POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

A study on the Covector Mapping Principle applying non-complete and pseudospectral methods

Laurea Magistrale in Space Engineering - Ingegneria Spaziale

Author: Silvia Busi

Advisor: Prof. Francesco Topputo

Co-advisor: Alessandra Mannocchi

Academic year: 2021-2022

1. Introduction

Two families of techniques are generally employed in the optimization of low-thrust trajectories, and are commonly classified as either direct or indirect methods. They have commonly been treated as separate approaches [4]. Solvers based on direct methods are capable of handling more complex problems, and are characterized by a wider convergence basin. However, trajectory optimization problems often involve a number of variables that is so large that can cause ill-conditioning [2]. On the other hand, indirect methods are generally considered as more accurate, but are characterized by poorer convergence properties [1]. Despite these different intrinsic features, the two families of methods stem from the same original continuous-time Optimal Control Problem (OCP). Therefore, they shall share an underlying connection, which might be exploited to overcome their intrinsic difficulties and lower computational time. An ideal algorithm would merge the robustness of direct methods with the accuracy of indirect ones, and this is exactly the aim of the Covector Mapping Principle (CMP), a novel approach to optimal control theory that works on the link between the dual variables of direct and indirect methods. However, researchers have demonstrated that only few methods provide a Covector Mapping Theorem (CMT). They are referred to as *complete methods*, and include *pseudospectral methods*. The natural question that arises is whether it is possible to employ *non-complete* methods to at least partially overcome the mentioned issues. The study developed in this thesis starts from optimization tools based on non-complete methods, investigating the presence of the bridge between dual variables. Then, the validity of the proposed link is verified passing both from a direct to an indirect method and viceversa. Finally, the obtained performances are compared with the results of the same problem solved with *Radau pseudospectral method*, which provides a CMT. The results demonstrate that the link can be exploited even without an analytical law.

2. Non-complete Method

In this section the analysis is performed adopting Hermite-Simpson collocation scheme in the context of direct formulation of DIRETTO [5] and SNOPT [3] solvers and variable step single-shooting Runge-Kutta 7-8 integration method within the indirect LT2.0 [6] optimization tool. The solvers formulate the OCP in Lagrange form, and a new version of DIRETTO has been developed to express the dynamics in Cartesian coordinates, in such a manner that the optimization tools leverage the same OCP

formulation. Note, however, that the solvers present a substantial difference, which is of paramount importance in the exploitation of the CMP. It consists in the application of a different discretization method, which influences the accuracy of the link between dual variables. However, the goal of this work is to exploit the link to improve convergence properties and lower computational cost, not to provide a CMT. Therefore, this difference needs to be understood and taken into consideration for the analysis. The aim of the developed algorithm is to prove both whether costates pass on the information of optimality to direct methods, while being an initial guess for Lagrange multipliers, and if robustness and flexibility of direct techniques can be exploited to solve the problem and to provide a good initial guess for the indirect costates. The main followed steps are described hereon.

1) Initialization The code allows to choose among different initial conditions coupled with associated boudary conditions, and corresponding time of flight. Before the beginning of the optimization process, the variables are adimensionalized according to characteristic units for numerical reasons. This step is common to LT2.0, DIRETTO, and SNOPT.

2) Run DIRETTO and LT2.0 A two-body Energy Optimal (EO) trajectory optimization problem is dealt both with LT2.0 and with DIRETTO without exchange of information between the solvers. The aim, indeed, is to ensure that the direct algorithm is as reliable as the indirect one. In the direct case, the resulting NLP problem is solved through $MATLAB^{®}$ `fmincon` interior-point algorithm to exploit sparsity of matrices.

Consider the optimal trajectory in Fig. 4, where it is evident that DIRETTO converges to LT2.0 optimal solution. The same can be stated for the control histories in Fig. 5, where throttle factor u, in-plane α and out-of-plane β angles are shown. Therefore, it can be affirmed that DIRETTO is reliable for the optimization process. However, it is verified that convergence to the solution obtained with LT2.0 is not reached if the number of nodes in DIRETTO is not sufficient. This issue is due to the very nature of direct collocation methods. Of course the number of discretization points shall be traded off with numerical issues arising from large-scale sparse matrix algebra, which could prevent the algorithm from reaching convergence.

