POLITECNICO
MILANO 1863
SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

DLR Institute of
Space Systems

Executive Summary

Hybrid Direct-Indirect Strategy for Optimal Landing Guidance of Reusable Rockets

Master of Science in Space Engineering

Author: **Fabio Spada**

Advisor: **Prof. Francesco Topputo**

Co-advisor: **Dr. Marco Sagliano**

Academic year: **2021-2022**

1. Introduction

On November 5, 2006, the International Space Station performed a zero-propellant reorientation maneuver that saved NASA $1,000,000. The relative Optimal Control Problem (OCP) was solved with a hybrid method: the employed algorithm solved a Nonlinear Programming (NLP) problem using a pseudospectral trsncription scheme; the Optimality Necessary Conditions associated with the original problem were at the same time satisfied; the optimization algorithm, therefore, merged the robustness of direct transcription with the accuracy of indirect methods, thanks to a Covector Mapping Theorem (CMT) [1]. ISS exploited environmental disturbances to complete a 90° reorientation without use of thrusters.

Eleven years later, on March, 2017, SpaceX reused successfully a first stage for the first time; the stage performed a vertical landing autonomously, on a barge two times wider than landing gear footprint. Precision was achieved with an online guidance algorithm that iteratively solved *convexified* formulations of the original pinpoint landing problem [2].

The following work gathers the algorithmic workhorses of the guidance milestones just described, and applies them to the Aerodynamic Powered Landing Problem; it merges a pseudospectral convex strategy based on Legendre-Gauss-Radau (LGR) transcription with a tailored indirect collocation scheme, managing to exploit aerodynamic disturbances to minimize fuel consumption during vertical landing: drag damps kinetic energy, while lift deflects trajectory in turn of thrusters. The elaborated strategy evaluates accurate solutions with reduced computational times; convex solver, based on previous studies led within the Institute of Space Sytems, DLR [3] generates guesses; the aforementioned CMT maps multipliers to costates and then feeds the indirect solver that refines them to obtain the problem optimum. The indirect solver is an original contribution inspired by remeshing strategies elaborated within DART Group, in Politecnico di Milano [4].

The present Executive Summary is organized as follows. The Aerodynamic Powered Landing Problem is formulated in section 2 along with the structure of the optimal solution; section 3 outlines two possible approaches to solve the problem: the former, a fully indirect approach, is discussed in section 4; the hybrid method builds on results obtained in section 4 and is presented in section 5; results are drawn in this section too. Section 6 summarizes the work and hints to further research topics.

2. Aerodynamic Powered Landing Problem

2.1. Problem formulation

A landing site-topocentric cartesian reference frame is accounted for, of type *East-North-Up*, with $z-$axis directed upwards; for a flat Earth model, therefore, initial and final conditions $\boldsymbol{x}_0$, $\boldsymbol{x}_f \in R^7$ and gravity vector $\boldsymbol{g} \in R^3$ read

$$\boldsymbol{x}_0 = \begin{bmatrix} \boldsymbol{r}_0 \\ \cdots \\ \boldsymbol{v}_0 \\ \cdots \\ m_0 \end{bmatrix} = \begin{bmatrix} 1200 \\ 600 \\ 5000 \\ -10 \\ 0 \\ -300 \\ 1000 \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix}1\\1\\1\end{matrix}}\right\} m \\ \left.\vphantom{\begin{matrix}1\\1\\1\end{matrix}}\right\} m/s \\ kg \end{matrix}$$

$$\boldsymbol{x}_f = \begin{bmatrix} \boldsymbol{r}_f \\ \boldsymbol{v}_f \\ m_f \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \text{free} \end{bmatrix}$$

$$\boldsymbol{g} = \begin{bmatrix} 0 \\ 0 \\ -9.81 \end{bmatrix} m/s^2$$

This most general case corresponds to a 3-D trajectory, on which no cartesian dimensional reduction can be applied, as represented in the following figure.

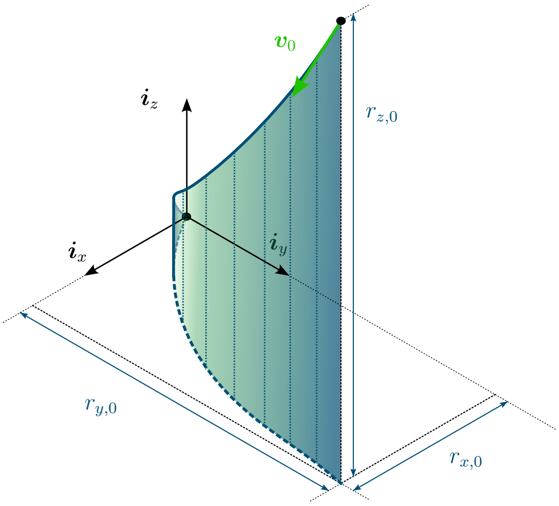

Figure 1: Candidate feasible trajectory

Mass depletion is modelled assuming constant specific impulse I_{sp} from the thruster; maximum thrust $T_{\max}$ is hypothesized not to depend on height. It is further useful introducing the thrust parameter $\hat{\alpha} \doteq (I_{\text{sp}} g_0)^{-1}$.

Thrust direction is identified by $\boldsymbol{u}_T \in R^3$, and thrust acceleration is therefore directed along such vector. $\boldsymbol{u}_T$ makes up for the first control. Aerodynamic controls are instead the *bank angle* σ and the *total angle of attack* α_{eff}. Such two angles identify, along with velocity versor $\boldsymbol{v}/v$, the *body axis direction* $\boldsymbol{u}_b \in R^3$. $u_b \doteq 1$, thus σ and α_{eff} are equivalent to $\boldsymbol{u}_b$.

