75th International Astronautical Congress (IAC), Milan, Italy, 14-18 October 2024.

IAC-24,A6,IP,98,x82679

Exploring functional connections theory and linearized approaches in collision avoidance maneuver design: a comparative study

David Pérez López[*a], **Andrea De Vittori, Pierluigi Di Lizia**[b], **Daniele Mortari**[c]

[a]*Bioengineering and Aerospace Engineering, Universidad Carlos III de Madrid, 28911 Madrid, ES*
[b]*Aerospace Science and Technology, Politecnico Di Milano, 20156 Milan, IT*
[c]*Aerospace Engineering, Texas A&M University, College Station, 77843 Texas, US*
[*]Corresponding Author: david.p.lopez@alumnos.uc3m.es

Abstract

The increasing number of objects in orbit is leading to more instances of close encounters between satellites and debris. In this setting, Space Situational Awareness (SSA) plays a crucial role in establishing guidelines to ensure the preservation of the current space environment and vital assets. One of these measures is the design of Collision Avoidance Maneuvers (CAMs) to tackle potential conjunctions between a primary spacecraft and a secondary object. In the state of the art, many works have analyzed impulsive and low-thrust CAM planning leveraging motion linearization to speed up the solution process. However, the application of the Theory of Functional Connections (TFC) to address this challenge remains largely unexplored in the literature. To fill this gap, this study investigates the design of CAM through an Energy-Optimal Control Problem (EOCP), framed with the aid of TFC. The TFC methodology derives the so-called constrained functionals that analytically satisfy the imposed constraints. By separating the set of constraints from the nonlinear system of Differential Equations (DEs), the TFC translates the Two-Point Boundary Value Problem (TPBVP) into an unconstrained optimization problem. This paper aims to perform a comparison between the TFC-based solution and a fully analytical approach applied to the linearized dynamics. Two strategies shape the TFC CAM policy. On the one hand, the outer-loop approach solves the Lagrange multiplier by optimizing the TFC solution with the terminal condition. On the other, the single-loop approach embeds the Lagrange multiplier in the set of unknowns and the terminal condition acts as an additional loss function. The algorithm performance is evaluated in terms of computational cost, accuracy, convergence, and the resulting impulse required to perform the maneuver. The CAMs can be planned to meet different terminal conditions at the Time of Closest Approach (TCA), such as the Miss Distance (MD) or the Square Mahalanobis Distance (SMD). The latter leverages Chan's Probability of Collision (PoC) to set the safeguard limit and assumes constant covariances, short-term encounters, and a spherical object approximation. Moreover, the dependence of the solution on the configuration parameters of the TFC algorithm, such as the discretization of the domain, the number of basis functions, and the type of orthogonal polynomials (Chebyshev, Legendre) is studied through several sensitivity analyses.

Keywords: Space Debris, Collision Avoidance Maneuver, Theory of Functional Connections, Optimal Control.

1. Introduction

Space debris overpopulation threatens to compromise the space environment as a viable human resource. Despite the fact that very few events are associated with collisions, it is expected that they will become the dominant source of space debris in the future. Notably, the space debris population would continue to increase even if no more satellites were put into orbit since collisions between the existing debris will continue to occur. As the number of objects in space increases, so does the probability of collision between them.

As part of the possible preventive actions, let us explore collision avoidance maneuvers, a countermeasure that is of interest to this study. These maneuvers fall under trajectory optimization, where the goal is to chart a path that meets specific initial and terminal conditions while minimizing relevant quantities, typically the amount of fuel needed to perform the maneuver.

The study of optimizing low-thrust Collision Avoidance Maneuvers (CAMs) is well-established. Gonzalo et al. [1] developed a semi-analytical solution to maximize miss distance by utilizing average dynamics, assuming continuous tangential thrust. C. Bombardelli and J. Hernando-Ayuso [2] tackle the optimal low-thrust collision avoidance problem for two objects in circular orbits with constant thrust magnitude. Andrea De Vittori et al. [3] derived a fully analytical approach for energy-optimal CAMs leveraging motion linearization in both Earth-centered inertial (ECI) and B-plane coordinates, providing a good guess for the fuel-optimal solution.

The Theory of Functional Connections (TFC) has been extensively applied in research on optimal control

problems and astrodynamics. In [4], R. Furfaro and D. Mortari introduce the use of TFC to solve Two-Point Boundary Value Problems (TPBVPs) derived from indirect optimal control methods. This pioneering application addresses various optimal guidance issues, including energy-efficient landings on planetary bodies with fixed TFC loop times and fixed-time optimal intercepts in target-interceptor scenarios. Schiassi et al. [5] present a unified method for determining energy-optimal landing trajectories on various planetary bodies, including planets, asteroids, and comets. In [6], H. Johnston and D. Mortari apply the TFC to the perturbed Lambert's problem. Their method starts with an unperturbed solution as a baseline and incorporates perturbations using constrained functionals. Johnston and Mortari [7] explore the application of the TFC to perturbed orbit propagation, extending from their work on Lambert's problem. The study finds that while TFC performs comparably to other techniques, it is particularly well-suited for boundary-value problems.

This thesis explores the application of TFC to solve the energy-optimal CAM problem, marking a significant first step towards integrating multiple constraints into the mission design. While traditional approaches have largely focused on linearized models and analytical tools, this research addresses the original nonlinear equations using a numerical scheme framed via the TFC. By demonstrating the capability to meet typical terminal conditions for reducing collision risk in orbital encounters, this work lays the groundwork for more advanced approaches. The performance of the TFC-based strategy is then evaluated by comparing it to the linear-analytical method introduced in [3]. This comparison seeks to determine whether the nonlinear solutions significantly differ from the linear ones and to assess the potential of TFC in this specific context.

2. Fundamental principles

This section seeks to provide a grasp of the essential principles that lay the foundations of CAMs, including the mathematical frameworks employed to approach the problem.

The dynamical model employed to simulate the motion of an object orbiting the Earth is the so-called Restricted Two-Body Problem (RTBP). Assuming that the object's mass is negligible compared to the Earth's mass, $(m_2 \ll m_1)$, the state-space equations describing the trajectory of m_2 in a reference frame (r.f.) centered at m_1 are:

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = -\dfrac{\mu}{r^3}\mathbf{r} \end{cases} \tag{2.1}$$

where $\mu = G\left(m_1 + m_2\right) \approx Gm_1$ is the gravitational parameter.

2.1 Conjunction definition

The state vectors of the centre of mass of the primary and secondary objects respectively, expressed in a generic r.f. $\mathfrak{R}$, are defined as $\mathbf{x}_p = (\mathbf{r}_p, \mathbf{v}_p)$ and $\mathbf{x}_s = (\mathbf{r}_s, \mathbf{v}_s)$. At a generic time, the motion of each object is given by Eq. (2.1). In this work, $\mathfrak{R}$ denotes the ECI r.f.. The shape and attitude of space objects, especially debris, are generally difficult to assess. For this reason, the collision domain of each body is modelled according to the spherical geometry hypothesis: $\mathrm{O_p}$ and $\mathrm{O_s}$ are regarded as spheres of radii R_p and R_s respectively, centred in their centre of mass.

2.1.1 Short-term encounter

An encounter is considered short-term if the conjunction duration is negligible compared to the orbital period of the objects involved. To evaluate such a condition, the conjunction duration (t_c) is defined according to [8], and represents the time required for the primary object to cross the 1σ relative position uncertainty ellipsoid in the η B-plane direction (see Sect. 2.1.2) at Time of Closest Approach (TCA) (which corresponds to the relative velocity direction) [9]:

$$t_c = \frac{2\sigma_\eta}{\|\mathbf{v}_p - \mathbf{v}_s\|} \tag{2.2}$$

Thus, considering t_c and the primary orbital period, a short-term encounter occurs when:

$$\frac{t_c}{T_p} \ll 1 \tag{2.3}$$

2.1.2 B-plane reference frame

The use of a B-plane coordinate system is a convenient method for conjunction analysis. The relative position between the objects, before and after performing a CAM, is usually measured in terms of such r.f.

First, let $\{x, y, z\}$ be the axes of an inertial r.f. centred at $\mathrm{O_p} - \mathrm{O_s}$ impact point, called the encounter frame, such that:

$$\mathbf{u}_x = \frac{\mathbf{v}_p}{\|\mathbf{v}_p\|}, \quad \mathbf{u}_z = \frac{\mathbf{v}_p \times \mathbf{v}_s}{\|\mathbf{v}_p \times \mathbf{v}_s\|}, \quad \mathbf{u}_y = \mathbf{u}_z \times \mathbf{u}_x. \tag{2.4}$$

Considering the conjunction duration as a small interval of time $\Delta t \ll 1$ around the impact event, the motion of both objects can be assumed to be uniform rectilinear with good approximation as per the short-term encounter hypothesis. The B-plane is centred at the secondary object $\mathrm{O_s}$, according to the formulation of [10]. In conjunction, it lies perpendicular to the objects' relative velocity vector $\mathbf{v}_p - \mathbf{v}_s$, thus contains both of them at TCA.

The introduction of a new coordinate set $\mathbf{b}_{3D} = [\xi, \eta, \zeta]^\top$, in such r.f., is handy for all the subsequent mathematical derivations. The unitary vectors are defined

as follows:

$$\mathbf{u}_\xi = \frac{\mathbf{v}_p \times \mathbf{v}_s}{\|\mathbf{v}_p \times \mathbf{v}_s\|}, \quad \mathbf{u}_\eta = \frac{\mathbf{v}_p - \mathbf{v}_s}{\|\mathbf{v}_p - \mathbf{v}_s\|}, \quad \mathbf{u}_\zeta = \mathbf{u}_\xi \times \mathbf{u}_\eta. \tag{2.5}$$

The unitary vectors define the rotation matrix from the inertial r.f. to the B-plane:

$$\mathbf{R}_{3\text{D}} = [\mathbf{u}_\xi,\ \mathbf{u}_\eta,\ \mathbf{u}_\zeta]^\top \tag{2.6}$$

