\mathrm{prop}}(t_f) - \mathbf{v}_f\right\|_2 - 10^0 \quad [\mathrm{m/s}] \tag{19b}$$



(a) Earth–Venus.

(b) Earth–Dionysus transfer.

Fig. 7 Regularized thrust magnitude and components for the considered transfers.

are defined. The chosen values are compatible with state-of-the-art autonomous navigation algorithms errors [25] given the large time of flights of the considered trajectories. The algorithms are considered to find a feasible solution when the condition

$$Q = \max(g, h) < 0 \tag{20}$$

is respected. In the case of the SCP solution, we want to use physically feasible controls (recall that the constraint in Eq. (2d) is not respected outside of the collocation points). Therefore, we integrate the dynamics with the obtained values of $[T_x, T_y, T_z]$ and we define the thrust magnitude as $\|\mathbf{T}\|_2 = \sqrt{T_x^2 + T_y^2 + T_z^2}$ at each time step. We refer to this set of thrust variables as *feasible* controls. Table 2 presents the convergence results. Usually, it is considered that the SCP algorithm reaches convergence when the nonlinear constraints violation at the collocation points of the candidate solution are below a certain threshold [9]. The columns Conv. SCP in the table refer to this type of convergence and it will also be indicated as *SCP*-convergence, whereas the columns Q_{SCP} and Q_{ST} refer to the criterion in Eq. (20).

With regard to the Earth–Venus transfer, the convergence of the standard SCP algorithm is in general low [9]. Surprisingly, the integration of the equations of motion with feasible SCP controls only respected the criterion in Eq. (20) once. On the other hand, the convergence of the shooting method satisfies the convergence criterion in many cases, often outperforming the SCP algorithm convergence itself, meaning that the thrust regularization procedure is robust against not-fully converged SCP solutions. With regard to the Earth–Dionysus transfer, none of the SCP computed (physically feasible) thrust profiles was able to satisfy the constraint in Eq. (20). Notably, the shooting method procedure results in many cases for which the condition in Eq. (20) is respected, even though in this second case the Q-convergence of the thrust regularization is lower than the SCP-convergence of the SCP algorithm itself.

Table 2 Convergence results

Case	Earth–Venus				Earth–Dionysus			
Nr.	Nodes	Conv. SCP, %	Q_{SCP}, %	Q_{ST}, %	Nodes	Conv. SCP, %	Q_{SCP}, %	Q_{ST}, %
1	50	53.0	0.0	35.0	100	82.0	0.0	44.0
2	100	32.0	0.0	53.0	150	76.0	0.0	58.0
3	150	26.0	0.0	44.0	200	79.0	0.0	68.0
4	200	21.0	0.0	43.0	250	77.0	0.0	68.0
5	250	18.0	1.0	33.0	300	80.0	0.0	69.0
Average	150	30.0	0.2	41.6	200	78.8	0.0	61.4

Table 3 Performance results

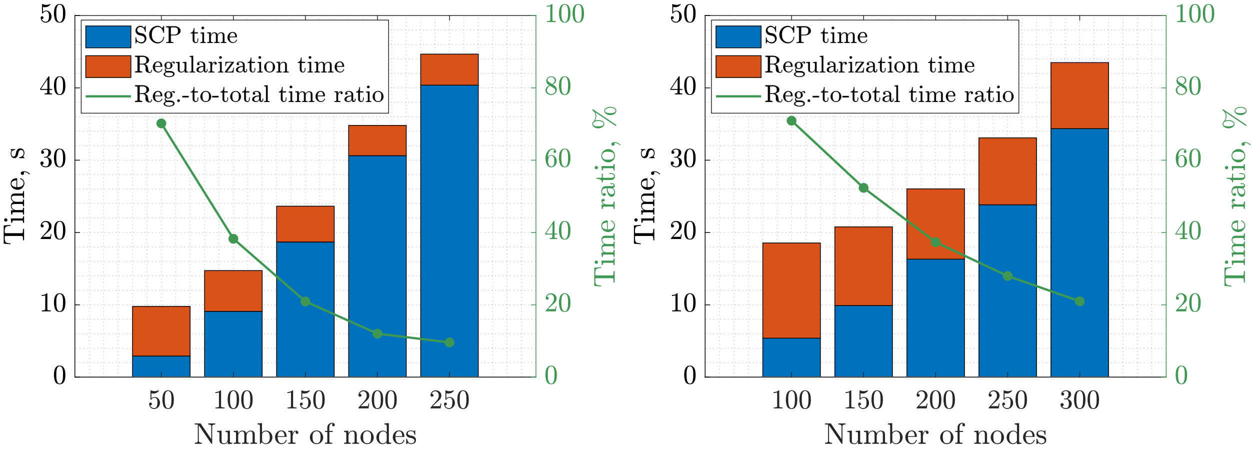
Case	Earth–Venus				Earth–Dionysus			
Nr.	Mass SCP, kg	ρ_m, %	$Time$, s	τ, %	Mass SCP, kg	ρ_m, %	$Time$, s	τ, %
1	1094.1	+1.14	9.8	70.1	2277.5	+0.56	18.6	70.9
2	1186.6	+0.21	14.7	38.3	2265.0	+0.31	20.8	52.4
3	1213.5	+0.22	23.6	20.9	2270.6	+0.17	26.0	37.3
4	1224.3	+0.11	34.8	12.0	2262.9	+0.22	33.1	27.9
5	1205.0	+0.12	44.7	9.6	2294.9	+0.20	43.5	20.9
Average	1184.7	+0.36	25.5	30.2	2274.2	+0.29	28.4	41.9