3) Research $\boldsymbol{\lambda}_{LT2.0} \leftrightarrow \boldsymbol{\lambda}_{DIR}$ At this point, a correct transformation between indirect costates and direct Lagrange multipliers shall be identified to properly exploit the link between dual variables. Therefore, this step is the most important and critical of the entire algorithm.

Notice that OCP formulation in LT2.0 is different from that of the direct approach, and this affects the research. In fact, the expression of the augmented cost functional of the EO problem involved in LT2.0 solution is

$$\hat{J} = J_{EO}^{L} + \int_{t_0}^{t_f} \boldsymbol{\lambda}^T (\mathbf{f}(\mathbf{x}, \mathbf{u}, t) - \dot{\mathbf{x}}(t)) \mathrm{d}t \qquad (1)$$

where $\boldsymbol{\lambda}$ are the costate variables required to introduce the dynamics constraints, and J_{EO}^{L} is the Lagrange cost functional, expressed as

$$J_{EO}^{L} = \int_{t_0}^{t_f} \frac{T_{max}}{I_{sp} g_0} u^2 dt \qquad (2)$$

Instead, the Lagrangian involved in the direct formulation within DIRETTO, SNOPT, and also Rps-DIRECT, explained in Section 3, is expressed as follows.

$$L(\mathbf{y}, \tilde{\boldsymbol{\lambda}}) = J_{EO}^{L} - \tilde{\boldsymbol{\lambda}}^T \mathbf{c}(\mathbf{y}) \qquad (3)$$

The vector of NLP variables is denoted with $\boldsymbol{y}$, whereas $\tilde{\boldsymbol{\lambda}}$ indicates the Lagrange multipliers, that do not have to be confused with the costates. Vector $\boldsymbol{c}$ embeds all the constraints of the problem.

By comparing Eqs. (1) and (3), it is clear that

$$\tilde{\boldsymbol{\lambda}}(t_k) \approx -\boldsymbol{\lambda}(t_k) \qquad k = 1, ..., N-1 \qquad (4)$$

where t_k is the discrete time instant. The last node corresponding to $k = N$ is not included in the dynamics constraints, since they are computed at the grid mid-points. The approximation in Eq. (4) is considered as acceptable, although it seems less accurate than results reported in literature [1], which assume that both direct and indirect methods employ the same numerical scheme.

Results in Fig. 1 show that the shape of dual variables is essentially the same, and an almost exact superposition is observable. However, a scaling factor equal to approximately 4.1 between costates $\boldsymbol{\lambda}$ and Lagrange multipliers $\tilde{\boldsymbol{\lambda}}$ has been applied. It is observed in Fig. 2, that the ratio $\tilde{\boldsymbol{\lambda}}/\boldsymbol{\lambda}$ for each of the state variables is approximately equal to 4.1, with deviations from 4.1 due to dual variables assuming values close to zero, thus approaching an indeterminate form.

