



POLITECNICO
MILANO 1863

RE.PUBLIC@POLIMI

Research Publications at Politecnico di Milano

Post-Print

This is the accepted version of:

F. Spada, M. Sagliano, F. Toppato
Direct-indirect Hybrid Strategy for Optimal Powered Descent and Landing
Journal of Spacecraft and Rockets, Vol. 60, N. 6, 2023, p. 1787-1804
doi:10.2514/1.A35650

The final publication is available at <https://doi.org/10.2514/1.A35650>

Access to the published version may require subscription.

When citing this work, cite the original published paper.

Permanent link to this version

<http://hdl.handle.net/11311/1246878>

Direct-Indirect Hybrid Strategy for Optimal Powered Descent and Landing

Fabio Spada*

Politecnico di Milano, Milan 20133, Italy

Marco Sagliano[†]

DLR, German Aerospace Center, Bremen 28359, Germany

Francesco Topputo[‡]

Politecnico di Milano, Milan 20156, Italy

A hybrid algorithm to solve optimal control problems is discussed in the present paper, and applied to the powered descent guidance (PDG) problem. A reference solution is first obtained via a convex direct solver, and is then used as guess for the primal-dual boundary value problem associated with the initial problem. In this context, a covector mapping theorem is used to map the multipliers of the direct solution to the corresponding discrete costates of the indirect method. Collocation based on *hp* pseudospectral scheme is employed for the **convex** direct step, while single shooting for the indirect step. A switching-detection technique further equips the shooting. **As opposed to the hybrid convex-indirect algorithm**, a state-of-the-art purely indirect algorithm is outlined: such approach merges the same single shooting approach with a homotopic continuation. The proposed methods are applied to the pinpoint landing formulation of the PDG, framed in a 3-D environment. Results are finally outlined, comparing the proposed hybrid strategy to the purely indirect approach. The outcome highlights the gain in computational times for the hybrid optimization technique over the fully homotopic scheme, demonstrating the validity of the former for landing trajectory optimization purposes.

I. Introduction

OPTIMAL control constitutes a widely adopted mathematical formulation of problems for a vast variety of engineering applications. Traditionally, indirect and direct methods for solving Optimal Control Problems (OCPs) have been opposed to each other; the former firstly augment the original primal dynamics to its equivalent Hamiltonian, namely primal-dual, then solve the associated Boundary Value Problem (BVP). The latter transcribe the cost function, constraints

Presented as Paper 2023-2502 at the AIAA Scitech 2023 Forum, National Harbor, MD & Online, January 23-27, 2023

*M.Sc. Graduate, Piazza Leonardo da Vinci, 32; fabio.spada@mail.polimi.it. Member AIAA (Corresponding Author).

[†]Senior GNC Research Engineer, Navigation and Control Department, Robert Hooke Str. 7; marco.sagliano@dlr.de. Senior Member AIAA.

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

and dynamics into discrete constraints, and cast the OCP as a Nonlinear Programming (NLP) problem. Indirect methods' workhorse is solution accuracy [1], as the BVP directly stems from the optimality necessary conditions of the original problem; on the other hand, they are penalized by reduced extension of convergence basin; convergence either to a non-physical or to a suboptimal solution is indeed likely [2]. Direct methods allow instead to easily include constraints, hence the versatility of algorithms based on such approaches. Eventually, the original problem can be convexified, thus reformulating the NLP as Convex Programming (CP) problem; this formulation represents a key technology for onboard computation purposes [3, 4], and allows the use of highly performing interior-point algorithms, ensuring convergence to the convex sub-problem optimum in polynomial time [5]. Lossless [6, 7] or successive convexification [8–12] techniques are the most common approaches in this context. However, solution refinement determines rapid increase of computational loads, whether OCP is transcribed as NLP or CP problem.

Solutions to the main drawbacks of both methods have been vastly analyzed in the latest years. Homotopic continuation techniques have been introduced for indirect solvers to overcome initial guess sensitivity [13]. A starting subproblem differing from the original one is solved first; the solution is then used as guess for a second subproblem, more similar to the original problem. Such process is repeated iteratively, up to when the second subproblem matches the original. Keys for the homotopic approach to succeed are 1) the formulation of the starting subproblem with a large enough convergence basin and 2) a design of continuation scheme ensuring that each solution lies in the convergence basin of the successive subproblem. The difference of the generic subproblem with respect to the original lies on a different functional formulation of the objective function, constraints or system dynamics. A homotopic parameter is used to gradually transform the starting functional back to the original one. In the context of offline optimal control, low thrust trajectories [14, 15], rocket ascent and descent [16, 17] and hypersonic reentry [18–21] make up valid application examples. Pseudospectral (PS) collocation methods have come into the limelight for what concerns direct methods: for smooth problems, their solution converges the OCP one with quasi-exponential rate [22], hence the requirement of few nodes only. Moreover, local collocation schemes allow to increase matrix sparsity, further alleviating computational weights [23, 24]. Validity of pseudospectral methods is confirmed by the variety of dynamic environments they are applied in, especially in combination with convex formulations: relative trajectory design [25, 26], deep-space trajectories [27], landing scenarios [28, 29], hypersonic entry [30] and multistage rocket guidance [31] are some examples from the aerospace field; proofs of their performance have further been provided for different engineering problems [32, 33]. Nonetheless, a complete coverage of requirements dictated by practical applications still lacks in both cases. For indirect methods, homotopically continuing a reference solution heavily penalizes computational times. For direct methods, polynomial shape functions cannot approximate non-regular dynamics accurately; discontinuities, however, characterize the optimal control profile in a vast majority of dynamical environments, hence the need for approaches tailored to the specific problem.

In this complex framework, a major breakthrough is represented by the work of Fahroo and Ross: tenet of their

research is the existence of a connection between the covectors of an OCP and the multipliers solving the corresponding NLP [34]; such nexus is referred to as Covector Mapping Principle (CMP). Its mathematical formulation, called Covector Mapping Theorem (CMT), varies with the employed transcription scheme. CMP is proven to exactly map Lagrange multipliers to discrete costates only for some pseudospectral methods, such as the Gauss-Pseudospectral [35] method and the Radau-Pseudospectral method [36]. Quadrature weights are employed to build the mapping between multipliers and costates. CMT has been previously used either to verify optimality of solution obtained from a direct method [37] or to obtain accurate guess costates for an indirect formulation from the equivalent direct one. ISS zero-propellant maneuver [38] constitutes a valuable example in this context.

Within indirect methods, CMT-based guess generation represents a promising approach. Indeed, the idea of warm-starting an indirect method using a direct method has been proposed by Bulirsch [39], however prior to the derivation of the CMT made by Ross [34]. In addition, if the original problem is reformulated as a convex one, guesses for the indirect solver are generated quickly and reliably, as demonstrated in [40]; in this last case, however, the exactness provided by the CMT has not been taken into account. In this work we propose to use an exact CMT, mainly thought for verification purposes, to combine convex and indirect methods into a hybrid algorithm. The first and main contribution of this paper is the development of a hybrid algorithm to solve the Powered Descent Guidance problem for a pinpoint landing formulation. As second contribution, we improve a previously-developed purely indirect approach based on homotopic continuation and apply it to the PDG problem. We then compare the performances of the original hybrid approach with the fully indirect homotopy-based algorithm; we demonstrate the supremacy of the proposed convex-indirect scheme over the state-of-art homotopic approach. For what concerns the hybrid strategy, the PDG problem is transcribed and solved first with an *hp* Radau-Pseudospectral method. Such direct step exploits a convexified formulation of the problem: successive convexifications to handle free final time are combined with the dynamics of the losslessly convexified formulation of the problem to handle the non-convex annular thrust constraint. The PDG OCP formulation as convex optimization problem ensures evaluating a guess with reduced computational times and convergence guarantees; the employed Radau-Pseudospectral method is chosen for its unique mapping properties between Lagrange multipliers and discrete costates [36]. The free-final-time formulation is necessary within the PDG problem as mass consumption is unimodal with respect to time of flight [6]; such formulation turns the linear dynamical constraints associated with the losslessly convexified PDG OCP into nonlinear, hence the need for successive convexifications. The following indirect step is based on a single shooting exploiting the analytical Jacobian of the dynamics; a switching detection technique is employed to integrate the primal-dual system with high accuracy over the bang-bang control profile. The indirect single shooting allows to cope with uncertainty over the number of switches whilst ensuring that the solution coincides with the theoretical *bang-bang* control profile. Indeed, the PDG OCP admits at maximum two switches, as analytically and practically demonstrated in literature [41, 42]; the use of a single shooting minimizes the dimension of the solved problem while further improving the convex step accuracy. Dynamics' analytical

Jacobian increases STM propagation accuracy when compared its numerical counterpart, resulting in a more robust correction step within the shooting.

Considering instead the homotopy-based algorithm, the objective function of the original formulation is generalized to a quadratic functional, and the corresponding optimal PDG problem is solved iteratively. The same shooting technique used for the indirect step of the hybrid approach is employed for each subproblem of the homotopy; primal-dual dynamics, analytical Jacobians and the switching detection technique are modified according to the generalized quadratic objective function. The chosen homotopic path gradually increases steepness in the optimal control profile, ensuring successful continuation of the OCP with reduced number of homotopic steps; use of analytical Jacobians is motivated by the same reasoning of the previous paragraph.

An overview of the contents division is here provided. In Sec. II the Optimal Control Problem with its dual Hamiltonian augmentation is provided and specialized for the Powered Descent Guidance problem. Section III is focused on the indirect method used to solve the problem outlined in Sec. II: the shooting approach is explained along with the techniques employed to integrate the bang-bang profile. Section IV develops the purely indirect algorithm, introducing the homotopic continuation scheme and the quadratic objective function characterizing the generic homotopic subproblem. In Sec. V the hybrid algorithm is presented and analyzed, providing a deep-dive into the guess-generation system based on the direct convex approach: the global PS method with relative CMT, the employed *hp* collocation scheme and the adopted convexification procedure are sequentially explained. Multipliers-to-costates mapping completes the section. In Sec. VI firstly performances of the hybrid technique and of the homotopic approach are compared; then, for what concerns the hybrid algorithm, accuracy improvements due to the indirect step with respect to the direct one are presented; finally, hybrid algorithm robustness is tested with two Monte-Carlo analyses. Conclusions are drawn in Sec. VII.

II. Optimal Powered Descent Guidance Problem

A. Optimal Control Problem and Hamiltonian Augmentation

Throughout the paper the notation is such that a boldface symbol indicates a vector, and the lightface form of the same symbol indicates the magnitude of this vector.

Let the analyzed physical system be autonomous, characterized at each $t \in [t_0, t_f]$, with t_0 assumed equal to 0, by the state $\mathbf{x}(t) \in \mathbb{R}^n$, and by the control $\mathbf{u}(t) \in \mathbb{R}^m$. States belong to space of continuous functions defined over $[0, t_f]$, namely $C([0, t_f], \mathbb{R}^n)$; controls are almost-everywhere continuous functions and belong to $C([0, t_f] - Q_t, \mathbb{R}^m)$, where Q_t contains the instants which control is discontinuous at. Let us consider fixed *initial conditions* for all states and fixed *final conditions* for the first k states. The subscripts $(\cdot)_0$ and $(\cdot)_f$ denote the quantity (\cdot) evaluated respectively at initial

and final time. The *boundary conditions* (BCs) therefore read

$$\mathbf{x}_0 = \bar{\mathbf{x}}_0 \quad \text{and} \quad x_{i,f} = \bar{x}_{i,f}, \quad i = 1, \dots, k \quad (1)$$

Given the right-hand side (RHS) $\mathbf{f} : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^n$, the corresponding *dynamical constraints* dictate the flow of the initial states, and read

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad \text{with} \quad t \in [0, t_f] \quad (2)$$

where the dependency of states and controls with respect to time has been dropped for notation ease.

Moreover, defining the scalar function $g_u : \mathbb{R}^{m+1} \rightarrow \mathbb{R}$, \mathbf{u} shall satisfy the scalar *pure control constraint* expressed by

$$g_u(\mathbf{u}, t) \leq 0 \quad \text{with} \quad t \in [0, t_f] \quad (3)$$

A control profile $\mathbf{u}(t)$ satisfying Eq. (3) is said to belong to the *set of admissible controls* $\mathcal{U} \subseteq \mathbb{R}^m$.

At last, let us introduce the *cost function*, or, more precisely, the *cost functional* [43], $\mathcal{J} : C([0, t_f], \mathbb{R}^n) \times C([0, t_f], \mathbb{R}^m) \times \mathbb{R} \rightarrow \mathbb{R}$. The cost function, with some abuse of notation regarding its domain, is defined as

$$\mathcal{J}(\mathbf{x}, \mathbf{u}, t_f) \doteq \int_0^{t_f} \mathcal{L}(\mathbf{x}, \mathbf{u}, t) dt \quad (4)$$

where $\mathcal{L} : \mathbb{R}^{n+m+1} \rightarrow \mathbb{R}$ is referred to as *path* (or *running*) *cost*.

Under the aforementioned assumptions, the Optimal Control Problem (OCP) is formalized by

$$\min_{\mathbf{u}} \mathcal{J}(\mathbf{x}, \mathbf{u}, t_f) \quad \text{s.t.} \quad \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ g_u(\mathbf{u}, t) \leq 0 \\ \mathbf{x}_0 = \bar{\mathbf{x}}_0 \\ x_{i,f} = \bar{x}_{i,f}, \quad i = 1, \dots, k \end{cases} \quad (5)$$

To dualize the problem we shall introduce the *costates*; it is furthermore convenient to define the *Hamiltonian*. Costates at generic instant t are indicated by $\lambda(t) \in \mathbb{R}^n$; the Hamiltonian $\mathcal{H} : \mathbb{R}^{2n+m+1} \rightarrow \mathbb{R}$ is defined according to

$$\mathcal{H}(\mathbf{x}, \mathbf{u}, \lambda, t) \doteq \mathcal{L}(\mathbf{x}, \mathbf{u}, t) + \lambda^T \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (6)$$

Remark 1: Costates have been introduced to define the Hamiltonian; actually, they are defined as the continuous-time multipliers of the dynamical constraints, rearranged as $\mathbf{f}(\mathbf{x}, \mathbf{u}) - \dot{\mathbf{x}} = \mathbf{0}$. Handling Problem (5) requires indeed an

augmented version of the cost functional \mathcal{J} , built including the dynamical constraints; the interested reader is referred to [2] for further details.

The *first-order optimality necessary conditions* can be obtained combining the *Euler-Lagrange (EL) equations for the OCP* and the *Pontryagin's Minimum Principle (PMP)* [2, 44], the former consisting in *costate dynamics* and *transversality conditions (TCs)*, *i.e.* boundary conditions on costates and Hamiltonian. The necessary conditions read

$$\left\{ \begin{array}{ll} \text{State dynamics:} & \dot{\mathbf{x}} = \nabla_{\mathbf{x}} \mathcal{H} \\ \text{Costate dynamics:} & \dot{\boldsymbol{\lambda}} = -\nabla_{\mathbf{x}} \mathcal{H} \doteq \mathbf{f}_{\boldsymbol{\lambda}} \\ \text{PMP:} & \mathbf{u}^* = \underset{\mathbf{u} \in \mathcal{U}}{\operatorname{argmin}} \left\{ \mathcal{H}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) \right\} \end{array} \right. \quad \text{with BCs and TCs} \quad \left\{ \begin{array}{l} \mathbf{x}_0 = \bar{\mathbf{x}}_0 \\ x_{i,f} = \bar{x}_{i,f}, \quad i = 1, \dots, k \\ \lambda_{j,f} = 0, \quad j = k+1, \dots, n \\ \mathcal{H}_f = 0 \end{array} \right. \quad (7)$$

Formulation in (7) corresponds to a *Two-Point Boundary Value Problem (TPBVP)* as the optimal control \mathbf{u}^* is expressed as function of system states and costates, *i.e.* the problem primal-dual variables.

The developed set of equations is now applied to the Powered Descent Guidance problem.