Moreover, the projection on the η-axis is achieved by:

$$\mathbf{R}_{2\text{D}} = [\mathbf{u}_\xi,\ \mathbf{u}_\zeta]^\top \tag{2.7}$$

2.1.3 Collision probability model

A probability constraint takes into account the presence of uncertainties in the knowledge of the objects' state. In a short-term scenario, where the relative motion follows a straight-line trajectory, the three-dimensional Probability of Collision (PoC) is confined to the volume of an infinite cylinder defined by the volume swept by a sphere of radius $R_A = R_P + R_S$ centered at the primary object. By performing integration along the entire extent of this volume (i.e., the direction of the relative velocity vector), PoC is described by a two-dimensional integral in the encounter B-plane. This problem was originally explored by Rice in [11]. As part of the overall solution to the collision probability problem, Chan later developed an absolutely convergent infinite series representation for the Rician Integral. It depends on parameters like the radius of the circular cross-section, the displacement of the circle from the Probability Density Function (PDF) center, and the standard deviation of the Rician PDF:

$$\text{PoC}(u, v) = e^{-\frac{v}{2}} \sum_{m=0}^{M} \frac{v^m}{2^m m!} \left[1 - e^{-\frac{u}{2}} \sum_{k=0}^{m} \frac{u^k}{2^k k!}\right] \tag{2.8}$$

where $m \in \mathbb{N}$ is the desired expansion order, $u \in \mathbb{R}^+$ is the ratio of the impact cross-sectional area to the area of the 1σ B-plane covariance ellipse:

$$u = \frac{s_A^2}{\sigma_\xi \sigma_\zeta \sqrt{1 - \rho_{\xi\zeta}^2}} \tag{2.9}$$

and $v \in \mathbb{R}^+$ is the Square Mahalanobis Distance (SMD):

$$\begin{aligned} v = \text{SMD} &= (\mathbf{r}_p - \mathbf{r}_s)^\top \mathbf{R}_{\text{b,2D}}^\top \mathbf{C}^{-1} \mathbf{R}_{\text{b,2D}} (\mathbf{r}_p - \mathbf{r}_s) \\ &= \mathbf{b}_p^\top \mathbf{C}^{-1} \mathbf{b}_p \end{aligned} \tag{2.10}$$

where $\mathbf{r}_p,\ \mathbf{r}_s$ are the primary and secondary position at conjunction, $\mathbf{b}_p$ the relative position of the primary object with respect to the secondary one in the B-plane r.f., and $\mathbf{C}$ the associated covariance matrix:

$$\mathbf{C}_{\xi\zeta} = \begin{bmatrix} \sigma_\xi^2 & \rho_{\xi\zeta}\sigma_\xi\sigma_\zeta \\ \rho_{\xi\zeta}\sigma_\xi\sigma_\zeta & \sigma_\zeta^2 \end{bmatrix} \tag{2.11}$$

In this work, the expression is truncated to $m = 3$, as done in [3].

2.2 Optimal control theory

The Optimal Control Problem (OCP) statement can be condensed into the following necessary conditions (for this problem):

$$\min_{\mathbf{u}(t)} \int_{t_0}^{t_f} \ell(\mathbf{x}(t), \mathbf{u}(t), t)\, dt \qquad \text{s.t.} \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \\ \mathbf{x}(t_0) = \mathbf{x}(t_0) \\ \Psi(\mathbf{x}(t_f), t_f) = 0 \end{cases} \tag{2.12}$$

Optimality consists in finding the unknown $\mathbf{u}^*(t)$ that produces the response $\mathbf{x}^*(t)$ when plugged into the right-hand side of the dynamic differential equations, such that the evaluation $\ell(\mathbf{x}^*(t), \mathbf{u}^*(t), t)$ yields the minimum cost.

It is convenient to introduce the Hamiltonian function associated with the OCP in Eq. (2.12), by adjoining the right-hand side of the differential equations to the cost integrand as:

$$\begin{aligned} H(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\lambda}(t), t) = &\ell(\mathbf{x}(t), \mathbf{u}(t), t) \\ &+ \boldsymbol{\lambda}^\top(t) \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \end{aligned} \tag{2.13}$$

with $\boldsymbol{\lambda}$ being the Lagrangian multiplier vector (also called costate vector).

The OCP necessary conditions translate to the so-called Euler-Lagrange equations, which can be formulated using the Hamiltonian:

$$\begin{cases} \dot{\mathbf{x}}(t) = \dfrac{\partial H}{\partial \boldsymbol{\lambda}} & \text{(2.14a)} \\ \dot{\boldsymbol{\lambda}}(t) = -\dfrac{\partial H}{\partial \mathbf{x}} & \text{(2.14b)} \\ 0 = \dfrac{\partial H}{\partial \mathbf{u}} & \text{(2.14c)} \end{cases}$$

with conditions:

$$\begin{cases} \Psi(\mathbf{x}(t_f), t_f) = 0 & \text{(2.15a)} \\ \boldsymbol{\lambda}(t_f) = \nu \dfrac{\partial \Psi}{\partial \mathbf{x}(t_f)} & \text{(2.15b)} \\ \mathbf{x}(t_0) = \mathbf{x}_0 & \text{(2.15c)} \end{cases}$$

The system given by Eqs. (2.14) and (2.15) includes a TPBVP, a Differential Algebraic Equation (DAE), and a terminal condition. If Eq. (2.14c) can be solved to produce the relation $\mathbf{u} = \mathbf{u}(\mathbf{x}, \boldsymbol{\lambda}, t)$, the system becomes a pure TPBVP. However, Euler-Lagrange equations only work with unbounded control. Pontryagin's Maximum Principle (PMP) [12] introduces the possibility of constraining the control to a set of admissible controls: $\mathbf{u} \in U$. This formulation leads to a new set of Euler-Lagrange equations. As a result, the third equation (2.14c) is replaced as follows:

$$0 = \frac{\partial H}{\partial \mathbf{u}} \implies \mathbf{u} = \arg\min_{\mathbf{u} \in U} H(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{u}, t) \tag{2.16}$$

2.3 Terminal conditions

The CAM problem is characterized by a terminal equality constraint imposed at TCA. Such a condition can be translated into a miss distance constraint or a probability constraint depending on whether the state-correlated uncertainties of the involved objects are negligible or not. The terminal conditions can be generally formulated in the following way:

$$\Psi(\mathbf{x}(t_f), t_f) = (\mathbf{r}_{pc} - \mathbf{r}_{sc})^\top \mathbf{K} (\mathbf{r}_{pc} - \mathbf{r}_{sc}) - \delta^2 = 0 \tag{2.17}$$

where $\mathbf{K}$ and δ depend on the type of constraint:

$$\text{MD:} \quad \mathbf{K} = \mathbf{I} \qquad \delta^2 = d^2 \tag{2.18}$$

$$\text{BP MD:} \quad \mathbf{K} = \mathbf{R}_{\text{b,2D}}^\top \mathbf{R}_{\text{b,2D}} \qquad \delta^2 = d_B^2 \tag{2.19}$$

$$\text{SMD:} \quad \mathbf{K} = \mathbf{R}_{\text{b,2D}}^\top \mathbf{C}^{-1} \mathbf{R}_{\text{b,2D}} \qquad \delta^2 = d_M^2 \tag{2.20}$$

The B-plane MD can also be tied to the isotropic PoC, which is obtained by retaining only the first term of the series ($m = 0$) in Eq. (2.8). The result is expressed as [13]:

$$\text{PoC}_{\text{iso}} = \exp\left\{ \ln\left(1 - e^{-\frac{R_A^2}{2\sigma^2}}\right) - \frac{1}{2}\frac{d_B^2}{\sigma^{*2}} \right\} \tag{2.21}$$

where $\sigma = \sqrt{\sigma_\xi \sigma_\zeta}$, and $\sigma^* = \max\{\sigma_\xi, \sigma_\zeta\}$. By solving for d_B in Eq. (2.21), the B-plane MD results as:

$$d_B = \sqrt{2\sigma^{*2} \ln\left(\frac{1 - e^{-\frac{R_A^2}{2\sigma^2}}}{\text{PoC}_{\text{iso}}}\right)} \tag{2.22}$$

Chan's formulation, in its direct form, lacks invertibility. However, according to [14] it is possible to derive the SMD for a given PoC. The Mahalanobis distance is a measure that represents how far a sample point is from the mean of the distribution in terms of standard deviations.

2.4 Analytical energy-optimal CAM design

The Energy Optimal (EO) CAM design, which is the central point of this work, is characterized by a low thrust control policy that aims at minimizing the overall energy. Based on [3], there exists a fully analytical solution to the EO CAM problem, which mainly hinges on motion linearization. This formulation adopts ECI coordinates and free thrust direction, where the control is defined by the acceleration vector $\mathbf{u} = [u_x, u_y, u_z]^\top$.

Adapting the theory introduced in Sect. 2.2, the cost function is defined as:

$$J = \int_{t_0}^{t_f} \frac{1}{2} \mathbf{u}^\top \mathbf{u} \, dt \tag{2.23}$$

and the terminal condition $\Psi(\mathbf{r}(t_f), t_f) = 0$ can be generally set according to Eq. (2.17).

Further, the augmented cost function shifts to:

$$\bar{J} = \nu \Psi(\mathbf{r}_f) + \int_{t_0}^{t_f} \left\{ \frac{1}{2} \mathbf{u}^\top \mathbf{u} + \boldsymbol{\lambda}^\top \left[\dot{\mathbf{x}} - \mathbf{f}(\mathbf{x}, \mathbf{u}) \right] \right\} dt \tag{2.24}$$

where time dependencies have been omitted for the sake of conciseness. The term $\nu \in \mathbb{R}$ is the additional Lagrangian multiplier paired with the the terminal constraint, $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_r, \boldsymbol{\lambda}_v]^\top \in \mathbb{R}^6$ alludes to the costate vector, and $\mathbf{f} \in \mathbb{R}^6$ embeds the problem dynamics $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ given by:

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = -\dfrac{\mu}{r^3}\mathbf{r} + \mathbf{u} \end{cases} \quad \text{ICs:} \begin{cases} \mathbf{r}(t_0) = \mathbf{r}_0 \\ \mathbf{v}(t_0) = \mathbf{v}_0 \end{cases} \tag{2.25}$$

The resulting Hamiltonian is:

$$H = \frac{1}{2}\mathbf{u}^\top \mathbf{u} + \boldsymbol{\lambda}_r^\top \mathbf{v} + \boldsymbol{\lambda}_v^\top \left(-\frac{\mu}{r^3}\mathbf{r} + \mathbf{u} \right) \tag{2.26}$$

The optimal control law results from the PMP as follows:

$$\mathbf{u} = -\boldsymbol{\lambda}_v \tag{2.27}$$

Finally, following the formulation of Eqs. (2.14) and Eqs. (2.15), the Euler-Lagrange equations read:

$$\begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = -\dfrac{\mu}{r^3}\mathbf{r} - \boldsymbol{\lambda}_v \\ \dot{\boldsymbol{\lambda}}_r = \dfrac{\mu}{r^3}\boldsymbol{\lambda}_v - \dfrac{3\mu \mathbf{r} \cdot \boldsymbol{\lambda}_v}{r^5}\mathbf{r} \\ \dot{\boldsymbol{\lambda}}_v = -\boldsymbol{\lambda}_r \end{cases} \tag{2.28}$$

$$\text{BCs:} \begin{cases} (\mathbf{r}_f - \mathbf{r}_s)^\top \mathbf{K} (\mathbf{r}_f - \mathbf{r}_s) - \delta^2 = 0 \\ \boldsymbol{\lambda}_r(t_f) = 2\nu \mathbf{K} (\mathbf{r}_f - \mathbf{r}_s) \\ \boldsymbol{\lambda}_v(t_f) = \mathbf{0} \\ \mathbf{r}(t_0) = \mathbf{r}_0 \\ \mathbf{v}(t_0) = \mathbf{v}_0 \end{cases}$$