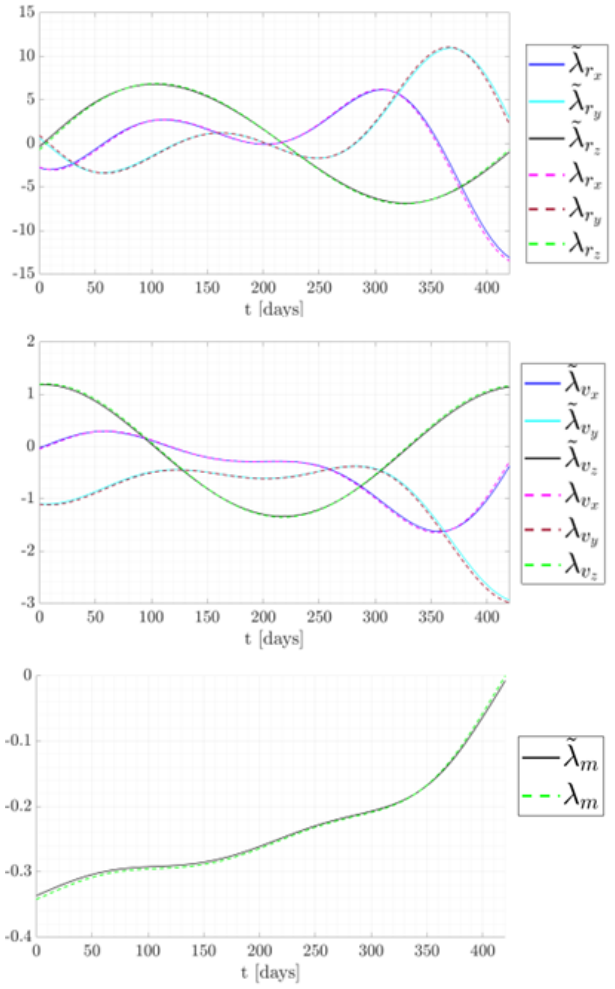

Figure 1: Map of costates with non-complete method. Lagrange multipliers (solid line) and costates (dashed line)

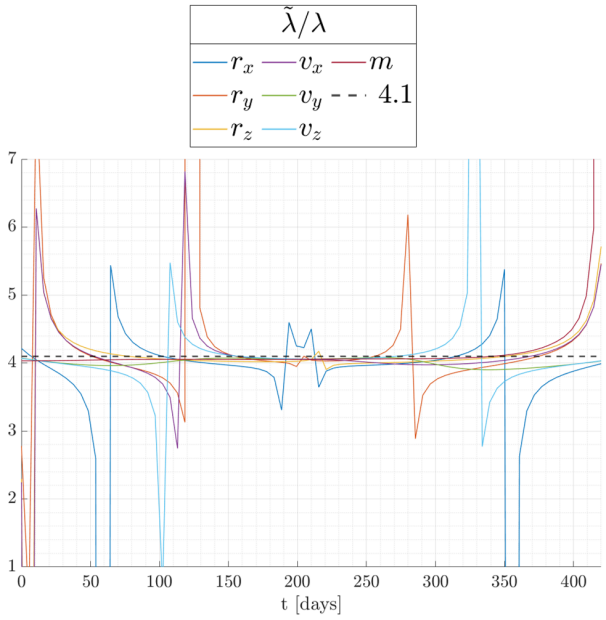

Figure 2: Scaling factor

Table 1: DIRETTO → LT2.0

	λ_{DIR}	$\lambda_{DIR_{mapped}}$
Step	0 → 12	0 → 7
f(x)	15.20 → 1.57e^{-26}	4.28 → 6.21e^{-27}
CPU Time†	0.627 s	0.372 s

† Relative to Intel® Core™ i9-11900H, RAM 16 GB

4) Run with different inputs Then, three different simulations are performed:
a) The initial costates obtained from the mapping of DIRETTO Lagrange multipliers are given as input to LT2.0, both with scaling factor, $\lambda_{DIR_{mapped}}$, and without it, λ_{DIR}, to verify whether LT2.0 still identifies a mathematical connection. The results in Table 1 show the value of the shooting function $f(x)$ at different simulation iterations and the computational time. Since number of both evaluated steps and CPU time is really limited in both cases, it is demonstrated that the link could be exploited avoiding the mapping.
b) Then, since it is not possible to provide DIRETTO with an initial guess of Lagrange multipliers, LT2.0 solution in terms of states and controls is discretized according to DIRETTO time intervals, and consequently used as input for the direct optimization process. The first step of the simulation within DIRETTO shows an objective function value comparable to the optimized one obtained by running DIRETTO with a free initial guess. However, constraints are not satisfied, and thus a new optimization procedure starts, leading to the same result obtained without inputs from LT2.0. Therefore, SNOPT tool is employed to prove whether a warm start is applicable to exploit the link from costates to Lagrange multipliers.
c) SNOPT allows to be given mapped costates from LT2.0 as input. The problem is then solved employing a SQP algorithm. Table 2 shows the results related to the non-initialization of SNOPT multipliers on the first column denoted with λ_{SN}, while those obtained with the warm start are reported on the second column ($\lambda_{LT2.0 \to SN}$). They show a marked improvement in the computational cost, proving the mapping to be successful also in the passage from indirect to direct methods. Note, however, that a consistent number of steps is required to reach the solution. This is mainly due to the difference in the discretization schemes adopted within the optimization tools. Moreover, it is important to highlight that both simulations show convergence to the solutions

