A. Mannocchi, C. Giordano, F. Topputo
Control-Constrained Indirect Optimization for Low-Thrust Trajectories with Duty Cycles
Journal of Guidance Control and Dynamics, Vol. 49, N. 5, p. 1369-1387, 2026 (published online 12/01/2026)
doi:10.2514/1.G008697

Control-constrained Indirect Optimization for Low-thrust Trajectories with Duty Cycles

Alessandra Mannocchi *, Carmine Giordano †, and Francesco Topputo ‡
Politecnico di Milano, Via La Masa 34, Milan, 20156, Italy

A new paradigm for space exploration is expected in the next years: deep-space missions will be performed in the future by several electric-propulsion-based missions. Fuel-optimal low-thrust trajectories show very long thrusting periods, due to the solution of the necessary conditions for optimality. These are not the best options when operational compliance is taken into account in mission design: coasting periods allow for orbit determination, science, telecommunications, and other space operations. This is particularly true for CubeSats, having limited thrusting, power, and pointing budgets. For this reason, electric-based spacecraft will be required to follow operational compliant trajectories: transfers with an alternation of coasting and thrusting periods imposed at pre-defined time instants, called duty cycles. Traditional trajectory optimization algorithms exhibit convergence problems when handling this kind of discontinuous constraint. In this work, an efficient and robust indirect method to compute operational compliant trajectories is formulated through the imposition of coasting arcs as control constraints and a triple continuation scheme. The algorithm is applied to over 50,000 trajectories to demonstrate its consistency and robustness, and to study the effects of duty cycles imposition on low-thrust trajectories. Results show that the operational compliant solutions can be computed in a fast and accurate way and that they are similar to the fuel-optimal ones with no operational constraint, both in terms of thrusting profile and propellant mass.

I. Introduction

SPACE sector will see a remarkable surge in activity in the next years. A significant role is played by deep-space exploration, where a new paradigm for space exploration is expected. An increasing number of interplanetary missions will be performed exploiting electric propulsion, as this technology allows greater flexibility in possible transfers, in addition to a great reduction in propellant mass due to its higher values of specific impulse, with respect to chemical propulsion [1]. A great example of this trend is the BepiColombo mission [2], launched in 2018 and now

Presented as Paper AIAA 2024-0631 at the 2024 SciTech Forum in Orlando, FL, 8–12 January 2024
*PhD Student, Department of Aerospace Science and Technology (DAER). AIAA member. Corresponding author: alessandra.mannocchi@polimi.it
†PostDoc Fellow, Department of Aerospace Science and Technology (DAER). AIAA member.
‡Full Professor, Department of Aerospace Science and Technology (DAER). AIAA senior member.

traveling towards Mercury. Electric propulsion is a good candidate also for CubeSats [3], a technology that has rapidly developed in the last years [4]. Several released-in-situ and/or deep-space CubeSat missions are expected to be launched in the next years by ESA (e.g., LUMIO [5], Milani [6], Juventas [7]). These shoebox-sized miniaturized satellites have stringent budgets in terms of thrusting, power, and pointing resources [8]. However, thanks to their low design, manufacturing, and launch costs [9], they have enabled parties not supported by huge capital to venture into space. By relying on off-the-shelf components, CubeSats also need less testing and validation procedures, cutting down costs and time-to-flight even more. For these reasons, deep-space stand-alone interplanetary CubeSats, probes able to travel to their final destination without the need of a carrying mothership, are expected to soon permeate the inner Solar System.

Two are the main drawbacks related to low-thrust missions. The first one is the requirement of a significant effort from on-ground flight dynamics, with regularly scheduled navigation and guidance operations. The second one is the noise for the spacecraft dynamics that can negatively affect the orbit determination from Earth. Electric propulsion trajectories are traditionally designed by formulating a low-thrust optimal trajectory problem (LOTP) [10, 11], a specialization of the optimal control problem (OCP) for time-continuous systems. The optimal solutions in terms of propellant mass selected for flying missions usually require the spacecraft to thrust for very long periods, with intermediate periods of coasting, whose length cannot be known or selected a priori, depending only on the optimality of the solution. These long thrusting periods introduce noise in the orbit determination, and, in the case of stand-alone interplanetary CubeSats, they prevent communications to the ground. With their limited budgets indeed, CubeSats are unable to perform operations, such as communication, ground-based orbit determination, and correction, or scientific experiments, while thrusting. To overcome these issues, electric-based spacecraft will be required to follow operational-compliant (OC) trajectories, consisting of a repetition of a regular pattern of alternating thrusting and coasting arcs, called duty cycles. OC trajectories can be designed in any way necessary to ease ground operations, for example miming the working week (e.g., 7 days duty cycles, with 6 days of thrust and 1 day of coasting).

No analytic solutions exist for the LOTP, but several numerical techniques, traditionally and mainly divided into direct and indirect methods [12], have been developed through the years to solve it. Direct methods [13–15] transcribe the OCP into a parameter optimization problem, and then use nonlinear programming to find the optimal solution. Direct methods are usually more flexible in the introduction of constraints, but require a higher number of nodes to properly describe the trajectory, and can lead easily to sub-optimal trajectories for this reason. Indirect methods [16, 17] aim at finding the solution of the necessary optimality conditions derived by the calculus of variations. They require the derivation of these equations and, having a small convergence region, require also a good initial guess for the solution process. Nevertheless, due to their nature, they lead to the exact solution of the OCP. The solution method of both formulations leverages the first, and sometimes second, order derivatives of the problem, required to be differentiable at least up to the first order [18, 19].

OC trajectories are subjected to time-dependent and discontinuous constraints, and for this reason, they are not

straightforward to be optimized through both these methods. The presence of duty cycles is then usually ignored in the preliminary phases of trajectory design. The problem of duty cycles in these phases is addressed with optimizations with only a limited percentage of the available maximum thrust - usually 80 to 90% - to leave space, in the following phases, for the introduction of these constraints.

The homotopy, or continuation, technique is a class of methods conceived to deal with discontinuous structures in the LOTP. They allow the solution of the original, difficult, and discontinuous OCP, starting from an easier one and gradually introducing complexity in the problem [20]. It is usually employed in indirect methods to overcome discontinuity problems as bang-bang control in fuel optimal solutions [21, 22], to impose varying maximum thrust models [23] and shutdown of the thruster due to eclipses [24]. Recently, the problem of forced coasting arcs imposition has become more and more relevant, and homotopic techniques have been employed to solve it in both indirect [25, 26] and direct formulations [27].

In this work, an innovative indirect formulation based on the modeling of the alternation of duty cycles in OC trajectories as control constraints is proposed. A triple continuation scheme is proposed to gradually increase the number of coasting arcs in the trajectory. The mathematical formulation of the constraint is derived and formalized, and coupled with an accurate integrating method capable of accurately describing the OC trajectory. In contrast to previously developed methods, this work introduces a straightforward mathematical formulation of the duty-cycle constraint, facilitating the derivation of the necessary conditions for the optimality of the indirect problem. Furthermore, the proposed triple continuation scheme enables the exact resolution of these conditions, thereby enabling the computation of the exact optimal solution without approximations. The solution method also relies on the use of analytical derivatives, which are computed and reported in the work, and a switching times detection technique, rapidly generating OC low-thrust transfers. This work proposes a method capable of modeling duty cycles with duration of any kind without any prior knowledge of the structure of the control law, obtaining as a result a throttle factor profile that mimics a fuel-optimal solution, i.e., showing thrust blocks close/around the location of the ones in the fuel-optimal solution. The optimality of the method is checked against a direct method previously developed by the authors [27]. The algorithm is used to solve optimal trajectories in two scenarios, the M-ARGO CubeSats one [28] and the RAMSES mission towards Apophis one [29], to compute about 50,000 trajectories and demonstrate its consistency and robustness, and to test its capabilities. The algorithm is shown to be able to compute OC trajectories with thrusting profiles similar to the solutions without the duty cycle imposition, with a limited increase in propellant mass. In summary, the main contributions of this paper are the following: firstly, the mathematical derivation of the necessary conditions for optimality for the time-dependent operational compliance through a control constraint formulation, conditions not derived by other solutions proposed to the problem [26]. Secondly, an updated version of the integration schemes proposed by the authors in [23, 24] to catch the dynamics of the problem. Finally, the paper presents the computations of actual low-thrust missions [28, 29] to precisely indicate their costs when duty cycles are foreseen in the mission analysis. This overcomes

the current limitations of their estimation according to the state of the art with a limit on the available maximum thrust and provides a reference for future electric-propulsion-based missions.

The paper is organized as follows. In Section II the optimal OC problem is stated and formalized. The necessary conditions for optimality and the conditions for the control constraint formulation are derived and presented. In Section III the methods exploited to solve the problem and ease the convergence are presented. These comprehend the model of the duty cycle switching function, the triple continuation scheme, as well as the integration flowchart. In Section IV, the method is applied to both M-ARGO CubeSat and RAMSES mission scenarios, to compute a large number of trajectories to test the performance of the method. Finally, in Section V, a critical discussion of the method proposed is presented.

II. Operational Compliant Optimal Control Problem

In this work, the heliocentric phase of an interplanetary orbit transfer is considered. The equations of the dynamics of a spacecraft in cartesian coordinates in an interplanetary two-body problem are

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \Rightarrow \begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = \mathbf{g}(\mathbf{r}) + u\frac{T_{max}}{m}\boldsymbol{\alpha} \\ \dot{m} = -u\frac{T_{max}}{I_{sp}g_0} \end{cases} \tag{1}$$

where $\mathbf{g}(\mathbf{r}) = -\mu\mathbf{r}/r^3$ is the gravitational acceleration and r is the norm of the position vector. Moreover, $\mathbf{r} = [x,\ y,\ z]$, $\mathbf{v} = [v_x,\ v_y,\ v_z]$ and m are respectively the spacecraft position, velocity, and mass included in the state vector $\mathbf{x} = [\mathbf{r},\ \mathbf{v},\ m]$. The control is provided by the unit vector $\boldsymbol{\alpha}(t)$, the thrust direction vector, and $u(t) \in [0, 1]$, the thrust throttle factor being 0 when the thruster is switched off, and 1 when it is fully on, such that $\mathbf{u}(t) = [\boldsymbol{\alpha}(t),\ u(t)]$. The values μ, g_0, I_{sp}, and T_{max} are the gravitational constant of the Sun, the Earth's gravitational acceleration at sea level, the specific impulse, and the maximum thrust, respectively.

In OC trajectory design the final goal is to switch off the electric engine according to a pre-defined duty cycle time-based schedule. The additional coasting arcs retrieved along the computed trajectories allow for easier operations from the ground and provide additional time for orbit determination and opportunistic science. To capture the logic of the enforced coasting arcs, a duty cycle switching function $S_{DC} = S_{DC}(t)$, is defined such that

$$\text{if} \quad S_{DC} < \epsilon_{DC} \quad \text{then} \quad u = 0 \tag{2}$$

$$\text{if} \quad S_{DC} \geq \epsilon_{DC} \quad \text{then} \quad u \in [0, 1] \tag{3}$$

where ϵ_{DC} is a constant pre-defined threshold. Assuming that the objective is to save the propellant mass m_p, the LOTP considered is a fuel-optimal (FO) problem [30], and for the rendezvous boundary conditions it reads

$$\underset{\mathbf{u}(t)\in U}{\text{minimize}} \quad J_f = \int_{t_i}^{t_f} L_f(\mathbf{x},\mathbf{u})\,\mathrm{d}t = \int_{t_i}^{t_f} \frac{T_{max}}{I_{sp}g_0} u\,\mathrm{d}t \tag{4}$$

$$\text{subject to} \quad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u}) \tag{5}$$

$$\mathbf{r}\,(t_i) = \mathbf{r}_i; \quad \mathbf{v}\,(t_i) = \mathbf{v}_i \quad \text{and} \quad m\,(t_i) = m_i \tag{6}$$

$$\mathbf{r}\left(t_f\right) = \mathbf{r}_T\left(t_f\right) \quad \text{and} \quad \mathbf{v}\left(t_f\right) = \mathbf{v}_T\left(t_f\right) \tag{7}$$

where $\mathbf{r}_i$, $\mathbf{v}_i$ and m_i are the prescribed initial conditions of the probes $\mathbf{x}_i = [\mathbf{r}_i,\, \mathbf{v}_i,\, m_i]$, and $\mathbf{r}_T(t_f)$ and $\mathbf{v}_T(t_f)$ are the known position and velocity of the target at the final time, which in the case considered is fixed. U is the set of admissible controls and, for this problem, it requires $\mathbf{u}(t)$ to be piece-wise continuous, i.e., $\mathbf{u}(t) \in \hat{\mathbb{C}}^0$, and to be bounded, as for the definition of the throttle factor. The FO problem aims at computing the optimal control function $\mathbf{u}^* = [\boldsymbol{\alpha}^*,\, u^*]$ that minimizes J_f under the boundary conditions at the initial and final times, t_i and t_f respectively, and other user-defined constraints. To do so, the necessary conditions for optimality are derived by introducing the Lagrange multipliers, or costates, $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_r,\, \boldsymbol{\lambda}_v,\, \lambda_m]$ associated with the states. To introduce the control constraint of an OC trajectory, expressed in Eqs. (2)- (3), a control constraint is considered as

$$C(u,t) = u\frac{T_{\max}}{I_{\mathrm{sp}}g_0}\max(0,\, \epsilon_{\mathrm{DC}} - S_{\mathrm{DC}}(t)) = 0 \tag{8}$$

Note that the constraint depends only on the throttle factor u and on the time. When $S_{\mathrm{DC}} \geq \epsilon_{\mathrm{DC}}$, the max function is 0, giving the freedom to the throttle factor u to be different from 0. When instead $S_{\mathrm{DC}} < \epsilon_{\mathrm{DC}}$, the max function is equal to $(\epsilon_{\mathrm{DC}} - S_{\mathrm{DC}})$. In this latter case, the throttle factor u has to be 0 to respect the constraint in Eq. (8). Assume that $(\epsilon_{\mathrm{DC}} - S_{\mathrm{DC}})$ is greater than 0: the need is to design S_{DC} according to this; it will be done and shown in Section III.A. This constraint corresponds to the throttle factor respecting

$$u = \begin{cases} 0 & \text{if} \quad S_{DC} < \epsilon_{DC} \\ [0,\, 1] & \text{otherwise} \end{cases} \tag{9}$$

To treat this constraint, it is adjoint to the Hamiltonian of the problem with a Lagrange multiplier $\mu(t)$. The Hamiltonian function is then defined as

$$H_f = L_f + \boldsymbol{\lambda}^T \cdot \mathbf{f} + \mu C = \boldsymbol{\lambda}_r \cdot \mathbf{v} + \boldsymbol{\lambda}_v \cdot \left(\mathbf{g}(\mathbf{r}) + u\frac{T_{max}}{m}\boldsymbol{\alpha}\right) + \lambda_m\left(-u\frac{T_{max}}{I_{sp}g_0}\right) + u\frac{T_{max}}{I_{sp}g_0} + \mu u\frac{T_{\max}}{I_{\mathrm{sp}}g_0}\max(0,\, \epsilon_{\mathrm{DC}} - S_{\mathrm{DC}}) \tag{10}$$

The Pontryagin minimum principle (PMP) [31], states that the control $\mathbf{u}^*$, to be optimal, has to minimize H_f at any time

instant. Accordingly, the optimal thrust direction $\boldsymbol{\alpha}^*$ is chosen as [30, 32]

$$\boldsymbol{\alpha}^* = -\frac{\boldsymbol{\lambda}_v}{\lambda_v} \quad \text{if} \quad \lambda_v \neq 0 \tag{11}$$

where λ_v is the norm of $\boldsymbol{\lambda}_v$. Introducing the optimal thrust direction $\boldsymbol{\alpha}^*$, the Hamiltonian results in

$$H_f = \boldsymbol{\lambda}_r \cdot \mathbf{v} + \boldsymbol{\lambda}_v \cdot \mathbf{g}(\mathbf{r}) + u\frac{T_{\max}}{I_{\text{sp}} g_0}\left[S_f + \mu \max(0,\, \epsilon_{\text{DC}} - S_{\text{DC}})\right] \tag{12}$$

where S_f is defined as the FO switching function, $S_f = 1 - \lambda_m - \lambda_v \frac{I_{\text{sp}} g_0}{m}$. According to the PMP, the optimal throttle factor u^* has to be selected in order to minimize the Hamiltonian H_f at each time instant, i.e., $\frac{\delta H_f}{\delta u} = 0$. To derive its law, consider the following cases:

1) $S_f > 0$ and $S_{\text{DC}} \geq \epsilon_{\text{DC}}$: in this case $u^* = 0$, with no constraints on μ;
2) $S_f > 0$ and $S_{\text{DC}} < \epsilon_{\text{DC}}$: in this case $u^* = 0$, with the Lagrangian multiplier being $\mu > 0$;
3) $S_f < 0$ and $S_{\text{DC}} \geq \epsilon_{\text{DC}}$: in this case $u^* = 1$, with no constraints on μ;
4) $S_f < 0$ and $S_{\text{DC}} < \epsilon_{\text{DC}}$: in this case $u^* = 0$, with the multiplier being $\mu > \frac{|S_f|}{\epsilon_{\text{DC}} - S_{\text{DC}}} > 0$;
5) $S_f = 0$ and $S_{\text{DC}} \geq \epsilon_{\text{DC}}$: in this case $u^* \in [0, 1]$, with no constraints on μ;
6) $S_f = 0$ and $S_{\text{DC}} < \epsilon_{\text{DC}}$: in this case $u^* = 0$, with the Lagrangian multiplier being $\mu > 0$.

