(Preprint) AAS 23-186

COMBINED CONVEX AND DIRECT SHOOTING OPTIMIZATION FOR LOW-THRUST TRAJECTORY GENERATION WITH ANALYTICAL THRUST PROFILE

Andrea C. Morelli [*], Alessandro Morselli[†], Carmine Giordano[‡], and Francesco Topputo[§]

Convex optimization for low-thrust trajectory generation guarantees high levels of reliability while keeping the computational time low. However, it is affected by one major disadvantage: constraints are only imposed and respected at the collocation points. Outside of these points, the control commands must be interpolated. Physical constraints on the thrust vector might therefore be violated. Consequently, the output of the algorithm could not be used directly for future autonomous guidance scenarios or for precise onground trajectory design. To address the issue, we develop a fast and reliable double-layer algorithm that exploits convex optimization to find a solution to the low-thrust trajectory optimization problem and transforms its control output into a perfectly bang-off-bang thrust profile where the thrust angles are expressed analytically using unique arbitrary-order polynomials inside each thrust arc, as this ensures that physical thrust constraints are always respected. The proposed strategy would represent an accurate and practical method for future autonomous guidance missions as well as for preliminary onground trajectory design.

INTRODUCTION

Autonomous low-thrust spacecraft guidance implies solving an optimal control problem onboard. This poses several challenges, as low-thrust trajectory optimization is a rather demanding task. Convex optimization[1] represents an interesting direct optimization approach because it provides high levels of reliability while assuring little computational effort.[2–4] It has recently been applied to different aerospace-related problems,[5,6] including powered descent guidance[7] and low-thrust trajectory optimization.[8,9] Still, one major disadvantage of convex optimization is that, like all direct collocation methods, it only provides the solution at the collocation points. This prevents the onboard deployment of the solution coming from this optimization approach *as is*, because interpolation between discretization points must be performed to obtain the solution at any time instant and therefore the physical constraints may be violated, making the required thrust commands infeasible or inaccurate. Moreover, since a finite number of discretization points is used, the switch on and

[*]Ph.D. Candidate, Department of Aerospace Science and Technology, Via La Masa 34, 20156 Milan, Italy. Email: andreacarlo.morelli@polimi.it.
[†]PostDoc Researcher, Department of Aerospace Science and Technology, Via La Masa 34, 20156 Milan, Italy. Email: alessandro.morselli@polimi.it.
[‡]PostDoc Researcher, Department of Aerospace Science and Technology, Via La Masa 34, 20156 Milan, Italy. Email: carmine.giordano@polimi.it.
[§]Full Professor, Department of Aerospace Science and Technology, Via La Masa 34, 20156 Milan, Italy. Email: francesco.topputo@polimi.it.

off times of the thrust arcs might not be exactly defined, and this further complicates the usage of convex optimization-based solutions in future hypothetical autonomous guidance scenarios.

We address the highlighted problems by proposing a double-layer algorithm that relies on convex optimization and exploits a direct shooting method to *regularize* the thrust magnitude profile and its angles such that the latter are expressed as unique arbitrary-order polynomials in each thrust arc and the thrust on and off times are exactly defined. This second layer represents an approach for low-thrust trajectory optimization problems similar to the strategy developed to transform impulsive maneuvers into finite-time ones.[10] Some works have addressed the *regularization* problem for direct trajectory optimization methods using the covector mapping theorem.[11] In particular, Hoffmann and Topputo[8] applied to convex optimization the approach developed by Agamawi et al.[12] for general direct transcription optimization problems. Both approaches use pseudospectral methods and consist of solving the same optimization problem (at least) twice with modified or denser discretization grids such that the thrust on and off times are accurately captured. Although these works solve the issue of exactly capturing the bang-bang structure of the thrust profile, they do not provide unique analytical expressions for the thrust angles in each thrust arc. Moreover, they are tailored for pseudospectral methods as they can exploit the covector mapping theorem, although only few discretization methods have this property. Finally, the aforementioned approaches result in (at least) two optimizations of the *whole* problem, which is not ideal for future hypothetical onboard applications as this may increase the computational burden of the guidance unit. Therefore, the utmost advantages of the proposed method with respect to literature are: first, it offers a fast (when compared to state-of-the-art direct and indirect methods) and reliable way to obtain near-optimal solutions for the low-thrust spacecraft trajectory optimization problem with unique analytical expressions for the thrust variables inside each thrust arc. This is obtained by exploiting the solution of the convex optimization-based layer to identify the thrust and coast arcs, and further only slightly optimizing it. Secondly, our method introduces an analytical procedure based on simple physical considerations to generate the initial guess for the switch on and off times of the thrust arcs. Finally, the regularized thrust profile is found by re-optimizing only simpler parts of the initial problem, thus ensuring a more reliable approach. We test the proposed algorithm in extensive simulations considering an Earth to Venus transfer and using an Hermite–Simpson (HS) discretization[9,13] for the convex optimization algorithm.