obtained with LT2.0 and DIRETTO, as observable in Fig. 4 and Fig. 5.

Table 2: SNOPT results

	λ_{SN}	$\lambda_{LT2.0 \to SN}$
# Iterations	18775	1385
CPU Time†	8.035 s	0.127 s

† Relative to Intel® Core™ i9-11900H, RAM 16 GB

3. Pseudospectral Method

When the analysis using the non-complete formulation is concluded, Radau pseudospectral method is implemented in RpsDIRECT code. It adopts LGR scheme. Dynamics is expressed in Cartesian coordinates and OCP is formulated in Lagrange form to be consistent with the optimization tools previously employed. Even in this algorithm, the discretization method is different from the one used in LT2.0. However, since the link between dual variables is described by the CMT, it is expected that the limit posed by the discretization is at least partially overcome. At this point, the procedure followed for the analysis is presented.

1) Initialization The same conditions established in the first step of the non-complete method algorithm are used. This allows to solve the identical trajectory optimization problem, allowing a comparison.

2) Run RpsDIRECT The EO problem is solved using interior-point algorithm within *MATLAB*® `fmincon`, and some issues are encountered to reach convergence. In fact, `fmincon` results really sensitive to the inputs of lower and upper bounds of NLP variables. It requires quite precise values for ranges of α and β to reach a continuous solution. However, the final result does not respect these limits, as they only influence continuity. In the case of DIRETTO a suitable range leading to a valid solution has been identified, while with RpsDIRECT more difficulties are faced. Fig. 3 presents the controls obtained for simulations with different ranges of α and β. It is clear that discontinuous and different controls lead neither to a valid solution nor to the same optimization history. Therefore, Lagrange multipliers do not overlap.

These results do not allow further studies, since they do not validate the algorithm, in the sense that a solution converging to LT2.0 one is not found. Nevertheless, it is an issue of `fmincon`, and thus it shall be overcome to analyze the pseudospectral method

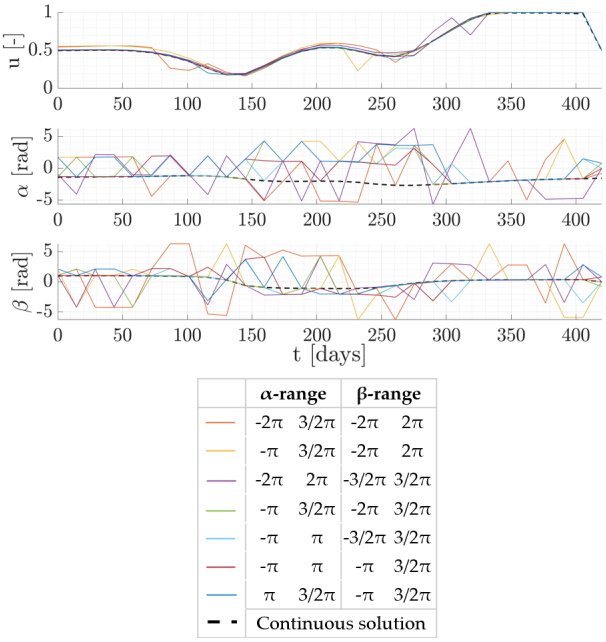

Figure 3: RpsDIRECT discontinuous controls with different bounds on α and β.

performance. Therefore, the initial guess for the controls is now changed from constant values to the values of DIRETTO optimized controls. The obtained results are shown in black-dashed lines in Fig. 3. The optimized trajectory in Fig. 4 clearly overlaps to the solution of LT2.0. However, the controls do not correspond exactly to those found in Section 2, as visible in Fig. 5. Moreover, notice that the controls at the last discretization point corresponding to t_f are not obtained. It is in accordance to LGR scheme, which does not enforce dynamics at the final bound of time domain.