Ultimately, these cases lead to the optimal control law

$$u^* = \begin{cases} 0 & \text{if} \quad S_f > 0 \quad \text{or} \quad S_{DC} < \epsilon_{DC} \\ \in [0,1] & \text{if} \quad S_f = 0 \quad \text{and} \quad S_{DC} \geq \epsilon_{DC} \\ 1 & \text{if} \quad S_f < 0 \quad \text{and} \quad S_{DC} \geq \epsilon_{DC} \end{cases} \tag{13}$$

According to Eq. (13), the optimal control of a FO solution exhibits a bang-bang, discontinuous profile [10], which makes the convergence of the solver difficult. A homotopy technique was introduced in [21] to smooth the control profile, and gradually introduce this discontinuity into the problem. The idea is to solve a LOTP having the following objective function

$$J_\epsilon = \int_{t_i}^{t_f} L_\epsilon(\mathbf{x}, \mathbf{u})\, \mathrm{d}t = \int_{t_i}^{t_f} \frac{T_{max}}{I_{sp} g_0}\left[u - \epsilon u(1-u)\right] \mathrm{d}t \tag{14}$$

where $\epsilon = 1$ corresponds to an energy optimal (EO) problem, and $\epsilon = 0$ corresponds to a FO problem. The problem is solved iteratively, gradually reducing ϵ and providing as an initial guess the solution to the previous problem, until $\epsilon = 0$

is reached. The Hamiltonian of these auxiliary problems reads

$$H_\epsilon = L_\epsilon + \boldsymbol{\lambda}^T \cdot \mathbf{f} + \mu C = \boldsymbol{\lambda}_r \cdot \mathbf{v} + \boldsymbol{\lambda}_v \cdot \mathbf{g}(\mathbf{r}) + u \frac{T_{\max}}{I_{\text{sp}} g_0} \left[S_\epsilon - \epsilon + \epsilon u + \mu \max(0,\, \epsilon_{\text{DC}} - S_{\text{DC}}) \right] \tag{15}$$

where S_ϵ is the switching function for the EO problem, and in particular $S_\epsilon = S_f$. In Eq. (15), the PMP has already been applied to retrieve the optimal thrust direction $\boldsymbol{\alpha}^*$, which is the same as Eq. (11). To derive the necessary conditions of optimality, it has to be

$$\frac{\delta H_\epsilon}{\delta u} = 0 \implies u^* = \frac{\epsilon - S_\epsilon - \mu \max(0,\, \epsilon_{\text{DC}} - S_{\text{DC}})}{2\epsilon} \tag{16}$$

To obtain the optimal control law for $u*$, consider the following cases:

1) $S_\epsilon > \epsilon$ and $S_{\text{DC}} \geq \epsilon_{\text{DC}}$: in this case $u^* = 0$, with no constraints on μ;
2) $S_\epsilon > \epsilon$ and $S_{\text{DC}} < \epsilon_{\text{DC}}$: in this case $u^* = 0$, with the Lagrangian multiplier being $\mu > 0$;
3) $S_\epsilon < -\epsilon$ and $S_{\text{DC}} \geq \epsilon_{\text{DC}}$: in this case $u^* = 1$, with no constraints on μ;
4) $S_\epsilon < -\epsilon$ and $S_{\text{DC}} < \epsilon_{\text{DC}}$: in this case $u^* = 0$, with the multiplier being $\mu > \frac{\epsilon - S_\epsilon}{\epsilon_{\text{DC}} - S_{\text{DC}}} > 0$;
5) $|S_\epsilon| < \epsilon$ and $S_{\text{DC}} \geq \epsilon_{\text{DC}}$: in this case $u^* = \frac{\epsilon - S_\epsilon}{2\epsilon}$, with no constraints on μ;
6) $|S_\epsilon| < \epsilon$ and $S_{\text{DC}} < \epsilon_{\text{DC}}$: in this case $u^* = 0$, with the multiplier being $\mu > \frac{\epsilon - S_\epsilon}{\epsilon_{\text{DC}} - S_{\text{DC}}} > 0$.

Ultimately, these cases lead to the optimal control law

$$u^* = \begin{cases} 0 & \text{if} \quad S_\epsilon > \epsilon \quad \text{or} \quad S_{DC} < \epsilon_{DC} \\ \frac{\epsilon - S_\epsilon}{2\epsilon} & \text{if} \quad |S_\epsilon| \leq \epsilon \quad \text{and} \quad S_{DC} \geq \epsilon_{DC} \\ 1 & \text{if} \quad S_\epsilon < -\epsilon \quad \text{and} \quad S_{DC} \geq \epsilon_{DC} \end{cases} \tag{17}$$

As $\boldsymbol{\alpha}^*$ and u^* depend on S_ϵ, and thus on both the state and costate $\mathbf{y} = [\mathbf{x},\, \boldsymbol{\lambda}]$, to find the optimal control they have to be integrated for the whole trajectory. The augmented equations of the dynamics are then considered

$$\dot{\mathbf{y}} = \mathbf{F}(\mathbf{y}) \Rightarrow \begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = \mathbf{g}(\mathbf{r}) + u^* \frac{T_{max}}{m} \boldsymbol{\alpha} \\ \dot{m} = -u^* \frac{T_{max}}{I_{sp} g_0} \\ \dot{\boldsymbol{\lambda}}_r = -\frac{3\mu}{r^5} (\mathbf{r} \cdot \boldsymbol{\lambda}_v) \mathbf{r} + \frac{\mu}{r^3} \boldsymbol{\lambda}_v \\ \dot{\boldsymbol{\lambda}}_v = -\boldsymbol{\lambda}_r \\ \dot{\lambda}_m = -\frac{u^* \lambda_v T_{max}}{m^2} \end{cases} \tag{18}$$

where $\boldsymbol{\alpha}^*$ has been substituted according to Eq. (11). The integration of Eqs. (18) has to satisfy the boundary conditions in Eqs. (6)– (7) at t_i and t_f. In addition to these, since no final condition is enforced on the mass value, the condition $\lambda_m\left(t_f\right) = 0$ has to be enforced as well [10]. All of this describes a two-point boundary value problem where the only unknown variable is the value of the costates at the initial time, $\boldsymbol{\lambda}_i = \boldsymbol{\lambda}\left(t_i\right)$.

Remark 1: Let $\mathbf{y}(t) = \boldsymbol{\varphi}_\epsilon\left(\mathbf{y}_i = [\mathbf{x}_i,\, \boldsymbol{\lambda}_i],\, t_i,\, t\right)$ be the solution flow for a specific ϵ value of Eqs. (18) integrated from the initial time t_i to a generic time instant t, with $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*$ and $u = u^*$ provided by Eq. (11) and Eq. (17) and $\mathbf{x}_i$ provided by Eqs. (6). The EO and FO indirect shooting problems read as

$$\text{Find } \boldsymbol{\lambda}_i^* \text{ such that } \mathbf{y}\left(t_f\right) = \boldsymbol{\varphi}_\epsilon\left(\left[\mathbf{x}_i,\, \boldsymbol{\lambda}_i^*\right], t_i, t_f\right) \text{ satifies } \begin{bmatrix} \mathbf{r}\left(t_f\right) - \mathbf{r}_T\left(t_f\right) \\ \mathbf{v}\left(t_f\right) - \mathbf{v}_T\left(t_f\right) \\ \lambda_m\left(t_f\right) \end{bmatrix} = \mathbf{0} \tag{19}$$

Synthetically, this problem translates into a zero-finding one, where the function to null is the flow $\mathbf{y}\left(t_f\right)$ with respect to the final boundary conditions and the variables to guess are the initial costates $\boldsymbol{\lambda}_i$. As zero-finding problems are known to benefit from the introduction of the analytical derivatives of the function to null, in this work they have been provided to the solver. They have been computed and can be found in the Appendix V.A.

III. Solution Method

The indirect shooting problem to be solved requires the solution of a zero-finding problem where the function to null is the flow $\mathbf{y}\left(t_f\right) = \left[\mathbf{x}\left(t_f\right),\, \boldsymbol{\lambda}\left(t_f\right)\right]$ with respect to the final boundary conditions in Eq. (19), and the variables to guess are the initial costates $\boldsymbol{\lambda}_i = \boldsymbol{\lambda}\left(t_i\right)$. To do so, the integration of the flow itself has to be computed at each iteration of the zero-finding problem. The introduction of the duty cycles discontinuity makes these computations difficult and for this reason, several precautions have been implemented to increase the accuracy and ease the convergence.

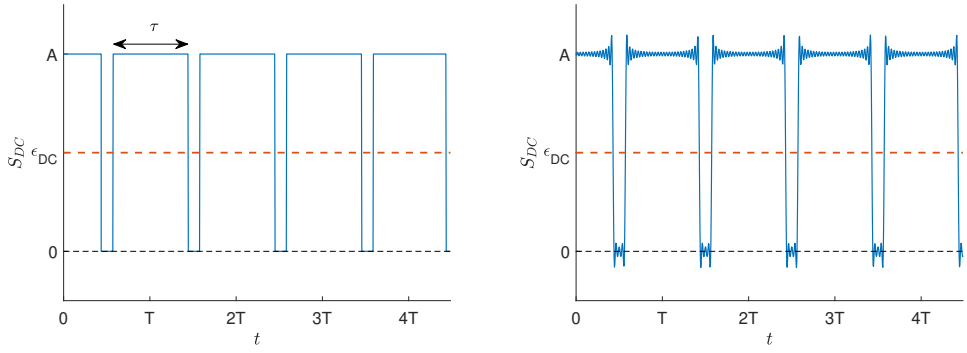
A. Duty Cycle Switching Function Model

To model $S_{DC}(t)$ a pulse wave function is selected. In particular, due to its greater regularity and smoothness with respect to the normal pulse wave function, an expansion based on the sinc function, i.e., $\mathrm{sinc}(t) = \sin(\pi t)/\pi t$, is employed as

$$S_{DC}(t) = \frac{A\tau}{T}\left(1 + 2\sum_{n=1}^{\infty}\left(\mathrm{sinc}\left(n\frac{\tau}{T}\right)\cos\left(2\pi n f t\right)\right)\right) \tag{20}$$

where A is the amplitude, T is the period, τ is the pulse length, and $f = 1/T$ is the frequency of the pulse wave. These values are tailored to enforce a pre-selected duration of the coasting arcs, e.g., to have duty cycles corresponding to a working week with 6 days of thrusting and 1 day of coasting, T is imposed to be 7 days, and τ to be 6 days (see Fig. 1(a) for reference). The expansion of the wave is truncated at a value n significantly high to have smoothness in the function

and thus reduce the influence of the noise due to the Gibbs phenomenon in the sinc function expansion (see Fig. 1(b)). For the same last reason, ϵ_{DC} is set to be half the value of the amplitude A. In this way, the phenomenon does not create inaccuracies during the integration. Note that $(\epsilon_{\mathrm{DC}} - S_{\mathrm{DC}})$ is greater than 0, as requested for Section II. In all the



(a) *Pulse wave modeling the duty cycles alternation.* (b) *Pulse wave modeling with sinc expansion.*

Fig. 1 Duty cycle switching function modeled as a pulse wave.

simulations in Section IV n has been imposed to 30, $A = 0.2$, and thus $\epsilon_{DC} = 0.1$. The values of T and τ have been varied to model different durations of the duty cycles for each simulation. The expansion selected naturally starts with half of a thrusting arc at the beginning of the trajectory. This configuration can be modified by shifting the time variable t of the desired amount in Eq. (20). Note that the formalism presented in this work applies to whatever is the shape of the function $S_{DC}(t)$, as long as it remains dependent only on time. Thus, different shapes of OC trajectories, even not periodic, can be imposed.

B. Switching Detection Technique

During the integration of Eq. (31) the detection of the switching time instants t_s is essential, as discontinuities cause accumulation of the integration error around those instants if they are not properly computed. Moreover, since our method relies on the accuracy of the gradient provided, it is necessary to introduce the compensation matrices $\mathbf{\Psi}_\epsilon\,(t_s)$ through Eq. (33) for every switching of S_ϵ. However, no prior information on the structure of the optimal thrusting profile is known, apart from the location of the duty cycles. Thus, a precise technique for the detection of the switches in the switching functions is introduced in this Subsection. Let t_k and t_{k+1} be two consecutive integration time instants, and $\mathbf{y}_k$ and $\mathbf{y}_{k+1}$ the correspondent states and costates. If, e.g., $(S_\epsilon\,(\mathbf{y}_k) - \epsilon)\,(S_\epsilon\,(\mathbf{y}_{k+1}) - \epsilon) < 0$, there exists $t_s \in [t_k,\, t_{k+1}]$ such that $S_\epsilon\,(t_s) = \epsilon$. Then, the switching detection technique based on a bisection method with accuracy 10^{-16} and a maximum of 50 iterations is implemented to detect t_s. As at the generic t_i the switching function $S_\epsilon\,(\mathbf{y}_i)$ depends on the state and costate at that time instant, integration of the trajectory up to that is necessary. The integration is performed with a fixed-step 7(8) Runge–Kutta integration scheme [33]. Note that the crossings of S_ϵ are detected not only when

$\epsilon = 0$, but for each ϵ value, to ease the location of the discontinuities when $\epsilon = 0$ is reached in the FO problem.

As regards the S_{DC} function, even if no compensation matrix is necessary for the crossing of ϵ_{DC}, the identification of the time instants when this happens is necessary for the accuracy of the integration. However, being S_{DC} function only of time, and being the final time t_f fixed in the considered problem, its switching time instants t_s can be identified regardless of the integration. For this reason, in this work, they are pre-computed as zeros of the function $(S_{DC} - \epsilon_{DC})$, and then enforced during the integration of Eq. (31), avoiding useless computation during the solution of the shooting problem. When a cross of ϵ_{DC} in S_{DC} is noted at t_k, the actual time instant t_k is compared to the saved vector of roots of the function $(S_{DC} - \epsilon_{DC})$, in order to select the closest switching time $t_{s,i}$ and locate it. However, when the integration step is too small, i.e., t_{k+1} and t_k are too close, it can happen that the located $t_{s,i}$ is not the correct one, but the one immediately after or before. This happens as $t_{s,i}$ are also pre-computed within a certain accuracy. This wrong location does not happen often, but when it does, it can fail the optimization itself, as the shooting integration is not performed correctly. Thus, a check on the time step, i.e., $t_{k+1} - t_k <$ 10E-14, is performed. When this condition is true, a bisection method with accuracy 10^{-16} and a maximum of 50 iterations is exploited to impose the correct value of t_s.

The logic of the switching technique is reported as Algorithm 1.

Algorithm 1 Switching detection technique.