B. Performance Assessment

Table 3 describes the performance of the proposed strategy in terms of final spacecraft mass and CPU time. The columns ρ_m report the quantity

$$\rho_m = \frac{m_f^{\mathrm{SCP}} - m_f^{\mathrm{ST}}}{m_0} \times 100. \tag{21}$$

The regularization procedure does not affect the optimality of the solutions. In the table, the column $Time$ refers to the total CPU time (SCP & thrust regularization) taken by the algorithm. The column τ indicates instead the ratio between the average time required by the thrust regularization procedure and the total time. The whole procedure (i.e., SCP plus thrust regularization) takes only seconds to converge on the considered machine. The left y-axis of Fig. 8 presents the average CPU times. As the number of nodes increases, the time required by the SCP algorithm increases while the time required by the shooting method decreases and therefore the regularization represents a lower fraction of the total CPU time (as the green lines show). Moreover, note that the sum of the SCP and thrust regularization times for a given number of nodes is often lower (or approximately equal) to the sole SCP time of the following case, especially for the



(a) Earth–Venus.

(b) Earth–Dionysus.

Fig. 8 Average regularization and SCP CPU times and regularization-to-total time ratio.

Earth–Venus case. In conclusion, Table 20 and Fig. 8 show that even though the thrust regularization increases the CPU time of the total procedure for a given number of SCP nodes, it also allows to use less nodes (because of its higher convergence according to Eq. (20) regardless of the value N assumes), and therefore to decrease the actual required CPU time.

V. Conclusions

In this work, a double-layer strategy to solve the low-thrust spacecraft trajectory optimization problem was proposed. The results show that our methodology outperforms the simple SCP optimization in terms of error on the final boundary conditions without any relevant degradation of the optimality of the solution and keeping the computational effort low. This finding is paramount for future autonomous guidance scenarios, because the accuracy of the solution may affect the correct execution of the thrust commands and could therefore strongly influence the amount of propellant required to reach the target celestial body. Our method being a post-processing procedure, the failure of the thrust regularization does not affect the convergence of the SCP algorithm, which can therefore still be used when a regularized solution is not available.

Acknowledgment

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] 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.

[2] Di Domenico, G., Andreis, E., Morelli, A. C., Merisio, G., Franzese, V., Giordano, C., Morselli, A., Panicucci, P., Ferrari, F., and Topputo, F., "The ERC-Funded EXTREMA Project: Achieving Self-Driving Interplanetary CubeSats," *Modeling and Optimization in Space Engineering: New Concepts and Approaches*, Springer, 2022, pp. 167–199. https://doi.org/10.1007/978-3-031-24812-2_6.

[3] 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.

[4] Morelli, A. C., Hofmann, C., and Topputo, F., "Robust Low-Thrust Trajectory Optimization Using Convex Programming and a Homotopic Approach," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 58, No. 3, 2021, pp. 2103–2116. https://doi.org/10.1109/TAES.2021.3128869.

[5] 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.