As a conclusion, it can be stated that the solution does not correspond to the exact one. Then, either other simulations changing the initial ranges of α and β are run, or CMT is applied to study whether it can still be exploited. The latter procedure is followed.

3) CMT Note that the algorithm works in the pseudotime domain τ, and therefore a mapping back to the physical time is computed to perform this step. The correct application of the CMT for LGR scheme is the most important step of this algorithm. It is computed as follows, with the addition of the same change of sign obtained in Eq. (4).

$$\begin{aligned} \tilde{\boldsymbol{\lambda}}_f &= \mathbf{D}_n^{\mathrm{T}} \boldsymbol{\Lambda} \\ \tilde{\boldsymbol{\lambda}}_i &= \frac{\boldsymbol{\Lambda}_i}{w_i} \quad i = 0, \ldots, n-1 \end{aligned} \tag{5}$$

In Eq. (5), $\mathbf{D}_n^T$ is the n^{th} row of pseudospectral

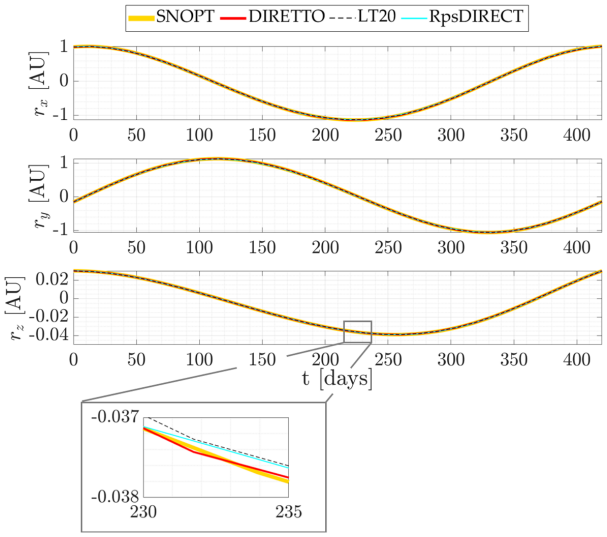

Figure 4: Optimized transfer.

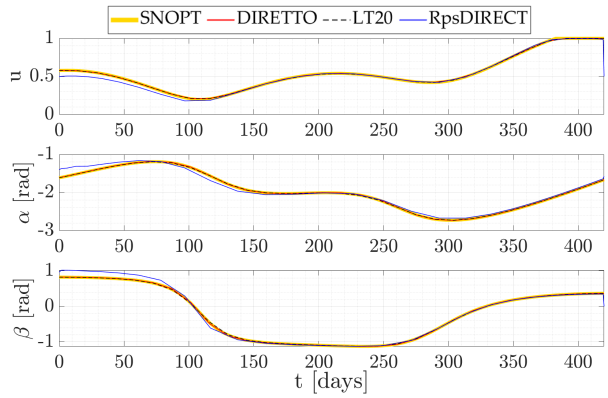

Figure 5: Optimized controls.

differentiation matrix $\boldsymbol{D}^T$, $\boldsymbol{\Lambda}_i$ are the multipliers associated with the dynamical constraints, and w_i the quadrature weights.

As expected, the shape of dual variables in Fig. 6 coincide. Nevertheless, the same scaling factor of 4.1 found in the research $\boldsymbol{\lambda}_{LT2.0} \leftrightarrow \boldsymbol{\lambda}_{DIRETTO}$ is present. Some differences are observable, particularly in the points where the controls do not overlap. It is in fact due to the nature of RpsDIRECT solution, namely the inaccurate convergence to LT2.0 results. However, next section analyzes whether the complete method still provides better performances when the mapped costates at t_0 are inputs for LT2.0 solver.

