SpaceOps-2025, ID # 85

Low-Thrust Manoeuvres Optimization Tools for mission analysis and operations

Adrián Lasa Sánchez[a]*, Antonio Lozano Lima[a], Roberto Sánchez Ramos[a], Denis Coccolo Góngora[a]

[a] *GMV, Tres Cantos, Spain*

* Corresponding Author: *alasa@gmv.com*

Abstract

This work presents the response of GMV's flight dynamics solution, **focusSuite**, to the problem of low-thrust trajectory design and operations. The implemented approach is constituted by two programs: IGEOR and OPEPOR. And it can be applied at a high variety of Earth mission scenarios such as electric orbit raising, relocation manoeuvres, end-of-life atmospheric re-entry or retirement to graveyard orbits. Both programs, that will be described in the present work, aim at sequentially optimizing low thrust trajectories to transfer a satellite from an initial orbit to a final target orbit, but each program has a different scope: IGEOR is intended to generate candidate solutions from scratch, whereas OPEPOR is in charge of refining such initial guesses and incorporates detailed environmental forces and sophisticated operational constraints. IGEOR uses an indirect method approach to provide the global optimum solution. However, due to the indirect method nature, IGEOR has some limitations in the dynamical model (supporting only central body and J2 perturbations) and it is able to consider a reduced number of constraints and thrust strategies (coasting arcs, thrust/Isp changes during eclipses). IGEOR uses an indirect method approach to provide the global optimum solution. However, due to the complexity of the underlying formulation, the dynamical model is limited to only central body and J2 perturbations. It also considers a reduced number of thrusting strategies (coasting arcs and thrust/Isp changes during eclipses). OPEPOR implements a hybrid method which allows to add a high variety of considerations. Instead of fully deriving the TBVP to obtain the analytical co-states of the equation of motion, this hybrid formulation simplifies the co-states dynamics into a polynomial evolution. As OPEPOR is a local optimizer, the user shall define an upper and lower boundary of the time of flight to avoid falling in a local optimum. These boundaries can be based on a previous IGEOR execution. The other optimization variables, the co-states, can be initialized internally by the programme. The higher flexibility of OPEPOR allows for a more detailed perturbation model, complex thrust strategies (coasting arcs, switch between different thrust/Isp levels based on different conditions), diverse considerations (e.g. firing time, perigee, apogee, Sun illumination requirements) and easier handling of time-varying target orbital elements.

Keywords: Low thrust, orbit raising, relocation, optimization, maneuver planning, mission analysis

Acronyms/Abbreviations

- Two-boundary value problem (TBVP)
- Non-linear programming problem (NLP)
- Equations of motion (EoMs)
- Self-adaptative differential evolution optimization algorithm (SADE)
- Interior point optimizer (IPOPT)
- Orbital averaging (OA)
- Continuous integration (CI)

1. Introduction

Optimizing low-thrust trajectories is a problem which has been studied for decades, mainly due to the lower fuel consumption of electric propulsion systems. Due to the complexity of the problem and increasing interest for space applications, solutions are continuously evolving. In this paper, GMV presents a robust, accurate and flexible tool that

can be applied to a wide range of scenarios covering multiple user needs. This solution is based on two programs, IGEOR and OPEPOR, which combine different methodologies in an efficient way to provide the required optimal results.

Trajectory optimization aims to find the thrust control law that allows a spacecraft to go from an initial orbit to a target orbit while a certain measurable quantity is minimized. This target is usually the total time of flight, to overcome the low thrust capability of electric propulsion systems, the consumed fuel mass or a mixture of both. At the same time, operational requirements such as eclipses handling, firing times or Sun illumination shall also be considered.

Most of the optimization methods that are used to solve low-thrust problems can be categorized as either indirect, direct or hybrid methods. Other methods, such as shape based or Lyapunov methods, are less used in the frame of terrestrial orbits due to their suboptimality and known difficulties with many revolutions' transfers.