Require: t_{k+1}, t_k, $\mathbf{y}_k$, $\mathbf{y}_{k+1}$
  if $S = S_{DC}$ **then**
    if $t_{k+1} - t_k < 10^{-14}$ **then**
      $t_a = t_k$, $S_a = S_{DC}(t_a)$
      $t_b = t_{k+1}$
      $t_c = (t_a + t_b)/2$, $S_c = S_{DC}(t_c)$
      while $|S_c - \epsilon_{DC}| > 10^{-16}$ and iterations ≤ 50 **do**
      bisection method on $[t_a, t_b]$
      end while
    else
      allocate the pre-computed $t_c \in [t_a, t_b]$
    end if
  else if $S = S_\epsilon$ **then**
    $t_a = t_k$, $\mathbf{y}_a = \mathbf{y}_k$, $S_a = S_\epsilon(\mathbf{y}_a, t_a)$
    $t_b = t_{k+1}$
    $t_c = (t_a + t_b)/2$, integrate up to $\mathbf{y}_c$, $S_c = S_\epsilon(\mathbf{y}_c, t_c)$
    while $|S_c - \epsilon| > 10^{-16}$ and iterations ≤ 50 **do**
      bisection method on $[t_a, t_b]$, integrating with 7(8) Runge-Kutta to get the flow $\mathbf{y}_i$ when necessary
    end while
  end if

C. Triple Continuation Scheme

The discontinuities introduced by both S_ϵ and S_{DC} prevent the solver from reaching convergence easily. This is particularly true as the number of duty cycles to impose increases and their duration decreases, i.e., as the forced coasting arcs become more and more frequent. For this reason, a triple continuation scheme inspired by the eclipses

imposition in [24] is implemented here. Firstly, the continuation on ϵ to solve the FO problem starting from the EO one, as described in Section II, is employed. This first ϵ-continuation starts solving the indirect shooting problem with objective function as in Eq. (14) with $\epsilon = 1$, corresponding to the EO problem. The solution obtained is exploited as an initial guess to solve the same problem but with a slightly lower value in ϵ. The process is iterated until the problem with $\epsilon = 0$ is solved, i.e., until the FO problem is solved.

The obtained solution is the FO trajectory without any duty cycle imposed. At this point, a second continuation scheme, the N_{max}-continuation, is employed to gradually enforce the coasting arcs required to obtain an OC trajectory. This approach, depicted in Fig. 2, gradually turns inactive coasting arcs into active ones. Essentially, the scheme works by increasing the maximum number of active forced coasting arcs N_{max} at each iteration starting from the FO solution obtained by the ϵ-continuation, which has $N_{max} = 0$.

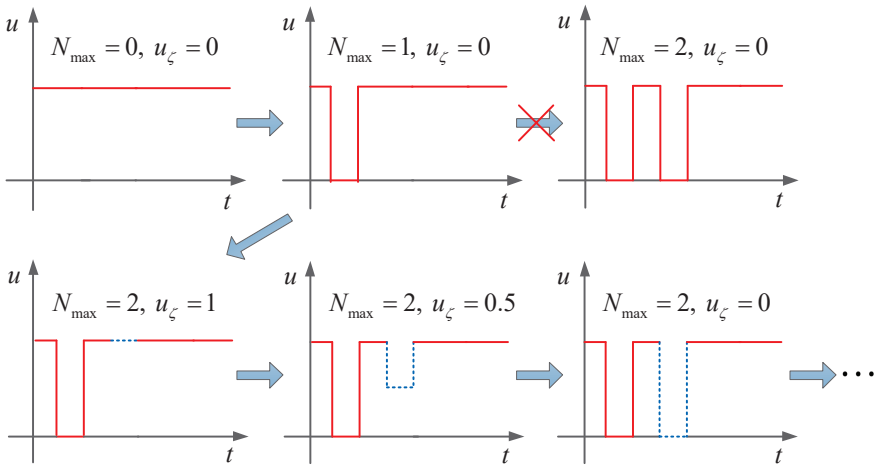


Fig. 2 N_{max}- and u_ζ-continuation schemes.

Let $N_s(t)$ be the number of the accumulated operational compliant coasting arcs at a certain time instant t, i.e., the coasting arcs due to duty cycles present in $[t_i, t]$. The value of $N_s(t)$ starts from a value of 0.5 if t_i is inside a time interval where $S_{DC} < \epsilon_{DC}$, and from a value of 0 if t_i is inside a time interval where $S_{DC} \geq \epsilon_{DC}$, and increases by 0.5 every time S_{DC} crosses ϵ_{DC}. A coasting arc is defined as active if it starts at t_a such that $N_s\,(t_a) \leq N_{max}$. Inactive coasting arcs are foreseen by the duty cycle switching function but are not imposed into the thrust profile as they happen at t_i such that $N_s\,(t_i) > N_{max}$. Thus, they affect the value of $N_s(t)$ but do not change the status of the engine. At each iteration, starting from the FO solution which has $N_{max} = 0$, N_{max} is increased by 1, and the obtained solution is used as an initial guess for the subsequent optimization. The N_{max}-continuation is stopped when the trajectory becomes OC, i.e., when N_{max} reaches the OC number of coasting arcs, N_{DC}. This value is computed as $N_{DC} = \lceil ToF/T \rceil$, where ToF is the time of flight and T is the period of the S_{DC} considered. In this way, more and more coasting arcs are turned to active at each iteration until the whole trajectory is OC.

However, it can happen that for a certain value of N_{max}, the N_{max}-continuation fails. For this reason, a third

continuation scheme, the u_ζ-continuation is introduced. The variable u_ζ is defined as the throttle factor in the N_{max}-th active coasting arc to impose, and it is the third parameter on which a continuation is performed. Following Fig. 2, starting from the FO solution (i.e., with $N_{max} = 0$ and $u_\zeta = 0$), the number of active coasting arcs N_{max} is increased from 0 to 1. The FO solution is used as the initial guess for this auxiliary problem (i.e., $N_{max} = 1$ and $u_\zeta = 0$). Supposing the solution to this problem is obtained, N_{max} is increased from 1 to 2, and the new auxiliary problem (i.e., $N_{max} = 2$ and $u_\zeta = 0$) is solved having as initial guess the trajectory solving the previous problem. Suppose that in this case, the solver fails (see Fig. 2). Then, the parameter u_ζ is imposed as 1, and a new auxiliary problem (i.e., $N_{max} = 2$ and $u_\zeta = 1$) is solved, having the same old initial guess. Virtually, the two auxiliary problems, the one with $N_{max} = 1$ and $u_\zeta = 0$ and the one with $N_{max} = 2$ and $u_\zeta = 1$, are the same, as even if the number of active coasting arcs is increased by 1 in the last problem, being $u_\zeta = 1$, no coasting arc is actually introduced. The value of u_ζ is then gradually reduced homotopically, keeping the same value of N_{max}, until $u_\zeta = 0$, and the solution for $N_{max} = 2$ and $u_\zeta = 0$ is obtained. Once this trajectory is computed, the N_{max}-continuation can proceed, the solution is used as an initial guess for the new problem with $N_{max} = 3$ and $u_\zeta = 0$, and the process can go on.

Note that the proposed methodology firstly exploits the ϵ-continuation and then N_{max}-continuation, and not the other way round, i.e., first the N_{max}-continuation and then the ϵ-continuation. This approach has been chosen so that a very ill-conditioned STM preventing the convergence, more likely to happen when several forced coasting arcs are present, would be encountered only at the final steps of the process.

Since the third continuation is on the gradual reduction of u_ζ, the minimum value of the throttle factor u_{min} in the N_{max}-th forced coasting arc is not always 0 as instead is in the EO and FO problems. Accordingly to the PMP [24], the optimal throttle factor equation turns from Eq. (17) into

$$u^* = \begin{cases} u_{min} & \text{if} \quad S_\epsilon > (1 - 2u_{min})\,\epsilon \qquad\quad \text{or} \quad S_{DC} < \epsilon_{DC} \\ \frac{\epsilon - S_\epsilon}{2\epsilon} & \text{if} \quad -\epsilon \le S_\epsilon \le (1 - 2u_{min})\,\epsilon \quad \text{and} \quad S_{DC} \ge \epsilon_{DC} \\ 1 & \text{if} \quad S_\epsilon < -\epsilon \qquad\qquad\quad \text{and} \quad S_{DC} \ge \epsilon_{DC} \end{cases} \tag{21}$$

where u_{min} is $u_\zeta = 1 \rightarrow 0$ in the N_{max}-the forced coasting arc to be imposed and 0 otherwise, that is

$$u_{min} = \begin{cases} u_\zeta & \text{if} \quad N_s > N_{max} - 1 \;\text{ and }\; N_s < N_{max} \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

It has to be noted that several continuation methods can be adopted to impose the coasting arcs. In particular, the hyperbolic tangent smoothing (HTS) method presented in [34] has been tested prior to the N_{max}-continuation. The basic idea is to make the switching function S_{DC} more and more sharp, continuing a parameter ρ in the hyperbolic tangent function approximating S_{DC}. However, when the coasting arcs become progressively more in number and

shorter in duration, the HTS continuation has been proven to fail in the convergence of the shooting problem. For this reason, the N_{max}- and u_ζ-continuation schemes have been adopted.

D. Integration Flowchart

To solve the indirect shooting method proposed at the end of Section II in *Remark 1*, the integration of Eqs. (18) has to be performed several times. The solver attempts to guess the correct value of $\boldsymbol{\lambda}_i^*$ to match the final boundary conditions on the flow, and thus to check if convergence is reached the trajectory has to be integrated up to t_f at each Newton's iteration. The integration flowchart employed, an augmented version of the one in [35] (depicted in blue in Fig. 3) is then exploited for accurate integration of states, costates, and the STM.

To ease the discussion and schematize the flow, firstly the logic of the engine status and the one related to the presence of the coasting arcs are defined. Let u_{type} be the engine status at a certain time instant t. It can assume three logical values according to the value of the throttle factor u, as

$$u_{type} = \begin{cases} \text{On} & \text{if} \quad u = 1 \\ \text{Medium} & \text{if} \quad u \in [u_{min},\, 1] \\ \text{Off} & \text{if} \quad u = u_{min} \end{cases} \tag{23}$$

where u_{min} is defined by Eq. (22). Let $\tilde{a}_{type}$ be the signal labeling a certain time instant t as in a coasting arc (Coast) or in a thrusting arc (Thrust), according to the switching function S_{DC}, as

$$\tilde{a}_{type} = \begin{cases} \text{Coast} & \text{if} \quad S_{DC} < \epsilon_{DC} \\ \text{Thrust} & \text{otherwise} \end{cases} \tag{24}$$

The label $\tilde{a}_{type}$ just gives information on the function S_{DC}, but not on the type of coasting arc, if it is active or not in the N_{max}-continuation. Thus, to consider the effects of the active coasting arcs, we define another parameter, a_{type}, labeling a certain time instant t as in an active coasting arc (Coast) or in a thrusting arc or inactive coasting arc (Thrust), according to the value of both the switching function S_{DC} and of N_s.

$$a_{type} = \begin{cases} \text{Coast} & \text{if} \quad S_{DC} < \epsilon_{DC} \text{ and } N_s \leq N_{max} \\ \text{Thrust} & \text{otherwise} \end{cases} \tag{25}$$

Obviously, a_{type} is the signal affecting the status of the engine u_{type}, and not $\tilde{a}_{type}$. As stated before, at the beginning of the integration, if the initial point is inside a coasting arc $N_s\,(t_i)$ is set equal to 0.5, while if it is outside, it is set to 0. Every time a switch of $\tilde{a}_{type}$ is detected, $N_s(t)$ is then increased by 0.5.

In Fig. 3 the flowchart for the integration is represented. The blocks in blue are the ones originally from [35]. This flowchart is inspired by the one from [23]. This flow is performed at every k-th time instant for the integration up to the

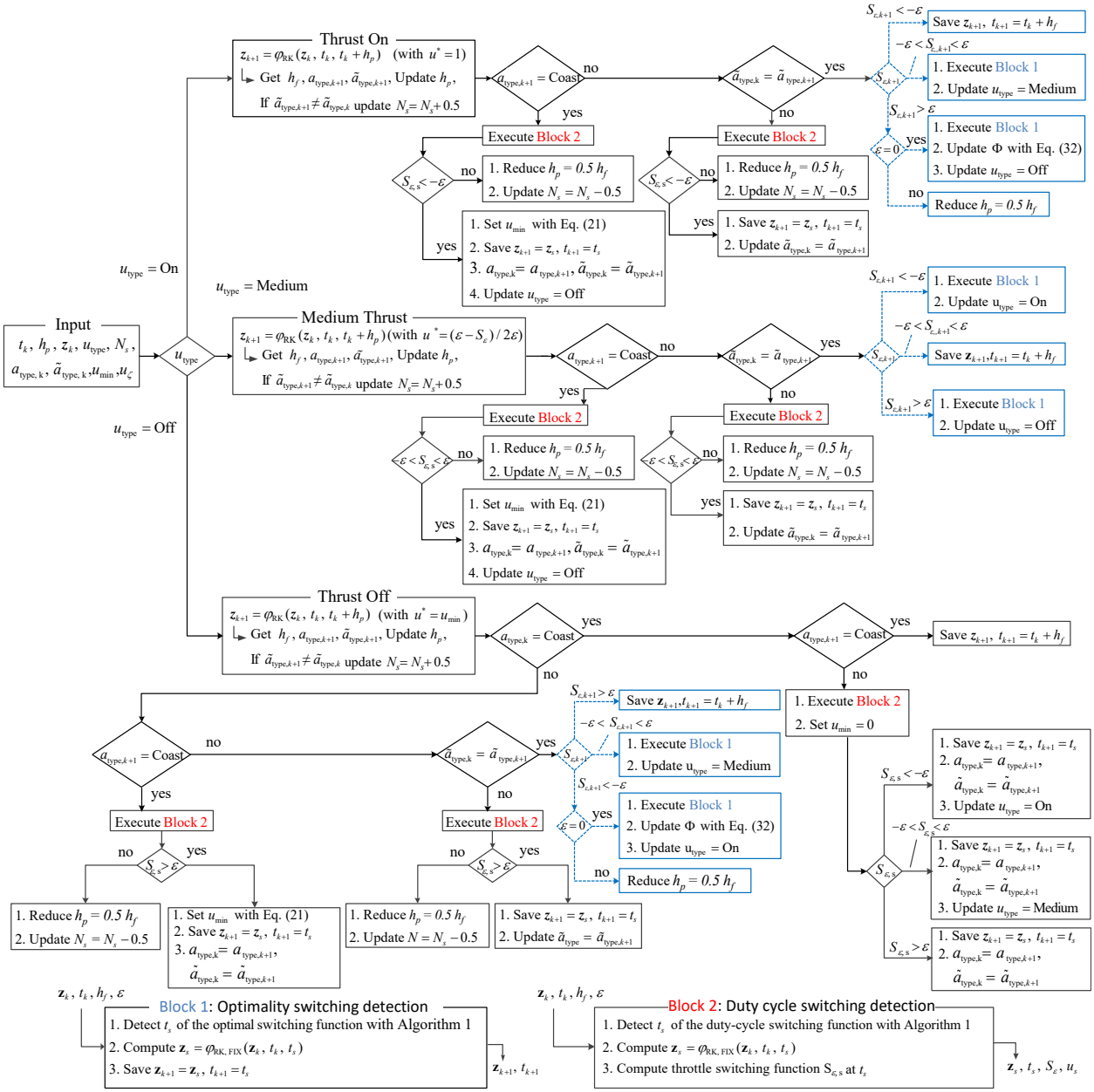


Fig. 3 Complete integration flowchart.

next $(k+1)$-th time instant takes. The integration takes as inputs:

1) t_k the time instant of reference,

2) h_p the guess for the time step predicted from the previous integration,

3) $\mathbf{z}_k$ the augmented state at t_k,

4) u_{type} the engine status at t_k,

5) N_s the number of accumulated coasting arcs up to t_k,

6) $a_{type,k}$ the label of t_k with respect to the active coasting arcs,

7) $\tilde{a}_{type,k}$, the label of t_k with respect to the accumulated coasting arcs,

8) u_{min} the minimum level of thrust throttle up to t_k,

9) u_ζ the thrust throttle in the N_{max}-th coasting arc in the u_ζ-continuation.

The values of ϵ and N_{max} are pre-set at the beginning of each computation depending on the step of the ϵ- and N_{max}-continuation schemes that are being performed. Depending on the value of u_{type}, three branches depart from these inputs. All of them start with an integration of the augmented state $\mathbf{z}_k$ with the different values of the throttle factor according to Eq. (23). The integration up to $\mathbf{z}_{k+1}$ is performed with a variable-step 7(8) Runge–Kutta integration scheme with a relative integration accuracy set at 10^{-11}. From this integration, a prediction on the time step h_p to be used at the next iteration is computed according to the variable-step 7(8) Runge–Kutta method, and updated: when the integration succeeds, the step is enlarged accordingly to the accuracy of the computed result. In particular, a modified version of the Runge–Kutta integrator has been implemented, such that h_p does not become too large, and at least three integration nodes are present in each coasting arc. This constraint is imposed to catch all the switches of S_{DC} when the duration of the coasting arcs is particularly short. Note that the integrated state $\mathbf{z}_{k+1}$ is computed at $t_k + h_f$, where h_f is adjusted internally by the variable-step Runge–Kutta scheme, and $h_f \neq h_p$.

