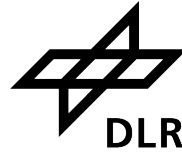




POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE



DLR

**Institute of
Space Systems**

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

Master of Science in Space Engineering

Author: **Fabio Spada**

Student ID: 967678

Advisor: Prof. Francesco Topputo

Co-advisor: Dr. Marco Sagliano

Academic Year: 2021-2022

Copyright © Fabio Spada October 2022
All Rights Reserved

This content is original, written by the Author, Fabio Spada. All non-original information, taken from previous works, is specified and recorded in the Bibliography.

When referring to this work, full bibliographic details must be given, i.e.:

F. Spada, “Hybrid Direct-Indirect Strategy for Optimal Terminal Guidance of Reusable Rockets”, 2022, M.Sc.Thesis, Politecnico di Milano, Faculty of Industrial and Information Engineering, Department of Aerospace Science and Technology, Degree in Space Engineering. Supervisor: F. Topputo, Co-supervisor: M. Sagliano.

*Cvava sero po tute
i kerava
jek sano ot mori
i taha jek jak kon kasta
vasu ti baro nebo
avi ker*

*Poserò la testa sulla tua spalla
e farò
un sogno di mare
e domani un fuoco di legna
perché l'aria azzurra
diventi casa*

*I'll lay my head on your shoulder
and I'll dream
a dream of sea
and tomorrow a fire of woods
so the blue air
will become home*

*Kon ovla so mutavia
kon ovla
ovla kon ascovi
me gava palan ladi
me gava
palan bura ot croiuti*

*Chi sarà a raccontare
chi sarà
sarà chi rimane
io seguirò questo migrare
seguirò
questa corrente di ali*

*Who will narrate
who will be
it will be who's gonna stay
I'll follow this migration
I'll follow
this stream of wings*

*Fabrizio De André, Ivano Fossati
Khorakhané (A forza di essere vento)*

A Bruna e Tonia

Acknowledgements

Six months have passed since I have arrived in Bremen, and the experiences I have lived have definitely been worth this 'jump'; if I have had such opportunity I shall thank Professor Francesco Topputo, the 'conductor'; after years of lessons, your advice and attention keep on being sources of inspiration.

A second huge thanks goes to my supervisor at DLR, Marco, my tenet: your passion and commitment have continuously pushed my curiosity and desire to learn more; thanks for the suggestions, and for having handed me the 'red pill'.

I then would like to thank the staff of the Navigation and Control department of the Institute of Space Systems, you have hosted me and made my workplace feel like home: your laughter, *die Mittagessen*, the talks; they all make part of the beautiful experiences I will carry with me.

Alfredo, Akrom, Francesco, Jennifer, José, Paria, Sahana, Sara, Stefano. It is thanks to you that I have enjoyed my weeks in Bremen; thanks for the beautiful memories I will carry with me about you and the time spent together.

Ciro, Giorgia, Greta, Jibril, Mattia, Michele, Pietro, Rick, Silvia, Staffo. Grazie per avermi mostrato come il mondo non è né bianco, né nero, ma che esistono molte sfumature, cangianti. La vostra presenza mi ha arricchito; con voi, più e più volte, mi sono riscoperto.

Cospito, Omar, Rizzi, Roberto. Siete entrati in punta di piedi e mi avete accompagnato nella mia crescita. Grazie per le pallonate e per gli schiaffi sul 'cuzzetto'; grazie per avermi dato degli splendidi esempi di amicizia.

Giulio e Mario, mi avete sempre stimolato a cercare novità e cambiamenti, avete accolto le mie idee più strampalate ed io ho fatto tesoro delle vostre; grazie.

Eliana, grazie per le discussioni, l'obiettività; grazie per mostrarmi ogni giorno come si prende la vita con leggerezza, per essere la mia 'socio a delinquere'.

Lucia ed Elio, i miei pilastri; grazie per l'ascolto e gli spunti di riflessione che mi date ogni giorno.

At last, I would like to thank the people I have not explicitly included, but that are part of my life, either nowadays, either within my memories. You've contributed to making me the person I am today. A last heartfelt thanks is for you.

Bremen, 13 September 2022

Abstract

The present work focuses on developing a fast and accurate algorithm to find feasible trajectories for reusable rockets' landing while minimizing their fuel consumption. Trajectory exploits aerodynamic forces, thus an Optimal Aerodynamic Powered Landing Problem is faced. A hybrid strategy is adopted, combining convex direct optimization with a novel indirect collocation scheme. A Covector Mapping Theorem is exploited to bridge the two methods. Development of the algorithm is organized in two steps: firstly, the structure of the optimal solution is derived solving the problem with a single shooting indirect method combined with a double homotopic continuation scheme; in second instance, an algorithm tailored on the optimal solution structure is presented and discussed. The suggested strategy is finally compared with the homotopic continuation scheme considering accuracy and computational times. Outcome is a net superiority of the designed algorithm over the homotopic technique; the power of a hybrid approach is therefore demonstrated over traditional solution methods.

Keywords: Hybrid Methods, Indirect Collocation, Convex Optimization, Rocket Trajectories, Aerodynamic Forces, Powered Landing

Sommario


Lo studio presentato in questa tesi si focalizza sullo sviluppo di un algoritmo rapido ed accurato per ottimizzazione di traiettorie; l'obiettivo è di individuare traiettorie di atterraggio per lanciatori riutilizzabili, compatibili con i vincoli di missione e che minimizzino il consumo di propellente. Le forze aerodinamiche sono sfruttate nell'ottimizzazione: il problema affrontato è quindi quello dell'Atterraggio Aerodinamico e Propulso Ottimale. Tecniche di ottimizzazione diretta convessa sono combinate con un innovativo schema indiretto di collocazione, delineando così una strategia di risoluzione ibrida; un Teorema di Mappatura dei Covettori è utilizzato da ponte di collegamento fra i due tipi di tecniche. Per lo sviluppo dell'algoritmo si sono resi necessari due passaggi: in prima istanza si è determinata la struttura della traiettoria ottimale usando uno *shooting* indiretto combinato con un doppio schema di continuazione omotopica; l'algoritmo definitivo è quindi 'cucito' sulla soluzione precedentemente trovata, così da poter gestire problemi con peculiarità simili. A conclusione, la velocità della strategia proposta è confrontata con quella dello schema di continuazione, mantenendo in considerazione l'accuratezza della soluzione trovata. I risultati mostrano una netta superiorità della strategia proposta rispetto alla tecnica di continuazione omotopica; la potenza di un approccio ibrido rispetto ad i tradizionali è quindi dimostrata.

Parole chiave: Metodi Ibridi, Collocazione Indiretta, Ottimizzazione Convessa, Traiettorie di Lanciatori, Forze Aerodinamiche, Atterraggio Propulso

Contents

Acknowledgements	v
Abstract	vii
Sommario	ix
Contents	xi
List of Figures	xv
List of Tables	xvii
List of Symbols	xix
1 Introduction	1
1.1 Historical framing	1
1.1.1 The forerunners of vertical landing	1
1.1.2 The <i>Faster, Better, Cheaper</i> concept	2
1.2 Algorithms state of the art	4
1.2.1 Offline optimization	4
1.2.2 Online optimization	5
1.3 Research contribution	5
1.4 Thesis structure	6
2 Mathematical fundamentals	9
2.1 Parametric Optimization	10
2.1.1 Nonlinear Programming	10
2.1.2 Convex Programming	11
2.1.3 Second-Order Cone Programming	13
2.2 Boundary Value Problem	14
2.2.1 One-dimensional case	14
2.2.2 Hamiltonian systems	15

2.3	Optimal Control Problem	15
2.3.1	Problem Statement	15
2.3.2	OCP as BVP - Optimality Necessary Conditions	17
2.3.3	OCP as NLP - Optimality Necessary conditions	21
3	Numerical methods	25
3.1	Traditional methods	27
3.1.1	Time-marching and Collocation techniques	27
3.1.2	Indirect and direct methods	28
3.2	Pseudospectral methods	29
3.2.1	Method overview	29
3.2.2	Survey of Legendre-type nodes	32
3.2.3	Radau direct collocation and CMT	33
3.2.4	Radau indirect collocation	38
3.3	OCP formulations wrap-up	40
4	Optimal Aerodynamic Powered Landing Problem	43
4.1	Problem formulation	44
4.2	Solution derivation	48
4.2.1	Optimization of thrust control	48
4.2.2	Optimization of aerodynamic controls	49
4.2.3	Dual problem formulation	51
4.3	Double homotopic scheme	52
4.4	Solution structure validation	55
4.4.1	Objective function homotopic scheme	55
4.4.2	Dynamics homotopic scheme	56
4.4.3	Summary	57
5	Hybrid Pseudospectral Algorithm	63
5.1	Direct convex collocation scheme	64
5.1.1	Direct step overview	69
5.2	Indirect collocation scheme	69
5.2.1	Indirect step overview	73
5.3	Final Results	74
6	Conclusions and Future Developments	85
6.1	Future developments	86

Contents	 xiii
Appendices	xxi
A Optimality Conditions with Path Constraints	xxi
A.1 Mixed constraints - Erdmann conditions	xxi
A.2 DAE index reduction	xxiii
A.2.1 Pure state constraints - Jump conditions	xxiii
B Optimal Thrust Program for Quadratic Objective Function	xxv
C Parameterization of Lift Versor in v and σ	xxvii
Bibliography	xxix

List of Figures

1.1	Overview of vertical landing vehicles - <i>From the left:</i> (a) DC-X performs first Earth vertical landing (b) Xombie completes large divert maneuver (c) New Shepard for the first time lands vertically after having reached space (d) First reused SpaceX Falcon 9 first stage lands after second successful mission (e) Co-simulation on Simulink-FlightGear of CALLISTO RTLS mission [10]	3
2.1	Second order cone constraint in \mathbb{R}^3 : $\ \mathbf{y}\ _2 \leq t$	14
3.1	Visual representation of Legendre-type nodes - 9 collocation points . .	34
3.2	Summary of Optimal Control Problem formulations	41
4.1	Trajectory not reducible to cartesian 2-D analysis	45
4.2	Generic \mathbf{L} and optimal \mathbf{L}^* associated with the same \mathbf{D} and obtained for different σ . Generic aerodynamic force \mathbf{F}_a and optimal one \mathbf{F}_a^* move on the paraboloid associated with the aerodynamic polar, in blue. \mathbf{L}^* minimizes $\mathbf{p}_v^T \mathbf{F}_a^*$	51
4.3	Double homotopy structure	54
4.4	\mathcal{J}_ε homotopic continuation scheme for control profile	55
4.5	\mathcal{J}_ε homotopic continuation scheme for optimal non \mathcal{C}^1 quantities . . .	57
4.6	$\mathbf{f}_{\bar{S}_{ref}}$ homotopic continuation scheme for thrust magnitude profile . .	58
4.7	$\mathbf{f}_{\bar{S}_{ref}}$ homotopic continuation scheme for α_{eff} profile	58
4.8	$\mathbf{f}_{\bar{S}_{ref}}$ homotopic continuation scheme for mass $m(t)$	59
4.9	$\mathbf{f}_{\bar{S}_{ref}}$ homotopic continuation scheme for trajectory	60
4.10	Comparison of thrust components between presented dynamic models	61
5.1	Complete hybrid algorithm structure	64
5.2	Thrust magnitude convexification	65
5.3	Direct step <i>zoom-in</i> , from Fig. 5.1 <i>On the left:</i> Solver structure <i>On the right:</i> Solver loop block - $\{1\}$	70

5.4	Remeshing procedure representation – \bar{k} is last iteration of the direct step; relative mesh on top. Remeshed domain for the first iteration of indirect step on bottom	72
5.5	Indirect step <i>zoom-in</i> , from Fig. 5.1	75
5.6	Final mass and final time iterations of direct step, with 5 segments and 5 collocation points per segment. <i>Iteration 1</i> : LCvx formulation; <i>Iterations 2-3</i> : Free t_f added; <i>Iterations 4-15</i> : \mathbf{D} added; <i>Iterations 16-24</i> : \mathbf{L} added	77
5.7	Optimal trajectory for progressive direct iterations	78
5.8	u_T for progressive direct iterations	78
5.9	α_{eff} for progressive direct iterations	79
5.10	Summary of relative errors in costates estimated with the direct convex method	79
5.11	Overview of touchdown position accuracy and relative touchdown velocities	81
5.12	Optimal landing with aerodynamic forces and thrust magnitude – Trajectory is projected on $x - y$ and $y - z$ planes along with aerodynamic forces	83
B.1	Cases for the application of the PMP to the quadratic formulation <i>Dark green</i> : $u_T^{(th)} < u_{T,\min}$ <i>Light blue</i> : $u_{T,\min} < u_T^{(th)} < u_{T,\max}$ <i>Violet</i> : $u_T^{(th)} > u_{T,\max}$	xxvi

List of Tables

4.1	Parameters summary	47
4.2	Results summary	59
5.1	Normalized error over estimated λ_m and computational times for different combinations of segments and collocation points per segment .	76
5.2	Indirect step performances for different collocation points per segment	80
5.3	Comparison of computational times between fully indirect homotopic approach and hybrid technique	82

List of Symbols

Mathematical symbols

τ	Pseudotime
\boldsymbol{x}	State vector
\boldsymbol{u}	Control vector
$\boldsymbol{\lambda}$	Costates vector
\boldsymbol{y}	Augmented state
\boldsymbol{f}	Dynamics right-hand side
\boldsymbol{X}	Discrete state
\boldsymbol{U}	Discrete control
Λ	Lagrange multiplier
\boldsymbol{Y}	Augmented discrete state
\boldsymbol{F}	Augmented right-hand side
\mathcal{J}	Objective function
φ	Hamiltonian system flow
\mathcal{H}	System Hamiltonian
w	Non-dimensional quadrature weights
\boldsymbol{D}	Pseudospectral differentiation matrix

Problem-specific symbols

t	Time	[s]
\boldsymbol{r}	Position	[m]
\boldsymbol{v}	Velocity	[m/s]
m	Mass	[kg]



\mathbf{u}_T	Thrust versor	$[-]$
T_{\max}	Maximum Thrust	$[N]$
I_{sp}	Engine specific impulse	$[s]$
g_0	Sea-level gravity acceleration	$[m/s^2]$
S_f	Switching function	$[-]$
\mathbf{D}	Drag	$[N]$
\mathbf{L}	Lift	$[N]$
C_L	Lift coefficient	$[-]$
C_D	Drag coefficient	$[-]$
$C_{D,0}, k, C_{L/\alpha}$	Further aerodynamic coefficients	$[-]$
ρ_0	Sea-level air density	$[kg/m^3]$
H	Atmospheric reference height	$[m]$
α_{eff}	Total angle of attack	$[^\circ]$
σ	Bank angle	$[^\circ]$
\mathbf{u}_b	Rocket body x-axis	$[-]$
ε	Objective function homotopic parameter	$[-]$
\bar{S}_{ref}	Reference surface homotopic parameter	$[-]$
Γ	Non-dimensional thrust amplitude slack	$[-]$
z	Logarithmic mass	$[kg]$
\mathbf{u}_a	Acceleration vector	$[m/s^2]$
ξ	Acceleration amplitude slack	$[m/s^2]$

Operators

$(\cdot) \preceq (\bullet)$	Component-wise inequality	-
$d(\cdot)$	First variation	$[\cdot]$
$\delta(\cdot)$	Virtual variation	$[\cdot]$
$\Delta(\cdot)$	Finite variation	$[\cdot]$
$\nabla_{(\bullet)}(\cdot)$	Gradient with respect to (\bullet) components	$[\cdot/\bullet]$
$\mathbf{J}_{(\bullet)}(\cdot)$	Jacobian with respect to (\bullet) components	$[\cdot/\bullet]$

1 | Introduction

Turns out the procrastinator has a guardian angel,
[...] someone called *The Panic Monster*.

Tim Urban

Inside the mind of a master procrastinator

1.1 Historical framing

1.1.1 The forerunners of vertical landing

The idea of launchers *vertical landing* deeps its roots in the Cold War, when Americans and Soviets came face to face to assert their own space dominion.

The Soviet mission *Luna 9* marked a milestone for planetary exploration: it performed the first ever soft landing, on Moon's surface on the 3th of February 1966 [1]. Soviets further pushed their probes towards other planets, demonstrating supremacy in expansion capabilities: four years later than *Luna 9*, *Venera 7* soft landed on Venus, and the following year *Mars 3* performed the first soft landing on Mars [2]. Americans focused instead on the Moon: *Surveyor 1* landed on the Moon [3] with few months delay over Soviets, but on 20th of July 1969 Americans gained back ground by setting human foot on Moon, with *Apollo 11*. Probe *Viking 1* was the first American probe to soft land on Mars, in 1976 [4]; this year also marked the end of Soviet program *Luna*: Soviets focused on *Venera* up the dissolution of the USSR. Mars environment became research environment for US Entry, Descent and Landing (EDL) technologies; progresses led to landing the rover *Curiosity* within an uncertainty ellipse of 12.5 km [5], improving by far results of previous space missions. On Earth surface, on the other hand, NASA successfully delegated further studies to Masten Space Systems' *Xombie* rocket (Fig. 1.1): tests in 2013 demonstrated online computation of large divert maneuvers; the algorithm G-FOLD managed success-

fully divert distances up to 750 m from an initial height of 400 m [6].

In addition to US, China recently made progress in landing vehicles development: *Chang'e 3* successfully landed on Moon surface in 2016 [7], while *Tianwen-1* deployed a rover on Mars surface at the beginning of 2021 [8].

1.1.2 The *Faster, Better, Cheaper* concept

Space probes and prototypes offered test benches to optimize landing performances, due to strict mission requirements. On the other hand, launchers industry could have benefited from such development: if landing capability had allowed launchers cheap reuse, the perspectives for space industry development would have flourished. This idea fitted in the management program of Daniel Goldin, NASA administrator in 1992: reusable cheap vehicles would have made missions 'faster, better, cheaper' [9].

On the 18th of August 1993, McDonnell Douglas' *DC-X* (Fig. 1.1) successfully landed vertically after a 100-meters-high 'hop'; it was the first vertical landing on Earth, and a huge opportunity for NASA; the agency endorsed the project, repeating hops with a new vehicle, the *DC-XA*. This was however dismantled in 1996 [11], signing the end of vertical landing concept within Goldin's management. On the other hand, in 1999 the Japan Aerospace Exploration Agency started the Reusable Vehicle Testing campaign, on the model of *DC-XA*. Three flight series were performed, in 1999, 2001 and 2003 [12].

We shall however wait until year 2015 to witness the first vertical landing of a proper space vehicle: Blue Origin's New Shepard, on 23rd November reached a peak altitude of 100.5 kilometers, then returned to landing site, precisely landing on the launchpad [13] (Fig. 1.1). After one year and a half, SpaceX successfully completed a satellite injection using a launcher with a recovered first stage: it was the first launch employing a used stage recovered with autonomous vertical landing [14] (Fig. 1.1).

Rocket reusability has therefore taken place outside the US. Different efforts have firstly focused on small demonstrators used for algorithms testing: it is the case of *EAGLE* [15], developed within DLR and of *FROG* [16], the counterpart from CNES; both had a tethering safety system and turbojet engines for propulsive requirements. EmboROCKETH, from ETH, is a more recent example of a similar design philosophy [17]. In a bigger picture, ArianeGroup plans on substituting the expendable *Ariane 6* with a reusable first stage-equipped rocket, *ArianeNext* [18]. In such framework different actors are being included: *Themis*, *CALLISTO* and *RETALT*. Launcher *Themis* represents a forerunner for first stage reuse [18], flight-



Figure 1.1: Overview of vertical landing vehicles - *From the left:*

- (a) DC-X performs first Earth vertical landing
- (b) Xombie completes large divert maneuver
- (c) New Shepard for the first time lands vertically after having reached space
- (d) First reused SpaceX Falcon 9 first stage lands after second successful mission
- (e) Co-simulation on Simulink-FlightGear of CALLISTO RTLS mission [10]

qualifying Prometheus engines; the trilateral project *CALLISTO* (Fig. 1.1) gathers DLR, JAXA and CNES [19, 20] to demonstrate Return-To-Landing-Site (RTLS) operations [21] with a suborbital rocket; the H2020-funded *RETALT* aims at improving technologies necessary for reusable launchers, with Elecnor Deimos leading the advancement within GNC subsystem [22].

1.2 Algorithms state of the art

Among the different mission aspects, it is evident trajectory optimization plays a major role when it comes to landing; the problem quickly escalates in complexity when constraints get included along launcher flight.

In such framework we can distinguish between *offline* and *online* optimization; the former insists on solution optimality, the latter shall ensure a feasible solution is generated with a sufficiently high update rate. *Offline optimization* algorithms can either use a direct formulation, either insist on indirect ones. Meaning of such definitions, along with successive ones, will be clarified later in the work. It is sufficient understanding the latter are usually hard to lead to convergence, yet offer the optimal solution; the former instead treat a discretized version of the continuous-time problem to be solved, employ a large number of variables, and allow for straight introduction of constraints. *Online optimization*, due to the robustness requirement, always builds on direct algorithms.

1.2.1 Offline optimization

Common direct formulations treat state variables and control as equal unknowns, leading to a *Nonlinear Programming Problem* (NLP). Such transcription is generalizable but does not guarantee successful convergence.

SPARTAN is a pseudospectral NLP solver from DLR [23]: developed within *SHEFEX-3* project, it features automatic variables scaling and dual numbers for exact first-order derivatives computation via operator overloading [24, 25].

GPOPS, *GPOPS-II* and *CGPOPS* are progressively complex algorithms developed within University of Florida. The first is a pseudospectral global collocation method [26]; the second features an *hp* mesh refinement strategy, with different discretization nodes; the last, written in C++, features hyper-dual numbers for second-order derivatives exact computation and mesh refinement strategy compatible with bang-bang control profiles [27].

Among indirect methods, shooting algorithms [28] have been combined with continuation schemes to provide solutions in highly constrained scenarios.

A different approach is provided by *DIDO*, a powerful algorithm building on pseudospectral indirect collocation schemes providing guess-free capability [29, 30]; valuable uses feature the *Zero-propellant maneuver* of the ISS [31].

1.2.2 Online optimization

A different approach is chosen to optimize trajectories *online*, i.e. integrating the optimal trajectory generation in the control loop. The employed methods usually deal with problem *convexification*[32]: problem is reformulated to increase convergence robustness and eliminate the issue of guess generation. In addition, care is provided to choosing the schemes that offer a better trade-off between accuracy and speed, to guarantee fast solution generation [33, 34].

The most general approach involves *sequential convexifications* [35, 36], while the most performing requires *lossless convexifications*, i.e. a reformulation of some originally nonconvex constraints to obtain an equivalent set of convex ones [37]. Both in sequential and lossless convexifications *Second-Order Cone* (SOC) constraints are commonly used: in sequential approach they can be used to introduce variable trust regions and virtual controls, counteracting linearization inaccuracy and artificial infeasibility risks[38]; in lossless approach they can be used to relax quadratic non-convex constraints [39]. For such flexibility, SOC constraints are widely diffused within different dynamical environments [40, 41, 42, 43, 44, 45].

Successive convexification-based algorithms are being tested on Blue Origin’s New Shepard, within NASA’s program *SPLICE* [32].

Lossless convexification has further been employed in the relevant application of Xombie: the rocket runs on the algorithm G-FOLD (Guidance for Fuel Optimal Large Divert), which enforces landing site position hardly [46, 6]. Case studies on MSL’s Skycrane have also demonstrated lossless convexification is feasible for minimization of landing site error [47]. Further developments have been applied to such algorithms within DLR: first studies have employed a global pseudospectral collocation which increases solution accuracy [40]; *hp* collocation schemes have later been employed as approaches to reduce CPU times leveraging Jacobian sparsity [41].

SpaceX’s Falcon 9, the current benchmark for pinpoint landing algorithms, is nominally guided exploiting convex problem formulation [48]; whether successive or lossless, this has never been publicly disclosed.

1.3 Research contribution

Within the presented framework, an ideal method would merge the solution accuracy of indirect methods with the robustness of the direct ones. The Covector Mapping Theorem (CMT) employed in DIDO provides a link between such two formulations: it constitutes a valuable tool to exploit a convex formulation of the landing problem;

moreover it has never been used to retrieve costates from a convexified form of the original problem feeding them to an indirect solver. Therefore the first research question.

1) Can the Covector Mapping Theorem provide a valid link between the solution of the convexified form of the vertical landing problem and the solution of the indirect form?

Typically, some optimal control problem solutions excel over the others since they manage to exploit environmental disturbances to alleviate authority of subsystems. In [39] a lossless convexification is employed to include aerodynamic forces in a 2-D environment; however, such formulation discards the effects of sideslip forces, which can instead be implemented within a 3-D environment; the second research question, therefore, follows:

2) How does the optimization of lift forces affect the optimal solution of the fuel-optimal landing problem? Does it lead to feasible solutions?

At last, few examples in literature employ indirect formulation in combination with collocation schemes [29]: collocations techniques are indeed usually faster than shooting ones. Moreover, vertical landing problems feature bang-bang type control profiles, thus forcing to tailor the collocation scheme to face possibly discontinuities. The last research question is formulated as

3) How can an indirect collocation scheme handle bang-bang type solutions of an optimal control problem? Which are its performances in terms of accuracy and speed?

1.4 Thesis structure

This thesis is organized to answer the research questions, providing the reader with the necessary means to understand the logic behind the algorithms design. Therefore Chap. 2 and 3 present the mathematical background and an overview of methods used to solve optimal control problems. In Chap. 4 the structure of the solution of the aerodynamic powered landing problem is presented, and the problem is solved using a continuation scheme coupled with an indirect shooting. Such step creates a benchmark to evaluate accuracy and computational speed of the algorithm presented in the next chapter; moreover, by assessing the solution structure we can gather enough information to tailor the indirect collocation scheme to the analyzed problem. The hybrid algorithm merging direct convex formulation with the mentioned indirect

collocation scheme is presented in Chap. 5. Research questions are answered in Chap. 6, along with possible future development strategies.

2 | Mathematical fundamentals

Find out what language this is. *Vei frotleet gaire, klafster bestarf eemur desorbaachtala.*

Creed Bratton
The Office US

The guidance strategy aims at finding a control law optimizing a given performance index in such a manner that resulting trajectory satisfies boundary conditions, and, eventually, constraints. The trajectory design problem can be indeed formulated as an optimization problem in the control domain, called *Optimal Control Problem*. Mathematical reference is found afterwards in Eq. (2.12).

In a nutshell

Given a controllable physical system, the *Optimal Control Problem* (OCP) consists of finding the control law that *optimizes* a scalar functional stemming from controls and physical states; states evolution obeys to controls and natural dynamics according to *dynamical constraints*, and shall satisfy *boundary conditions* (BCs). Eventually, constraints on the applicable control and on the admitted states shall be satisfied. Such generalization of the OCP is required to account for different physically relevant constraints [28, 49].

OCP is aimed then at 1) optimizing a function and at 2) ensuring the physical states evolve according to system dynamics and satisfy boundary conditions. Mathematical methods solving the OCP either leverage an optimization problem, either augment the dynamics, embedding the optimization within dynamics itself.

In the second case the original problem shall be *dualized* and optimality conditions define the *Boundary Value Problem* to be solved.

In a nutshell

A *Boundary Value Problem* (BVP) consists of finding an n -dimensional state, over a provided domain, satisfying differential conditions within the domain and punctual conditions at two or more points. A function satisfying such conditions is a solution of the BVP [50].

The second option consists of embedding the dynamics constraints and BCs in problem formulation, focusing on optimizing the objective function; the original problem shall be *transcribed* into its discrete-time form, and punctual dynamics constraints define the *nonlinear programming* problem to be solved.

In a nutshell

Minimizing a nonlinear scalar function dependant on a finite number of parameters whilst respecting a finite set of nonlinear constraints on the parameters is the aim of the *nonlinear programming* (NLP) *problem*.

BVPs and NLP problems represent then the main tools to approach the solution of Optimal Control Problems; OCPs, in turn, make up the most intuitive formulation of the guidance problem.

The present chapter is organized as follows: the first two sections provide the mathematical formulations of NLP problems, along with two of its subclasses, and BVPs, specialized to the case of Hamiltonian systems; the third provides an overview of the OCP, along with its forms as BVP or NLP problem.

2.1 Parametric Optimization

2.1.1 Nonlinear Programming

Let us consider a vector $\mathbf{x} \in \mathbb{R}^n$, defined as *NLP vector*, and a scalar function $F(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$, called *objective function*.

Constraints are classified as *equality constraints* and *inequality constraints*, respectively represented by $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^p$ and $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^q$. The constraints may be

assembled in the *constraints vector* $\mathbf{c} \in \mathbb{R}^{p+q}$ as

$$\mathbf{c}(\mathbf{x}) = \begin{bmatrix} \mathbf{h}(\mathbf{x}) \\ \mathbf{g}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}) \\ \vdots \\ h_p(\mathbf{x}) \\ g_1(\mathbf{x}) \\ \vdots \\ g_q(\mathbf{x}) \end{bmatrix} \quad (2.1)$$

\mathbf{c} may be possibly divided in its active component $\mathbf{c}_a \in \mathbb{R}^a$, with $a < n$ and its inactive component $\mathbf{c}_i \in \mathbb{R}^i$, with $i + a = p + q$. A point $\bar{\mathbf{x}}$ for which the Jacobian of the active constraints $\mathbf{J}_{\bar{\mathbf{x}}}\mathbf{c}_a$ is full rank is said to satisfy the Linear Independence Constraint Qualification (LICQ) [51].

The NLP problem can be mathematically formulated as follows [52]

$$\min_{\mathbf{x} \in \mathbb{R}^n} F(\mathbf{x}) \quad \text{s.t.} \quad \begin{cases} \mathbf{h}(\mathbf{x}) = \mathbf{0} \\ \mathbf{g}(\mathbf{x}) \preceq \mathbf{0} \end{cases} \quad (2.2)$$

2.1.2 Convex Programming

Despite its mathematical statement simplicity, solving the NLP problem becomes rapidly challenging with the increase of NLP vector size [52]; the non linearity of the objective function and the existence of different local minima play a central role in the complexity of the NLP solution. On the other hand, as peremptorily stated by Rockafellar [53], the property that grants main advantages when solving an optimization problem is its *convexity* rather than *linearity*: a problem is defined as *convex* when it is in the form

$$\min_{\mathbf{x} \in \mathbb{R}^n} f_0(\mathbf{x}) \quad \text{s.t.} \quad \begin{cases} f_1(\mathbf{x}) \leq b_1 \\ \vdots \\ f_m(\mathbf{x}) \leq b_m \\ f_{m+1}(\mathbf{x}) = b_{m+1} \\ \vdots \\ f_{m+n}(\mathbf{x}) = b_{m+n} \end{cases} \quad (2.3)$$

and $f_0, \dots, f_{m+n} : \mathbb{R}^n \rightarrow \mathbb{R} \forall i = 0, \dots, m+n$ are *convex*, namely satisfy the *Jensen's inequality*¹

$$f_i(\alpha \mathbf{x} + \beta \mathbf{y}) \leq f_i(\alpha \mathbf{x}) + f_i(\beta \mathbf{y}) \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \forall \alpha, \beta \in \mathbb{R} \quad \text{s.t.} \quad \begin{cases} \alpha + \beta = 1 \\ \alpha \geq 0 \\ \beta \geq 0 \end{cases} \quad (2.4)$$

Linearity represents indeed a more restrictive property than *convexity*: this suggests that a convexified form of an NLP problem represents a better trade-off between computational advantages and generalizability of the technique with respect to a linearized form. Indeed, the original problem can in some cases be rearranged in a *lossless* convexified form [37]: appropriate transformations allow for transferring the non-convexities from some constraints (*e.g.* non-convex dynamics) to other constraints. Such last non-convex constraints can be *relaxed* into convex ones: the solution of such final form, can be demonstrated to be *equivalent* to the original problem one. This is why such convex formulation is referred to as *lossless*.

Remark 2.1: Problems featuring high non linearities are likely to be not losslessly convexifiable; however, in practical applications, a *sequential programming* strategy is always employed, leading to a *successive* convexification approach [56, 57]: this corresponds to convexifying the problem at each iteration of the optimization, tackling eventual information loss due to the convexification process with multiple sequential optimizations.