B. Powered Descent Guidance problem formulation

A 3-DoF scenario is now considered; a Vertical Takeoff-Vertical Landing vehicle is accounted for such purpose and the parameters associated with propulsion system performances are chosen coherently with standard propulsion systems. Aerodynamic forces are neglected and gravity field is modelled as constant. Vehicle motion is referred to a cartesian Downrange-Crossrange-Altitude (DCA) reference frame, centered in the landing site location. Altitude is alternatively referred to as Height. Assembling state as $\mathbf{x} \doteq [\mathbf{r}^T, \mathbf{v}^T, m]^T$, dynamics results

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{v} \\ u_T \frac{T_{\max}}{m} \mathbf{i}_T + \mathbf{g} \\ -\alpha u_T \end{bmatrix} \doteq \mathbf{f}(\mathbf{x}, u_T, \mathbf{i}_T) \quad \text{where} \quad \alpha = \frac{T_{\max}}{I_{sp} g_0} \quad \text{and} \quad \mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ -g_0 \end{bmatrix} \quad (8)$$

Specific impulse $I_{sp} = 320 \text{ s}$ corresponds to a semicryogenic architecture [45], $g_0 = 9.807 \text{ m/s}^2$ is sea level gravity acceleration. Thrust vector versor \mathbf{i}_T and non-dimensional thrust magnitude u_T constitute problem control variables. Both initial and final position and velocity vectors are assigned, while mass is provided at the beginning only, as its final

value shall be optimized. Boundary conditions, where the superscript $(\cdot)^{nd}$ denotes a non-dimensional quantity, read

$$\mathbf{r}_0^{nd} = \begin{bmatrix} 0.24 \\ 0.12 \\ 1 \end{bmatrix}, \quad \mathbf{v}_0^{nd} = \begin{bmatrix} -0.66 \\ 0 \\ -1.98 \end{bmatrix}, \quad \mathbf{m}_0^{nd} = 1 \quad \text{and} \quad \mathbf{r}_f = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{v}_f = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (9)$$

and the vector $\bar{\mathbf{x}}_0^{nd} \doteq [0.24, 0.12, 1, -0.66, 0, -1.98, 1]^T$ gathers the non-dimensional initial conditions. Initial height is considered as reference length; landing time of the free-final time quadratic objective function-optimal problem, outlined in Sec. IV, is instead considered as reference time. Reference velocity results from the ratio of the two, while reference mass corresponds to initial mass. Null final velocity and displacement with respect to the landing site characterize the present landing formulation, namely a *pinpoint landing*. Initial conditions force trajectory to lie on a curved surface, as represented in Fig. 1; therefore the analysis is applied to the most general case for the presented dynamical environment. Thrust limitations are associated with lower throttling limit and upper thrust magnitude, hence

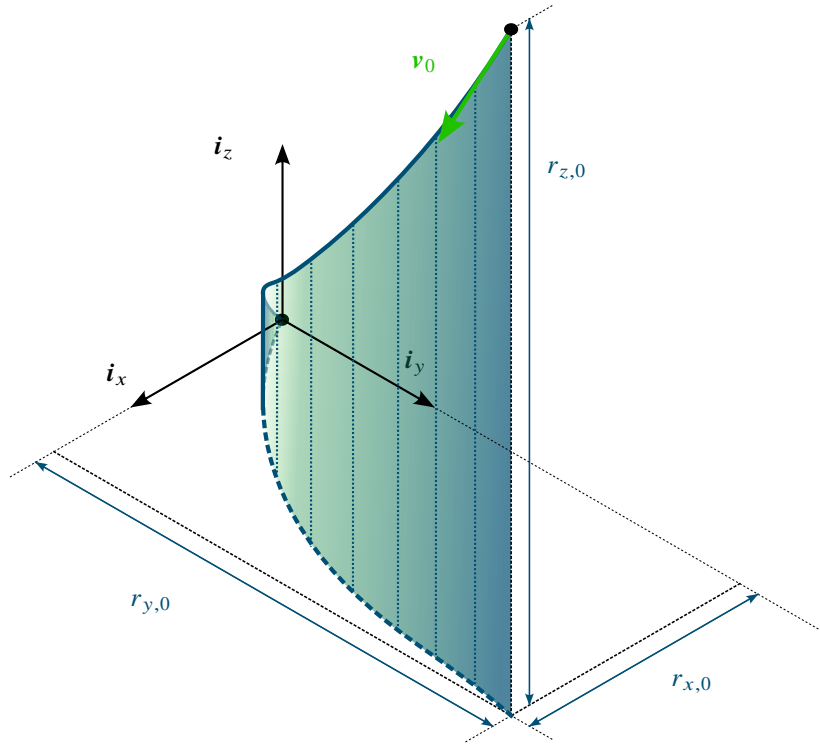


Fig. 1 Example of a trajectory lying on a curved surface

$$u_{T,\min} \leq u_T \leq u_{T,\max} \quad \text{with} \quad \begin{cases} u_{T,\min} = 0.3 \\ u_{T,\max} = 1 \end{cases} \quad (10)$$

Definition of i_T as unit vector requires

$$i_T = 1 \quad (11)$$

Usual aim of an optimal landing problem is the minimization of the propellant mass while ensuring the final landing site is correctly targeted. Moreover, the direct step of the hybrid algorithm requires an Acceleration-Optimal (AO) formulation, which is less intuitive than the Fuel Optimal (FO), but equivalent to it [6]. *Indeed, the direct step makes use of the lossless convex formulation of the 3-DoF PDG problem, which in turn requires the AO formulation.* The *cost functional* \mathcal{J} is hence coherently formulated, thus reading

$$\mathcal{J} = \int_0^{t_f} \frac{T_{\max}}{m} u_T dt \quad (12)$$

with free final time t_f , since the objective function is unimodal with respect to t_f [6].

We would like to highlight that in this analysis we did not include typical path constraints such as the glideslope limitation, constituting a pure state constraint that cannot be active over a finite interval of time as shown in literature [46], nor the thrust direction constraint, which constitutes a pure control constraint, as they do not intrinsically alter the TPBVP formulation in Eq. (7).

The Acceleration-Optimal PDG problem \mathbb{P} can be hence formalized as

$$\min_{u_T, i_T, t_f} \mathcal{J} \quad \text{s.t.} \quad \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u_T, i_T) \\ u_{T,\min} \leq u_T \leq u_{T,\max} \\ i_T = 1 \\ \text{BCs in Eqs. (9)} \end{cases} \quad (13)$$

The Hamiltonian \mathcal{H} associated with Problem (13) is hereafter defined

$$\mathcal{H} \doteq \frac{T_{\max}}{m} u_T + \lambda_r^T \mathbf{v} + \lambda_v^T \left(u_T \frac{T_{\max}}{m} i_T + \mathbf{g} \right) - \lambda_m \alpha u_T \quad (14)$$

The PMP, therefore, provides

$$i_T^*(\lambda) = -\frac{\lambda_v}{\lambda_v} \quad \text{and} \quad u_T^*(\mathbf{x}, \lambda) = \begin{cases} u_{T,\min} & \text{if } S > 0 \\ \in [u_{T,\min}, u_{T,\max}] & \text{if } S = 0 \\ u_{T,\max} & \text{if } S < 0 \end{cases} \quad (15)$$

where the relation on the left stems from Lawden's *primer vector theory* [47], while the function S depends on

primal-dual states only, and is referred to as *switching function*. Its expression, for the considered dynamics and i_T^* , reads

$$S(\mathbf{x}, \lambda) = \frac{T_{\max}}{m} (1 - \lambda_v) - \lambda_m \alpha \quad (16)$$

Since S is non-null almost everywhere [41], u_T^* is discontinuous, then referred to as *bang-bang*.

Both primal and dual dynamical constraints can be reformulated employing Eqs. (15) and (16), according to the following

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \lambda) = \begin{bmatrix} \mathbf{v} \\ -u_T^* \frac{T_{\max}}{m} \frac{\lambda_v}{\lambda_v} + \mathbf{g} \\ -\alpha u_T^* \end{bmatrix} \quad \text{and} \quad \dot{\lambda} = \mathbf{f}_\lambda(\mathbf{x}, \lambda) = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ -\lambda_r \\ \frac{u_T^* T_{\max}}{m^2} (1 - \lambda_v) \end{bmatrix} \quad (17)$$

Problem in (13) is finally dualized, and the TPBVP embedding the PMP is outlined here

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \lambda) \\ \dot{\lambda} = \mathbf{f}_\lambda(\mathbf{x}, \lambda) \end{cases} \quad \text{with BCs} \quad \begin{cases} \mathbf{x}_0^{nd} = \bar{\mathbf{x}}_0^{nd} \\ \mathbf{r}_f = \mathbf{0}_{3 \times 1} \\ \mathbf{v}_f = \mathbf{0}_{3 \times 1} \\ \lambda_{m,f} = 0 \\ \mathcal{H}_f = 0 \end{cases} \quad (18)$$

Note that, whereas the benchmark presented here is well-known, and therefore useful to understand the behavior of the proposed methodology, the use of a higher-fidelity model for the dynamics (e.g. including the rocket's rotational degrees of freedom) rapidly increases the complexity of retrieving the optimal control by using the PMP, and is therefore beyond the scope of this paper.

III. Acceleration-Optimal PDG Indirect Single Shooting

The present section deepens the analysis of the shooting method employed to solve Problem (18), along with the techniques to correctly handle the bang-bang control profile. A technique built on the legacy of [14], developed for low-thrust optimization, is applied to the vacuum PDG dynamics.

A. Single shooting technique

States \mathbf{x} and costates λ can be grouped in the vector $\mathbf{y} = [\mathbf{x}^T, \lambda^T]^T \in \mathbb{R}^{14}$. Let $\varphi_{\mathbf{y}}([\bar{\mathbf{x}}_0, \lambda_0], 0; t)$ denote the flow [48] of the initial primal-dual states according to dynamics defined in Problem (18). It results $\mathbf{y} \doteq \varphi_{\mathbf{y}}([\bar{\mathbf{x}}_0, \lambda_0], 0; t)$,

therefore $\mathbf{y} = \mathbf{y}(\mathbf{y}_0, t) = \mathbf{y}(\lambda_0, t)$ since initial states and time are fixed. The TPBVP can be recast in \mathbb{P}_I as follows

$$\text{Find } [\lambda_0^T, t_f]^T \text{ such that } \mathbf{y}(t) \text{ satisfies } \begin{cases} \mathbf{r}(\lambda_0, t_f) = \mathbf{0}_{3 \times 1} \\ \mathbf{v}(\lambda_0, t_f) = \mathbf{0}_{3 \times 1} \\ \lambda_m(\lambda_0, t_f) = 0 \\ \mathcal{H}(\lambda_0, t_f) = 0 \end{cases} \quad (19)$$

The employed approach consists of an *indirect single shooting*. First, initial costates and final time are guessed, and a high-order explicit integration scheme is used to propagate the primal-dual dynamics until final time: the initial guesses are indeed 'shot'. Secondly, the violation of terminal conditions $\Delta_f \in \mathbb{R}^8$ is evaluated, and used to correct initial guesses.

The propagation is performed using an 8th order Dormand-Prince Runge-Kutta fixed-step integrator equipped with a switching detector; such choice is justified by the nature of the optimal solution: since the dynamics are locally non-regular, the multi-step integrator can not handle the local non-continuous control profile correctly.

As far as a corrector step is considered, constraints on final conditions are strongly nonlinear, therefore an analytical first-order approximation is employed to robustly correct the initial states and time, avoiding unnecessary numerical approximations. Let the superscript $(\cdot)^{k_m}$ mark the quantity (\cdot) estimated at the generic k_m^{th} iteration of the single shooting main loop. Let the vector $\Delta_0^{k_m} \in \mathbb{R}^8$ gather the approximated deviations of initial conditions at step k_m from the solution. The following equation defines the linearized mapping from $\Delta_0^{k_m}$ to $\Delta_f^{k_m}$.

$$\Delta_f^{k_m} \doteq \begin{bmatrix} \delta \mathbf{r}_f \\ \delta \mathbf{v}_f \\ \delta \lambda_{m,f} \\ \delta \mathcal{H}_f \end{bmatrix}^{k_m} = \begin{bmatrix} \mathbf{\Phi}_{\lambda, [\mathbf{r}, \mathbf{v}, \lambda_m]} & \begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \\ \dot{\lambda}_m \end{bmatrix} \\ \nabla_{\mathbf{y}}^T \mathcal{H}_f \mathbf{\Phi}_{\lambda, \mathbf{y}} & \dot{\mathcal{H}}_f \end{bmatrix}^{k_m} \begin{bmatrix} \delta \lambda_0 \\ \delta t_f \end{bmatrix}^{k_m} \doteq \mathbb{J}^{k_m} \begin{bmatrix} \delta \lambda_0 \\ \delta t_f \end{bmatrix}^{k_m} \doteq \Delta_0^{k_m} \quad (20)$$

Block 1 Block 2

The *State Transition Matrix* (STM) $\mathbf{\Phi}$ is defined in the following paragraph. The Jacobian of the final constraints with respect to the free variables, here represented by \mathbb{J}^{k_m} , is built using two blocks.

1. Block 1 - Flow-related components

Given a reference trajectory and the initial perturbation $\delta \mathbf{y}_0$, the STM $\mathbf{\Phi}(t_0, t)$ maps $\delta \mathbf{y}(t)$ over $[0, t]$ according to $\delta \mathbf{y}(t) \doteq \mathbf{\Phi}(t_0, t) \delta \mathbf{y}_0$. The STM constitutes the first-order approximation of the perturbations' dynamics; given the n^{th} order identity matrix $\mathbb{I}_{n \times n}$, if the primal-dual states belong to $C^1([0, t_f], \mathbb{R}^{14})$, its components obey to the following

dynamics [49]

$$\begin{cases} \dot{\Phi}(t_0, t) = \mathbf{J}_y \mathbf{F}(\mathbf{y}) \Phi(t_0, t) \\ \Phi(t_0, t_0) = \mathbb{I}_{14 \times 14} \end{cases} \quad \text{where} \quad \mathbf{F}(\mathbf{y}) \doteq \begin{bmatrix} \mathbf{f}(\mathbf{y}) \\ \mathbf{f}_\lambda(\mathbf{y}) \end{bmatrix} \quad (21)$$

Block 1 of \mathbb{J}^{k_m} employs $\Phi_{\lambda, [r, v, \lambda_m]}$ for the first upper sub-block, namely the components of Φ mapping perturbations of costates to perturbations of position, velocity and comass λ_m . The lower sub-block, chains 1) $\nabla_y \mathcal{H}_f$ which identifies the dependency of final \mathcal{H} with respect to the augmented states and 2) $\Phi_{\lambda, y}$, that maps perturbations on costates to perturbations on final augmented state.

For the Acceleration-Optimal formulation, the primal-dual dynamics Jacobian is provided by

$$\mathbf{J}_y \mathbf{F} = \left[\begin{array}{ccc|ccc} \mathbf{0}_{3 \times 3} & \mathbb{I}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \Omega_{v, m} & \mathbf{0}_{3 \times 3} & \Omega_{v, \lambda_v} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 \\ \hline \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & -\mathbb{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \Omega_{\lambda_m, m} & \mathbf{0}_{1 \times 3} & \Omega_{\lambda_m, \lambda_v} & 0 \end{array} \right], \quad \begin{cases} \Omega_{v, m} = \frac{T_{\max} u_T^* \lambda_v}{m^2 \lambda_v} \\ \Omega_{v, \lambda_v} = \frac{T_{\max} u_T^*}{m} \left(-\frac{\mathbb{I}_{3 \times 3}}{\lambda_v} + \frac{\lambda_v \lambda_v^T}{\lambda_v^3} \right) \\ \Omega_{\lambda_m, m} = -2 \frac{T_{\max} u_T^*}{m^3} (1 - \lambda_v) \\ \Omega_{\lambda_m, \lambda_v} = -\frac{T_{\max} u_T^* \lambda_v^T}{m^2 \lambda_v} \end{cases} \quad (22)$$

2. Block 2 - Final time-related components

Block 2 components are trivial: variation of the time horizon leads, in the first-order approximation, to a variation in final conditions proportional to the total time derivatives. In such fashion, the variation of the Hamiltonian is again composed, as $\mathcal{H}_f = \mathcal{H}(\mathbf{y}_f)$; therefore $\dot{\mathcal{H}}_f = \nabla_y^T \mathcal{H}_f \dot{\mathbf{y}}$.