- Indirect methods: These methods use the calculus of variations to describe a TBVP which is later discretized into a NLP. The optimality of the solution is guaranteed by derivation of the necessary mathematical conditions. Moreover, they usually provide more accuracy than direct methods. Nevertheless, their main disadvantage is their small convergence radius and that any change in the dynamics or conditions (e.g. constraints) of the problem requires a full rederivation of the equations.
- Direct methods: These methods discretize the problem into NLP without deriving any optimality condition. The thrust law is usually parameterized based on a polynomial (e.g. Hermite, Gauss-Legendre), but any kind of parametrization can be employed. Their advantages are their robustness to poor initial guesses and the easy accommodation of changes in the dynamics or conditions of the problem. But opposite to indirect methods, they tend to converge to local optimums and provide less accurate results.
- Hybrid methods: This category includes any combination of direct and indirect methods. Usually, they discretize similarly to direct methods, but partially incorporating optimality conditions of indirect methods. Note that global optimality is not guaranteed as they lack full optimality conditions.

2. IGEOR

IGEOR employs an indirect method to generate an optimal solution of a simplified problem according to the processes described in [1] and [2]. It transforms the problem into a TBVP by means of calculus of variations. IGEOR implements a single shooting method where the optimization variables are the initial values of the costate vector and the variable L_1, which is directly related with the number of revolutions.

This program aims to compute an initial draft of the desired low-thrust trajectory to transfer a satellite from an initial orbit to its final target orbit. As an initial guesser, its main purpose is to rapidly compute a feasible trajectory that can be later refined by OPEPOR, which only requires the approximate time of flight as input. However, IGEOR might be used without further refinements in case its functionalities already meet the mission requirements.

2.1. Dynamics Modelling

The dynamics is defined in equinoctial elements to avoid singularities when eccentricity and/or inclination are equal to zero. The equations of motion (EoMs) that define the spacecraft dynamics are:

$$\begin{cases} \dfrac{d\boldsymbol{x}}{dt} = \boldsymbol{C}(\boldsymbol{x}) + \boldsymbol{F}(\boldsymbol{x})\left[\dfrac{\boldsymbol{u}}{m} + \boldsymbol{\gamma}(\boldsymbol{x}, t)\right] \\ \dfrac{dm}{dt} = -\dfrac{u}{g_0 I_{sp}} \end{cases} \tag{1}$$

$$\boldsymbol{C}(\boldsymbol{x}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \sqrt{\dfrac{\mu}{x_1^3}} B^2 \end{pmatrix}^T \tag{2}$$

$$\boldsymbol{F}(\boldsymbol{x}) = \sqrt{\frac{x_1}{\mu}}\frac{1}{B}\begin{pmatrix} 0 & 2x_1 & 0 \\ Bs_L & (B+1)c_L + x_2 & -x_3(x_4 s_L - x_5 c_L) \\ -Bc_L & (B+1)s_L + x_3 & x_2(x_4 s_L - x_5 c_L) \\ 0 & 0 & \frac{Ac_L}{2} \\ 0 & 0 & \frac{As_L}{2} \\ 0 & 0 & x_4 s_L - x_5 c_L \end{pmatrix} \tag{3}$$

$$\begin{cases} s_L = \sin x_6 \\ c_L = \cos x_6 \\ A = 1 + x_4^2 + x_5^2 \\ B = 1 + x_2 c_L + x_3 s_L \end{cases} \tag{4}$$

Where $\boldsymbol{x}$ represents the six modified equinoctial elements (p, f, g, j, k, L), $\boldsymbol{C}$ is the contribution of central gravity forces, $\boldsymbol{F}$ is the contribution due to external forces, $\boldsymbol{u}$ is the thrust vector in RTN frame, m is the wet spacecraft mass, $\boldsymbol{\gamma}$ is a vector representing the sum of perturbed acceleration vectors in RTN frame, g_0 is the gravitational acceleration at sea level and I_{sp} is the specific impulse at sea level.