Eq. (2.28) forms a TPBVP, which can be addressed analytically by linearizing the dynamics around the nominal uncontrolled Keplerian trajectory. The aim is to pinpoint the initial costates $\boldsymbol{\lambda}_r(t_0) = \boldsymbol{\lambda}_{r0}$, $\boldsymbol{\lambda}_v(t_0) = \boldsymbol{\lambda}_{v0}$ as to reduce the TPBVP to an Initial Value Problem (IVP). In [3], this is done by using the State Transition Matrix (STM) $\boldsymbol{\Phi}(t_0, t_f)$, which maps initial state variations into the corresponding final ones. The resulting equations are:

$$\begin{aligned} \boldsymbol{\lambda}_{r0} =& \mathbf{D}^{-1}[(\mathbb{I} - 2\nu \mathbf{D}\mathbf{B}^{-1}\mathbf{R}_{\text{2D}}^\top \mathbf{C}^{-1}\mathbf{R}_{\text{2D}})^{-1} \\ & (-2\nu \mathbf{D}\mathbf{B}^{-1}\mathbf{R}_{\text{2D}}^\top \mathbf{C}^{-1}\mathbf{R}_{\text{2D}}\mathbf{r}_s + \mathbf{r}_p) - \mathbf{r}_p] \\ \boldsymbol{\lambda}_{v0} =& -\boldsymbol{\Phi}_{44}^{-1}\boldsymbol{\Phi}_{43}\boldsymbol{\lambda}_{r0} \end{aligned} \tag{2.29}$$

with $\mathbf{B} = \boldsymbol{\Phi}_{33} - \boldsymbol{\Phi}_{34}\boldsymbol{\Phi}_{44}^{-1}\boldsymbol{\Phi}_{43}$, and $\mathbf{D} = \boldsymbol{\Phi}_{13} - \boldsymbol{\Phi}_{14}\boldsymbol{\Phi}_{44}^{-1}\boldsymbol{\Phi}_{43}$.

2.5 The Theory of Functional Connections: an overview

The TFC, introduced by D. Mortari [15], is a generalized interpolation scheme enabling functional interpola-

tion. This provides the mathematical framework to generate functionals (functions of functions) that analytically satisfy all imposed linear constraints and represent the real-valued set of functions satisfying such constraints. For a general introduction to the topic, including theoretical background and examples of practical applications, please consult [16].

One of the applications of the TFC framework is the numerical solution of Differential Equations (DEs). In general, the TFC method is planted between the two previous approaches (analytical and numerical), because it solves the equations numerically while the constraints are analytically embedded in the constrained functionals. The conventional numerical methods (excluding IVP) must explore the entire function space to identify a unique solution. Conversely, the TFC method solves the problem in the reverse order of the analytical approach. First, the candidate solution is approached by defining the constrained functional, which represents a reduction of the function space to a set of all the functions satisfying the Ordinary Differential Equation (ODE)'s constraints. Then, the codomain of the constrained functional is used to find the unique solution of the ODE. That is to say, if the free function $g(x)$ is assumed to cover the function space of the solution, then the constrained functional projects this function into a reduced set of constraints. Overall, the potential of the TFC framework to solve DEs lies in its ability to transform constrained problems into unconstrained ones by deriving constrained functionals that inherently satisfy the set of constraints. To further understand this concept, let us consider a general ODE subject to n linear constraints:

$$F\Big(x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, ..., \frac{d^ny}{dx^n}, \Big) = 0 \tag{2.30}$$

By deriving the constrained functionals, the constraints of Eq. (2.30) are decoupled from the solution of the ODE, which is transformed into:

$$\tilde{F}\Big(x, g, \frac{dg}{dx}, \frac{d^2g}{dx^2}, ..., \frac{d^ng}{dx^n}, \Big) = 0 \tag{2.31}$$

and the solution is obtained by finding the function $g(x)$ satisfying Eq. (2.31). The following sections elaborate on the major steps to solve this new equation.

2.5.1 Defining the free function

The free function is defined within the so-called basis domain, z. Since numerical bases are typically defined in closed domains, it is essential to establish the mapping between these basis domains and the ODE problem domain with some function $z = z(x)$. Selecting a free function fundamentally involves identifying the best (differentiable) function approximator. A linear combination of orthogonal polynomials can be leveraged for their advantageous numerical properties.

Since all functions are linear in their unknown coefficients a_k, the general expansion can be expressed as:

$$g(x) = \boldsymbol{\xi}^\top \mathbf{h}(z) \tag{2.32}$$

where $\boldsymbol{\xi} = \{a_0, ..., a_k, ..., a_{m-1}\}^\top$ and $\mathbf{h}(z)$ is the vector function of the m polynomials.

For the aforementioned orthogonal polynomials, the basis domain is $z \in [-1, 1]$, which may be different from the problem domain $x \in [x_0, x_f]$. Thus, both functions can be linearly mapped using the following relations:

$$z = z_0 + \frac{z_f - z_0}{x_f - x_0}(x - x_0) \iff x = x_0 + \frac{x_f - x_0}{z_f - z_0}(z - z_0) \tag{2.33}$$

The derivatives of $g(x)$ are determined as:

$$\frac{d^ng}{dx^n} = \boldsymbol{\xi}^\top \frac{d^n\mathbf{h}(z)}{dz^n}\left(\frac{dz}{dx}\right)^n \tag{2.34}$$

where, by using:

$$c := \frac{dz}{dx} = \frac{z_f - z_0}{x_f - x_0} \tag{2.35}$$

the expression of Eq. (2.34) can be simplified to:

$$\frac{d^ng}{dx^n} = c^n \boldsymbol{\xi}^\top \frac{d^n\mathbf{h}(z)}{dz^n} \tag{2.36}$$

By defining the free function and its derivatives according to Eqs. (2.32) and (2.34), respectively, the transformed ODE derived in Eq. (2.31) reduces to:

$$\tilde{F}(x, \boldsymbol{\xi}) = 0 \tag{2.37}$$

2.5.2 Domain discretization and loss vector

To solve for the unknown coefficients $\boldsymbol{\xi}$ numerically, the problem domain (and thus the basis domain) must be discretized. By doing so, Eq. (2.37) becomes a system of algebraic equations, which can be linear or nonlinear, depending on the original nature of Eq. (2.30). The new problem can be described with the so-called loss vector at the discretized points:

$$\mathbb{L}(\boldsymbol{\xi}) = \Big[\tilde{F}(x_0, \boldsymbol{\xi}) \dots \tilde{F}(x_k, \boldsymbol{\xi}) \dots \tilde{F}(x_f, \boldsymbol{\xi})\Big]^\top = \mathbf{0} \tag{2.38}$$

When using Chebyshev or Legendre orthogonal polynomials, the discretization scheme involves more than merely selecting uniformly spaced points. The optimal approach in these cases is to use Chebyshev-Gauss-Lobatto nodes (also known as Chebyshev nodes of the second kind). This distribution results in a much slower increase on the condition number of the matrix to be inverted in the Least Squares (LS) problem as the number of basis functions, m, increases. For $N + 1$ domain points, the nodes are calculated as:

$$z_j = -\cos\left(\frac{j\pi}{N}\right), \quad for \;\; j = 0, 1, 2, ..., N \tag{2.39}$$

2.5.3 Least squares solution

In the case of a linear ODE, the constrained functional and its derivatives appear linearly and consequently remain linear in the unknown $\boldsymbol{\xi}$ term. This leads to the following equation:

$$\mathbb{A}\boldsymbol{\xi} = \mathbf{b} \tag{2.40}$$

where matrix $\mathbb{A}$ is composed of a linear combination of the terms linear in the unknown coefficients. In terms of the loss function $\tilde{F}$, $\mathbb{A}$ is simply the Jacobian of the loss vector:

$$\mathbb{J} = \frac{\partial \mathbb{L}}{\partial \boldsymbol{\xi}} = \left[\frac{\partial \tilde{F}(x_0, \boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \cdots \frac{\partial \tilde{F}(x_k, \boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \cdots \frac{\partial \tilde{F}(x_f, \boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right]^\top \tag{2.41}$$

which is independent of $\boldsymbol{\xi}$, since $\mathbb{L}$ is linear in $\boldsymbol{\xi}$. On the other hand, vector $\mathbf{b}$ is simply the loss vector evaluated at $\boldsymbol{\xi} = \mathbf{0}$:

$$\mathbf{b} = -\mathbb{L}(\mathbf{0}) = \left[\tilde{F}(x_0, \mathbf{0}) \dots \tilde{F}(x_k, \mathbf{0}) \dots \tilde{F}(x_f, \mathbf{0})\right]^\top \tag{2.42}$$

Finally, Eq. (2.40) becomes:

$$\mathbb{J}\boldsymbol{\xi} = -\mathbb{L}(\mathbf{0}) \tag{2.43}$$

The solution to the linear problem posed by Eq. (2.43) can be assessed directly using any LS method. However, in the case of a nonlinear ODE, Eq. (2.38) is nonlinear in $\boldsymbol{\xi}$. A system of nonlinear algebraic equations must be solved by an iterative LS method such as the Gauss-Newton algorithm, where each iteration step is formulated similarly to Eq. (2.43) as follows:

$$\mathbb{J}(\boldsymbol{\xi_i})\Delta\boldsymbol{\xi}_i = -\mathbb{L}(\boldsymbol{\xi_i}) \tag{2.44}$$

and the update of $\boldsymbol{\xi}$ is given by:

$$\boldsymbol{\xi}_{i+1} = \boldsymbol{\xi}_i + \Delta\boldsymbol{\xi}_i \tag{2.45}$$

The LS step $\Delta\boldsymbol{\xi}_i$ can be computed from Eq. (2.44) using any linear LS method. Additionally, it is necessary to define the converge criterion for the iterative process. This work considers the following stopping condition:

$$max[|\mathbb{L}(\boldsymbol{\xi})|] < \epsilon \tag{2.46}$$

Overall, the distinction between solving a linear versus a nonlinear ODE is simplified to the difference between Eq. (2.43) and Eq. (2.44). The linear case requires only one iteration, compared to the multiple iterations of the nonlinear case.