The corrector exploits the constraints' Jacobian to correct the initial condition estimation, using therefore a Newton-Raphson method to solve iteratively Problem (19). The solution at the generic k_m^{th} step solves the system

$$\mathbb{J}^{k_m} \begin{bmatrix} \lambda_0 \\ t_f \end{bmatrix}^{k_m} = \mathbb{J}^{k_m} \begin{bmatrix} \lambda_0 \\ t_f \end{bmatrix}^{k_m-1} - \xi \Delta_f^{k_m} \quad (23)$$

where the fixed scalar $\xi \in [0, 1]$ is tuned to define the radius of the trust region. The loop is stopped when $\left\| \Delta_0^{k_m} \right\|_\infty < \text{tol}$, being tol fixed to $\text{tol} = 10^{-5}$. **The correction step reported in Eq. (23) and used across this work allows to include the free-final-time contributions within the Fuel-Optimal-equivalent problem formulation, expanding the original Fuel Optimal formulation in [14].**

Remark 2: Given $z \doteq [y^T, \text{vec}(\Phi)^T]^T$ where vec transforms a matrix into a column-wise vector, the augmented state dynamics shall be further augmented propagating STM according to

$$\dot{z} = \begin{bmatrix} \dot{y} \\ \text{vec}(\dot{\Phi}) \end{bmatrix} = \begin{bmatrix} F(y) \\ \text{vec}(\mathbf{J}_y F \Phi) \end{bmatrix} \doteq \mathcal{F}(z) \quad (24)$$

Propagation of the STM is possible only numerically, and an analytical form is used for $\mathbf{J}_y F$. While the STM is required for the corrector step, number of its components grows quadratically with the number of states. For our application, $z \in \mathbb{R}^{210}$, since $\text{vec}(\Phi) \in \mathbb{R}^{196}$; using an implicit propagator results in demanding computational time requirements.

Remark 3: STM dynamics in Eq. (21) is valid within thrusting arcs. At the generic switching time t_{sw} , instead, Jacobian of primal-dual dynamics is discontinuous. Jacobian $\Psi(t_{\text{sw}}) \doteq \mathbf{J}_{y(t_{\text{sw}}^-)} y(t_{\text{sw}}^+)$ handles the discontinuity according to

$$\Phi(0, t) = \Phi(t_{\text{sw}, t}^+) \Psi(t_{\text{sw}}) \Phi(0, t_{\text{sw}}^-) \quad (25)$$

and is computed with the following relation [49]

$$\Psi(t_{\text{sw}}) = \mathbb{I}_{14 \times 14} + [\dot{y}(t_{\text{sw}}^+) - \dot{y}(t_{\text{sw}}^-)] \frac{\nabla_y^T S(t_{\text{sw}})}{\dot{S}(t_{\text{sw}})} \quad (26)$$

Equation (25) may be applied multiple times in case multiple switches are present over the time domain; \dot{S} is instead computed in the same fashion as $\dot{\mathcal{H}}$, i.e. $\dot{S} \doteq \nabla_y^T S \dot{y}$.

B. Switching detection approach

The described technique requires a switching detection routine. The reason for such claim is twofold: 1) correction accuracy benefits from accurately computing the STM and, consequently, from accurately estimating the switching time; 2) propagation accuracy of states and costates benefits from accurate evaluation of the switching time: states and costates do not belong to C^∞ with respect to time at switching points, due to discontinuities in the optimal control profile. Therefore, as previously mentioned, non-stiff single-step and multi-step integrators can not handle such discontinuities; rather than using a stiff integrator, in this work we employ the switching detection to place an integration node exactly at the switching point, hence allowing a non-stiff single step to integrate the primal-dual dynamics minimizing the integration error across the control switch.

During the integration, the sign of the switching function is monitored: according to Eq. (15), S changes sign whenever u_T^* switches. If a sign change of S is detected during the generic step $[t_1, t_2]$, a Newton-Raphson zero-finding algorithm looks for the zero of S in the interval $[t_1, t_2]$; at the generic k_s^{th} iteration of the switching time detection loop,

therefore, it simply reads

$$t_{sw}^{k_s} = t_{sw}^{k_s-1} - \frac{S(t_{sw}^{k_s-1})}{\dot{S}(t_{sw}^{k_s-1})} \quad (27)$$

and the algorithm is considered converged when $\Delta^{k_s} t_{sw} \doteq t_{sw}^{k_s} - t_{sw}^{k_s-1} < \text{tol}$. tol in this case equals 10 eps , where eps is the machine epsilon.

The overall integration workflow is explained in Algorithm 1. The superscript k indicates the generic step of the integration loop, $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(0)}$ denote respectively the dynamics corresponding to maximum control and to idle control. In Algorithm 1, z_0 is built using the known initial state x_0 , the assigned STM $\Phi(t_0, t_0)$. Initialization of λ_0 is instead complex, due to the ill-conditioning of the primal-dual dynamics. The two guess-generation techniques to counteract such problematic are described in the following two sections. The former exploits an indirect homotopy, and is based on state-of-the-art homotopy techniques well employed both in rocket descent [17] and low-thrust optimization [14]; the latter exploits the convex direct step and the Covector Mapping Theorem, thus providing the novel hybrid method.

Algorithm 1 Integration workflow with external switching detection technique

```

 $z^k \leftarrow z_0 \quad S^k \leftarrow S(z^k)$ 
while  $t^k < t_f$  do
  if  $S^k > 0$  then  $\mathcal{F}^k \leftarrow \mathcal{F}^{(0)}$ ; else if  $S^k < 0$  then  $\mathcal{F}^k \leftarrow \mathcal{F}^{(1)}$ ; end if
   $z^{k+1} \leftarrow \varphi_z^k(t^k, t^{k+1})$   $\triangleright \varphi_z^k(t^k, t^{k+1}) \doteq z(z^k, t^k; t^{k+1})$  is  $z$  flow from  $t^k$  through  $\mathcal{F}^k$ 
   $S^{k+1} \leftarrow S(y^{k+1})$   $\triangleright t^{k+1}$  is arbitrarily chosen as  $t^k + 0.1 \text{ s}$ 
  if  $\text{sign}(S^{k+1}) = \text{sign}(S^k)$  then
     $t^k \leftarrow t^{k+1} \quad z^k \leftarrow z^{k+1}$ 
  else if  $\text{sign}(S^{k+1}) \neq \text{sign}(S^k)$  then
    find  $t_{sw}$  solving  $S = 0$ ,  $t_{sw}^{\text{guess}} = t^k$   $\triangleright$  See Sec. III.B for details
     $[y^{k+1}, \text{vec}(\Phi_-^{k+1})]^T \leftarrow \varphi_z^k(t^k, t_{sw})$ 
     $\Phi^{k+1} = \Psi(t_{sw}) \Phi_-^{k+1}$   $\triangleright$  See Eq. (26) for  $\Psi$  evaluation
     $t^k \leftarrow t_{sw} \quad z^k \leftarrow [y^{k+1}, \text{vec}(\Phi^{k+1})]^T$  Impose  $\mathcal{F}^k$  depending on the detected switch type
  end if
end while

```

IV. Indirect homotopy-based guess generation

The present section presents the generic homotopic subproblem, then outlines the continuation scheme employed for the fully-indirect algorithm.

A. Quadratic PDG Problem and shooting formulation updates

Let the quantity $(\cdot)_h$ denote the generic variable (\cdot) associated with the generalized quadratic formulation here presented.

The machinery developed in the previous sections needs to be partially reformulated to increase problem convergence basin. Indeed, a generalized cost functional is introduced: setting the *homotopy parameter* $\varepsilon \neq 1$ in the homotopic

cost functional in Eq. (28), convergence robustness of the PDG dual formulation increases with respect to the Acceleration-Optimal case.

$$\mathcal{J}_h = \int_0^{t_f} \left[(1 - \varepsilon) \left(\frac{T_{\max}}{m} u_T \right)^2 + \varepsilon \frac{T_{\max}}{m} u_T \right] dt \quad \text{where } \varepsilon \in [0, 1] \quad (28)$$

For $\varepsilon = 0$ the PDG problem is purely quadratic, and is therefore referred to as *Energy Optimal* (EO); for $\varepsilon = 1$ the original Acceleration-Optimal formulation is retrieved, hence the lower convergence performances.

The Hamiltonian is modified accordingly, and, if no thrust limitations are accounted, the problem features finite optimal thrust magnitude u_T^{th} . Both read as follows.

$$\begin{aligned} \mathcal{H}_h &= (1 - \varepsilon) \left(\frac{T_{\max}}{m} u_T \right)^2 + \varepsilon \frac{T_{\max}}{m} u_T + \lambda_r^T \mathbf{v} + \lambda_v^T \left(u_T \frac{T_{\max}}{m} \mathbf{i}_T + \mathbf{g} \right) - \lambda_m \alpha u_T \\ u_T^{th}(\mathbf{x}, \lambda, \varepsilon) &= \frac{1}{2(1 - \varepsilon)} \left[\frac{m^2}{T_{\max}^2} \lambda_m \alpha + \frac{m}{T_{\max}} (\lambda_v - \varepsilon) \right] \end{aligned} \quad (29)$$

hence the application of the PMP provides Eq. (30); it constrains u_T in Eq. (29), but leaves unaltered the optimal thrust direction outlined in Eq. (15).

$$u_{T,h}^*(\mathbf{x}, \lambda, \varepsilon) = \begin{cases} u_{T,\min} & \text{if } u_T^{th} < u_{T,\min} \\ u_T^{th} & \text{if } u_T^{th} \in [u_{T,\min}, u_{T,\max}] \\ u_{T,\max} & \text{if } u_T^{th} > u_{T,\max} \end{cases} \quad (30)$$

The corresponding TPBVP reads

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}_h(\mathbf{x}, \lambda, \varepsilon) \\ \dot{\lambda} = \mathbf{f}_{\lambda,h}(\mathbf{x}, \lambda, \varepsilon) \end{cases} \quad \text{with BCs} \quad \begin{cases} \mathbf{x}_0^{nd} = \bar{\mathbf{x}}_0^{nd} \\ \mathbf{r}_f = \mathbf{0}_{3 \times 1} \\ \mathbf{v}_f = \mathbf{0}_{3 \times 1} \\ \lambda_{m,f} = 0 \\ \mathcal{H}_{h,f} = 0 \end{cases} \quad (31)$$

While \mathbf{f}_h differs from \mathbf{f} for the optimal thrust magnitude program only, $\mathbf{f}_{\lambda,h}$ also features a different functional form

with respect to f_λ .

$$f_{\lambda,h}(\mathbf{x}, \lambda, \varepsilon) = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ -\lambda_r \\ \frac{u_{T,h}^* T_{\max}}{m^2} (\varepsilon - \lambda_v) + 2 \frac{(u_{T,h}^* T_{\max})^2}{m^3} (1 - \varepsilon) \end{bmatrix} \quad (32)$$

Let us refer to the three thrust levels outlined in Tab. 1.

Table 1 Thrust levels for generalized \mathcal{J}_h

Thrust level	$u_{T,h}^*$
(1)	u_{\min}
(2)	u_T^{th}
(3)	u_{\max}

With respect to the Acceleration Optimal, the generalized formulation features arcs propelled by non-constant thrust: u_T^{th} changes continuously and the interval $[u_{t,\min}, u_{t,\max}]$ has non-null duration, hence constant thrust arcs are patched by non-constant thrust ones, namely arcs of thrust level (2).

The generalized *single shooting problem* $\mathbb{P}_{1,h}$ is reformulated as follows.

$$\text{Find } [\lambda_0^T, t_f]^T \text{ such that } \mathbf{y}_h(t) \text{ satisfies } \begin{cases} \mathbf{r}(\lambda_0, t_f) = \mathbf{0}_{3 \times 1} \\ \mathbf{v}(\lambda_0, t_f) = \mathbf{0}_{3 \times 1} \\ \lambda_m(\lambda_0, t_f) = 0 \\ \mathcal{H}_h(\lambda_0, t_f) = 0 \end{cases} \quad (33)$$

The correction step machinery is not modified, and the related Jacobian \mathbb{J}_h features the updated dynamics RHS and Hamiltonian.

The propagation step, instead, requires Jacobians with different functional forms depending on the thrust level.

Specifically, for thrust levels (1) and (3), the Jacobian of dynamics reads

$$\mathbf{J}_y^{(1,3)} \mathbf{F}_h = \left[\begin{array}{ccc|ccc} \mathbf{0}_{3 \times 3} & \mathbb{I}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \boldsymbol{\Omega}_{v,m}^{(1,3)} & \mathbf{0}_{3 \times 3} & \boldsymbol{\Omega}_{v,\lambda_v}^{(1,3)} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 \\ \hline \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & -\mathbb{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \boldsymbol{\Omega}_{\lambda_m,m}^{(1,3)} & \mathbf{0}_{1 \times 3} & \boldsymbol{\Omega}_{\lambda_m,\lambda_v}^{(1,3)} & 0 \end{array} \right], \quad \left\{ \begin{array}{l} \boldsymbol{\Omega}_{v,m}^{(1,3)} = \frac{T_{\max} u_{T,h}^* \lambda_v}{m^2} \\ \boldsymbol{\Omega}_{v,\lambda_v}^{(1,3)} = \frac{T_{\max} u_{T,h}^*}{m} \left(-\frac{\mathbb{I}_{3 \times 3}}{\lambda_v} + \frac{\lambda_v \lambda_v^T}{\lambda_v^3} \right) \\ \boldsymbol{\Omega}_{\lambda_m,m}^{(1,3)} = -2 \frac{u_{T,h}^* T_{\max}}{m^3} (\varepsilon - \lambda_v) + \\ \quad -6 \frac{(u_{T,h}^* T_{\max})^2}{m^4} (1 - \varepsilon) \\ \boldsymbol{\Omega}_{\lambda_m,\lambda_v}^{(1,3)} = -\frac{T_{\max} u_{T,h}^* \lambda_v^T}{m^2} \end{array} \right. \quad (34)$$

For thrust level (2), instead, the non-null gradient components of $u_{T,h}^* = u_T^{th}$ shall be taken into account; therefore the Jacobian corresponding to thrust level (2) is provided by the following equations

$$\mathbf{J}_y^{(2)} \mathbf{F}_h = \left[\begin{array}{ccc|ccc} \mathbf{0}_{3 \times 3} & \mathbb{I}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \boldsymbol{\Omega}_{v,m}^{(2)} & \mathbf{0}_{3 \times 3} & \boldsymbol{\Omega}_{v,\lambda_v}^{(2)} & \boldsymbol{\Omega}_{v,\lambda_m}^{(2)} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \boldsymbol{\Omega}_{m,m}^{(2)} & \mathbf{0}_{1 \times 3} & \boldsymbol{\Omega}_{m,\lambda_v}^{(2)} & \boldsymbol{\Omega}_{m,\lambda_m}^{(2)} \\ \hline \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & -\mathbb{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \boldsymbol{\Omega}_{\lambda_m,m}^{(2)} & \mathbf{0}_{1 \times 3} & \boldsymbol{\Omega}_{\lambda_m,\lambda_v}^{(2)} & \boldsymbol{\Omega}_{\lambda_m,\lambda_m}^{(2)} \end{array} \right], \quad \left\{ \begin{array}{l} \frac{\partial u_{T,h}^*}{\partial m} = \frac{1}{2(1-\varepsilon)} \left[2 \frac{m}{T_{\max}^2} \lambda_m \alpha + \frac{1}{T_{\max}} (\lambda_v - \varepsilon) \right] \\ \nabla_{\lambda_v} u_{T,h}^* = \frac{1}{2(1-\varepsilon)} \frac{m}{T_{\max}} \frac{\lambda_v}{\lambda_v} \\ \frac{\partial u_{T,h}^*}{\partial \lambda_m} = \frac{1}{2(1-\varepsilon)} \frac{m^2}{T_{\max}^2} \alpha \end{array} \right. \quad (35)$$

$$\left\{ \begin{array}{lll} \frac{\partial \dot{v}}{\partial u_{T,h}^*} = -\frac{T_{\max}}{m} \frac{\lambda_v}{\lambda_v} & \frac{\partial \dot{\lambda}_m}{\partial u_{T,h}^*} = \frac{T_{\max}}{m^2} (\varepsilon - \lambda_v) + 4 \frac{u_{T,h}^* T_{\max}^2}{m^3} (1 - \varepsilon) & \\ \boldsymbol{\Omega}_{v,m}^{(2)} = \boldsymbol{\Omega}_{v,m}^{(1,3)} + \frac{\partial \dot{v}}{\partial u_{T,h}^*} \frac{\partial u_{T,h}^*}{\partial m} & \boldsymbol{\Omega}_{v,\lambda_v}^{(2)} = \boldsymbol{\Omega}_{v,\lambda_v}^{(1,3)} + \frac{\partial \dot{v}}{\partial u_{T,h}^*} \nabla_{\lambda_v}^T u_{T,h}^* & \boldsymbol{\Omega}_{v,\lambda_m}^{(2)} = \frac{\partial \dot{v}}{\partial u_{T,h}^*} \frac{\partial u_{T,h}^*}{\partial \lambda_m} \\ \boldsymbol{\Omega}_{m,m}^{(2)} = -\alpha \frac{\partial u_{T,h}^*}{\partial m} & \boldsymbol{\Omega}_{m,\lambda_v}^{(2)} = -\alpha \nabla_{\lambda_v}^T u_{T,h}^* & \boldsymbol{\Omega}_{m,\lambda_m}^{(2)} = -\alpha \frac{\partial u_{T,h}^*}{\partial \lambda_m} \\ \boldsymbol{\Omega}_{\lambda_m,m}^{(2)} = \boldsymbol{\Omega}_{\lambda_m,m}^{(1,3)} + \frac{\partial \dot{\lambda}_m}{\partial u_{T,h}^*} \frac{\partial u_{T,h}^*}{\partial m} & \boldsymbol{\Omega}_{\lambda_m,\lambda_v}^{(2)} = \boldsymbol{\Omega}_{\lambda_m,\lambda_v}^{(1,3)} + \frac{\partial \dot{\lambda}_m}{\partial u_{T,h}^*} \nabla_{\lambda_v}^T u_{T,h}^* & \boldsymbol{\Omega}_{\lambda_m,\lambda_m}^{(2)} = \frac{\partial \dot{\lambda}_m}{\partial u_{T,h}^*} \frac{\partial u_{T,h}^*}{\partial \lambda_m} \end{array} \right.$$

Remark 4: The presented formulas are valid as long as $\varepsilon \neq 1$. The formulation corresponding to such value has however already been treated in Sec. III, and it substitutes the quadratic PDG when $\varepsilon = 1$. Moreover, switching times are substituted by saturation times; here the thrust profile is non-smooth, therefore a saturation detection is employed, and the value of u_T^{th} is monitored; when it steps across a saturation value $u_{T,\text{sat}}$ (may be either $u_{T,\min}$ or $u_{T,\max}$), a zero-finding algorithm finds t_{sat} such that $u_T^{th}(t_{\text{sat}}) - u_{T,\text{sat}} = 0$. A Newton-Raphson method similar to the one described

in Sec. III.B is employed for such purpose. However, since thrust magnitude is continuous, no Jacobian dynamics discontinuity is present, hence no STM jump Ψ is added.