Remark 2.2: Convex problems solvability easiness over non-convex ones is mainly associated with two factors: 1) since the *optimum* of a convex problem is its *global optimum*, the guess choice does not influence the minimum that is reached; 2) if coupled with proper algorithms, convex problems are solved with a number of operations growing polynomially with respect to the problem dimensions; such algorithms include *primal-dual interior-point methods*, as employed in the solver ECOS [58]; NLP problems, instead, feature exponential growth of operations number [59].

Several programming problems may be transformed into specific subsets of convex problems; a problem hierarchy with increasing complexity and generality features,

¹Johan Jensen's work focused actually on the specific form of (2.4) with $\alpha = \beta = 0.5$ [54]; nonetheless, *Jensen's inequality* definition is often employed as referring to (2.4) [55], a generalized expression.

in order, *linear problems*, *quadratic problems* and *geometric programming*; *generalized constraints* and *generalized objective function problems* eventually account for vectorial expressions of the objective function and constraints [55]. The choice of the convex programming technique to convexify the problem with is traded-off accounting for convexification accuracy against efficient solving algorithms availability.

2.1.3 Second-Order Cone Programming

Convexification of the starting NLP problem as a particular quadratic one, namely a *Second-Order Cone Programming* (SOCP) problem, has proved useful for different applications; its general form is then hereafter described.

Let $\mathbf{x} \in \mathbb{R}^n$ be the optimization variable and let $\mathbf{f} \in \mathbb{R}^n$ be the cost function coefficients vector; in addition, $\mathbf{A}_i \in \mathbb{R}^{n_i \times n}$, $\mathbf{b}_i \in \mathbb{R}^{n_i}$, $\mathbf{c}_i \in \mathbb{R}^n$, $d_i \in \mathbb{R}$ are associated with quadratic inequality constraints, while $\mathbf{P} \in \mathbb{R}^{p \times n}$, $\mathbf{p} \in \mathbb{R}^p$ and $\mathbf{Q} \in \mathbb{R}^{q \times n}$ and $\mathbf{q} \in \mathbb{R}^q$ are respectively related to linear inequality and linear equality constraints². The general SOCP problem results then

$$\min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{f}^T \mathbf{x} \quad \text{s.t.} \quad \begin{cases} \|\mathbf{A}_i \mathbf{x} + \mathbf{b}_i\|_2 \leq \mathbf{c}_i^T \mathbf{x} + d_i & i = 1, \dots, m \\ \mathbf{P} \mathbf{x} \preceq \mathbf{p} \\ \mathbf{Q} \mathbf{x} = \mathbf{q} \end{cases} \quad (2.5)$$

The inequality $\|\mathbf{A}\mathbf{x} + \mathbf{b}\|_2 \leq \mathbf{c}^T \mathbf{x} + d$, with $\mathbf{A} \in \mathbb{R}^{r \times n}$, is defined as *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. 2.1 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 region bordered on the bottom by the represented surface, a cone in \mathbb{R}^3 .

²Linear 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 [55], thus linear inequalities are explicitly defined.

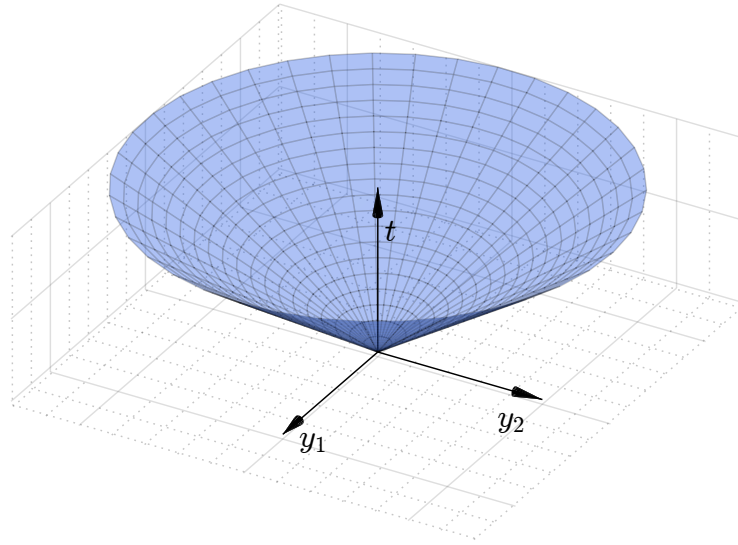


Figure 2.1: Second order cone constraint in \mathbb{R}^3 : $\|\mathbf{y}\|_2 \leq t$

2.2 Boundary Value Problem

2.2.1 One-dimensional case

Let the vector $\mathbf{x}(t) \in \mathbb{R}^n$ with $t_0 < t < t_f$ flow according to the time varying vector field $\mathbf{f} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$. Given the two vectors $\mathbf{x}_0 \in \mathbb{R}^n$ and $\mathbf{x}_f \in \mathbb{R}^n$, the *Two-Point Boundary Value Problem* (TPBVP) for a set of first order ODEs can be mathematized as follows [50]

$$\text{Find } \mathbf{x}(t) \text{ s.t. } \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) \text{ and } \begin{cases} \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{x}(t_f) = \mathbf{x}_f \end{cases} \quad (2.6)$$

The TPBVP can be generalized to the *Multi-Point Boundary Value Problem* (MP-BVP) by adding punctual constraints on $\mathbf{x}(t)$ at an arbitrary number i of instants t_i over the compact $[t_0, t_f]$.

Well-posedness and *well-conditioning* of a BVP are of paramount importance in the context of practical applications: a *well-posed* BVP is characterized by 1) *existing* and *unique* solution; 2) continuous dependance of the solution on the provided data [60]. A *well-conditioned* BVP features instead small variations in the solution for small variations in boundary conditions and dynamics equation.

2.2.2 Hamiltonian systems

BVPs associated with *Hamiltonian systems* play a central role in the context of OCPs: they make up the means to embed optimality conditions in problem formulation, but intrinsically feature ill-conditioning.

A Hamiltonian system is a system of the form [61]

$$\dot{\mathbf{p}} = -\nabla_{\mathbf{q}}\mathcal{H}(\mathbf{q}, \mathbf{p}) \quad \dot{\mathbf{q}} = \nabla_{\mathbf{p}}\mathcal{H}(\mathbf{q}, \mathbf{p}) \quad (2.7)$$

where $\mathbf{q} \in \mathbb{R}^n$ is the vector of *generalized coordinates* and $\mathbf{p} \in \mathbb{R}^n$ is the vector of coordinates' *conjugate momenta*.

\mathbf{q} and \mathbf{p} dynamics can be hardly ever decoupled; whatever boundary condition is then added to (2.7), coordinates and momenta dynamics shall be integrated in the same time direction. This leads to afore-mentioned ill-conditioning problems: loosely speaking, the faster the forward-time convergence for coordinates, the faster the forward-time divergence of the conjugate momenta; strictly speaking, coordinates and momenta ODEs have an *exponential dichotomy* [50].

A simple geometric interpretation allows for deeper insights: if $\mathcal{H} \in C^2(U \subseteq \mathbb{R}^{2n})$, the divergence of Hamiltonian differential operator is null; flow it is then said to make up for a *symplectic* transformation, namely it makes up a hypervolume-preserving operator in the phase space of coordinates and momenta [62]. Hence the coupled converging-diverging behaviour of \mathbf{p} and \mathbf{q} .

Summing up, due to the opposite stability features of coordinates and momenta for a Hamiltonian system, small changes in the boundary conditions do not imply small changes in the solution of the associated BVP; the BVP itself is then ill-conditioned.

2.3 Optimal Control Problem

2.3.1 Problem Statement

Let us consider a physical system, described by the continuous-time varying state $\mathbf{x}(t) \in \mathbb{R}^n$, with $t \in [t_0, t_f]$, and characterized by a vector of known parameters $\mathbf{p} \in \mathbb{R}^l$; its dynamics are described by $\mathbf{f} : \mathbb{R}^{n+m+l+1} \rightarrow \mathbb{R}^n$ according to the *dynamical constraint*

$$\dot{\mathbf{x}} = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}, t] \quad (2.8)$$

and driven by the continuous-time control $\mathbf{u}(t) \in \mathbb{R}^m$.

Dropping, for notation ease, the explicit dependency with respect to time, it is

possible to define a functional of $\{\mathbf{x}, \mathbf{u}, t\}$, the *objective function* $\mathcal{J} : \mathbb{R}^{n+m+1} \rightarrow \mathbb{R}$ which reads

$$\mathcal{J}(\mathbf{x}, \mathbf{u}, t) = \phi(\mathbf{x}_f, t_f) + \int_{t_0}^{t_f} \mathcal{L}(\mathbf{x}, \mathbf{u}, t) dt \quad (2.9)$$

where $\phi : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ is the *terminal cost* and $\mathcal{L} : \mathbb{R}^{n+m+1} \rightarrow \mathbb{R}$ is the *path cost* [51]. Along the overall interval $[t_0, t_f]$, \mathbf{x} and \mathbf{u} shall further satisfy the *path constraints* expressed by $\mathbf{g} : \mathbb{R}^{n+m+1} \rightarrow \mathbb{R}^p$, according to

$$\mathbf{g}(\mathbf{x}, \mathbf{u}, t) \preceq \mathbf{0} \quad \text{categorized as} \quad \begin{cases} \text{Pure control constraints:} & \mathbf{g}_{p,u}(\mathbf{u}, t) \preceq \mathbf{0} \\ \text{Pure state constraints}^3: & \mathbf{g}_{p,x}(\mathbf{x}, t) \preceq \mathbf{0} \\ \text{Mixed constraints:} & \mathbf{g}_m(\mathbf{x}, \mathbf{u}, t) \preceq \mathbf{0} \end{cases} \quad (2.10)$$

Defining an inequality constraint as *active* if it is *weakly satisfied* and *inactive* otherwise, the jacobian $\partial \mathbf{g}_a / \partial \mathbf{u}^T$ of the *active* inequality constraints \mathbf{g}_a , shall be full rank. Necessary condition for such hypothesis to hold is that *pure state constraints* are not present. Challenge of both cases follows in the work.

Control and states shall satisfy the *initial* and *terminal constraint*, linked to $\boldsymbol{\psi}_0 : \mathbb{R}^{n+m+1} \rightarrow \mathbb{R}^q$ and $\boldsymbol{\psi}_f : \mathbb{R}^{n+m+1} \rightarrow \mathbb{R}^r$ with $q, r < n^4$, which respectively read

$$\begin{aligned} \boldsymbol{\psi}_0(\mathbf{x}_0, \mathbf{u}_0, t_0) &= \mathbf{0} \\ \boldsymbol{\psi}_f(\mathbf{x}_f, \mathbf{u}_f, t_f) &= \mathbf{0} \end{aligned} \quad \text{where} \quad \begin{cases} \mathbf{x}_0 = \mathbf{x}(t_0) \\ \mathbf{u}_0 = \mathbf{u}(t_0) \\ \mathbf{x}_f = \mathbf{x}(t_f) \\ \mathbf{u}_f = \mathbf{u}(t_f) \end{cases} \quad (2.11)$$

Controls satisfying (2.10) and (2.11) are said to belong to the *set of admissible controls* $\mathcal{U} \subseteq \mathbb{R}^m$ [65, 51].

³*Pure state constraints* are usually denoted as $\mathbf{S}(\mathbf{x}, t) \preceq \mathbf{0}$ [63, 64]; symbol \mathbf{S} is therefore adopted in the remainder of the present work.

⁴The relation linking q, r and n refers to the constraint usually found when dealing with practical applications: boundary conditions as equations equalling the initial and final states to constant or time dependant variables.

The Optimal Control Problem (OCP) can be finally mathematically formalized [52].

$$\min_{\mathbf{u}} \mathcal{J}(\mathbf{x}, \mathbf{u}, t) \quad \text{s.t.} \quad \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{p}, t) \\ \mathbf{g}(\mathbf{x}, \mathbf{u}, t) \preceq \mathbf{0} \\ \boldsymbol{\psi}_0(\mathbf{x}_0, \mathbf{u}_0, t_0) = \mathbf{0} \\ \boldsymbol{\psi}_f(\mathbf{x}_f, \mathbf{u}_f, t_f) = \mathbf{0} \end{cases} \quad (2.12)$$

Remark 2.3: Expression provided in (2.9) coincides with the *Bolza formulation* of the problem, which is of interest due to the penalization of both final state and integral costs. It can be nonetheless shown [51] that provided formulation is equivalent to the *Lagrange formulation* or the *Mayer formulation*, which account respectively for the *path cost* or for the *terminal cost* only.

Remark 2.4: Despite the problem solution including only the control, no assumption on the final time t_f value has been made: the problem, as for landing applications, may be a *free-final time* one, with solution $\mathbf{u}(t)$ valid as long as the *terminal constraints* are satisfied; t_f adds, in this generalized case, as further degree of freedom.

Remark 2.5: For the remainder of this work, the initial time will always be considered as known and fixed.

The two features of the OCP are evident from statement (2.12): on the one hand there is one functional to be optimized, on the other dynamics and boundary constraints to be satisfied. The two following sections explicitate the two possible formulations of the OCP, along with the associated optimality necessary conditions.

2.3.2 OCP as BVP - Optimality Necessary Conditions

The OCP is by definition a continuous-time problem; the optimality can be embedded in a BVP: first the original problem shall be dualized, then the dynamics and the boundary conditions for the *primal-dual* system are retrieved from the first-order optimality necessary conditions: the employed process follows.

OCP dualization

For simplicity, let the dynamic function \mathbf{f} embed the parameters \mathbf{p} in its own definition. The constrained problem outlined in (2.12) can be condensed in the minimization of an *augmented cost function* $\hat{\mathcal{J}}$: the path cost is substituted by its *Legendre transform* based on the *costates* $\boldsymbol{\lambda} \in \mathbb{R}^n$ and the *path constraints Lagrange multipliers* $\boldsymbol{\mu} \in \mathbb{R}^p$ while the terminal cost is substituted by its *Legendre transform* based on the *initial and terminal constraints Lagrange multipliers* $\boldsymbol{\nu}_0 \in \mathbb{R}^q, \boldsymbol{\nu}_f \in \mathbb{R}^r$ [61, 66, 67]; from $\hat{\mathcal{J}}$ definition in (2.13), it is evident that, insofar as constraints as satisfied, $\hat{\mathcal{J}}$ optimality implies \mathcal{J} optimality. $\hat{\mathcal{J}}$ is indeed defined as

$$\hat{\mathcal{J}} = \Phi_f + \boldsymbol{\nu}_0^T \boldsymbol{\psi}_0 + \int_{t_0}^{t_f} \left\{ \mathcal{L}(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T \{ \mathbf{f}(\mathbf{x}, \mathbf{u}, t) - \dot{\mathbf{x}} \} + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}, \mathbf{u}, t) \right\} dt \quad (2.13)$$

where $\Phi_f = \phi(\mathbf{x}_f, t_f) + \boldsymbol{\nu}_f^T \boldsymbol{\psi}_f$; using for $\boldsymbol{\lambda}$ a similar notation as the employed in (2.11), the former expression can be rearranged as follows

$$\hat{\mathcal{J}} = \Phi_f + \boldsymbol{\nu}_0^T \boldsymbol{\psi}_0 + \boldsymbol{\lambda}_0^T \mathbf{x}_0 - \boldsymbol{\lambda}_f^T \mathbf{x}_f + \int_{t_0}^{t_f} \left\{ \mathcal{L}(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}, \mathbf{u}, t) + \dot{\boldsymbol{\lambda}}^T \mathbf{x} \right\} dt \quad (2.14)$$

The Hamiltonian of the system can be further introduced by transforming the Lagrangian term in a similar fashion as what done to obtain $\hat{\mathcal{J}}$ from \mathcal{J} ; the Hamiltonian $\mathcal{H} : \mathbb{R}^{2n+m+p+1} \rightarrow \mathbb{R}$ reads indeed

$$\mathcal{H}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\mu}, t) = \mathcal{L}(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\mu}^T \mathbf{g}(\mathbf{x}, \mathbf{u}, t) \quad (2.15)$$

Thanks to the application of the Legendre transform on the cost function terms the Lagrangian *path cost* \mathcal{L} is naturally substituted by its dual [61] Hamiltonian \mathcal{H} . The switch from a *Lagrangian* approach to a *Hamiltonian* is completed by the optimality necessary conditions, which mirror the problem dualization.

Setting to zero the *differential* of $\hat{\mathcal{J}}$ with respect to the state, the control and eventually the final time in the admissible directions allows to retrieve conditions on the *adjoint variables*, while accounting for the *adjoint variables* allows to retrieve back the constraints imposed on the non-dualized form of the OCP.

Euler-Lagrange Equations

Following a suggestion from Bryson's vast work [66] and exploiting the Hamiltonian leads to

$$\begin{aligned}
 d\hat{\mathcal{J}} = & \left[\nabla_{\mathbf{x}_f}^T \Phi_f - \boldsymbol{\lambda}_f^T \right] d\mathbf{x}_f + \left[\boldsymbol{\nu}_0^T \nabla_{\mathbf{x}_0} \psi_0 + \boldsymbol{\lambda}_0^T \right] d\mathbf{x}_0 + \left[\frac{\partial \Phi_f}{\partial t_f} - \dot{\boldsymbol{\lambda}}_f^T \mathbf{x}_f + \mathcal{H}(t_f) + \dot{\boldsymbol{\lambda}}_f^T \mathbf{x}_f \right] dt_f + \\
 & \left[\boldsymbol{\nu}_0^T \frac{\partial \psi_0}{\partial t_0} + \dot{\boldsymbol{\lambda}}_0^T \mathbf{x}_0 - \mathcal{H}(t_0) - \dot{\boldsymbol{\lambda}}_0^T \mathbf{x}_0 \right] dt_0 + \int_{t_0}^{t_f} \left\{ \left(\nabla_{\mathbf{x}}^T \mathcal{H} + \dot{\boldsymbol{\lambda}}^T \right) \delta \mathbf{x} + \nabla_{\mathbf{u}}^T \mathcal{H} \delta \mathbf{u} \right\} dt
 \end{aligned} \tag{2.16}$$

where the boxed terms cancel each other out.

Imposing $d\hat{\mathcal{J}} = 0$ allows to retrieve the *Hamiltonian form of the Euler-Lagrange equations for the OCP*:

$$\left\{ \begin{array}{l} \text{State dynamics:} \\ \text{Costate dynamics:} \\ \text{Control equation:} \end{array} \right. \quad \left\{ \begin{array}{l} \dot{\mathbf{x}} = \nabla_{\boldsymbol{\lambda}} \mathcal{H} \\ \dot{\boldsymbol{\lambda}} = -\nabla_{\mathbf{x}} \mathcal{H} \\ \mathbf{0} = \nabla_{\mathbf{u}} \mathcal{H} \end{array} \right. \quad \text{with boundary conditions} \quad \left\{ \begin{array}{l} \mathbf{x}_0 = \mathbf{x}_0^* \\ t_0 = t_0^* \\ \boldsymbol{\lambda}_f = \nabla_{\mathbf{x}_f} \Phi_f \\ \mathcal{H}_f = -\frac{\partial \Phi_f}{\partial t_f} \end{array} \right. \tag{2.17}$$

Additional constraints on the path multipliers components shall be added [68], referring to (2.13) or (2.14), which read

$$\begin{aligned}
 \mu_i &= 0 & \text{if } g_i(\mathbf{x}, \mathbf{u}, t) < 0 \\
 \mu_i &> 0 & \text{if } g_i(\mathbf{x}, \mathbf{u}, t) = 0
 \end{aligned} \tag{2.18}$$

Such additional constraints complement the augmentation of the Lagrangian with the mixed path constraints \mathbf{g} (2.15) to provide the Hamiltonian formulation of the *Erdmann corner conditions* [63]. The concept is further analyzed in Appendix A.

Remark 2.6: Coherently with statement in **Remark 2.5**, boundary conditions have been defined in (2.17) for practical application purposes: in OCP application fields it is usual to have both the initial state and the initial time imposed; on the

other hand, free-final state and free-final time features determine the constraints on costate and Hamiltonian final values.

Remark 2.7: States at boundaries may be defined either fully or partially; conditions on missing state components are respectively substituted by conditions on dual costate components.

Remark 2.8: As hinted in the problem statement, two hypotheses underlie the optimality necessary conditions expressed in (2.17): 1) *pure state constraints* shall not be present among path constraints, 2) the Jacobian of the active constraints shall be full rank. For the purposes of complete reentry scenarios optimization, such hypotheses shall possibly be relaxed; for the former, *jump conditions*, namely discontinuities over costates and Hamiltonian, shall be satisfied; for the latter, an *index reduction* in the set of *Differential Algebraic system of Equations* (DAE) shall be instead taken into account [65]. The two techniques are reported in Appendix A. For the purposes of terminal landing optimization, however, such hypothesis hold, thus allowing the aforementioned simplification.

Formulation in (2.17) corresponds, as previously mentioned, to a *Two-Point Boundary Value Problem* (TPBVP); in addition, system dynamics is hardly ever linear with respect to the state, thus the ill-conditioning problems mentioned in Sec. 2.2.2 arise. At last, the control equation introduces a further algebraic constraint. The resulting set of equations is a DAEs: the control equation is then often reelaborated to obtain an explicit expression of the control; a *Pure Two-Point Boundary Value Problem* (PTPBVP) is thus obtained, as the control can be expressed as function of system states and costates. Another approach consists of substituting the control equation with its equivalent, the *Pontryagin's Minimum Principle* (PMP).

Pontryagin's Minimum Principle

Pontryagin's Minimum Principle states the necessary condition for control law $\mathbf{u}^*(t)$ optimality: simply speaking, $\mathbf{u}^*(t)$ shall minimize the Hamiltonian, provided \mathbf{u}^* belongs to the set of admissible controls \mathcal{U} [69]. The corresponding mathematical expression results

$$\mathbf{u}^* = \underset{\mathbf{u} \in \mathcal{U}}{\operatorname{argmin}} \left\{ \tilde{\mathcal{H}}(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}, t) \right\} \quad \text{with} \quad \tilde{\mathcal{H}} = \mathcal{L}(\mathbf{x}, \mathbf{u}, t) + \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \quad (2.19)$$

The condition expressed by the *control equation* applies to the OCP provided that path constraints are embedded in the definition of the Hamiltonian, and is verified when control constraints are inactive; it instead fails if the stationary point of the Hamiltonian with respect to controls does not belong to \mathcal{U} . Pontryagin's Minimum Principle is, indeed, a generalization of the Hamiltonian control equation; it still holds if Hamiltonian is formulated for simplicity ignoring pure control path constraints, as in (2.19).

2.3.3 OCP as NLP - Optimality Necessary conditions

The dynamics and boundary conditions of the OCP can be transformed in a finite set of constraints by time-discretization; in such way one can treat the OCP as nonlinear programming problem: the *transcription* allows for switching to the discrete-time formulation, and parametric optimization algorithms can be employed for solving the problem.

OCP transcription

Let us discretize the free-bounded time interval $[t_0, t_f]$ into a finite sequence of N successive time instants t_i with $i = 1, \dots, N$, and let \mathbf{x}_i and \mathbf{u}_i be the state and the control at t_i ; such discretization points are referred as *nodes*⁵. Let us assemble the *NLP vector* $\mathbf{y} \in \mathbb{R}^s$, with $s = N(n + m) + 2$, as followingly expressed

$$\mathbf{y} = \left[\underbrace{\mathbf{x}_1^T, \dots, \mathbf{x}_N^T}_{\mathbf{X}}, \underbrace{\mathbf{u}_1^T, \dots, \mathbf{u}_N^T}_{\mathbf{U}}, t_0, t_f \right]^T \doteq \left[\mathbf{X}^T, \mathbf{U}^T, t_0, t_f \right]^T \quad (2.20)$$

Continuous-time dynamics can be *transcribed*, i.e. it can be imposed at a fixed number of instants t_j , with $j = 1, \dots, M$ over the domain. In addition, the *objective function* can be discretized, and the eventual path cost integrated numerically; the functional \mathcal{J} can be approximated by the parametric function $\bar{\mathcal{J}}(\mathbf{y})$, written as follows

$$\bar{\mathcal{J}}(\mathbf{y}) = \phi(\mathbf{x}_N, t_N) + \sum_{i=1}^N w_i \mathcal{L}(\mathbf{x}_i, \mathbf{u}_i, t_i) \quad (2.21)$$

being w_i , with $i = 1, \dots, N$ the i -th component of the integration weights vector \mathbf{w} ; it has further been supposed for simplicity the last nodes t_N to coincide with t_f .

Discrete path constraints can be divided into *equality constraints* and *inequality*

⁵Not necessarily $t_1 = t_0$ or $t_N = t_f$

constraints, and dynamic constraints can be embedded in the former; both are then respectively represented by $\mathbf{h} : \mathbb{R}^s \rightarrow \mathbb{R}^p$ and $\mathbf{g} : \mathbb{R}^s \rightarrow \mathbb{R}^q$, such that

$$\mathbf{h}(\mathbf{y}) = \mathbf{0} \quad (2.22)$$

$$\mathbf{g}(\mathbf{y}) \preceq \mathbf{0} \quad (2.23)$$

The discrete-time version of the OCP can be at last mathematically formulated as follows

$$\min_{\mathbf{y} \in \mathbb{R}^s} \bar{\mathcal{J}}(\mathbf{y}) \quad \text{s.t.} \quad \begin{cases} \mathbf{h}(\mathbf{y}) = \mathbf{0} \\ \mathbf{g}(\mathbf{y}) \preceq \mathbf{0} \end{cases} \quad (2.24)$$

Karush–Kuhn–Tucker conditions

Solving the OCP consists then of solving a constrained optimization problem.

The cost function is augmented by means of the vector of Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^p$ and $\boldsymbol{\nu} \in \mathbb{R}^q$, leading to the Lagrangian $L : \mathbb{R}^{s+p+q} \rightarrow \mathbb{R}$ defined as

$$L(\mathbf{y}, \boldsymbol{\lambda}) = \bar{\mathcal{J}}(\mathbf{y}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{y}) + \boldsymbol{\nu}^T \mathbf{g}(\mathbf{y}) \quad (2.25)$$

Lagrangian definition is useful to introduce the first order necessary conditions of optimality, known as *Karush–Kuhn–Tucker* (KKT) conditions: if \mathbf{y}^* is a local minimum point satisfying the LICQ, there exist unique vectors $\boldsymbol{\lambda}^*$, $\boldsymbol{\nu}^*$ such that

$$\left\{ \begin{array}{l} \nabla_{\mathbf{y}} L = \mathbf{n}(\mathbf{y}^*) + \mathbf{G}(\mathbf{y}^*)^T \boldsymbol{\lambda}^* + \mathbf{H}(\mathbf{y}^*)^T \boldsymbol{\nu}^* = \mathbf{0} \\ \nabla_{\boldsymbol{\lambda}} L = \mathbf{h}(\mathbf{y}^*) = \mathbf{0} \\ \nabla_{\boldsymbol{\nu}} L = \mathbf{g}(\mathbf{y}^*) \preceq \mathbf{0} \\ \boldsymbol{\nu}^* \succeq \mathbf{0} \\ \boldsymbol{\nu}^{*T} \mathbf{g}(\mathbf{y}^*) = \mathbf{0} \end{array} \right. \quad (2.26)$$

where \mathbf{G} , \mathbf{H} and \mathbf{n} are respectively the *equality and inequality constraints Jacobian matrices* and the *cost function gradient*, mathematically defined as

$$\mathbf{G}(\mathbf{y}) = \begin{bmatrix} \frac{\partial g_1}{\partial y_1} & \cdots & \frac{\partial h_1}{\partial y_s} \\ \vdots & & \\ \frac{\partial g_p}{\partial y_1} & \cdots & \frac{\partial h_p}{\partial y_s} \end{bmatrix}, \quad \mathbf{H}(\mathbf{y}) = \begin{bmatrix} \frac{\partial h_1}{\partial y_1} & \cdots & \frac{\partial g_1}{\partial y_s} \\ \vdots & & \\ \frac{\partial h_q}{\partial y_1} & \cdots & \frac{\partial g_q}{\partial y_s} \end{bmatrix},$$

$$\mathbf{n}(\mathbf{y}) \equiv \nabla_{\mathbf{y}} \bar{\mathcal{J}} = \begin{bmatrix} \frac{\partial \bar{\mathcal{J}}}{\partial y_1} \\ \vdots \\ \frac{\partial \bar{\mathcal{J}}}{\partial y_s} \end{bmatrix} \quad (2.27)$$

3 | Numerical methods

- What a filthy job!
- Could be worse.
- How?
- Could be raining.

Dr. Frankenstein and Igor
Young Frankenstein

Two different mathematical formulations of the OCP have been discussed, yet no explicit insight about employed methods has been provided. A first techniques classification, however, naturally stems from the different OCP mathematical formulations; *direct methods* and *indirect methods* can then be distinguished.

In a nutshell

A *direct method* optimizes the objective function by solving a zero-finding problem involving both controls and states, while an *indirect method* finds the states and costates satisfying the optimality necessary conditions, thus dictating the optimal control law.

The former indeed *directly* looks for the control law, while the latter *indirectly* finds it by using the control equation and the solution of the MPBVP. Direct methods, then, operate on an approximated NLP formulation of the OCP; they *discretize, then optimize*; direct methods offer wide basins of attraction and have thus been traditionally used for their robustness. On the other hand, indirect ones operate on the equivalent BVP formulation, thus *optimizing, then discretizing* during the integration process [65]; for the reasons mentioned in Sec. 2.2.2, it is difficult to achieve convergence with an indirect method; on the other hand its solution satisfies OCP optimality conditions by definition, thus it usually features high accuracy.

A second distinction arises from the nature of the integration schemes employed for solving the problem, allowing to distinguish between *shooting methods* and *colloca-*

tion methods.

In a nutshell

A *shooting* or *time-marching method* simulates the system dynamics sequentially while a *collocation method* integrates the dynamics simultaneously as dynamic constraints are imposed on the overall integration domain [70]; integration is therefore performed either explicitly or implicitly. Points in which dynamics is reinforced are defined *collocation points*.

Collocation and shooting techniques both allow for imposing dynamics: they can be thus employed either to reinforce dynamic constraints as in direct methods, either to solve BVPs as in indirect methods.

Traditional collocation techniques work with integral constraints, and nodes differ from collocation points; in the latest years, *pseudospectral* collocation techniques have gained the attention of research community.

In a nutshell

A *pseudospectral* (PS) method, also known as *orthogonal collocation* method [70], is a collocation technique. It is based on a) approximating solution as summation of Lagrange polynomials and b) reinforcing dynamics at points that coincide with roots of orthogonal polynomials.

From PS methods definition, collocation points coincide with nodes: the punctual nature of dynamic constraints allows for their straightforward reinforcement. In addition, the orthogonal polynomials employment provides pseudospectral methods with quasi-exponential convergence rate [71].

pseudospectral methods play a central role in one of the major recent advancements in the context of optimal trajectories design, the *Covector Mapping Principle*.

In a nutshell

Given a sequence of approximations of the OCP converging to the original problem, the *Covector Mapping Principle* (CMP) states that, under certain circumstances, there exist of a sequence of solutions for the approximated problem converging to the solution of the original one [72]. Such circumstances are provided by a *Covector Mapping Theorem* (CMT).

It turns that simple *Covector Mapping Theorems* can be demonstrated when the

original problem is transcribed with a pseudospectral method [73], offering an implementable solution for trajectory optimization purposes [74, 75]. The CMP offers a unique possibility to solve the starting OCP by cherry-picking attraction-basin width from direct methods and solution accuracy from indirect.

The present chapter deepens the topics afore described, starting firstly with a presentation of the traditionally employed techniques for solution of OCPs. PS methods features are then outlined and analyzed; a survey of nodes on which PS methods are based follows, while the last subsection specializes the employed PS scheme to a direct method, the relative CMT and an indirect method.

3.1 Traditional methods

As mentioned in the introduction, methods to solve optimal control problems can be either direct or indirect, either time-marching or collocation-based ones. A first distinction between the integration techniques couple is presented, then a rapid survey of direct and indirect methods is carried out.

3.1.1 Time-marching and Collocation techniques

A *time-marching* method propagates the initial conditions of a provided *Initial Value Problem* (IVP) by means of sequential integration techniques, divided in *Multi-Stage* and *Multi-Step* methods.