PROBLEM STATEMENT

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 (EoM) of such spacecraft can be written as

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

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. μ is the gravitational parameter of the primary body, $\mathbf{T} = [T_x, T_y, T_z]^\top$ is the thrust vector (with $T(t)$ being its euclidean norm), 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 nonlinear and nonconvex, they have to be modified for the problem to be expressed in a convex form. According to literature,[4,14,15] the convexified low-thrust trajectory optimization (LTO) problem reads

$$\underset{\mathbf{u}(t)}{\text{minimize}} \quad -w(t_f) \tag{2a}$$

$$\text{subject to:} \quad \dot{\mathbf{x}}(t) = \mathbf{f}_0(\bar{\mathbf{x}}(t)) + \mathbf{A}(\bar{\mathbf{x}}(t))\left(\mathbf{x}(t) - \bar{\mathbf{x}}(t)\right) + \mathbf{B}\mathbf{u}(t) \tag{2b}$$

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

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

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

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

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

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

$\mathbf{x} = [\mathbf{r}; \mathbf{v}; w]$ and $\mathbf{u} = [\tau_x, \tau_y, \tau_z, \Gamma]^\top = [\boldsymbol{\tau}; \Gamma]$ are the new state and control variables, respectively; in particular, $w(t)$ is the modified mass variable. t_0 and t_f are the initial and final transfer times. T_{max} is the 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 term $\mathbf{f}_0(\bar{\mathbf{x}}(t))$ is the natural two-body dynamics evaluated at the reference trajectory, and the matrices $\mathbf{A}$ and $\mathbf{B}$ are the jacobians of the decoupled dynamics[4] with respect to the state and control, respectively. The original optimal control problem is solved through the sequential convex programming (SCP),[2] an iterative technique that considers a sequence of convex subproblems of the form (2a)–(2h).

The Thrust Regularization Problem

Once the LTO problem has been solved, the optimized control history is available. Regardless of how many discretization points are used, being them a finite number, the thruster switch on and off times might not be captured precisely by direct methods. On top of that, consider Eq. (2d): it expresses the physical constraint on the thrust variables $\sqrt{\tau_x^2(t) + \tau_y^2(t) + \tau_z^2(t)} = \Gamma(t)$. Due to discretization, it is only satisfied at the collocation points t_j, with $j = 1, \dots, N$. In other words, the aforementioned constraint defect is always under the SCP selected convergence threshold at the collocation points, whereas it overcomes the threshold at the points where the thrust variables are interpolated. Since we use the Hermite–Simpson discretization, the thrust variables are linearly interpolated in-between two nodes. According to our simulations, when an acceptable number of collocation nodes N is used, the maximum constraint violation is approximately the $0.5 - 7\%$ (strongly depending on N itself) of the maximum available thrust, i.e., it is in the same order of magnitude of the state-of-the-art low-thrust engines execution errors.[16] This is undesirable, especially for future autonomous guidance missions: the guidance commands have to be as accurate as possible, hopefully several orders of magnitude less than typical real-thrusters execution errors. One may claim that the solution to this problem could be to increase the number of discretization points; the correspondent CPU time, however, also increases significantly, in particular with an higher derivative with respect to the decrease of the error's one.