Aerodynamic forces $\boldsymbol{D}, \boldsymbol{L} \in R^3$ read

$$\boldsymbol{D} = -\frac{1}{2}\,\rho(r_z)\,S_{ref}\,C_D\,(\alpha_{\text{eff}})\;v\;\boldsymbol{v}$$
$$\boldsymbol{L} \approx \frac{1}{2}\,\rho(r_z)\,S_{ref}\,C_{L/\alpha}\;[\boldsymbol{v} \otimes \boldsymbol{u}_b\,(\alpha_{\text{eff}}, \sigma)] \otimes \boldsymbol{v}$$

where *lift coefficient* C_L for an axisymmetric body, the *drag coefficient* C_D and variable atmospheric density ρ are modelled according to

$$C_L(\alpha_{\text{eff}}) = C_{L/\alpha}\alpha_{\text{eff}}$$
$$C_D(\alpha_{\text{eff}}) = C_{D,0} + kC_L^2(\alpha_{\text{eff}})$$
$$\rho\,(r_z) = \rho_0 e^{-r_z/H}$$

Dynamics right-hand side, therefore, reads

$$\begin{bmatrix} \boldsymbol{v} \\ \dfrac{T_{max}}{m}\boldsymbol{u}_T + \boldsymbol{g} + \dfrac{\boldsymbol{D}}{m} + \dfrac{\boldsymbol{L}}{m} \\ -\hat{\alpha} T_{max} u_T \end{bmatrix} = \boldsymbol{f}\,(\boldsymbol{x}, \boldsymbol{u}_T, \alpha_{\text{eff}}, \sigma)$$

Total angle of attack, defined as positive, is limited to $\alpha_{\max}$. Thrust authority is bounded as well, due to limits on mass depletion rates. Thus

$$\alpha_{\text{eff}} \leq \alpha_{\max} = 15^\circ$$
$$u_T \geq u_{T,\min} = 0.3$$
$$u_T \leq u_{T,\max} = 1$$

At last, aim is minimizing fuel consumption, or, equivalently, thrust acceleration integral [5]. The objective function reads then

$$\mathcal{J} = \int_0^{t_f} \frac{\hat{\alpha}}{m} T_{max} u_T \,\mathrm{d}t$$

The Aerodynamic Powered Landing Problem is formulated followingly

$$\min_{\alpha_{\text{eff}}, \sigma, \boldsymbol{u_T}, t_f} \mathcal{J} \quad \text{s.t.} \quad \begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f} \\ \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ \boldsymbol{x}(t_f) = \boldsymbol{x}_f \\ \alpha_{\text{eff}} \leq \alpha_{\max} \\ u_T \geq u_{T,\min} \\ u_T \leq u_{T,\max} \end{cases} \tag{1}$$

Thruster and aerodynamic parameters are summarized in table 1.

Parameter	Value	
I_{sp}	320	$[s]$
$T_{\max}$	40	$[kN]$
S_{ref}	0.44	$[m^2]$
ρ_0	1.225	$[{}^{kg}/{}_{m^3}]$
H	7200	$[m]$
$C_{D,0}$	0.5	[-]
k	2.5	[-]
$C_{L/\alpha}$	2.2	[-]

Table 1: Parameters summary

2.2. Optimal Control Structure

The optimal thrust magnitude u_T^* follows a *bang-bang* profile, while the optimal total angle of attack α_{eff}^* follows a *continuous-bang* profile. Optimal thrust direction and σ^*, instead, minimize projections respectively of thrust acceleration and of aerodynamic acceleration on a specific direction, defined by the *primer vector* $\boldsymbol{p}_v$. To better understand, the following digression contextualizes and builds on the cited concepts.

Primer vector is defined as $\boldsymbol{p}_v \doteq \boldsymbol{\lambda_v}/\lambda_v$, and $\boldsymbol{\lambda_v}$ can be in turn defined *dualizing* the original problem. Indeed, the OCP in eq. (1) can be transformed in a Two Point Boundary Value Problem (TPBVP): defining the system Hamiltonian $\mathcal{H} \doteq \mathcal{J} + \boldsymbol{\lambda}^{\text{T}}\boldsymbol{f}$, with vector $\boldsymbol{\lambda} \in R^7$ gathering the *costates*, the dynamics of $\boldsymbol{\lambda}$ obey

$$\dot{\boldsymbol{\lambda}} \doteq \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} \boldsymbol{\lambda}_r \\ \boldsymbol{\lambda}_v \\ \lambda_m \end{bmatrix} = -\begin{bmatrix} \nabla_{\boldsymbol{r}}\mathcal{H} \\ \nabla_{\boldsymbol{v}}\mathcal{H} \\ \nabla_m\mathcal{H} \end{bmatrix}$$

with the boundary $\boldsymbol{\lambda}_m = 0$, $\mathcal{H}_f = 0$. The optimal control minimizes, at each instant, $\mathcal{H}$: this condition, the Pontryagin's Minimum Principle (PMP), dictates the structure of the optimal control profiles.