Once the integration is performed, the values of $\tilde{a}_{type,k+1}$ and $a_{type,k+1}$ are updated as well, according to Eqs. (24) and (25). If $\tilde{a}_{type,k+1} \neq \tilde{a}_{type,k}$, the number of accumulated coasting arcs is updated as $N_s = N_s + 0.5$, as a switch of the duty cycle switching function S_{DC} has been detected. From this point, different branches of the decision tree are followed according to the value of u_{type}. The branches for the throttle factor On and Medium follow the same logic, so just the first one is described here. A zoom of this branch from Fig. 3 is reported to ease the comprehension in Fig. 4.

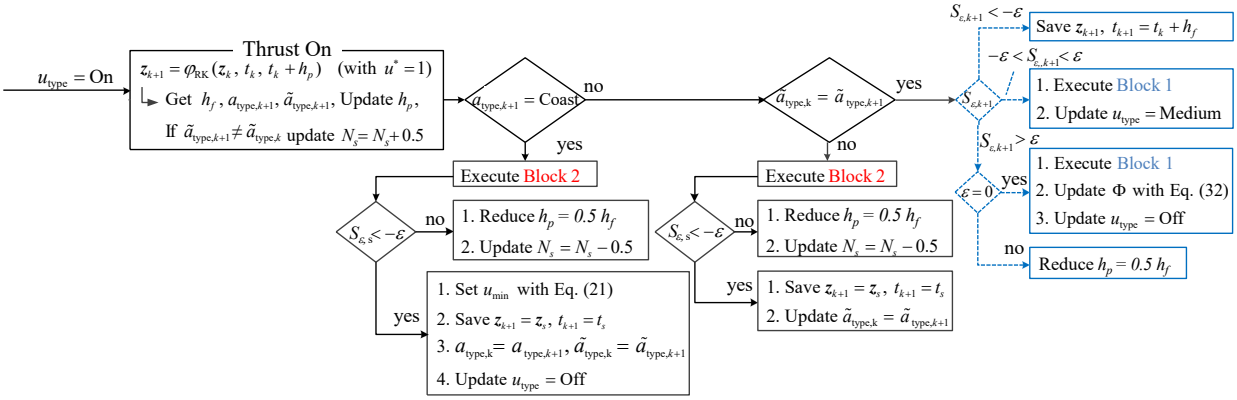


Fig. 4 Focus on the u_{type} = On branch of the integration flowchart.

Firstly, the value of $a_{type,k+1}$ is checked: if it is Coast, it means that S_{DC} is below ϵ_{DC}. To be on the branch u_{type} = On, a_{type} had to be previously set on Thrust, thus, a switch in the S_{DC} has happened. This switch has to be detected for proper integration. This happens in Block 2 in Fig. 3, where Algorithm 1 is employed to detect the switching time t_s and a fixed-step 7(8) Runge–Kutta integration scheme is exploited to compute the state $\mathbf{z}_s$ at t_s. The optimal

switching function $S_{\epsilon,s}$ at t_s is computed, and used to evaluate if also S_ϵ had a switch. In the case of $S_{\epsilon,s} \geq -\epsilon$, the optimal switching function had a jump from On to Medium or Off. To properly detect this switch, the guess integration interval is reduced such that $h_p = 0.5h_f$, N_s is rolled back to its previous value, and the integration is performed again from t_k with the reduced time interval. In the case of $S_{\epsilon,s} < -\epsilon$, no switch in the optimal switching function happened. Thus, u_{min} is set through Eq. (22), the new augmented state and t_s are saved, $\tilde{a}_{type,k}$ and $a_{type,k}$ are updated, and the u_{type} is set to Off. At the next integration step, the branch for u_{type} Off will be followed.

In the case $a_{type,k+1}$ is still on Thrust, the check is now shifted on $\tilde{a}_{type,k+1}$. If it is not equal to $\tilde{a}_{type,k}$, a switch in S_{DC} at t_s has to be detected as before. Then, similarly, a check on $S_{\epsilon,s}$ is made to ensure that also a switch in S_ϵ is not present. If the switch is present the integration is rolled back and performed again, if it is not the case, the integration continues with an updated value of $\tilde{a}_{type,k}$. In case $\tilde{a}_{type,k+1} = \tilde{a}_{type,k}$ no switches on S_{DC} are present, i.e., no changes in neither a_{type} and $\tilde{a}_{type}$ are present. The focus moves then to the detection of eventual switches in just S_ϵ. Thus, the blue branch in Fig. 5 is executed. Basically, the value $S_{\epsilon,k+1} = S_\epsilon(t_{k+1})$ is checked to understand if a switch in u_{type} has to be taken into account. Eventually, the switching time t_s is computed with Algorithm 1 in Block 1, and eventual updates in the STM are computed through Eq. (33). The branch for u_{type} Medium follows the same logic, but the switches of S_ϵ that are tested on the optimal switching function remaining in $-\epsilon < S_{\epsilon,k+1} < \epsilon$, and those eventual jumps have not to be compensated in the STM.

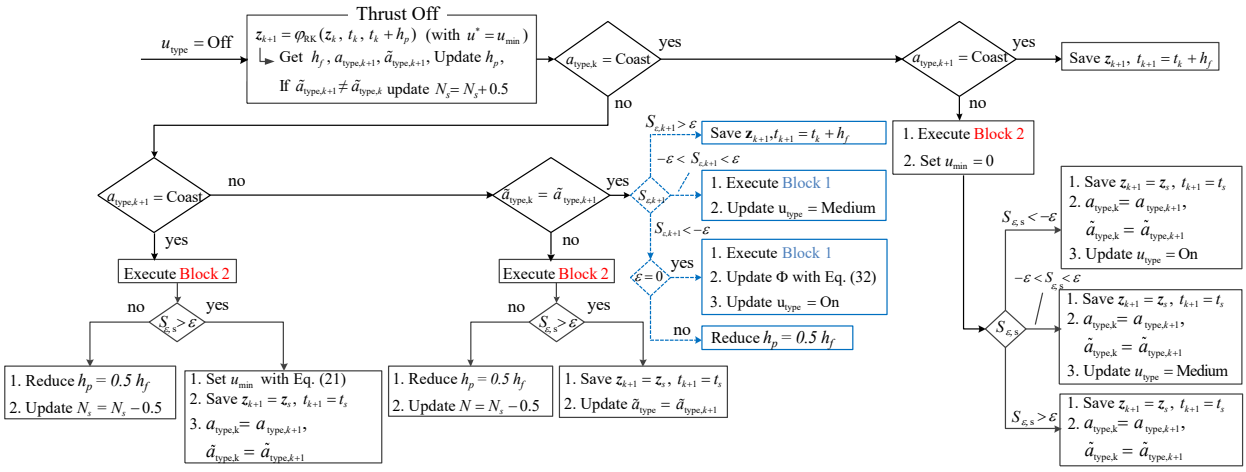


Fig. 5 Focus on the u_{type} = Off branch of the integration flowchart.

In Fig. 5 a zoom of the branch for u_{type} = Off from Fig. 3 is reported. When the branch to follow is u_{type} = Off, the integration flowchart has to take into account more possible paths, as the switching off of the engine could be caused both by the values of S_ϵ and S_{DC} at t_k. Firstly, the integrator checks $a_{type,k}$ to understand if the engine is switched off at t_k due to S_{DC} or S_ϵ. If it switched off due to S_{DC}, i.e., $a_{type,k}$ = Coast, the value of $a_{type,k+1}$ is checked, to understand if a switch in the duty cycle switching function S_{DC} has been caught during the integration up to t_{k+1}. If $a_{type,k}$ is still Coast, no further actions are needed, the augmented state is saved and the integration continues. If

$a_{type,k+1}$ is Thrust, Block 2 is performed to detect the switching time t_s and u_{min} is set to 0. Then, a check on the value of $S_{\epsilon,s}$ is made to state if a switch in S_ϵ is present as well. The new values of the augmented state, of $\tilde{a}_{type,k}$ and of $a_{type,k}$ are updated, as well as the label of u_{type} if necessary. No updates in the STM are needed in any case, as the switch to Thrust has been caused by S_{DC}, and from t_{k+1} the integration proceeds just considering the optimality of S_ϵ.

In case the engine is switched off due to S_ϵ, i.e., $a_{type,k} =$ Thrust, again the value of $a_{type,k+1}$ is checked. If it is Coast, the switching time t_s of S_{DC} has to be detected through Block 2, and then, according to the value of $S_{\epsilon,s}$ the integration is rolled back reducing the time step and re-performed, or saved and continues with u_{min} set through Eq. (22). If instead $a_{type,k+1} =$ Thrust, the eventual switch on $\tilde{a}_{type}$ is checked. In case the switch happened, again Block 2 is performed, t_s is computed and $S_{\epsilon,s}$ is used to understand if the optimal switching function switched as well, and, in case, whether the integration has to be re-performed or not. In case $\tilde{a}_{type,k} = \tilde{a}_{type,k+1}$, the original block in blue in Fig. 5 is executed, to check eventual switches in S_ϵ and include eventual correction in the STM with Eq. (33).

As the integration flowchart here explained is quite cumbersome to catch at first approach, the reader is invited to follow the integration by selecting one particular case and to reason on the different decision paths the integration algorithm has to take at each time step t_k. The main logic behind the integration flowchart is that each switching time has to be detected for accurate integration, the cause of each switching has to be detected, and no more than one switching can be detected at each time step.

IV. Results

In this section, the indirect shooting method of *Remark 1*, is solved with the addition of the OC problem presented in Section II and through the methods presented in Section III. It is has been applied to different mission scenarios to test its robustness and consistency, in particular to M-ARGO and RAMSES missions. Moreover, it has been tested against a direct collocation method previously developed by the authors. The results are presented hereafter.

A. M-ARGO CubeSat Scenario

The Miniaturised Asteroid Remote Geophysical Observer (M-ARGO) [28] will be the first CubeSat designed by the European Space Agency to perform its interplanetary travel towards a near-Earth asteroid (NEA) alone. The designed CubeSat is a 12U model, with three main objectives: to perform the first fully autonomous interplanetary CubeSat cruise between Earth and an asteroid, to characterize the physical properties of the asteroid, and to assess future resource exploitation potential. Being fully autonomous, this CubeSat will be required to follow an OC trajectory, for its stringent pointing, propellant, and power budgets. M-ARGO will be launched as a piggyback payload on an Earth escape trajectory with $C_3 \approx 0$, toward the L_2 Sun–Earth Lagrangian point (SEL2). From here, after a parking phase in a SEL2 halo orbit, the solar electric propulsion will be exploited for an interplanetary cruise to the target asteroid. In particular, five asteroids have been downlisted by ESA as potential targets. Here, only the nominal trajectory towards asteroid 2010

UE51 has been considered. The M-ARGO CubeSat mission assumptions are summarized in Table 1.

Table 1 M-ARGO mission scenario assumptions.

Parameter	Value
Asteroid target	2010 UE51
Departure date	Sep 17, 2023
ToF	581 days
Thruster	Ion thruster
Thruster input power	[80-130] W
T_{max}	1.3 mN
I_{sp}	3500 s
m_0	28.2 kg

The simulations in this Subsection have been performed with a normalization of the variables with the constants listed in Table 2. The ephemerides of the asteroid 2010 UE51 are retrieved from the Spacecraft Planet Instrument Camera-matrix Events (SPICE) kernels of the HORIZONS system [36]. The departure date of the transfer is set on September 17, 2023, with a time of flight, ToF, of 581 days. The CubeSat is supposed to depart from SEL2 and rendezvous with the asteroid, thus matching both its position and velocity. It has been supposed to impose to this interplanetary transfer several duty cycle schemes of increasing frequency of occurrence, to test the robustness of the method with increasing number of discontinuities.

Table 2 Constants used for the normalization.

Parameter	Value
Sun gravitational constant, μ	1.327124×10^{11} km^3/s^2
Earth gravitational acceleration, g_0	9.80665 m/s^2
Length unit, AU	1.495978×10^{8} km
Time unit, TU	5.022643×10^{6} s
Velocity unit, VU	29.784692 km/s
Mass unit, MU	28.2 kg

The simulations in this Subsection have been performed with an Intel i9-9980HK, a total RAM of 16 GB. Matlab R2023b has been exploited to solve the shooting problem, through a trust-region-dogleg algorithm [37]. The integration code has been compiled into MEX file to speed up simulations. To perform the first ϵ-continuation from the EO to the FO problem, as described in Subsection III.C, it has been imposed a decrease in ϵ of $\Delta\epsilon = 0.05$ for each iteration. For u_ζ instead, a variation $\Delta u_\zeta = 0.1$ is selected. At each iteration, slightly increased values of the variations $\Delta\epsilon$ and Δu_ζ are used if the solver succeeds, otherwise the variations are halved. In particular, $\Delta\epsilon_{new} = 1.01\Delta\epsilon_{old}$ and $\Delta u_{\zeta,new} = 1.01\Delta u_{\zeta,old}$ if the solver succeeded at the previous iteration. In any case, no more than 100 iterations are allowed for both ϵ- and u_ζ-continuations. The continuations is also considered failed if $\Delta u_\zeta < 0.0005$ or $\Delta\epsilon < 10^{-8}$. To

find the initial converging guess for the EO problem with $\epsilon = 1$ the Adjoint Control Transformation (ACT) initialization has been employed [38] to find the first solution of the EO problem until a converging initial guess is obtained. From this guess, the triple continuation scheme is performed to obtain first the FO solution, and then the FO OC solution for four different duty cycle schemes: one with $T = 30$ days and $\tau = 25$ days, M-ARGO.A, one with $T = 15$ days and $\tau = 10$ days, M-ARGO.B, one with $T = 7$ days and $\tau = 6$ days, M-ARGO.C, and one with $T = 5$ days and $\tau = 4$ days, M-ARGO.D. The rationale behind the choice of the schemes used in the simulations is double. To analyze the effects of different T and τ values and different T/τ values, and to align with real-world application cases, to give some useful insights for the next generation of low-thrust missions. It is in particular the case of the combination $T = 7$ days and $\tau = 6$ days, foreseen at the moment in the workflow of the Bepi Colombo mission [2]. The results of the simulations are reported in Table 3.

Table 3 M-ARGO simulations results.

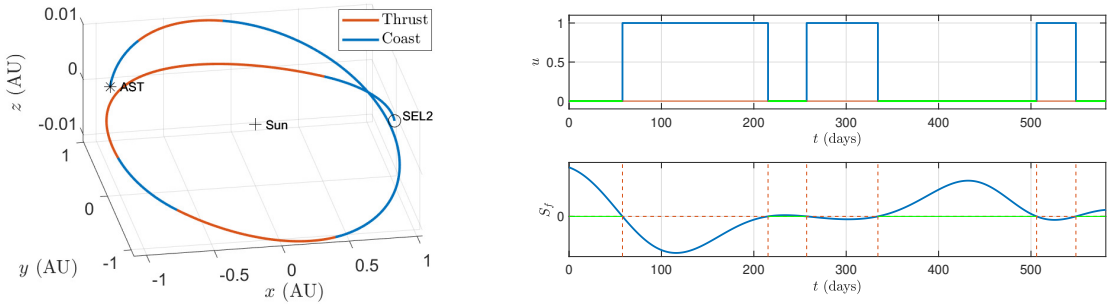
Name	**T** (days)	τ (days)	$(\mathbf{T}-\tau)/\mathbf{T}$	$\mathbf{m_p}$ (kg)	$\mathbf{\Delta m_p}$ (%)	**CPU time** (s)
FO	-	-	-	0.907208	-	28.32
M-ARGO.A	30	25	0.1667	0.913540	0.6980	5.15
M-ARGO.B	15	10	0.3333	0.942919	3.9364	11.59
M-ARGO.C	7	6	0.1429	0.911911	0.5184	72.62
M-ARGO.D	5	4	0.2000	0.916394	1.0126	87.68

The value m_p is the required propellant mass, while the increment in propellant mass Δm_p is computed with respect to the m_p of the FO solution of 0.907208 kg as in Table 3. It can be noted that Δm_p does not increase monotonically with the increase of the frequency of the duty cycles i.e., as T decreases. The higher Δm_p is instead the one of trajectory M-ARGO.B, being 3.9364%, related to the imposition of the scheme with $T = 15$ days and $\tau = 10$ days. The increment in propellant mass can indeed be related to the ratio of forced coasting time over the total time allocated for a duty cycle, $(T - \tau)/T$. This ratio for the scheme with $T = 15$ days and $\tau = 10$ days is the highest, imposing one-third of the time of the duty cycle as coasting time, free of thrust. It can be noted instead, as the scheme $T = 7$ days and $\tau = 6$ days, having the lowest ratio $(T - \tau)/T$, shows the lowest Δm_p as well.