We address the aforementioned issues by formulating the so-called thrust regularization problem. For all the thrust segments i (with $i = 1, \dots, N_{\text{T}}$) that can be identified in the SCP solution of a low-thrust space trajectory optimization problem, we want to find the exact switch on and off times ($t_{\text{ON}}^{(i)}$ and $t_{\text{OFF}}^{(i)}$, respectively) and to express the thrust angles with polynomials of arbitrary order p, such that the variables T_x, T_y, and T_z can be described with analytical expressions of the form

$$\mathbf{T}^{(i)}(t) = \begin{bmatrix} T_x^{(i)}(t) \\ T_y^{(i)}(t) \\ T_z^{(i)}(t) \end{bmatrix} = T_{\max} \begin{bmatrix} \cos[\beta^{(i)}(t)] \cos[\alpha^{(i)}(t)] \\ \cos[\beta^{(i)}(t)] \sin[\alpha^{(i)}(t)] \\ \sin[\beta^{(i)}(t)] \end{bmatrix}, \tag{3}$$

where

$$\alpha(t) = a_0 + a_1 t + \cdots + a_p t^p \tag{4a}$$

$$\beta(t) = b_0 + b_1 t + \cdots + b_p t^p \tag{4b}$$

are the thrust angles defined as in Figure 1.

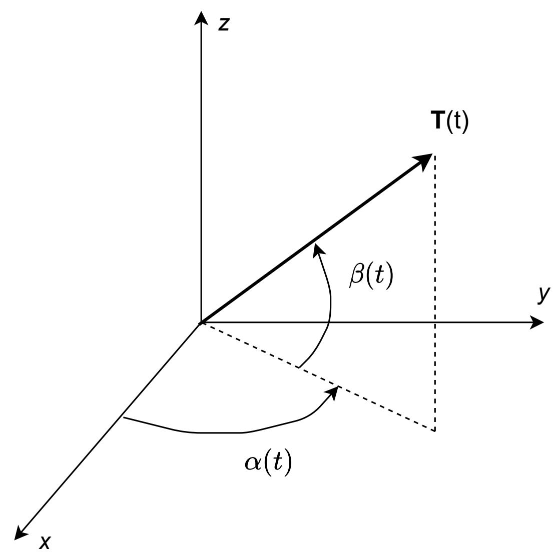

Figure 1: Definition of thrust angles $\alpha(t)$ and $\beta(t)$.

METHODOLOGY

Figure 4 summarizes the main steps of the proposed approach. As the SCP algorithm does not consist in the innovative part of this work, it will not be described in details in the next sections. The interested reader can refer to previous works for a thorough description of the resolution strategy used for the problem in Eqs. (2).[8,9,17] The SCP algorithm can either converge (when the maximum constraints violation is below a certain threshold ε_{SCP} such that $0 < \varepsilon_{\text{SCP}} \ll 1$) or stop without having found a feasible solution. In order to understand the potential of the proposed double-layer approach, we perform the thrust regularization procedure even in the cases for which the SCP algorithm did not fully converge. In other words, the direct shooting optimization is executed even when the maximum constraint violation is lower than a certain other threshold $\varepsilon_{\text{TR}} > \varepsilon_{\text{SCP}}$.

Thrust Arcs Identification

In order to perform the thrust regularization procedure, the thrust arcs of the SCP algorithm solution must be identified, as a different shooting optimization is run for each of them. With the