Let an initial state $\mathbf{x}_0 \in \mathbb{R}^{n_s}$ evolve according to $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ through the interval $[t_0, t_f]$, and let us want to evaluate state \mathbf{x}_{k+1} at time t_{k+1} . A multi-stage method exploits the state evaluation \mathbf{x}_k at time t_k and additional evaluations of \mathbf{f} in the interval $[t_k, t_{k+1}]$; a multi-step method exploits propagated states $[\mathbf{x}_{k-n}, \dots, \mathbf{x}_k]$ at previous time instants $[t_{k-n}, \dots, t_k]$. Runge-Kutta (RK) and Adam-Bashfort-Moulton (ABM) schemes are valuable examples of respectively multi-stage and multi-step methods. A traditional *collocation* method approximates the state evolution with an analytic function with unknown coefficients¹, and retrieves coefficients by reinforcing dynamics at *collocation points*. Such procedure is referred to as *implicit integration*, and the choice of the approximating function, namely the integration scheme, dictates method accuracy.

¹First Order Hold (FOH) discretization is an exception, as a method of this kind parameterizes only the controls and integrates the dynamics

Remark 3.1: Location of collocation points is theoretically arbitrary; however, the optimal choice of collocation points changes with the integration scheme: desired accuracy is obtained if collocation points coincide with the optimal ones for the employed scheme [76]. Optimal location of collocation points never coincides with discretization nodes.

Considering the same example cited before, let t_j be the generic collocation point over the integration interval, while t_i denotes the generic node; furthermore, let $\mathbf{a} = [a_1, \dots, a_m] \in \mathbb{R}^m$ be the vector of function coefficients such that $\mathbf{x} = \mathbf{x}(\mathbf{a})$; since at nodes $\mathbf{x}_i = \mathbf{x}_i(\mathbf{a})$ and at collocation points $\mathbf{x}_j = \mathbf{x}_j(\mathbf{a})$, one can invert the first equation and obtain $\mathbf{x}_j = \mathbf{x}_j(\mathbf{x}_i)$. Dynamics is integrated by nulling the *defect* ζ_j at each collocation point, according to the following equation

$$\dot{\mathbf{X}}_j(\mathbf{X}_i) = \mathbf{f}(\mathbf{X}_j, t_j) = \mathbf{f}(\mathbf{X}_i, t_j) \implies \zeta_j = \dot{\mathbf{X}}_j(\mathbf{X}_i) - \mathbf{f}(\mathbf{X}_i, t_j) = \mathbf{0} \quad (3.1)$$

where $\dot{\mathbf{X}}_j$ is an approximation of states derivatives at t_j which expression depends on the employed scheme, and $\mathbf{X}_i \in \mathbb{R}^{n_s n_n}$ is the vector of states at the n_n nodes. Eq. (3.1) constitutes a nonlinear system of equations: solution is refined over whole domain at each iteration, thus integration is performed with a parallel approach. State is usually approximated as sum of polynomials; collocation is referred to as *global* or *local* if such polynomials are respectively defined on the overall domain or on subintervals of it.

3.1.2 Indirect and direct methods

Indirect methods solve the BVP associated to the dual system, with boundary conditions dictated by optimality necessary conditions. Following approaches are then often adopted.

Indirect single shooting Employment of a time-marching integration to impose boundary conditions on a problem is often referred to as *shooting*. States \mathbf{x} and costates $\boldsymbol{\lambda}$ are 'shot' from one of time domain bounds, according to the flow φ of the Hamiltonian dynamical system; they are then corrected proportionally to boundary conditions violation at the opposite bound: such correction step is often based on linearization of dynamics flux and therefore referred to as *differential correction*. As such function is nonlinear, however, the differential correction shall be iterated; when the required accuracy is at last reached, the control can be retrieved from states and costates histories.

Indirect multiple shooting The previous approach may be improved with a *multiple-shooting* approach: propagation time is subdivided into subintervals $[t_k, t_{k+1}]$, states \mathbf{x}_k and costates $\boldsymbol{\lambda}_k$ are 'shot' over their respective subinterval and the equality constraint $\varphi([\mathbf{x}_k; \boldsymbol{\lambda}_k], t_k; t_{k+1}) = [\mathbf{x}_{k+1}; \boldsymbol{\lambda}_{k+1}]$ is imposed for each subinterval. Such approach improves rejection of Hamiltonian systems ill-conditioning and their *hyper-sensitivity* to initial conditions.

Indirect collocation An alternative approach to the above mentioned consists in transcribing the dynamics of the Hamiltonian system and solve the associated BVP through a collocation technique; collocation was indeed traditionally thought of as a way to solve Boundary Value Problems [52].

Direct methods are based on state and control discretization and approximation, and attempt to minimize the OCP objective function; they are distinguished between *control parameterization methods*, which approximate the controls only, and *state and control parametrization methods* which approximate both controls and states. The following approaches are commonly adopted [77].

Direct shooting Such methods belong to *control parameterization methods*, as the control is parameterized using a specific functional form with coefficients that are determined by the optimization process. Such coefficients constitute then the components of the NLP vector.

In the same fashion as for indirect methods, either a *single shooting* or a *multiple shooting* can be adopted; in the second case the continuity is reinforced at subintervals boundaries only for states.

Direct collocation Direct collocation methods versatility has allowed them to gain popularity in the field of Optimal Control, and they usually parameterize both states and controls. For sparsity reasons, local collocation is usually adopted, with high number of nodes preferred over high-order polynomials to approximate states and controls.

3.2 Pseudospectral methods

3.2.1 Method overview

pseudospectral methods build the solution of an optimal control problem as a series expansion, and the most commonly employed is Lagrange decomposition.

Global collocation

Let us firstly normalize physical time t to pseudotime τ , according to the affine mapping

$$\tau = \frac{2}{t_f - t_0}t - \frac{t_f + t_0}{t_f - t_0} \quad t \in [t_0, t_f] \quad (3.2)$$

which maps $[t_0, t_f]$ to $[-1, 1]$. Notice τ is scaled from t according to the factor $\frac{2}{t_f - t_0}$. Let the time domain be discretized in $n + 1$ nodes t_i , with $i = 0, \dots, n$. The generic state $\mathbf{x}(t)$ is then approximated by a summation of $n + 1$ Lagrange polynomials $\tilde{L}(t)$ of degree n according to

$$\mathbf{x}(t) \approx \sum_{i=0}^n \mathbf{X}_i \tilde{L}_i(t) \quad \text{where} \quad \tilde{L}_i(t) = \prod_{\substack{k=0 \\ k \neq i}}^n \frac{t - t_k}{t_i - t_k} \quad (3.3)$$

and $\mathbf{X}_i = \mathbf{x}(t_i)$ as Lagrange polynomials satisfy the *isolation property*.

With respect to traditional collocation schemes, the Lagrange expansion allows to identify the n_c collocation points as a subset of the $n + 1$ nodes, simplifying the relationship between the unknowns and the differential approximation. The *pseudospectral differentiation matrix* $\mathbf{D} \in \mathbb{R}^{n_s n_c \times n_s(n+1)}$ can be defined over the normalized pseudotime domain, independently on the value of final time t_f . Considering a zero-based notation, Its j^{th} block row and i^{th} block column component $\mathbf{D}_{j,i} \in \mathbb{R}^{n_s \times n_s}$ results

$$\mathbf{D}_{j,i} = \mathbf{J}_{\mathbf{X}_i} \mathbf{X}'_j = \frac{d\tilde{L}_i}{d\tau}(\tau_j) \mathbb{I}_{n_s} = \frac{t_f - t_0}{2} \dot{\tilde{L}}_i(t_j) \mathbb{I}_{n_s} \quad (3.4)$$

where $(\bullet)'$ denotes derivative of quantity \bullet with respect to τ . It is being referred to block rows and columns rather than rows and columns since normalized state pseudotime derivatives depend only on the considered collocation point, thus they are equal with each other at each collocation point. For fixed discretization of pseudotime domain, \mathbf{D} does neither depend on t_0 , nor on t_f : transformation in Sec. 3.2.1 is indeed affine. If control $\mathbf{U}_j = \mathbf{u}(t_j) \in \mathbb{R}^u$ is added and RHS is supposed time-independant, Eq. (3.1) can be indeed re-elaborated as

$$\frac{2}{t_f - t_0} \mathbf{D}_j \mathbf{X} - \mathbf{f}(\mathbf{X}_j, \mathbf{U}_j) = \mathbf{0} \quad (3.5)$$

where $\mathbf{D}_j = [\mathbf{D}_{j,0}, \dots, \mathbf{D}_{j,n}]$ is the differentiation matrix for the $(j+1)^{\text{th}}$ collocation point and $\mathbf{X} \in \mathbb{R}^{n_s(n+1)}$ gathers states at nodes.

At last, similarly to Eq. (2.21), the generic running cost $\mathcal{L}(\mathbf{x}, \mathbf{u}, t)$ can be reformulated according to

$$\int_{t_0}^{t_f} \mathcal{L}(\mathbf{x}, \mathbf{u}) dt \approx \frac{t_f - t_0}{2} \sum_{i=0}^n w_i \mathcal{L}(\mathbf{X}_i, \mathbf{U}_i, t_i) \quad (3.6)$$

using the integration weights vector \mathbf{w} , which depends on the choice of nodes, analyzed in Sec. 3.2.2.

Local collocation

When a local collocation approach is pursued, time domain is divided in h segments and solution is approximated as summation of piecewise Lagrange polynomials, since an independent collocation is performed over each segment. Time-dependant functions are then patched at the *knots*, i.e. the inner time bounds.

Let the superscript s , with $s = 1, \dots, h$, denote the s^{th} segment; then time domain $[t_0, t_f]$ can be divided in h subdomains $[t_0^{(s)}, t_n^{(s)}]$, such that $t_n^{(s-1)} = t_0^{(s)}$. h pseudotime subdomains can be then defined according to

$$\tau^{(s)} = \frac{2}{t_n^{(s)} - t_0^{(s)}} \left[t - t_0^{(s)} \right] - \frac{t_n^{(s)} + t_0^{(s)}}{t_n^{(s)} - t_0^{(s)}} \quad t \in [t_0^{(s)}, t_n^{(s)}] \quad (3.7)$$

thus normalizing each segment over its own duration. In such manner, if each segment is characterized by the same number of nodes $n + 1$, the segmental differential matrix $\mathbf{D}^{(s)}$ can be evaluated once for all. Let

$$\begin{array}{ll} \mathbf{X}_i^{(s)} & \mathbf{U}_i^{(s)} & s = 1, \dots, h \\ & & i = 0, \dots, n \end{array} \quad (3.8)$$

denote the generic state and control at $(i + 1)^{\text{th}}$ node, over the s^{th} segment. At knots states shall be patched, thus $\mathbf{X}_n^{(s-1)} = \mathbf{X}_0^{(s)}$. Moreover, Eq. (3.5) is rewritten as

$$\frac{2}{t_n^{(s)} - t_0^{(s)}} \mathbf{D}_j \mathbf{X}^{(s)} - \mathbf{f} \left[\mathbf{X}_j^{(s)}, \mathbf{U}_j^{(s)} \right] = \mathbf{0} \quad \text{with } s = 1, \dots, h \quad (3.9)$$

as well as Eq. (3.6), which becomes

$$\int_{t_0}^{t_f} \mathcal{L}[\mathbf{x}(t), \mathbf{u}(t), t] dt \approx \sum_{s=1}^h \frac{t_n^{(s)} - t_0^{(s)}}{2} \sum_{i=0}^n w_i \mathcal{L}[\mathbf{X}_i^{(s)}, \mathbf{U}_i^{(s)}, t_i^{(s)}] \quad (3.10)$$

At this stage it is useful to introduce the different types of nodes, as their specific features guide the choice of the scheme for the present thesis.

3.2.2 Survey of Legendre-type nodes

pseudospectral methods are characterized by specific positions of collocation points along a domain: such points coincide with the roots of a polynomial belonging to a family of orthogonal polynomials, defined over the prescribed domain. Each set of collocation points is further associated to a different quadrature rule, thus to quadrature weights. Moreover, since every collocation point is a node, unknowns too are located at roots of such orthogonal polynomials; this implies that for specific orthogonal polynomials, the KKT conditions of a direct pseudospectral method match the discretized optimality conditions of the original OCP. Such result represents a major breakthrough in optimal control theory: it establishes a connection between two methods, direct and indirect ones, that have traditionally been opposed against each other.

Among the orthogonal polynomials, Legendre-type ones allow to match, at least partially, the KKT and the discrete optimality conditions: a Legendre polynomial L_n of n^{th} degree satisfies the following *Sturm-Liouville* equation

$$[(1 - \tau^2) L_n'(\tau)]' + n(n + 1)L_n(\tau) = 0 \quad (3.11)$$

and Legendre-type collocation points are obtained as roots of Legendre polynomials, linear combinations of Legendre polynomials of different orders, or their derivatives.

Lobatto nodes: Historically speaking, *Legendre-Gauss-Lobatto* (LGL) nodes have been firstly employed [29]; among the $n + 1$ collocation points over the pseudotime domain, first and last collocation points are $\tau_0 = -1$, $\tau_n = 1$, while the remaining ones, namely τ_k , with $k = 1, \dots, n - 1$, are the $n - 1$ roots of $L_n'(\tau)$. Nodes completely coincide with collocation points. Such choice allows to collocate dynamics at both time domain bounds, providing the most intuitive approach when looking for the

optimal control profile over an assigned time domain. On the other hand, the KKT conditions relative to multipliers at domain bounds couple dynamics and boundary conditions, thus determining noisy costates and a suboptimal solution.

Gauss nodes: For this last reason, research focus has moved towards *Legendre-Gauss* (LG) nodes [78]; the n collocation points for such approach are the n roots of $L_{n+1}(\tau)$ and do not include neither $\tau = -1$, nor $\tau = 1$, while $\tau_0 = -1$ is included among nodes. They avoid the aforementioned problem affecting LGL nodes, and feature perfect matching between the KKT and discrete optimality conditions. On the other hand, dynamics is not collocated at the initial time bound, thus control can not be optimized on the overall time domain; moreover states and control at final time are not available as optimization variables, thus boundary conditions can not be reinforced completely during the optimization.

Radau nodes: Most recent advances have tried and merge the positive aspects of the previous approaches, leading to the *Legendre-Gauss-Radau* (LGR) nodes [79]: $n + 1$ nodes include $\tau_n = 1$ and the n roots of $L_{n+1}(\tau) + L_n(\tau)$, while collocation points miss $\tau = 1$. Such architecture allows for precise mapping of the KKT conditions and discrete optimality conditions, while ensuring boundary conditions can be imposed and satisfied. Since LGR do not allow collocation at final time, the *flipped Legendre-Gauss-Radau* (fLGR) nodes have been proposed; collocation points and nodes are symmetric with respect to the LGR ones: they are respectively the n roots of $L_{n+1}(\tau) - L_n(\tau)$ and the collocation points with the addition of $\tau_0 = -1$.

Nodes and collocation points are represented in Fig. 3.1: it can be noticed that LGR type nodes are not symmetric with respect to the pseudotime domain center, as well as that LG nodes are missing the final time bound. In addition, LGL is the only nodes group for which the number of collocation points equals number of nodes.

At this stage, it is evident LGR and fLGR nodes offer an excellent trade-off to develop a hybrid and efficient optimal control algorithm: the following subsections therefore build onto the choice of Radau nodes.

3.2.3 Radau direct collocation and CMT

The mathematical structure developed in Sec. 3.2.1 can now be specialized to a global Radau-type collocation scheme. Analysis is led for a global collocation scheme for simplicity, but can be generalized with tools provided in Sec. 3.2.1.

Lagrange polynomials can be evaluated for the LGR and fLGR nodes, as well as the

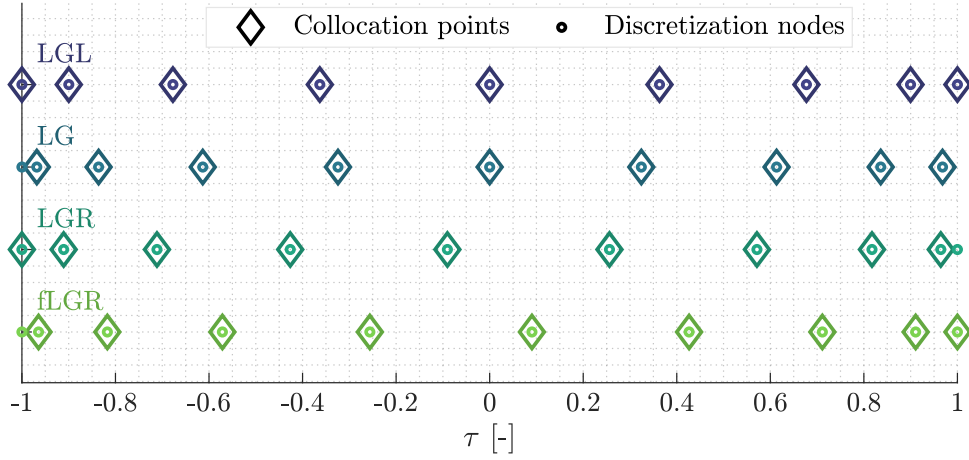


Figure 3.1: Visual representation of Legendre-type nodes - 9 collocation points

differentiation matrix \mathbf{D} . Both approaches include $\tau_0 = -1$ and $\tau_n = 1$ among the $n + 1$ discretization nodes, thus subscripts 0 and n refer to such time bounds.

Remark 3.2: Both LGR and fLGR have, given n collocation points, $n + 1$ nodes. Due to its own definition, matrix \mathbf{D} is rectangular, with 1 block column more with respect to the block rows.

The quadrature weights in Eq. (3.6) are instead defined by

$$w_i^{\text{LGR}} = \begin{cases} \frac{2}{n^2} & i = 0 \\ \frac{(1 - \tau_i)}{n^2 L_{n-1}(\tau_i)^2} & i = 1, \dots, n-1 \\ 0 & i = n \end{cases} \quad w_i^{\text{fLGR}} = w_{n-i}^{\text{LGR}} \quad i = 0, \dots, n \quad (3.12)$$

Remark 3.3: Both LGR and fLGR feature a null weight at the node where dynamics is not collocated. Let us consider a running cost dependant on the control: it features null derivative with respect to the final control or to the initial one, respectively for the LGR nodes and the fLGR nodes. If no equality constraint on control at such instant is applied, the problem becomes singular, and this control value can not be uniquely determined by the optimizer; this problem is said to be *unregulated*.

Remark 3.4: Similarly to nodes, weights for LGR and fLGR are symmetric; the respective KKT conditions are then symmetric as well, thus the two schemes are completely equivalent. This allows to treat only the LGR scheme, simplifying the following analysis.

Let us reconsider the general NLP problem in Eq. (2.24). For our purposes, the equality constraints gathered in vector \mathbf{h} can be specified as *dynamics constraints* and as *final boundary conditions*, while the inequality, gathered in \mathbf{g} , can be specified as *pure control constraints* \mathbf{g}_u . Moreover, it is hypothesized for simplicity that $t_0 = t_0^* = 0$, $\mathbf{X}_0 = \mathbf{X}_0^*$ and that final condition constraints are independent on controls.

Running cost in Eq. (3.6), exploiting weights outlined in Eq. (3.12), can be plugged into the general definition of NLP cost in Eq. (2.21), providing

$$\bar{\mathcal{J}}(\mathbf{X}, \mathbf{U}, t) = \phi(\mathbf{X}_f, t_f) + \frac{t_f}{2} \sum_{i=0}^n w_i^{LGR} \mathcal{L}(\mathbf{X}_i, \mathbf{U}_i, t_i) \quad (3.13)$$

The, the NLP for a LGR pseudospectral method results

$$\min_{\mathbf{U} \in \mathbb{R}^{u(n+1)}} \bar{\mathcal{J}}(\mathbf{X}, \mathbf{U}, t) \quad \text{s.t.} \quad \left\{ \begin{array}{l} \frac{2}{t_f} \mathbf{D}_i \mathbf{X} - \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i) = \mathbf{0} \\ \mathbf{g}_u(\mathbf{U}_i, t_i) \preceq \mathbf{0} \\ \psi_f(\mathbf{X}_f, t_f) = \mathbf{0} \end{array} \right\} \quad i = 0, \dots, n-1 \quad (3.14)$$

Let us now introduce \mathbf{N}_f , the multipliers associated with final conditions, \mathbf{M}_i , the multipliers related to pure control constraints and $\mathbf{\Lambda}_i$, the multipliers related dynamical constraints. \mathbf{M}_i and $\mathbf{\Lambda}_i$ are referred to instant t_i . Similarly, function explicit dependancies are dropped and substituted with the subscript i .

The *Lagrangian* \mathbf{L} of Eq. (3.14) reads then

$$\mathbf{L} = \phi_f + \mathbf{N}_f^\top \psi_f + \frac{t_f}{2} \sum_{i=0}^{n-1} \left(w_i \mathcal{L}_i + \mathbf{M}_i^\top \mathbf{g}_{u,i} \right) - \sum_{i=0}^{n-1} \mathbf{\Lambda}_i^\top \left(\mathbf{D}_i \mathbf{X} - \frac{t_f}{2} \mathbf{f}_i \right) \quad (3.15)$$

where dynamical constraints have been rescaled to mirror the quadrature applied exactly in the context of the continuous-time OCP and the superscript LGR has been dropped from weights for notation simplification.

Karush-Kuhn-Tucker conditions can be then retrieved. They read

$$\begin{aligned}
& \text{for } i = 1, \dots, n-1 & \frac{2}{t_f} \mathbf{D}_i^T \boldsymbol{\Lambda} &= w_i \nabla_{\mathbf{X}_i} \mathcal{L}_i + \mathbf{J}_{\mathbf{X}_i}^T \mathbf{f}_i \boldsymbol{\Lambda}_i \\
& \text{for } i = 0, \dots, n-1 & \left\{ \begin{array}{l} \mathbf{0} = w_i \nabla_{\mathbf{U}_i} \mathcal{L}_i + \mathbf{J}_{\mathbf{U}_i}^T \mathbf{f}_i \boldsymbol{\Lambda}_i + \mathbf{J}_{\mathbf{U}_i}^T \mathbf{g}_{u,i} \mathbf{M}_i \\ \frac{2}{t_f} \mathbf{D}_i \mathbf{X} = \mathbf{f}(\mathbf{X}_i, \mathbf{U}_i) \\ \mathbf{0} = \mathbf{M}_i \text{ if } \mathbf{g}_{u,i} \prec \mathbf{0} \\ \mathbf{0} \succ \mathbf{M}_i \text{ if } \mathbf{g}_{u,i} = \mathbf{0} \end{array} \right. & \\
& & \mathbf{D}_n^T \boldsymbol{\Lambda} &= \nabla_{\mathbf{X}_f} \phi_f + \mathbf{J}_{\mathbf{X}_f}^T \boldsymbol{\psi}_f \mathbf{N}_f \\
& & \mathbf{0} &= \boldsymbol{\psi}_f \\
& \frac{\partial}{\partial t_f} \left[\frac{t_f}{2} \sum_{i=0}^{n-1} (w_i \mathcal{L}_i + \boldsymbol{\Lambda}_i^T \mathbf{f}_i + \mathbf{M}_i^T \mathbf{g}_{u,i}) \right] &= -\frac{\partial \phi_f}{\partial t_f} - \mathbf{N}_f^T \frac{\partial \boldsymbol{\psi}_f}{\partial t_f}
\end{aligned} \tag{3.16}$$

where \mathbf{D}_i^T is $(i+1)^{\text{th}}$ row of matrix \mathbf{D}^T , thus explaining the presence of \mathbf{D}_n^T . Lagrange multipliers in Eq. (3.16) stem from the same primal problem as continuous costates and multipliers, outlined in the OCP formulation as BVP, in Eq. (2.17). However, in the first case domain is discretized and optimality conditions are then introduced for the discrete problem; in the second, optimality conditions are introduced for the original problem, and transcription can be developed on the primal-dual formulation afterwards. Such difference proves fundamental, since, *discretization and optimization do not commute* [73]: an additional mapping is necessary to link dual variables of the discrete-time problem with discrete dual variables of the continuous-time problem.

A Covector Mapping Theorem for the LGR collocation scheme is derivable by transcribing dynamics of the Hamiltonian system in Eq. (2.17) with the same LGR collocation scheme outlined for the primal system. Let $\tilde{\boldsymbol{\lambda}}_i$ and $\tilde{\boldsymbol{\mu}}_i$ as costates and path multipliers at $(i+1)^{\text{th}}$ collocation point and \mathbf{D}^\dagger as the differentiation operator for the costates. Since dynamics constraints are only reinforced at the n collocation points, costates are approximated by a polynomial of $(n-1)^{\text{th}}$ degree; this requires differentiation matrix to fit such degree, namely \mathbf{D}^\dagger . Moreover, the final costate $\tilde{\boldsymbol{\lambda}}_f$ can only be extrapolated. The approximated continuous-time optimality conditions,

then, read

$$\begin{aligned}
& \text{for } i = 1, \dots, n-1 \quad -\frac{2}{t_f} \mathbf{D}_i^\dagger \tilde{\boldsymbol{\lambda}} = \nabla_{\mathbf{x}} \mathcal{L}_i + \mathbf{J}_{\mathbf{x}}^\top \mathbf{f}_i \tilde{\boldsymbol{\lambda}}_i \\
& \text{for } i = 0, \dots, n-1 \quad \left\{ \begin{array}{l} \mathbf{0} = \nabla_{\mathbf{u}} \mathcal{L}_i + \mathbf{J}_{\mathbf{u}}^\top \mathbf{f}_i \tilde{\boldsymbol{\lambda}}_i + \mathbf{J}_{\mathbf{u}}^\top \mathbf{g}_{u,i} \tilde{\boldsymbol{\mu}}_i \\ \frac{2}{t_f} \mathbf{D}_i \mathbf{X} = \mathbf{f}_i \\ \mathbf{0} = \tilde{\boldsymbol{\mu}}_i \quad \text{if } \mathbf{g}_{u,i} \prec \mathbf{0} \\ \mathbf{0} \succ \tilde{\boldsymbol{\mu}}_i \quad \text{if } \mathbf{g}_{u,i} = \mathbf{0} \end{array} \right. \quad (3.17) \\
& \tilde{\boldsymbol{\lambda}}_f = \nabla_{\mathbf{x}_f} \phi_f + \mathbf{J}_{\mathbf{x}_f}^\top \boldsymbol{\psi}_f \boldsymbol{\nu}_f \\
& \mathbf{0} = \boldsymbol{\psi}_f \\
& \frac{\partial}{\partial t_f} \left[\frac{t_f}{2} \sum_{i=0}^{n-1} w_i (\mathcal{L}_i + \tilde{\boldsymbol{\lambda}}_i^\top \mathbf{f}_i + \tilde{\boldsymbol{\mu}}_i^\top \mathbf{g}_{u,i}) \right] = -\frac{\partial \phi_f}{\partial t_f} - \boldsymbol{\nu}_f^\top \frac{\partial \boldsymbol{\psi}_f}{\partial t_f}
\end{aligned}$$

where \mathcal{H}_f is obtained deriving with respect to t_f the quadrature of \mathcal{H} over the collocation points.

It can be proven that \mathbf{D}^\dagger is a differential operator for the space of polynomials of degree $n-1$ if its components are built according to the following equations [79]

$$\begin{aligned}
\mathbf{D}_{0,0}^\dagger &= -\mathbf{D}_{0,0} - \frac{1}{w_0} \mathbb{I}_{n_s} \\
\mathbf{D}_{i,j}^\dagger &= -\frac{w_j}{w_i} \mathbf{D}_{j,i} \quad \text{otherwise}
\end{aligned} \quad (3.18)$$

Let \mathbf{W} be the block diagonal matrix of quadrature weights w_i such that $\mathbf{W}_{i,i} = w_i \mathbb{I}_{n_s}$, then Eq. (3.18) implies that

$$\mathbf{D}_i^\top = -w_i \mathbf{D}_i^\dagger \mathbf{W}^{-1} \quad \implies \quad \frac{2}{t_f} \mathbf{D}_i^\top \boldsymbol{\Lambda} = -w_i \frac{2}{t_f} \mathbf{D}_i^\dagger \tilde{\boldsymbol{\lambda}} \quad (3.19)$$

Simply speaking, second equation in Eq. (3.19) implies the approximate costate dynamics is equivalent to the dual KKT conditions, provided multipliers are scaled with the quadrature weights. The mapping in Eq. (3.20), thus, constitutes the

Covector Mapping Theorem for the LGR collocation scheme².

$$\begin{aligned}
 \tilde{\lambda}_f &= \mathbf{D}_n^T \boldsymbol{\Lambda} \\
 \tilde{\lambda}_i &= \frac{\Lambda_i}{w_i} \quad i = 0, \dots, n-1 \\
 \tilde{\mu}_i &= \frac{\mathbf{M}_i}{w_i} \quad i = 0, \dots, n-1 \\
 \boldsymbol{\nu}_f &= \mathbf{N}_f
 \end{aligned} \tag{3.20}$$

Application of Eq. (3.20) allows to exactly match Eq. (3.16) and Eq. (3.17). It therefore proves the existence of a link between the indirect and direct formulations of the OCP [29]: an accurate estimation of multipliers for a direct method can be then used to guess the costates of the equivalent indirect formulation.

Remark 3.5: Hypothesis of fixed initial states suit the problem analyzed in such work for a global collocation type: in case of local collocation, initial conditions are known for the first segment only. Nonetheless, CMT in Eq. (3.20) is still valid if initial time and states are free, as well as if mixed constraints are present [79].

While embedding optimality necessary conditions, Radau direct collocation still lies among the high-order direct methods, and this determines two important setbacks that shall be accounted for. At first, the control policy is an optimization variable, thus its optimal value is determined by the optimization process, without explicit dependancy on states and costates; such dependancy can be embedded in an indirect method, with benefits over final accuracy and computational cost. In second instance, the discrete nature of the direct method implies a suboptimal solution when dealing with bang-bang type profiles, if the specific structure of the solution is not accounted for. In such case an indirect method allows for precise detection of switching positions, thus representing the most intuitive choice to accurately solve a bang-bang problem.

3.2.4 Radau indirect collocation

A fast yet accurate method to solve OCPs is represented by the concept of indirect collocation; the approach presented previously for direct methods can be easily

²A similar expression can be retrieved for the fLGR collocation scheme, still affine in costates and continuous time multipliers

extended to the dual states.

Let us consider a global collocation with $n + 1$ nodes. Costates are treated with the same approach as states, thus

$$\boldsymbol{\lambda}(t) \approx \sum_{i=0}^n \tilde{\boldsymbol{\lambda}}_i \tilde{L}_i(t) \quad (3.21)$$

and the differentiation matrices of primal and dual variables coincide. For such formulation the block differentiation matrix $\mathbf{D}_{j,i}$ results $\mathbf{D}_{j,i} \in \mathbb{R}^{2n_s \times 2n_s}$.

The nodal states and costates estimations can be grouped in the nodal *augmented state* $\mathbf{Y}_i \in \mathbb{R}^{2n_s}$, as well as the *augmented right hand side* $\mathbf{F} : \mathbb{R}^{2n_s} \rightarrow \mathbb{R}^{2n_s}$ can be assembled. They respectively read

$$\mathbf{Y}_i = \begin{bmatrix} \mathbf{X}_i \\ \tilde{\boldsymbol{\lambda}}_i \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} \nabla_{\boldsymbol{\lambda}} \tilde{\mathcal{H}} \\ -\nabla_{\mathbf{x}} \tilde{\mathcal{H}} \end{bmatrix} \quad (3.22)$$

Applying the PMP, the optimal nodal control \mathbf{U}_i^* results $\mathbf{U}_i^* = \mathbf{U}_i^*(\mathbf{Y}_i)$ and the nodal Hamiltonian $\tilde{\mathcal{H}}_i = \tilde{\mathcal{H}}_i(\mathbf{Y}_i)$.