Therefore, the statements at the beginning of this section are mathematized as follows. For the thrust control, PMP requires $\boldsymbol{i}_T^* \doteq \boldsymbol{u}_T/u_T = -\boldsymbol{p}_v$, and

$$u_T^* = \begin{cases} u_{T,\min} & \text{if} \quad S_f > 0 \\ u_{T,\max} & \text{if} \quad S_f < 0 \\ \in (u_{T,\min}, u_{T,\max}) & \text{if} \quad S_f = 0 \end{cases}$$

where the *switching function* S_f is defined as

$$S_f \doteq \frac{1}{m}\left(1 - \frac{\lambda_v}{\hat{\alpha}}\right) - \lambda_m$$

For the treated problem S_f is non-null almost everywhere. For the aerodynamic controls, the quantity to be minimized is $\boldsymbol{p}_v^{\text{T}}(\boldsymbol{D} + \boldsymbol{L})$; for unbounded σ (as in our case) optimal $\boldsymbol{L}^*$ belongs to the plane defined by $\boldsymbol{p}_v$ and $\boldsymbol{v}$; specifically, it is aligned and opposed to $\boldsymbol{p}_{v,\perp}$, the component of $\boldsymbol{p}_v$ orthogonal to $\boldsymbol{v}$. The procedure is graphically represented in fig. 2, where $\boldsymbol{F}_a$ denotes $\boldsymbol{F}_a \doteq \boldsymbol{D} + \boldsymbol{L}$ and the blue surface is the aerodynamic polar $\boldsymbol{F}_a$ is constrained to lie on.

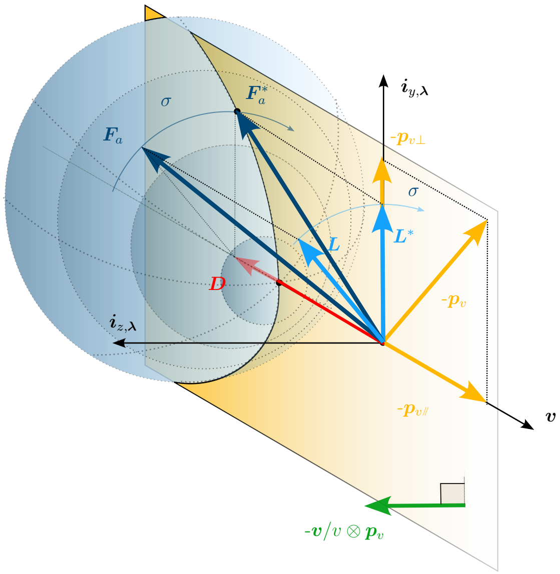
σ
$\boldsymbol{F}_a^*$
$\boldsymbol{i}_{y,\lambda}$
$\boldsymbol{F}_a$
-$\boldsymbol{p}_{v\perp}$
σ
$\boldsymbol{L}$
$\boldsymbol{L}^*$
-$\boldsymbol{p}_v$
$\boldsymbol{D}$
$\boldsymbol{i}_{z,\lambda}$
-$\boldsymbol{p}_{v/\!/}$
$\boldsymbol{v}$
-$\boldsymbol{v}/v \otimes \boldsymbol{p}_v$

Figure 2: Optimal $\boldsymbol{L}$ representation.

Optimal total angle of attack α_{eff}^* obeys instead, for the given aerodynamic polar, to the relation

$$\alpha_{\text{eff}}^* = \begin{cases} \dfrac{\tan\varsigma}{2kC_{L/\alpha}} & \text{if} \quad \tan\varsigma < 2kC_{L/\alpha}\alpha_{\max} \\ \alpha_{\max} & \text{else} \end{cases}$$

where ς defines the angle between $\boldsymbol{p}_v$ and $\boldsymbol{v}$. Optimal bank angle σ^* is imposed by finding the corresponding optimal $\boldsymbol{u}_b^*$. Rodrigues' formula is employed in such context: $\boldsymbol{v}/v$ is rotated around eigenaxis $-\boldsymbol{v}/v \otimes \boldsymbol{p}_v$ about angle α_{eff} to obtain $\boldsymbol{u}_b^*$.

3. Numerical Methods

It is now worth introducing the underlying idea behind the analyzed algorithms proposed in this work. Figure 3 provides a complete overview of the possible strategies to approach the solution of the OCP.

$$\begin{tikzcd}
B^{\lambda} \arrow[r, shift left, "\text{Discretize}"] & B^{\lambda N} \arrow[l, shift left, "\text{Convergence}"] \arrow[d, shift right] \arrow[l, bend right=60, blue] \\
& B^{N\lambda} \arrow[u, shift right, "\text{CMT}"'] \arrow[u, bend right=60, blue] \\
B \arrow[uu, "\text{Dualize}"] \arrow[uu, bend left=60, red] \arrow[r, shift left, "\text{Discretize}"] \arrow[r, bend right=60, blue] & B^{N} \arrow[l, shift left, "\text{Convergence}"] \arrow[u, "\text{Dualize}"'] \arrow[u, bend right=60, blue]
\end{tikzcd}$$

Figure 3: Summary of Optimal Control Problem formulations
In red: Pure indirect strategy
In blue: Hybrid strategy

B, stated in eq. (1), refers to the common problem formulation, and the superscripts indicate variants of it:

- B^{λ} is the TPBVP formulation, obtained with dualization, as described in section 2.2. It is the formulation tackled by *indirect methods*.
- B^{N} is the NLP formulation, *i.e.* the optimization problem obtained discretizing the domain with nodes, and computing states and controls at such points. It is the formulation approached by *direct methods.*
- $B^{N\lambda}$ is the zero-finding problem obtained imposing optimality conditions on problem B^{N}, and represents the final mathematical formulation tackled by common softwares, as the open-source convex solver ECOS. Optimal controls make part of the output of the problem.
- $B^{\lambda N}$ is the discrete-time version of B^{λ}, as nodes are introduced and optimality conditions are imposed at such points. Optimal controls are found applying the PMP on the output of the problem.