Dimensionless variables are applied to (1) to make it numerically more convenient for optimization:

$$\hat{x}_1 = \frac{x_1}{d} \quad \hat{m} = \frac{m}{m_0} \quad \hat{u} = \frac{u}{u_{nom}} \quad \hat{t} = t\sqrt{\frac{d}{\mu}}\frac{u_{nom}}{m_0} \tag{5}$$

Where d is a fixed value of the same order than the target semi-latus rectum (p), m_0 is the initial wet spacecraft mass and u_{nom} is the nominal thrust level.

Additionally, a change of the independent variable is introduced, s. This variable is 0 at the initial time and 1 at the final time. To be able to define s, a new constant must be introduced, L_1 (to be optimized), which is directly proportional to the number of revolutions.

$$x_6 = \frac{\mu m_0}{a_t^2 \cdot u_{nom}} L_1 s + x_{6_0} = \frac{L_1 s}{\epsilon} + x_{6_0} \tag{6}$$

Note that (6) defines $x_6 \in [0, \infty)$, so if it is divided by 2π, the number of revolutions is obtained.

Introducing (5) and (6) into (1), EoMs are defined as:

$$\begin{cases} \frac{d\hat{\boldsymbol{x}}}{ds} = \frac{L_1}{\epsilon \hat{G}(\hat{\boldsymbol{x}})}\left(\hat{\boldsymbol{C}}(\hat{\boldsymbol{x}}) + \hat{\boldsymbol{F}}(\hat{\boldsymbol{x}})\left[\frac{\hat{\boldsymbol{u}}}{\hat{m}} + \hat{\boldsymbol{\gamma}}(\hat{\boldsymbol{x}})\right]\right) \\ \frac{d\hat{m}}{ds} = -\frac{L_1}{\epsilon \hat{G}(\hat{\boldsymbol{x}})}\sqrt{\frac{\mu}{d}}\frac{1}{g_0 I_{sp}}\hat{u} = -\frac{L_1}{\epsilon \hat{G}(\hat{\boldsymbol{x}})}\kappa \hat{u} \\ \frac{d\hat{t}}{ds} = \frac{L_1}{\epsilon \hat{G}(\hat{\boldsymbol{x}})} \\ \frac{dL_1}{ds} = 0 \end{cases} \tag{7}$$

$$\hat{G}(\hat{\boldsymbol{x}}) = \frac{d\hat{x}_6}{d\hat{t}} = \hat{\boldsymbol{C}}_{\boldsymbol{6}}(\hat{\boldsymbol{x}}) + \hat{\boldsymbol{F}}_{6,1-3}(\hat{\boldsymbol{x}})\left[\frac{\hat{\boldsymbol{u}}}{\hat{m}} + \hat{\boldsymbol{\gamma}}(\hat{\boldsymbol{x}}, t)\right] \tag{8}$$

Due to the complexity of accommodating a flexible set of perturbations in indirect methods, IGEOR only considers the possibility to add Earth oblateness perturbation based on a J_2 model.

$$\hat{\boldsymbol{\gamma}}(\hat{\boldsymbol{x}}) = -J_2 \mu \frac{R_E^2}{r^4}\begin{pmatrix} \frac{3}{2}\left(1 - \frac{12}{A^2}(x_4 s_L - x_5 c_L)^2\right) \\ -\frac{12}{A^2}(x_4 s_L - x_5 c_L)(x_4 c_L + x_5 s_L) \\ -\frac{6}{A^2}(x_4 s_L - x_5 c_L)(1 - x_4^2 - x_5^2) \end{pmatrix} \tag{9}$$

Where J_2 is the Earth oblateness parameter, R_E is the Earth radius and r is the orbital radius.