The continuous time EL conditions in Eq. (2.7) is then rearranged, providing the following discrete time BVP, where $\mathbf{Y} \in \mathbb{R}^{2n_s(n+1)}$.

$$\text{Find } [\mathbf{Y}^T, t_0, t_f] \quad \text{s.t.} \quad \begin{cases} \frac{2}{t_f - t_0} \mathbf{D}_j \mathbf{Y} - \mathbf{F}(\mathbf{Y}_j) = \mathbf{0} \\ \text{for } j = 0, \dots, n-1 \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{X}_0 = \mathbf{x}_0^* \\ t_0 = t_0^* \\ \tilde{\boldsymbol{\lambda}}_n = \nabla_{\mathbf{x}_f} \Phi_f \\ \tilde{\mathcal{H}}_n = -\frac{\partial \Phi_f}{\partial t_f} \end{cases} \quad (3.23)$$

Remark 3.6: Problem in Eq. (3.23) is well posed provided the Jacobian of the dynamical and of boundary defects is not singular, if a simple Newton-Raphson method is employed; boundary conditions, indeed, constitute $2n_s + 2$ constraints, while dynamical constraints make up for $2n_s n$ constraints. Vector $[\mathbf{Y}^T, t_0, t_f]$ is therefore determined univocally.

3.3 OCP formulations wrap-up

The whole developed mathematical plant is graphically summarized in Fig. 3.2 [73]. More specifically, B refers to the common problem formulation, and the superscripts indicate variants of the problem itself:

- **Problem B** is the original Optimal Control Problem, as stated in Eq. (2.12): no numerical method is introduced, and the formulation is in the continuous time.
- **Problem B^λ** is the BVP dual formulation of problem B , as formulated in Eq. (2.17). The problem is still in its continuous time formulation, and the optimal control profile is coupled to states and costates thanks to the control equation/PMP; it is the formulation solved by *indirect methods*.
- **Problem B^N** is the NLP form of problem B , thus the formulation where states and controls are restricted to their nodal values. It corresponds to the formulation in Eq. (2.24), therefore it is the one approached by *direct methods*. Its solution is closer to the one of B the more accurate the discretization scheme.
- **Problem $B^{N\lambda}$** is the zero-finding version of B^N , as multipliers are introduced and associated KKT conditions are outlined, as in Eq. (3.16). It represents the set of equations most commercial softwares [58, 80] effectively solve, and directly provides the unknown control profile.
- **Problem $B^{\lambda N}$** is the discrete time version of B^λ , as nodes are introduced and optimality conditions are imposed at such points, as in Eq. (3.17) or in Eq. (3.23). Its solution is closer to one of B^λ the more accurate the discretization scheme. The control profile is retrieved from optimality necessary conditions after problem has been solved.

The exact solution of the original problem B coincides with the one from B^λ . Moreover, optimality conditions for B actually define B^λ , which then configures as the problem to be solved. The most direct way to do this, namely $B \rightarrow B^\lambda$, is often followed and it is highlighted in red in Fig. 3.2; yet initialization of costates and dynamical system ill-conditioning constitute relevant problems, since such approach coincides with a pure indirect method.

An alternative is represented by the longest path, $B \rightarrow B^N \rightarrow B^{N\lambda} \rightarrow B^{\lambda N} \rightarrow B^\lambda$,

which exploits the CMT (e.g. Eq. (3.20)) of the employed discretization scheme, therefore using a direct method as guess generation mechanism for the indirect one; the process is represented by the blue path in Fig. 3.2. Such approach allows to bypass the problems of a purely indirect method employing a direct one, constituting a *hybrid method* embedding accuracy and robustness [72].

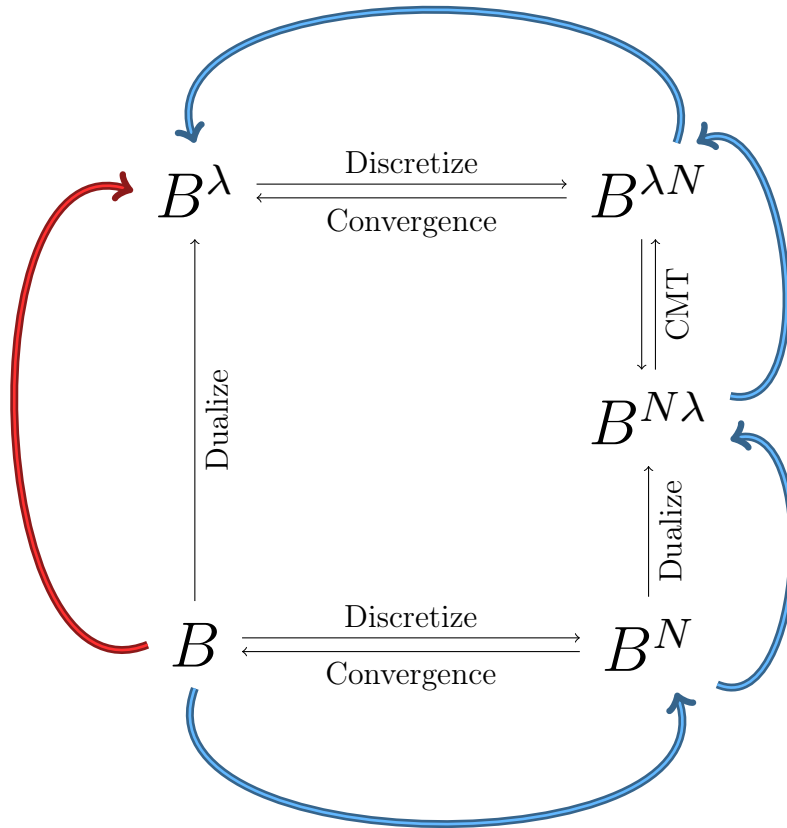


Figure 3.2: Summary of Optimal Control Problem formulations

In red: Pure indirect strategy

In blue: Hybrid strategy

4 | Optimal Aerodynamic Powered Landing Problem

La curiosità dove porta
È pericolosa

Mattia Ricci, *via Whatsapp*TM

Landing a probe on a body surface is a complex task that requires performing precise maneuvers over a short time horizon. In addition, maneuvers authority represents a relevant requirement for the control system.

The present work focuses on planetary *powered pinpoint landing*:

In a nutshell

Pinpoint landing consists of targeting a precise surface location, with final prescribed velocity with respect to the landing site. Full state knowledge is therefore required, as well as full system controllability. A *powered landing* employs thrust to reach final state; aim of thrust program is typically the minimization of the overall fuel consumption.

Pinpoint landing is the only reliable strategy if null landing position error is targeted. More generic landing approaches, indeed, do not allow to directly control the final position, *e.g.* the *soft landing* technique imposes null velocity at touchdown, but does not account for touchdown location [81].

Landing phase requires targeting the final landing site as well as damping the kinetic energy possessed by the vehicle. These operations shall deal with an upper bound over thrust, and a lower limit on thrust itself, due to limited engine throttleability. In such framework the optimal thrust program is represented by a *bang-off-bang* profile in the most general case.

In a nutshell

A *bang-bang* profile is a command program characterized by alternating maximum and minimum control magnitude arcs. The *switching times* are bounds of such arcs, thus marking discontinuities in control program.

In the specific case of a 3-D rocket landing scenario, at maximum 2 full thrust arcs are required, the former to redirect the platform towards landing site, the latter to brake before touchdown; the intermediate is an idle thrust arc [81]. An off-bang profile is instead sufficient if initial velocity is approximately headed towards the landing site.

In such scenario, aerodynamic forces are often included as disturbance. However, if aerodynamic forces were exploited correctly, one could further optimize fuel consumption: *drag* damps kinetic energy, and could thus minimize the duration of the last full-thrust arc; *lift* acts as centripetal force deflecting trajectory, and could minimize or even substitute the first full thrust arc.

An indirect *single shooting* is preliminarily employed to assess such claim; a continuation scheme based on a *double homotopy* augments such indirect formulation to increase its robustness.

In a nutshell

Two functions are *homotopic* if one can be continuously deformed into the other; such deformation is called *homotopy* [82].

In our case, a first homotopy deforms the objective function, turning the \mathcal{C}^1 optimal thrust program to discontinuous; a second introduces the aerodynamic forces, therefore modifying the switches location.

The present chapter is articulated as follows: Sec. 4.1 outlines the aerodynamic powered pinpoint landing problem, and in Sec. 4.2 the PMP is applied to the problem itself. The double homotopic approach is discussed in Sec. 4.3, while Sec. 4.4 draws the physical results due to addition of aerodynamic forces and the performances of the presented indirect method.

4.1 Problem formulation

Landing phase is usually started at a distance negligible with respect to planet curvature radius; *flat Earth* model is therefore employed. In addition, a 3-D model

is sought for in the context of this work: a landing site-topocentric cartesian reference frame is accounted for, of type *East-North-Up*, with z -axis directed upwards; chosen initial conditions force trajectory to lie on a curved surface, preventing from any dimensional reduction to simplify the problem, as graphically represented in Fig. 4.1.

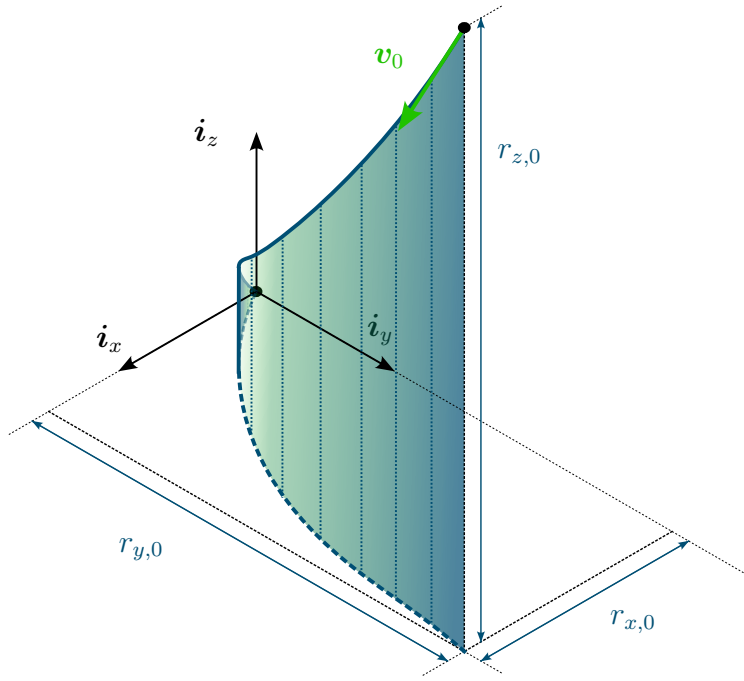


Figure 4.1: Trajectory not reducible to cartesian 2-D analysis

With such hypotheses, initial and final conditions $\mathbf{x}_0, \mathbf{x}_f \in \mathbb{R}^7$ and gravity vector $\mathbf{g} \in \mathbb{R}^3$ read

$$\mathbf{x}_0 = \begin{bmatrix} \mathbf{r}_0 \\ \dots \\ \mathbf{v}_0 \\ \dots \\ m_0 \end{bmatrix} = \left. \begin{array}{l} \left. \begin{bmatrix} 1200 \\ 600 \\ 5000 \end{bmatrix} \right\} m \\ \left. \begin{bmatrix} -10 \\ 0 \\ -300 \end{bmatrix} \right\} m/s \\ \left. \begin{bmatrix} 1000 \end{bmatrix} \right\} kg \end{array} \right\} \mathbf{x}_f = \begin{bmatrix} \mathbf{r}_f \\ \mathbf{v}_f \\ m_f \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \text{free} \end{bmatrix} \quad \mathbf{g} = \begin{bmatrix} 0 \\ 0 \\ -9.81 \end{bmatrix} m/s^2 \quad (4.1)$$

Mass depletion is modelled assuming constant specific impulse I_{sp} from the thruster; maximum thrust T_{max} is hypothesized not to depend on height.

For what concerns controls, thrust direction is identified by $\mathbf{u}_T \in \mathbb{R}^3$, and thrust acceleration is therefore directed along such vector. \mathbf{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 *body axis direction* $\mathbf{u}_b \in \mathbb{R}^3$ as function of velocity direction \mathbf{v}/v : a first rotation about \mathbf{v}/v of angle σ transforms the *cross-wind* direction $\hat{\mathbf{i}}_{\text{cw}}$ to $\hat{\mathbf{i}}_{\text{cw}}$, then rotation about $\hat{\mathbf{i}}_{\text{cw}}$ of angle α_{eff} rotates \mathbf{v}/v to \mathbf{u}_b . σ and α_{eff} , therefore, are equivalent to \mathbf{u}_b .

Equation (4.2) defines the *lift coefficient* C_L for an axisymmetric body and the polar model linking the *drag coefficient* C_D to C_L .

$$C_L(\alpha_{\text{eff}}) = C_{L/\alpha}\alpha_{\text{eff}}, \quad C_D(\alpha_{\text{eff}}) = C_{D,0} + kC_L^2(\alpha_{\text{eff}}) \quad (4.2)$$

For easiness, air density is modelled through the following exponential relationship, although US76 model [83] could be used for better accuracy.

$$\rho(r_z) = \rho_0 \exp\left(-\frac{r_z}{H}\right) \quad (4.3)$$

Complete dynamics, therefore, read

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{r} \\ \dot{\mathbf{v}} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \frac{T_{\text{max}}}{m}\mathbf{u}_T + \mathbf{g} + \frac{\mathbf{D}(r_z, \mathbf{v}, \alpha_{\text{eff}})}{m} + \frac{\mathbf{L}(r_z, \mathbf{v}, \alpha_{\text{eff}}, \sigma)}{m} \\ -\frac{T_{\text{max}}}{I_{\text{sp}}g_0}u_T \end{bmatrix} = \mathbf{f}(\mathbf{x}, \mathbf{u}_T, \alpha_{\text{eff}}, \sigma) \quad (4.4)$$

where

$$\begin{aligned} \mathbf{D}(r_z, \mathbf{v}, \alpha_{\text{eff}}) &= -\frac{1}{2}\rho(r_z)S_{\text{ref}}C_D(\alpha_{\text{eff}})\mathbf{v}\mathbf{v} \\ \mathbf{L}(r_z, \mathbf{v}, \alpha_{\text{eff}}, \sigma) &\approx \frac{1}{2}\rho(r_z)S_{\text{ref}}C_{L/\alpha}[\mathbf{v} \otimes \mathbf{u}_b(\alpha_{\text{eff}}, \sigma)] \otimes \mathbf{v} \end{aligned} \quad (4.5)$$

Total angle of attack, defined as positive [84], is limited to α_{max} ; Sec. 4.2 outlines how limitation on α_{eff} constrains control \mathbf{u}_b . Thrust authority is bounded as well due to limits on mass depletion rates, accounting for a maximum engine throttleability of 30%. Thus

$$\begin{aligned} \alpha_{\text{eff}} &\leq \alpha_{\text{max}} &= 15^\circ \\ 0.3 &= u_{T,\text{min}} \leq u_T \leq u_{T,\text{max}} &= 1 \end{aligned} \quad (4.6)$$

Aim of thrust program is optimizing the final mass, or, equivalently, minimizing the thrust acceleration integral: this second approach, indeed, physically corresponds to optimizing the final logarithmic mass, equivalent to the the first approach as the natural logarithm is a monotonic function [37]. For reasons clarified in Chap. 5, optimization of thrust acceleration integral is pursued, leading to the following objective function

$$\mathcal{J}(\mathbf{x}, \mathbf{u}_T, t_f) = \int_0^{t_f} \frac{1}{m} \frac{T_{max}}{I_{sp} g_0} u_T dt \quad (4.7)$$

The problem can be therefore formulated as in Eq. (4.8)

$$\min_{\alpha_{eff}, \sigma, \mathbf{u}_T, t_f} \mathcal{J}(\mathbf{x}, \mathbf{u}_T, t_f) \quad \text{s.t.} \quad \begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}_T, \alpha_{eff}, \sigma) \\ \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{x}(t_f) = \mathbf{x}_f \\ \alpha_{eff} \leq \alpha_{max} \\ u_{T,min} \leq u_T \leq u_{T,max} \end{cases} \quad (4.8)$$

Thruster and aerodynamic parameters are summarized in Tab. 4.1.

Table 4.1: Parameters summary

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	[-]

Remark 4.1: In such scenarios, a commonly employed reference frame is the topocentric *Downrange-Crossrange-Altitude* (DCA): Downrange direction connects initial position to final landing site, Altitude is orthogonal to planet surface and headed upwards and Crossrange completes the right-hand reference frame. The

optimal initial velocity is indeed headed towards landing site, thus allowing a 2-D formulation of the pinpoint landing problem. The interest for a 3-D model, however, is the reason why such approach is discarded.

Remark 4.2: Thrust direction is constrained to the body axis due to limited TVC authority; such limitation, however, imposes strict requirements over body direction, not allowing for an optimal exploitation of the aerodynamic forces. In addition, x-body axis required to point upwards due to attitude control limitations, thus thrust should feature positive z component. Inclusion of such second constraint is left for future analyses. \mathbf{u}_b and \mathbf{u}_T are left unlinked, and \mathbf{u}_T unconstrained.

4.2 Solution derivation

Application of Pontryagin's Minimum Principle allows to retrieve the optimal control profile for the introduced problem. Since Eq. (4.6) draws pure control constraints, multipliers associated to such constraints can be ignored in the reduced Hamiltonian $\tilde{\mathcal{H}}$. For the analyzed problem, $\tilde{\mathcal{H}}$ reads

$$\tilde{\mathcal{H}} = \boldsymbol{\lambda}_r^T \mathbf{v} + \boldsymbol{\lambda}_v^T \mathbf{g} + \underbrace{\frac{T_{max}}{m} \boldsymbol{\lambda}_v^T \mathbf{u}_T + \frac{T_{max}}{I_{sp} g_0} \left(\frac{1}{m} - \lambda_m \right) u_T}_{\text{Thrust-related}} + \underbrace{\frac{1}{m} \boldsymbol{\lambda}_v^T [\mathbf{D}(\alpha_{eff}) + \mathbf{L}(\alpha_{eff}, \sigma)]}_{\text{Aerodynamic-related}} \quad (4.9)$$

where the dependency of \mathbf{D} and \mathbf{L} on aerodynamic angles has been highlighted. PMP imposes optimizing the thrust related terms and the aerodynamic related ones. Since such two terms are independent, the optimization can be performed separately.

4.2.1 Optimization of thrust control

Let $\tilde{\mathcal{H}}_T$ be the thrust-dependent term of the Hamiltonian. \mathbf{u}_T denotes the normalized thrust orientation, and can be expressed as product of its norm u_T and its direction \mathbf{i}_T . Moreover, let us define the *primer vector* as the normalized coveLOCITY $\mathbf{p}_v = \boldsymbol{\lambda}_v / \lambda_v$. Since T_{max}/m and u_T are positive, one shall minimize the product $\boldsymbol{\lambda}_v^T \mathbf{i}_T$ to minimize the Hamiltonian. Therefore, according to Lawden's theory [85], it shall be

$$\mathbf{i}_T = -\mathbf{p}_v \quad \text{thus} \quad \tilde{\mathcal{H}}_T = \frac{T_{max}}{I_{sp} g_0} \underbrace{\left(\frac{1}{m} - \lambda_m - \frac{I_{sp} g_0}{m} \lambda_v \right)}_{S_f} u_T \quad (4.10)$$

The bracketed term S_f is called *switching function*: to minimize $\tilde{\mathcal{H}}_T$, optimal thrust magnitude u_T^* shall stick to the following thrust program, essentially dictated by the sign of the switching function S_f

$$u_T^* = \begin{cases} u_{T,\min} & \text{if } S_f > 0 \\ u_{T,\max} & \text{if } S_f < 0 \\ \in (u_{T,\min}, u_{T,\max}) & \text{if } S_f = 0 \end{cases} \quad (4.11)$$

which corresponds to a *bang-bang* control profile is condition $S_f = 0$ holds almost everywhere.

Remark 4.3: If the third condition in Eq. (B.3) is verified over a finite interval of time $[t_1, t_2]$, with $t_1 \neq t_2$, u_T can not be determined with the PMP. Such *singularity* is solved employing a generalization of the second-order optimality condition, called *generalized Legendre-Clebsch condition* [86]. The second-order optimality condition has been ignored in such work since finite maxima are not present in standard trajectory optimization problems. Moreover, literature shows rocket landing problems do not usually suffer from control singularity [87]; generalized Legendre-Clebsch condition is therefore bypassed.

4.2.2 Optimization of aerodynamic controls

Let $\tilde{\mathcal{H}}_a$ be the aerodynamics-dependent term of the Hamiltonian. The control angles σ and α_{eff} are optimized in two steps, firstly σ and α_{eff} in second instance.

Bank angle optimization

The optimal *bank angle* program maintains body axis direction \mathbf{u}_b in the plane defined by \mathbf{p}_v and velocity \mathbf{v}^1 [88].

To understand this, let us firstly consider the drag only; it is possible to rearrange \mathbf{p}_v as $\mathbf{p}_v = \mathbf{p}_{v\parallel} + \mathbf{p}_{v\perp}$, the component $\mathbf{p}_{v\parallel}$ being parallel to \mathbf{v} , $\mathbf{p}_{v\perp}$ being orthogonal

¹Bank angle control gets singular if $\mathbf{p}_v \otimes \mathbf{v} = 0$, but such condition is never verified exactly in a 3-D model

to \mathbf{v} . The problem and the solution are outlined followingly.

$$\min_D \mathbf{p}_v^T \mathbf{D} \quad \text{therefore} \quad \begin{cases} p_{v\parallel} v < 0 & \xrightarrow{\text{Eq. (4.5)}} & p_{v\parallel} D > 0 & \implies & \text{minimize } D \\ p_{v\parallel} v > 0 & \xrightarrow{\text{Eq. (4.5)}} & p_{v\parallel} D < 0 & \implies & \text{maximize } D \end{cases} \quad (4.12)$$

\mathbf{D} does not depend on σ , thus $p_{v\parallel} D$ is not regulated with σ .

Let us now add the lift: the function to be minimized becomes $\mathbf{p}_v^T (\mathbf{D} + \mathbf{L}) = p_{v\parallel} D + \mathbf{p}_{v\perp}^T \mathbf{L}$. If lift \mathbf{L} is orthogonal to plane defined by \mathbf{p}_v and \mathbf{v} , then $\mathbf{p}_{v\perp}^T \mathbf{L} = 0$: results obtained for the drag-only analysis are unchanged. However, if σ is not bounded, there always exists an interval $[\sigma_1, \sigma_2]$ such that 1) $\mathbf{p}_{v\perp}^T \mathbf{L}(\sigma^*) < 0 \forall \sigma^* \in (\sigma_1, \sigma_2)$, 2) $\mathbf{p}_{v\perp}^T \mathbf{L}(\sigma_1) = \mathbf{p}_{v\perp}^T \mathbf{L}(\sigma_2) = 0$, 3) $\sigma^* = (\sigma_1 + \sigma_2)/2$ minimizes $\mathbf{p}_{v\perp}^T \mathbf{L}(\sigma^*)$. Procedure is graphically reported in Fig. 4.2.

Therefore there always exists an optimal σ minimizing the contribution of the aerodynamic forces on the Hamiltonian. Imposing the optimal σ corresponds to 1) imposing \mathbf{u}_b to lie in the plane defined by \mathbf{p}_v and \mathbf{v} , 2) guaranteeing that \mathbf{L} obtained with positive angle of attack satisfies $\mathbf{p}_{v\perp}^T \mathbf{L} < 0$. Optimization of α_{eff} is thus reduced to a 2-D analysis, on a half plane.

Total angle of attack optimization

Let us consider the plane defined by \mathbf{p}_v and \mathbf{v} . Optimal \mathbf{L} and \mathbf{D} have been demonstrated to lie on such plane. Let ς be the angle between $-\mathbf{p}_v$ and \mathbf{v} . Moreover, let us consider the polar model outlined in Eq. (4.2). If $0 < \varsigma < \pi/2$, the optimal α_{eff} minimizes the projection of $\mathbf{D} + \mathbf{L}$ on \mathbf{p}_v provided that

$$\tan \varsigma = \frac{\partial D}{\partial L} \stackrel{\text{Eq. (4.2)}}{=} 2kC_{L/\alpha} \alpha_{\text{eff}}^* \quad \text{therefore} \quad \alpha_{\text{eff}}^* = \frac{\tan \varsigma}{2kC_{L/\alpha}} \quad (4.13)$$

For $\pi/2 \leq \varsigma < \pi$ the optimal α_{eff} is unbounded since the relationship between C_D and C_L is parabolic.

Adding the pure control constraint in Eq. (4.6), the optimal angle of attack program satisfies

$$\alpha_{\text{eff}}^* = \begin{cases} \frac{\tan \varsigma}{2kC_{L/\alpha}} & \text{if } \tan \varsigma < 2kC_{L/\alpha} \alpha_{\text{max}} \\ \alpha_{\text{max}} & \text{if } \tan \varsigma \geq 2kC_{L/\alpha} \alpha_{\text{max}} \end{cases} \quad (4.14)$$

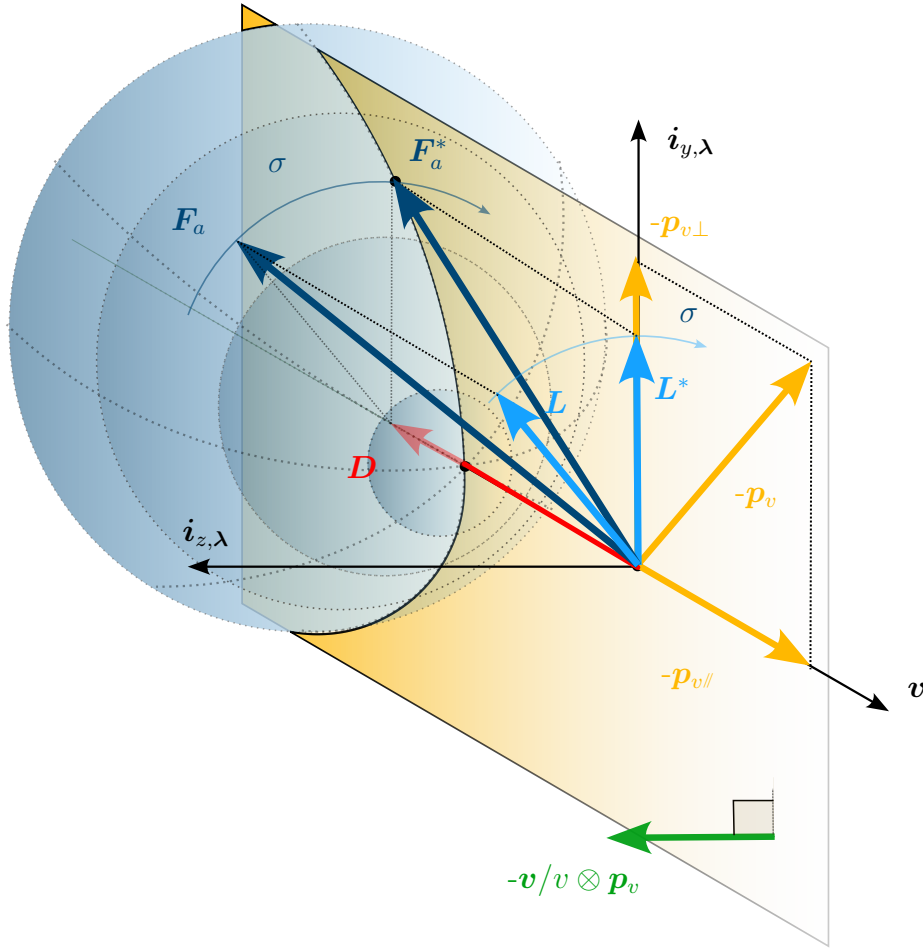


Figure 4.2: Generic L and optimal L^* associated with the same D and obtained for different σ . Generic aerodynamic force F_a and optimal one F_a^* move on the paraboloid associated with the aerodynamic polar, in blue. L^* minimizes $\mathbf{p}_v^T F_a^*$

As outlined before, optimal bank angle σ^* is imposed by finding the corresponding optimal \mathbf{u}_b^* . Therefore, using Rodrigues' formula, it results

$$\begin{cases} \mathbf{e}^* = -\frac{\mathbf{v}}{v} \otimes \mathbf{p}_v \\ \mathbf{u}_b^* = \frac{1}{v} [\cos \alpha_{\text{eff}} \mathbf{v} + \sin \alpha_{\text{eff}} (\mathbf{e} \otimes \mathbf{v}) + (1 - \cos \alpha_{\text{eff}}) (\mathbf{e} \cdot \mathbf{v}) \mathbf{e}] \end{cases} \quad (4.15)$$

4.2.3 Dual problem formulation

At last, let us introduce the TPBVP that is solved, the dual version of Eq. (4.8). Controls are expressed as function of primal and dual variables using Eq. (4.10), Eq. (B.3), Eq. (4.13) and Eq. (4.15). Therefore $\mathbf{f}(\mathbf{x}, \mathbf{u}_T, \alpha_{\text{eff}}, \sigma) = \mathbf{f}(\mathbf{x}, \boldsymbol{\lambda})$, and

$\mathbf{f}_\lambda(\mathbf{x}, \boldsymbol{\lambda}) \doteq -\nabla_{\mathbf{x}} \tilde{\mathcal{H}}(\mathbf{x}, \boldsymbol{\lambda})$. In addition $\mathbf{y}(t) \doteq [\mathbf{x}(t)^\top, \boldsymbol{\lambda}(t)^\top]^\top$, and $\mathbf{F}(\mathbf{x}, \boldsymbol{\lambda}) \doteq [\mathbf{f}(\mathbf{x}, \boldsymbol{\lambda})^\top, \mathbf{f}_\lambda(\mathbf{x}, \boldsymbol{\lambda})^\top]^\top$. The TPBVP formulation for the presented problem results then

$$\text{Find } \mathbf{y}(t), t_f \text{ s.t. } \dot{\mathbf{y}} = \mathbf{F}(\mathbf{y}) \text{ and } \begin{cases} \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{r}(t_f) = \mathbf{r}_f \\ \mathbf{v}(t_f) = \mathbf{v}_f \\ \boldsymbol{\lambda}_m(t_f) = 0 \\ \tilde{\mathcal{H}}(t_f) = 0 \end{cases} \quad (4.16)$$

A single shooting approach can be employed to solve Eq. (4.16); given the flow of the controlled primal-dual system $\boldsymbol{\varphi}(\mathbf{y}_0, t_0; t) = \mathbf{y}(t)$, the problem is reformulated as

$$\text{Find } \boldsymbol{\lambda}_0, t_f \text{ such that } \boldsymbol{\varphi}(t) \text{ satisfies } \begin{cases} \mathbf{r}(t_f) = \mathbf{r}_f \\ \mathbf{v}(t_f) = \mathbf{v}_f \\ \boldsymbol{\lambda}_m(t_f) = 0 \\ \tilde{\mathcal{H}}(t_f) = 0 \end{cases} \quad (4.17)$$

Boundary conditions correspond to nonlinear constraints with respect to $\boldsymbol{\lambda}_0$ and t_f ; $\boldsymbol{\lambda}_0$ and t_f shall be therefore guessed.

4.3 Double homotopic scheme

Finding an educated guess for $\boldsymbol{\lambda}_0$ represents a problem, as costates lack of clear physical meaning. Moreover, it has been experimentally verified that solving Eq. (4.17) is difficult with a *multi-start technique*, namely randomly guessing $\boldsymbol{\lambda}_0$ and t_f . A double continuation scheme is therefore adopted²: a first continuation scheme, feasible for a first initialization with random number generation, steps from a quadratic objective function to the linear one [90, 91]; a second scheme introduces aerodynamic forces [92] with a linear scaling.

²In literature, the concept of *double homotopy* commonly refers to employment of two different continuations schemes to tackle the same problem [89]; in our case, instead, two different problems are solved in series, using different continuation rules

Objective function homotopy Let $\varepsilon \in [0, 1]$ denote the *homotopic parameter* such that the generic objective of the homotopic scheme \mathcal{J}_ε results

$$\mathcal{J}_\varepsilon \doteq \int_0^{t_f} \left[\varepsilon \left(\frac{1}{m} \frac{T_{max}}{I_{sp} g_0} u_T \right) + (1 - \varepsilon) \left(\frac{1}{m} \frac{T_{max}}{I_{sp} g_0} u_T \right)^2 \right] dt \quad (4.18)$$

Therefore $\mathcal{J}_{\varepsilon_{EO}}$, with $\varepsilon_{EO} = 0$, has a running cost quadratic with respect to the control, while $\mathcal{J}_{\varepsilon_{AO}}$, with $\varepsilon_{AO} = 1$, corresponds to the original objective function in Eq. (4.7). Therefore one can weight the quadratic running cost and the linear one changing ε . Substituting \mathcal{J} with \mathcal{J}_ε in Eq. (4.8) determines a different structure of the optimal thrust program with respect to Eq. (B.3). Lawden's theory is indeed still valid, but $\tilde{\mathcal{H}}_{T,\varepsilon}$ is quadratic in u_T ; control becomes of bang-continuous-bang type. Laws dictating thrust program are reported in Appendix B.