4) Run LT2.0 If the CMP works correctly, convergence to the optimal solution shall be reached in few iterations.

Two simulations are run within LT2.0 solver: the first does not involve the scaling factor, while the second one does. The inputs are therefore denoted as

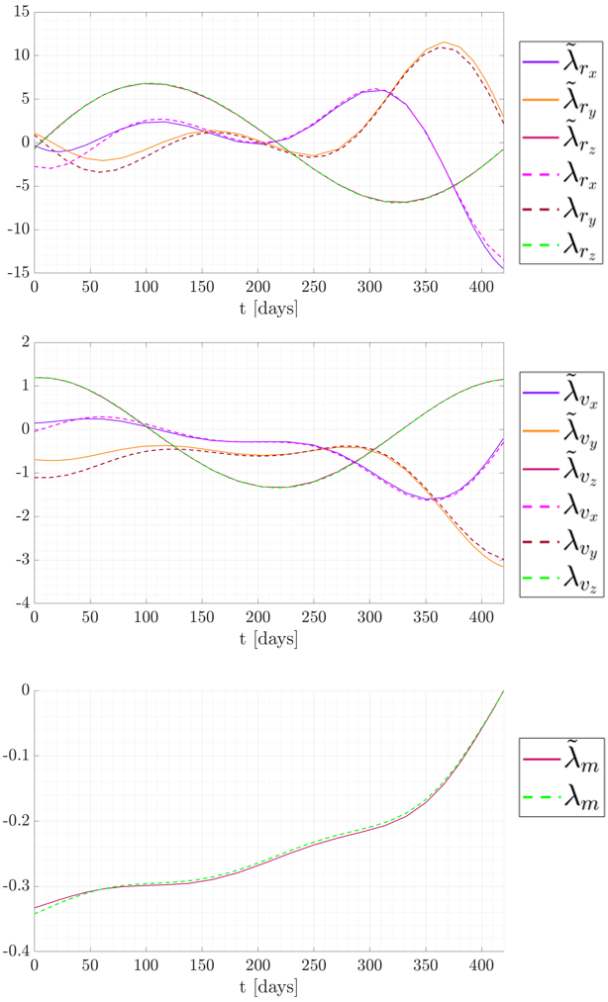

Figure 6: Map of costates with Radau PS method. Costates from CMT (solid line) and costates from LT2.0 (dashed line)

$\boldsymbol{\lambda}_{\boldsymbol{CMT}}$ and $\boldsymbol{\lambda}_{\boldsymbol{CMT_{mapped}}}$ respectively, and the results are shown in Table 3.

Table 3: CMT → LT2.0

	$\boldsymbol{\lambda}_{\boldsymbol{CMT}}$	$\boldsymbol{\lambda}_{\boldsymbol{CMT_{mapped}}}$
Step	$0 \rightarrow 10$	$0 \rightarrow 8$
$\mathbf{f}(\mathbf{x})$	$0.179 \rightarrow 1.60\text{e}^{-20}$	$0.163 \rightarrow 1.68\text{e}^{-20}$
CPU Time†	0.582 s	0.437 s

† Relative to Intel® Core™ i9-11900H, RAM 16 GB

Notice that both number of steps and computational time are comparable to the results obtained with the non-complete method. Nevertheless, the value of the shooting function is significantly lowered, due to the application of the analytical mapping offered by the CMT. Moreover, although the solution of RpsDIRECT is not exact, optimality is reached within LT2.0 tool, allowing for the application of the pseu-

dospectral method even with the issues arising from `fmincon`. The results in terms of controls are shown in Fig. 7. Notice that only the results obtained with $\boldsymbol{\lambda}_{CMT}$ are reported, as $\boldsymbol{\lambda}_{CMT_{mapped}}$ leads to the same solution.