The propagation algorithm with saturation detection is summarized in Algorithm 2. The operator lev extracts the thrust level from the theoretical optimal thrust magnitude, according to Eq. (30) and Tab. 1. The superscript of \mathcal{F}_h indicates the thrust level the augmented RHS corresponds to.

Algorithm 2 Integration workflow with external saturation detection technique

```

 $z^k \leftarrow z_0 \quad u_T^{th,k} \leftarrow u_T^{th}(z^k)$ 
while  $t^k < t_f$  do
  if  $u_T^{th,k} < u_{T,\min}$  then  $\mathcal{F}_h^k \leftarrow \mathcal{F}_h^{(1)}$ ; else if  $u_T^{th,k} > u_{T,\max}$  then  $\mathcal{F}_h^k \leftarrow \mathcal{F}_h^{(3)}$ ; else  $\mathcal{F}_h^k \leftarrow \mathcal{F}_h^{(2)}$ ; end if
   $z^{k+1} \leftarrow \varphi_{h,z}^k(t^k, t^{k+1})$   $\triangleright \varphi_{h,z}^k(t^k, t^{k+1}) \doteq z(z^k, t^k; t^{k+1})$  is  $z$  flow from  $t^k$  through  $\mathcal{F}_h^k$ 
   $u_T^{th,k+1} \leftarrow u_T^{th}(y^{k+1})$   $\triangleright t^{k+1}$  is arbitrarily chosen as  $t^k + 0.1 s$ 
  if  $\text{lev}(u_T^{th,k+1}) = \text{lev}(u_T^{th,k})$  then
     $t^k \leftarrow t^{k+1} \quad z^k \leftarrow z^{k+1}$ 
  else if  $\text{lev}(u_T^{th,k+1}) \neq \text{lev}(u_T^{th,k})$  then
    find  $t_{\text{sat}}$  solving  $u_T^{th} - u_{T,\text{sat}} = 0$ ,  $t_{\text{sat}}^{\text{guess}} = t^k$ 
     $[y^{k+1}, \text{vec}(\Phi_-^{k+1})]^T \leftarrow \varphi_{h,z}^k(t^k, t_{\text{sw}})$ 
     $t^k \leftarrow t_{\text{sw}} \quad z^k \leftarrow [y^{k+1}, \text{vec}(\Phi_-^{k+1})]^T$  Impose  $\mathcal{F}_h^k$  depending on the detected switch type
  end if
end while

```

Solution and convergence properties of Problem in (31) depend on the specific functional form of \mathcal{F}_h , which in turn depends on the homotopic parameter ε . The single shooting converges in few steps for $\varepsilon = 0$, given arbitrary initial conditions for both λ_0 and t_f . The purely quadratic problem is therefore used as first subproblem for the indirect homotopy, described in the following subsection.

B. Homotopy-based purely indirect algorithm

The idea underlying the homotopy is straightforward: the solution of the k_h^{th} subproblem can be successfully used as guess for the $(k_h + 1)^{\text{th}}$, as long as the two problems are sufficiently similar. In the present case, the closer ε^{k_h} to ε^{k_h+1} , the more similar the two subproblems. As suggested in literature [50], the sequence $[\varepsilon^1, \dots, \varepsilon^{n+1}]$ is taken as equally spaced on a logarithmic scale, thus guaranteeing successful continuation for sufficiently small distance between ε^{k_h} and ε^{k_h+1} . A correctly designed homotopy converges to the original problem, therefore it shall be $\varepsilon^{n+1} \doteq 1$; in addition, for the suggested convergence properties of the PDG problem for $\varepsilon = 0$, it is chosen $\varepsilon^1 \doteq 0$.

The full indirect algorithm is schematized in the flowchart in Fig. 2. Notice the notation $\mathbb{P}_{I,h}(\varepsilon)$, as the problem $\mathbb{P}_{I,h}$ solved at each iteration changes with ε .

The presented homotopic scheme is capable of efficiently handling the reduced basin of convergence extension of the Fuel-Optimal problem. Yet, as exposed later in the results Sec. VI, the number of required homotopic steps is

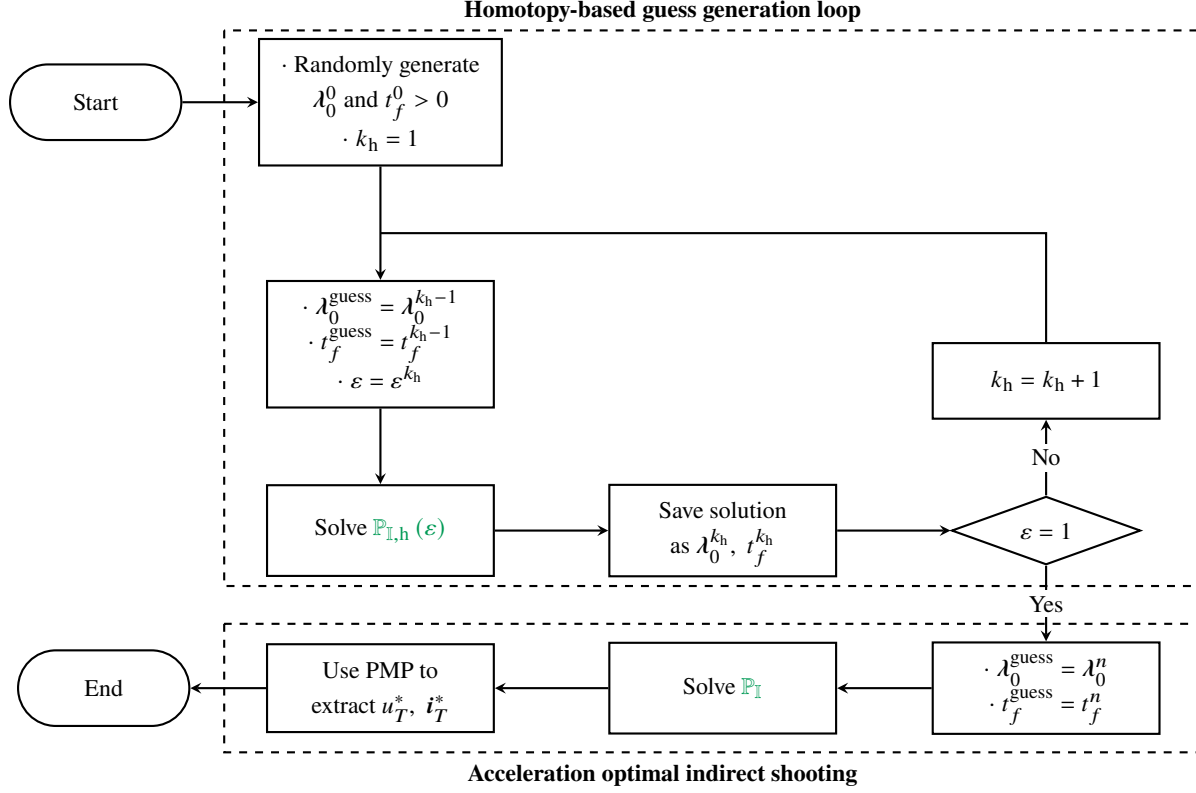


Fig. 2 Fully indirect homotopy-based algorithm overview

relevant; as a consequence, computational times of the guess generation mechanism widely overcome the time required to solve \mathbb{P}_I .

The hybrid strategy discussed in the following section solves this issue.

V. Convex pseudospectral-based guess generation

The present section starts by presenting in Sec. V.A the Second-Order Cone Programming (SOCP) problems, and the employed collocation scheme with relative Covector Mapping Theorem in Sec. V.B. It then develops the lossless convex version of the Fuel-Optimal problem and the successive convexifications are applied to the transcribed version of the problem to treat the free final time in Sec. V.C. The hybrid algorithm is completed in Sec. V.D and its schematic overview provided in Fig. 4.

A. Second-Order Cone Programming

Second-Order Cone Programming problems make up a specific subclass of convex problems, and have recently grown in popularity in trajectory optimization*; indeed, they offer a feasible formulation for OCPs from different dynamical environments [6, 26, 30, 51, 52].

*Notation employed in this subsection partially clashes with the rest of the paper, and is limited to the summary of SOCP problems.

Let \mathbf{x} momentarily denote a vector of parameters $\mathbf{x} \in \mathbb{R}^n$. Let $\boldsymbol{\varphi} \in \mathbb{R}^n$ be the cost function coefficients vector; in addition, $\mathbf{A}_i \in \mathbb{R}^{n_i \times n}$, $\mathbf{b}_i \in \mathbb{R}^{n_i}$, $\mathbf{c}_i \in \mathbb{R}^n$, $\mathbf{d}_i \in \mathbb{R}$, with $i = 1, \dots, m$, are associated with quadratic inequality constraints, while $\mathbf{p} \in \mathbb{R}^p$, $\mathbf{P} \in \mathbb{R}^{p \times n}$ and $\mathbf{q} \in \mathbb{R}^q$, $\mathbf{Q} \in \mathbb{R}^{q \times n}$ are respectively related to linear inequality and linear equality constraints[†]. The general SOCP problem results then

$$\min_{\mathbf{x} \in \mathbb{R}^n} \boldsymbol{\varphi}^T \mathbf{x} \quad \text{s.t.} \quad \begin{cases} \|\mathbf{A}_i \mathbf{x} + \mathbf{b}_i\|_2 \leq \mathbf{c}_i^T \mathbf{x} + d_i & i = 1, \dots, m \\ \mathbf{P} \mathbf{x} \leq \mathbf{p} \\ \mathbf{Q} \mathbf{x} = \mathbf{q} \end{cases} \quad (36)$$

where the symbol \leq defines the component-wise inequality. The inequality $\|\mathbf{A} \mathbf{x} + \mathbf{b}\|_2 \leq \mathbf{c}^T \mathbf{x} + d$, with $\mathbf{A} \in \mathbb{R}^{r \times n}$, defines a generic *second-order cone constraint*: it indeed defines a second-order cone in \mathbb{R}^{r+1} for the variables \mathbf{y} , t obtained with the affine transformation $\mathbf{y} = \mathbf{A} \mathbf{x} + \mathbf{b}$, $t = \mathbf{c}^T \mathbf{x} + d$. Fig. 3 provides a simplified representation of this concept: the region of admissible couples $[\mathbf{y}, t]$, with $\mathbf{y} \in \mathbb{R}^2$, is identified by the blue region, a cone in \mathbb{R}^3 .

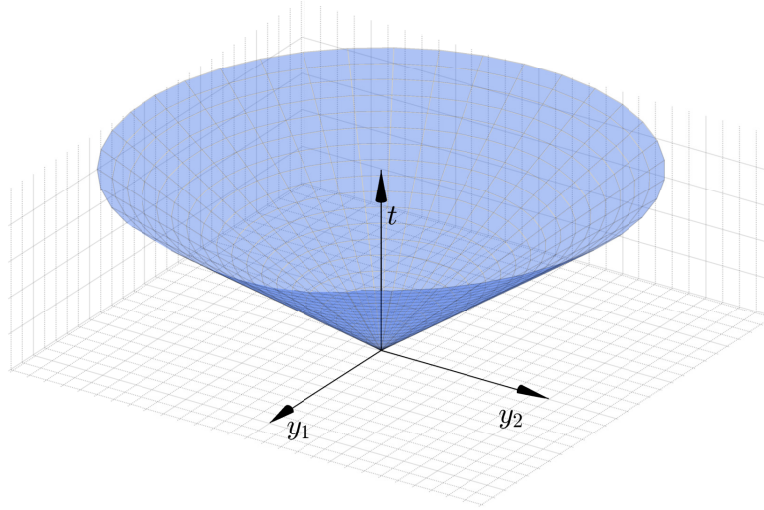


Fig. 3 Second order cone constraint in \mathbb{R}^3 : $\|\mathbf{y}\|_2 \leq t$

Ideally, one could aim at transforming the problem via a *lossless convexification*, relaxing constraints or dynamics in a convex equivalent and obtaining a different problem featuring the same optimal solution as the original one. However, such result is not always possible; a more general approach, based on *successive convexifications*, helps in this context: non-convex functions are convexified, and a *convexify and optimize* approach is iteratively applied until solution converges to the desired one within prescribed tolerance. Our approach combines lossless convex terms and

[†]Linear inequality constraints actually define cones on orthants in n -dimensional spaces, thus can be seen as subsets of second-order cone constraints; nonetheless, pure linear constraints are treated by solvers with dedicated approaches [53], thus linear inequalities are explicitly defined.

successive convexifications: the lossless convexification is applied to the continuous-time version of the problem, while the successive convexifications are applied to the transcribed equivalent.

B. *hp* Radau Pseudospectral Transcription scheme

A Legendre-Gauss-Radau (LGR) scheme is employed in this work: Lagrange polynomials approximate states and controls, and are based on 1) the roots of the Legendre-Radau polynomials and 2) the right limit of the pseudospectral time used for the integration.

For a global collocation approach, we introduce the pseudotime $\tau \in [-1, 1]$, mapped from the physical time according to the following affine transformation

$$\tau = \frac{2}{t_f - t_0}t - \frac{t_f + t_0}{t_f - t_0} \quad t \in [t_0, t_f] \quad (37)$$

the Legendre-Radau polynomial R_n is built as

$$R_n(\tau) = \tilde{L}_n(\tau) + \tilde{L}_{n-1}(\tau) \quad (38)$$

where \tilde{L}_n is the n^{th} order Legendre polynomial. Let τ_k be the k^{th} root of the polynomial in Eq. (38), and t_k the corresponding physical time. States $\mathbf{x} \in \mathbb{R}^{n_s}$ are approximated by a summation of $n + 1$ Lagrange polynomials P of degree n according to

$$\mathbf{x}(t) \approx \sum_{i=0}^n X_i P_i(t) \quad P_i(t) = \prod_{\substack{k=0 \\ k \neq i}}^n \frac{t - t_k}{t_i - t_k} \quad (39)$$

Time domain is indeed discretized in $n + 1$ nodes. In Eq. (39), $X_i = \mathbf{x}(t_i)$ since Lagrange polynomials satisfy the *isolation property*. The *pseudospectral differential operator* \mathbf{D} is obtained by differentiating with respect to τ , and i^{th} column component relative to time t_k is provided by

$$\mathbf{D}_{k,i} = \frac{dP_i}{d\tau}(\tau_k) \mathbb{I}_{n_s \times n_s} = \frac{t_f - t_0}{2} \dot{P}_i(t_k) \mathbb{I}_{n_s \times n_s} \quad (40)$$

Rescaling the differential operator from pseudotime to time, continuous-time dynamical constraints are approximated by

$$\frac{2}{t_f - t_0} \mathbf{D}_i \mathbf{X} - \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i) = \mathbf{0} \quad i = 0, \dots, n-1 \quad (41)$$

Remark 5: Dynamical constraints are reinforced at collocation points only: as far as LGR collocation is concerned, $\tau = 1$ is not a root of $R_n(\tau)$, thus dynamics is not reinforced at the corresponding physical instant, namely $t_f = t_n$. Nonetheless, Lagrange base polynomials include the polynomial associated with final time: this 1) allows for reinforcing

the boundary conditions at both time bounds, and 2) make the differentiation operator \mathbf{D} rectangular, with dimensions $n \times (n + 1)$.