The CPU time required to compute the FO solution in the first row of Table 3 depends only on the ACT initialization: it is not related to the proposed algorithm and is reported just for the sake of completeness. Differently from Δm_p, the CPU time to compute the solution increases with the frequency of the duty cycles. This is in line with the increase of the nodes required to integrate the trajectory: as explained in Subsection III.D, a modified version of the Runge–Kutta integration scheme is exploited such that at least three integration nodes are collocated in each coasting arc. The CPU time is, however, still contained for the worst-case scenario of trajectory D, with $T = 5$ days and $\tau = 4$ days, requiring less than 2 minutes to compute the solution.

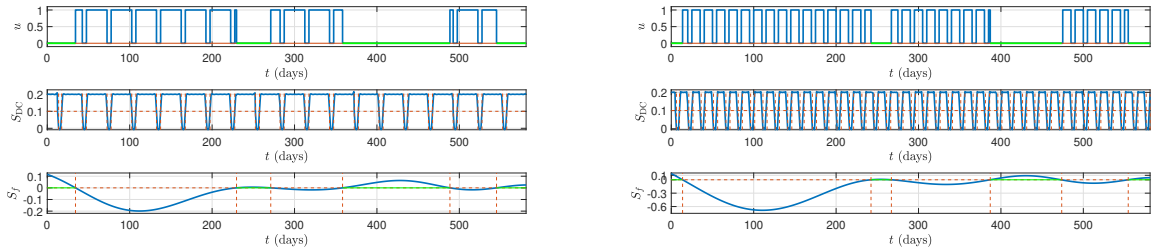
The trajectories in the EclipJ2000 reference frame, as well as the u profiles, the optimal switching function S_f, and

the duty cycle switching function S_{DC}, are reported in Figs. 6 to 8. Note that, as in the rest of the Section, the notation S_f instead of S_ϵ has been preferred as the solution of the trajectories foreseen $\epsilon = 0$, i.e., they are the FO OC cruises. For the trajectories in the EclipJ2000 reference frame. SEL2 stands for the departure point, AST stands for the arrival point at the asteroid, and the Sun's position is reported for reference. For the u and S_f profiles, the green lines highlight the zones where $S_f > 0$, i.e., when the throttle factor is zero not because of the duty cycle scheme imposition, but for the conditions for optimality of the FO problem.



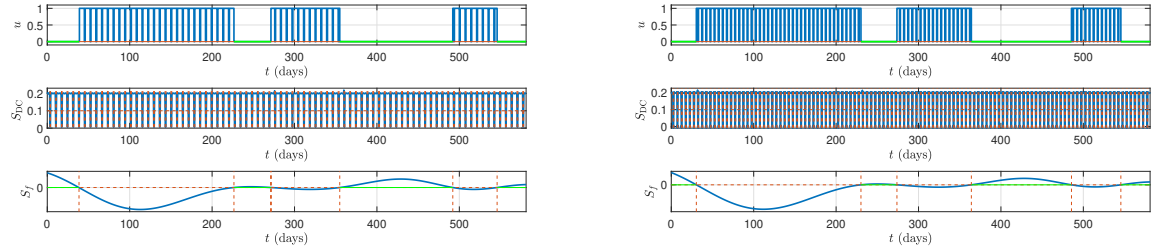
(a) *Trajectory in the EclipJ2000 reference frame.* (b) *Throttle factor and optimal switching function profiles.*

Fig. 6 Solution for M-ARGO FO trajectory.



(a) *M-ARGO.A, T = 30 days and τ = 25 days.* (b) *M-ARGO.B, T = 15 days and τ = 10 days.*

Fig. 7 Solutions for trajectories M-ARGO.A and M-ARGO.B.



(a) *M-ARGO.C, T = 7 days and τ = 6 days.* (b) *M-ARGO.D, T = 5 days and τ = 4 days.*

Fig. 8 Solutions for trajectories M-ARGO.C and M-ARGO.D.

It can be noted that all the OC thrust profiles mimic the shape of the FO solution in Fig. 6(b), i.e., the long thrusting arcs of the FO solution are made OC, suggesting that a sub-optimal OC solution for the FO problem has been reached. In particular, the solver is able to switch off the engine taking into account both the duty cycles and the optimality of the FO trajectory, i.e., maintaining the optimal location of the thrusting blocks visible in Fig. 6(b), these are spread into longer blocks to retrieve the thrust forced into the trajectories. This explains the very low increases in m_p shown in Table 3 for each solution.

B. Direct Method Comparison

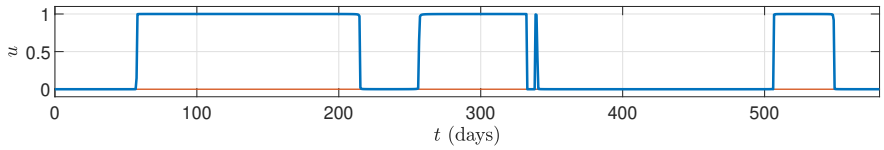
To validate the results obtained in the precedent Subsection, a comparison of the same test case of Table 1 with the same duty cycle schemes has been performed employing the method previously developed by the authors in [27]. In this work, a direct method based on collocation and on a hotomopic approach was developed to compute OC trajectories. The direct method imposes the forced coasting arcs of the duty cycles increasing their weight in the objective function in subsequent iterations. For more information, please refer to [27]. The direct method has been modified to include different schemes of duty cycles, as it was previously thought to compute only OC trajectories with a scheme with $T = 7$ days and $\tau = 6$ days.

Table 4 Direct method simulations results and comparison with the indirect method.

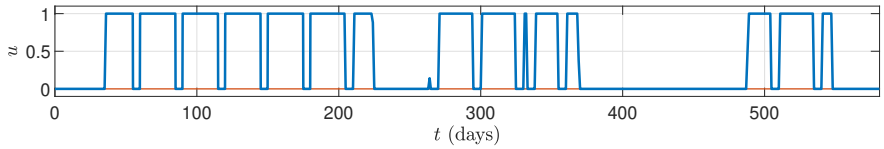
Name	**T** (days)	τ (days)	$\mathbf{m_p}$ (kg)	$\mathbf{\Delta m_{p,D}}$ (%)	$\mathbf{\Delta m_{p,I}}$ (%)	**Iterations**	**CPU time** (s)
FO.1	-	-	0.907220	-	0.0013	-	257.42
M-ARGO.A.1	30	25	0.914285	0.7788	0.0816	3	859.98
M-ARGO.B.1	15	10	0.934268	2.9814	-0.9175	6	1685.54
M-ARGO.B.2	15	10	0.943163	3.9619	0.0259	7	2785.24
M-ARGO.C.1	7	6	0.912322	0.5624	0.0451	2	872.98
M-ARGO.D.1	5	4	0.918302	1.2215	0.2082	3	1510.94

The results of the simulations with the direct approach are in Table 4 and depicted in Fig. 9. They have been reported: the propellant mass m_p required for the trajectory, the increment in propellant mass $\Delta m_{p,D}$ with respect to the FO solution computed with the direct method, i.e., the 0.907220 kg of solution FO.1, the increment in propellant mass $\Delta m_{p,I}$ with respect to the corresponding values computed for each trajectory with the indirect method in Table 3, the number of hotomopic iterations required by the direct method to get to the OC solution, and the CPU times required to compute all the iterations for each duty cycle scheme. Note that, again, the time used to compute the FO.1 solution is reported just for the sake of completeness, but it is representative of the performance of neither the direct nor the indirect method. Regarding the CPU times, the machine used to compute the trajectories with the direct method is the same as in the previous Subsection. It can be noted, however, that they are much higher with respect to the values reported in Table 3. On average, at least 1 order of magnitude more time is required to get to the DC solution with the direct method

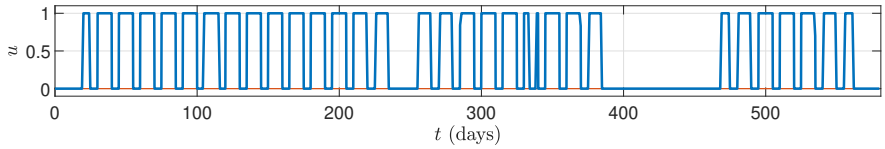
with respect to the indirect method. This is due to the high number of nodes required by the direct method to converge.



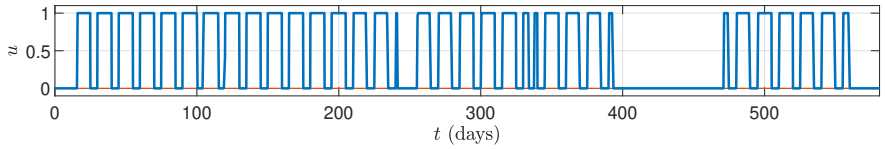
(a) *Thottle factor profile for trajectory FO.1.*



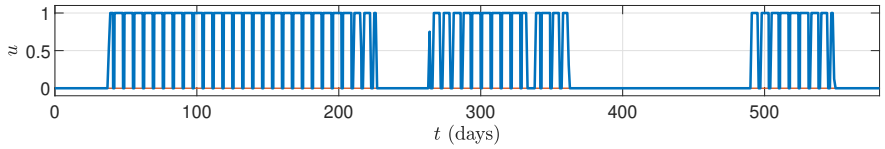
(b) *Thottle factor profile for trajectory M-ARGO.A.1, with* $T = 30$ *days and* $\tau = 25$ *days.*



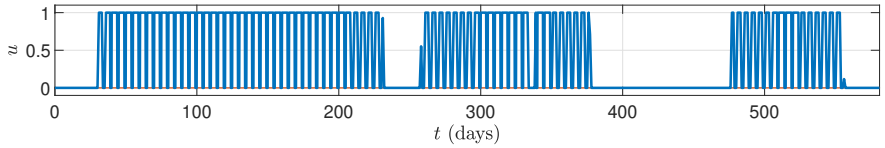
(c) *Thottle factor profile for trajectory M-ARGO.B.1, with* $T = 15$ *days and* $\tau = 10$ *days.*



(d) *Thottle factor profile for trajectory M-ARGO.B.2, the one with doubled time grid, with* $T = 15$ *days and* $\tau = 10$ *days.*



(e) *Thottle factor profile for trajectory M-ARGO.C.1, with* $T = 7$ *days and* $\tau = 6$ *days.*



(f) *Thottle factor profile for trajectory M-ARGO.D.1, with* $T = 5$ *days and* $\tau = 4$ *days.*

Fig. 9 Solutions computed with the direct method.

Again, the trajectories with the higher ratio $(T - \tau)/T$ show a higher increase in propellant mass with respect to the corresponding FO solution, $\Delta m_{p,D}$. This confirms the conclusions drawn in the previous Subsection. It can be also noted that the propellant mass computed by the direct method is higher than the one computed with the indirect method, i.e., $\Delta m_{p,I} > 0$. This is true for each duty cycle scheme, apart from the one with $T = 15$ days and $\tau = 10$ days: in this case, two solutions have been reported in Table 4, M-ARGO.B.1 and M-ARGO.B.2. Trajectory M-ARGO.B.1 was obtained as first solution from the direct method, with $\Delta m_{p,I} = -0.9175$, meaning a decrease in the propellant

mass with respect to the value computed by the indirect shooting method. As this result was not expected, as indirect methods lead to more optimal solutions by nature, another trajectory, with a time grid for the direct collocation with a doubled number of nodes, has been computed, leading to trajectory M-ARGO.B.2, having $\Delta m_{p,I} = 0.0259$. This case underlines the high dependence of the direct method on the number of nodes used to transcribe the problem, while the indirect algorithm previously presented, thanks to the variable-step 7(8) Runge–Kutta integration scheme and the precise integration flowchart of Subsection III.D, autonomously adjusts the number of the nodes in order to obtain the best representation of the solution.

The profiles of the throttle factor of the trajectories computed with the direct method in Table 4 have been reported in Fig. 9. In general, the results obtained with the direct method are very similar to the solutions presented in the previous Subsection IV.A, with expected discrepancies due to the difference in number of nodes and integration techniques. It can be noted that the throttle factor profiles are very similar to the ones of Figs. 6-8. However, some noise in the direct computations, i.e., very short thrust peaks and/or thrust peaks not reaching $u = 1$, is pretty evident in basically every solution of Fig. 9. This is an expected result due to the collocation of the dynamic equations with an a-priori-selected number of nodes, typical of direct methods.

C. RAMSES Mission Reachability analysis

To extensively test the algorithm on a large number of trajectories, it has been applied also to the RAMSES mission scenario, re-computing the porkchop in [29] and studying the effects of the application of different schemes of duty cycle to the cost of the interplanetary phase of this mission. The European Space Agency proposed the mission RAMSES (Rapid Apophis Mission for SEcurity and Safety) to reach the asteroid (99942) Apophis, a potentially hazardous asteroid with a diameter of about 370 meters that will pass within 31,000 km of Earth's surface, closer than geosynchronous satellites, on April 13, 2029 [39]. The close encounter with the Earth will subject Apophis to significant tidal torques, changing both its astrodynamical and physical properties [29]. RAMSES mission is supposed to rendez vous with the asteroid before the close encounter to collect data about these changes while they are happening.

Table 5 RAMSES mission scenario assumptions.

Parameter	Value
Asteroid target	(99942) Apophis
Earliest departure date	Nov 1, 2026
Latest departure date	Jun 13, 2028
Latest arrival date	Feb 13, 2029
ToF_{max}	800 days
T_{max}	60 mN
I_{sp}	1500 s
m_0	500 kg
$m_{p,max}$	73 kg

The mission analysis assumptions of [29] are reported in Table 5. Most of them come from the mission's requirements, such as the maximum propellant mass $m_{p,max}$, the dimension of the porkchop window, and, of course, the target. The spacecraft is supposed to be inserted directly into an interplanetary transfer to Apophis by a launcher. Considering the characteristics of Ariane 6.2 and 6.4 launchers*, the infinity velocity v_∞ at the Earth sphere-of-influence interface has been constrained to be lower than 4 km/s, with free declination δ and right ascension α in the J2000 reference frame. The launch conditions (v_∞, α, and δ) for each trajectory in the porkchop, 11,883 in total, have been computed with a first layer of convex optimization, whose results have then been exploited as informed initial guesses for a second layer of indirect optimization to find the FO and time-optimal (TO) solutions, as explained in depth in [29]. The result is the porkchop shown in Fig. 10, reporting the propellant mass m_p for each trajectory in the departure window considered. On the thick blue line lie the TO solutions, the region dashed in blue is the unfeasible area where no solutions are found because of requiring less ToF with respect to the TO solutions or because of the difficulty of the solver to converge. The thick dashed black lines contour the zone where $m_p < 73$ kg, the oblique black lines correspond to the solutions arriving 6 months, 5 months, 4 months, and 3 months prior to the close encounter, while the oblique continuous red line corresponds to -2 months from the close encounter. After this line, the porkchop is patched with a grey area, indicating rendezvous happening after the latest arrival date on February 13, 2029, and thus not acceptable for system design according to the mission's requirement Table 5.

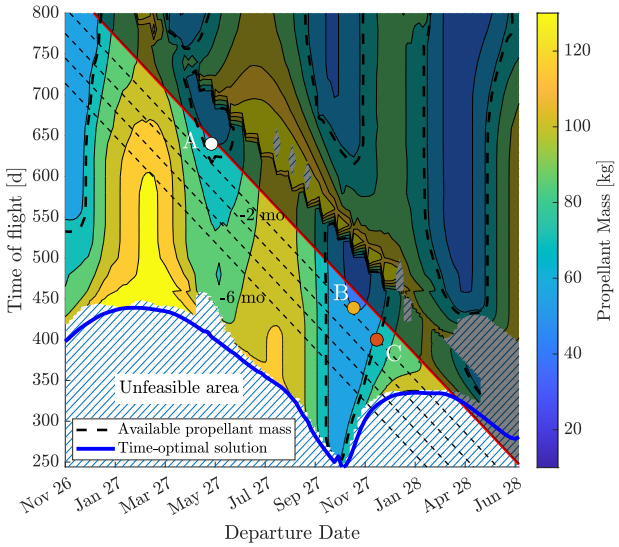


Fig. 10 RAMSES propellant mass porkchop plot (from [29]).

A variable step grid in the ToF dimension is exploited, to help the convergence close to the TO solutions line. From ToF of 244 days up to 460 days, the time step has been considered as 1 day, while a 20-day interval is considered from ToF = 460 days on. Contrarily, the time step for the departure dates is fixed to 5 days. The three letters A, B, and C

*Ariane 6 User Manual I2R0: link here (last accessed: February 22, 2024)

in Fig. 10 indicate the nominal trajectories selected for system sizing. They are reported as trajectories RAMSES.A, RAMSES.B, and RAMSES.C with their characteristics in Table 6.