2.2. Optimal Control Problem

IGEOR describes the minimum time problem as a TBVP following an indirect formulation:

$$\begin{cases} Min_u\ t_f \\ \dfrac{d\hat{\boldsymbol{x}}}{ds} = \dfrac{L_1}{\epsilon\hat{G}(\hat{\boldsymbol{x}})}\left(\hat{\boldsymbol{C}}(\hat{\boldsymbol{x}}) + \hat{\boldsymbol{F}}\,(\hat{\boldsymbol{x}})\left[\dfrac{\hat{\boldsymbol{u}}}{\hat{m}} + \hat{\boldsymbol{\gamma}}(\hat{\boldsymbol{x}})\right]\right) \\ \dfrac{d\hat{m}}{ds} = -\dfrac{L_1}{\epsilon\hat{G}(\hat{\boldsymbol{x}})}\kappa\hat{u} \\ \dfrac{d\hat{t}}{ds} = \dfrac{L_1}{\epsilon\hat{G}(\hat{\boldsymbol{x}})} \\ \dfrac{dL_1}{ds} = 0 \\ \boldsymbol{x}(t_0) = \boldsymbol{x_0} \quad \boldsymbol{x}(t_f) = \boldsymbol{x_f} \\ m(t_0) = m_0 \quad m(t_f) = free \\ t_f = free \end{cases} \tag{10}$$

The Hamiltonian can be defined as:

$$H = \boldsymbol{\lambda}_{\hat{\boldsymbol{x}}_{1-5}} \cdot \frac{d\hat{\boldsymbol{x}}_{1-5}}{ds} + \lambda_{\hat{t}}\frac{d\hat{t}}{ds} + \lambda_{L_1}\frac{dL_1}{ds} \tag{11}$$

Where $\boldsymbol{\lambda}_{\hat{\boldsymbol{x}}_{1-5}}$, $\lambda_{\hat{t}}$ and λ_{L_1} are the costate associated to each respective state variable. Despite their lack of physical meaning, costate variables can be seen as a thrust profile that changes with a weighted combination of the state variables.

Introducing (7) into (11), the expression is reduced to:

$$H = \frac{L_1}{\epsilon\hat{G}(\hat{\boldsymbol{x}})}\left(\hat{\boldsymbol{F}}_{1-5,1-3}\,(\hat{\boldsymbol{x}})\left[\frac{\hat{\boldsymbol{u}}}{\hat{m}} + \hat{\boldsymbol{\gamma}}(\hat{\boldsymbol{x}})\right] \cdot \boldsymbol{\lambda}_{\hat{\boldsymbol{x}}_{1-5}} + \lambda_{\hat{t}}\right) \tag{12}$$

According to the calculus of variations, the optimal control law is obtained by minimizing the Hamiltonian with respect to the control variable, $\hat{\boldsymbol{u}}$. As a result, the optimal control law can be written as:

$$\frac{\hat{\boldsymbol{u}}_{opt}}{\left\|\hat{\boldsymbol{u}}_{opt}\right\|} = \frac{\boldsymbol{\lambda}^{\boldsymbol{T}}_{\hat{\boldsymbol{x}}_{1-5}}\hat{\boldsymbol{F}}_{1-5,1-3}\,(\boldsymbol{x})}{\left\|\boldsymbol{\lambda}^{\boldsymbol{T}}_{\hat{\boldsymbol{x}}_{1-5}}\hat{\boldsymbol{F}}_{1-5,1-3}\,(\boldsymbol{x})\right\|} \tag{13}$$

Note that *H* is minimized with respect to the thrust modulus when $\hat{u} = \hat{u}_{max}$. This means that the thrust will be active and at the maximum level available. Additionally, analysing (13), it can be seen that a costate vector can be multiplied by any positive scalar without modifying the thrust direction. However, this is not a problem for an indirect method since the magnitude of the costate vector is set by its EoMs [3].

Finally, the costate dynamics can be obtained by differentiating the negative Hamiltonian (12) with respect to the state variable associated with the costate.