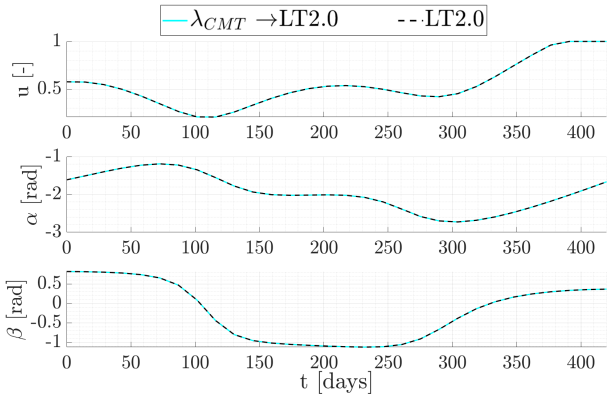

Figure 7: Optimized controls $\boldsymbol{\lambda}_{CMT} \rightarrow$ LT2.0

4. Conclusions

To conclude, it is verified that an empirical link between Lagrange multipliers and costate variables has been found. It can be exploited within non-complete methods passing from both direct to indirect method and viceversa, allowing to lower the computational time and to overcome the intrinsic limitations of the two families of approaches. The comparison with the complete Radau pseudospectral method shows that the CMT provides a value for the shooting function close to zero already at the first step of iteration. However, the computational cost is comparable to the one obtained employing non-complete approach. These remarks lead to conclude that it is possible to exploit the link between dual variables in spite of dealing with a non-complete method. Therefore, this result allows to pave the way for the continuation of this study in the future. Some open points constituting matter for future research are present:

1. `Fmincon` lower and upper bounds First of all, investigation on the $MATLAB^{®}$ function would allow to understand its functioning, and therefore establish a path to be followed to avoid discontinuities in the optimized controls.

2. Increase the number of LGR discretization points Concerning instead the pseudospectral formulation, it has been noticed that a high number of nodes causes ill-conditioning of Legendre polynomials. This issue can be overcome by dividing the time domain into two or more segments, and then performing global collocation on each of them. In this manner, controls may converge to the solution obtained with the indirect solver, further improving the performance of the algorithm.

3. Scaling factor To conclude, it has been proved that the mappings performed with both algorithms present a scaling factor of approximately 4.1. LT2.0 is only partially influenced by this factor, as the number of steps in the simulations does not significantly vary, and convergence is always reached. Moreover, the formulation of the problems is considered correct, as the shape of Lagrange multipliers and costates is the same. Therefore, it can be stated that the scaling factor does not significantly affect the performances of the indirect solver. Notice that this factor might derive from a different scaling applied internally in LT2.0 optimization process. Therefore, a new indirect algorithm should be implemented to understand if the problem lies in LT2.0 formulation.

References

[1] J. T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society for Industrial and Applied Mathematics, second edition, 2010. doi: 10.1137/1.9780898718577.

[2] B. A. Conway. *Spacecraft Trajectory Optimization*. Cambridge University Press, 2010. doi: 10.1017/CBO9780511778025.

[3] P. E. Gill, E. Wong, W. Murray, and M. A. Saunders. *User's Guide for SNOPT Version 7.7: Software for Large-Scale Nonlinear Programming*. 2018.

[4] D. Morante, M. Sanjurjo Rivo, and M. Soler. A Survey on Low-Thrust Trajectory Optimization Approaches. *Aerospace*, 8(3), 2021. doi: 10.3390/aerospace8030088.

[5] F. Topputo, D. A. Dei Tos, K. V. Mani, S. Ceccherini, C. Giordano, V. Franzese, and Y. Wang. Trajectory Design in High-Fidelity Models. In *ICATT*, Oberpfaffenhofen, Germany, 11 2018. 7th International Conference on Astrodynamics Tools and Techniques.

[6] Y. Wang. *Efficient Indirect Optimization of Low-Thrust Trajectories with Interior-Point Constraints*. PhD thesis, Politecnico di Milano., 2 2022.