Aerodynamic forces homotopy Let $\bar{S}_{ref} \in [0, 1]$ denote the non-dimensional reference surface. Dynamics right-hand side \mathbf{f} can be generalized by $\mathbf{f}_{\bar{S}_{ref}}$ such that the acceleration terms $\mathbf{f}_{\bar{S}_{ref},v}$ read

$$\mathbf{f}_{\bar{S}_{ref},v} = \frac{T_{max}}{m} \mathbf{u}_T + \mathbf{g} + \bar{S}_{ref} \frac{\mathbf{D} + \mathbf{L}}{m} \quad (4.19)$$

Velocity terms and the mass term are left unchanged. Similarly as before, $\bar{S}_{ref} = 1$ corresponds to the initial dynamics; on the other hand $\bar{S}_{ref} = 0$ corresponds to dynamics without aerodynamic forces: substituting \mathbf{f} with $\mathbf{f}_{\bar{S}_{ref}}$ in Eq. (4.8) alters the magnitude of aerodynamic forces, without modifying the structure of the corresponding aerodynamic-related hamiltonian $\tilde{\mathcal{H}}_{a,\bar{S}_{ref}}$. Optimal α_{eff}^* and σ^* programs are therefore left unmodified.

At this stage, functions with subscripts $\varepsilon, \bar{S}_{ref}$ stem from the generalized Hamiltonian $\tilde{\mathcal{H}}_{\varepsilon,\bar{S}_{ref}}$, reading

$$\tilde{\mathcal{H}}_{\varepsilon,\bar{S}_{ref}} = \boldsymbol{\lambda}_r^T \mathbf{v} + \boldsymbol{\lambda}_v^T \mathbf{g} + \tilde{\mathcal{H}}_{T,\varepsilon} + \tilde{\mathcal{H}}_{a,\bar{S}_{ref}} \quad (4.20)$$

$B_{\varepsilon, \bar{S}_{ref}}^\lambda$ thus denotes the generalized primal-dual TPBVP, outlined followingly

$$\text{Find } \boldsymbol{\lambda}_0, t_f \text{ such that } \boldsymbol{\varphi}_{\varepsilon, \bar{S}_{ref}}(t) \text{ satisfies } \begin{cases} \mathbf{r}(t_f) = \mathbf{r}_f \\ \mathbf{v}(t_f) = \mathbf{v}_f \\ \lambda_m(t_f) = 0 \\ \tilde{\mathcal{H}}_{\varepsilon, \bar{S}_{ref}}(t_f) = 0 \end{cases} \quad (4.21)$$

The followed strategy is outlined in Fig. 4.3. $\varepsilon^{(i)}$, with $i = 0, \dots, n$, is the generic

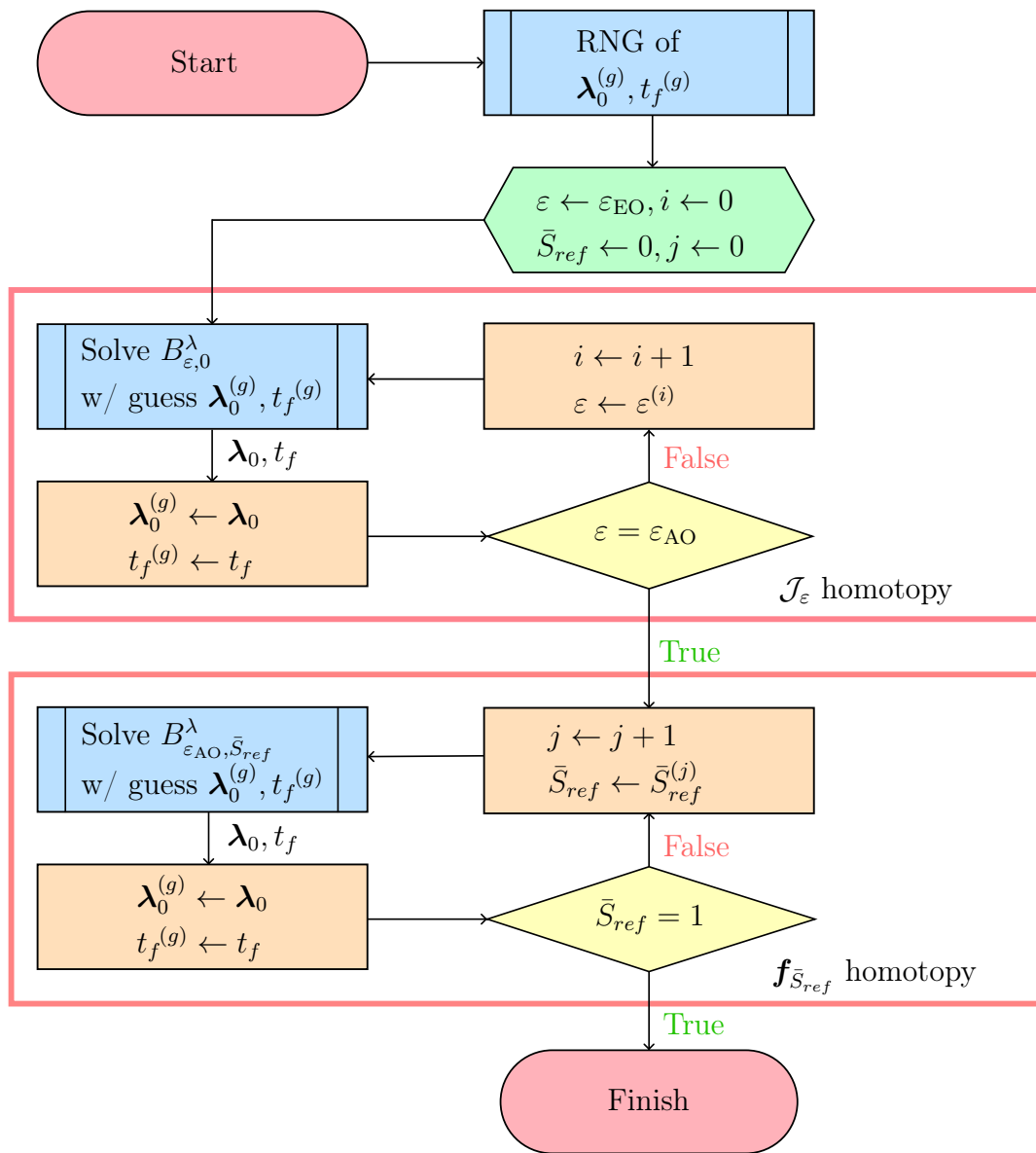


Figure 4.3: Double homotopy structure

term of the monotonic series $\{\varepsilon^{(i)}\}$ with $\varepsilon^{(0)} = \varepsilon_{EO}, \varepsilon^{(n)} = \varepsilon_{AO}$; similarly $\bar{S}_{ref}^{(j)}$, with $j = 0, \dots, m$, is the generic term of the monotonic series $\{\bar{S}_{ref}^{(j)}\}$ with $\bar{S}_{ref}^{(0)} = 0, \bar{S}_{ref}^{(m)} = 1$. Convergence of each subproblem is guaranteed by the correct spacing within series $\{\varepsilon^{(i)}\}$ and $\{\bar{S}_{ref}^{(j)}\}$: a logarithmic scale is employed for the former [93], while a linear scaling is sufficient for the latter.

4.4 Solution structure validation

In the present section the results and iterations of both homotopic schemes are reported. The zero-finding problem associated to the TPBVP is solved at each step employing MATLAB's Optimization Toolbox `fsolve`; dynamics is propagated employing the variable-order Adam–Bashfort–Moulton scheme; switching detection is embedded in the dynamics function, therefore a multi-step integrator better handles control discontinuity.

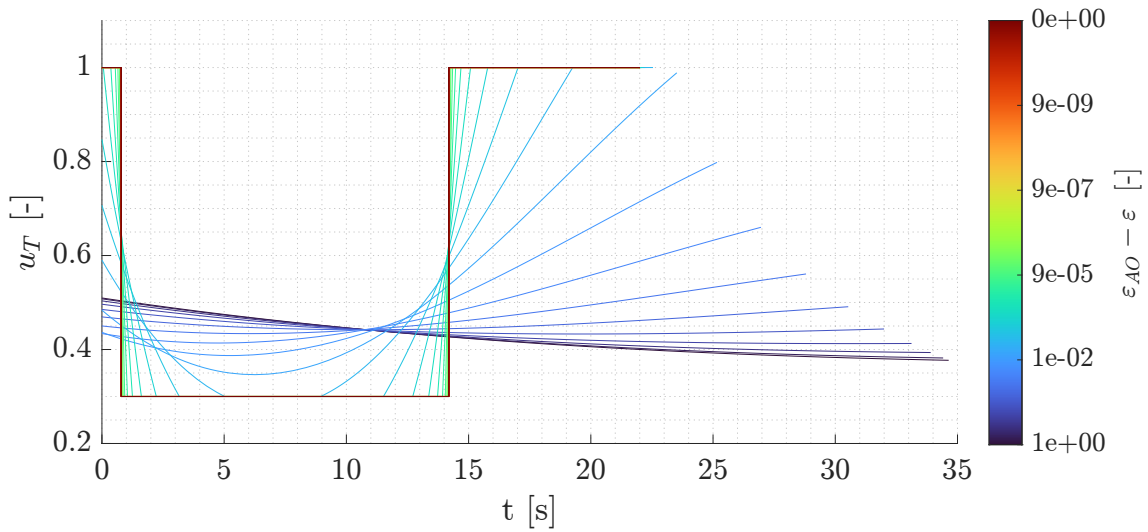


Figure 4.4: \mathcal{J}_ε homotopic continuation scheme for control profile

4.4.1 Objective function homotopic scheme

As discussed, the first performed step is the homotopy of the objective function \mathcal{J}_ε . Iterations reported in Fig. 4.4 confirm how the optimal control profile passes from continuous to *bang-idle-bang*: the blue control profile corresponds to ε satisfying $\varepsilon_{AO} - \varepsilon = 1$; therefore ε coincides with $\varepsilon_{EO} \doteq 0$. Similarly, the dark red control

profile corresponds to ε satisfying $\varepsilon_{AO} - \varepsilon = 0$, *i.e.* $\varepsilon = \varepsilon_{AO} \doteq 1$.

Mass iterations for the optimal control profile are shown in Fig. 4.5a: mass $m(t)$ is not \mathcal{C}^1 , as it presents corner points at control switches. Similar corner points are observed in the comass $\lambda_m(t)$ as well, in Fig. 4.5b.

Remark 4.4: Such non regularities remain in the successive homotopy; particular attention shall be paid to them, as their presence may imply accuracy reduction for the employed collocation scheme.

Remark 4.5: Since dynamic variables are in the same order of magnitude, dynamics are integrated without adimensionalization: λ_m is dual with respect to m , not m_{nd} ; for representation purposes, however, the latter and other physical non-dimensional quantities have been employed.

4.4.2 Dynamics homotopic scheme

In the representations that follow the first homotopy passages are reported in light grey to allow for a precise comparison of how the optimal solution evolves.

It is evident how the optimal control (Fig. 4.6) moves from a *bang-idle-bang* structure to an *idle-bang* structure: as expected, aerodynamic forces allow to alleviate thrust utilization. This, in turn, reflects on the trajectory; as represented in Fig. 4.9, the fuel optimal path gets distorted from an approximately straight one: the rocket tends to maximize the downwards acceleration to maximize the influence of aerodynamic forces³ and exploit them to deflect trajectory and damp kinetic energy. For this reason, α_{eff} is not maximized in the first seconds of descent: this trades-off initial aerodynamic braking for energy damping at lower altitude, with greater speed and thicker atmosphere. α_{eff} along the continuation scheme is represented in Fig. 4.7.

The outcome of such fuel sparing is a non-negligible save in the required fuel mass, as reported in Fig. 4.8; with respect to the case without aerodynamic forces, final mass sparing amounts to over the 4% of the initial one. Comparing such strategy with a case with null α_{eff} and drag only, improvement of nearly 2% of the final mass is observed. On the other hand, an important countereffect arises, as visualized in Fig. 4.10: inclusion and optimization of aerodynamic forces directs the primer vector \mathbf{p}_v upwards, therefore optimal thrust is oriented downwards. This is evidently not compatible with common rocket pointing requirements: rocket shall always head upwards to allow for the vertical landing to be completed safely. An overview of the

³Indeed $D \approx v^2$, $L \approx v^2$

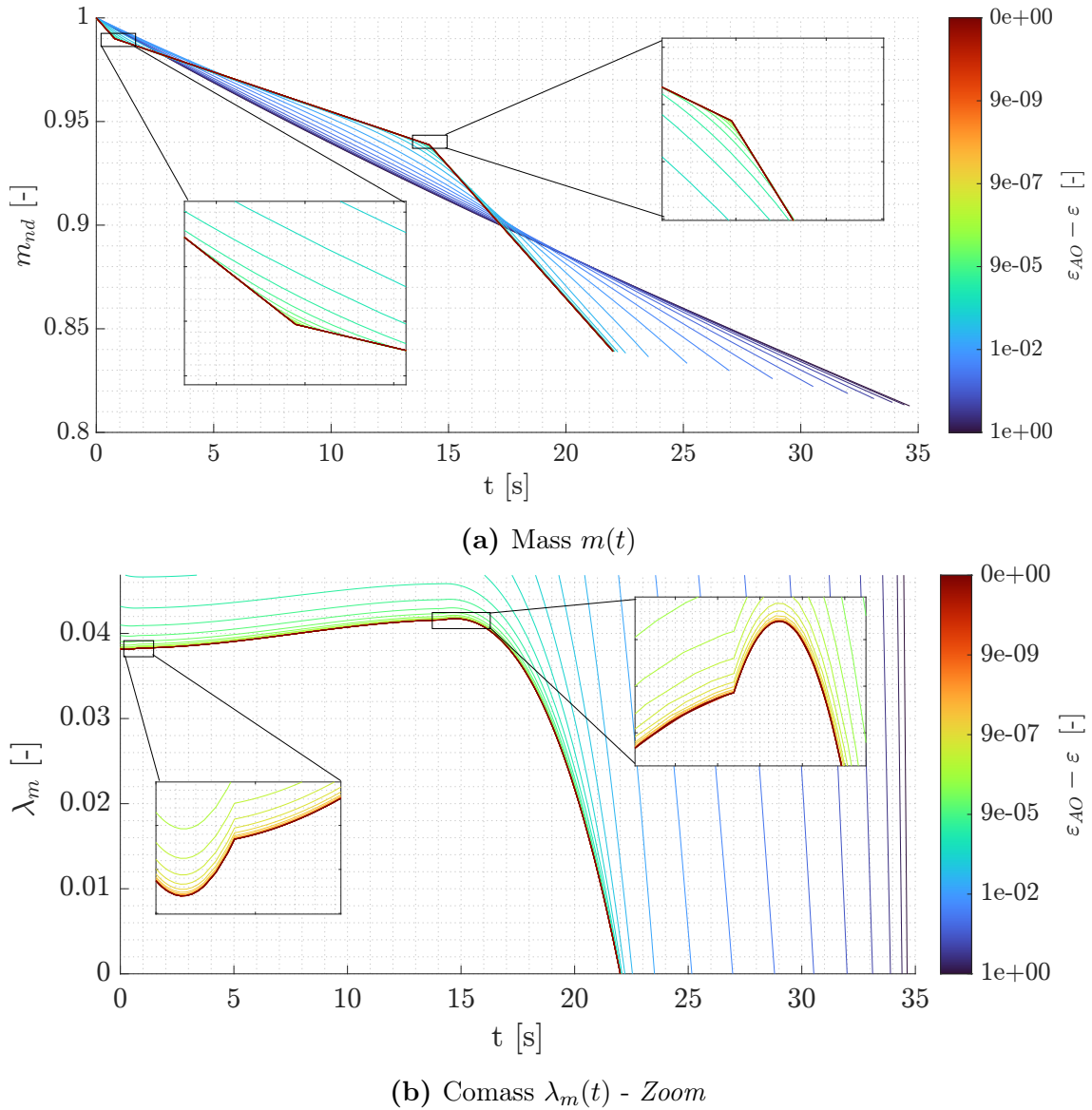


Figure 4.5: \mathcal{J}_ε homotopic continuation scheme for optimal non \mathcal{C}^1 quantities

strategies to counteract such problem is discussed in Chap. 6. In the drag-only case, instead, thrust is always directed upwards.

4.4.3 Summary

Merit figures, along with additional information, are reported in Tab. 4.2. Optimal direction of the body axis \mathbf{u}_b^* has been discarded: limitation of α_{\max} and hypothesis of rocket body cylindrical shape guarantee upwards pointing.

Despite limitations associated with thrust direction, the aerodynamic powered land-

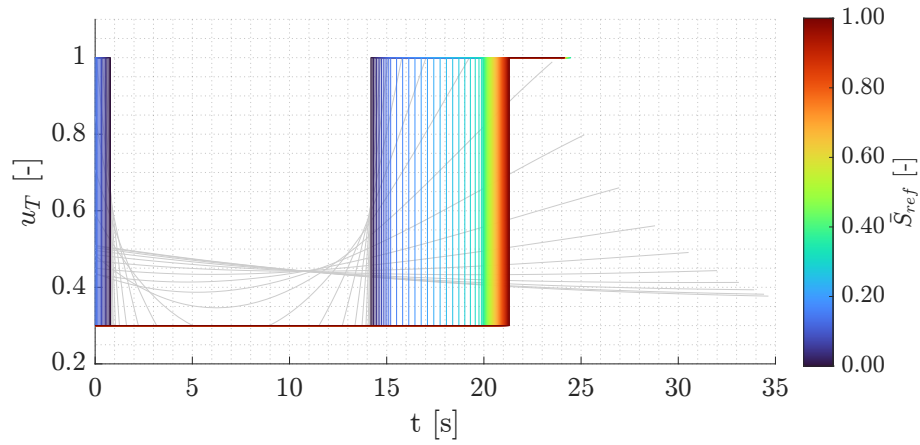


Figure 4.6: $f_{\tilde{S}_{ref}}$ homotopic continuation scheme for thrust magnitude profile

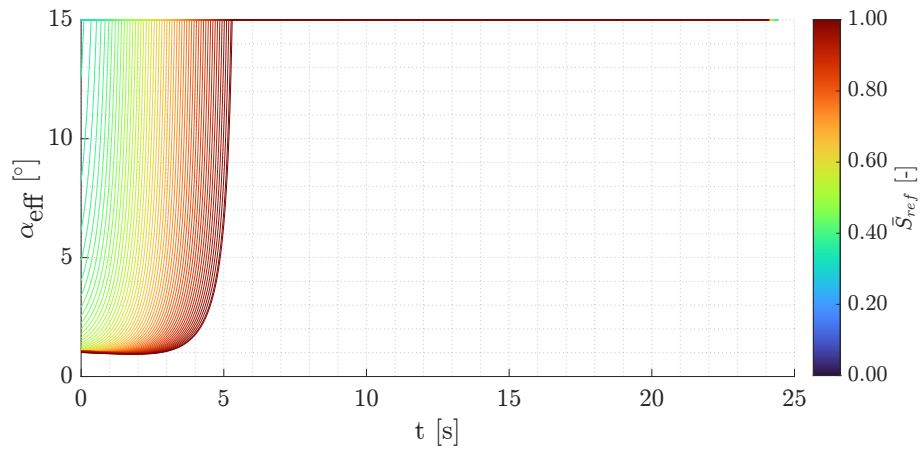


Figure 4.7: $f_{\tilde{S}_{ref}}$ homotopic continuation scheme for α_{eff} profile

ing problem structure offers, if solved, a strong proof of concept for an innovative algorithm: highly nonlinear dynamics and non regularity of the dynamic quantities represent challenging features for a collocation scheme. The following chapter, therefore, builds on the knowledge acquired within the present one to develop a hybrid collocation scheme.

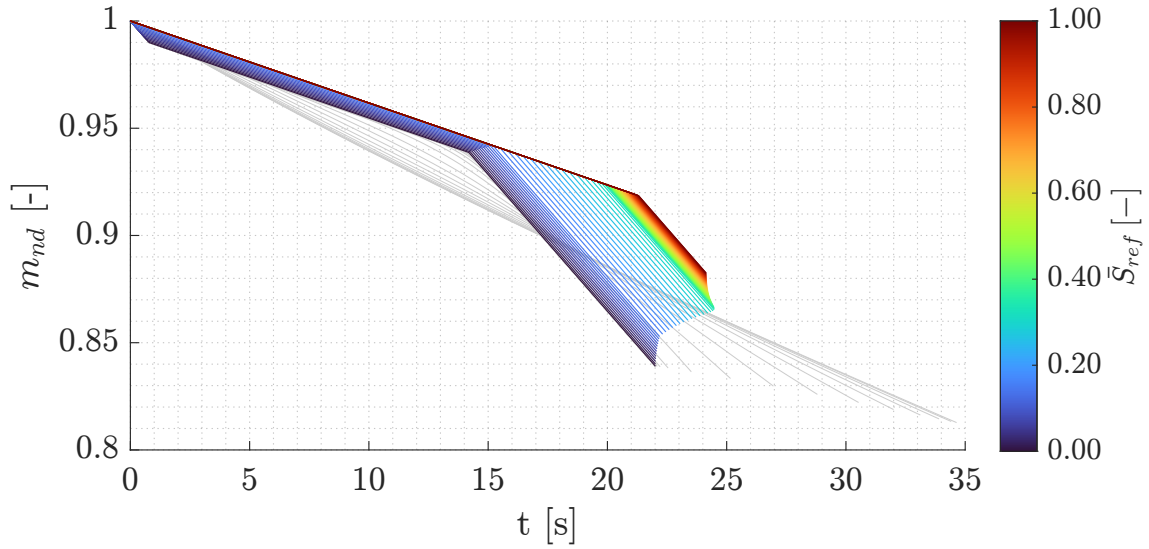


Figure 4.8: $f_{\bar{S}_{ref}}$ homotopic continuation scheme for mass $m(t)$

Table 4.2: Results summary

Dynamics	u_T profile	α_{eff} profile	m_f [kg]	t_f [s]	t_{sw} [s]
No aerodynamics	<i>Bang-idle-bang</i>	-	839.1	22.01	14.19
Drag only	<i>Idle-Bang</i>	Reinforced null	864.3	24.05	19.14
Full aerodynamics	<i>Idle-Bang</i>	<i>Continuous-Bang</i>	882.5	24.14	21.31

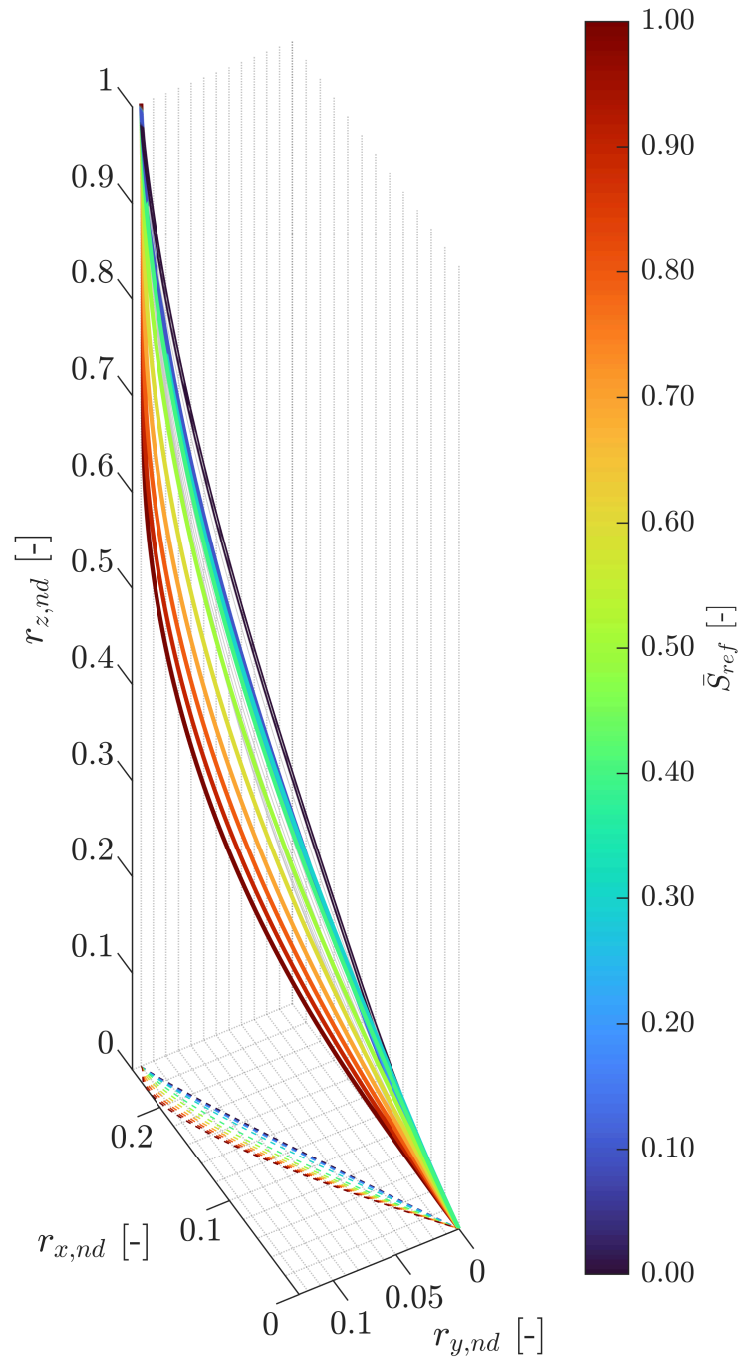


Figure 4.9: $f_{\bar{S}_{ref}}$ homotopic continuation scheme for trajectory

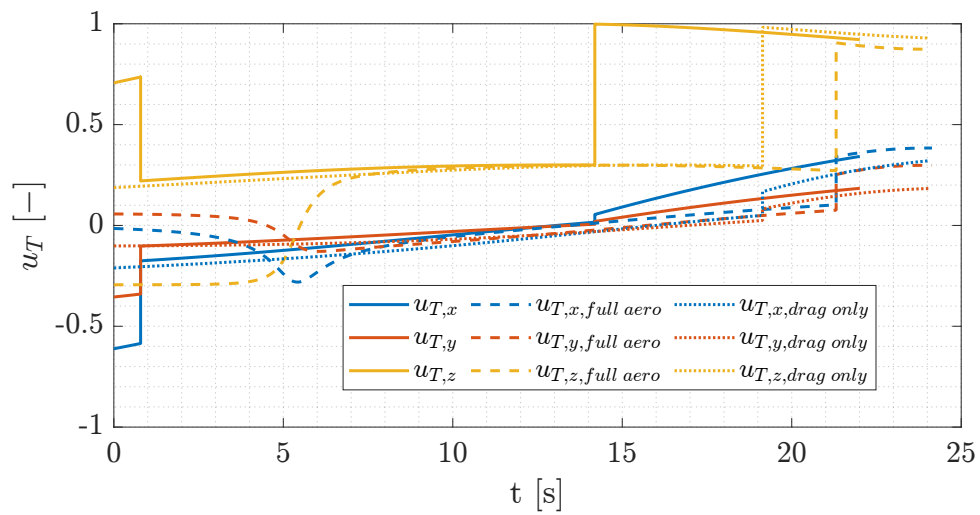


Figure 4.10: Comparison of thrust components between presented dynamic models

5 | Hybrid Pseudospectral Algorithm

This is my horse Fritz.

Dr. King Schultz
Django Unchained

As previously discussed, the highly nonlinear dynamics and the *bang–bang* optimal control structure have dictated requirements over the integrator capabilities: choosing the best one has proven to be of utmost importance. Minimizing computational time without sacrificing solution accuracy requires therefore a tailored algorithm structure, to prevent the complexities mentioned above to degrade algorithm performances.

It has been chosen to merge a convex form of the aerodynamic powered landing problem with an indirect collocation scheme. The whole strategy is represented in Fig. 5.1. Following the flowchart direction, we first approach the problem $B^{N\lambda}$: previous studies have demonstrated the accuracy and speed of methods based on lossless convexifications (LCvx) and on pseudospectral transcription [40, 41]. In this work the lossless scheme is augmented with successive convexifications (SCvx); these allow to 1) use a *free final time* formulation and 2) include *aerodynamic forces*. Multipliers are then mapped to costates using the CMT outlined in Eq. (3.20), which is polynomial with respect to multipliers and time bounds of the segments. Applying it, thus bridging the direct step with the indirect one, is of minor computational requirements. Costates are however dual to the non-physical states required by the LCvx [37]: an additional mapping is therefore applied, following indications in [94]. An indirect solver based on pseudospectral collocation is finally employed as refinement strategy since 1) it does not treat discontinuous functions, the control, but only \mathcal{C}^0 functions, the dynamics; 2) it can exploit the multipliers found at each collocation point from the direct step. Such indirect collocation is further equipped

with a tailored *switching detection scheme* and the meshing strategy allows handling non-regularities associated with C^0 states and costates. Deeper analysis of the direct and indirect step are reported respectively in Sec. 5.1 and in Sec. 5.2. In Sec. 5.3, performances of the complete algorithm are provided.

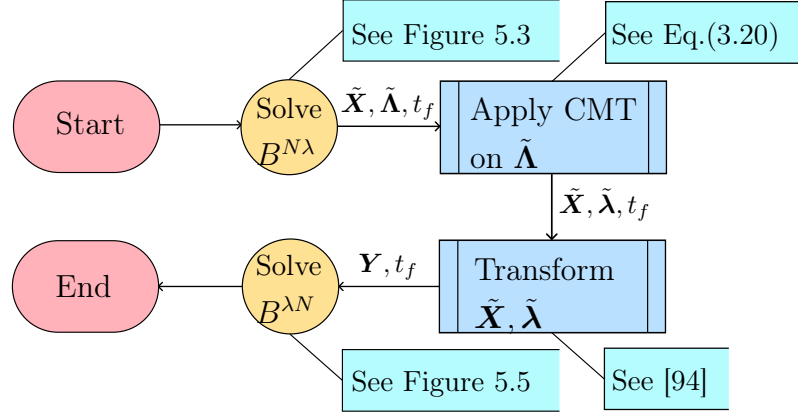


Figure 5.1: Complete hybrid algorithm structure

5.1 Direct convex collocation scheme

Let us consider the original problem formulated in Eq. (4.8). Such problem contains sources of non convexity that are handled with combined approaches: 1) a lossless convexification handles the non convexity associated to thrust magnitude constraints 2) a linearization handles the aerodynamic contributions and the final time contribution associated with the pseudospectral transcription.

Thrust magnitude lossless convexification

With respect to the original problem, we ignore at first the aerodynamic forces, and hypothesize fixed final time \bar{t}_f . The main issue arises from the limitation on thrust magnitude Eq. (4.6). In Fig. 5.2a, the acceptable domain is highlighted when projected onto the 2-D plane $u_{T,x} - u_{T,y}$, and it constitutes a non-convex set. Moreover, let us introduce the control $\Gamma \in \mathbb{R}$ such that

$$\frac{\|\mathbf{T}\|_2}{T_{\max}} = u_T \leq \Gamma \quad \text{and} \quad u_{T,\min} \leq \Gamma \leq u_{T,\max} \quad (5.1)$$

Then the non-convex feasible set in Fig. 5.2a is mapped to the convex feasible set in Fig. 5.2b, which constitutes a 3-D projection of the 4-D conic constraint in the first relation of Eq. (5.1). Demonstration that such relaxation is lossless is provided in [37]: if one solves the optimal control problem associated with this new relaxed constraint, first relation in Eq. (5.1) is satisfied tightly, therefore the initial thrust magnitude constraint is satisfied.