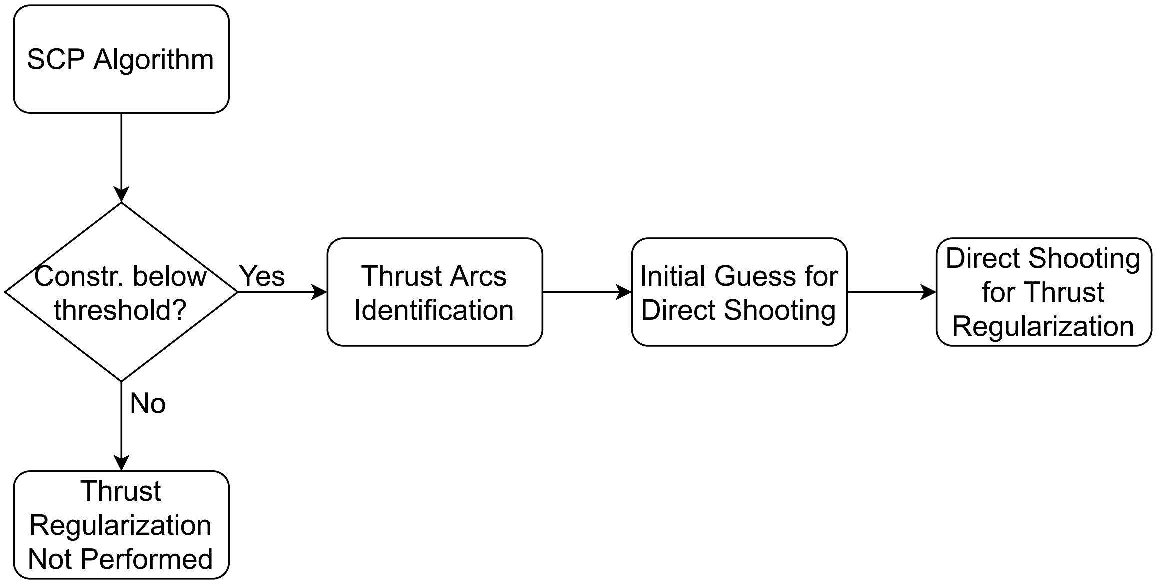

Figure 2: Flowchart of the proposed approach.

expression *thrust arcs identification* we refer to the process of defining the switch on and off times $t_1^{(i)}$ and $t_2^{(i)}$ of each thrust arc $i = 1, \dots, N_\mathrm{T}$. Note that the times $t_1^{(i)}$ and $t_2^{(i)}$ are not exactly defined, as the SCP algorithm is a direct method and therefore it is extremely unlikely that the switching points are placed exactly at a discretization node. Consequently, we consider a threshold ε_T under which the thrust magnitude is considered to be 0. As a result, the times $t_1^{(i)}$ and $t_2^{(i)}$ are artificially defined as the instants when the thrust magnitude becomes higher or lower than the selected threshold, respectively. The outcome of the described process is a *thrust mask*, uniquely defined by the parameters $t_1^{(i)}$ and $t_2^{(i)}$, and N_T.

Initial Guess Generation

Let us now indicate with $t_\mathrm{ON}^{(i)}$ and $t_\mathrm{OFF}^{(i)}$ the optimized values for the switch on and off times associated with the thrust arc i (as found by the shooting method). The direct shooting method requires an initial guess for these times, as well as for the coefficients of the polynomials that describe the thrust angles and the SCP algorithm does not directly provide any initial guess for the thrust angles coefficients. Therefore, we construct a new and more accurate initial guess for the shooting method considering the following assumptions. For each thrust arc i,

1) the total change of spacecraft velocity $\Delta V_\mathrm{SCP}^{(i)}$ as obtained from the SCP algorithm, computed as

$$\Delta V_\mathrm{SCP}^{(i)} = \int_{t_1^{(i)}}^{t_2^{(i)}} \Gamma(t)\,\mathrm{d}t, \tag{5}$$

must be equal to the total change of the spacecraft's velocity $\Delta V^{(i)}$ associated with the initial guess thrust profile, such that

$$\Delta V^{(i)} = \int_{t_a^{(i)}}^{t_b^{(i)}} \Gamma(t)\,\mathrm{d}t = \Delta V_\mathrm{SCP}^{(i)}, \tag{6}$$

where $t_a^{(i)}$ and $t_b^{(i)}$ are the switch on and off times of the thrust arc i associated with the new initial guess.

2) the barycentric time $\underline{t}_{\text{SCP}}^{(i)}$ of the SCP solution, defined as

$$\underline{t}_{\text{SCP}}^{(i)} = \frac{\displaystyle\int_{t_1^{(i)}}^{t_2^{(i)}} t\Gamma(t)\,\mathrm{d}t}{\displaystyle\int_{t_1^{(i)}}^{t_2^{(i)}} \Gamma(t)\,\mathrm{d}t}, \tag{7}$$

must be equal to the barycentric time $\underline{t}_c^{(i)}$ associated with the thrust profile of the new initial guess.