3. The TFC applied to the energy-optimal CAM

This section demonstrates the application of the TFC to solve the nonlinear EO CAM equations, illustrating the practical implementation of the fundamental principles explored in the previous chapter.

3.1 Problem formulation with TFC

The nonlinear EO-CAM problem is stated by the set of Euler-Lagrange equations and boundary conditions of Eq. (2.28). Through the use of TFC, the linear constraints can be embedded into the constrained functionals, thus the first-order necessary conditions reduce to the following loss functions, for $i = 1, 2, 3$:

$$\begin{cases} F_{r_i} \longrightarrow \dot{r}_i - v_i = 0 \\ F_{v_i} \longrightarrow \dot{v}_i + \dfrac{\mu}{r^3} r_i + \lambda_{v_i} = 0 \\ F_{\lambda_{r_i}} \longrightarrow \dot{\lambda}_{r_i} - \dfrac{\mu}{r^3}\lambda_{v_i} + \dfrac{3\mu \mathbf{r}\cdot\boldsymbol{\lambda}_v}{r^5} r_i = 0 \\ F_{\lambda_{v_i}} \longrightarrow \dot{\lambda}_{v_i} + \lambda_{r_i} = 0 \\ F_{NL} \longrightarrow (\mathbf{r}_f - \mathbf{r}_s)^\top \mathbf{K} (\mathbf{r}_f - \mathbf{r}_s) - \delta^2 = 0 \end{cases} \tag{3.1}$$

whith $r = \sqrt{|r_1|^2 + |r_2|^2 + |r_3|^2}$. It is important to note that the last equation corresponds to the first boundary condition of the original system. This constraint is treated as an additional loss function due to its nonlinear nature. Consequently, the constrained functionals are derived from the remaining linear constraints, arranged such that the constants are collected on one side:

$$\text{Linear BCs:} \begin{cases} r_i(t_0) = r_{0_i} \\ v_i(t_0) = v_{0_i} \\ 2\nu \mathbf{K}_i \mathbf{r}_s = 2\nu \mathbf{K}_i \mathbf{r}(t_f) - \lambda_{r_i}(t_f) \\ \lambda_{v_i}(t_f) = 0 \end{cases} \tag{3.2}$$

where $\mathbf{K}_i$ is the i-th row of the $\mathbf{K}$ matrix. It is crucial to acknowledge that the third relation is a component constraint, i.e., a constraint involving several dependent variables, in this case: r_1, r_2, r_3 and λ_{r_i}. The way to proceed is to decide which constrained functional will embed the component constraint. Theoretically, a valid constrained functional will be produced regardless of which dependent variable is chosen. Here, let us assign the component constraint to λ_r.

Since each variable has only one associated constraint, the general expression of the constrained functionals results as follows:

$$\chi_i(t) = g_{\chi_i}(t) + \phi_\chi(t)\rho_{\chi_i} \tag{3.3}$$

for $\chi = r, v, \lambda_r, \lambda_v$.

The projection functions are effortlessly derived as:

$$\begin{aligned} &\rho_{r_i}(t, g_i^r(t)) = r_{0_i} - g_{r_i}(t_0) \\ &\rho_{v_i}(t, g_i^v(t)) = v_{0_i} - g_{v_i}(t_0) \\ &\rho_{\lambda_{r_i}}(t, g_{\lambda_{r_i}}(t), g_{r_1}(t), g_{r_2}(t), g_{r_3}(t)) \\ &\quad = 2\nu \mathbf{K}_i \big(\mathbf{r}_s - \mathbf{r}(t_f)\big) + g_{\lambda_{r_i}}(t_f) \\ &\rho_{\lambda_{v_i}}(t, g_{\lambda_{v_i}}(t)) = -g_{\lambda_{v_i}}(t_f) \end{aligned} \tag{3.4}$$

The switching functions result from applying the definition:

$$\phi_\chi(t) = s(t)\alpha_\chi \tag{3.5}$$

selecting the support function $s(t) = 1$ for all variables.

Leveraging the scalar multiplication property of the constraint operator (see Chapter 3 of [16]), the expression $\mathfrak{C}_\chi[\phi_\chi(t)] = 1$ can be rewritten as:

$$\alpha_\chi \mathfrak{C}_\chi[s(t)] = 1 \tag{3.6}$$

and the unknown coefficients are:

$$\alpha_r = 1 \quad \alpha_v = 1 \quad \alpha_{\lambda_r} = -1 \quad \alpha_{\lambda_v} = 1 \tag{3.7}$$

Finally, the constrained functionals are given by:

$$\begin{aligned} r_i(t) &= g_{r_i}(t) + \big(r_{0_i} - g_{r_i}(t_0)\big) \\ v_i(t) &= g_{v_i}(t) + \big(v_{0_i} - g_{v_i}(t_0)\big) \\ \lambda_{r_i}(t) &= g_{\lambda_{r_i}}(t) - g_{\lambda_{r_i}}(t_f) + 2\nu \mathbf{K}_i \big(\mathbf{r}(t_f) - \mathbf{r}_s\big) \\ \lambda_{v_i}(t) &= g_{\lambda_{v_i}}(t) - g_{\lambda_{v_i}}(t_f) \end{aligned} \tag{3.8}$$

According to the TFC mathematical framework, since all the expressions in Eq. (3.8) have been derived using $s(t) = 1$, the vector of polynomials dispenses with the first term (degree 0) and thus is the same in all cases: $\mathbf{h}_\chi(z) = \mathbf{h}(z)$. Decomposing the free function and mapping the time-independent variable into the basis domain, the constrained functionals and their first derivatives result as:

Position

$$\begin{aligned} r_i(z) &= \big[\mathbf{h}(z) - \mathbf{h}(z_0)\big]^\top \boldsymbol{\xi}_{r_i} + r_{0_i} \\ \dot{r}_i(z) &= \big[c\mathbf{h}_z(z)\big]^\top \boldsymbol{\xi}_{r_i} \end{aligned} \tag{3.9}$$

Velocity

$$\begin{aligned} v_i(z) &= \big[\mathbf{h}(z) - \mathbf{h}(z_0)\big]^\top \boldsymbol{\xi}_{v_i} + v_{0_i} \\ \dot{v}_i(z) &= \big[c\mathbf{h}_z(z)\big]^\top \boldsymbol{\xi}_{v_i} \end{aligned} \tag{3.10}$$

Position costate

$$\begin{aligned} \lambda_{r_i}(z) =& \big[\mathbf{h}(z) - \mathbf{h}(z_f)\big]^\top \boldsymbol{\xi}_{\lambda_{r_i}} - 2\nu \mathbf{K}_i \mathbf{r}_s \\ &+ 2\nu \Big\{ K_{i1} \Big(\big[\mathbf{h}(z_f) - \mathbf{h}(z_0)\big]^\top \boldsymbol{\xi}_{r_1} + r_{0_1} \Big) \\ &\quad + K_{i2} \Big(\big[\mathbf{h}(z_f) - \mathbf{h}(z_0)\big]^\top \boldsymbol{\xi}_{r_2} + r_{0_2} \Big) \\ &\quad + K_{i3} \Big(\big[\mathbf{h}(z_f) - \mathbf{h}(z_0)\big]^\top \boldsymbol{\xi}_{r_3} + r_{0_3} \Big) \Big\} \\ \dot{\lambda}_{r_i}(z) =& \big[c\mathbf{h}_z(z)\big]^\top \boldsymbol{\xi}_{\lambda_{r_i}} \end{aligned} \tag{3.11}$$

Velocity costate

$$\begin{aligned} \lambda_{v_i}(z) &= \big[\mathbf{h}(z) - \mathbf{h}(z_f)\big]^\top \boldsymbol{\xi}_{\lambda_{v_i}} \\ \dot{\lambda}_{v_i}(z) &= \big[c\mathbf{h}_z(z)\big]^\top \boldsymbol{\xi}_{\lambda_{v_i}} \end{aligned} \tag{3.12}$$

The TFC problem is defined by the system of Eq. (3.1) and the constrained functionals of Eqs. (3.9), (3.10), (3.11) and (3.12). It is evident that the problem is nonlinear since the unknowns appear nonlinearly in the second, third and last equations of Eq. (3.1). Therefore, the solution requires the application of the Non-Linear Least Squares (NLLS) optimizer.

Upon analyzing the equations, it becomes clear that the unknowns in this system include not only the coefficients of the polynomial expressions, $\boldsymbol{\xi}_{\chi_i}$, but also the Lagrange multiplier, ν. Depending on the method used to handle the latter, there are two main approaches to address this problem: the Single-Loop Algorithm (SLA) and the Double-Loop Algorithm (DLA). The first method only involves the LS iterative loop to solve for all the unknowns at the same time, that is, including ν, thus increasing the search space. The second method features an outer iterative loop that solves for ν by optimizing the residual of the nonlinear loss function F_{NL}. At each step, the candidate multiplier is inserted into the inner LS loop to solve for the polynomial coefficients.

Particular attention should be given to problems involving state vectors in dynamical systems, where some components are derivatives of other components [17]. When formulating the problem elements using TFC, it would be a mistake trying to merge the position $\mathbf{r}(t)$ and velocity $\mathbf{v}(t)$ constrained functionals given in Eq. (3.8) and assume that the first equation of the system in Eq. (3.1) is redundant because the velocity will always be the derivative of the position as per the following single constrained functional:

$$r_i(t) = g_{r_i}(t) + \big(r_{0_i} - g_{r_i}(t_0)\big) + (t - t_0)\big(v_{0_i} - \dot{g}_{r_i}(t_0)\big) \tag{3.13}$$

While this expression satisfies both initial constraints for $\mathbf{r}(t)$ and $\mathbf{v}(t)$, it does not necessarily satisfy the dynamical equations obtained from the Hamiltonian formulation given in Eq. (3.1). The same applies to the costates and the relation given by the fourth equation of the system.