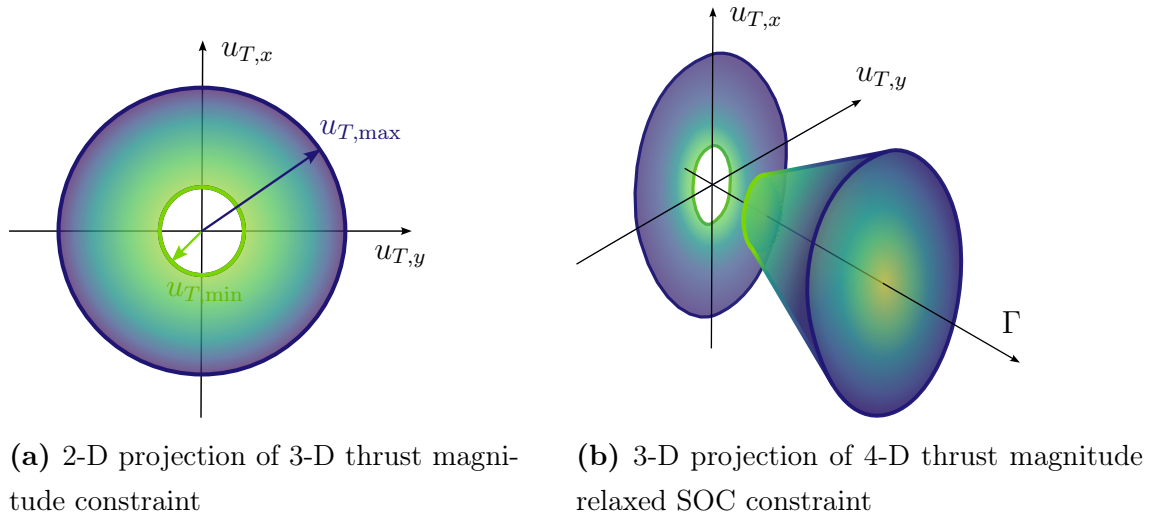


Figure 5.2: Thrust magnitude convexification

Applying the following change of variables

$$\mathbf{u}_a \doteq \frac{T_{\max}}{m} \mathbf{u}_T \quad \xi \doteq \frac{T_{\max}}{m} \Gamma \quad z \doteq \log(m) \quad (5.2)$$

dynamics is rewritten as

$$\frac{d}{dt} \begin{pmatrix} \mathbf{r} \\ \mathbf{v} \\ z \end{pmatrix} \doteq \frac{d\tilde{\mathbf{x}}}{dt} = \tilde{\mathbf{f}}_{na}(\tilde{\mathbf{x}}, \mathbf{u}_a, \xi) = \begin{bmatrix} \mathbf{v} \\ \mathbf{u}_a + \mathbf{g} \\ \xi \\ -\frac{\xi}{I_{sp}g_0} \end{bmatrix} \quad (5.3)$$

the objective function \mathcal{J} reads

$$\mathcal{J}(\xi) = \int_0^{\bar{t}_f} \frac{\xi}{I_{sp}g_0} dt \quad (5.4)$$

and constraints in Eq. (5.1) become

$$\|\mathbf{u}_a\|_2 \leq \xi \quad (5.5)$$

$$u_{T,\min} e^{-z} \leq \xi \leq u_{T,\max} e^{-z} \quad (5.6)$$

At last, the non convex constraint in Eq. (5.6) can be convexified into

$$\begin{cases} \xi \geq \xi_{\min} \doteq u_{T,\min} e^{-z_0} \left[1 - (z - z_0) + \frac{(z - z_0)^2}{2} \right] \\ \xi \leq \xi_{\max} \doteq u_{T,\max} e^{-z_0} [1 - (z - z_0)] \end{cases} \quad (5.7)$$

where $z_0 \doteq \log(m_0 - u_{T,\max} t / I_{sp} g_0)$.

The problem defined by the objective function in Eq. (5.4), satisfying dynamics in Eq. (5.3), respectful of boundary conditions, and respectful of control constraints in Eq. (5.5) and in Eq. (5.7) constitutes a convex optimization problem when transcribed with a pseudospectral method.

Aerodynamic forces successive convexifications

The algorithm treats the aerodynamic forces with successive linear convexifications, augmenting the control components presented in the previous paragraph with aerodynamic controls.

Adding aerodynamic contributions, the dynamics right-hand side reads

$$\tilde{\mathbf{f}}(\tilde{\mathbf{x}}, \mathbf{u}_a, \xi, \alpha_{\text{eff}}, \sigma) = \begin{bmatrix} \mathbf{v} \\ \mathbf{u}_a + \mathbf{g} + \mathbf{a}_{aero} \\ -\frac{\xi}{I_{sp} g_0} \end{bmatrix} \quad \text{where} \quad \mathbf{a}_{aero} = \frac{\mathbf{D} + \mathbf{L}}{e^z} \quad (5.8)$$

With respect to the indirect formulation, however, the lift expression is explicitly formulated with respect to the bank angle; constraints on the aerodynamic controls are reduced to bounds over the angle of attack, and, eventually, on the bank angle. Considering the lift force, only its direction \mathbf{i}_L is dependant on the bank angle σ : the procedure mapping the velocity vector \mathbf{v} and σ to \mathbf{i}_L is reported in Appendix C. \mathbf{i}_L is a sinusoidal vectorial function of σ . Therefore

$$\mathbf{L}(r_z, \mathbf{v}, \alpha_{\text{eff}}, \sigma) = \frac{1}{2} \rho(r_z) S_{ref} C_{L/\alpha} \alpha_{\text{eff}} v^2 \mathbf{i}_L(\mathbf{v}, \sigma) \quad (5.9)$$

which shall be coupled with box constraints over σ to guarantee correct convergence of the algorithm

$$-\pi < \sigma < \pi \quad (5.10)$$

In addition, lower bound on α_{eff} shall be added to guarantee $\alpha_{\text{eff}} \geq 0^\circ$.

Remark 5.1: Employing \mathbf{u}_b Sec. 4.1 as control variable is theoretically possible. However, \mathbf{u}_b identifies a direction, therefore it shall be $\|\mathbf{u}_b\|_2 = 1$; this is a strongly non convex constraint. No lossless relaxation has been elaborated, therefore the formulation with α_{eff} and σ is preferred.

Let us gather the controls in the vector \mathbf{u} , defined as

$$\mathbf{u} \doteq \left[\mathbf{u}_a^T, \xi, \alpha_{\text{eff}}, \sigma \right]^T \quad (5.11)$$

The functions $\tilde{\mathbf{x}}(t)^{(k)}$ and $\mathbf{u}(t)^{(k)}$ define the k^{th} term of a series of reference solutions of the aerodynamic powered descent problem with fixed final time \bar{t}_f . Each of these solutions satisfies the dynamics in Eq. (5.8) linearized about the previous reference solution of the series. Therefore $\tilde{\mathbf{x}}^{(k)}$ and $\mathbf{u}^{(k)}$ satisfy

$$\dot{\tilde{\mathbf{x}}}^{(k)} = \tilde{\mathbf{f}}_l^{(k)} [\tilde{\mathbf{x}}^{(k)}, \mathbf{u}^{(k)}] = \mathbf{A}^{(k)} \tilde{\mathbf{x}}^{(k)} + \mathbf{B}^{(k)} \mathbf{u}^{(k)} + \mathbf{e}^{(k)} \quad (5.12)$$

where

$$\mathbf{A}^{(k)} \doteq \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbb{I}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \dots & \dots & \dots \\ \mathbf{J}_{\tilde{\mathbf{x}}} \mathbf{a}_{aero}^{(k-1)} & & \\ \dots & & \\ \mathbf{0}_{1 \times 7} & & \end{bmatrix}, \quad \mathbf{B}^{(k)} \doteq \begin{bmatrix} & & \mathbf{0}_{3 \times 6} & & \\ \dots & \dots & \dots & \dots & \dots \\ \mathbb{I}_{3 \times 3} & & \mathbf{0}_{3 \times 1} & & \mathbf{J}_{[\alpha_{\text{eff}}, \sigma]} \mathbf{a}_{aero}^{(k-1)} \\ \dots & & \dots & \dots & \dots \\ \mathbf{0}_{1 \times 3} & & -\frac{1}{I_{\text{sp}} g_0} & & \mathbf{0}_{1 \times 2} \\ \dots & & \dots & & \dots \end{bmatrix},$$

$$\mathbf{e}^{(k)} \doteq \tilde{\mathbf{f}} [\tilde{\mathbf{x}}^{(k)}, \mathbf{u}^{(k)}] - \mathbf{A}^{(k)} \tilde{\mathbf{x}}^{(k-1)} - \mathbf{B}^{(k)} \mathbf{u}^{(k-1)} \quad (5.13)$$

Aerodynamics are therefore handled imposing at each k^{th} step the linearized dynamics in Eq. (5.12). Reference to *successive convexifications* arises as the OCP is reformulated at each step linearizing dynamics with respect to a different reference solution.

Final time successive convexifications

Final time is handled in a similar manner as done for the aerodynamic forces. However, its contribution is due to the transcription process. The direct collocation scheme employs an *hp* pseudospectral discretization, building on [41]. Let us use the same notation employed in Sec. 3.2.1; in addition to nodal states and controls, $\tilde{\mathbf{X}}^s$ gathers the states over s^{th} segment. At k^{th} iteration, left-hand side of Eq. (5.12) is transcribed using an LGR pseudospectral scheme and linearized about previous iteration; therefore

$$\left\{ \begin{array}{l} \frac{2h}{t_f^{(k-1)}} \mathbf{D}_i \tilde{\mathbf{X}}^{(s,k)} = \mathbf{A}_i^{(s,k)} \tilde{\mathbf{X}}_i^{(s,k)} + \\ \quad + \mathbf{B}_i^{(s,k)} \mathbf{U}_i^{(s,k)} + \mathbf{C}_i^{(s,k)} t_f^{(k)} + \tilde{\mathbf{e}}_i^{(s,k)} \\ \mathbf{C}_i^{(s,k)} \doteq \frac{2h}{[t_f^{(k-1)}]^2} \mathbf{D}_i \tilde{\mathbf{X}}^{(s,k-1)} \\ \tilde{\mathbf{e}}_i^{(s,k)} \doteq \mathbf{e}_i^{(s,k)} - \frac{2h}{t_f^{(k-1)}} \mathbf{D}_i \tilde{\mathbf{X}}^{(s,k-1)} \end{array} \right. \quad \begin{array}{l} s = 1, \dots, h \\ i = 0, \dots, n \end{array} \quad (5.14)$$

The system of equations in Eq. (5.14) is linear with respect to unknowns, therefore it can be used as constraint within a convex formulation. The objective function is integrated with a pseudospectral quadrature scheme and its contribution with respect to final time linearized as well; in such form, it can be used as objective of a convex optimization problem. Therefore

$$\bar{\mathcal{J}} [\mathbf{U}^{(k)}, t_f^{(k)}] = \frac{1}{2h} \left[\sum_{s=0}^h \sum_{i=0}^n w_i \xi_i^{(s,k-1)} \right] t_f^{(k)} + \frac{t_f^{(k-1)}}{2h} \sum_{s=0}^h \sum_{i=0}^n w_i \xi_i^{(s,k)} \quad (5.15)$$

Let $\Delta(\bullet)^{(k)} \doteq (\bullet)^{(k)} - (\bullet)^{(k-1)}$. Two fixed trust region are included in problem formulation: the first trust region ζ_{t_f} relates to t_f at each iteration via the constraint $\|\Delta t_f^{(k)}\|_2 \leq \zeta_{t_f}$; similarly, the second trust region $\zeta_{\alpha_{\text{eff}}}$ is imposed through the constraint $\|\Delta \alpha_{\text{eff}}^{(s,k)}\|_2 \leq \zeta_{\alpha_{\text{eff}}}$, with $i = 0, \dots, n$ and $s = 1, \dots, h$. Constraints are nonlinear, therefore linearization carries approximation inaccuracies: the solution of the linearized problem may differ sensibly from the solution of the original one; in our case, this leads to oscillations about the optimal solution, preventing the algorithm from reaching convergence. The need to limit variations of t_f and nodal α_{eff} follows. Equation (5.14) and Eq. (5.15) can be integrated in problem formulation along with the trust region constraints. This completes the convex pseudospectral collocation scheme.

5.1.1 Direct step overview

The convex optimization problem solved at each iteration of the direct step reads

$$\min_{\mathbf{U}^{(k)}, t_f^{(k)}} \bar{\mathcal{J}} [\mathbf{U}^{(k)}, t_f^{(k)}] \quad \text{s.t.} \quad \left\{ \begin{array}{l} \text{Dynamics in Eq. (5.14)} \\ \tilde{\mathbf{X}}_0^{(1,k)} = \tilde{\mathbf{x}}_0 \\ \tilde{\mathbf{X}}_n^{(h,k)} = \tilde{\mathbf{x}}_f \\ \tilde{\mathbf{X}}_n^{(s-1,k)} = \tilde{\mathbf{X}}_0^{(s,k)} \\ \|\mathbf{u}_{a_i}^{(s,k)}\|_2 \leq \xi_i^{(s,k)} \\ \xi_i^{(s,k)} \geq \xi_{\min_i}^{(s,k)} \\ \xi_i^{(s,k)} \leq \xi_{\max_i}^{(s,k)} \\ 0 \leq \alpha_{\text{eff}_i}^{(s,k)} \leq \alpha_{\max} \\ -\pi \leq \sigma_i^{(s,k)} \leq \pi \\ \|\Delta \alpha_{\text{eff}_i}^{(s,k)}\|_2 \leq \zeta_{\alpha_{\text{eff}}} \\ \|\Delta t_f^{(k)}\|_2 \leq \zeta_{t_f} \end{array} \right. \quad (5.16)$$

The employed strategy is represented in Fig. 5.3: iterations are stopped if $|\Delta t_f^{(k)}| < \text{tol}$, and multipliers are extracted at the last step. The problem is solved with sequentially more complex dynamics: guess time for the LCvx formulation is retrieved from the 1-D problem analytic formulation from [95]; then time is left varying. \mathbf{D} and \mathbf{L} are only added afterwards, and separately. Trust regions over α_{eff} are set to $\alpha_{\max}/10$ and $\alpha_{\max}/50$ for the steps respectively with drag only with both lift and drag. An LGR scheme is employed for transcription, thus collocating dynamics at the first time instant; the open-source convex solver ECOS [58] is used as back-end solver for the problem Eq. (5.16), setting its tolerances `abstol`, `feastol`, `reltol` = 10^{-10} . Strict tolerances allow to minimize the error associated with sub-problem solution.

5.2 Indirect collocation scheme

Within the indirect step, problem in Eq. (4.16) is transcribed and dynamics is collocated with a pseudospectral scheme; the resulting zero-finding problem is solved sequentially with the Newton-Raphson method. Employment of a pseudospectral

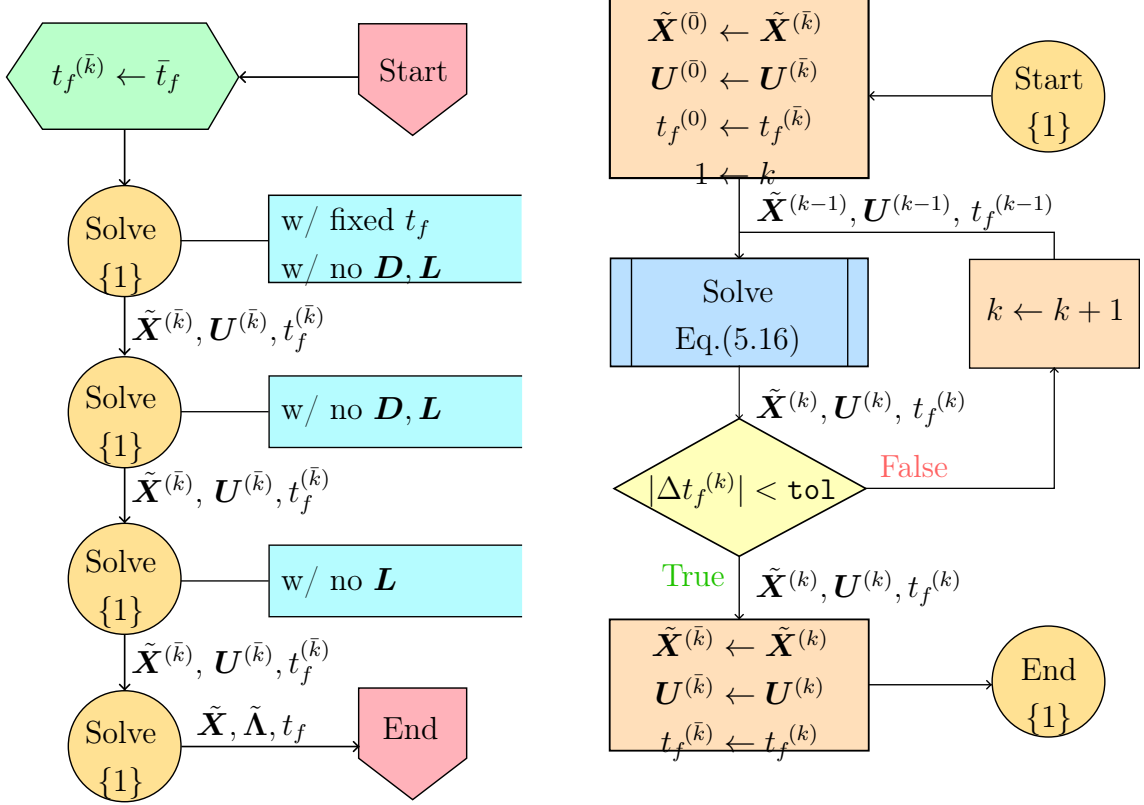


Figure 5.3: Direct step *zoom-in*, from Fig. 5.1

On the left: Solver structure

On the right: Solver loop block - {1}

scheme allows high solution accuracy with few nodes employment. Dynamics is linearized at each iteration step: the transcribed problem corresponds therefore to a determined linear system; this ensures swift sequential solution of the BVP. The same procedure is applied to $\tilde{\mathcal{H}}_f$, thus constraining the degree of freedom provided by t_f .

Results outlined in Chap. 4 show $m(t)$ and $\lambda_m(t)$ are not \mathcal{C}^1 ; this is due to the structure of the optimal thrust magnitude profile. Collocation shall therefore deal with such non-regularities with a tailored approach. To do this, domain is meshed exploiting the guess from the direct method, and structure of such mesh is hold during the BVP solution procedure. Moreover, the resulting linear system is rescaled to minimize *ill-conditioning* of the dual system [24].

Remark 5.2: A further higher-order non regularity arises from the saturation of α_{eff} ; since $\alpha_{\text{eff}} \in \mathcal{C}^0$, this would ensure $m(t), \lambda_m(t) \in \mathcal{C}^1$. Inaccuracies associated

with thrust switch angular points are however dominant, and therefore treated.

Meshing strategy for thrust non-regularities handling

Local non-regularities can be accurately approached if local collocation is adopted: knots shall be placed at switches, exploiting the piecewise structure of the integrated dynamics. With respect to standard hp collocation methods [96], solution refinement is guaranteed by an r scheme: such method modifies location of knots rather than the interpolating polynomials order or the number of segments [97].

As shown in Chap. 4, a single switching instant t_{sw} shall be handled; it is therefore added to the unknowns vector. The switching function S_f is exploited as additional necessary constraint: from variational formulation, indeed, $S_f = 0$ whenever thrust switches; therefore, the related constraint results

$$S_f [\mathbf{y} (t_{\text{sw}})] = 0 \quad (5.17)$$

The previous nonlinear constraint is linearized sequentially during the BVP solving procedure; with the inclusion of such constraint, the final problem is still a linear system. Imposition of final hamiltonian and of Eq. (5.17) reads

$$\begin{cases} \mathbb{A}_{\tilde{\mathcal{H}}_f}^{(k)} \Delta \mathbf{Y}_n^{(2,k)} = \mathbb{E}_{\tilde{\mathcal{H}}_f}^{(k)} \\ \mathbb{A}_{\tilde{\mathcal{H}}_f}^{(k)} \doteq \nabla_{\mathbf{y}}^T \tilde{\mathcal{H}}_f^{(k-1)} \\ \mathbb{E}_{\tilde{\mathcal{H}}_f}^{(k)} \doteq -\tilde{\mathcal{H}}_f^{(k-1)} \end{cases} \quad (5.18)$$

$$\begin{cases} \mathbb{A}_{S_f}^{(k)} \Delta \mathbf{Y}_n^{(1,k)} = \mathbb{E}_{S_f}^{(k)} \\ \mathbb{A}_{S_f}^{(k)} \doteq \nabla_{\mathbf{y}}^T S_f^{(k-1)} \\ \mathbb{E}_{S_f}^{(k)} \doteq -S_f^{(k-1)} \end{cases}$$

Mesh initialization is based on [45]: switching function is approximated with the piecewise linear function $S_{f,\text{pw}}$ that interpolates the values mapped from the guess; then $t_{\text{sw}}^{(0)} \doteq S_{f,\text{pw}}^{-1}(0)$. Remeshing procedure is reported in Fig. 5.4; number of segments and collocation points have been chosen for visual clarity only. The new mesh is constrained to number of segments equal to the number of switches augmented by 1; for our problem 2 segments are therefore sufficient. This allows each segment to feature a constant thrust magnitude profile; to further validate effectiveness of such technique, the PMP is still applied at each node, without externally constraining thrust magnitude. With the r collocation, dynamics are imposed using different scal-

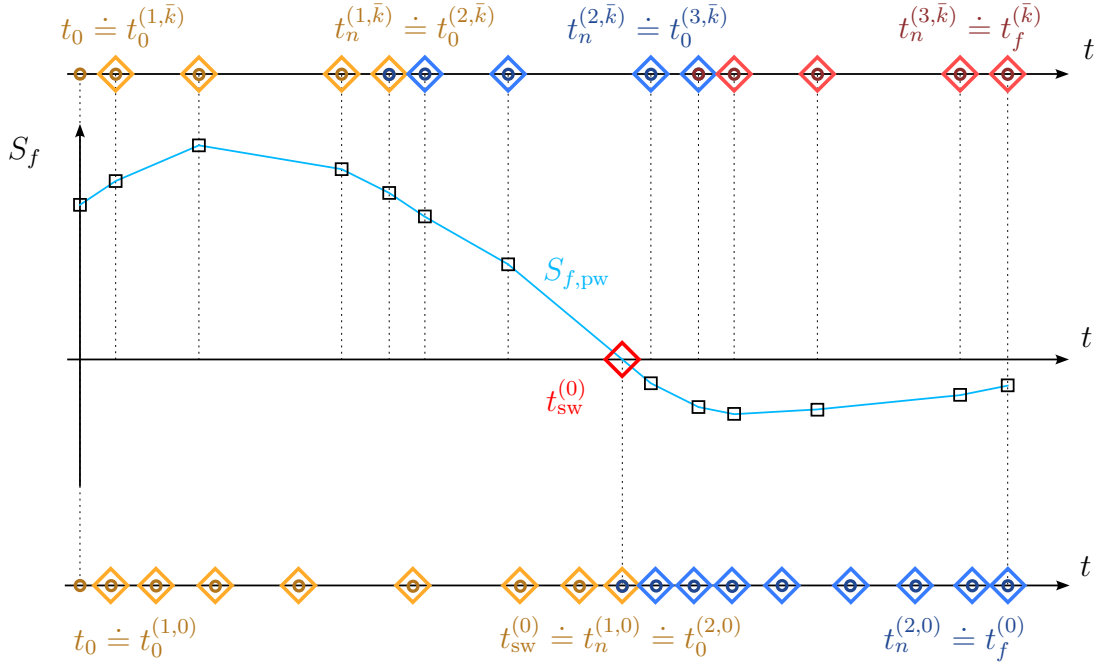


Figure 5.4: Remeshing procedure representation – \bar{k} is last iteration of the direct step; relative mesh on top. Remeshed domain for the first iteration of indirect step on bottom

ing factors for the two segments, given their different lengths. Therefore, dynamics at the generic k^{th} reads

$$\begin{cases}
 \mathbb{D}_i^{(s,k)} \Delta \mathbf{Y}^{(s,k)} = \mathbb{A}_i^{(s,k)} \Delta \mathbf{Y}^{(s,k)} + \\
 \quad \quad \quad + \mathbb{C}_i^{(s,k)} \left[\Delta t_f^{(s,k)} - \Delta t_0^{(s,k)} \right] + \mathbb{E}_i^{(s,k)} \\
 \mathbb{D}_i^{(s,k)} \doteq \frac{2}{t_f^{(s,k-1)} - t_0^{(s,k-1)}} \mathbf{D}_i \\
 \mathbb{A}_i^{(s,k)} \doteq \mathbf{J}_y \mathbf{F}_i^{(s,k-1)} \\
 \mathbb{C}_i^{(s,k)} \doteq \frac{2}{\left[t_f^{(s,k-1)} - t_0^{(s,k-1)} \right]^2} \mathbf{D}_i \mathbf{Y}^{(s,k-1)} \\
 \mathbb{E}_i^{(s,k)} \doteq \mathbf{F}_i^{(s,k-1)} - \mathbb{D}_i^{(s,k)} \mathbf{Y}^{(s,k-1)}
 \end{cases} \quad \begin{array}{l} s = 1, 2 \\ i = 0, \dots, n \end{array} \quad (5.19)$$

where $t_0^{(1,k)} = 0 \forall k$ to satisfy boundary conditions; moreover $t_0^{(2,k)} = t_f^{(1,k)} \doteq t_{\text{sw}}^{(k)}$ and $t_f^{(2,k)} \doteq t_f^{(k)}$.

5.2.1 Indirect step overview

Let the augmented unknowns vector $\tilde{\mathbf{Y}} \doteq [\mathbf{Y}^T, t_{sw}, t_f]^T$ include the final time and switching time. An fLGR collocation scheme is employed, as dynamics imposition at final node increases accuracy in the estimation of $\tilde{\mathcal{H}}$. Problem is further scaled according to the Jacobian Rows Normalization (JRN) presented in [24], thus decreasing ill-conditioning. The generic solved problem at k^{th} iteration results

$$\tilde{\mathbf{A}}_{\text{lin}}^{(k)} \Delta \tilde{\mathbf{Y}}^{(k)} = \tilde{\mathbf{b}}_{\text{lin}}^{(k)} \quad \text{where} \quad \begin{cases} \tilde{\mathbf{A}}_{\text{lin}}^{(k)} \doteq \mathbf{K}_f \mathbf{A}_{\text{lin}}^{(k)} \\ \tilde{\mathbf{b}}_{\text{lin}}^{(k)} \doteq \mathbf{K}_f \mathbf{b}_{\text{lin}}^{(k)} \\ \mathbf{K}_{f,i,j} \doteq \begin{cases} \frac{1}{\|\mathbf{A}_{\text{lin},i}^{(k)}\|_2} & \text{if } i = j \\ 0 & \text{else} \end{cases} \end{cases} \quad (5.20)$$

and

$$\mathbf{A}_{\text{lin}}^{(k)} = \begin{bmatrix} \mathbf{A}_{\text{dyn},\mathbf{y}}^{(k)} & \vdots & \mathbf{A}_{\text{dyn},t}^{(k)} \\ \dots & \dots & \dots \\ \mathbf{A}_{\text{knot}} & \vdots & \\ \dots & \dots & \\ \mathbf{A}_{BCs} & \vdots & \\ \dots & \dots & \\ \mathbf{A}_{\tilde{\mathcal{H}}_f}^{(k)} & \vdots & \mathbf{0}_{(4n_s+2) \times 2} \\ \dots & \dots & \\ \mathbf{A}_{S_f}^{(k)} & \vdots & \end{bmatrix} \quad (5.21)$$

$$\mathbf{A}_{\text{dyn},\mathbf{y}}^{(k)} = \begin{bmatrix} \mathbb{D}^{(1,k)} - \text{blk}_{i=1,\dots,n} [\mathbb{A}_i^{(1,k)}] & \vdots & \mathbf{0}_{[2n_s n] \times [2n_s(n+1)]} \\ \dots & \dots & \dots \\ \mathbf{0}_{[2n_s n] \times [2n_s(n+1)]} & \vdots & \mathbb{D}^{(2,k)} - \text{blk}_{i=1,\dots,n} [\mathbb{A}_i^{(2,k)}] \end{bmatrix} \quad (5.22)$$

$$\mathbf{A}_{\tilde{\mathcal{H}}_f}^{(k)} = \begin{bmatrix} \mathbf{0}_{1 \times [2n_s(n+1)]} & \vdots & \mathbf{0}_{1 \times 2n_s n} & \mathbb{A}_{\tilde{\mathcal{H}}_f}^{(k)} \end{bmatrix} \quad (5.23)$$

$$\mathbf{A}_{S_f}^{(k)} = \begin{bmatrix} \mathbf{0}_{1 \times 2n_s n} & \mathbb{A}_{S_f}^{(k)} & \vdots & \mathbf{0}_{1 \times [2n_s(n+1)]} \end{bmatrix} \quad (5.24)$$

$$\mathbf{A}_{\text{dyn},t}^{(k)} = \begin{bmatrix} -\mathbb{C}^{(1,k)} & \vdots & \mathbf{0}_{[2n_s n] \times 1} \\ \dots & \dots & \dots \\ \mathbb{C}^{(2,k)} & \vdots & -\mathbb{C}^{(2,k)} \end{bmatrix} \quad (5.25)$$

$$\mathbf{b}_{\text{lin}}^{(k)} = \begin{bmatrix} \mathbb{E}^{(1,k)} \\ \mathbb{E}^{(2,k)} \\ \mathbf{0}_{4n_s \times 2} \\ \mathbb{E}_{\mathcal{H}_f}^{(k)} \\ \mathbb{E}_{S_f}^{(k)} \end{bmatrix} \quad (5.26)$$

Remaining constant matrices \mathbf{A}_{knot} and \mathbf{A}_{BCs} are the constant Jacobians respectively of the knotting constraints and of state boundary conditions with respect to vector \mathbf{Y} . Moreover, the operator $\text{blk}_{i=1,\dots,n}(\bullet)$ assembles a rectangular $n \times (n+1)$ block matrix placing at each i^{th} block row and $(i+1)^{\text{th}}$ block column the argument (\bullet) evaluated at the $(i+1)^{\text{th}}$ node.

$\tilde{\mathbf{Y}}$ is updated at each iteration step using the fixed trust region $\zeta_{\tilde{\mathbf{Y}}}$, thus providing

$$\tilde{\mathbf{Y}}^{(k)} = \tilde{\mathbf{Y}}^{(k-1)} + \zeta_{\tilde{\mathbf{Y}}} \Delta \tilde{\mathbf{Y}}^{(k)} \quad (5.27)$$

and the linear system in Eq. (5.20) is solved using MATLAB's built-in linear system solver `mldivide`, employing an LU factorization. BVP is considered solved if

$$\max \left[\text{abs} \left(\Delta \tilde{\mathbf{Y}}^{(k)} \right) \right] < \text{tol} \quad (5.28)$$

where $\text{abs}(\bullet)$ provides the component-wise absolute value of (\bullet) and $\max(\bullet)$ selects maximum component of (\bullet) . The complete indirect step is reported in Fig. 5.5.

5.3 Final Results

Accuracy and computational times of the direct step need to be traded off. The first refers to the estimation of the guess: to evaluate guess accuracy, compass estimated using 10 segments and 20 collocation points is taken as reference solution; segments and collocation points are varied, and the maximum relative error of the new λ_m history with respect to the reference λ_m is computed. Results are reported in Tab. 5.1. Maximum iterations are fixed to 50, for each step of the problem: dashed lines represent settings that have not reached convergence. Computational times have been evaluated averaging 10 runs for each setting.