3) The thrust magnitude is always equal to the maximum value T_{max}.

From the aforementioned considerations, a system of equations can be derived and solved, where the only unknowns are $t_a^{(i)}$ and $t_b^{(i)}$, i.e., the most accurate initial guesses for the times $t_{\text{ON}}^{(i)}$ and $t_{\text{OFF}}^{(i)}$. Once the initial guess for the thrust on and off times has been defined, the initial guess for the thrust angles coefficients must be taken care of. First, the thrust variables of the SCP algorithm $[\tau_x, \tau_y, \tau_z, \Gamma]$ are transformed into the canonical thrust variables[14] $[T_x, T_y.T_z, \|\mathbf{T}\|_2]$; after that, a change of coordinates is performed to obtain the values of the thrust angles at the SCP collocation points. Finally, a least-square polynomial interpolation is executed to obtain the coefficients of the thrust angles polynomials.

Direct Shooting Optimization

Figure 3 shows the scheme of the comparison between the thrust profile as obtained by the SCP algorithm and the regularized thrust profile. The red filled circles represent the thrust magnitude at the nodal points (recall that in the HS discretization, the thrust variables are interpolated linearly in each trajectory segment). The previously-defined instants $t_1^{(i)}$, $t_{\text{ON}}^{(i)}$, $t_{\text{OFF}}^{(i)}$, and $t_2^{(i)}$ identify three time intervals. The idea of the proposed strategy is summarized in the following point and performed for each thrust arc.

- First, starting from the spacecraft state as found by the SCP algorithm and correspondent to the time $t_1^{(i)}$, the natural spacecraft dynamics is propagated from $t_1^{(i)}$ to $t_{\text{ON}}^{(i)}$;
- then, the low-thrust dynamics in Eq. (1) with the regularized thrust variables is propagated from $t_{\text{ON}}^{(i)}$ to $t_{\text{OFF}}^{(i)}$;
- finally, the natural dynamics is further propagated from $t_{\text{OFF}}^{(i)}$ to $t_2^{(i)}$;
- the algorithm stops when the spacecraft state at time $t_2^{(i)}$ as obtained by the shooting method is equal (up to a predefined user-selected threshold) to the spacecraft state at time $t_2^{(i)}$ as computed by the SCP algorithm.

Therefore, the idea at the basis of our approach is to impose that the spacecraft state before and after each thrust arc is the same before and after the thrust regularization procedure. This is eventually

done by adjusting the values for the true switch on and off times as well as the values for the thrust angles coefficients. Note that the procedure is executed separately for each thrust arc; this means that the user can choose whether to perform the regularization for the whole thrust profile or not. As already previously mentioned, this is of great advantage when considering future hypothetical autonomous guidance mission scenarios, as the computational burden of the simple SCP algorithm is only slightly increased.
Figure 4 summarizes the strategy. In the flowchart, k indicates the shooting algorithm iterations, and $\mathbf{a}^{(i)}$ and $\mathbf{b}^{(i)}$ are vectors collecting the thrust angles coefficients $\alpha(t)$ and $\beta(t)$, respectively. Moreover, Δs indicates the elapsed time since the beginning of the optimization, and ε_{ST}, s_{max}, and k_{max} are the constraints, elapsed time, and iterations thresholds. In fact, to limit the overall duration of the computation, each shooting optimization terminates when either the constraints are satisfied or a maximum number of solver iterations or elapsed time overcome a certain threshold.