3.2 Single-loop approach

The SLA involves ν and F_{NL} in the LS optimization process, treating the former as an unknown and the latter as an additional loss function. In this way, the set of loss functions is given by Eq. (3.1) and the vector of unknowns:

$$\begin{aligned} \Xi = \Big[&\boldsymbol{\xi}_{r_1}^\top,\ \boldsymbol{\xi}_{r_2}^\top,\ \boldsymbol{\xi}_{r_3}^\top,\ \boldsymbol{\xi}_{v_1}^\top,\ \boldsymbol{\xi}_{v_2}^\top,\ \boldsymbol{\xi}_{v_3}^\top, \\ &\boldsymbol{\xi}_{\lambda_{r_1}}^\top,\ \boldsymbol{\xi}_{\lambda_{r_2}}^\top,\ \boldsymbol{\xi}_{\lambda_{r_3}}^\top,\ \boldsymbol{\xi}_{\lambda_{v_1}}^\top,\ \boldsymbol{\xi}_{\lambda_{v_2}}^\top,\ \boldsymbol{\xi}_{\lambda_{v_3}}^\top,\ \nu \Big]^\top \end{aligned} \tag{3.14}$$

of size $(12m + 1) \times 1$, with m being the number of polynomials in $\mathbf{h}(z)$. Finally, the loss vector contains F_{NL}, which is defined at t_f, and the rest of the loss functions

are evaluated at the discretized domain points, as follows:

$$\mathbb{L} = \left[\mathbf{F}(t_0,\Xi)^\top \dots \mathbf{F}(t_f,\Xi)^\top, F_{NL}(\Xi)\right]^\top \tag{3.15}$$

of size $(12N+1)\times 1$, where N is the number of domain points and:

$$\mathbf{F}(t,\Xi) = \Big[F_{r_1},\ F_{r_2},\ F_{r_3},\ F_{v_1},\ F_{v_2},\ F_{v_3},\ F_{\lambda_{r_1}},\ F_{\lambda_{r_2}},\ F_{\lambda_{r_3}},\ F_{\lambda_{v_1}},\ F_{\lambda_{v_2}},\ F_{\lambda_{v_3}}\Big]^\top\Big|_{(t,\Xi)} \tag{3.16}$$

Finally, the loss Jacobian derived from Eq. (3.15) is a $(12N+1)\times(12m+1)$ matrix defined as follows:

$$\mathbb{J} = \frac{\partial \mathbb{L}}{\partial \boldsymbol{\xi}} = \left[\frac{\partial \mathbf{F}(t_0,\Xi)}{\partial \boldsymbol{\xi}} \cdots \frac{\partial \mathbf{F}(t_f,\Xi)}{\partial \boldsymbol{\xi}}, \frac{\partial F_{NL}}{\partial \boldsymbol{\xi}}\right]^\top \tag{3.17}$$

Once the loss vector and loss Jacobian are defined, the solution can be addressed by applying NLLS in two main different ways: by explicit implementation of the Gauss-Newton algorithm defined in Eq. (2.44), and by means of the *lsqnonlin*() MATLAB function:

- The Gauss-Newton iterative loop requires inverting the loss Jacobian, which is typically not a square matrix. This is achieved by considering the following options:
  1. *pinv*(): Moore-Penrose pseudo-inverse with Singular Value Decomposition (SVD) decomposition.
  2. *lsqminnorm*(): which not only provides a solution that minimizes the value of $\|\mathbb{J}(\boldsymbol{\xi_i})\Delta\boldsymbol{\xi}_i + \mathbb{L}(\boldsymbol{\xi_i})\|$ but also $\|\Delta\boldsymbol{\xi}_i\|$.
  3. *equilibrate*(): to improve the condition number of the problem by scaling $\mathbf{A} = (\mathbb{J}(\boldsymbol{\xi}_i)^\top \mathbb{J}(\boldsymbol{\xi}_i))$.
- The *lsqnonlin*() MATLAB function is a NLLS solver that finds the minimum of the sum of squares of the residuals provided by the loss vector. This function only requires the loss vector to compute the solution.

Since the system is nonlinear, one common step to all the iterative methods described above is the initialization of the unknowns by means of a first guess $\boldsymbol{\xi}^0$, such that both the position and velocity are estimated to match the state evolution of the ballistic trajectory, while the costates and the Lagrangian multiplier are set equal to zero. The position and velocity unknown coefficients, $\boldsymbol{\xi}_r^0$ and $\boldsymbol{\xi}_v^0$ respectively, are defined through linear least squares fitting using the unperturbed propagated trajectory along the domain points $(\mathbf{r}_{ref}(z), \mathbf{v}_{ref}(z))$, except for the initial conditions, at z_0, which are analytically satisfied by the constrained functionals. From Eqs. (3.9) and (3.10):

$$\begin{bmatrix} [\mathbf{h}(z_1)-\mathbf{h}(z_0)]^\top \\ \vdots \\ [\mathbf{h}(z_k)-\mathbf{h}(z_0)]^\top \\ \vdots \\ [\mathbf{h}(z_f)-\mathbf{h}(z_0)]^\top \end{bmatrix} \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_k \\ \vdots \\ a_m \end{bmatrix}}_{\boldsymbol{\xi}_{r_i}^0} = \begin{bmatrix} r_{ref_i}(z_1) - r_{0_i} \\ \vdots \\ r_{ref_i}(z_k) - r_{0_i} \\ \vdots \\ r_{ref_i}(z_f) - r_{0_i} \end{bmatrix}$$

$$\begin{bmatrix} [\mathbf{h}(z_1)-\mathbf{h}(z_0)]^\top \\ \vdots \\ [\mathbf{h}(z_k)-\mathbf{h}(z_0)]^\top \\ \vdots \\ [\mathbf{h}(z_f)-\mathbf{h}(z_0)]^\top \end{bmatrix} \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_k \\ \vdots \\ a_m \end{bmatrix}}_{\boldsymbol{\xi}_{v_i}^0} = \begin{bmatrix} v_{ref_i}(z_1) - v_{0_i} \\ \vdots \\ v_{ref_i}(z_k) - v_{0_i} \\ \vdots \\ v_{ref_i}(z_f) - v_{0_i} \end{bmatrix} \tag{3.18}$$

The linear systems of Eq. (3.18) can be solved in one step using any of the methods described earlier, for example, the *lsqminnorm*() function. The solutions for $i = 1,2,3$ lead to the first estimation of the vector of unknowns:

$$\boldsymbol{\xi}^0 = \left[\boldsymbol{\xi}_{r_1}^{0^\top},\ \boldsymbol{\xi}_{r_2}^{0^\top},\ \boldsymbol{\xi}_{r_3}^{0^\top},\ \boldsymbol{\xi}_{v_1}^{0^\top},\ \boldsymbol{\xi}_{v_2}^{0^\top},\ \boldsymbol{\xi}_{v_3}^{0^\top},\ \mathbf{0}_{(6m+1)\times 1}^\top\right]^\top \tag{3.19}$$

Lastly, units are adjusted as an additional effort to reduce the condition number of the LS problem. Motivated by the appearance of Δt as the denominator of $c = \Delta z/\Delta t$ in some components of the loss Jacobian, time units are converted from seconds to hours. After testing the problem conditioning improvement for several unit conversions, including adimensionalization, this proves to be one of the most optimal approaches to effectively reduce the condition number of the matrix and thus stabilize the solution convergence.

3.3 Double-loop approach

The DLA splits the optimization process into two different steps. Unlike in the SLA, here ν and F_{NL} are separated from the LS optimization process in an attempt to reduce the complexity of such iterative loop in terms of search space and residuals. In this case, F_{NL} serves as the residual of the $fsolve$() MATLAB solver, which also requires a first guess of the Lagrangian multiplier, ν^0. At the k-th iteration, the inner LS iterative loop is solved using the corresponding estimation ν^k. However, this time the TFC problem is defined by the set of loss functions given in Eq. (3.1) except for F_{NL}, and the following vec-

tor of unknowns:

$$\Xi = \Big[\boldsymbol{\xi}_{r_1}^\top,\, \boldsymbol{\xi}_{r_2}^\top,\, \boldsymbol{\xi}_{r_3}^\top,\, \boldsymbol{\xi}_{v_1}^\top,\, \boldsymbol{\xi}_{v_2}^\top,\, \boldsymbol{\xi}_{v_3}^\top, \\ \boldsymbol{\xi}_{\lambda_{r_1}}^\top,\, \boldsymbol{\xi}_{\lambda_{r_2}}^\top,\, \boldsymbol{\xi}_{\lambda_{r_3}}^\top,\, \boldsymbol{\xi}_{\lambda_{v_1}}^\top,\, \boldsymbol{\xi}_{\lambda_{v_2}}^\top,\, \boldsymbol{\xi}_{\lambda_{v_3}}^\top\Big]^\top \tag{3.20}$$

of size $12m \times 1$. The loss vector is:

$$\mathbb{L} = \Big[\mathbf{F}(t_0, \Xi)^\top \dots \mathbf{F}(t_k, \Xi)^\top \dots \mathbf{F}(t_f, \Xi)^\top\Big]^\top \tag{3.21}$$

of size $12N \times 1$, where $\mathbf{F}(t, \Xi)$ is given in Eq. (3.16).

In this case, the loss Jacobian is a $12N \times 12m$ matrix defined as follows:

$$\mathbb{J} = \frac{\partial \mathbb{L}}{\partial \boldsymbol{\xi}} = \left[\frac{\partial \mathbf{F}(t_0, \Xi)}{\partial \boldsymbol{\xi}} \cdots \frac{\partial \mathbf{F}(t_k, \Xi)}{\partial \boldsymbol{\xi}} \cdots \frac{\partial \mathbf{F}(t_f, \Xi)}{\partial \boldsymbol{\xi}}\right]^\top \tag{3.22}$$

The inner loop of DLA can be solved by applying any of the NLLS methods described for SLA in Sect. 3.2.

4. Results

The following tests have been implemented in MATLAB R2023b and run on OMEN Laptop 15-en1xxx laptop (CPU: AMD Ryzen 7 5800H of 3201 MHz, GPU: NVIDIA GeForce RTX 3060). The conjunction scenario studied in this chapter, called LEOH2HMD, is a real encounter developed for the Electrocam project sponsored by the European Space Agency (ESA) and provided by GMV, where both primary and secondary objects orbit the Low Earth Orbit (LEO) region. This scenario is characterized in Tables 4-1 and 4-2.

Table 4-1. LEOH2HMD conjunction characterization.

Conjunction Data Message	
MD	48.1 m
PoC	0.107
TCA	2020-02-03 T02:24:00.003140
Pos. uncert.	$\sim 10 - 10^2$ m
Vel. uncert.	~ 0.1 m/s

Table 4-2. LEOH2HMD orbital parameters of the primary and secondary objects.

Element	**Primary**	**Secondary**
a [m]	7.552e3	7.575e3
e [-]	0.0012	0.0100
i [deg]	87.92	89.38
Ω [deg]	1.94	179.03
ω [deg]	127.52	112.63
θ [deg]	5.11	294.68

Table 4-3 lists the configuration parameters employed in each test (identified by the section heading numbering). The notation *var* indicates the parameters evaluated (varied) during the sensitivity analyses. In summary: $\Delta\theta$ is the true anomaly, N is the number of domain points, m is the number of basis functions, and ϵ is the convergence tolerance used in NLLS. Moreover, "poly" denotes the type of orthogonal polynomials employed.