The optimality conditions of B actually define B^{λ}, thus solution of B^{λ} coincides with the one of B. The most direct path to solve B^{λ}, namely $B \rightarrow B^{\lambda}$, is highlighted in red in fig. 3; initialization of costates and dynamical system ill-conditioning constitute relevant problems: an algorithm following this path is used to verify how the general control structure in section 2.2 specializes to our case, but additional strategies shall be used to circumvent the aforementioned setbacks.

An alternative is represented by the longest path, $B \rightarrow B^{N} \rightarrow B^{N\lambda} \rightarrow B^{\lambda N} \rightarrow B^{\lambda}$: it exploits the mapping, called Covector Mapping Theorem, between the multipliers of the NLP and the costatates of the TPBVP; the process is represented by the blue path in fig. 3. Such approach is followed by the proposed hybrid method, where B^{N} is the convex problem form, $B^{N\lambda}$ is solved using ECOS and $B^{\lambda N}$ is tackled by the indirect collocation scheme.

4. Pure indirect strategy

Solution of the presented problem is first evaluated using a single shooting technique: $\boldsymbol{\lambda}_0$ and t_f are tuned such that the flow of $\boldsymbol{x}_0$ and $\boldsymbol{\lambda}_0$ through the Hamiltonian dynamics $\boldsymbol{\varphi}$ satisfies the final boundary conditions over states, costate λ_m and Hamiltonian $\mathcal{H}$. However, the optimal u_T is discontinuous; this results in a small basin of convergence for the original zero-finding problem. Two homotopic continuation schemes are therefore adopted in series:

- The first is applied on the objective function: it generalizes $\mathcal{J}$ with $\mathcal{J}_{\varepsilon}$, an objective function including a quadratic term in u_T; the continuation scheme starts from the purely quadratic objective function ($\varepsilon = \varepsilon_{EO} \doteq 0$) and reaches the purely linear one ($\varepsilon = \varepsilon_{AO} \doteq 1$), corresponding to the original problem. Aerodynamic forces are discarded from dynamics. The first step is initialized using Random Number Generation (RNG).
- The second is applied on the aerodynamic forces, generalizing $\boldsymbol{f}$ with $\boldsymbol{f}_{\bar{S}_{ref}}$. A weighting factor $\bar{S}_{ref} \in [0,1]$ is employed in $\boldsymbol{f}_{\bar{S}_{ref}}$ to scale aerodynamic forces linearly; $\bar{S}_{ref} = 0$ and $\bar{S}_{ref} = 1$ correspond respectively to the cases with no aerodynamic forces and with full aerodynamic forces.

Optimal control profiles for the first homo-

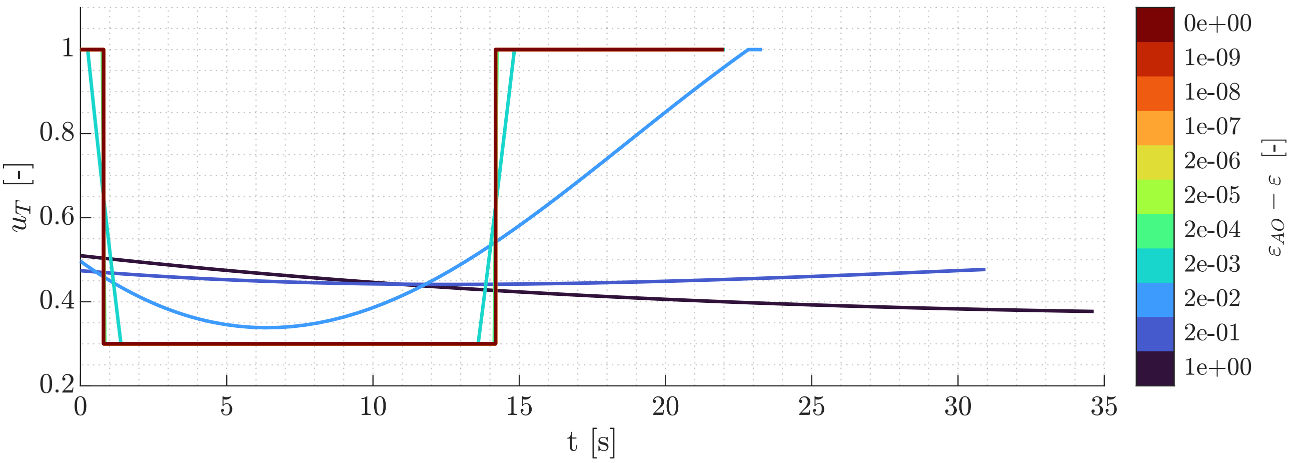

Figure 4: $\mathcal{J}_\varepsilon$ homotopic continuation scheme for u_T profile