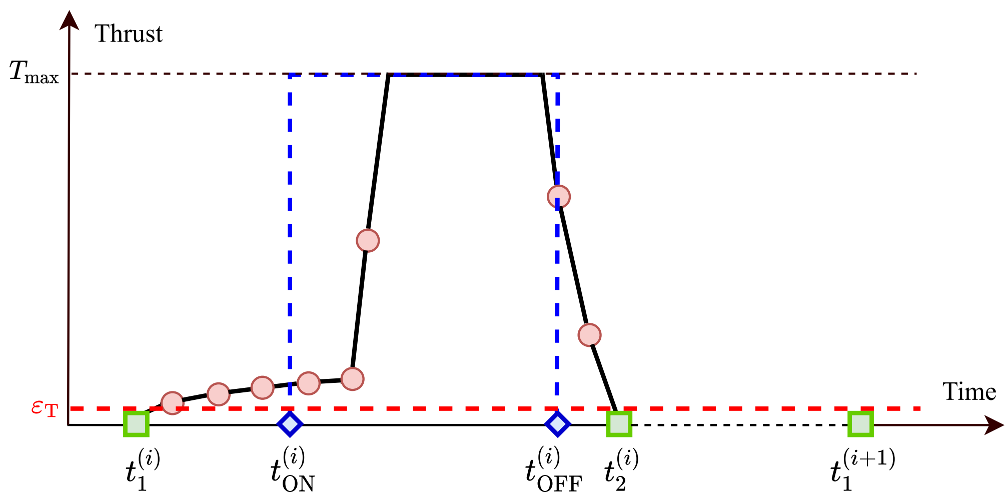

Figure 3: Comparison of SCP (black solid line) and regularized (blue dashed line) thrust profiles.

NUMERICAL SIMULATIONS

The performance of the proposed algorithm is assessed in several simulations where we want to regularize the entire thrust profile. The Earth-Venus transfer is considered as test case. Table 1 shows the transfer parameters. The normalization factors are LU = 1 Astronomical Unit for position variables and VU $= \sqrt{\mu/\text{LU}}$ for velocities.
To address the quality of the proposed algorithm, we perform the following analysis. We select different number of discretization nodes and we run 100 simulations for each of them, providing the SCP algorithm with simple, perturbed cubic interpolation-based initial guesses.[4] The objective of the analysis is to compare the convergence, the obtained final mass, and the CPU time required by the simple SCP algorithm and the shooting method procedure. As previously stated, 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 did not satisfy the convergence criteria. This is to check whether the proposed methodology can increase the convergence of the standard SCP. Table 2 presents the values for all the parameters of the proposed approach.
To assess whether the SCP algorithm or thrust regularization return a feasible solution, the follow-

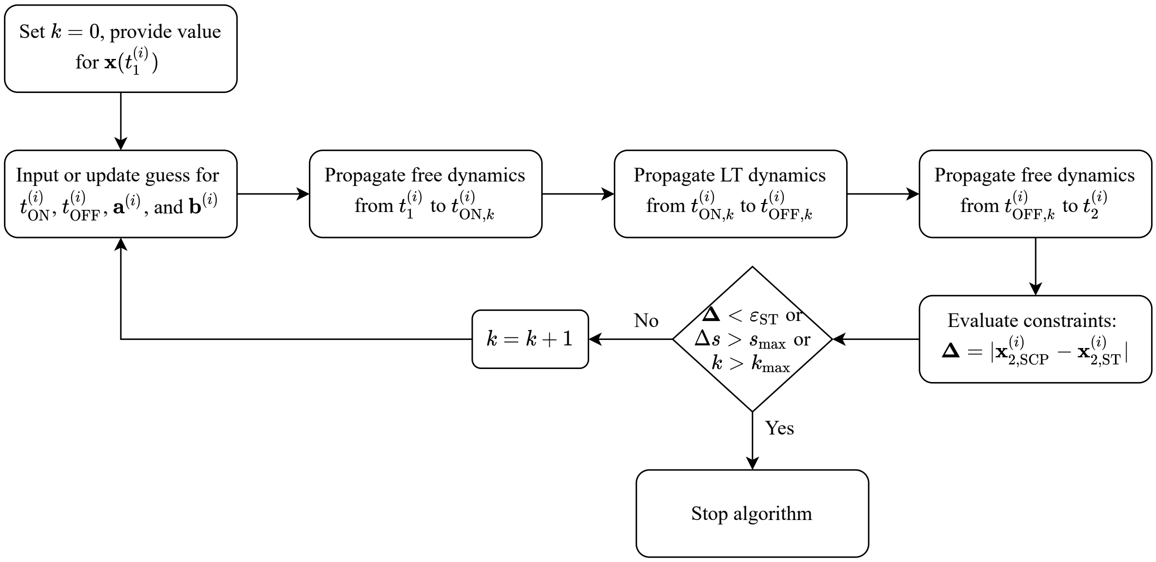

Figure 4: Flowchart of the proposed approach.