Among the shown configurations, the one with 5 segments and 5 collocation nodes per segment is chosen: it grants accurate costates with scarce computational times. Converged values of the objective function and of final time are reported, as function

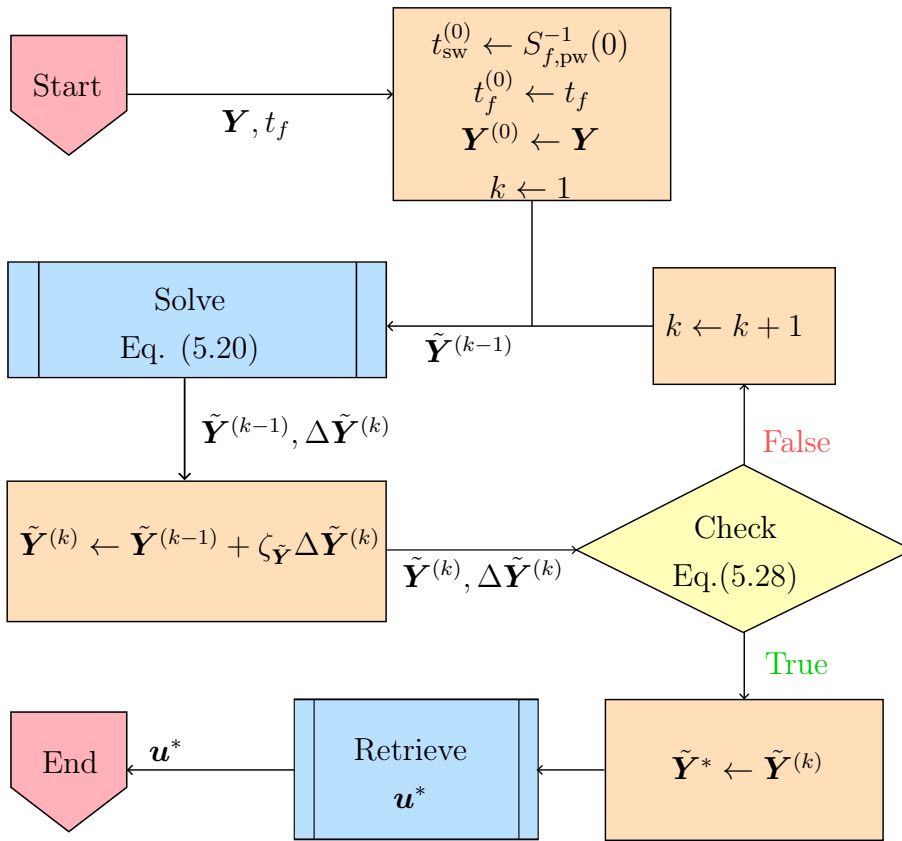


Figure 5.5: Indirect step *zoom-in*, from Fig. 5.1

Table 5.1: Normalized error over estimated λ_m and computational times[†] for different combinations of segments and collocation points per segment

		Segments			
		1	2	5	10
Collocation points p. s.	5	-	0.391	1.060	3.050
	10	0.406	1.347	-	11.414
	20	1.264	3.947	13.951	43.491
	5	-	17.08	5.79	5.72
	10	32.20	9.54	-	2.12
	20	11.2	3.60	5.72	/

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

of the iterations, in Fig. 5.6. Trust region over final time value is not reinforced before introduction of aerodynamic forces; this allows to improve speed performances.

Remark 5.3: With respect to results reported in Tab. 4.2, the final mass reached in Fig. 5.6 for the drag-only case is 1.5% higher. This behaviour is caused by the fact simulations shown in Fig. 5.6 have upper bound α_{\max} of 15° ; in the homotopic formulation, instead, in the drag-only analysis it is reinforced $\alpha_{\text{eff}} = 0$ as such value is consistent with the hypothesis of lift neglect.

Optimal trajectories for the original problem and for the only-drag and full aerodynamic formulations are reported in Fig. 5.7: addition of lift, namely iterations 16, 20 and 24, corresponds to curved optimal trajectories, as shown by the ground tracks represented with dashed lines.

Furthermore, convergence of most relevant controls is reported in Fig. 5.8 and in Fig. 5.9, respectively the profiles of u_T and of α_{eff} . Dynamics at final nodes is not collocated, and relative integration weights are null. The optimizer is therefore not able to optimize such values, the control is *unregulated*, thus discarded in the analysis and not represented. Moreover, uncertainty over thrust switching time is directly related

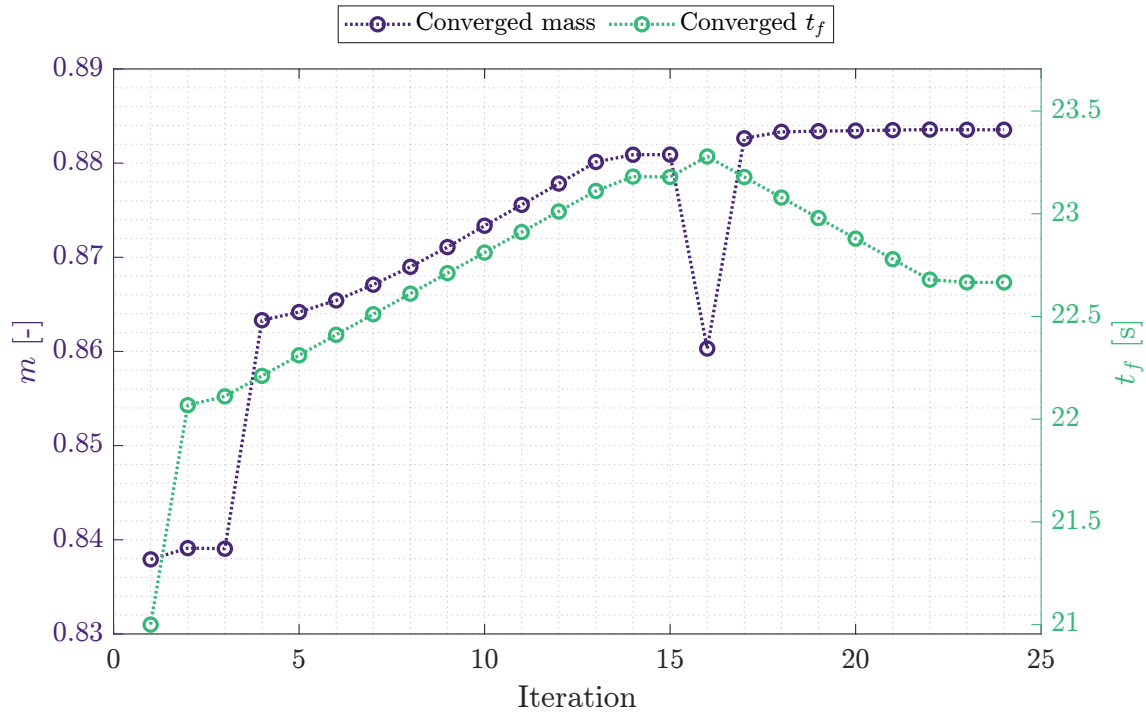


Figure 5.6: Final mass and final time iterations of direct step, with 5 segments and 5 collocation points per segment. *Iteration 1:* LCvx formulation; *Iterations 2-3:* Free t_f added; *Iterations 4-15:* \mathbf{D} added; *Iterations 16-24:* \mathbf{L} added

to the mesh; with the selected mesh such uncertainty amounts approximately to 0.5 s for the final converged control profile. At last, estimation of costates is reported in Fig. 5.10; λ^* , the optimal costates, are retrieved from the solution reported in Sec. 4.4. 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; $\partial\tilde{\mathcal{H}}/\partial r_x = \partial\tilde{\mathcal{H}}/\partial r_y = 0$, thus λ_{r_x} and λ_{r_y} can be corrected in a single iteration within the indirect algorithm.

Let the guess from the direct method be fixed. The computational time of the indirect step is dominated by the magnitude of the trust regions and by the number of adopted collocation points. Accuracy is instead dominated by the collocation points only. However, employment of higher number of collocation points allows to increase the size of the trust regions before incurring in numerical instability: linear approximation accuracy is favored by high number of collocation points. Results are therefore reported in Tab. 5.2, where e_{t_f} and $e_{t_{sw}}$ denote the error in the estimated t_f and t_{sw} , evaluated with respect to the benchmark described in the double homotopy.

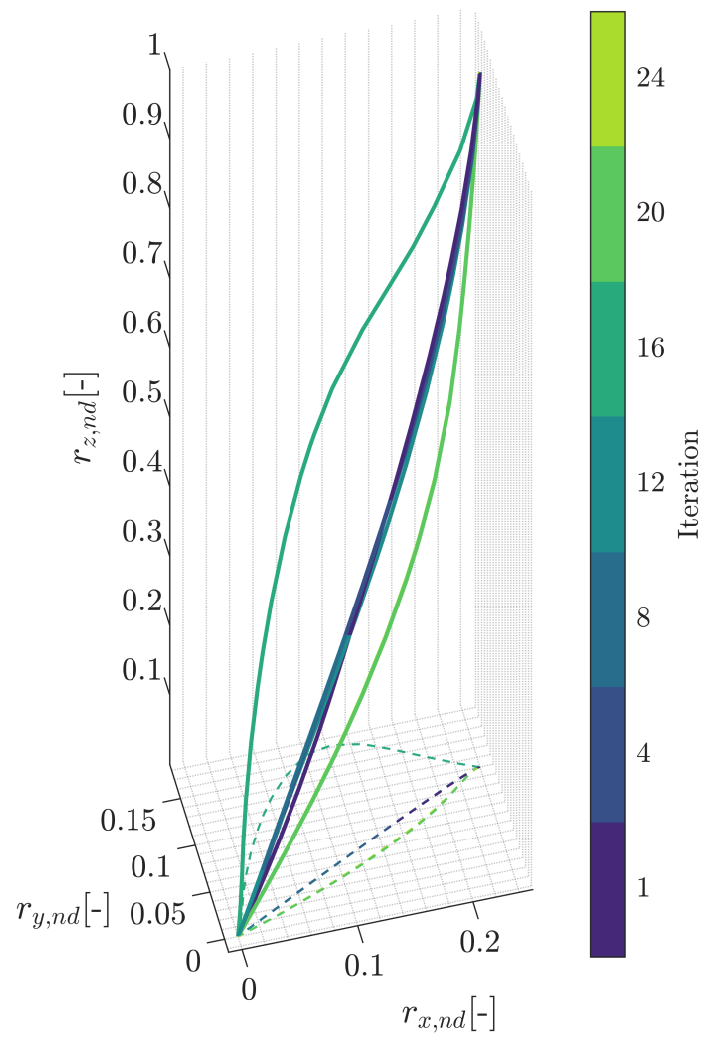


Figure 5.7: Optimal trajectory for progressive direct iterations

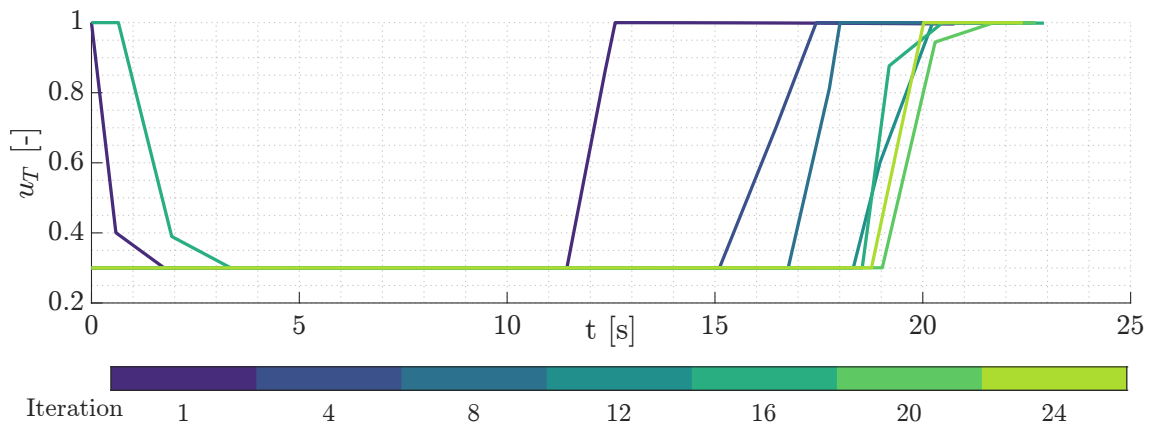


Figure 5.8: u_T for progressive direct iterations

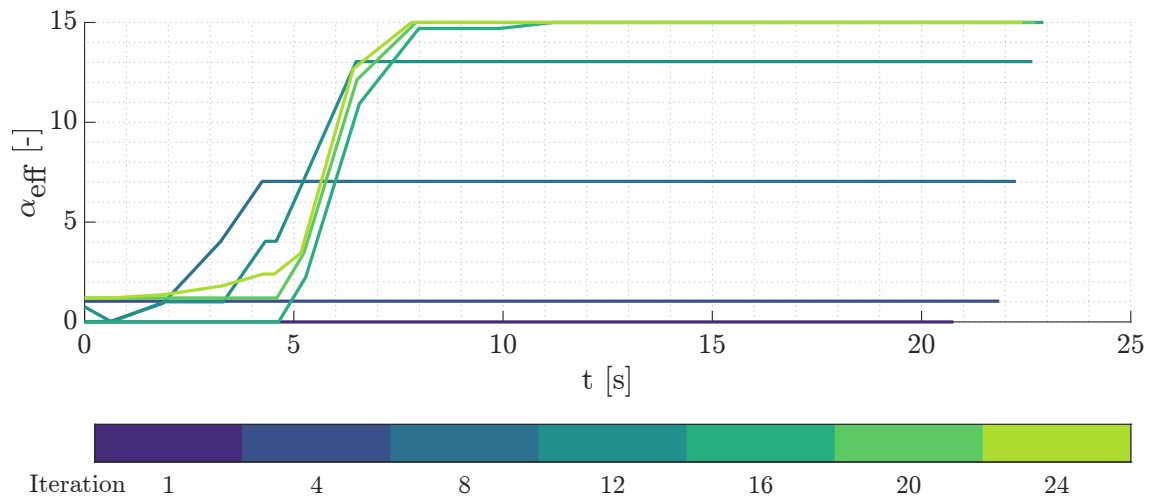


Figure 5.9: α_{eff} for progressive direct iterations

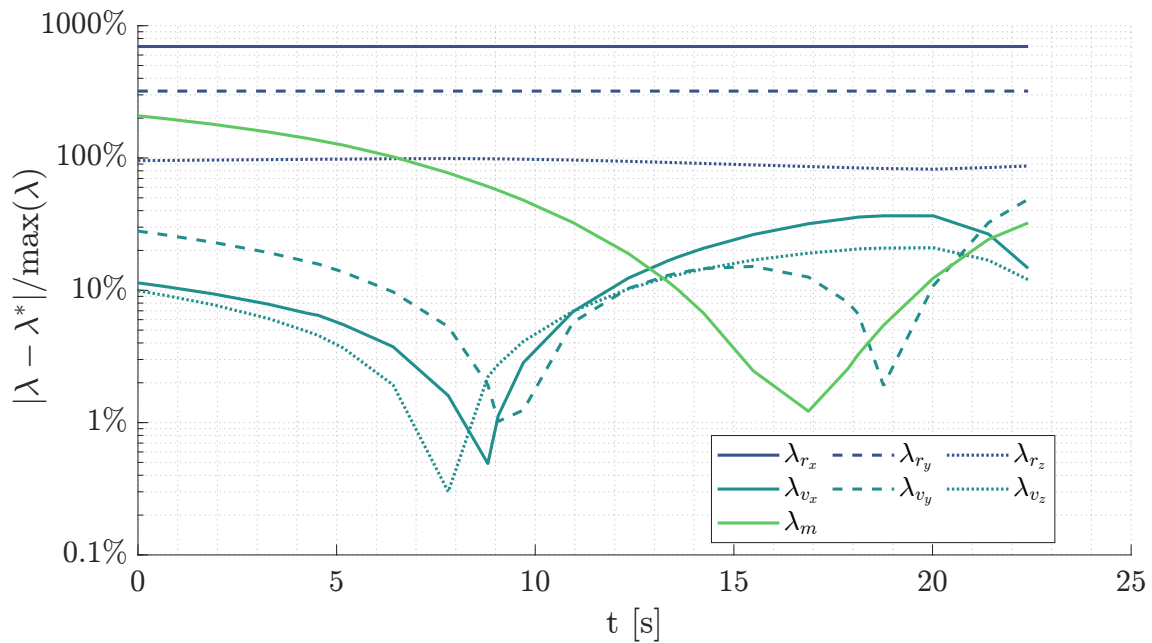


Figure 5.10: Summary of relative errors in costates estimated with the direct convex method

$\zeta_{\tilde{\mathbf{Y}},\max}$ denotes the maximum trust region value allowing the given setting to reach convergence. Tolerance parameter `tol` is set to 10^{-5} .

For the same sets of collocation points, moreover, an accuracy analysis over the

Table 5.2: Indirect step performances for different collocation points per segment

	Collocation points per segment			
	10	20	40	80
CPU time [†] [s]	1.2008	2.2047	5.1953	15.3827
e_{t_f} [s]	0.056	0.120	0.037	0.005
$e_{t_{sw}}$ [s]	0.077	0.109	0.038	0.003
$\zeta_{\tilde{\mathbf{Y}},\max}$	0.6	0.8	0.85	0.85
m_f [kg]	882.49	882.60	882.55	882.52

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

final condition satisfaction is carried out: controls are retrieved using the PMP and closed loop rocket dynamics is simulated. Specifically, all controls are interpolated using cubic approximations, with the exception of the thrust magnitude; following suggestions in [42, 34], thrust magnitude is treated employing a zero-order-hold which mirrors the *bang-bang* structure of the control profile. Results are reported in Fig. 5.11: on the left the absolute error over final touchdown position e_{r_f} is reported, computed at the final instant of simulation; on the right the corresponding touchdown velocity; this is evaluated during the simulation, within the integrator when the condition $r_z = 0$ m is detected. Results show maximum error in accuracy slightly higher than 1 m over final landing site position; such lack of convergence to 0 can be explained considering the corner point in α_{eff} profile is not grasped by the collocation scheme and by the applied cubic interpolation. 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 e_{r_f} which determines non null touchdown velocity, and its error is in the order of 0.2% of the initial height.

Finally, it is worth comparing the discussed strategy with the single shooting-based homotopic scheme: computational times are reported in Tab. 5.3. Considering the homotopic approach, 11 subproblems are solved to shift from the energy optimal

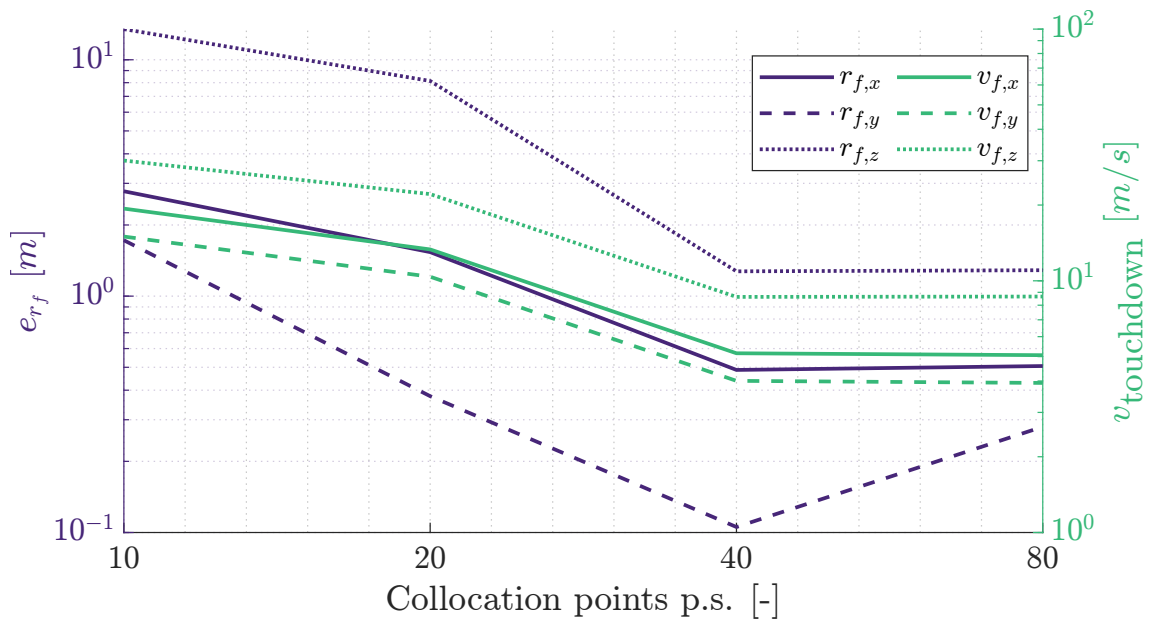


Figure 5.11: Overview of touchdown position accuracy and relative touchdown velocities

solution to the acceleration optimal one, and 11 iterations are required to add the aerodynamic contributions.

Remark 5.4: To face the homotopic continuation problem smartly, states and costates should be augmented with their State Transition Matrix (STM) to be used in the correction step, thus increasing convergence robustness [98]; this, however, would require propagating the additional 196 STM components. External switching detection techniques have been moreover tested and verified [98]; they pay the price of solving a nonlinear zero-finding problem at each switching detection, yet allow integrator to integrate the *bang-bang* dynamics correctly. The presented approach exploits MATLAB functions, which are not tailored to handling propagation and correction of discontinuous dynamics.

Despite the homotopic approach computational times being penalized by the employed COTS functions, Tab. 5.3 highlights a great gap between the single shooting and the hybrid strategy, thus demonstrating superiority of the presented technique. Computational times relative to hybrid methods are reevaluated in a full hybrid method loop, with 5 collocation points and 5 segments for the direct step and 40 collocation points per segment for the indirect step.

At last, in Fig. 5.12 the solution extracted from the direct method is presented,

Table 5.3: Comparison of computational times between fully indirect homotopic approach and hybrid technique

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

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

along with the aerodynamic forces and with a visual representation of the thrust magnitude. Aerodynamic forces are projected on plane $x - y$, and on plane $y - z$; it is evident lift allows for trajectory deflection; thus, as stated multiple times in the present work, lift is determinant to limit thruster use towards braking purposes.

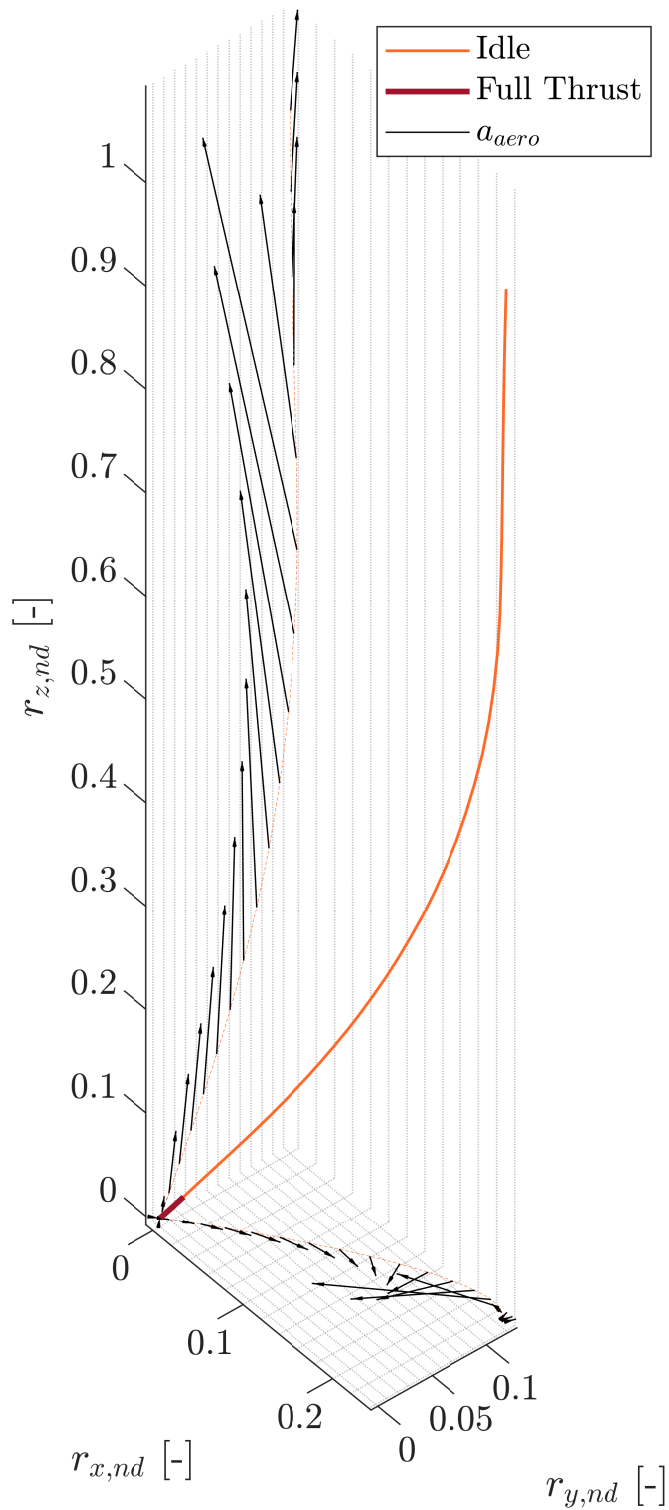


Figure 5.12: Optimal landing with aerodynamic forces and thrust magnitude – Trajectory is projected on $x - y$ and $y - z$ planes along with aerodynamic forces

6 | Conclusions and Future Developments

After all... it's just a trick.
Yes, it's just a trick.

Jep Gambardella
Great Beauty

An innovative strategy combining the workhorses of direct and indirect methods has been presented and discussed; as well, extensive analysis has focused on the Aerodynamic Powered Landing Problem, outlining its positive aspects and limits. The research questions posed in Sec. 1.3 are therefore hereafter answered.

1) The Covector Mapping Theorem has proved its validity when an estimation of the costates is sought for. This feature is not altered neither by lossless, nor by successive convex formulations. The CMT, however, fails at providing accurate costates for complex formulations in a straightforward manner: employment of additional constraints to facilitate direct method convergence shall be well weighted; such constraints shall be ensured not to be active at convergence.

2) Optimization of lift forces allows to sensibly increase the gains in spared propellant mass; it is therefore a potentially valuable tool in rocket trajectory optimization. Nonetheless, the bare formulation of this problem carries intrinsically non-feasibility in rocket attitude that shall be counteracted with a more constrained scenario.

3) An *r-method* is a powerful tool to include non-regularities of primal-dual dynamics in an indirect formulation; lower order non-regularities within the searched control laws can be therefore efficiently handled, even in case of a *bang-bang* structures. A collocation scheme based on *r-method* grants high accuracy with fairly low computational requirements.

Such answers build on the work previously performed within the optimal control research field; specifically, coupling of convexification and CMT adds to the pillars

laid in the Optimal Powered Landing field [73, 40]; embedding a switching detection within an indirect collocation scheme opposes the direct-based techniques to handle *bang-bang* structures [27, 45]; PMP use reduces the dimension of the solved problem, eliminating the controls. Finally, the benefits of lift optimization in a 3-D environment ideally extend the previous analyses developed within the 2-D powered landing scenario [39].

6.1 Future developments

In the present work some points have been left open, thus constituting matter for future research. They are therefore outlined followingly, stemming from the research questions and building on the provided answers.

1a) How does the CMT behave if successive convexifications are handled augmenting the objective function with slacks on trust regions and virtual controls? [36] Does it still provide estimates in the same order of magnitude as the real costates?

1b) In [79] the CMT for the fLGR scheme is demonstrated excluding *pure state constraints*. Does it still hold if pure state constraints are added? What effect does their inclusion have on the estimated costates?

2a) What is the best approach to include lift, in a 3-D model, featuring thrust as upwards directed? Is constraining thrust direction feasible, thus modifying PMP to include conic constraints? Which performances are obtained if a first segment of engine shut off is instead allowed?

2b) How to include in the indirect problem formulation limitations relative to maximum thrust deflection angle with respect to body axis? In the present work the two directions have been kept separated, but explicit constraints between them allow for precise satisfaction of technological limitations.

3a) Which are the accuracy performances of the indirect collocation scheme if non-regularities of higher order with respect to discontinuities are accounted for? Then, How to tune efficiently the number of nodes per segment if the segments feature different lengths?

3b) How does the proposed collocation scheme behave if multiple bangs are present in the solution? Switching function has multiple roots, thus particular attention shall be paid to handling the constraints associated with null S_f .

3c) How can the collocation scheme speed be improved? And how much time can be spared with non constant trust regions?

A | Optimality Conditions with Path Constraints

Aim of the following appendix is to describe the necessary conditions arising from path constraints introduction in the optimal control problem. They have already been partially treated with the PMP introduction; nonetheless, it is worth providing a deeper analysis to fully understand their meaning and the reasons behind the inclusion of the path constraints in the Hamiltonian. Further clarifications and explanations are available in literature [63, 64].

A.1 Mixed constraints - Erdmann conditions

Let us consider a generic scalar *mixed constraint*, hypothesized to be active, with scalar control; it reads

$$g_m(\mathbf{x}, u, t) = 0 \quad (\text{A.1})$$

Given the optimal solution, neighbouring solutions shall still satisfy $\delta g_m = 0$ and the *perturbation differential equations* for $\delta \mathbf{x}$. Substituting the virtual variation of the control from $\delta g_m = 0$ in the perturbation equations one obtains [63]

$$\frac{d}{dt}(\delta \mathbf{x}) = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}} - \frac{\partial \mathbf{f}}{\partial u} \left(\frac{\partial g_m}{\partial u} \right)^{-1} \frac{\partial g_m}{\partial \mathbf{x}} \right] \delta \mathbf{x} \quad (\text{A.2})$$

which is the constraint-compliant time derivative of the variation $\delta \mathbf{x}$. Accordingly, costate dynamics read

$$\dot{\boldsymbol{\lambda}} = - \left[\frac{\partial \mathcal{L}}{\partial \mathbf{x}} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} - \frac{\partial \mathbf{f}}{\partial u} \left(\frac{\partial g_m}{\partial u} \right)^{-1} \frac{\partial g_m}{\partial \mathbf{x}} \right]^T \boldsymbol{\lambda} \quad (\text{A.3})$$

where the inequality expressed by the following equation shall be satisfied to guarantee each admissible δu determines the cost function to increase

$$\frac{(\partial\mathcal{L}/\partial u) + \boldsymbol{\lambda}^T(\partial\mathbf{f}/\partial u)}{\partial g_m/\partial u} < 0 \quad (\text{A.4})$$

Finally, costate boundary conditions and transversality conditions read

$$\boldsymbol{\lambda}_f = \frac{\partial\Phi_f}{\partial\mathbf{x}_f^T} \quad \mathcal{L}_f + \boldsymbol{\lambda}_f^T\dot{\mathbf{x}}_f = -\frac{\partial\Phi_f}{\partial t_f} \quad (\text{A.5})$$

When constraint is instead inactive, (A.3) is simply substituted by

$$\dot{\boldsymbol{\lambda}} = -\left(\frac{\partial\mathcal{L}}{\partial\mathbf{x}} + \frac{\partial\mathbf{f}}{\partial\mathbf{x}}\right)^T \boldsymbol{\lambda} \quad (\text{A.6})$$

Whenever the constraint switches from inactive to active, indeed, costate dynamics switches from (A.6) to (A.3), with the additional constraint in (A.4). This set of conditions is known as *Erdmann corner conditions*. It is evident that these conditions imply discontinuities in the control, thus in the state and costate dynamics. Nonetheless, state, costate remain time-continuous.

By introducing the Hamiltonian as in (2.15), the control equation provides for active constraint [64]

$$\mu = -\frac{(\partial\mathcal{L}/\partial u) + \boldsymbol{\lambda}^T(\partial\mathbf{f}/\partial u)}{\partial g_m/\partial u} \quad (\text{A.7})$$

thus μ shall satisfy $\mu > 0$ when mixed constraint is active.

On the other hand, for inactive mixed constraint, μ shall be $\mu = 0$ for the Hamiltonian form of the costate dynamics to be equivalent to (A.6).

It is finally evident that including the mixed constraints in the Hamiltonian makes the Hamiltonian formulation equivalent to the Lagrangian one, provided that conditions (2.18) are added to the optimality necessary conditions. *Erdmann corner conditions* have been included in problem formulation.

Remark A.1: *Erdmann conditions* do not alter the TPBV nature of the rearranged OCP. On the other hand, the uncertainty over number and position of junction points increases the complexity of practically solving the TPBVP; the same difficulty arises from the PMP, as it stems from inequality constraints as well. Practical ad-hoc techniques shall be then employed to overcome such difficulty [98].

A.2 DAE index reduction

Until this point it has been hypothesized the jacobian of active constraints $\partial \mathbf{g}_a / \partial \mathbf{u}^T$ to be full rank. Intervals over which such condition is not satisfied are called *singular subarcs*. Solving the singular case problem requires a differential approach to overcome the higher complexity.

In the Lagrangian augmentation \mathbf{g}_a can be substituted with its lowest order time derivative $\mathbf{g}_a^{(q)}$ such that $\partial \mathbf{g}_a^{(q)} / \partial \mathbf{u}^T$ is full rank; in addition, lower order derivatives shall be nulled at an internal point in the singular subarc. Such procedure is defined as *DAE index reduction* [65]. An example of the aforementioned procedure follows, in the application of a case of practical interest.