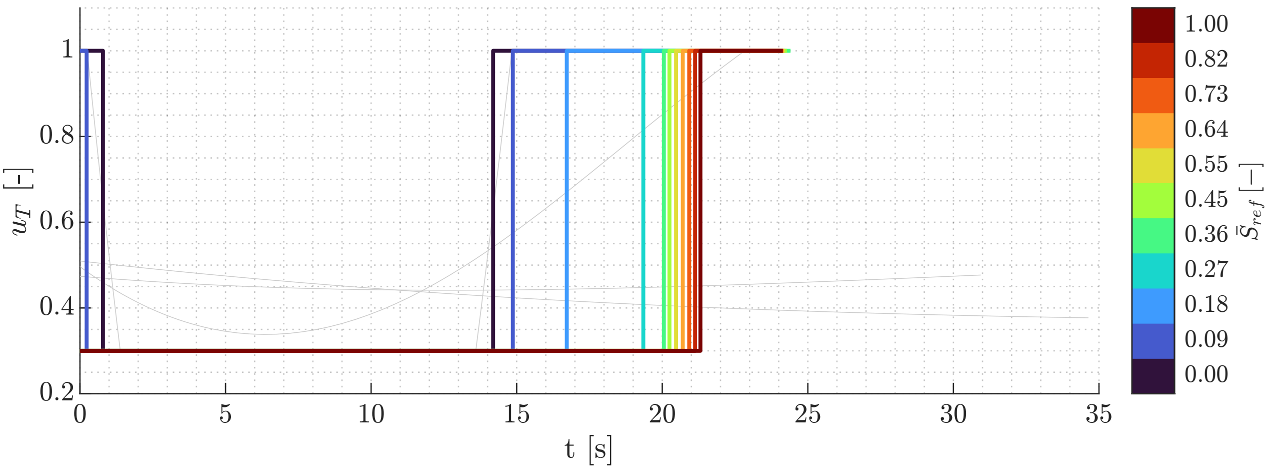

Figure 5: $\boldsymbol{f}_{\bar{S}_{ref}}$ homotopic continuation scheme for u_T profile

topic loop are represented in fig. 4, while the results for the aerodynamics continuation are represented in fig. 5. Variation of ε shifts the control profile from a $\mathcal{C}^\infty$ control profile to a *bang-off-bang* discontinuous profile; varying instead $\bar{S}_{ref}$, the profile degenerates in an *off-bang* type. The decreasing duration of the thrusting arcs through the iterations further hints to a decreasing fuel consumption. Indeed, as reported in table 2, addition of aerodynamic forces allows propellant savings amounting approximately to the 4% of the initial rocket mass.

Dynamics	u_T profile	m_f [kg]
No $\boldsymbol{D}, \boldsymbol{L}$	*Bang-idle-bang*	839.1
Full $\boldsymbol{D}, \boldsymbol{L}$	*Idle-Bang*	882.5

Table 2: Main outcomes from indirect method

5. Hybrid strategy

5.1. Algorithm overview

With respect to the pure indirect strategy, the hybrid one cuts out the homotopic continuation substituting it with the convex step. The Covector Mapping Theorem employed to link the direct and indirect steps stems from the employed LGR transcription and from the number of collocation points, *i.e.* the discrete instants at which dynamics is reinforced. For n collocation points, the CMT reads

$$\begin{aligned} \tilde{\boldsymbol{\lambda}}_f &= \frac{2}{t_f}\mathbf{D}_n^{\mathrm{T}}\tilde{\boldsymbol{\Lambda}} \\ \tilde{\boldsymbol{\lambda}}_i &= \frac{\tilde{\boldsymbol{\Lambda}}_i}{\tilde{w}_i} \qquad i = 0, \dots, n-1 \end{aligned}$$

where $\tilde{\boldsymbol{\Lambda}}$ are the multipliers at the collocation points. In addition $\tilde{w}_i$, with $i = 0, \dots, n-1$, is the i^{th} nodal quadrature weight w_i scaled with t_f according to $\tilde{w}_i = t_f w_i/2$. $\tilde{\boldsymbol{X}}$ gathers the states at nodes; $\mathbf{D}$ is the pseudospectral differentiation matrix approximating dynamics at each i^{th} collocation point according to $\dot{\tilde{\boldsymbol{X}}}_i \approx \dfrac{2}{t_f}\mathbf{D}_i\tilde{\boldsymbol{X}}$; finally $\mathbf{D}_n^{\mathrm{T}}$ is last row of $\mathbf{D}^{\mathrm{T}}$. The marker $\tilde{}$ refers to a non physical parameterization of states used within the direct solver; prior to feeding the indirect solver $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{\lambda}}$ are then mapped to the physical domain. The overall pipeline is represented in fig. 6.

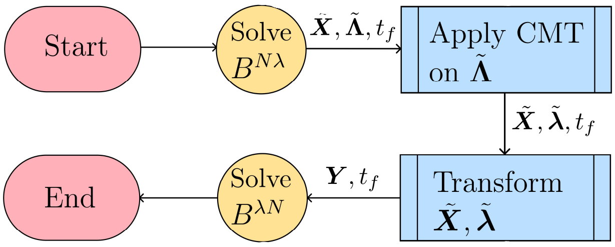
Start
Solve $B^{N\lambda}$
$\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{\Lambda}}, t_f$
Apply CMT on $\tilde{\boldsymbol{\Lambda}}$
$\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{\lambda}}, t_f$
Transform $\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{\lambda}}$
$\boldsymbol{Y}, t_f$
Solve $B^{\lambda N}$
End

Figure 6: Algorithm simplified pipeline

The **convex step** ($B^{N\lambda}$) merges two convexification techniques, a *lossless convexification* (LCvx) that handles non-convex thrust magnitude constraints and *successive convexifications* (SCvx) that address aerodynamic and free final time contributions. Lift is here explicitly formulated in terms of σ rather than $\boldsymbol{u}_b$ to avoid the non-convex constraint $\boldsymbol{u}_b = 1$.