Table 1: Simulation values for the Earth–Venus transfer.[18]

Parameter	Earth–Venus
$\mathbf{r}_0$, LU	$[0.97083220, 0.23758440, -1.67106 \times 10^{-6}]^\top$
$\mathbf{v}_0$, VU	$[-0.25453902, 0.96865497, 1.50402 \times 10^{-5}]^\top$
m_0, kg	1500
$\mathbf{r}_f$, LU	$[-0.32771780, 0.63891720, 0.02765929]^\top$
$\mathbf{v}_f$, VU	$[-1.05087702, -0.54356747, 0.05320953]^\top$
m_f, kg	free
$T_{\max}$, N	0.33
I_{sp}, s	3800
t_f, days	1000

ing converge criterion is used: we propagate the equations of motion in Eq. (1) with the obtained controls and we check whether *both* the conditions

$$\left\|\mathbf{r}_{\text{prop}}(t_f) - \mathbf{r}_f\right\| < 1000\,\text{km} \tag{8a}$$
$$\left\|\mathbf{v}_{\text{prop}}(t_f) - \mathbf{v}_f\right\| < 1\,\text{m/s} \tag{8b}$$

are respected, where $(\cdot)_{\text{prop}}$ indicates te propagated dynamics. This is because we want to understand whether the obtained solution would be good enough to be run onboard to get the spacecraft to the target celestial body. The selected threshold values are deemed acceptable for this kind of analysis, considering the long times of flight of the selected transfer.

Note that in the case of the SCP solution, we want to use physically feasible controls at any time instant (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 vari-

Table 2: Parameters for the thrust regularization algorithm.

Parameter	Value
Shooting convergence threshold ε_{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
α and β polynomials order p	2
SCP convergence threshold ε_{SCP}	1×10^{-6}
Thrust regularization flag threshold ε_{ST}	1×10^{-4}

Table 3: Convergence results for the Earth-Venus transfer.

Nodes	Conv. SCP alg., %	Conv. SCP integr., %	Conv. shoot., %
100	32.0	0.0	53.0
150	26.0	0.0	44.0
200	21.0	0.0	43.0

ables as *feasible* controls.

Table 3 presents the convergence results. The column *Conv. SCP alg.* refers to the cases for which the SCP algorithm's convergence criteria in Ref.[4] were satisfied (i.e., maximum violation of the constraints *at the nodes* under a predefined threshold ε_{SCP} as reported in Table 2), whereas the column *Conv. SCP integr.* refers to the criteria in Eqs. (8). Results show an undisputed superiority of the thrust regularization procedure in combination with SCP with respect to simply using the SCP algorithm. Notably, in none of the cases the SCP algorithm was able to ensure a final error lower than the desired threshold when feasible controls are considered. Therefore, if directly executed, the solution coming from SCP would soon result in large displacements from the planned trajectories. Simulations also highlight how the optimality of the post-processed solution is not significantly affected (the difference is $< 1\%$ with respect to the initial spacecraft mass).

Moreover, the CPU time required by the regularization procedure is only a fraction of the overall time (usually in the order of $10 - 15\%$), which is almost always lower than 1 minute on an Intel Core i7-10510U CPU @ 1.80GHz. Once again, this aspect proves that the implemented strategy does not substantially increase the computational burden of the simple SCP algorithm. On top of that, also consider that in autonomous guidance scenarios, only the first portion of the thrust profile is actually executed as a new reference trajectory to follow would be computed again after navigation operations are performed. Consequently, the computational impact of the thrust regularization is even lower with respect to what has been said few lines above.

Finally, Fig. 5 shows an example of regularized thrust profile. Although the SCP profiles poorly capture the bang-off-bang structure due to the small number of nodes, the regularization procedure was successfully completed.