Using local collocation guarantees higher convergence rate for non-smooth solution [23] and alleviates computational loads with respect to global collocation due to an increased sparsity of the matrix associated with the dynamics. Local collocation can be developed by dividing the whole domain in segments, and employing over each segment the procedure explained for the global collocation. **The convex collocation scheme employed in this work, based on work in [24], is therefore referred to as *hp* pseudospectral.**

If time domain is subdivided in \bar{n} segments, $[t_0^j, t_p^j]$ denotes the i^{th} segment, each discretized in $p + 1$ nodes. Then

$$\begin{array}{cc} X_i^j & U_i^j \\ & j = 1, \dots, \bar{n} \\ & i = 0, \dots, p \end{array} \quad (42)$$

identify the states and controls at the i^{th} node of the j^{th} segment, respectively. The integral of the path cost features the multiple segments division, as

$$\int_0^{t_f} \mathcal{L}(\mathbf{x}, \mathbf{u}, t) dt \approx \frac{t_p^{\bar{n}} - t_0^1}{2\bar{n}} \sum_{j=1}^{\bar{n}} \sum_{i=0}^p w_i \mathcal{L}(X_i^j, U_i^j, t_i^j) \quad \text{where} \quad w_i = \begin{cases} \frac{2}{n+1^2} & i = 0 \\ \frac{(1 - \tau_i)}{n^2 \tilde{L}_n(\tau_i)^2} & i = 1, \dots, n-1 \end{cases} \quad (43)$$

thus providing, if substituted in Eq. (4), the *discrete cost function* for the *hp* collocation $\mathcal{J}_d^{hp} = \mathcal{J}_d^{hp}(\mathbf{X}^{hp}, \mathbf{U}^{hp}, t^{hp})$.

The PS NLP problem can be then written as

$$\min_{\mathbf{U}^{hp} \in \mathbb{R}^{(p+1) \times n}} \mathcal{J}_d^{hp}(\mathbf{X}^{hp}, \mathbf{U}^{hp}, t^{hp}) \quad \text{s.t.} \quad \left\{ \begin{array}{l} \frac{2}{t_p^j - t_0^j} \mathbf{D}_{i+1}^j X_i^j - f(X_i^j, U_i^j) = \mathbf{0} \\ g_u(U_i^j, t_i^j) \leq 0 \\ \text{Boundary conditions in Eq. (5)} \\ \left. \begin{array}{l} t_p^{j-1} = t_0^j \\ X_p^{j-1} = X_0^j \end{array} \right\} \end{array} \right. \quad \begin{array}{l} j = 1, \dots, \bar{n} \\ i = 0, \dots, p \\ \\ j = 2, \dots, \bar{n} \end{array} \quad (44)$$

where the last two constraints make up for the *linking conditions*. In addition, for the same number of nodes p in each segment, the matrix \mathbf{D} is the same for all the segments; therefore

$$\mathbf{D}^j = \mathbf{D} \quad (45)$$

holds, and the index j can be dropped.

The CMT for the pseudospectral scheme can be easily demonstrated for a single segment formulation, and is valid for the hp pseudospectral scheme as well.

In the one segment case, the *Lagrangian* $\tilde{\mathcal{L}}$ of the NLP reads

$$\tilde{\mathcal{L}} = \frac{t_f - t_0}{2} \sum_{i=0}^{n-1} \left[w_i \mathcal{L}(X_i, U_i, t_i) + M_i g_u(U_i, t_i) \right] - \sum_{i=0}^{n-1} \Lambda_i^T \left[D_i X - \frac{t_f - t_0}{2} f(X_i, U_i) \right] \quad (46)$$

where M_i and Λ_i are the multipliers respectively related to pure control constraints and dynamical constraints at instant t_i . Dynamical constraints have been rescaled to mirror the quadrature applied exactly in the context of the continuous-time OCP.

Optimality necessary conditions for the solution of Problem in (44), the *Karush–Kuhn–Tucker* (KKT) conditions, are obtained by nulling the gradients of Eq. (46). The KKT conditions concerning stationarity with respect to X_i, U_i and t_f are reported in Eq. (47), where X_0 is fixed, coherently with boundary conditions.

$$\begin{aligned} \frac{2}{t_f - t_0} D_i^T \Lambda &= w_i \nabla_{X_i} \mathcal{L}_i + J_{X_i}^T f_i \Lambda_i & i = 1, \dots, n-1 \\ \mathbf{0} &= w_i \nabla_{U_i} \mathcal{L}_i + J_{U_i}^T f_i \Lambda_i + \nabla_{U_i}^T g_{u,i} M_i & i = 0, \dots, n-1 \\ D_n^T \Lambda &= \mathbf{0} \\ \frac{\partial}{\partial t_f} \left[\frac{t_f - t_0}{2} \sum_{i=0}^{n-1} (w_i \mathcal{L}_i + \Lambda_i^T f_i + M_i g_{u,i}) \right] &= 0 \end{aligned} \quad (47)$$

where D_i^T is $(i+1)^{\text{th}}$ row of matrix D^T and $f_i \doteq f(X_i, U_i)$.

A Covector Mapping Theorem for the LGR collocation scheme is derivable by transcribing dynamics of the Hamiltonian system of formulation in (7) with the same LGR collocation scheme outlined for the primal system; among the resulting equations, the dynamical constraints and the boundary conditions of interest are reported followingly, where $\tilde{\lambda}_i$ and $\tilde{\mu}_i$ are the discrete costates and discrete path multipliers at i^{th} collocation point and D^\dagger is the differentiation operator for the costates.

$$\begin{aligned} -\frac{2}{t_f - t_0} D_i^\dagger \tilde{\lambda} &= \nabla_x \mathcal{L}_i + J_x^T f_i \tilde{\lambda}_i & i = 1, \dots, n-1 \\ \mathbf{0} &= \nabla_u \mathcal{L}_i + J_u^T f_i \tilde{\lambda}_i + \nabla_u^T g_{u,i} \tilde{\mu}_i & i = 0, \dots, n-1 \\ \tilde{\lambda}_n &= \mathbf{0} \\ \frac{\partial}{\partial t_f} \left[\frac{t_f - t_0}{2} \sum_{i=0}^{n-1} w_i (\mathcal{L}_i + \tilde{\lambda}_i^T f_i + \tilde{\mu}_i g_{u,i}) \right] &= 0 \end{aligned} \quad (48)$$

Since $t_n = t_f$ is not a collocation point for the employed scheme, \mathcal{H}_f is obtained deriving with respect to t_f the quadrature of \mathcal{H} over the collocation points, and $\tilde{\lambda}_n$ is extrapolated exploiting $\tilde{\lambda}$.

Identification of conditions in Eq. (47) with conditions in Eq. (48) is completed by analyzing the differential operators \mathbf{D}^\dagger and \mathbf{D}^\top . It can be proven that \mathbf{D}^\dagger is a differential operator for the space of polynomials of degree $n - 1$, provided its components are built according to the following equations [36]

$$\begin{aligned} \mathbf{D}_{0,0}^\dagger &= -\mathbf{D}_{0,0} - \frac{1}{w_0} \mathbb{I}_{n_s \times n_s} \\ \mathbf{D}_{i,j}^\dagger &= -\frac{w_j}{w_i} \mathbf{D}_{j,i} \quad \text{otherwise} \end{aligned} \quad (49)$$

Defining \mathbf{W} as the diagonal matrix of quadrature weights w_i such that $\mathbf{W}_{i,i} = w_i \mathbb{I}_{n_s \times n_s}$, Eq. (49) determines the following relations

$$\mathbf{D}_i^\top = -w_i \mathbf{D}_i^\dagger \mathbf{W}^{-1} \implies \frac{1}{w_i} \frac{2}{t_f - t_0} \mathbf{D}_i^\top \boldsymbol{\Lambda} = -\frac{2}{t_f - t_0} \mathbf{D}_i^\dagger \tilde{\lambda} \quad (50)$$

This proves that the left hand-side of the discrete costate dynamics in Eq. (48) is equivalent to the left-hand side of the stationarity condition with respect to state in Eq. (47), provided that a scaling through the quadrature weight is applied. The affine transformation outlined in Eq. (51) exploits nodal LGR quadrature weights to map multipliers to costates, thus making up the CMT for the employed transcription scheme: its application allows to exactly match Eq. (47) and Eq. (48), proving the existence of an exact link between the indirect and direct formulations of the OCP. [The interested reader is referred to the full derivation of the employed CMT, in \[36\].](#)

$$\begin{aligned} \tilde{\lambda}_n &= \mathbf{D}_n^\top \boldsymbol{\Lambda} \\ \tilde{\lambda}_i &= \frac{\boldsymbol{\Lambda}_i}{w_i} \quad i = 0, \dots, n-1 \\ \tilde{\mu}_i &= \frac{M_i}{w_i} \quad i = 0, \dots, n-1 \end{aligned} \quad (51)$$

C. Convex powered descent guidance problem

Let us consider the change of variables (on the left), objective function (in the center) and acceleration constraints (on the right) in Eq. (52). The problem associated with such contributions is the lossless convex version of problem \mathbb{P} : the non-convexity of thrust limitations is relaxed by 1) the use of u_T , 2) the Second-Order Cone constraint on \mathbf{u} and σ

and 3) the linearization and parabolic approximations of bounds over σ .

$$\begin{aligned} \mathbf{u} &\doteq \frac{u_T T_{\max}}{m} \mathbf{i}_T \\ \sigma &\doteq \frac{u_T T_{\max}}{m} \\ \tilde{m} &\doteq \log(m) \end{aligned} \quad \mathcal{J}_{\text{cvx}} = \int_0^{t_f} \sigma dt \quad \text{and} \quad \begin{cases} u \leq \sigma \\ \sigma \leq \rho_u e^{-\tilde{m}_u} [1 - (\tilde{m} - \tilde{m}_u)] \\ \sigma \geq \rho_l e^{-\tilde{m}_l} [1 - (\tilde{m} - \tilde{m}_l) + 0.5(\tilde{m} - \tilde{m}_l)^2] \\ \tilde{m}_u \doteq \log(m_0 - \alpha u_{T,\min} t); \rho_u \doteq u_{T,\max} T_{\max} \\ \tilde{m}_l \doteq \log(m_0 - \alpha u_{T,\max} t); \rho_l \doteq u_{T,\min} T_{\max} \end{cases} \quad (52)$$

Such approach limits the non-convex terms in the dynamics formulation and speeds up the direct step. The modified state \mathbf{x}_{cvx} dynamics reads then

$$\dot{\mathbf{x}}_{\text{cvx}} = \begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \\ \dot{\tilde{m}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{u} + \mathbf{g} \\ -\frac{\sigma}{I_{sp} g_0} \end{bmatrix} = \mathbf{f}_{\text{cvx}}(\mathbf{x}_{\text{cvx}}, \mathbf{u}, \sigma) \quad (53)$$

The problem \mathbb{P} can be then formulated in lossless convex form \mathbb{P}_{LCvx} as

$$\min_{\mathbf{u}, \sigma, t_f} \mathcal{J}_{\text{cvx}} \quad \text{s.t.} \quad \begin{cases} \dot{\mathbf{x}}_{\text{cvx}} = \mathbf{f}_{\text{cvx}}(\mathbf{x}_{\text{cvx}}, \mathbf{u}, \sigma) \\ \text{Constraints in Eq. (52)} \\ \mathbf{r}_f = \mathbf{0}_{3 \times 1} \\ \mathbf{v}_f = \mathbf{0}_{3 \times 1} \end{cases} \quad (54)$$

Remark 6: The problem obtained from the convexification is Acceleration Optimal (AO), justifying the original problem formulation of \mathbb{P} as AO.

Problem in (54) is transcribed with the pseudospectral scheme described in Sec. V.B, and the lossless convexification ensures its solution coincides with the original problem one. However, the objective function, the dynamics LHS and the constraints over σ depend on time; this happens in such way that the contribution of free final time is non-convex when problem in (54) is transcribed. Successive convexifications are therefore required: time-related terms of transcribed \mathbb{P} are linearized about the guessed final time, the obtained convex subproblem is solved and the solution constitutes the guess for a successive subproblem. *Such sequential transcription of free-final-time contributions, introduced for discrete-time dynamics in [8] and adapted to the hp pseudospectral collocation in [11], is in this work rearranged as analytically exact, as per Eq. (55) and Eq. (56).*

Let $\mathbf{X}_{\text{cvx}}^k, \mathbf{U}^k, \Sigma^k, t_f^k$ denote the solution of the k^{th} subproblem of the succession. Moreover, the \bar{n} segments domain

is divided in are arbitrarily fixed as equal. Therefore, for the $(k + 1)^{\text{th}}$ subproblem, dynamical constraints provided by (41) are linearized according to the following system of equations, and read

$$\frac{2\bar{n}}{t_f^k} \mathbf{D}_i \mathbf{X}_{\text{cvx}}^{k+1} = \mathbf{f}_{\text{cvx}}(\mathbf{X}_{\text{cvx}}^{k+1}, \mathbf{U}^{k+1}, \Sigma^{k+1}) + \frac{2}{(t_f^k)^2} \mathbf{D}_i \mathbf{X}_{\text{cvx}}^k \left(t_f^{k+1} - t_f^k \right) \quad \text{with } i = 0, \dots, n-1 \quad (55)$$

which corresponds to a linear constraint on $\mathbf{X}_{\text{cvx}}^{k+1}, \mathbf{U}^{k+1}, \Sigma^{k+1}, t_f^{k+1}$, and the condition $t_0 = 0$ has been taken back into account. For simplicity the superscript indicating the segment is dropped. Similarly, the objective function is linearized according to

$$\mathcal{J}_{\text{cvx}}^{k+1} = \frac{t_f^k}{2\bar{n}} \sum_{j=1}^{\bar{n}} \sum_{i=0}^p w_i \Sigma_i^{k+1} + \frac{1}{2\bar{n}} \sum_{j=1}^{\bar{n}} \sum_{i=0}^p w_i \Sigma_i^k \left(t_f^{k+1} - t_f^k \right) \quad (56)$$

Constraints over σ in Eq. (52) are time dependant: however, it has been experimentally verified that imposing such constraints with fixed final time does not sensibly alter estimated multipliers, which constitute the ultimate aim of the convex direct optimization.

Remark 7: We stress that retaining the final-time-dependant factor $\frac{2\bar{n}}{t_f}$ as multiplying the term $\mathbf{D}_i \mathbf{X}_{\text{cvx}}$ allows to decouple the final time from the dynamics RHS \mathbf{f}_{cvx} , with evident improvements over algorithm implementation simplicity.

The problem \mathbb{P}_{SCvx} solved at the generic $(k + 1)^{\text{th}}$ direct convex step reads, therefore

$$\min_{\mathbf{U}^{k+1}, \Sigma^{k+1}, t_f^{k+1}} \mathcal{J}_{\text{cvx}}^{k+1} \quad \text{s.t.} \quad \left\{ \begin{array}{l} \text{Scaled}^{\ddagger} \text{dynamical constraints in Eq. (55)} \\ \text{Constraints in Eq. (52) fixing } t_f \text{ at } t_f^k \\ \mathbf{R}_f = \mathbf{0}_{3 \times 1} \\ \mathbf{V}_f = \mathbf{0}_{3 \times 1} \end{array} \right. \quad (57)$$

Remark 8: The use of sequential convex methods may allow to adapt the convexification-based guess generation employed in this work to more complex scenarios. High maturity has been indeed reached for what concerns both attitude guidance and highly constrained formulations ([9, 54–57]).

D. Convex pseudospectral-based hybrid algorithm

Tools used in the hybrid method have been discussed and outlined. Starting from the algorithm end, the indirect single shooting has the guess provided by the convex direct collocation scheme. Such scheme is outlined in the box *Convex pseudospectral-based guess generation* in Fig. 4: the free-final-time formulation \mathbb{P}_{SCvx} is solved iteratively,

hence requires a first guess; this is provided by a fixed-final-time lossless formulation \mathbb{P}_{LCvx} (solved exactly in a single iteration thanks to its linear formulation). Final time for such last problem shall be fixed to a meaningful value t_f^0 . We opt for solving the free-final-time Fuel-Optimal 1-DoF PDG problem obtained from Problem (13), retaining only the dynamics and boundary conditions along the z -axis. t_f^0 is then fixed to the optimal final time of such 1-DoF problem, and computed as explained in the next paragraph.