$$\begin{cases} \dfrac{d\boldsymbol{\lambda}_{\hat{\boldsymbol{x}}_i}}{ds} = \dfrac{H}{\hat{G}^2(\hat{\boldsymbol{x}})}\dfrac{\delta\hat{G}(\hat{\boldsymbol{x}})}{\delta\hat{\boldsymbol{x}}_i} - \dfrac{L_1}{\epsilon\hat{G}(\hat{\boldsymbol{x}})}\left[\dfrac{\boldsymbol{\lambda}^{\boldsymbol{T}}_{\hat{\boldsymbol{x}}_{1-5}}}{\hat{m}} \cdot \dfrac{\delta\hat{\boldsymbol{F}}_{1-5,1-3}\,(\hat{\boldsymbol{x}})}{\delta\hat{\boldsymbol{x}}_i}\dfrac{\hat{\boldsymbol{u}}_{opt}}{\left\|\hat{\boldsymbol{u}}_{opt}\right\|} + \left(\dfrac{\delta\hat{\boldsymbol{F}}_{1-5,1-3}\,(\hat{\boldsymbol{x}})}{\delta\hat{\boldsymbol{x}}_i}\hat{\boldsymbol{\gamma}}(\hat{\boldsymbol{x}}) + \hat{\boldsymbol{F}}_{1-5,1-3}\,(\hat{\boldsymbol{x}})\dfrac{\delta\hat{\boldsymbol{\gamma}}(\hat{\boldsymbol{x}},)}{\delta\hat{\boldsymbol{x}}_i}\right)\boldsymbol{\lambda}_{\hat{\boldsymbol{x}}_{1-5}}\right] \\ \dfrac{d\lambda_{L_1}}{ds} = -\dfrac{H}{L_1} \\ \dfrac{d\lambda_{\hat{t}}}{ds} = 0 \end{cases} \tag{14}$$

As the Hamiltonian is independent of time, the dynamics of the costate $\lambda_{\hat{t}}$ in (14) reveals that this costate does not play any role in the optimization and the calculus of variations stablishes that its value is -1. It also implies that $\frac{d\hat{t}}{ds}$ is not necessary for optimization. In any case, the derivative must be computed to know the transfer time at each integration node.

Transversality conditions from the calculus of variations state that, if a variable is free at both extremes such as L_1, its costate boundary conditions are equal to 0 ($\lambda_{L_1}(t_0) = \lambda_{L_1}(t_1) = 0$).

2.3. Cost Function and Optimization Algorithms

IGEOR uses SADE to obtain a first estimate of the optimization variables from scratch. This solution can be later (manually or automatically) refined by IPOPT algorithm. SADE, as a heuristic algorithm, has drawbacks to find an accurate solution of the problem. But it can efficiently sweep the search space to narrow it down. On the contrary, IPOPT requires an initial guess in the surroundings of the optimal solution to be able to converge, although it produces more accurate results and complies better with constraints than SADE.

SADE and IPOPT manage cost function and constraints differently. The former requires the definition of only one scalar function, while the latter can consider each condition separately. Consequently, a cost function is defined for each method:

$$J_{SADE} = F_{cs} \tag{15}$$

$$J_{IPOPT} = t_f \tag{16}$$

Where F_{cs} represents the sum of all the constraint functions involved in the problem.

IGEOR formulation allows to remove the time from the cost function because the transfer is integrated with $s \in [0,1]$ and the constraint $\lambda_{L_1}(t_1) = 0$ is related with the total time of flight through the number of revolutions. However, this is only applied to SADE to reduce the effort of the optimizer. As IPOPT handles cost function and constraint separately, time of flight is maintained as part of the cost function.

2.4. Constraint and Impositions

The main constraint of the optimization is to reach the target orbit. As explained in previous sections, SADE and IPOPT handle constraints in a different way. SADE tries to minimize the constraint function as a penalty function included in the cost function (17) (a threshold is defined in this function as a convergence criteria) while IPOPT can stablish upper and lower boundaries to each target orbital element (18).

$$SADE \Rightarrow F_{cs_{TO}} = \sum_{i=1}^{6} w_i \left(x_{i_f}^k - x_{i_p}^k\right)^2 + \lambda_{L1}^2 \tag{17}$$

$$IPOPT \Rightarrow \begin{cases} w_i\, Tol \geq w_i \left\| x_{i_f}^k - x_{i_p}^k \right\| \; for\; i = 1,6 \\ \lambda_{L1} = 0 \end{cases} \tag{18}$$

Where $\boldsymbol{x}^k$ is the modified equinoctial state vector, the subindex f represents the target final conditions, p represents the final propagated state and w is a weight value associated to the corresponding orbital element.