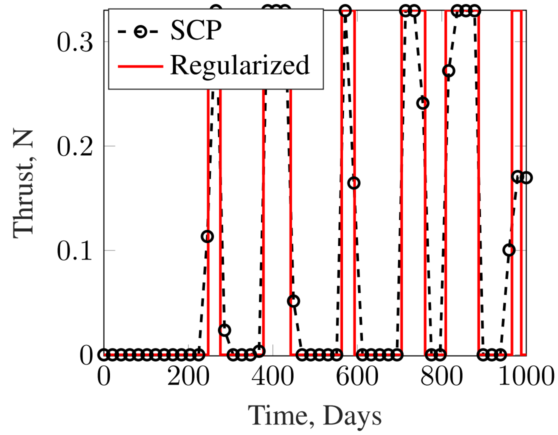

Figure 5: Comparison of thrust profiles as obtained by the SCP algorithm with 50 nodes and after regularization for the Earth-Venus transfer.

CONCLUSIONS

This work presents a double-layer algorithm based on convex optimization to obtain precise and regularized thrust commands for low-thrust spacecraft transfers. It locates in the context of the research aimed to shift flight-related tasks such as spacecraft guidance directly onboard. Results show that the proposed approach is efficient both in terms of convergence and required computational time.

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] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

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

[3] R. Bonalli, A. Cauligi, A. Bylard, and M. Pavone, "Gusto: Guaranteed sequential trajectory optimization via sequential convex programming," *2019 International conference on robotics and automation (ICRA)*, IEEE, 2019, pp. 6741–6747.

[4] C. Hofmann, A. C. Morelli, and F. Topputo, "On the Performance of Discretization and Trust-Region Methods for On-Board Convex Low-Thrust Trajectory Optimization," *AIAA Scitech 2022 Forum*, 2022, p. 1892.

[5] D. Malyuta, Y. Yu, P. Elango, and B. Açıkmeşe, "Advances in trajectory optimization for space vehicle control," *Annual Reviews in Control*, Vol. 52, 2021, pp. 282–315.

[6] X. Liu, P. Lu, and B. Pan, "Survey of convex optimization for aerospace applications," *Astrodynamics*, Vol. 1, Sep. 2017, pp. 23–40, 10.1007/s42064-017-0003-8.

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

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

[9] A. C. Morelli, C. Hofmann, and F. Topputo, "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, 10.1109/TAES.2021.3128869.

[10] D. A. Dei Tos, M. Rasotto, F. Renk, and F. Topputo, "LISA Pathfinder mission extension: A feasibility analysis," *Advances in Space Research*, Vol. 63, No. 12, 2019, pp. 3863–3883.

[11] Q. Gong, I. M. Ross, W. Kang, and F. Fahroo, "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.

[12] Y. M. Agamawi, W. W. Hager, and A. V. Rao, "Mesh refinement method for solving bang-bang optimal control problems using direct collocation," *AIAA Scitech 2020 Forum*, 2020, p. 0378.

[13] P. Williams, "Hermite–Legendre–Gauss–Lobatto Direct Transcription in Trajectory Optimization," *Journal of Guidance, Navigation, and Control*, Vol. 32, Aug. 2009, pp. 1392–1395, 10.2514/1.42731.

[14] Z. Wang and M. J. Grant, "Minimum-Fuel Low-Thrust Transfers for Spacecraft: A Convex Approach," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54, Mar. 2018, pp. 2274–2290, 10.1109/TAES.2018.2812558.

[15] Z. Wang and M. J. Grant, "Optimization of minimum-time low-thrust transfers using convex programming," *Journal of Spacecraft and Rockets*, Vol. 55, No. 3, 2018, pp. 586–598.

[16] V. Franzese, F. Topputo, F. Ankersen, and R. Walker, "Deep-space optical navigation for M-ARGO mission," *The Journal of the Astronautical Sciences*, Vol. 68, No. 4, 2021, pp. 1034–1055.

[17] C. Hofmann, A. C. Morelli, and F. Topputo, "Performance Assessment of Convex Low-Thrust Trajectory Optimization Methods," *Journal of Spacecraft and Rockets*, 2022. In Advance, 10.2514/1.A35461.

[18] F. Jiang, H. Baoyin, and J. Li, "Practical Techniques for Low-Thrust Trajectory Optimization with Homotopic Approach," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 1, 2012, 10.2514/1.52476.