The optimal thrust profile for the 1-DoF case features one bang only [58], the first thrust arc being characterized by idle-thrust level, the second arc by full-thrust level[§]. The system in Eq. (58) imposes that the initial height ($h_{0,\text{frw}}$) and vertical speed ($\dot{h}_{0,\text{frw}}$) propagated for time τt_f shall be equal to the same physical quantities (h_{bkw} and \dot{h}_{bkw}), propagated backwards in time for $(1 - \tau) t_f$, from landing conditions. This allows to compute the initial guess t_f^0 for the final time in few iterations, without heavily penalizing the computational time of the overall algorithm.

$$\begin{cases} h_{0,\text{frw}}(\tau, t_f) - h_{\text{bkw}}(\tau, t_f) = 0 \\ \dot{h}_{0,\text{frw}}(\tau, t_f) - \dot{h}_{\text{bkw}}(\tau, t_f) = 0 \end{cases} \quad (58)$$

The value tol for the stopping criterion of the successive convexifications iterations in Fig. 4 is imposed on the final time variation, and fixed to 10^{-2} s.

At last, discrete comass estimated from the convex step $\tilde{\lambda}_{\tilde{m}}$ is dual to the logarithmic mass \tilde{m} : the following mapping is applied to re-establish the duality with respect to the physical mass [59].

$$\tilde{\lambda}_m = \frac{1}{e^{\tilde{m}}} \tilde{\lambda}_{\tilde{m}} \quad (59)$$

Algorithm is schematized in Fig. 4.

VI. Results and Performances

In this section we first compare the presented algorithms in terms of performance. We then analyze the accuracy improvements provided by the indirect shooting after the convex direct step. Finally, we test the robustness of the proposed hybrid strategy through Monte-Carlo analyses. We firstly verify how the algorithm handles uncertainties with respect to guesses of final time computed as in Sec. V.D; then the full hybrid algorithm is tested against dispersions in initial conditions.

[‡]Coherently with formulation in Eq. (46), the dynamical constraints are provided to the optimizer as multiplied by the factor $t_f^k/2$

[§][58] considers null lower bound on thrust magnitude; however, it also demonstrates that fuel consumption is minimized if the final time is minimized; in doing so, the hypothesis of null lower bound on thrust magnitude is not employed. The optimal thrust profile features then an *idle-thrust*, *full-thrust* profile even for non-null lower thrust magnitude bounds.

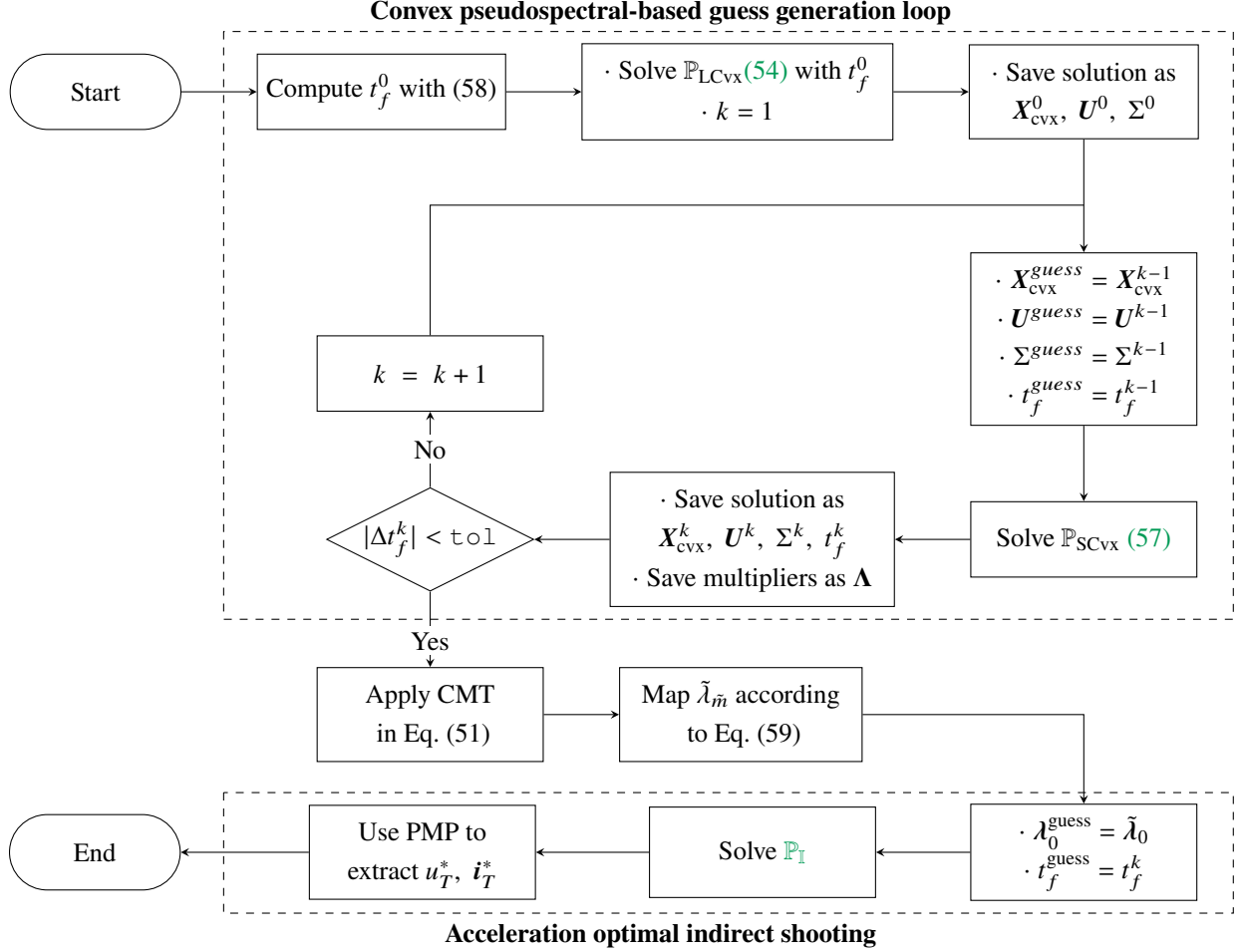


Fig. 4 Convex-based hybrid algorithm overview

A. Hybrid and Fully Indirect algorithms comparison

Results in this subsection are related to values of the trust region radius ξ in Eq. (23) equal to 1. No convergence issues are indeed observed for nominal initial conditions. As previously mentioned, the continuation scheme shall be correctly tuned to grant successful convergence at each iteration. For our purpose, it is sufficient using 10 homotopic steps, *i.e.* the 11th coinciding with the Acceleration-Optimal problem. Moreover, the homotopic parameter associated with the last step results $\varepsilon^n = 0.998$. The continuation steps of the control and mass profiles are reported in Fig. 5. For the considered boundary conditions and problem parameters, the optimal thrust magnitude profile appears from the results to require a single-bang control program; the full-thrust arc damps kinetic energy and the minimum-thrust arc is kept as long as possible so to minimize fuel consumption while guaranteeing correct landing site targeting.

Exhaustive comparison between the two guess generation mechanisms shall both consider computational time and accuracy; the latter indirectly affects the required iterations to solve the Acceleration-Optimal shooting. As far as the convex approach is considered, a discretization with 5 segments and 10 collocation points for each segment is chosen to

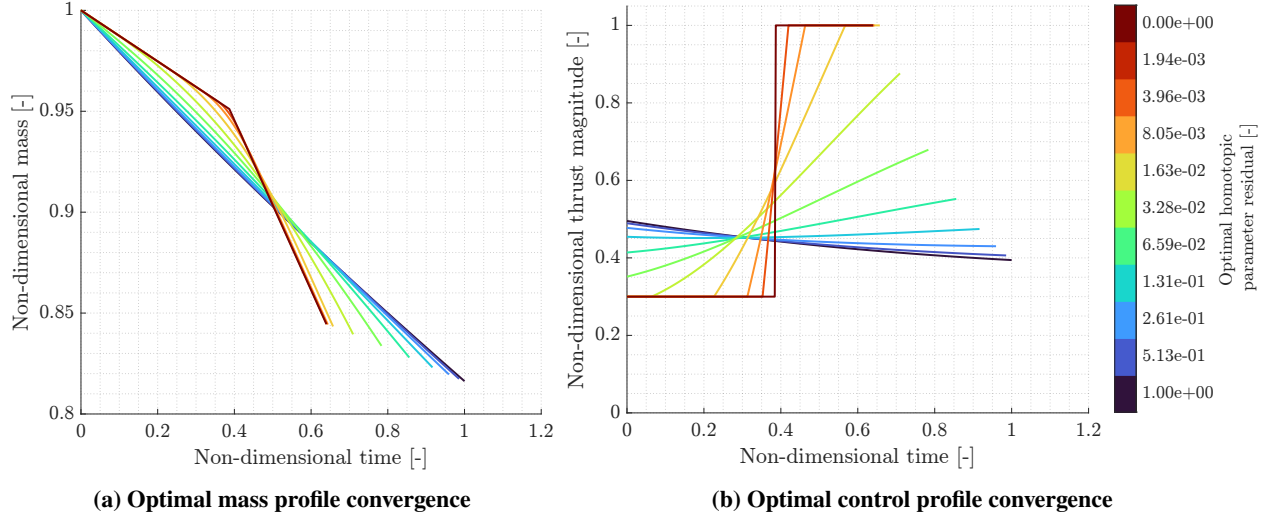


Fig. 5 Homotopic approach - Mass and control

grant good accuracy without penalizing computational times; the solver ECOS is used to solve each convex subproblem [60]. The absolute, relative and optimality tolerance parameters are fixed to 10^{-5} . The integration step employed in each indirect shooting integration is fixed to 0.1 s. Computational times are summarized in Tab. 2; for the hybrid algorithm fixed initial conditions are used, as it has been observed that the 1-DoF t_f computation always converges in a few iterations; within the indirect approach, instead, the initial costates are guessed within an interval $[-1, 1]$ with a random multi-start technique. Reported data correspond to a batch of 300 optimization runs for what concerns the hybrid

Table 2 Acceleration Optimal powered landing computational times

		Computational time [†] [s]
Hybrid algorithm	1-D t_f computation	$4.1 \cdot 10^{-3}$
	Convex Guess Generation	$5.15 \cdot 10^{-1}$
	CMT and costate mapping	$8.3 \cdot 10^{-4}$
	Indirect step	$1.66 \cdot 10^0$
Purely Indirect algorithm	Homotopic Guess Generation	$2.47 \cdot 10^1$
	Indirect step	$1.20 \cdot 10^0$

[†] Relative to Dell XPS w/ 2.6 GHz Intel Core i7-9750H, 16 GB 2666 MHz DDR4.

algorithm; considering the purely indirect, the population for the data in Tab. 2 is made up by the successfully converged 135 optimization runs out of a batch of 200. The computational time required for the homotopic scheme approximately amounts to the 2000% of the time needed for the Acceleration-Optimal shooting; the improvement attainable with a different guess generation mechanism is evident, bearing in mind that the overall computational time scale of the purely indirect algorithm is higher than 25 s. For what concerns the hybrid approach, instead, the guess generation time is drastically reduced, for an overall computational time smaller than 2.3 s. In addition to this, the CMT and costate

mapping feature negligible computational times, thus confirming the computational efficiency of the presented hybrid approach. At last, the estimation of the 1-D final time adds a degree of complexity to the guess generation algorithm without relevantly affecting the overall CPU time.

The accuracy of the guess-generation mechanisms is at last compared. Figure 6 highlights the performances of the convex-based guess generation, being the relative guesses associated with the dashed lines. Comass and cospeed accuracies are respectively better than the last and of the second-last iteration of the continuation scheme; coposition accuracy depends on the components, and is approximately comparable to the seventh step of the homotopic continuation. For what concerns the final time, not reported in Fig. 6, the percentual normalized error amounts to the 0.3% for the convex approach, and to 0.14% for the last iteration of the homotopic step. The guesses computed with the convex PS approach ensure convergence of the following Indirect Acceleration-Optimal Shooting; the lower accuracy with respect to the ones computed with the indirect continuation explains the difference in average computational time of the Indirect step for the hybrid algorithm. The advantages of the proposed hybrid algorithm are evident: computational

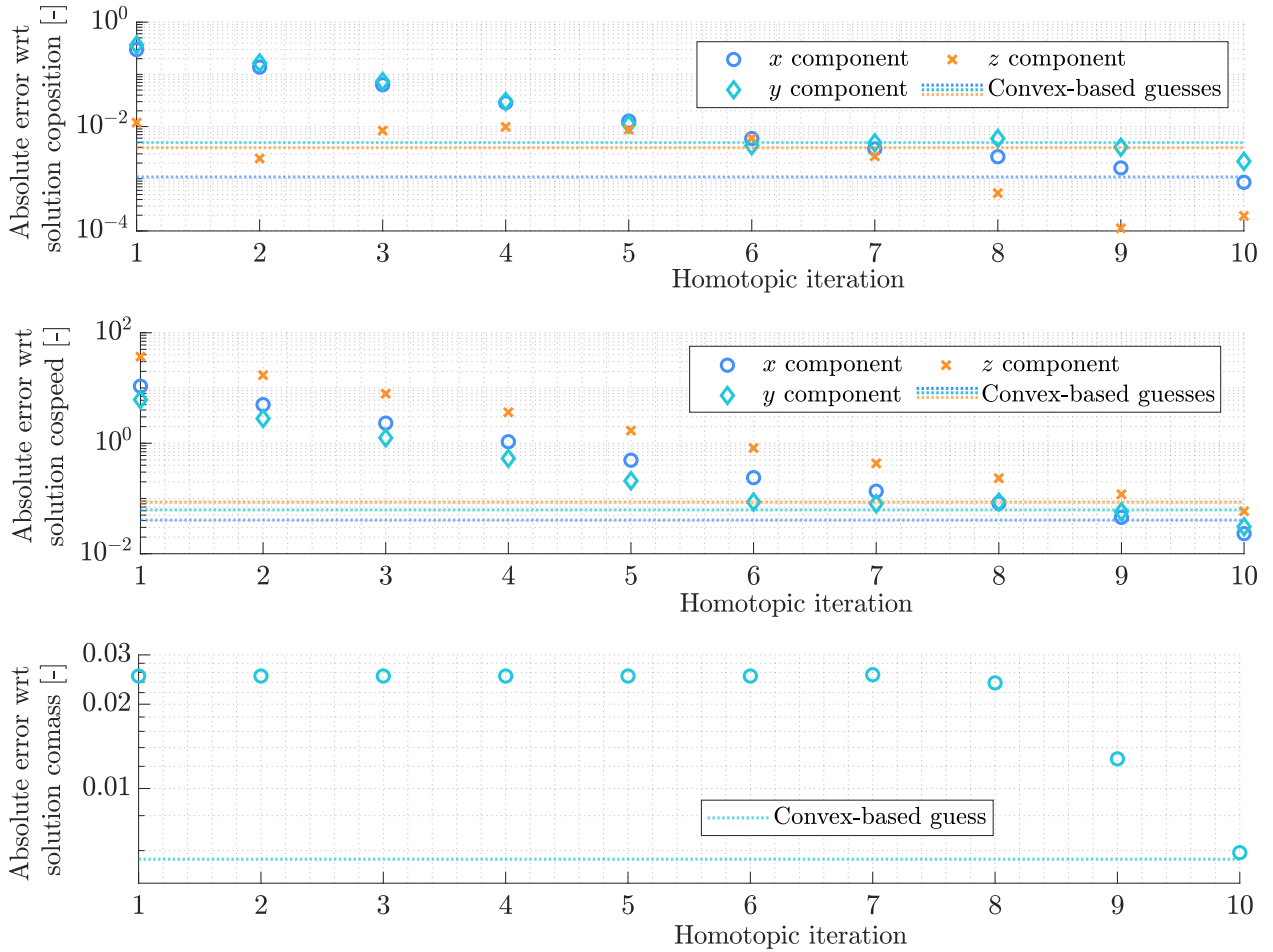


Fig. 6 Comparison of iterations of continuation scheme and costate guesses generated via the convex-based approach

time is sensibly improved with respect to a purely indirect approach tailored to the specific problem, while accuracy is nearly left unaltered. The validity of the presented convex pseudospectral-based guess generation algorithm is therefore confirmed for nominal initial conditions.