Table 6 Characteristics of the selected nominal trajectories for RAMSES mission.

ID	Departure date	Arrival date	**ToF** (days)	$\mathbf{m_p}$ (kg)
RAMSES.A	May 10, 2027	Feb 8, 2029	640	70.43026
RAMSES.B	Nov 11, 2027	Jan 23, 2029	439	54.95489
RAMSES.C	Dec 11, 2027	Jan 14, 2029	400	72.22668

In this work, the same assumptions of Table 5 are considered for the re-computation of the porkchop in the presence of different schemes of duty cycles. The FO solutions computed in [29] are used as initial guesses for the algorithm presented in this work, to avoid excessive CPU time in the guessing of a good initial guess through ACT for each of the 11,883 trajectories in the original porkchop. Four different duty cycle schemes have been imposed: $T = 30$ days and $\tau = 25$ days, $T = 20$ days and $\tau = 17$ days, $T = 15$ days and $\tau = 10$ days, and $T = 7$ days and $\tau = 6$ days. They are reported in Table 7, with the respective $(T - \tau)/T$ ratios. Thus, with respect to the previous Subsections, a new scheme having $T = 20$ days and $\tau = 17$ days has been considered. Note that in this case $(T - \tau)/T = 0.1500$. This scheme has been preferred to the one with $T = 5$ days and $\tau = 4$ days used in the previous Subsections for the re-computation of the porkchop because for having a similar $(T - \tau)/T$ ratio but being less computationally expensive. In total, about 50,000 trajectories are computed by applying the algorithm presented.

Each of the four duty cycle schemes has been applied both activating and de-activating the first continuation from EO to FO (see Subsection III.C). Having the FO solutions from the original porkchop, in theory, no EO-to-FO continuation should be necessary. Nevertheless, tests have shown that activating all of the three continuations produces a slightly higher number of converged cases. The FO solutions computed for the original porkchop indeed did not include the time discretization due to the imposition of S_{DC} function constraint (see Subsection III.D, on the imposition of a minimum number of nodes in each coasting arcs). Thus, for trajectories particularly difficult to solve, a first solution of the EO problem with the more stringent time grid seems to ease the convergence.

Each porkchop has gone through 4 iterations of computation. The first one, iteration 0, involved the solution of the problem imposing as an initial guess the corresponding FO solution from the porkchop in Fig. 10, and activating all of the three continuations in Subsection III.C. The other three have been exploited to fill the gaps of the non-converged trajectories and obtain more comprehensive porkchops. At each of the iterations 1, 2, and 3, the up-to-8 converged trajectories around an non-converged one from the previous iteration are used as initial guesses, until the convergence, if happening, is reached. An example of this warm start is shown in Fig. 11, where the red crosses represent non-converged trajectories, and the green circles the converged ones. Consider the non-converged trajectory in B2: at iteration 1, it has 5 possible initial guesses to warm start its convergence. After iteration 1, trajectory B2 has not converged, but the

trajectory A1 has. Thus, at iteration 2, trajectory A1 only is used to warm-start the solution of trajectory B2. This process is performed for each trajectory, for each iteration from 1 to 3, for each duty cycle scheme. Since the converged trajectories yet present the discontinuous thrust profiles of the associated duty cycle scheme, for iterations 1, 2, and 3 the problem has been solved by switching off the EO to FO continuation and imposing for the second and third continuation N_{max} to the maximum number and $u_\zeta = 0$.

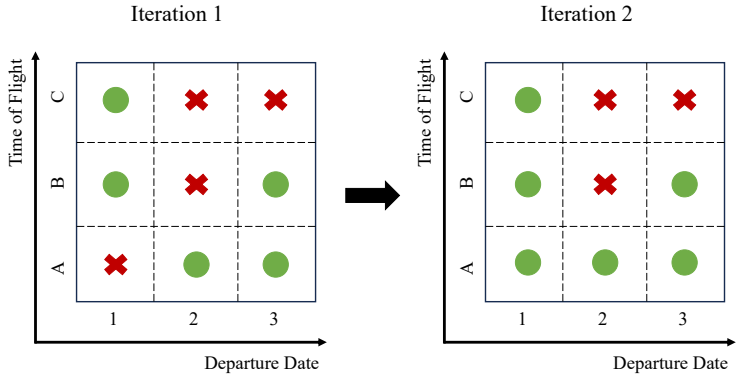


Fig. 11 Continuation scheme used for iterations 1, 2, and 3 for each porkchop.

To reduce the computational time, both parallel computation, with 60 cores of an 80-core Intel Xeon CPU E5@2.10GHz workstation[†], and compilation into MEX file of the integration for the shooting problem have been exploited. Matlab R2022b has been exploited for the computations on the machine. The results of these computations are reported in Table 7. The number N_{conv} represents the absolute number of converged cases in each porkchop, at each iteration. The number N_{unf} represents the number of non-converged cases because under the new TO limit in the ToF, i.e., the unfeasible trajectories. Indeed, since coasting periods are imposed in the thrusting profiles, the TO blue line shown in Fig. 10 is expected to move back to higher ToFs in porkchop with duty cycles, as more trajectories became unfeasible for having insufficient ToF. The percentage of converged cases $\%\, N_{conv}$ are computed with respect to the total number of feasible solutions from the original solutions, i.e., it is the percentage of N_{conv} on (11883 - N_{unf}).

For each duty cycle scheme, as the iterations go on the number of converged cases N_{conv} and $\%\, N_{conv}$ increase, while the number of unfeasible ones N_{unf} decreases, all approaching a plateau with the third iteration. It has to be noted in particular as the convergence rate $\%\, N_{conv}$ is strictly correlated to the ratio $(T - \tau)/T$, being the lowest, 93.85% at the third iteration, for the highest ratio, 0.3333, and the highest, 98.62%, for the lowest ratio, 0.1429. Thus, this ratio seems to be related not only to the increment in propellant mass but also to the success ratio of the algorithm.

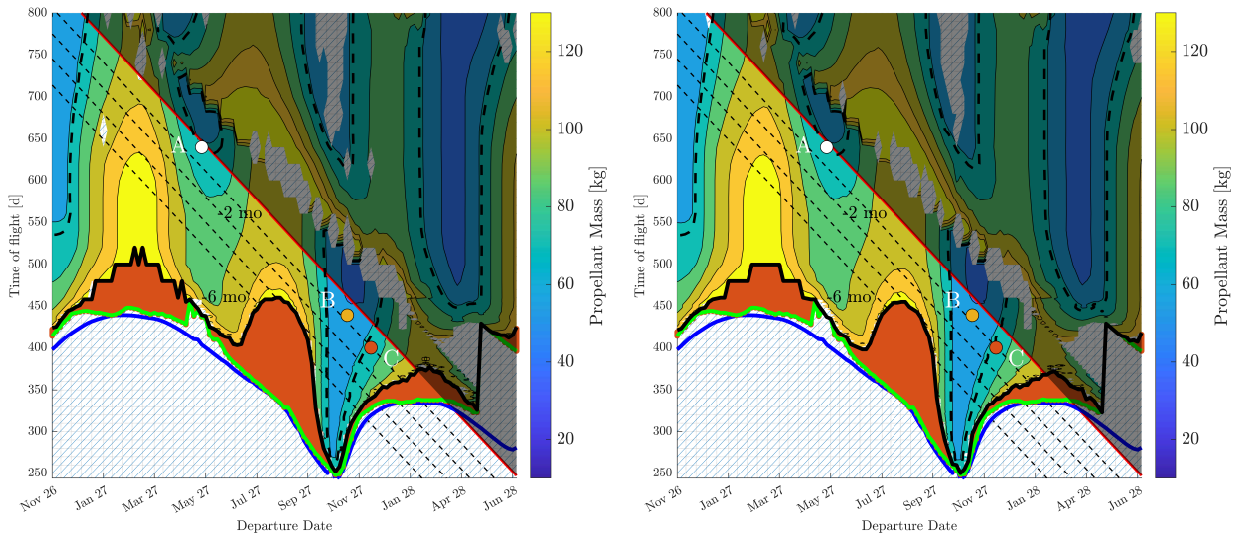
The porkchops related to these results are shown in Figs. from 12 to 13. In them, the legend is the same as in Fig. 10. The blue line still represents the TO solutions of the original porkchop without the duty cycles. A green line has been added to retrace the limit of the original porkchop, while the thick black line retraces the border of the new

[†]This is the Euler workstation at the Department of Aerospace Science and Technology, Politecnico di Milano.

Table 7 Convergence rates for the porkchops with the different duty cycle schemes.

T (days)	τ (days)	$(\mathbf{T}-\tau)/\mathbf{T}$	It.	$\mathbf{N_{conv}}$	$\mathbf{N_{unf}}$	% $\mathbf{N_{conv}}$
30	25	0.1667	0	7393	3756	90.97
			1	7957	3665	97.04
			2	8060	3641	97.79
			3	8083	3632	97.96
20	17	0.1500	0	7912	3344	92.66
			1	8453	3204	97.40
			2	8530	3183	98.05
			3	8552	3176	98.22
15	10	0.3333	0	2778	8432	80.50
			1	3216	8370	91.55
			2	3294	8363	93.58
			3	3312	8354	93.85
7	6	0.1429	0	8146	3305	94.96
			1	8598	3134	98.27
			2	8654	3102	98.55
			3	8668	3094	98.62

porkchop, i.e., representative of the new TO line for the OC trajectories of the new porkchops. In between them, the red area shows where the new unfeasible cases, N_{unf}, are located.

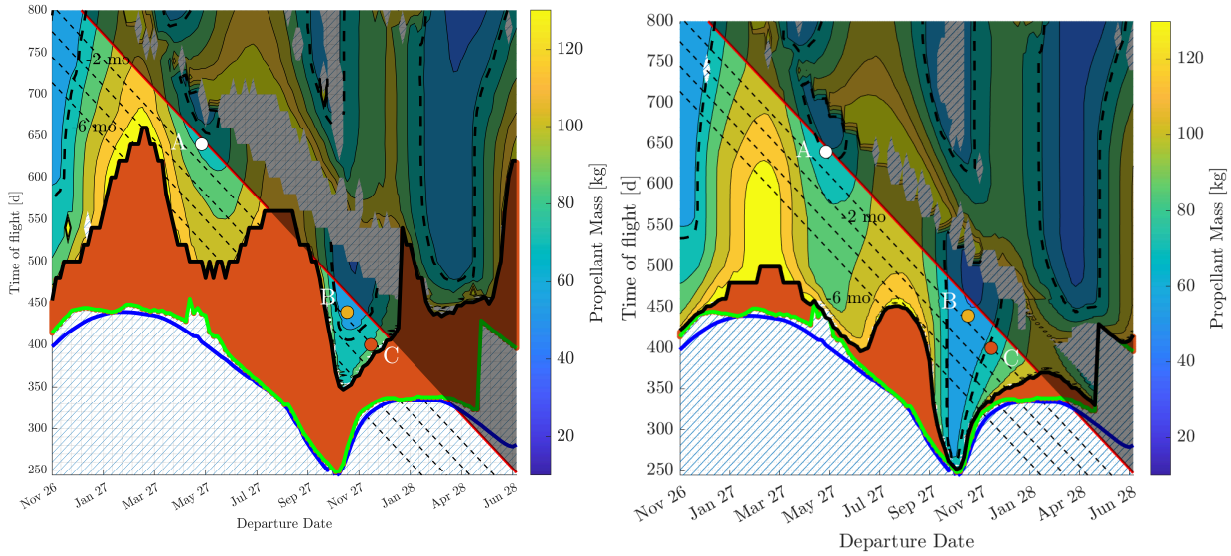


(a) $T = 30$ *days and* $\tau = 25$ *days duty cycle scheme.*

(b) $T = 20$ *days and* $\tau = 17$ *days duty cycle scheme.*

Fig. 12 Porkchops with the $T = 30$ days and $\tau = 25$ days and $T = 20$ days and $\tau = 17$ days duty cycle schemes.

It can be noted that the new TO line moves back to higher ToFs as expected, in particular for the $T = 15$ days and $\tau = 10$ days scheme, where the highest number of unfeasible cases are present. The regression of the TO line suggests



(a) $T = 15$ *days and* $\tau = 10$ *days duty cycle scheme.*

(b) $T = 7$ *days and* $\tau = 6$ *days duty cycle scheme.*

Fig. 13 Porkchops with the $T = 15$ days and $\tau = 10$ days and $T = 7$ days and $\tau = 6$ days duty cycle schemes.

that, for future computations, a more refined time step for ToFs higher than 460 days would ease the convergence and produce more smooth results in the porkchops. For all the schemes, the convergence around the flyby dates (identifiable in the diagonal step discontinuities of the propellant mass in Fig. 10) is exacerbated by the duty cycles imposition. This is particularly visible, again, for the $T = 15$ days and $\tau = 10$ days scheme, in Fig. 13(a).

In Table 8 the results in terms of propellant mass and CPU times are reported. The value $\mu(\Delta m_p)$ is the average increment in propellant mass all over every porkchop at the third iteration with respect to the FO solution computed in the original porkchop in [29], while $\sigma(\Delta m_p)$ is its standard deviation. It is confirmed that the largest increment in propellant mass is related to a higher $(T - \tau)/T$ ratio. For three of the four duty cycle schemes the average increase in propellant mass is 10%, while for the last, the one with $T = 15$ days and $\tau = 10$ days, this increase more than double. The CPU time, in hours, is the time exploited to compute each of the whole porkchop, summing the times for all four iterations. It can be noted that it increases with the decrease of T, as observed in the above Subsections. This pattern is not respected only in the case, again, with $T = 15$ days and $\tau = 10$ days, where the very low number of converged cases (see Table 7) decreases the total CPU time. Indeed, the value reported in Table 8 considers only the CPU time required for the converged cases, while the one for the non-converged and unfeasible ones is excluded from the counting because considered not relevant for the sake of this work. For example, the total CPU time required to compute the converged and non-converged cases for iteration 0 of the scheme with $T = 7$ days and $\tau = 6$ days has been 1374.98 hours, circa 3.5 times higher than the one reported in the table. On average, the non-converged cases require more time because several iterations are performed by the algorithm to find a solution. The value μ(CPU time) is the average time required to compute one converged trajectory in the porkchop, while σ(CPU time) is the standard deviation. As observed in the

previous Subsections, these values increase with the increase of the frequency of the duty cycles.

Table 8 Propellant mass and CPU time data for the porkchops with the different duty cycle schemes.

T (days)	τ (days)	$\mu(\mathbf{\Delta m_p})$ (%)	$\sigma(\mathbf{\Delta m_p})$ (%)	**CPU time** (h)	μ(**CPU time**) (s)	σ(**CPU time**) (s)
30	25	10.85	9.05	67.57	30.09	42.14
20	17	9.86	8.26	106.31	47.35	57.54
15	10	21.83	18.23	58.36	63.44	82.36
7	6	9.32	7.94	395.29	164.17	200.94

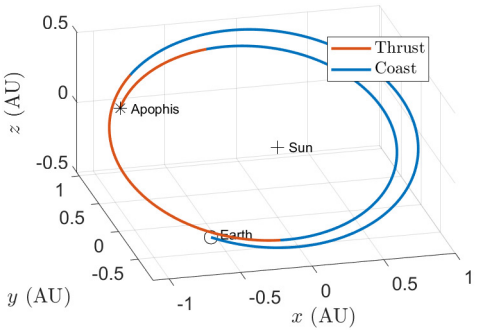
The three trajectories RAMSES.A, RAMSES.B, and RAMSES.C of Table 6 remain feasible in all of the four new porkchops, meaning that they do not fall below the new TO line. Their new characteristics, in terms of propellant mass, m_p, and increase of propellant mass with respect to the values in Table 6, Δm_p, are reported in Table 9.

Table 9 Characteristics of the selected nominal trajectories for RAMSES mission with duty cycles imposition.