- the LCvx builds on [5] and, for fixed final time and ignoring aerodynamic forces, provides a problem that convex solvers handle with no approximations. Its working principle is based on 1) a change of variables (hence the need for $\tilde{}$), 2) a formulation of the fuel optimal landing problem as acceleration optimal and 3) a relaxation of the non-convex thrust magnitude constraint.
- the SCvx technique comes into play as aerodynamic and final time contributions are linearized sequentially: the convex optimization problem is therefore solved multiple times, and requires initialization. Moreover, *trust regions* ζ_{t_f} and $\zeta_{\alpha_{\text{eff}}}$ constrain variations of t_f and α_{eff} at each iteration, according to

  $$|\Delta t_f| \le \zeta_{t_f}$$
  $$|\Delta \alpha_{\text{eff}}| \le \zeta_{\alpha_{\text{eff}}}$$

  This approach limits linearization inaccuracy errors, thus improving algorithm robustness.

Therefore the convex loop linearizes the original problem and solves the resulting subproblem sequentially and iteratively; such procedure is iterated as long as Δt_f satisfies $|\Delta t_f| > \texttt{tol}$. At last, nonlinear contributions are added to the solved problem one after the other, thus grant algorithm convergence: the pure LCvx form is solved first, guessing fixed final time t_f from a 1-D model; free final time contribution is added afterwards, then drag and, finally, lift. The procedure is reported in fig. 7, where $\{1\}$ denotes the described SCvx + LCvx problem, the superscript $\bar{k}$ conventionally indicates solution at convergence and $\{1\}$ denotes the problem $BN\lambda$

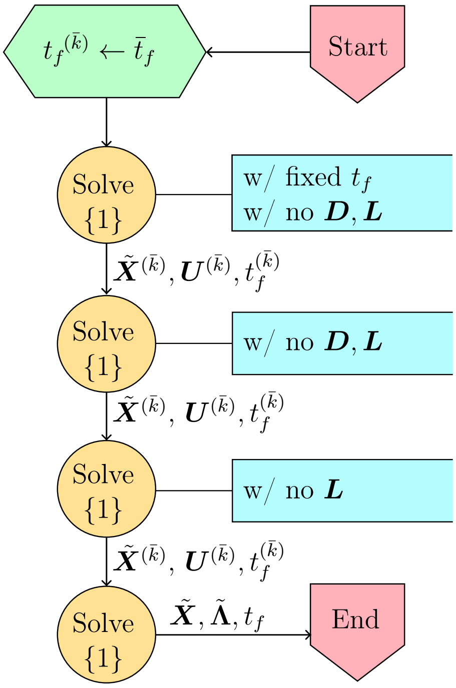
$t_f^{(\bar{k})} \leftarrow \bar{t}_f$
Start
Solve $\{1\}$
w/ fixed t_f
w/ no $\boldsymbol{D}, \boldsymbol{L}$
$\tilde{\boldsymbol{X}}^{(\bar{k})}, \boldsymbol{U}^{(\bar{k})}, t_f^{(\bar{k})}$
Solve $\{1\}$
w/ no $\boldsymbol{D}, \boldsymbol{L}$
$\tilde{\boldsymbol{X}}^{(\bar{k})}, \boldsymbol{U}^{(\bar{k})}, t_f^{(\bar{k})}$
Solve $\{1\}$
w/ no $\boldsymbol{L}$
$\tilde{\boldsymbol{X}}^{(\bar{k})}, \boldsymbol{U}^{(\bar{k})}, t_f^{(\bar{k})}$
Solve $\{1\}$
$\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{\Lambda}}, t_f$
End

Figure 7: $B^{N\lambda}$ – Sequential problems pipeline

The **indirect step** ($B^{\lambda N}$) is a collocation scheme based on an r-method; switching time t_{sw} is included in the unknowns and the closure condition for it is provided by the constraint

$$S_f(t_{\text{sw}}) = 0$$

Since optimal thrust profile requires one switch only, two segments are sufficient to discretize the domain. Comass, mass and covelocities are not $\mathcal{C}^1$ due to the *bang* in the control; however, separate polynomials are employed to approximate states and costates on the two subarcs;

non-regularities are handled in this way.

To initialize the mesh, the switching function S_f is evaluated at each node and interpolated along $[0, t_f]$. The obtained estimation $\bar{S}_f$ is then inverted to find t_{sw} satisfying $\bar{S}_f(t_{\text{sw}}) = 0$. Domain is then remeshed and the mesh structure is hold. The zero-finding problem is handled via sequential linearizations, using analytical Jacobians and a fixed trust region $\zeta_{\tilde{\boldsymbol{Y}}}$ acting on each component of $\tilde{\boldsymbol{Y}} \doteq \left[\boldsymbol{X}^{\mathrm{T}}, \boldsymbol{\lambda}^{\mathrm{T}}, t_{\text{sw}}, t_f\right]^{\mathrm{T}}$. Jacobian is further normalized at each iteration step to decrease problem conditioning numbers. The generic problem solved within the indirect step at k^{th} step is

$$\tilde{\mathbf{A}}^{(k)}_{\text{lin}} \Delta \tilde{\boldsymbol{Y}}^{(k)} = \tilde{\mathbf{b}}^{(k)}_{\text{lin}} \begin{cases} \tilde{\mathbf{A}}^{(k)}_{\text{lin}} \doteq \boldsymbol{K}_f \mathbf{A}^{(k)}_{\text{lin}} \\ \tilde{\mathbf{b}}^{(k)}_{\text{lin}} \doteq \boldsymbol{K}_f \mathbf{b}^{(k)}_{\text{lin}} \\ \boldsymbol{K}_{f_{i,j}} \doteq \begin{cases} 0 & \text{if} \quad i \neq j \\ \dfrac{1}{\left\|\mathbf{A}^{(k)}_{\text{lin},i}\right\|_2} & \text{else} \end{cases} \end{cases}$$