Table 4-3. Testing configuration.

	4.1.1	**4.1.2**	**4.2.1**	**4.2.2**	**4.2.3**	**4.3**	**4.4**
$\boldsymbol{\Delta\theta}$	0.1 : 0.05 : 2		0.2 : 0.2 : 8				
$\mathbf{N}$	[30,90]		[30,200]				
$\mathbf{m}$	[20,60]		[20,150]				
$\boldsymbol{\epsilon}$	1e−9				*var*	1e−5	
poly	Chebyshev		*var*	Chebyshev			

4.1 TFC: strategy comparison

This section aims to identify the most computationally efficient TFC approach. The primary strategies for addressing the TFC solution are the SLA and the DLA. Both require applying NLLS: to the single loop in the former and to the inner loop in the latter. As outlined in Sect. 3.2, the main candidates to tackle this optimization are the Gauss-Newton method and the *lsqnonlin*() MATLAB function, with the former supporting multiple solvers. The first step is to identify the best choice to maximize convergence for the Gauss-Newton method. Next, the best NLLS solver is determined by testing both Gauss-Newton and *lsqnonlin*() in SLA. The selected method is then implemented in the inner loop of DLA for the final comparison between SLA and DLA. Four tests are carried out, one per terminal condition. The results illustrate the computational cost expressed as a function of the true anomaly.

Both N and m are defined as linear functions of θ to adapt the problem's complexity to the true anomaly. For scenarios with minimal maneuver anticipation, fewer resources are sufficient. Although increasing these values generally leads to more robust solutions, it also significantly increases processing time. Conversely, as maneuver anticipation increases, the problem domain expands, requiring larger values of N to resize the domain discretization and larger values of m to enhance the solution accuracy. Both the lower and upper values are selected to ensure system convergence for the minimum and maximum true anomalies while keeping the values as low as possible. However, in some instances, the algorithm may fail to converge to a solution. If this occurs, it is assumed that the values of N and m are insufficient, and the op-

timization process is restarted with the maximum values. If convergence is still not achieved, it is concluded that convergence is unattainable, regardless of the TFC configuration.

4.1.1 Gauss-Newton solver

The Gauss-Newton solver for the NLLS problem adopts three possible implementations in this study: *pinv* (pseudoinverse with SVD), *lsqminnorm* (pseudoinverse with Complete Orthogonal Decomposition (COD)), and *equilibrate* (scaling to reduce the condition number). The results are obtained using SLA. Fig. 4-1 includes four diagrams, each corresponding to a different terminal condition. On average, the *lsqminnorm*() function demonstrates the best computational performance, especially as the complexity of the problem increases. This trend highlights its superior robustness in handling more demanding cases. Table 4-4 summarizes their success rates in terms of achieving convergence. Both *pinv*() and *lsqminnorm*() present the best ratio with a 100 % of success, followed by *scale*() with the 98.7 %.

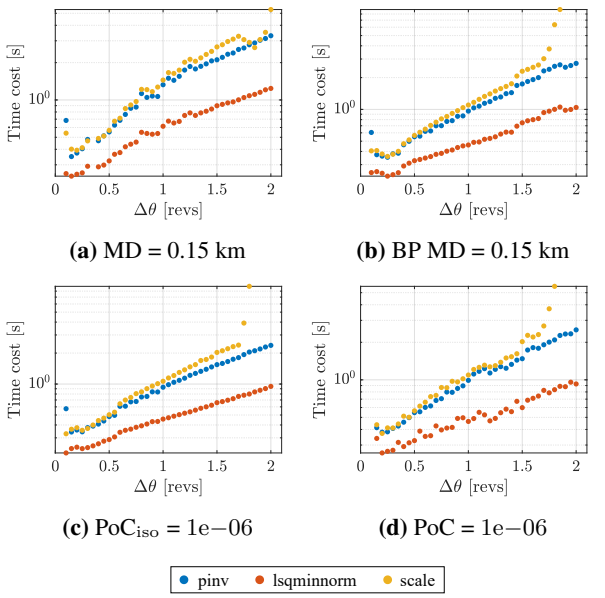


Fig. 4-1. Computational cost of the Gauss-Newton solvers for each terminal condition.

4.1.2 NLLS solver: Gauss-Newton vs lsqnonlin

The previous test revealed the *lsqminnorm*() function to be the most efficient choice for integrating the Gauss-Newton solver. In this test, the performance of the candidate NLLS solvers is compared to determine the best option. Fig. 4-2 proves that Gauss-Newton is the fastest NLLS solver in SLA compared to *lsqnonlin*(). As a result, Gauss-Newton is also used as the solver for the inner loop of DLA. Finally, when comparing both algorithms, SLA consistently demonstrates superior computational performance, especially as problem complexity increases. The reason why DLA does not improve the performance of SLA despite simplifying the equations by moving ν and F_{NL} to the outer loop is that the inner system remains nonlinear in terms of the unknown coefficients $\boldsymbol{\xi}$. The additional computational effort introduced by the *fsolve*() function is not even countervailed. Table 4-5 summarizes their success rates in terms of achieving convergence. The DLA presents the best ratio with 100 % of success, closely followed by SLA with 98.7 %.

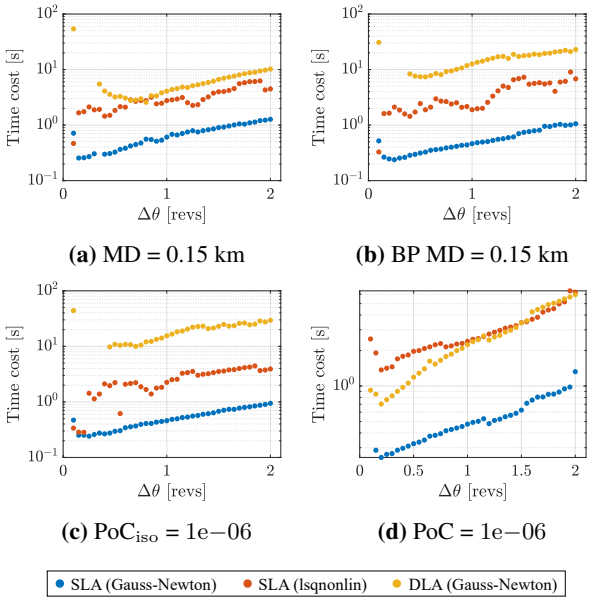


Fig. 4-2. Computational cost of the cadidate algorithms for each terminal condition.

Table 4-4. Convergence success rate [%] of the Gauss-Newton solvers.

	MD	BP MD	PoC$_{iso}$	PoC	Total
pinv	97.4	100	100	97.4	98.7
lsqminnorm	97.4	100	100	97.4	98.7
scale	97.4	92.3	89.7	87.1	91.6

Table 4-5. Convergence success rate [%] of the candidate algorithms.

	MD	BP MD	PoC$_{iso}$	PoC	Total
SLA (G-N)	97.4	100	100	97.4	98.7
DLA (G-N)	100	100	100	100	100
SLA (lsqnonlin)	87.1	86.8	82.0	100	88.9

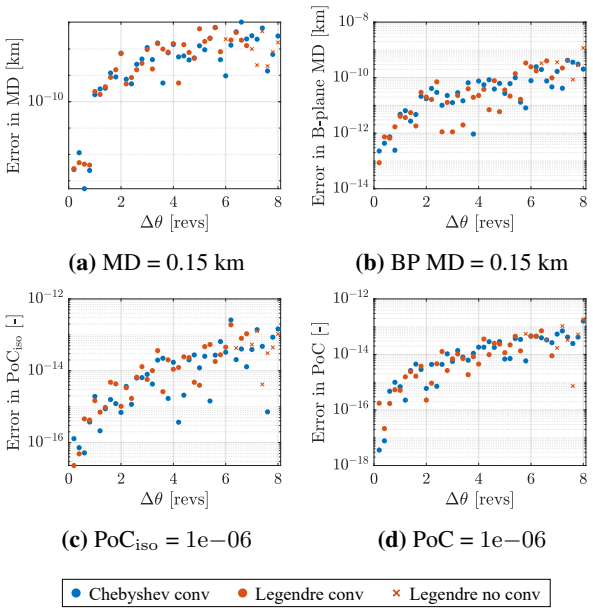
4.2 TFC: sensitivity analyses

In the previous section, various algorithms are evaluated based solely on their computational performance. The SLA with the Gauss-Newton solver, utilizing the *lsqminnorm*() function, is identified as the most efficient TFC approach, due to its superior time efficiency and a convergence success rate that is nearly 100%. This section offers a detailed examination of the results through a series of sensitivity analyses.

4.2.1 Orthogonal polynomials

This test evaluates the absolute error of solutions for the same conjunction using both Chebyshev and Legendre polynomials across a range of true anomalies. Additionally, it records the number of iterations needed for each method to converge.

Figure 4-3 illustrates the error associated with each terminal condition as a function of maneuver anticipation. This metric is calculated by determining the absolute difference between the imposed constraint and the actual value obtained. It is important to note that the real value is not derived directly from the TFC solution. Although the solution convergence ensures that the residual of the condition, namely F_{NL}, is below the known tolerance $epsilon$, the real value is instead obtained by propagating the initial conditions provided by TFC to the TCA. This allows to analyse how effectively the initial conditions achieve the desired outcomes when integrating the dynamics of Eq. (3.1). The error increases by several orders of magnitude with the true anomaly, although it is still acceptable if compared to the uncertainty levels.

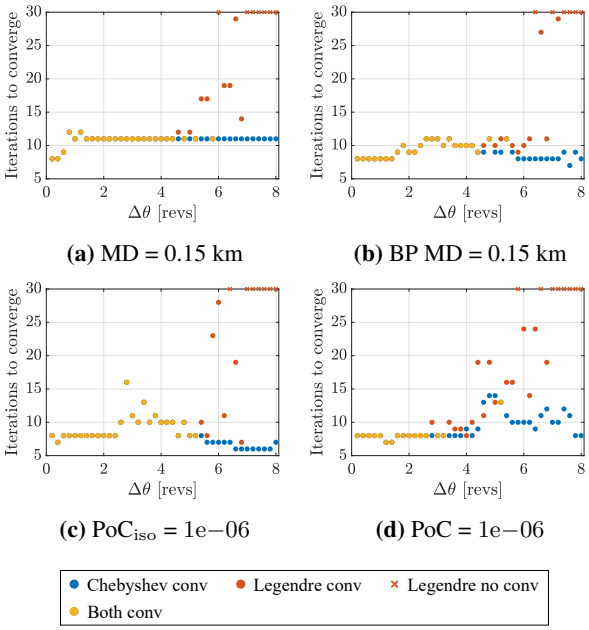


(a) MD = 0.15 km **(b)** BP MD = 0.15 km

(c) PoC$_{iso}$ = 1e−06 **(d)** PoC = 1e−06

Fig. 4-3. Absolute error in the terminal condition using Chebyshev and Legendre polynomials.