IGEOR formulation established that the first five orbital elements must be always targeted to ensure optimality and convergence. So, when a target orbital element is not defined (ex: ω in circular orbits), the programme still needs a value to guarantee convergence. In those cases, a 0 value can be set as it does not assume a problem to the optimization.

On the other hand, the indirect nature of IGEOR makes that adding any new constraint or change in the orbital dynamics requires a rederivation of the problem. However, the addition of some imposition can be tolerated by the formulation as they impact in the dynamic is reduced:

- Eclipses: A secondary engine level (thrust and I_{sp}) can be defined during eclipses (being possible to define zero thrust during this condition).
- Thrust degradation: A parametric degradation can be defined for nominal and secondary engine levels based on the thrusting time with a linear or step evolution.
- Mass waste: A constant fuel loss can be defined during non-thrusting intervals.
- Coasting arcs: Non-thrusting periods can be defined by date, transfer time or when crossing a delimited threshold delimited such as specific semi-major axis, apogee, or perigee.

3. OPEPOR

OPEPOR employs a hybrid formulation allowing the addition of numerous perturbation forces, constraints and impositions without changes in the formulation of the optimization problem. This is achieved thanks to parametrize the thrust control law according to a piecewise linear function of the costate evolution. It also applies a single shooting method is used, where the optimization variables are the total time of flight and a set of time distributed costate vectors.

OPEPOR results a powerful software able to compute accurate low-thrust trajectories considering diverse conditions. However, due to its hybrid formulation, it may converge to local optimums. To overcome this risk, it is recommended that the user stablishes proper upper and lower boundaries for the total time of flight. For this purpose, GMV proposes to use IGEOR, which can provide an optimum time of flight for a similar case.

3.1. Dynamical Modelling

OPEPOR uses the same dynamical model than IGEOR, except for the variable change described in (6).

$$\begin{cases} \dfrac{d\hat{\boldsymbol{x}}}{d\hat{t}} = \hat{\boldsymbol{C}}(\hat{\boldsymbol{x}}) + \hat{\boldsymbol{F}}\,(\hat{\boldsymbol{x}})\left[\dfrac{\hat{\boldsymbol{u}}}{\hat{m}} + \hat{\boldsymbol{\gamma}}(\hat{\boldsymbol{x}})\right] \\ \dfrac{d\hat{m}}{d\hat{t}} = -\dfrac{L_1}{\epsilon \hat{G}(\hat{\boldsymbol{x}})}\kappa\hat{u} \end{cases} \tag{19}$$

Opposite to IGEOR, OPEPOR can accommodate changes in the optimization problem easily thanks to its hybrid nature. Therefore, it offers a selectable set of orbital perturbations which can be considered: non-spherical Earth gravity field, solid Earth tides, ocean tides, third body gravitational forces, indirect J2/Moon interaction, general relativistic effects, solar radiation pressure, aerodynamic forces, empirical accelerations, and non-gravitational forces (payload radiation and thermal imbalance).

3.2. Optimal Control Problem

Minimum time problem in OPEPOR is defined as:

$$\begin{cases} Min_u\, t_f \\ \dfrac{d\hat{\boldsymbol{x}}}{d\hat{t}} = \hat{\boldsymbol{C}}(\hat{\boldsymbol{x}}) + \hat{\boldsymbol{F}}\,(\hat{\boldsymbol{x}})\left[\dfrac{\hat{\boldsymbol{u}}}{\hat{m}} + \hat{\boldsymbol{\gamma}}(\hat{\boldsymbol{x}})\right] \\ \dfrac{d\hat{m}}{d\hat{t}} = -\kappa\hat{u} \\ \boldsymbol{x}(t_0) = \boldsymbol{x_0} \quad \boldsymbol{x}(t_f) = \boldsymbol{x_f} \\ m(t_0) = m_0 \quad m(t_f) = free \\ t_f = free \end{cases} \tag{20}$$