and

$$\mathbf{A}^{(k)}_{\text{lin}} = \left[\begin{array}{c:c} \mathbf{A}^{(k)}_{\text{dyn},\boldsymbol{y}} & \mathbf{A}^{(k)}_{\text{dyn},t} \\ \hdashline \mathbf{A}_{knot} & \\ \mathbf{A}_{BCs} & \\ \mathbf{A}^{(k)}_{\tilde{\mathcal{H}}_f} & \mathbf{0}_{(4n_s+2)\,\mathrm{x}\,2} \\ \mathbf{A}^{(k)}_{S_f} & \end{array}\right]$$

The different contributions within Jacobian $\mathbf{A}^{(k)}_{\text{lin}}$ mirror the different constraints.

Controls are finally extracted and accuracy of the final solution assessed using a Zero-Order-Hold for thrust magnitude and cubic approximations for the remaining controls.

5.2. Final results

Evaluation of performances of the direct step shall take into account computational time, as well as accuracy of costates estimation. As far as the former is considered, computational time have been timed for different grid configurations, and are reported in table 3; the configuration with 5 segments and 5 collocation points per segment (CPpS) has been selected as nominal: the maximum percentual deviation of λ_m with respect to the configuration with 20 nodes and 10 segments amounts to 5% only. Improvement of such parameter does not pay off the sensible increases in CPU time.

Table 3: Computational times[†] for different combinations of segments and collocation points per segment (CPpS)

			Segments			
			1	2	5	10
CPpS	5	CPU time[†][s]	-	0.4	1.1	3.0
	10		0.4	1.3	-	11.4
	20		1.2	3.9	13.9	43.4

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

For what concerns accuracy, instead, results are shown in fig. 8 for the 5 segments-5 CPpS configuration: $\boldsymbol{\lambda}^*$ is retrieved from the pure indirect solution. It is evident the orders of magnitude are correctly grasped for covelocities and comass; as well, order of magnitude is correctly estimated for λ_{r_z}, while errors grow sensibly for λ_{r_x} and λ_{r_y}. This setback, however, is of minor importance in the economy of the complete algorithm: $\partial\mathcal{H}/\partial r_x = \partial\mathcal{H}/\partial r_y = 0$, thus λ_{r_x} and λ_{r_y} can be corrected in a single iteration within the indirect algorithm.

Considering the indirect step, instead, 40 collocation points per segment offer a fair trade-off between computational times and error over t_f and t_{sw}; for the same sets of collocation points an accuracy analysis over the final condition is carried out: controls are retrieved using the PMP and rocket dynamics is simulated. Results are reported in table 4: absolute error over final touchdown position e_{r_f} is computed at the final instant of simulation; the corresponding touchdown velocity v_{TD} is reported as well; this is evaluated during the simulation, within the integrator when the condition $r_z = 0\ m$ is detected. It does not correspond, therefore, to the final velocity evaluated by the solver. Results show maximum accuracy slightly higher than 1 m over final landing site position. Vertical touchdown velocity levels down at 9 m/s, corresponding to the 3% of the initial vertical velocity; the need for online optimization, even ignoring unmodelled dynamics, is evident. Yet, performances offered employing 40 collocation points per segment are extremely promising: it is the error over final position that determines

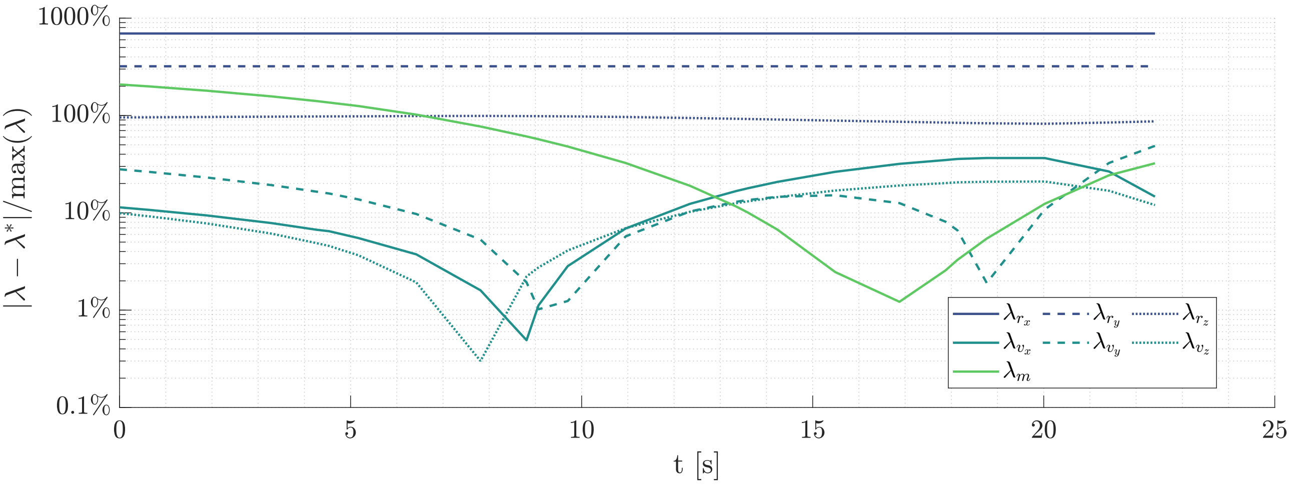

Figure 8: Summary of relative errors in costates estimated with the direct convex method