Figure 4-4 illustrates the number of iterations required for convergence in each scenario. For low values of true anomaly, both Chebyshev and Legendre polynomials converge in the same number of steps (denoted in yellow to enhance visibility where results overlap). However, as the true anomaly grows, the convergence performance of Legendre polynomials deteriorates, often reaching the maximum iteration limit and thus failing to converge.



(a) MD = 0.15 km **(b)** BP MD = 0.15 km

(c) PoC$_{iso}$ = 1e−06 **(d)** PoC = 1e−06

Fig. 4-4. Iterations to converge using Chebyshev and Legendre polynomials.

Tables 4-6 and 4-7 present a summary of the results, focusing on convergence success rates and Mean Absolute Errors (MAEs). While Chebyshev polynomials show slightly better MAE results, their performance is significantly superior in terms of convergence. As the true anomaly increases, Legendre polynomials experience instances of non-convergence. However, the error values for both types of polynomials remain similar even in those cases. Based on this analysis, Chebyshev polynomials emerge as a superior choice over Legendre polynomials for solving the problem. They not only demonstrate more consistent convergence but also maintain accuracy across various scenarios, making them the preferred option for selecting the orthogonal polynomial type.

Table 4-6. Convergence success rate [%] for each polynomial type.

	MD	BP MD	PoC$_{iso}$	PoC	Total
Che	100	100	100	100	100
Leg	82.5	85	82.5	80	82.5

Table 4-7. MAE for each polynomial type.

	MD	BP MD	PoC_{iso}	PoC
Che	1.3e−8 km	7.3e−11 km	2.7e−14	2.1e−14
Leg	1.2e−8 km	1.1e−10 km	2.9e−14	2.2e−14

4.2.2 Terminal condition

In this analysis, the focus is on the value of the terminal condition. Based on the conclusions drawn from the previous analysis, Chebyshev polynomials are selected as the orthogonal polynomial type for this study. The following figures only display solutions where convergence is achieved.

Fig. 4-5 shows that the results are consistent across all cases in terms of absolute error at TCA, demonstrating the algorithm's reliability.

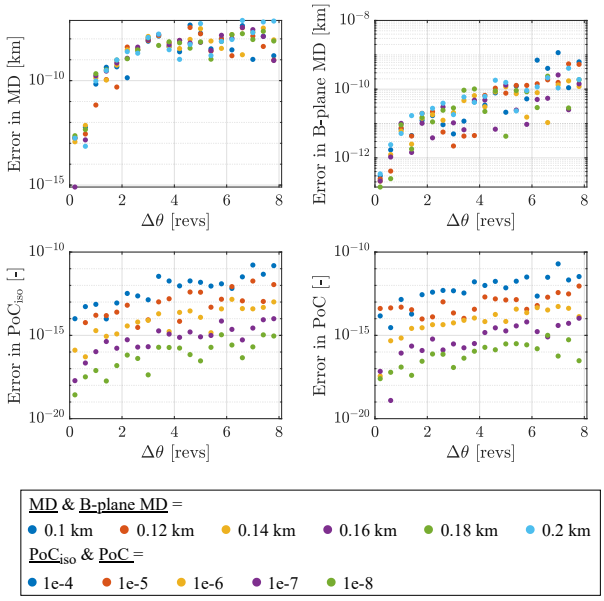


Fig. 4-5. Absolute error for multiple terminal conditions.

Figure 4-6 illustrates the number of iterations needed for convergence in each scenario. To enhance visibility where some results overlap, a slight jitter effect is applied to the dots. The vertical axis is divided into regions corresponding to specific integer iteration counts, allowing for clear distinction between coincident results. Interestingly, larger true anomalies typically require the same or fewer iterations for convergence compared to smaller true anomalies. The exception is the miss distance scenario, where more iterations are necessary as true anomaly increases. Tables 4-8 and 4-9 present a summary of the results, focusing on convergence success rates and MAEs.

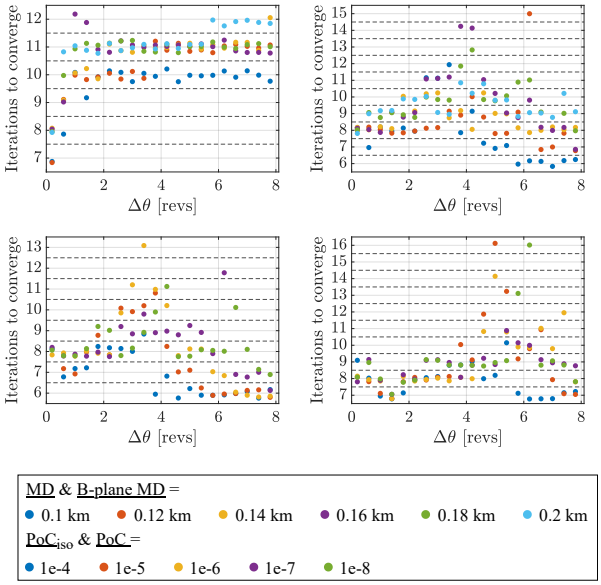


Fig. 4-6. Iterations to converge for multiple terminal conditions.

Table 4-8. MAE for MD [1e−08 km] and B-plane MD [1e−10 km].

δ [km]	0.1	0.12	0.14	0.16	0.18	0.20
MD	0.85	0.82	0.87	0.81	0.64	1.74
BP MD	1.70	1.06	0.51	0.43	0.55	0.89

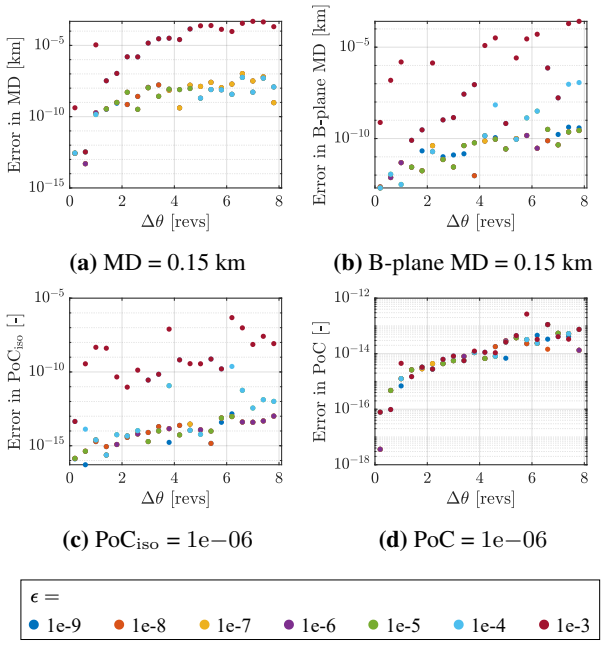
Table 4-9. MAE for PoC_{iso} [1e−12] and PoC [1e−12].

δ	1e−4	1e−5	1e−6	1e−7	1e−8
PoC_{iso}	2.6593	0.2842	0.0261	0.0020	0.0002
PoC	2.0269	0.1436	0.0160	0.0020	0.0001

4.2.3 Convergence tolerance

This test focuses on the convergence tolerance to stop the NLLS iterative loop. Convergence is achieved in all instances, so success rate results are not included.

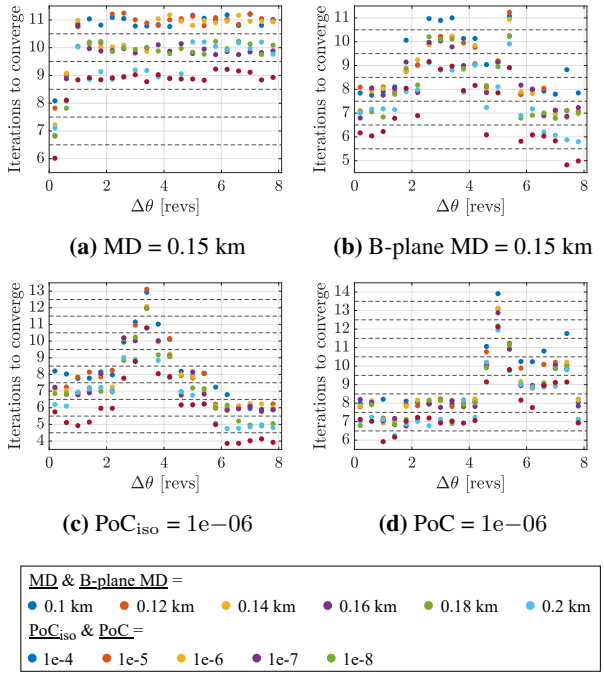
In general, as observed in Figs. 4-7 and 4-8, lower tolerance levels generally require fewer iterations but lead to larger errors. Interestingly, increasing the tolerance to 1e−05 produces results comparable to those at 1e−09. However, at higher tolerances, such as 1e−04 and 1e−03, the errors increase significantly by several orders of magnitude. An exception to this trend is the PoC terminal condition, which consistently shows similar results in both error and iteration counts across all tolerance levels.



(a) MD = 0.15 km **(b)** B-plane MD = 0.15 km

(c) PoC$_{iso}$ = 1e−06 **(d)** PoC = 1e−06

Fig. 4-7. Absolute error for multiple convergence tolerances.



(a) MD = 0.15 km **(b)** B-plane MD = 0.15 km

(c) PoC$_{iso}$ = 1e−06 **(d)** PoC = 1e−06

Fig. 4-8. Iterations to converge for multiple convergence tolerances.

4.3 TFC: case study

This section presents an analysis of the results obtained by applying TFC to a specific CAM problem. Convergence is achieved in all samples.

Fig. 4-9 shows the B-plane landing position of the primary object after performing the maneuver. This position depends on the anticipation time before the TCA. In all scenarios, the final position falls within the ellipse defined for a PoC of 1×10^{-6}. However, the exact position varies around the ellipse based on the true anomaly considered in each case. The unperturbed final position of the primary is also represented, in order to compare the effect of the maneuver. The unperturbed final position of the primary object is also shown for comparison.