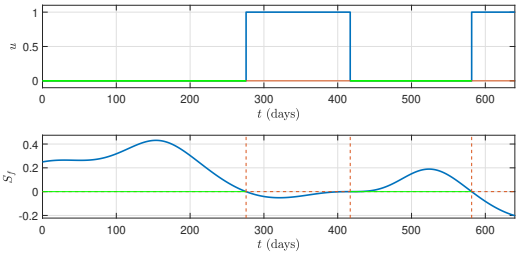
ID	**T** (days)	τ (days)	$\mathbf{m_p}$ (kg)	$\mathbf{\Delta m_p}$ (%)
RAMSES.A	30	25	72.03746	2.2820
	20	17	72.03909	2.2843
	15	10	81.39247	15.5646
	7	6	71.79096	1.9320
RAMSES.B	30	25	57.92087	5.3971
	20	17	57.02260	3.7626
	15	10	64.49493	17.3598
	7	6	56.92094	3.5776
RAMSES.C	30	25	75.77176	4.9083
	20	17	75.82496	4.9819
	15	10	87.83010	21.6034
	7	6	75.57287	4.6329

It can be noted that the highest propellant mass increase is again related to the scheme with $T = 15$ days and $\tau = 10$ days, while the lowest to the scheme with $T = 7$ days and $\tau = 6$ days. For the two other schemes, it can be noted that the increase in propellant mass is very similar between the two for each trajectory. The trajectory profiles are reported in Figs. from 14 to 16, where the same of legend of Subsection IV.A is used.

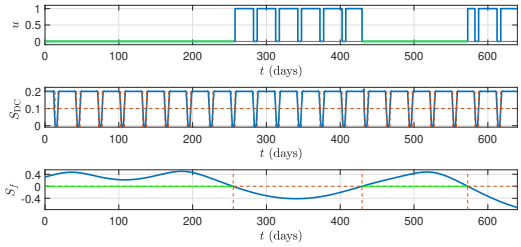
For all the trajectories above, the imposition of the duty cycle scheme splits the pre-existing thrusting blocks in the corresponding FO solutions into several, shorter, blocks, a behavior observed also in [27]. For the scheme with $T = 15$ days and $\tau = 10$ days, the one having the highest $(T - \tau)/T$ ratio, for trajectories RAMSES.A and RAMSES.B the thrusting profiles show additional thrusting blocks (Figs. 14(e) and 15(e)), while for trajectory RAMSES.C, the one having the shortest ToF and fullest thrusting profile, i.e., there is only one long coasting arc, the last two thrusting blocks merge in a unique one (Fig. 16(e)).



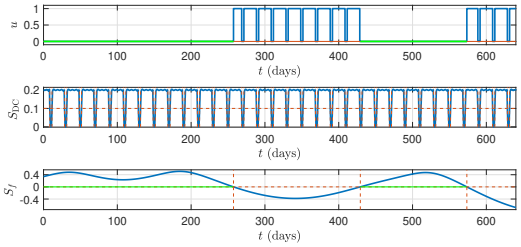
(a) *FO Trajectory RAMSES.A in the J2000 reference frame.*



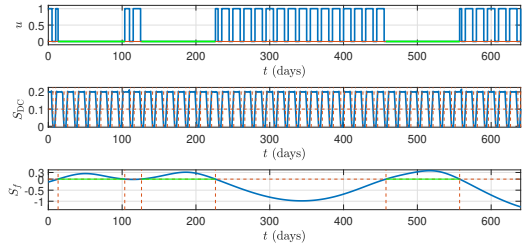
(b) *FO throttle factor and optimal switching function profiles.*



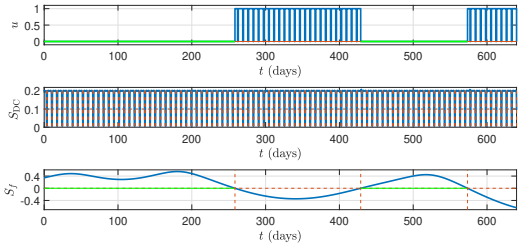
(c) *Duty cycle scheme with* $T = 30$ *days and* $\tau = 25$ *days.*



(d) *Duty cycle scheme with* $T = 20$ *days and* $\tau = 17$ *days.*



(e) *Duty cycle scheme with* $T = 15$ *days and* $\tau = 10$ *days.*

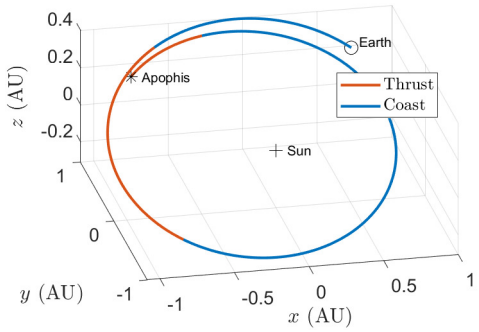


(f) *Duty cycle scheme with* $T = 7$ *days and* $\tau = 6$ *days.*

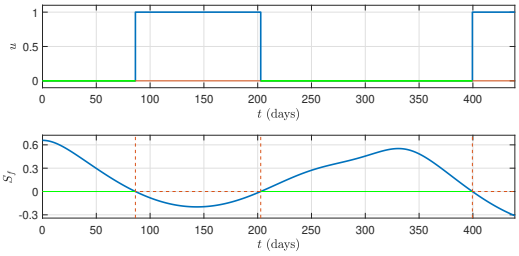
Fig. 14 RAMSES.A trajectory with different duty cycle schemes.

V. Conclusion

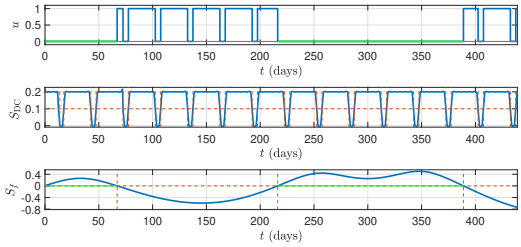
In this work, the problem of the operational compliant (OC) trajectories, low-thrust trajectories characterized by a pre-defined alternation in time of thrusting and coasting arcs, the duty cycles, is investigated. These trajectories are highly discontinuous in nature, and their computation is not easy. A novel formulation of the OC constraint in indirect trajectory optimization is thus presented in this paper. The optimization is based on a triple continuation scheme, on the imposition of the forced coast arcs through control constraint mathematical formalization, and the use of analytic derivatives for robustness and accuracy. The necessary conditions for optimality are derived for the fuel optimal problem. The proposed method is applied to two real-case scenarios, the M-ARGO CubeSat and the mission RAMSES towards the Apophis asteroid, with tests on more than 50,000 trajectories. Results show that the algorithm can generate OC trajectories in a fast way, without any need for prior knowledge of the OC thrust profile. In particular, computational



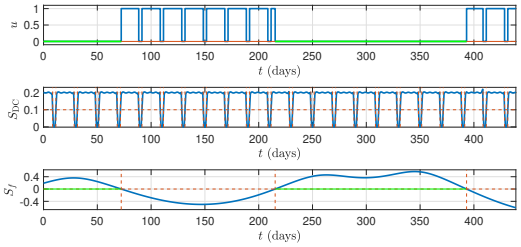
(a) *FO Trajectory RAMSES.B in the J2000 reference frame.*



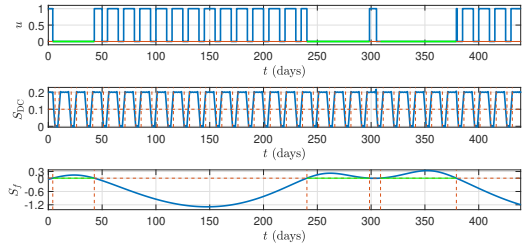
(b) *FO throttle factor and optimal switching function profiles.*



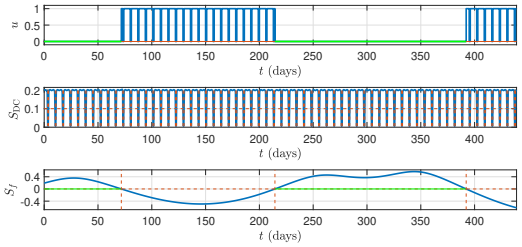
(c) *Duty cycle scheme with* $T = 30$ *days and* $\tau = 25$ *days.*



(d) *Duty cycle scheme with* $T = 20$ *days and* $\tau = 17$ *days.*



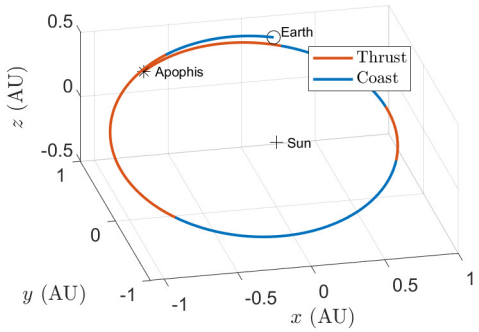
(e) *Duty cycle scheme with* $T = 15$ *days and* $\tau = 10$ *days.*



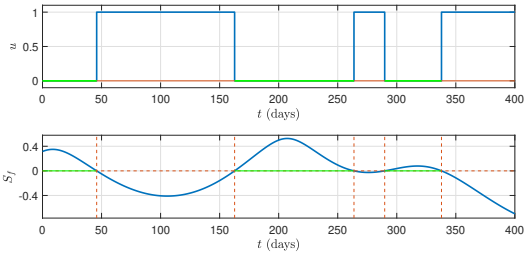
(f) *Duty cycle scheme with* $T = 7$ *days and* $\tau = 6$ *days.*

Fig. 15 RAMSES.B trajectory with different duty cycle schemes.

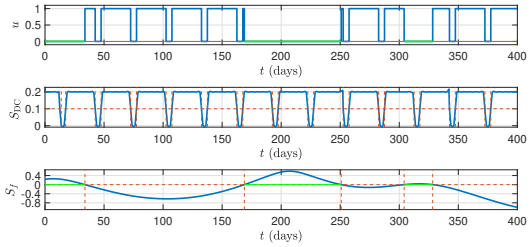
times span from 30 s to 160 s on average per trajectory. The proposed approach computes trajectories with very little increase of propellant mass with respect to the fuel-optimal solution. The lowest increase found is less than 1%, while the highest is around 20%. In particular, it is shown that the highest increase in propellant mass is due to a higher ratio between the imposed coasting time over the duration of the whole duty cycle. The algorithm is shown to be robust and consistent with previous results obtained by the authors. Specifically, the hypothesis that higher values of increases in propellant mass are related to trajectories having yet a high propellant mass cost and having a fuller thrusting profile with almost no long coasting arcs, is confirmed.



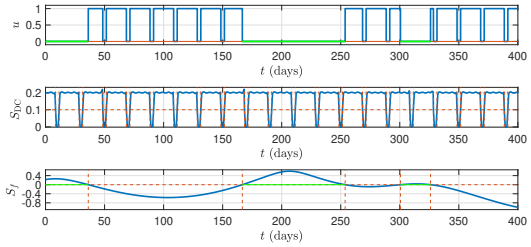
(a) *FO Trajectory RAMSES.C in the J2000 reference frame.*



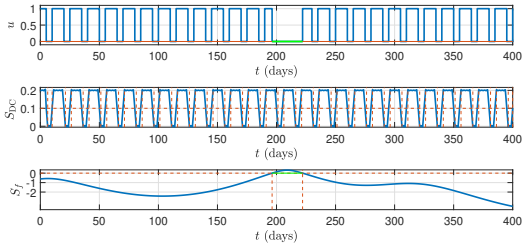
(b) *FO throttle factor and optimal switching function profiles.*



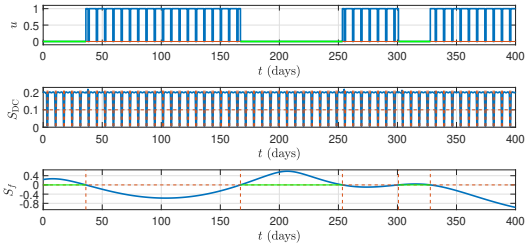
(c) *Duty cycle scheme with* $T = 30$ *days and* $\tau = 25$ *days.*



(d) *Duty cycle scheme with* $T = 20$ *days and* $\tau = 17$ *days.*



(e) *Duty cycle scheme with* $T = 15$ *days and* $\tau = 10$ *days.*



(f) *Duty cycle scheme with* $T = 7$ *days and* $\tau = 6$ *days.*

Fig. 16 RAMSES.C trajectory with different duty cycle schemes.

Appendix

A. Analytical Derivatives

Zero-finding problems are usually solved with some kind of Newton's iteration. This method leverages the first-order derivative of the shooting function to adjust the guess for the problem's variables. To increase the accuracy and robustness of the computation, instead of relying on approximations of these derivatives, in this work analytic derivatives of the shooting problem are computed and provided. In particular, deriving the boundary conditions of Eqs. (19) with respect to zero-finding variable $\boldsymbol{\lambda}_i$ means considering some elements of the state transition matrix (STM) $\boldsymbol{\Phi}(t_i, t)$. The STM maps small variations in the initial conditions $\delta\mathbf{y}_i$ over a generic time instant $t > t_i$, i.e., $\delta\mathbf{y}(t) = \boldsymbol{\Phi}(t_i, t)\,\delta\mathbf{y}(t_i)$.

The STM is subjected to the variational equation

$$\dot{\mathbf{\Phi}}\left(t,t_i\right) = D_y\mathbf{F}\mathbf{\Phi}\left(t,t_i\right) \quad \text{with} \quad \mathbf{\Phi}\left(t_i,t_i\right) = \mathbb{I}_{14} \tag{26}$$

where $\mathbb{I}_n$ represents a square identity matrix of dimension n, and $D_y\mathbf{F}$ is the Jacobian matrix of $\mathbf{F}(\mathbf{y})$, and has has two different expressions, one for $u^* = 0$ or $u^* = 1$, and one for $u^* = \left(\epsilon - S_\epsilon\right)/2\epsilon$. The general form of $D_y\mathbf{F}$ is

$$D_y\mathbf{F} = \begin{bmatrix} \mathbb{O}_3 & \mathbb{I}_3 & \mathbf{O}_3 & \mathbb{O}_3 & \mathbb{O}_3 & \mathbf{O}_3 \\ \mathbb{G} & \mathbb{O}_3 & \mathbb{M}_1 & \mathbb{O}_3 & \mathbb{M}_2 & \mathbb{M}_3 \\ \mathbf{O}_3^T & \mathbf{O}_3^T & \mathbf{M}_4 & \mathbf{O}_3^T & \mathbb{M}_5 & \mathbf{M}_6 \\ -\frac{\delta(\mathbb{G}^T\cdot\boldsymbol{\lambda}_v)}{\delta\mathbf{r}} & \mathbb{O}_3 & \mathbf{O}_3 & \mathbb{O}_3 & -\mathbb{G}^T & \mathbf{O}_3 \\ \mathbb{O}_3 & \mathbb{O}_3 & \mathbf{O}_3 & -\mathbb{I}_3 & \mathbb{O}_3 & \mathbf{O}_3 \\ \mathbf{O}_3^T & \mathbf{O}_3^T & \mathbf{M}_7 & \mathbf{O}_3^T & \mathbb{M}_8 & \mathbf{M}_9 \end{bmatrix} \tag{27}$$

where $\mathbb{O}_n$ represents a square null matrix of dimension $n \times n$, while $\mathbf{O}_n$ represents null vectors of dimension $n \times 1$. In Eq. (27) the matrix $\mathbb{G}(\mathbf{r})$ is defined as

$$\mathbb{G}(\mathbf{r}) = \frac{\partial \mathbf{g}(\mathbf{r})}{\partial \mathbf{r}} = \begin{bmatrix} \frac{3\mu x^2}{r^5} - \frac{\mu}{r^3} & \frac{3\mu xy}{r^5} & \frac{3\mu xz}{r^5} \\ \frac{3\mu xy}{r^5} & \frac{3\mu y^2}{r^5} - \frac{\mu}{r^3} & \frac{3\mu yz}{r^5} \\ \frac{3\mu xz}{r^5} & \frac{3\mu yz}{r^5} & \frac{3\mu z^2}{r^5} - \frac{\mu}{r^3} \end{bmatrix} \tag{28}$$

while the terms $\mathbf{M}_i$ and $\mathbb{M}_i$ define the different forms of $D_y\mathbf{F}$. For $u^* = 0$ or $u^* = 1$ some of them are null, as

$$\begin{aligned} &\mathbb{M}_1 = \frac{\boldsymbol{\lambda}_v}{\lambda_v}\frac{T_{max}u^*}{m^2}, && \mathbb{M}_2 = -\frac{u^*T_{max}}{m}\left(\frac{\mathbb{I}_3}{\lambda_v} - \frac{\boldsymbol{\lambda}_v\cdot\boldsymbol{\lambda}_v^T}{\lambda_v^3}\right), && \mathbb{M}_3 = \mathbf{O}_3 \\ &\mathbf{M_4} = 0, && \mathbb{M}_5 = \mathbf{O}_3^T, && \mathbf{M_6} = 0 \\ &\mathbf{M_7} = \frac{2u^*T_{max}\lambda_v}{m^3}, && \mathbb{M}_8 = -\frac{u^*T_{max}}{m^2}\frac{\boldsymbol{\lambda}_v^T}{\lambda_v}, && \mathbf{M_9} = 0 \end{aligned} \tag{29}$$

while for $u^* = (\epsilon - S_\epsilon)/2\epsilon$ all the terms are nonzero, as

$$\begin{aligned}
\mathbb{M}_1 &= \frac{\boldsymbol{\lambda}_v}{\lambda_v}\frac{T_{max}u^*}{m^2} + \frac{T_{max}I_{sp}g_0}{2m^3\epsilon}\boldsymbol{\lambda}_v, & \mathbb{M}_2 &= -\frac{u^*T_{max}}{m}\left(\frac{\mathbb{I}_3}{\lambda_v} - \frac{\boldsymbol{\lambda}_v\cdot\boldsymbol{\lambda}_v^T}{\lambda_v^3}\right) - \frac{\boldsymbol{\lambda}_v\cdot\boldsymbol{\lambda}_v^T}{\lambda_v^2}\frac{I_{sp}g_0T_{max}}{2m^2\epsilon} \\
\mathbb{M}_3 &= -\frac{\boldsymbol{\lambda}_v}{\lambda_v}\frac{T_{max}}{2m\epsilon}, & \mathbf{M_4} &= \frac{T_{max}\lambda_v}{2m^2\epsilon}, & \mathbb{M}_5 &= -\frac{T_{max}}{2m\epsilon}\frac{\boldsymbol{\lambda}_v^T}{\lambda_v} \\
\mathbf{M_6} &= -\frac{T_{max}}{2\epsilon I_{sp}g_0}, & \mathbf{M_7} &= \frac{2u^*T_{max}\lambda_v}{m^3} - \frac{T_{max}I_{sp}g_0\lambda_v^2}{2m^4\epsilon} \\
\mathbb{M}_8 &= -\frac{u^*T_{max}}{m^2}\frac{\boldsymbol{\lambda}_v^T}{\lambda_v} - \frac{T_{max}I_{sp}g_0}{2m^3\epsilon}\boldsymbol{\lambda}_v^T, & \mathbf{M_9} &= -\frac{T_{max}\lambda_v}{2m^2\epsilon}
\end{aligned} \tag{30}$$