[6] Schiassi, E., D'Ambrosio, A., Drozd, K., Curti, F., and Furfaro, R., "Physics-informed neural networks for optimal planar orbit transfers," *Journal of Spacecraft and Rockets*, Vol. 59, No. 3, 2022, pp. 834–849. https://doi.org/10.2514/1.A35138.

[7] Boyd, S., and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004, pp. 21–187.

[8] Federici, L., Zavoli, A., and Furfaro, R., "Comparative Analysis of Reinforcement Learning Algorithms for Robust Interplanetary Trajectory Design," *International Conference on Applied Intelligence and Informatics*, Springer, 2022, pp. 133–149. https://doi.org/10.1007/978-3-031-25755-1_9.

[9] 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. https://doi.org/10.2514/1.A35461.

[10] Bonalli, R., Cauligi, A., Bylard, A., and Pavone, M., "GuSTO: Guaranteed Sequential Trajectory optimization via Sequential Convex Programming," *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6741–6747. https://doi.org/10.1109/ICRA.2019.8794205.

[11] Liu, X., Lu, P., and Pan, B., "Survey of convex optimization for aerospace applications," *Astrodynamics*, Vol. 1, No. 1, 2017, pp. 23–40. https://doi.org/10.1007/s42064-017-0003-8.

[12] Gong, Q., Ross, I. M., Kang, W., and Fahroo, F., "Connections between the covector mapping theorem and convergence of pseudospectral methods for optimal control," *Computational Optimization and Applications*, Vol. 41, No. 3, 2008, pp. 307–335. https://doi.org/10.1007/s10589-007-9102-4.

[13] Agamawi, Y. M., Hager, W. W., and Rao, A. V., "Mesh Refinement Method for Solving Bang-Bang Optimal Control Problems Using Direct Collocation," *AIAA Scitech 2020 Forum*, 2020. https://doi.org/10.2514/6.2020-0378.

[14] Tang, G., Jiang, F., and Li, J., "Fuel-Optimal Low-Thrust Trajectory Optimization Using Indirect Method and Successive Convex Programming," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54, No. 4, 2018, pp. 2053–2066. https://doi.org/10.1109/TAES.2018.2803558.

[15] Pei, C., You, S., Dai, R., and Rea, J. R., "A Unified Optimization Algorithm for Bang-bang Optimal Control," *AIAA Scitech 2022 Forum*, 2022. https://doi.org/10.2514/6.2022-0953.

[16] Ricciardi, L. A., and Vasile, M., "Direct transcription of optimal control problems with finite elements on Bernstein basis," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 2, 2019, pp. 229–243. https://doi.org/10.2514/1.G003753.

[17] Dei Tos, D. A., Rasotto, M., Renk, F., and Topputo, F., "LISA Pathfinder mission extension: A feasibility analysis," *Advances in Space Research*, Vol. 63, No. 12, 2019, pp. 3863–3883. https://doi.org/10.1016/j.asr.2019.02.035.

[18] Lu, P., "What is guidance?" *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 7, 2021, pp. 1237–1238. https://doi.org/10.2514/1.G006191.

[19] 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.

[20] Mao, Y., Szmuk, M., and Açıkmeşe, B., "Successive convexification of non-convex optimal control problems and its convergence properties," *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 3636–3641. https://doi.org/10.1109/CDC.2016.7798816.

[21] Scharf, D. P., Açıkmeşe, B., Dueri, D., Benito, J., and Casoliva, J., "Implementation and experimental demonstration of onboard powered-descent guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 213–229. https://doi.org/10.2514/1.G000399.

[22] Franzese, V., Topputo, F., Ankersen, F., and Walker, R., "Deep-space optical navigation for M-ARGO mission," *The Journal of the Astronautical Sciences*, Vol. 68, No. 4, 2021, pp. 1034–1055. https://doi.org/10.1007/s40295-021-00286-9.

[23] 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.

[24] Ross, I. M., "A historical introduction to the convector mapping principle," *Advances in the Astronautical Sciences*, 2006, pp. 1257–1278.

[25] Andreis, E., Franzese, V., and Topputo, F., "Onboard Orbit Determination for Deep-Space CubeSats," *Journal of Guidance, Control, and Dynamics*, 2022, pp. 1466–1479. https://doi.org/10.2514/1.G006294.