A.2.1 Pure state constraints - Jump conditions

Presence of *pure state constraints* is a sufficient condition for active constraints jacobian singularity to be verified. In addition, they are widely diffused in advanced OCP formulations of the powered descent and landing problem [99, 84, 36].

The process developed in section A.1 following the additional conditions provided in section A.2 can be employed to treat *pure state constraints*.

Let us consider an active scalar constraint, which reads

$$S(\mathbf{x}, t) = 0 \tag{A.8}$$

Similarly to what done before, Hamiltonian could be augmented with S ; however, S does not explicitly depend on the control: control equations solution would not be influenced by the presence of S in the Hamiltonian. On the other hand, including $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u, t)$ in (A.8) would allow to include the control, thus allowing for treating the pure state constraint as done before for the mixed constraint. This can be done by substituting (A.8) with an equivalent differential expression.

Let t_1 and t_2 be the times corresponding to constraint *entering* and *exiting corner*; enforcing (A.8) from t_1 to t_2 is equivalent to nulling the first $q - 1$ derivatives of S

at t_1 and nulling q -th derivative along the constrained arc:

$$S(\mathbf{x}, t) = 0 \quad \text{with } t \in [t_1, t_2] \quad \iff \quad \begin{cases} \mathbf{N}(\mathbf{x}_1, t_1) = \begin{bmatrix} S^{(0)} \\ S^{(1)} \\ \vdots \\ S^{(q-1)} \end{bmatrix}_{t=t_1} = \mathbf{0} \\ S^{(q)}(\mathbf{x}, u, t) = 0 \quad \text{with } t \in [t_1, t_2] \end{cases} \quad (\text{A.9})$$

where $S^{(q)}$ is the first S time derivative explicitly including the control u ; in addition $\mathbf{x}_1 = \mathbf{x}(t_1)$; such constraint is called *q-th order state constraint*.

At this stage $S^{(q)}$ substitutes g_m in the analysis of section A.1, and it can be embedded in the Hamiltonian [64]; it is left imposing $\mathbf{N}(\mathbf{x}_1, t_1) = \mathbf{0}$, which makes up an *interior-point constraint*.

Interior point constraints can be treated as boundary conditions: terminal cost shall be augmented with $\mathbf{N}(\mathbf{x}_1, t_1)$, and associated vector of Lagrange multipliers $\boldsymbol{\nu}_1 \in \mathbb{R}^q$; integral term over $[t_0, t_f]$ can be substituted by the summation of integrals over the intervals $[t_0, t_{1-}]$, $[t_{1+}, t_{2-}]$ and $[t_{2+}, t_f]$; nulling $d\hat{\mathcal{J}}$ requires that

$$\begin{aligned} \boldsymbol{\lambda}_{1-}^T &= \boldsymbol{\lambda}_{1+}^T + \boldsymbol{\nu}_1^T \frac{\partial \mathbf{N}}{\partial \mathbf{x}_1} \\ \boldsymbol{\lambda}_{2-} &= \boldsymbol{\lambda}_{2+} \\ \mathcal{H}_{1-} &= \mathcal{H}_{1+} - \boldsymbol{\nu}_1^T \frac{\partial \mathbf{N}}{\partial t_1} \\ \mathcal{H}_{2-} &= \mathcal{H}_{2+} \end{aligned} \quad (\text{A.10})$$

Such interior point conditions are defined *jump conditions* and shall be added to the boundary conditions in (2.17) to complete problem formulation. A more punctual demonstration of (A.10) is provided by [63].

Remark A.2: *Jump conditions* consist of interior-point constraints: these modify the nature of the OCP equivalent BVP; the double constraint is indeed substituted by a multiple constraint. A problem with internal active pure state constraints is indeed a MPBVP: discontinuities in costates and Hamiltonian sum to difficulties dictated by the non-definition of constrained subarcs boundaries.

B | Optimal Thrust Program for Quadratic Objective Function

The following derivation follows the structure of the optimal control retrieval for the thrust program. Lawden's primer vector theory is still valid. Letting $\hat{\alpha} = T_{max}/I_{sp}g_0$, $\tilde{\mathcal{H}}_{T,\varepsilon}$ results

$$\tilde{\mathcal{H}}_{T,\varepsilon} = \hat{\alpha} \left(\frac{\varepsilon}{m} - \lambda_m - \frac{I_{sp}g_0}{m} \lambda_v \right) u_T + (1 - \varepsilon) \frac{1}{m^2} \hat{\alpha}^2 u_T^2 \quad (\text{B.1})$$

Since $\varepsilon < 1$, then $\frac{\partial^2 \tilde{\mathcal{H}}_{T,\varepsilon}}{\partial u_T^2} > 0 \forall \varepsilon$. The theoretical optimal thrust magnitude $u_T^{(th)}$ can be therefore retrieved imposing $\frac{\partial \tilde{\mathcal{H}}_{T,\varepsilon}}{\partial u_T} = 0$. Such last condition provides

$$\begin{aligned} 0 = \hat{\alpha} \left(\frac{\varepsilon}{m} - \lambda_m - \frac{I_{sp}g_0}{m} \lambda_v \right) + (1 - \varepsilon) \frac{2}{m^2} \hat{\alpha}^2 u_T^{(th)} &\implies \\ \implies u_T^{(th)} = \frac{\frac{\varepsilon}{m} - \lambda_m - \frac{I_{sp}g_0}{m} \lambda_v}{(1 - \varepsilon) \frac{2}{m^2} \hat{\alpha}} & \quad (\text{B.2}) \end{aligned}$$

However, (B.2) does not include saturation of u_T , which shall therefore be added on top of such condition. The optimal thrust profile, compatible with thrust limitations, results therefore

$$u_T^* = \begin{cases} u_{T,\min} & \text{if } u_T^{(th)} < u_{T,\min} \\ u_T^{(th)} & \text{if } u_{T,\min} < u_T^{(th)} < u_{T,\max} \\ u_{T,\max} & \text{if } u_T^{(th)} > u_{T,\max} \end{cases} \quad (\text{B.3})$$

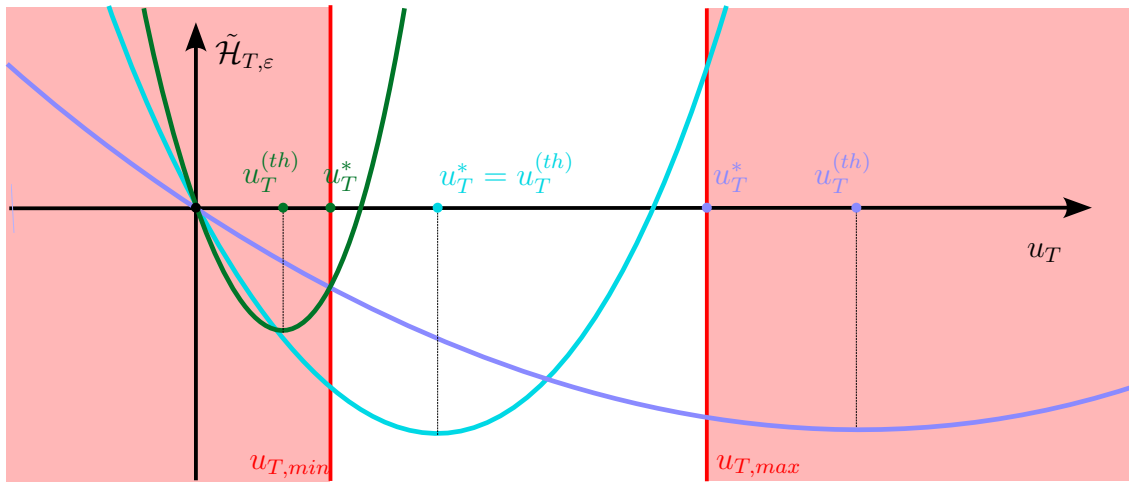


Figure B.1: Cases for the application of the PMP to the quadratic formulation

$$\begin{aligned}
 \text{Dark green: } & u_T^{(th)} < u_{T,\min} \\
 \text{Light blue: } & u_{T,\min} < u_T^{(th)} < u_{T,\max} \\
 \text{Violet: } & u_T^{(th)} > u_{T,\max}
 \end{aligned}$$

Cases in (B.3) are graphically reported in Figure B.1; the three parabolas correspond to three different expressions of the function in (B.1): they all feature positive concavity and have a root in the origin.

C | Parameterization of Lift Versor in \mathbf{v} and σ

Parameterization of lift versor \mathbf{i}_L requires accounting for velocity \mathbf{v} components if an aerodynamic angles-like expression of \mathbf{i}_L is pursued. Indeed, \mathbf{i}_L is orthogonal to vector \mathbf{v} : its expression in the inertial frame is therefore dependant on \mathbf{v} components. Bank angle is not constrained: any versor orthogonal and rigid with respect to \mathbf{v} is sufficient to introduce the bank angle influence. In our case, \mathbf{v}/v is obtained rotating the z -versor of the inertial frame \mathbf{i}_z according to

$$\mathbf{v}/v = \begin{bmatrix} \cos \delta & 0 & \sin \delta \\ 0 & 1 & 0 \\ -\sin \delta & 0 & \cos \delta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix} \mathbf{i}_z = \begin{bmatrix} \cos \delta \cos \gamma \\ -\sin \gamma \\ \sin \delta \cos \gamma \end{bmatrix} \quad (\text{C.1})$$

The previous equation leads

$$\tan \delta = \frac{v_z}{v_x} \quad \text{hence, for } -\pi/2 < \delta < \pi/2, \quad \begin{cases} \cos \delta = \frac{1}{\sqrt{1 + \frac{v_z^2}{v_x^2}}} \\ \sin \delta = \frac{v_z/v_x}{\sqrt{1 + \frac{v_z^2}{v_x^2}}} \end{cases} \quad (\text{C.2})$$

Bank angle identifies rotations about \mathbf{v}/v ; the matrix $\mathbb{R}_\sigma^I(\mathbf{v}, \sigma)$ of the eigenrotations of bank angle σ in the inertial reference frame exploits therefore \mathbf{v}/v as eigenaxis and σ as Euler angle, and satisfies

$$\mathbf{v}/v = \mathbb{R}_\sigma^I(\mathbf{v}, \sigma) \mathbf{v}/v \quad (\text{C.3})$$

\mathbf{i}_L can be finally assembled applying rotations Eq. (C.1) first and Eq. (C.3) in second instance on the inertial x -versor \mathbf{i}_x . They read

$$\begin{aligned} \mathbf{i}_L(\mathbf{v}, \sigma) &= \mathbb{R}_\sigma^I(\mathbf{v}, \sigma) \begin{bmatrix} \cos \delta & 0 & \sin \delta \\ 0 & 1 & 0 \\ -\sin \delta & 0 & \cos \delta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \\ &= \mathbb{R}_\sigma^I(\mathbf{v}, \sigma) \begin{bmatrix} \cos \delta \\ 0 \\ -\sin \delta \end{bmatrix} = \mathbb{R}_\sigma^I(\mathbf{v}, \sigma) \begin{bmatrix} \frac{1}{\sqrt{1 + \frac{v_z^2}{v_x^2}}} \\ 0 \\ -\frac{v_z/v_x}{\sqrt{1 + \frac{v_z^2}{v_x^2}}} \end{bmatrix} \end{aligned} \quad (\text{C.4})$$

Bibliography

- [1] Slyuta, E. “The Luna program”. In: *Sample Return Missions*. Ed. by Andrea Longobardo. Elsevier, 2021. Chap. 3. DOI: 10.1016/B978-0-12-818330-4.00003-3.
- [2] Siddiqi, A.A. *Beyond Earth: A Chronicle of Deep Space Exploration, 1958-2016*. National Aeronautics and Space Administration, Office of Communications, NASA History Division, 2018.
- [3] National Aeronautics and Space Administration Science and Technology Division. *Astronautics and Aeronautics, 1966*. NASA, 1967.
- [4] Soffen, G. A. and Snyder, C. W. “The First Viking Mission to Mars”. In: *Science* 193 (1976). DOI: 10.1126/science.193.4255.759.
- [5] Prakash, R. and Burkhart, P. Dan and Chen, A. and Comeaux, K. A. and Guernsey, C. S. and Kipp, D. M. and Lorenzoni, L. V. and Mendeck, G. F. and Powell, R. W. and Rivellini, T. P. and Martin, A. M. S. and Sell, S. W. and Steltzner, A. D. and Way, D. W. “Mars Science Laboratory Entry, Descent, and Landing System Overview”. In: *2008 IEEE Aerospace Conference*. 2008. DOI: 10.1109/AERO.2008.4526283.
- [6] Scharf, D. P. and Regehr, M. W. and Vaughan, G. M. and Benito, J. and Ansari, H. and Aung, M. and Johnson, A. and Casoliva, J. and Mohan, S. and Dueri, D. and Açıkmese, B. and Masten, D. and Nietfeld, S. “ADAPT demonstrations of onboard large-divert Guidance with a VTVL rocket”. In: *2014 IEEE Aerospace Conference*. 2014. DOI: 10.1109/AERO.2014.6836462.
- [7] Li, S. and Jiang, X. and Tao, T. “Guidance Summary and Assessment of the Chang’e-3 Powered Descent and Landing”. In: *Journal of Spacecraft and Rockets* 53.2 (2016). DOI: 10.2514/1.A33208.
- [8] Huang, X. and Li, M. and Wang, X. and Hu, J. and Zhao, Y. and Guo, M. and Xu, C. and Liu, W. and Wang, Y. and Hao, C. and Xu, L. “The Tianwen-1 Guidance, Navigation, and Control for Mars Entry, Descent, and Landing”. In: *Space: Science & Technology 2021* (2021). DOI: 10.34133/2021/9846185.

- [9] Larrimer, B. I. *Promise Denied - NASA's X-34 and the Quest for Cheap, Reusable Access to Space*. NASA Aeronautics Book Series. National Aeronautics and Space Administration, 1994.
- [10] Sagliano, M. "Open-Source Visualization of Reusable Rockets Motion: Approaching Simulink - FlightGear Co-simulation". In: *AIAA Scitech 2021 Forum* (2020).
- [11] Charette, R. O. and Steinmeyer, D. A. and Smiljanic, R. R. "Delta clipper lessons learned for increased operability in reusable space vehicles". In: *AIP Conference Proceedings*. Vol. 420. 1998. DOI: 10.1063/1.54901.
- [12] Nonaka, S. and Nishida, Y. and Kato, H. and Ogawa, H. and Inatani, Y. "Vertical Landing Aerodynamics of Reusable Rocket Vehicle". In: *Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan* 10 (2012), pp. 1–4. DOI: 10.2322/tastj.10.1.
- [13] Wagner, E. "Research Flights on Blue Origin's New Shepard". In: *Gravitational and Space Research* 9 (2021). DOI: 10.2478/gsr-2021-0005.
- [14] Reddy, V. S. "The SpaceX Effect". In: *New Space* 6.2 (2018). DOI: 10.1089/space.2017.0032.
- [15] Sagliano, Marco and Dumke, Michael and Theil, Stephan. "Simulations and Flight Tests of a New Nonlinear Controller for the EAGLE Lander". In: *Journal of Spacecraft and Rockets* 56 (2019). DOI: 10.2514/1.A34161.
- [16] Rmili, B. and Monchaux, D. and Boisneau, O. and Hassin, J. and Querry, S. and Besson, S. and Poirey, G. and Boré, R. and Hamada, I. and Amrouchi, H. and Franc, J. and Barreau, M. and Mercadié, N. and Labois, T. and Grinco, D. "FROG, a Rocket for GNC demonstrations: Firsts flights attempts of the FROG turbojet version and preparation of the future mono-propellant rocket engine". In: *EUCASS 2019*. July 2019. DOI: 10.13009/EUCASS2019-197.
- [17] Spannagl, L. and Hampp, E. and Carron, A. and Sieber, J. and Pascucci, C. A. and Zraggen, A. U. and Domahidi, A. and Zeilinger, M. N. "Design, Optimal Guidance and Control of a Low-cost Re-usable Electric Model Rocket". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021. DOI: 10.1109/IROS51168.2021.9636430.
- [18] deMirand, A. P. and Bahu, J. and Louaas, E. "Ariane Next, a vision for a reusable cost efficient European rocket". In: *EUCASS 2019*. July 2019. DOI: 10.13009/EUCASS2019-949.

- [19] Dumont, E. and Ecker, T. and Chavagnac, C. and Witte, L. and Windelberg, J. and Klevanski, J. and Giagkozoglou, S. “CALLISTO - Reusable VTVL launcher first stage demonstrator”. In: *Space Propulsion Conference 2018*. 2018.
- [20] 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”. In: *EUCASS 2019*. July 2019.
- [21] Stappert, S. and Wilken, J. and Bussler, L. and Sippel, M. “A Systematic Comparison of Reusable First Stage Return Options”. In: *EUCASS 2019*. July 2019.
- [22] De Zaiacomo, G. and Blanco Arnao, G. and Bunt, R. and Bonetti, D. “Mission engineering for the RETALT VTVL launcher”. In: *CEAS Space Journal* (2022). DOI: 10.1007/s12567-021-00415-y.
- [23] Huneker, L. and Sagliano, M. and Arslantas, Y. “SPARTAN: An Improved Global Pseudospectral Algorithm for High-Fidelity Entry-Descent-Landing Guidance Analysis”. In: *30th ISTS*. 2015.
- [24] Sagliano, M. “Performance analysis of linear and nonlinear techniques for automatic scaling of discretized control problems”. In: *Operations Research Letters* 42 (2014). DOI: 10.1016/j.orl.2014.03.003.
- [25] Sagliano, M. and Theil, S. and Bergsma, M. and D’Onofrio, V. and Whittle, L. and Viavattene, G. “On the Radau pseudospectral method: theoretical and implementation advances”. In: *CEAS Space Journal* 9 (June 2017). DOI: 10.1007/s12567-017-0165-5.
- [26] Rao, A. V. and Benson, D. A. and Darby, C. and Patterson, M. A. and Francolin, C. and Sanders, I. and Huntington, G. T. “GPOPS, A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method”. In: 37 (2010). DOI: 10.1145/1731022.1731032.
- [27] Agamawi, Y. M. and Rao, A. V. “CGPOPS: A C++ Software for Solving Multiple-Phase Optimal Control Problems Using Adaptive Gaussian Quadrature Collocation and Sparse Nonlinear Programming”. In: *ACM Trans. Math. Softw.* 46 (2020). DOI: 10.1145/3390463.
- [28] Brendel, E. and Hérissé, B. and Bourgeois, E. “Optimal guidance for Toss Back concepts of Reusable Launch Vehicles”. In: *EUCASS 2019*. July 2019. DOI: 10.13009/EUCASS2019-232.

- [29] 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.
- [30] Ross, I. M. *Enhancements to the DIDO Optimal Control Toolbox*. 2020. DOI: 10.48550/ARXIV.2004.13112.
- [31] Bedrossian, N. S. and Bhatt, S. and Kang, W. and Ross, I. M. “Zero-Propellant Maneuver Guidance”. In: *IEEE Control Systems Magazine* 29.5 (2009). DOI: 10.1109/MCS.2009.934089.
- [32] Malyuta, D. and Reynolds, T. P. and Szmuk, M. and Lew, T. and Bonalli, R. and Pavone, M. and Açıkmeşe, B. *Convex Optimization for Trajectory Generation*. 2021. DOI: 10.48550/ARXIV.2106.09125.
- [33] Malyuta, D. and Reynolds, T. and Szmuk, M. and Mesbahi, M. and Açıkmeşe, B. and Carson, J. M. “Discretization Performance and Accuracy Analysis for the Rocket Powered Descent Guidance Problem”. In: *AIAA Scitech Forum*. 2019. DOI: 10.2514/6.2019-0925.
- [34] Hofmann, C. and Morelli, A. C. and Topputo, F. “On the Performance of Discretization and Trust-Region Methods for On-Board Convex Low-Thrust Trajectory Optimization”. In: *AIAA Scitech Forum*. 2022. DOI: 10.2514/6.2022-1892.
- [35] Mao, Y. and Szmuk, M. and Xu, X. and Açıkmeşe, B. *Successive Convexification: A Superlinearly Convergent Algorithm for Non-convex Optimal Control Problems*. 2018. DOI: 10.48550/ARXIV.1804.06539.
- [36] 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”. In: *AIAA Scitech Forum*. Jan. 2021. DOI: 10.2514/6.2021-0862.
- [37] 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.
- [38] Yang, R. and Liu, X. “Comparison of Convex Optimization-Based Approaches to Solve Nonconvex Optimal Control Problems”. In: *AIAA SciTech Forum*. 2019. DOI: 10.2514/6.2019-1666.
- [39] Liu, X. “Fuel-Optimal Rocket Landing with Aerodynamic Controls”. In: *Journal of Guidance, Control, and Dynamics* 42 (2019). DOI: 10.2514/1.G003537.

- [40] Sagliano, M. “Pseudospectral Convex Optimization for Powered Descent and Landing”. In: *Journal of Guidance, Control and Dynamics* 41.2 (Feb. 2018). DOI: 10.2514/1.G002818.
- [41] 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.
- [42] Malyuta, D. and Reynolds, T. and Szmuk, M. and Açikmeşe, B. and Mesbahi, M. “Fast Trajectory Optimization via Successive Convexification for Spacecraft Rendezvous with Integer Constraints”. In: *AIAA Scitech Forum*. Jan. 2020. DOI: 10.2514/6.2020-0616.
- [43] Liu, X. and Shen, Z. and Lu, P. “Entry Trajectory Optimization by Second-Order Cone Programming”. In: *Journal of Guidance, Control and Dynamics* 39.2 (Feb. 2016). DOI: 10.2514/1.G001210.
- [44] Sagliano, M. and Mooij, E. “Optimal drag-energy entry guidance via pseudospectral convex optimization”. In: *Aerospace Science and Technology* 117 (July 2021). DOI: 10.1016/j.ast.2021.106946.
- [45] 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.
- [46] Açikmeşe, 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”. In: *Concepts and Approaches for Mars Exploration*. Ed. by LPI Editorial Board. LPI Contributions. 2012.
- [47] Blackmore, L. and Açikmeşe, B. and Scharf, D. P. “Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization”. In: *Journal of Guidance, Control and Dynamics* 33.4 (Aug. 2010). DOI: 10.2514/1.47202.
- [48] 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.
- [49] Sagliano, M. “Development of a Novel Algorithm for High Performance Reentry Guidance”. PhD thesis. Universität Bremen, Jan. 2016.

- [50] Ascher, U.M. and Mattheij, R. M. M. and Russell, R. D. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Society for Industrial and Applied Mathematics, 1995.
- [51] Longuski, J.M. and Guzmán, J. and Prussing, J.E. *Optimal Control with Aerospace Applications*. Springer, 2014.
- [52] Betts, J. T. “Survey of Numerical Methods for Trajectory Optimization”. In: *Journal of Guidance, Control and Dynamics* 21.2 (Apr. 1998). DOI: 10.2514/2.4231.
- [53] Rockafellar, R.T. “Lagrange Multipliers and Optimality”. In: *SIAM Review* 35.2 (June 1993).
- [54] Jensen, J. L. W. V. “Sur les fonctions convexes et les inégalités entre les valeurs moyennes”. In: *Acta Mathematica* 30 (1906). DOI: 10.1007/BF02418571.
- [55] Boyd, S. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.
- [56] Szmuk, M. and Eren, U. and Açıkmeşe, B. “Successive Convexification for Mars 6-DoF Powered Descent Landing Guidance”. In: *AIAA Guidance, Navigation and Control Conference*. Jan. 2017. DOI: 10.2514/6.2017-1500.
- [57] M. Szmuk, T. P. Reynolds, and B. Açıkmeşe. “Successive Convexification for Real-Time Six-Degree-of-Freedom Powered Descent Guidance with State-Triggered Constraints”. In: *Journal of Guidance, Control, and Dynamics* 43 (2020). DOI: 10.2514/1.G004549.
- [58] Domahidi, A. and Chu, E. and Boyd, S. “ECOS: An SOCP Solver for Embedded Systems”. In: *European Control Conference (ECC)*. 2013. DOI: 10.23919/ECC.2013.6669541.
- [59] Nesterov, Y. and Nemirovskii, A. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.
- [60] Evans, L. C. *Partial Differential Equations*. American Mathematical Society, 2010.
- [61] Lanczos, C. *The Variational Principles of Mechanics*. University of Toronto Press, 1949.
- [62] Hairer, E. and Lubich, C. and Wanner, G. *Geometric Numerical Integration - Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, 2005.

- [63] Bryson, A. E. and Denham, W. F. and Dreyfus, S. E. “Optimal Programming Problems with Inequality Constraints I: Necessary Conditions for Extremal Solutions”. In: *AIAA Journal* 1.11 (Nov. 1963). DOI: 10.2514/3.2107.
- [64] Pesch, H.J. “A Practical Guide to the Solution of Real-Life Optimal Control Problems”. In: *Control and Cybernetics* 23 (July 1994). DOI: 10.1.1.53.5766.
- [65] Betts, J. T. *Practical Methods for Optimal Control and Estimation Using Non-linear Programming*. Society for Industrial and Applied Mathematics, 2010.
- [66] Bryson, A. E. and Ho, Y. *Applied Optimal Control - Optimization, Estimation and Control*. Taylor & Francis, 1975.
- [67] Whittle, P. “Likelihood and Cost as Path Integrals”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 53.11 (1991). DOI: 10.1111/j.2517-6161.1991.tb01842.x.
- [68] Ghilardi, L. *A Study on the Covector Mapping Principle for Low-Thrust Trajectory Optimisation*. M.Sc. Thesis. Politecnico di Milano, Faculty of Industrial and Information Engineering, Department of Aerospace Science and Technology, Degree in Space Engineering. Supervisor: F. Topputo, Co-supervisors: C. Hofmann, A. Mannocchi. 2021.
- [69] Pontryagin, L.S. *The Mathematical Theory of Optimal Processes*. John Wiley & Sons, 1962.
- [70] Rao, A. V. “A Survey of Numerical Methods for Optimal Control”. In: *Advances in the Astronautical Sciences* 135 (2010).
- [71] Kang, W. “The rate of convergence for a pseudospectral optimal control method”. In: *2008 47th IEEE Conference on Decision and Control*. 2008. DOI: 10.1109/CDC.2008.4738608.
- [72] Ross, I. M. “A Historical Introduction to the Covector Mapping Principle”. In: *Advances in the Astronautical Sciences*. Vol. 123. 2006.
- [73] Ross, I. M. and Fahroo, F. “Legendre Pseudospectral Approximations of Optimal Control Problems”. In: *New Trends in Nonlinear Dynamics and Control and their Applications*. 2003. DOI: {10.1007/978-3-540-45056-6_21}.
- [74] Rea, J. “Launch Vehicle Trajectory Optimization Using a Legendre Pseudospectral Method”. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. June 2003. DOI: {10.2514/6.2003-5640}.

- [75] Ross, I. M. “How to Find Minimum-Fuel Controllers”. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*. June 2004. DOI: {10.2514/6.2004-5346}.
- [76] Herman, A. L. and Conway, B. A. “Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules”. In: *Journal of Guidance, Control, and Dynamics* 19.3 (1996). DOI: {10.2514/3.21662}.
- [77] Topputo, F. and Zhang, C. “Survey of Direct Transcription for Low-Thrust Space Trajectory Optimization with Applications”. In: *Abstract and Applied Analysis* 2014 (June 2014). DOI: 10.1155/2014/851720.
- [78] Huntington, G. T. “Advancement And Analysis of a Gauss Pseudospectral Transcription for Optimal Control Problems”. PhD thesis. Massachusetts Institute of Technology, July 2007.
- [79] Garg, D. “Advances in global pseudospectral methods for optimal control”. PhD thesis. University of Florida, Aug. 2011.
- [80] Patterson, M. A. and Rao, A. V. “GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using Hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming”. In: *ACM Transactions on Mathematical Software* 41.1 (July 2013). DOI: 10.1145/2558904.
- [81] Lu, P. “Propellant-Optimal Powered Descent Guidance”. In: *Journal of Guidance, Control, and Dynamics* 41 (Dec. 2017), pp. 1–14. DOI: 10.2514/1.G003243.
- [82] .
- [83] NOAA. *U.S. Standard Atmosphere, 1976*. Tech. rep. NOAA-S/T 76-1562. Washington, D.C.: NOAA, NASA, USAF, Oct. 1976.
- [84] Simplicio, P. and Marcos, A. and Bennani, S. “Reusable Launchers: Development of a Coupled Flight Mechanics, Guidance, and Control Benchmark”. In: *Journal of Guidance, Control and Dynamics* 57.1 (Feb. 2020). DOI: 10.2514/1.A34429.
- [85] Lawden, D. F. “Optimal trajectories for space navigation”. In: 1963.
- [86] Robbins, H. M. “A Generalized Legendre-Clebsch Condition for the Singular Cases of Optimal Control”. In: *IBM Journal of Research and Development* 11.4 (1967). DOI: 10.1147/rd.114.0361.

- [87] Leparoux, C. and Hérisse, B. and Jean, F. *Structure of optimal control for planetary landing with control and state constraints*. 2022. DOI: 10.48550/ARXIV.2204.06794.
- [88] Vinh, N. X. “General theory of optimal trajectory for rocket flight in a resisting medium”. In: *Journal of Optimization Theory and Applications* 11.2 (1973). DOI: 10.1007/BF00935883.
- [89] Pan, B. and Lu, P. and Pan, X. and Ma, Y. “Double-Homotopy Method for Solving Optimal Control Problems”. In: *Journal of Guidance, Control, and Dynamics* 39.8 (2016). DOI: 10.2514/1.G001553.
- [90] Bertrand, R. and Epenoy, R. “New Smoothing Techniques for Solving Bang–Bang Optimal Control Problems — Numerical Results and Statistical Interpretation”. In: *Optimal Control Applications and Methods* 23.4 (2002). DOI: 10.1002/oca.709.
- [91] Jiang, F. and Baoyin, H. and Li, J. “Practical Techniques for Low-Thrust Trajectory Optimization with Homotopic Approach”. In: *Journal of Guidance, Control, and Dynamics* 35 (2012). DOI: 10.2514/1.52476.
- [92] Lu, P. and Sun, H. Tsai, B. “Closed-Loop Endoatmospheric Ascent Guidance”. In: *Journal of Guidance, Control and Dynamics* 26.2 (Apr. 2003). DOI: 10.2514/2.5045.
- [93] Morelli, A. C. and Hofmann, C. and Topputo, F. “Robust Low-Thrust Trajectory Optimization Using Convex Programming and a Homotopic Approach”. In: *IEEE Transactions on Aerospace and Electronic Systems* (Oct. 2021). DOI: 10.1109/TAES.2021.3128869.
- [94] Taheri, E. and Arya, V. and Junkins, J. L. “Costate mapping for indirect trajectory optimization”. In: *Astrodynamics* 5 (2021). DOI: 10.1007/s42064-021-0114-0.
- [95] Meditch, J. “On the problem of optimal thrust programming for a lunar soft landing”. In: *IEEE Transactions on Automatic Control* 9.4 (1964). DOI: 10.1109/TAC.1964.1105758.
- [96] Patterson, M. A. and Rao, A. V. “GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using Hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming”. In: *ACM Trans. Math. Softw.* 41 (2014). DOI: 10.1145/2558904.

- [97] McRae, D. S. “r-Refinement grid adaptation algorithms and issues”. In: *Computer Methods in Applied Mechanics and Engineering* 189 (2000). DOI: [https://doi.org/10.1016/S0045-7825\(99\)00372-2](https://doi.org/10.1016/S0045-7825(99)00372-2).
- [98] Zhang, C. and Topputo, F. and Bernelli-Zazzera, F. and Zhao, Y. “Low-Thrust Minimum-Fuel Optimization in the Circular Restricted Three-Body Problem”. In: *Journal of Guidance, Control and Dynamics* 38.8 (Aug. 2015). DOI: 10.2514/1.G001080.
- [99] Simplicio, P. and Marcos, A. and Bennani, S. “Guidance of Reusable Launchers: Improving Descent and Landing Performance”. In: *Journal of Guidance, Control and Dynamics* 42.10 (Aug. 2019). DOI: 10.2514/1.G004155.