To be exploited in the zero-finding problem $\boldsymbol{\Phi}(t_i, t)$ has to be integrated with states and costates as it is subject to a variational equation. Defining $\mathbf{z} = [\mathbf{y},\ \mathrm{vec}(\boldsymbol{\Phi})]$ the 210-dimensional vector containing the 14 variables of $\mathbf{y}$ and the 196 elements of $\boldsymbol{\Phi}$, where the vec operator converts a matrix into a vector, the equation to integrate becomes

$$\dot{\mathbf{z}} = \mathbf{C}(\mathbf{z}) \Rightarrow \begin{cases} \dot{\mathbf{y}} = \mathbf{F}(\mathbf{y}) \\ \mathrm{vec}\left(\dot{\boldsymbol{\Phi}}\right) = \mathrm{vec}\left(D_y\mathbf{F}\boldsymbol{\Phi}\right) \end{cases} \tag{31}$$

However, the STM $\boldsymbol{\Phi}$ maps states and costates along a continuous trajectory, and the bang-bang profile given by Eq. (17) creates discontinuities in the flow. At each switching time t_1, t_2, ..., t_N, where either S_ϵ crosses 0 or S_{DC} crosses ϵ_{DC}, compensation matrices in the STM $\boldsymbol{\Psi}(t_1)$, $\boldsymbol{\Psi}(t_2)$, ..., $\boldsymbol{\Psi}(t_N)$ have to be added using the chain rule [38]

$$\boldsymbol{\Phi}\left(t_f, t_i\right) = \boldsymbol{\Phi}\left(t_f, t_N^+\right)\boldsymbol{\Psi}\left(t_N\right)\boldsymbol{\Phi}\left(t_N^-, t_{N-1}^+\right)\boldsymbol{\Psi}\left(t_{N-1}\right)\ldots\boldsymbol{\Psi}\left(t_2\right)\boldsymbol{\Phi}\left(t_2^-, t_1^+\right)\boldsymbol{\Psi}\left(t_1\right)\boldsymbol{\Phi}\left(t_1^-, t_i\right) \tag{32}$$

where N is the total number of switching times. Suppose that the discontinuity is detected at t_s either for the switching function S_ϵ crossing 0 or for S_{DC} crossing ϵ_{DC}, and thus the optimal control u^* jumps from 0 to 1 or vice versa. In both cases $\mathbf{y}$ is continuous, but $\dot{\mathbf{y}}$ is discontinuous as the dynamics of $\mathbf{v}(t)$ is influenced by u^*. It can be proved [38] that the compensation matrix $\boldsymbol{\Psi}(t_s)$ is given by

$$\boldsymbol{\Psi}\left(t_s\right) = \frac{\partial\mathbf{y}\left(t_s^+\right)}{\partial\mathbf{y}\left(t_s^-\right)} = \mathbb{I}_{14} + \left(\dot{\mathbf{y}}\left(t_s^+\right) - \dot{\mathbf{y}}\left(t_s^-\right)\right)\frac{1}{\dot{S}\left(t_s^-\right)}\frac{\partial S}{\partial\mathbf{y}}\bigg|_{t_s^-} \tag{33}$$

where S can represent S_{DC} or S_ϵ. In the case of the optimal switching function S_ϵ, the compensation matrix is indicated as $\boldsymbol{\Psi}_\epsilon(t_s)$, and its computed with Eq. (33) where the derivatives of S_ϵ are

$$\frac{\partial S_\epsilon}{\partial\mathbf{y}} = \left[\mathbf{O}_6^T,\ \frac{I_{sp}g_0}{m^2}\lambda_v,\ \mathbf{O}_3^T,\ -\frac{I_{sp}g_0}{m}\frac{\boldsymbol{\lambda_v}^T}{\lambda_v},\ -1\right], \quad \dot{S}_\epsilon = \frac{I_{sp}g_0}{m\lambda_v}\boldsymbol{\lambda}_r\cdot\boldsymbol{\lambda}_v \tag{34}$$

If the jump for u^* from 0 to 1 or vice versa at t_s is instead due to the switching function S_{DC} crossing ϵ_{DC}, the compensation matrix is indicated as $\mathbf{\Psi}_{DC}(t_s)$. Since $\partial S_{DC}/\partial \mathbf{y} = \mathbf{0}$, it results that

$$\mathbf{\Psi}_{DC}(t_s) = \frac{\partial \mathbf{y}\left(t_s^+\right)}{\partial \mathbf{y}\left(t_s^-\right)} = \mathbb{I}_{14} \tag{35}$$

Indeed $\mathbf{\Psi}(t_s)$ maps the effect of variations of $\mathbf{y}\left(t_s^-\right)$ in variations in $\mathbf{y}\left(t_s^+\right)$, and since S_{DC} does not depend on $\mathbf{y}$ but only on time, no effects are expected on the states or the costates. Therefore, during the integration of Eq. (31), no compensations to the STM are necessary for switching in the throttle factor due to S_{DC}, but only for switching due to S_ϵ.

Funding Sources

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 864697). A. M. and F. T. would also like to acknowledge the funding received under ESA Contracts No. 4000136010/21/NL/GLC/my.

References

[1] Jehn, R., Companys, V., Corral, C., García Yárnoz, D., and Sánchez, N., "Navigating BepiColombo during the weak-stability capture at Mercury," *Advances in Space Research*, Vol. 42, No. 8, 2008, pp. 1364–1369. https://doi.org/10.1016/j.asr.2008.01.011.

[2] Montagnon, E., Budnik, F., Casale, M., De la Fuente, S., Martinez, S., Murakami, G., Ogawa, M., Seki, T., Steiger, C., and Yamashita, M., "BepiColombo ground segment and mission operations," *Space Science Reviews*, Vol. 217, No. 32, 2021, pp. 1–23. https://doi.org/10.1007/s11214-021-00805-y.

[3] Pascoa, J. C., Teixeira, O., and Filipe, G., "A review of propulsion systems for CubeSats," *ASME International Mechanical Engineering Congress and Exposition*, V001T03A039, Vol. 1, Pittsburg, PA, 2018. https://doi.org/10.1115/IMECE2018-88174.

[4] Walker, R., Binns, D., Bramanti, C., Casasco, M., Concari, P., Izzo, D., Feili, D., Fernandez, P., Fernandez, J. G., Hager, P., Koschny, D., Pesquita, V., Wallace, N., Carnelli, I., Khan, M., Scoubeau, M., and Taubert, D., "Deep-space CubeSats: thinking inside the box," *Astronomy & Geophysics*, Vol. 59, No. 5, 2018, pp. 5.24–5.30. https://doi.org/10.1093/astrogeo/aty232.

[5] Cervone, A., Topputo, F., Speretta, S., Menicucci, A., Turan, E., Di Lizia, P., Massari, M., Franzese, V., Giordano, C., Merisio, G., Labate, D., Pilato, G., Costa, E., Bertels, E., Thorvaldsen, A., Kukharenka, A., Vennekens, J., and Walker, R., "LUMIO: A CubeSat for observing and characterizing micro-meteoroid impacts on the Lunar far side," *Acta Astronautica*, Vol. 195, 2022, pp. 309–317. https://doi.org/10.1016/j.actaastro.2022.03.032.

[6] Ferrari, F., Franzese, V., Pugliatti, M., Giordano, C., and Topputo, F., "Trajectory options for Hera's Milani CubeSat around (65803) Didymos," *The Journal of the Astronautical Sciences*, Vol. 68, No. 4, 2021, pp. 973–994. https://doi.org/10.1007/s40295-021-00282-z.

[7] Goldberg, H. R., Karatekin, Ö., Ritter, B., Herique, A., Tortora, P., Prioroc, C., Gutierrez, B. G., Martino, P., and Carnelli, I., "The Juventas CubeSat in support of ESA's Hera mission to the asteroid Didymos," *Proceedings of the Small Satellite Conference*, SSC19-WKIV-05, Vol. 1, Utah State University, Logan, UT, 2019.

[8] Martin-Mur, T. J., Gustafson, E. D., and Young, B. T., "Interplanetary CubeSat navigational challenges," *25th International Symposium on Space Flight Dynamics (ISSFD)*, Munich, Germany, 2015.

[9] Woellert, K., Ehrenfreund, P., Ricco, A. J., and Hertzfeld, H., "Cubesats: Cost-effective science and technology platforms for emerging and developing nations," *Advances in Aerospace Research*, Vol. 47, No. 4, 2011, pp. 663–684. https://doi.org/10.1016/j.asr.2010.10.009.

[10] Bryson, A., and Ho, Y.-C., *Applied optimal control*, Taylor & Francis, London, 1975. https://doi.org/10.1201/9781315137667, Chaps. 1–3.

[11] Hull, D. G., *Optimal control theory for applications*, Springer Science & Business Media, Berlin, Germany, 2013. https://doi.org/10.1007/978-1-4757-4180-3, Chaps. 1–4.

[12] 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.

[13] Enright, P. J., and Conway, B. A., "Discrete approximations to optimal trajectories using direct transcription and nonlinear programming," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 994–1002. https://doi.org/10.2514/3.20934.

[14] Hargraves, C. R., and Paris, S. W., "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 338–342. https://doi.org/10.2514/3.20223.

[15] Dei Tos, D. A., and Topputo, F., "High-fidelity trajectory optimization with application to saddle-point transfers," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 6, 2019, pp. 1343–1352. https://doi.org/10.2514/1.G003838.

[16] Kechichian, J. A., "Optimal low-Earth-orbit-geostationary-Earth-orbit intermediate acceleration orbit transfer," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 4, 1997, pp. 803–811. https://doi.org/10.2514/2.4116.

[17] Ranieri, C. L., and Ocampo, C. A., "Indirect optimization of three-dimensional finite-burning interplanetary transfers including spiral dynamics," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 2, 2009, pp. 445–455. https://doi.org/10.2514/1.38170.

[18] Morante, D., Sanjurjo Rivo, M., and Soler, M., "A survey on low-thrust trajectory optimization approaches," *Aerospace*, Vol. 8, No. 3, 2021, p. 88. https://doi.org/10.3390/aerospace8030088.

[19] Betts, J. T., *Practical methods for optimal control and estimation using nonlinear programming*, SIAM, Philadelphia, PA, 2010. Chaps. 1, 3.

[20] Jiang, F., Baoyin, H., and Li, J., "Practical techniques for low-thrust trajectory optimization with homotopic approach," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 1, 2012, pp. 245–258. https://doi.org/10.2514/1.52476.

[21] Bertrand, R., and Epenoy, R., "New smoothing techniques for solving bang–bang optimal control problems—numerical results and statistical interpretation," *Optimal Control Applications and Methods*, Vol. 23, No. 4, 2002, pp. 171–197. https://doi.org/10.1002/oca.709.

[22] Epenoy, R., and Bertrand, R., "Optimal control and smoothing techniques for computing minimum fuel orbital transfers and rendezvous," *18th International Symposium on Space Flight Dynamics (ISSFD)*, Vol. 548, Munich, Germany, 2004, pp. 131–136.

[23] Wang, Y., and Topputo, F., "Indirect optimization of power-limited asteroid rendezvous trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 5, 2022, pp. 131–136. https://doi.org/10.2514/1.G006179.

[24] Wang, Y., and Topputo, F., "Indirect optimization of fuel-optimal many-revolution low-thrust transfers with eclipses," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 59, No. 1, 2022, pp. 39–51. https://doi.org/10.1109/TAES.2022.3189330.

[25] Epenoy, R., "Fuel optimization of low-thrust trajectories under thrust and coast times constraints: a novel indirect approach," *Acta Astronautica*, Vol. 206, 2023, pp. 218–232. https://doi.org/10.1016/j.actaastro.2023.02.030.

[26] Nurre, N. P., and Taheri, E., "Duty-cycle-aware low-thrust trajectory optimization using embedded homotopy," *Acta Astronautica*, Vol. 212, 2023, pp. 630–642. https://doi.org/10.1016/j.actaastro.2023.08.022.

[27] Mannocchi, A., Giordano, C., and Topputo, F., "A Homotopic Direct Collocation Approach for Operational-Compliant Trajectory Design," *The Journal of the Astronautical Sciences*, Vol. 69, 2022, pp. 1649–1665. https://doi.org/10.1007/s40295-022-00351-x.

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

[29] Morelli, A. C., Mannocchi, A., Giordano, C., Ferrari, F., and Topputo, F., "Initial Trajectory Assessment of a low-thrust option for the RAMSES Mission to (99942) Apophis," *Advances in Space Research*, Vol. 73, No. 8, 2024, pp. 4241–4253. https://doi.org/10.1016/j.asr.2024.02.005.

[30] Conway, B. A., *Spacecraft trajectory optimization*, Cambridge University Press, Cambridge, UK, 2010. https://doi.org/10.1017/CBO9780511778025, Pages 17–20.

[31] Conway, B. A., "A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems," *Journal of Optimization Theory and Applications*, Vol. 152, 2012, pp. 271–306. https://doi.org/10.1007/s10957-011-9918-z.

[32] Longuski, J. M., Guzmán, J. J., and Prussing, J. E., *Optimal control with aerospace applications*, Springer, Berlin, Germany, 2014. https://doi.org/10.1007/978-1-4614-8945-0, Pages 199–202.

[33] Verner, J. H., "Numerically optimal Runge–Kutta pairs with interpolants," *Numerical Algoithms*, Vol. 53, 2010, pp. 383–396. https://doi.org/10.1007/s11075-009-9290-3.

[34] Taheri, E., and Junkins, J. L., "Generic smoothing for optimal bang-off-bang spacecraft maneuvers," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 11, 2018, pp. 2470–2475. https://doi.org/10.2514/1.G003604.

[35] 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.

[36] Giorgini, J. D., and Yeomans, D. K., "On-line system provides accurate ephemeris and related data," Tech. Rep. NPO-20416, NASA, 1999.

[37] Powell, M. J. D., "A Fortran subroutine for solving systems of nonlinear algebraic equations," , No. AERE-R-5947, 1968.

[38] Russell, R. P., "Primer vector theory applied to global low-thrust trade studies," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 2, 2007, pp. 460–472. https://doi.org/10.2514/1.22984.

[39] Sokolov, L., Bashakov, A., Borisova, T., Petrov, N., Pitjev, N., and Shaidulin, V. S., "Impact trajectories of the asteroid Apophis in the 21st century," *Solar System Research*, Vol. 46, 2012, pp. 291–300. https://doi.org/10.1134/S0038094612040077.