B. Convex direct step and indirect step comparison

The indirect step proves fundamental for refining the solution estimated with the convex direct step. The control profiles after both steps are represented in Fig.7a. Dots correspond to the collocation points employed within the direct

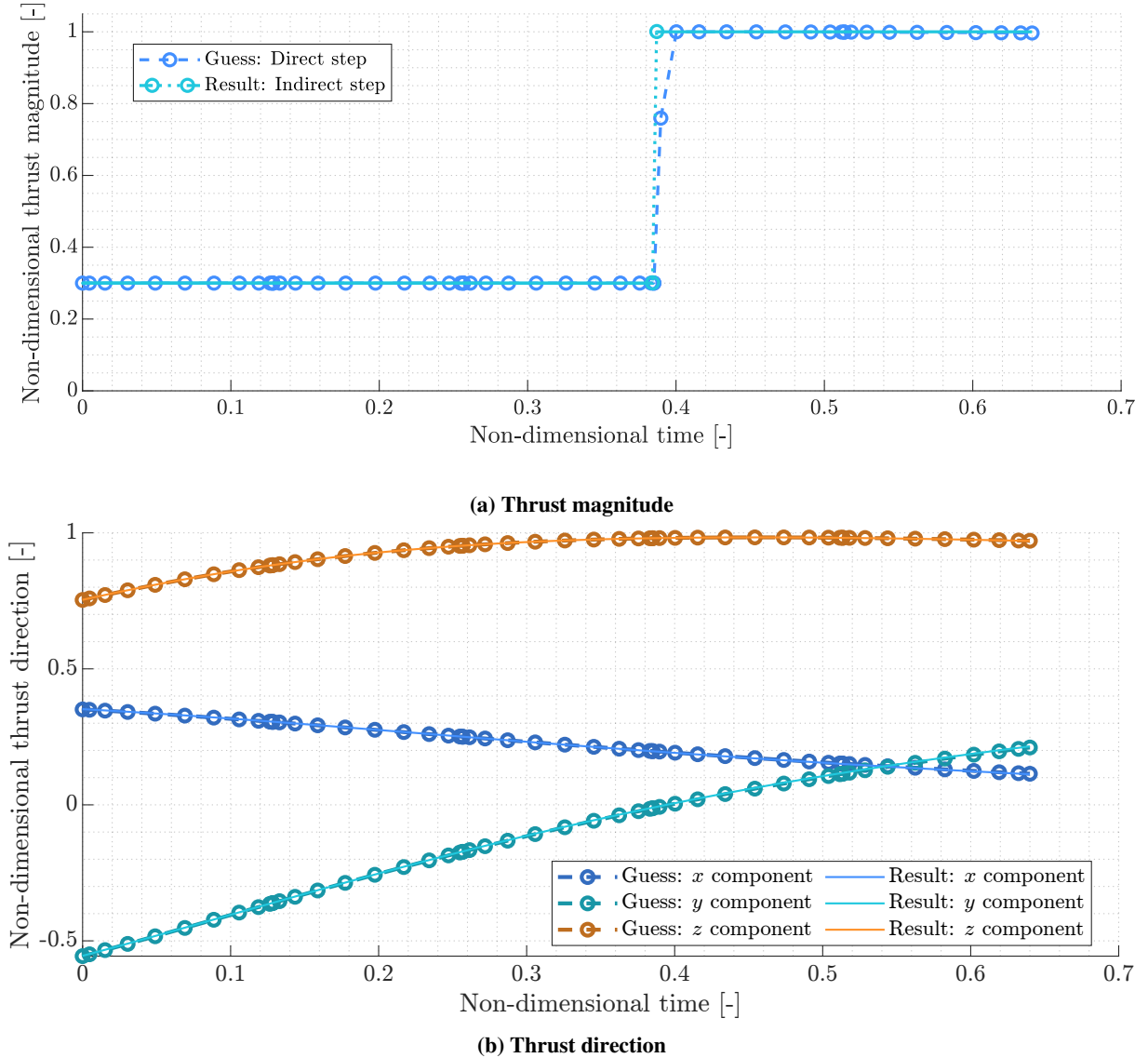


Fig. 7 Thrust obtained at the end of the direct and the indirect steps of the hybrid algorithm

step, while the control profile after the indirect step is continuous, as obtained with a shooting strategy. The discontinuity point makes an exception. Final control is not represented for the direct step, as dynamics is not collocated at final

time. The change from a nearly discontinuous profile, estimated with the convex step, to a completely discontinuous one can be appreciated; uncertainty over ignition time within the convex guess corresponds approximately to the 3% of the total time of flight. Thrust directions are instead reported in Fig. 7b, where it is shown that the direct step already provides a thrust direction close to the optimal. Moreover, the indirect step guarantees a sensitive improvement over final conditions satisfaction. Results are reported in Tab. 3. Dynamics has been propagated with a variable order Adams-Bashforth-Moulton propagator (propagation relative tolerance fixed to 10^{-12}) being thrust commands approximated with zero-order hold scheme.

Table 3 Final conditions[†] of propagated guess solution and final solution

Propagated final state	Convex Direct Step	Indirect Step
\mathbf{r}_f^{nd}	$[0.83; -7.50; -9.67] 10^{-3}$	$[1.05; -0.01; 3.27] 10^{-3}$
\mathbf{v}_f^{nd}	$[0.37; -2.88; -3.43] 10^{-2}$	$[0.31; 0.14; 1.89] 10^{-3}$

[†] \mathbf{r}_f^{nd} and \mathbf{v}_f^{nd} are both fixed to $[0; 0; 0]$ within the optimization.

C. Hybrid Algorithm Robustness analysis

Given the architecture of the direct convex step of the hybrid algorithm, the only physical quantity that can not be handled exactly by solving \mathbb{P}_{LCvx} is the free final time t_f . The first MC campaign focuses therefore on dispersions in t_f^0 values; it is then verified that chaining the solutions of problems $\mathbb{P}_{LCvx} - \mathbb{P}_{SCvx}$ is an efficient approach to handle such guess uncertainties. Results are reported in Fig. 8, where successful runs are depicted in light blue, and infeasibilities are highlighted with red lines.

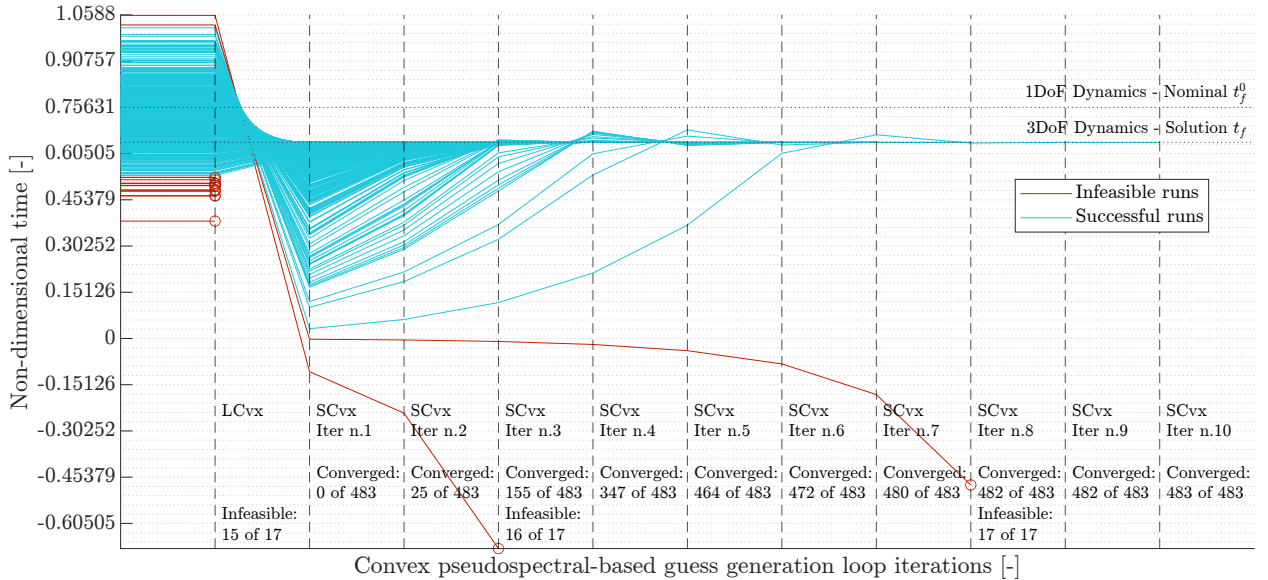


Fig. 8 Monte-Carlo analysis - Convergence of final time as function of iterations

The nominal final time t_f^0 found according to Sec. V.D is perturbed with a normal gaussian distribution of 1-sigma uncertainty of $0.15\bar{t}_f^0$. 500 runs are executed and for each run it is retrieved 1) the history of convergence of t_f and 2) the number of iterations of the convex guess generation loop.

A few considerations can be outlined analyzing Fig. 8. In first instance all the successful runs converge to the same t_f value: the consistency of the final time values found through the convex guess generation loop is therefore confirmed. Secondly, employed dispersions in t_f^0 are extremely conservative, and are easily overcome by the 1-DoF strategy proposed in Sec. V.D. Nonetheless, the sequence $\mathbb{P}_{\text{LCvx}} - \mathbb{P}_{\text{SCvx}}$ handles such dispersions well, demonstrating the efficiency of the convex guess generation loop, with a total number of successful runs higher than the 96%. We stress, however, that the proposed 1-DoF strategy grants even higher accuracy with respect to the one used to generate dispersions. In third instance, more than the 92% of the runs converges within 6 optimizations, *i.e.* 1 LCvx iteration and 5 SCvx iterations, while nearly all the infeasibilities are detected within the LCvx step. Such results demonstrate the accuracy of the sequential approach with respect to management of the final time contributions. At last, we highlight that the 17 infeasibilities are not intrinsic to the physical problem, but are associated with the way the convex PDG problem is posed. 1) Infeasibilities detected at the LCvx step can be handled by inserting *virtual controls*, as widely demonstrated in literature [8, 11]; these would allow the LCvx problem to be solveable. 2) The remaining 2 infeasibilities arise as the final time becomes negative at the first iteration of the SCvx; inserting *trust regions* to handle t_f linearization error [61] has been proven successful in previous works as well [8, 11]. Still, the 1-DoF strategy is conceived to avoid the previous numerical infeasibilities; the application of the suggested solutions is therefore out of the scope of this work.

The hybrid algorithm is then tested against initial conditions uncertainty. Specifically, considered dispersions are reported in Tab. 4: uncertainty over initial position, velocity and mass is taken into account.

Table 4 Monte-Carlo analysis - Uncertainties in initial conditions

Initial state	Dispersion [†]
Downrange	[0.228, 0.252]
Crossrange	[0.117, 0.123]
Height	[0.95, 1.05]
Downrange rate	[−0.686, −0.634]
Crossrange rate	[−0.013, 0.013]
Height rate	[−2.02, −1.94]
Mass	[0.98, 1.02]

[†] Corresponding to uniform distribution.

Dispersions in initial physical states sensibly impact on the evolution of costates; this in turn affects the accuracy of the STM in approximating dynamics of perturbations. Eventually, inaccuracies result in solution divergence within the indirect shooting. However, it has been verified that a trust region radius ξ in Eq. (23) smaller than 1 can handle

approximation errors, hence preventing from divergence. Fixing ξ , Tab. 5 reports computational times and number of successful runs for different ξ values: performances for the Indirect Shooting step are reported; trust region is indeed applied within the indirect step only.

Table 5 Correlation between ξ and AO Shooting performance

ξ	AO Shooting CPU Time [†] [s]	Successful runs [‡]
1	$1.97 \cdot 10^0$	93 %
0.5	$5.40 \cdot 10^0$	96 %
0.25	$7.66 \cdot 10^0$	100 %

[†] Relative to Dell XPS w/ 2.6 GHz Intel Core i7-9750H, 16 GB 2666 MHz DDR4.

[‡] Batch of 500 as statistical sample.

Results are graphically reported in Figs. 9 and 10. Fig. 9b highlights the sensitivity of the optimal switching time with respect to initial conditions; Fig. 9a shows instead that the final masses feature smaller dispersions with respect to the initial ones: final masses approximately lie in the interval $[0.83, 0.86]$. Considering instead Fig. 10, landing trajectories are represented on the left, while the speed components envelopes are reported on the right. The free final time and switching time formulation can be as well appreciated considering the velocity components.

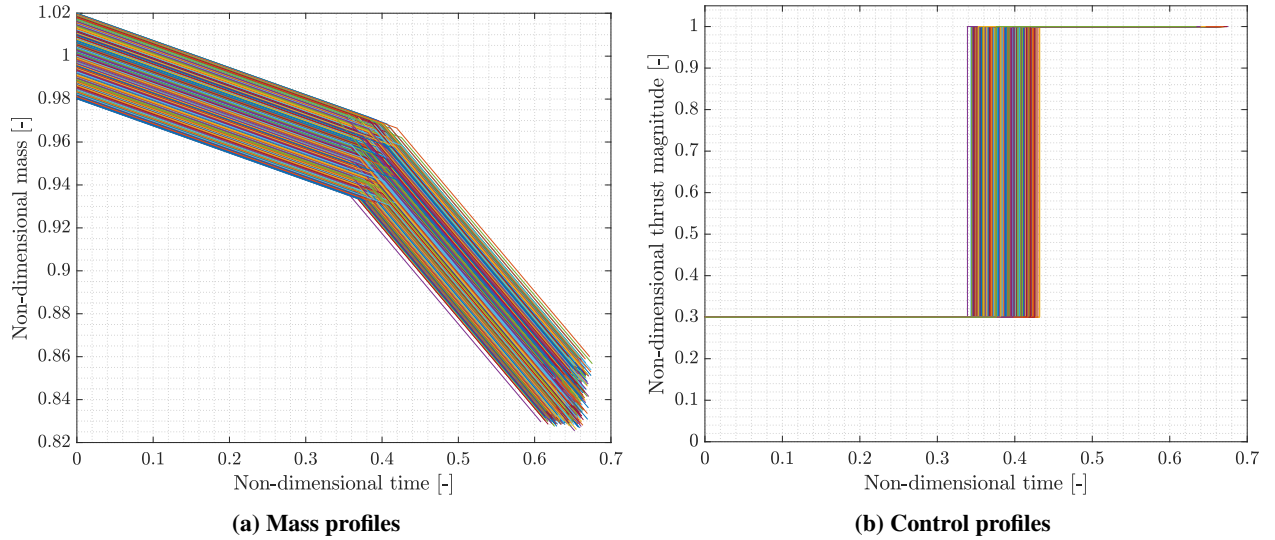


Fig. 9 Monte-Carlo analysis - Mass and control profiles

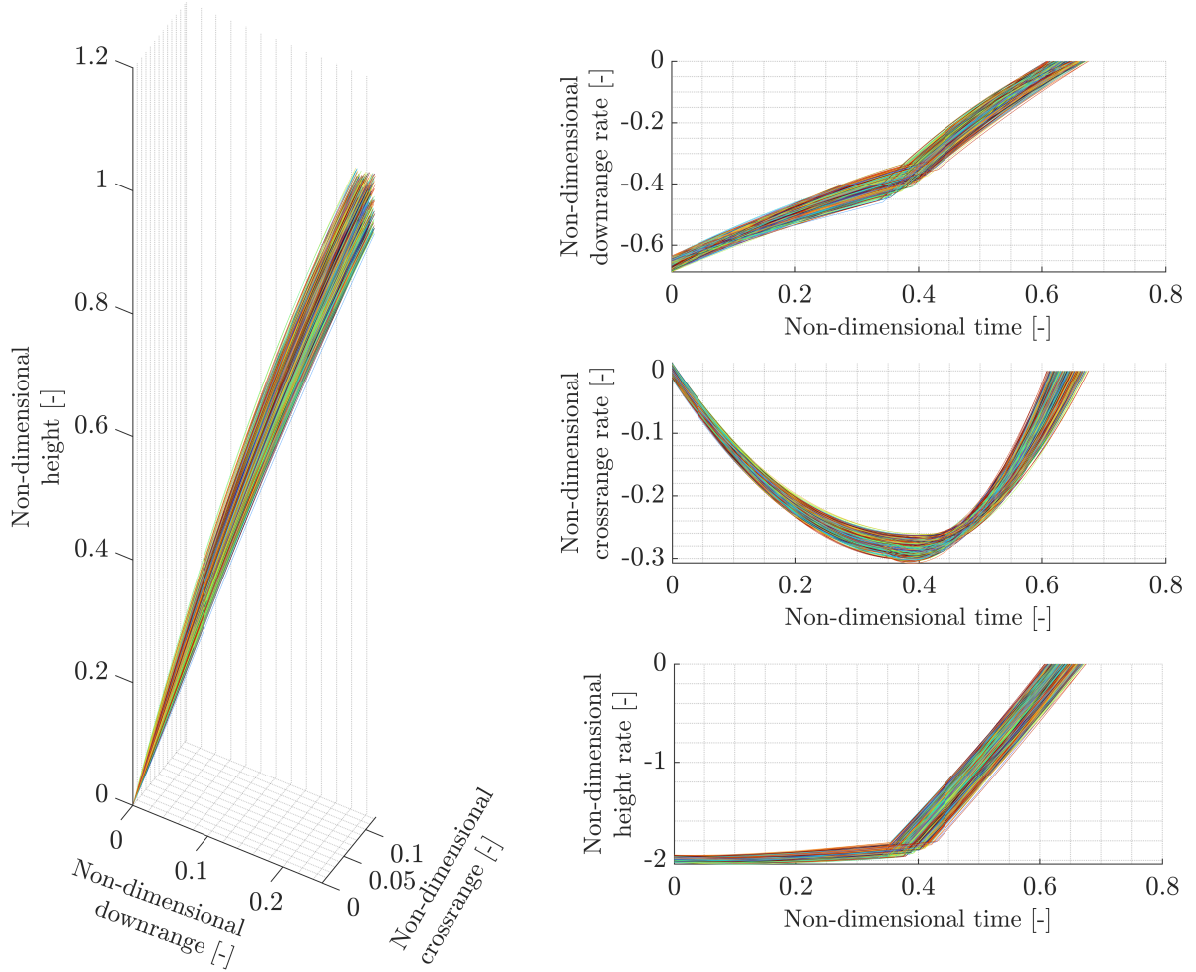


Fig. 10 Monte-Carlo analysis - Trajectories and velocity profiles

VII. Conclusions

In the present paper a novel hybrid strategy has been introduced and compared to a state-of-the-art purely indirect continuation-based technique. Both the two steps characterizing the hybrid scheme and the continuation-based technique have been extensively presented; results have been finally outlined for a landing scenario, modelled as a 3-D point mass. Rapidity of the hybrid strategy highlights its suitability for preliminary solution generation purposes; in addition, the gain with respect to the purely indirect homotopic approach demonstrates the superiority of the combination of convex optimization and indirect shooting over the state-of-the-art of offline guidance strategies.