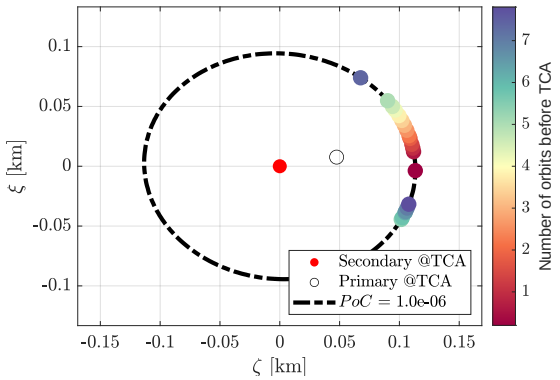


Fig. 4-9. Landing position on B-plane after CAM.

To verify the optimality of a solution to an autonomous problem, it is essential to ensure that the Hamiltonian remains constant over time, as required by the Euler-Lagrange equations. Figure 4-10 demonstrates this by plotting the MAE of $H(t)$ relative to its mean value. As shown, all results stay below 2e−10, indicating that all solutions are indeed optimal.

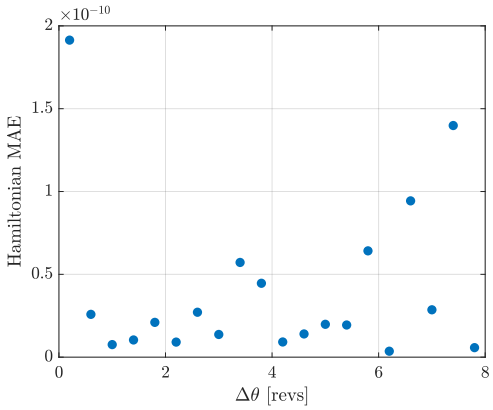


Fig. 4-10. MAE of Hamiltonian with respect to its mean value.

4.4 Comparative study

This section is devoted to compare the performance of TFC with the analytical approach detailed in Sect. 2.4. As discussed in [3], the analytical formulation leads to a

fourth-degree polynomial in ν, yielding four valid solutions. By solving for ν, the initial conditions can be determined and then propagated to evaluate the final outcomes. The optimal solution is identified by integrating the control policy (acceleration) over time, selecting the one with the minimum total change in velocity Δv. Unlike this analytical approach, TFC does not explicitly provide all four solutions but rather converges to a single one of them. The specific solution depends on the initial conditions and parameters used in the problem setup. Figure 4-11 demonstrates that while TFC consistently converges to a valid solution, it does not always yield the most optimal one.

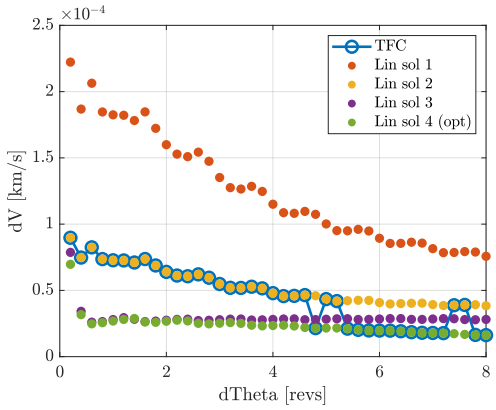


Fig. 4-11. Total change in velocity of the maneuver: comparison

Fig. 4-12 provides a comparative analysis of the computational cost between the two approaches. The analytical solutions outperforms the TFC ones by an entire order of magnitude for lower true anomaly values and by two orders of magnitude as the true anomaly increases. This result highlights the efficiency advantage of the analytical method, particularly as the complexity of the problem grows.

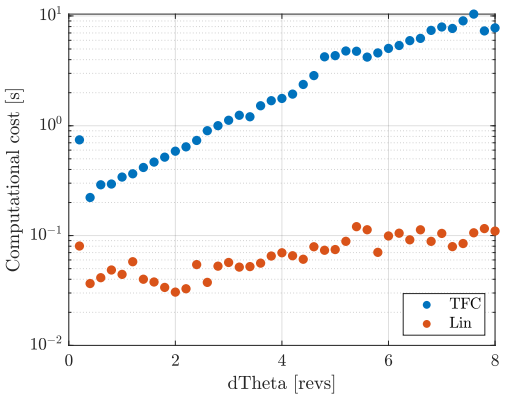


Fig. 4-12. Time spent to compute the solutions using TFC and the analytical approach.

In contrast, as shown in Fig. 4-13, the TFC approach yields a significantly lower error in the actual PoC at TCA after propagating the solutions, despite its higher computational cost. Nevertheless, the errors produced by the analytical results remain well within acceptable limits.

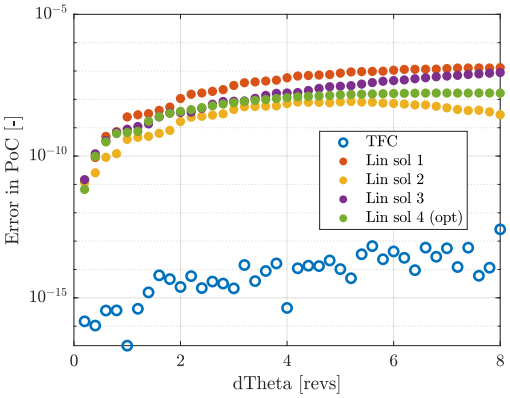


Fig. 4-13. Error in the terminal condition (PoC = 1e−6) using TFC and the analytical approach.

5. Conclusions

The research demonstrates that TFC is a viable method for solving the nonlinear differential equations derived from applying optimal control theory to the CAM problem. A key insight from the development process is that the condition number of the optimization problem is significantly influenced by the choice of time units, which, in turn, affects the loss Jacobian components. To address this, all algorithms were implemented using hours instead of seconds, which proved to enhance stability.

Among the various strategies explored, the SLA with the Gauss-Newton solver, specifically using the *lsqminnorm*() function, stands out as the most efficient in terms of both computational cost and convergence success rate. Conversely, the DLA did not provide substantial improvements, as the simplifications it introduces do not effectively linearize the equations.

Sensitivity analyses revealed that Chebyshev polynomials are superior to Legendre polynomials for this application, offering better performance and robustness. This robustness is particularly evident when varying the terminal condition values and solving the NLLS iterative loop at low tolerance levels. The TFC method not only meets the terminal constraint at TCA with acceptable accuracy but also produces solutions that satisfy the optimality condition of autonomous problems, demonstrated by a constant Hamiltonian over time.

However, the comparative study highlighted two main drawbacks of the TFC approach. First, solution convergence in TFC is highly dependent on the initial guesses and problem configuration parameters, making it difficult to consistently identify the most impulse-efficient control

law. In contrast, the analytical method provides all four possible solutions, enabling a direct comparison of total velocity change to select the most optimal solution. Second, TFC has a significantly higher computational cost, with times ranging from one to two orders of magnitude longer than the analytical method, depending on the maneuver anticipation. This difference is primarily due to the extensive resource demands of the NLLS solver, particularly the *lsqminnorm*() function, which accounts for more than 80% of the total computation time (as identified by the MATLAB profiler tool). Despite these challenges, TFC successfully produces valid solutions within an acceptable timeframe and achieves a lower error in the terminal condition compared to the analytical method. To further enhance the performance of TFC, future work could focus on optimizing the code by implementing the NLLS solver in a faster programming language, such as C, and exploring alternative optimization techniques.

The main potential of TFC is given by the possibility of incorporating additional constraints that do not have an analytical solution, offering a more flexible approach than traditional analytical methods. The approach developed in this study can be expanded or adapted to embed additional constraints within a more comprehensive trajectory mission design, potentially integrating CAM into an all-in-one solution in case of a likely conjunction, for instance, considering the return to the original orbit or accounting for eclipses to prevent the spacecraft from thrusting during shadowed portions of the orbit.

References

[1] Gonzalo Gomez J, Colombo C, Di Lizia P. A semi-analytical approach to low-thrust collision avoidance manoeuvre design. In *70th International Astronautical Congress (IAC 2019)*, 2019, 1–9.

[2] Hernando-Ayuso J, Bombardelli C. Low-Thrust Collision Avoidance in Circular Orbits. *Journal of Guidance, Control, and Dynamics*, 2020, 44, doi: 10.2514/1.G005547.

[3] De Vittori A, Palermo MF, Lizia PD, Armellin R. Low-Thrust Collision Avoidance Maneuver Optimization. *Journal of Guidance, Control, and Dynamics*, 2022, 45(10): 1815–1829, doi:10.2514/1.G006630.

[4] Furfaro R, Mortari D. Least-squares solution of a class of optimal space guidance problems via Theory of Connections. *Acta Astronautica*, 2020, 168: 92–103, doi:https://doi.org/10.1016/j.actaastro.2019.05.050.

[5] Schiassi E, D'Ambrosio A, Johnston H, Furfaro R, Curti F, Mortari D. Complete Energy Optimal Landing on Small and Large Planetary Bodies via Theory of Functional Connections. 2020.

[6] Johnston H, Mortari D. The Theory of Connections Applied to Perturbed Lambert's Problem. 2018.

[7] Johnston H, Mortari D. Orbit Propagation via the Theory of Functional Connections. 2019.

[8] Chan FK. *Spacecraft collision probability*, American Institute of Aeronautics and Astronautics, Inc.2008.

[9] Chan K. Comparison of methods for spacecraft collision probability computations. In *International Space Conference of Pacific-basin Societies (ISCOPS), Montreal, Canada*, 2018.

[10] Valsecchi G, Milani A, Gronchi GF, Chesley S. Resonant returns to close approaches: Analytical theory. *http://dx.doi.org/10.1051/0004-6361:20031039*, 2003, 408, doi:10.1051/0004-6361:20031039.

[11] Rice SO. Mathematical analysis of random noise. *The Bell System Technical Journal*, 1944, 23(3): 282–332, doi:10.1002/j.1538-7305.1944.tb00874.x.

[12] Vinter R. *Optimal Control and Pontryagin's Maximum Principle*. 2013, 1–9, doi:10.1007/978-1-4471-5102-9_200-1.

[13] Reiter JA, Spencer DB. Solutions to rapid collision-avoidance maneuvers constrained by mission performance requirements. *Journal of Spacecraft and Rockets*, 2018, 55(4): 1040–1048.

[14] De Vittori A. Enhanced Collision Avoidance Strategies in the Near-Earth Environment. Ph.D. thesis, Politecnico Di Milano, 2024.

[15] Mortari D. The Theory of Connections: Connecting Points. *Mathematics*, 2017, 5(4), doi:10.3390/math5040057.

[16] Johnston HR. The theory of functional connections: A journey from theory to application. Ph.D. thesis, Texas A&M University, 2021.

[17] Mortari D, Furfaro R. Univariate Theory of Functional Connections Applied to Component Constraints. *Mathematical and Computational Applications*, 2021, 26(1), doi:10.3390/mca26010009.