non null touchdown velocity, and its error is in the order of 0.2‰ of the initial height. Moreover, the proposed approach overcomes the pure indirect approach by far when it comes to computational times. Despite the latter being penalized by non-tailored functions from MATLAB, the gap is significant: the proposed architecture handles the same problem with much lower effort, as shown in table 5. The optimal trajectory is reported fig. 9: full thrust and idle arcs are highlighted; aerodynamic forces are projected, as well as trajectory, on the planes $x-y$ and $y-z$. The orientation of aerodynamic forces confirms their capability do damp kinetic energy while deflecting the trajectory: at all points $\boldsymbol{a}_{\text{aero}}$ features a component opposed to velocity and a centripetal one.

Table 4: Overview of touchdown position accuracy and relative touchdown velocities

		$e_{r_{f,z}}$ [m]	v_{TD} [m/s]
CPpS	10	13.5	30.0
	20	8.1	22.1
	40	1.2	8.6
	80	1.2	8.6

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

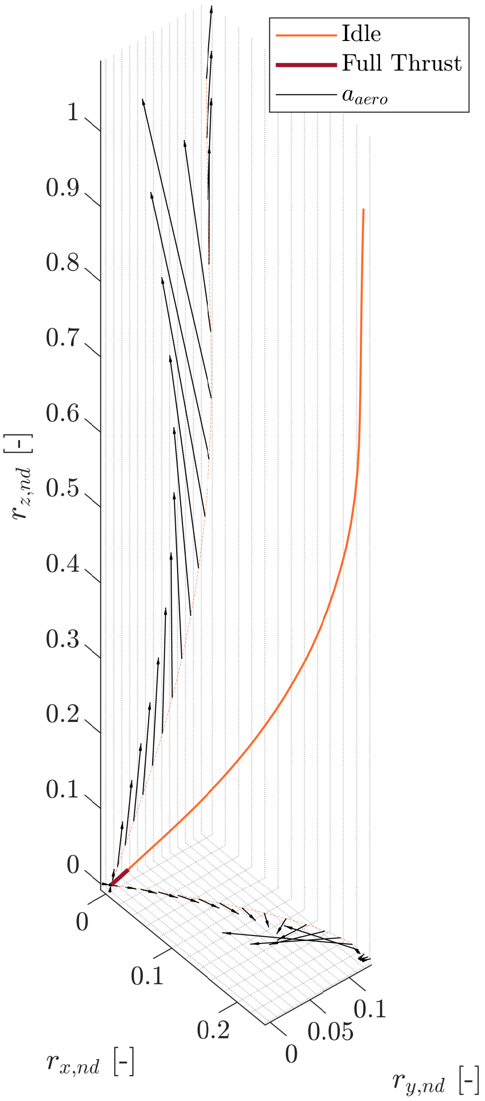

Figure 9: Optimal landing with aerodynamic forces and thrust magnitude

Table 5: Comparison of computational times between purely indirect homotopic approach and hybrid technique

		CPU Time†$[s]$
Indirect homotopy	$\bar{\mathcal{J}}_\varepsilon$ homotopy	256.7
	$\boldsymbol{f}_{S_{ref}}$ homotopy	124.9
Hybrid method	Direct step	1.1708
	CMT	$6\ 10^{-5}$
	$\tilde{\boldsymbol{\lambda}}$ mapping	$5\ 10^{-4}$
	Indirect step	4.7608

† Relative to Dell XPS w/ 2.6 GHz Intel Core i7, 16 GB 2666 MHz DDR4

6. Conclusions

A hybrid algorithm to assess solution of a bang-bang problem with accuracy and limited computational time has been designed and tested. Method has demonstrated net supremacy over a traditional approach. The indirect collocation scheme has allowed to reduce sensibly the time required to solve the First Order optimality necessary conditions; the convex step has provided, within limited time, a guess accurate enough to allow the indirect solver to converge. Future directions include development of a first strategy to constrain u_T pointing direction and of a second one to limit the relative angle between $\boldsymbol{u}_b$ and $\boldsymbol{u}_T$, mirroring a finite TVC capability.

References

[1] Fahroo, F. and Ross, I. M. “Costate Estimation by a Legendre Pseudospectral Method”. In: *Journal of Guidance, Control and Dynamics* 24.3 (Apr. 2001). DOI: 10.2514/2.4709.

[2] Blackmore, L. “Autonomous precision landing of space rockets”. In: *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium*. Vol. 46. Washington, DC: The National Academies Press, 2017. DOI: 10.17226/23659.

[3] Sagliano, M. “Generalized *hp* Pseudospectral-Convex Programming for Powered Descent and Landing”. In: *Journal of Guidance, Control and Dynamics* 42.7 (July 2019). DOI: 10.2514/1.G003731.

[4] Hofmann, C. and Topputo, F. “Rapid Low-Thrust Trajectory Optimization in Deep Space Based on Convex Programming”. In: *Journal of Guidance, Control and Dynamics* 44.7 (July 2021). DOI: 10.2514/1.G005839.

[5] Açıkmeşe, B. and Ploen, S. R. “Convex Programming Approach to Powered Descent Guidance for Mars Landing”. In: *Journal of Guidance, Control and Dynamics* 30.5 (Oct. 2007). DOI: 10.2514/1.27553.