Results from the Monte-Carlo analyses offer a clear view of algorithm performance under dispersions. Accuracy of costate generation mechanism fused with a trust region-augmented indirect step successfully initialize and handle the highly nonlinear flow of primal-dual dynamics; convergence percentages of the considered statistical sample seal the reliability of the proposed hybrid technique.

References

- [1] Rao, A. V., “A Survey of Numerical Methods for Optimal Control,” *Advances in the Astronautical Sciences*, Vol. 135, 2010.
- [2] Bryson, A. E. and Ho, Y., *Applied Optimal Control - Optimization, Estimation and Control*, Taylor & Francis, 1975.
- [3] Açıkmeşe, B. and Casoliva, J. and Carson, J. M. and Blackmore, L., “G-FOLD: A Real-Time Implementable Fuel Optimal Large Divert Guidance Algorithm for Planetary Pinpoint Landing,” *Concepts and Approaches for Mars Exploration*, edited by LPI Editorial Board, 2012.
- [4] Blackmore, L., “Autonomous precision landing of space rockets,” *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium*, Vol. 46, Washington, DC: The National Academies Press, 2017. <https://doi.org/10.17226/23659>.
- [5] Nesterov, Y. and Nemirovskii, A., *Interior-Point Polynomial Algorithms in Convex Programming*, Society for Industrial and Applied Mathematics, 1994.
- [6] Açıkmeşe, B. and Ploen, S. R., “Convex Programming Approach to Powered Descent Guidance for Mars Landing,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007. <https://doi.org/10.2514/1.27553>.
- [7] Blackmore, L. and Açıkmeşe, B. and Scharf, D. P., “Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization,” *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, 2010. <https://doi.org/10.2514/1.47202>.
- [8] Szmuk, M. and Eren, U. and Açıkmeşe, B., “Successive Convexification for Mars 6-DoF Powered Descent Landing Guidance,” *AIAA Guidance, Navigation and Control Conference*, 2017. <https://doi.org/10.2514/6.2017-1500>.
- [9] Szmuk, M. and Reynolds, T. P. and Açıkmeşe, B., “Successive Convexification for Real-Time Six-Degree-of-Freedom Powered Descent Guidance with State-Triggered Constraints,” *Journal of Guidance, Control, and Dynamics*, Vol. 43, 2020. <https://doi.org/10.2514/1.G004549>.
- [10] Sagliano, M. and Tsukamoto, T. and Maces, A. and Seelbinder, D. and Ishimoto, S. and Dumont, E., “Guidance and Control Strategy for the CALLISTO Flight Experiment,” *EUCASS 2019*, 2019. <https://doi.org/10.13009/EUCASS2019-284>.
- [11] Sagliano, M. and Heidecker, A. and Hernández, J. M. and Farì, S. and Schlotterer, M. and Woicke, S. and Seelbinder, D. and Dumont, E., “Onboard Guidance for Reusable Rockets: Aerodynamic Descent and Powered Landing,” *AIAA Scitech Forum*, 2021. <https://doi.org/10.2514/6.2021-0862>.
- [12] Botelho, A. and Martinez, M. and Recupero, C. and De Zaiacomo, G., “Design of the landing guidance for the retro-propulsive vertical landing of a reusable rocket stage,” *CEAS Space Journal*, 2022. <https://doi.org/10.1007/s12567-022-00423-6>.
- [13] 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. <https://doi.org/10.1002/oca.709>.

- [14] Zhang, C. and Topputo, F. and Zhao, Y., “Low-Thrust Minimum-Fuel Optimization in the Circular Restricted Three-Body Problem,” *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 8, 2015. <https://doi.org/10.2514/1.G001080>.
- [15] Pan, B., Pan, X., and Lu, P., “Finding Best Solution in Low-Thrust Trajectory Optimization by Two-Phase Homotopy,” *Journal of Spacecrafts and Rockets*, Vol. 56, No. 1, 2019. <https://doi.org/10.2514/1.A34144>.
- [16] Lu, P. and Sun, H. Tsai, B., “Closed-Loop Endoatmospheric Ascent Guidance,” *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, 2003. <https://doi.org/10.2514/2.5045>.
- [17] Brendel, E. and Hérissé, B. and Bourgeois, E., “Optimal guidance for Toss Back concepts of Reusable Launch Vehicles,” *EUCASS 2019*, 2019. <https://doi.org/10.13009/EUCASS2019-232>.
- [18] Mansell, J. R. and Grant, M. J., “Adaptive Continuation Strategy for Indirect Hypersonic Trajectory Optimization,” *Journal of Spacecraft and Rockets*, Vol. 55, No. 4, 2018. <https://doi.org/10.2514/1.A34013>.
- [19] Mall, K. and Grant, M. J. and Taheri, E., “Uniform Trigonometrization Method for Optimal Control Problems with Control and State Constraints,” *Journal of Spacecraft and Rockets*, Vol. 57, No. 5, 2020. <https://doi.org/10.2514/1.A34624>.
- [20] Mall, K. and Taheri, E., “Three-Degree-of-Freedom Hypersonic Reentry Trajectory Optimization Using an Advanced Indirect Method,” *Journal of Spacecraft and Rockets*, Vol. 59, No. 5, 2022. <https://doi.org/10.2514/1.A34893>.
- [21] Vedantam, M. and Akella, M. R. and Grant, M. J., “Multistage Stabilized Continuation for Indirect Optimal Control of Three-Dimensional Hypersonic Trajectories,” *Journal of Spacecraft and Rockets*, Vol. 59, No. 6, 2022. <https://doi.org/10.2514/1.A35313>.
- [22] Kang, W., “The rate of convergence for a pseudospectral optimal control method,” *2008 47th IEEE Conference on Decision and Control*, 2008. <https://doi.org/10.1109/CDC.2008.4738608>.
- [23] Darby, C. and Hager, W. and Rao, A., “An hp-Adaptive Pseudospectral Method for Solving Optimal Control Problems,” *Optimal Control Applications and Methods*, Vol. 32, 2011. <https://doi.org/10.1002/oca.957>.
- [24] Sagliano, M., “Generalized *hp* Pseudospectral-Convex Programming for Powered Descent and Landing,” *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 7, 2019. <https://doi.org/10.2514/1.G003731>.
- [25] Huntington, G., and Rao, A., “Optimal Reconfiguration of Spacecraft Formations Using the Gauss Pseudospectral Method,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, 2008. <https://doi.org/10.2514/1.31083>.
- [26] Lu, P., and Liu, X., “Autonomous Trajectory Planning for Rendezvous and Proximity Operations by Conic Optimization,” *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 2, 2013. <https://doi.org/10.2514/1.58436>.
- [27] Hofmann, C. and Topputo, F., “Rapid Low-Thrust Trajectory Optimization in Deep Space Based on Convex Programming,” *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 7, 2021. <https://doi.org/10.2514/1.G005839>.

- [28] Sagliano, M., "Pseudospectral Convex Optimization for Powered Descent and Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 2, 2018. <https://doi.org/10.2514/1.G002818>.
- [29] Wang, J., and Cui, N., "A Pseudospectral-Convex Optimization Algorithm for Rocket Landing Guidance," *AIAA Scitech Forum*, 2018. <https://doi.org/10.2514/6.2018-1871>.
- [30] Sagliano, M. and Mooij, E., "Optimal drag-energy entry guidance via pseudospectral convex optimization," *Aerospace Science and Technology*, Vol. 117, 2021. <https://doi.org/10.1016/j.ast.2021.106946>.
- [31] Garrido, J. and Sagliano, M., "Ascent and Descent Guidance of Multistage Rockets via Pseudospectral Methods," *AIAA Scitech Forum*, 2021. <https://doi.org/10.2514/6.2021-0859>.
- [32] Malyuta, D., Reynolds, T., Szmuk, M., Mesbahi, M., Açıkmeşe, B., and Carson, J., "Discretization Performance and Accuracy Analysis for the Rocket Powered Descent Guidance Problem," *AIAA Scitech Forum*, 2019. <https://doi.org/10.2514/6.2019-0925>.
- [33] Hofmann, C., Morelli, A. C., and Toppato, F., "On the Performance of Discretization and Trust-Region Methods for On-Board Convex Low-Thrust Trajectory Optimization," *AIAA Scitech Forum*, 2022. <https://doi.org/10.2514/6.2022-1892>.
- [34] Fahroo, F. and Ross, I. M., "Costate Estimation by a Legendre Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 3, 2001. <https://doi.org/10.2514/2.4709>.
- [35] Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method," *Journal of Guidance, Control, and Dynamics*, Vol. 29, 2006. <https://doi.org/10.2514/1.20478>.
- [36] Garg, D., Patterson, M., Francolin, C., Darby, C., Huntington, G. T., Hager, W., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation of Finite-Horizon and Infinite-Horizon Optimal Control Problems via a Radau Pseudospectral Method," *Computational Optimization and Applications*, Vol. 49, 2011. <https://doi.org/10.1007/s10589-009-9291-0>.
- [37] Josselyn, S., and Ross, I. M., "Rapid Verification Method for the Trajectory Optimization of Reentry Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 3, 2003. <https://doi.org/10.2514/2.5074>.
- [38] Bedrossian, N. S., Bhatt, S., Kang, W., and Ross, I. M., "Zero-Propellant Maneuver Guidance," *IEEE Control Systems Magazine*, Vol. 29, No. 5, 2009. <https://doi.org/10.1109/MCS.2009.934089>.
- [39] Bulirsch, R. and Nerz, E. and Pesch, H. J. and von Stryk, O., *Combining Direct and Indirect Methods in Optimal Control: Range Maximization of a Hang Glider*, Birkhäuser Basel, Basel, 1993, pp. 273–288. https://doi.org/10.1007/978-3-0348-7539-4_20.
- [40] Bonalli, R., Cauligi, A., Bylard, A., and Pavone, M., "GuSTO: Guaranteed Sequential Trajectory optimization via Sequential Convex Programming," 2019. <https://doi.org/10.1109/ICRA.2019.8794205>.
- [41] Lu, P., "Propellant-Optimal Powered Descent Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 41, 2017, pp. 1–14. <https://doi.org/10.2514/1.G003243>.

- [42] Liu, X., “Fuel-Optimal Rocket Landing with Aerodynamic Controls,” *Journal of Guidance, Control, and Dynamics*, Vol. 42, 2019. <https://doi.org/10.2514/1.G003537>.
- [43] Longuski, J.M. and Guzmán, J. and Prussing, J.E., *Optimal Control with Aerospace Applications*, Springer, 2014.
- [44] Pontryagin, L.S., *The Mathematical Theory of Optimal Processes*, John Wiley & Sons, 1962.
- [45] Sforza, P. M., *Manned Spacecraft Design Principles*, Butterworth-Heinemann, Boston, 2016.
- [46] You, S. and Dai, R. and Rea, J. R., “Theoretical Analysis of Fuel-Optimal Powered Descent Problem with State Constraints,” *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 12, 2022. <https://doi.org/10.2514/1.G006815>.
- [47] Lawden, D. F. , *Optimal trajectories for space navigation*, Butterworths, London, 1963.
- [48] Hairer, E. and Lubich, C. and Wanner, G., *Geometric Numerical Integration - Structure-Preserving Algorithms for Ordinary Differential Equations*, Springer, 2005.
- [49] Russell, R. P., “Primer Vector Theory Applied to Global Low-Thrust Trade Studies,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 2, 2007. <https://doi.org/10.2514/1.22984>.
- [50] Morelli, A. C. and Hofmann, C. and Topputo, F., “Robust Low-Thrust Trajectory Optimization Using Convex Programming and a Homotopic Approach,” *IEEE Transactions on Aerospace and Electronic Systems*, 2021. <https://doi.org/10.1109/TAES.2021.3128869>.
- [51] Wang, Z. and Grant, J., “Optimization of Minimum-Time Low-Thrust Transfers Using Convex Programming,” *Journal of Spacecraft and Rockets*, Vol. 55, No. 3, 2017. <https://doi.org/10.2514/1.A33995>.
- [52] Liu, X. and Shen, Z. and Lu, P., “Entry Trajectory Optimization by Second-Order Cone Programming,” *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 2, 2016. <https://doi.org/10.2514/1.G001210>.
- [53] Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004.
- [54] Szmuk, M. and Açıkmeşe, B., *Successive Convexification for 6-DoF Mars Rocket Powered Landing with Free-Final-Time*, ??? <https://doi.org/10.2514/6.2018-0617>.
- [55] Zhang, Y. and Huang, J. and Cui, H., “Trajectory Design via Convex Optimization for Six-Degree-of-Freedom Asteroid Powered Landing,” *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 4, 2021, pp. 779–792. <https://doi.org/10.2514/1.G004434>.
- [56] Sagliano, M. and Lu, P. and Seelbinder, D. and Theil, S., “Six-Degree-of-Freedom Rocket Landing Optimization by Augmented Convex-Concave Decomposition,” *AIAA SCITECH 2023 Forum*, ??? <https://doi.org/10.2514/6.2023-2005>.
- [57] Kamath, A. G., Elango, P., Kim, T., Mceowen, S., Yu, Y., Carson, J. M., Mesbahi, M., and Açıkmeşe, B., “Customized Real-Time First-Order Methods for Onboard Dual Quaternion-based 6-DoF Powered-Descent Guidance,” *AIAA SCITECH 2023 Forum*, ??? <https://doi.org/10.2514/6.2023-2003>.

- [58] Meditch, J., “On the problem of optimal thrust programming for a lunar soft landing,” *IEEE Transactions on Automatic Control*, Vol. 9, No. 4, 1964. <https://doi.org/10.1109/TAC.1964.1105758>.
- [59] Taheri, E. and Arya, V. and Junkins, J. L., “Costate mapping for indirect trajectory optimization,” *Astrodynamics*, Vol. 5, 2021. <https://doi.org/10.1007/s42064-021-0114-0>.
- [60] Domahidi, A. and Chu, E. and Boyd, S., “ECOS: An SOCP Solver for Embedded Systems,” *European control Conference (ECC)*, 2013. <https://doi.org/10.23919/ECC.2013.6669541>.
- [61] Yang, R. and Liu, X., *Comparison of Convex Optimization-Based Approaches to Solve Nonconvex Optimal Control Problems*, AIAA SciTech Forum, 2019. <https://doi.org/10.2514/6.2019-1666>.