The Hamiltonian can be described as:

$$H = \boldsymbol{\lambda}_{\hat{\boldsymbol{x}}_{1-5}} \cdot \frac{d\hat{\boldsymbol{x}}_{1-5}}{dt} \tag{21}$$

Following the same procedure as before with (13), the optimal control law is defined as:

$$\frac{\hat{\boldsymbol{u}}_{opt}}{\|\hat{\boldsymbol{u}}_{opt}\|} = \frac{\boldsymbol{\lambda}_{\hat{\boldsymbol{x}}_{1-5}}\hat{\boldsymbol{F}}_{1-5,1-3}\,(\boldsymbol{x})}{\|\boldsymbol{\lambda}_{\hat{\boldsymbol{x}}_{1-5}}\hat{\boldsymbol{F}}_{1-5,1-3}\,(\boldsymbol{x})\|} \tag{22}$$

Assuming a costate dynamics not dependant on the Hamiltonian implies that optimality conditions are not kept. As a consequence, no rederivation is required when modifying the problem to solve. As a result, OPEPOR, instead of deriving the costate dynamics as in the indirect approach, the costate evolution is here replaced by a piecewise linear polynomial as described in Fig. 1. This new costate dynamic requires to know the time of flight prior to any evaluation of each shooting, as it is required to distribute the costate vectors that will form part of the optimization variables.

Initially the software distributes the costate vector nodes equally in time (black nodes in Fig. 1), but this is not mandatory. In fact, OPEPOR allows a re-optimization from an epoch later than the original. In this case, OPEPOR will compute a new costate node at the new initial epoch (white node in Fig. 1) based on the previous solution. This new node will be used along with the subsequent ones for the optimization, while preserving their relative distribution respect to the time of flight.

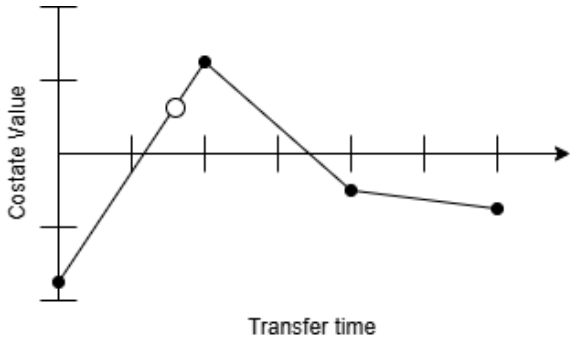

Fig. 1: OPEPOR costate dynamics (3 costate arcs), including a new node created for re-optimization

3.3. Cost Function and Optimization Algorithm

IGEOR gives an idea of the total time of flight to set proper bounds in OPEPOR, but an initial guess for the costate vectors is also needed. To generate that initial guess, the same approach than IGEOR is recommended: optimization with a SADE algorithm to obtain a first estimate, then followed by a second optimization with IPOPT.

OPEPOR cost function is defined as follow:

$$J_{SADE} = \sqrt{w_{t_f} t_f + F_{cs}} \tag{23}$$

$$J_{IPOPT} = w_{t_f} t_f \tag{24}$$

3.4. Constraints and Impositions

OPEPOR defines target constraints as IGEOR in (17) and (18). But opposite to IGEOR, OPEPOR costate EoMs do not regulate costate magnitude because they are not derived from the calculus of variations. Consequently, costate control variables must be bounded in the optimization process (usually $\lambda \in [-1,1]$). Additionally, it is recommended to constraint the norm of each costate control vector to be unitary ($\left\|\boldsymbol{\lambda}_{\hat{\boldsymbol{x}}_{1-5}}\right\| = 1$) as it helps to delimit the search space [3]. SADE execution does not make use of this constraint, but it forces this condition computing the unitary vector for each costate vector of the generated random population.

Additionally, OPEPOR supports the next constraints:

- Maximum eccentricity during transfer.
- Minimum and maximum perigee and apogee heights during a transfer period.
- Maximum thrust angular rate.

And in addition to the impositions considered in IGEOR, OPEPOR also implements the following impositions:

- Firing times: Maximum and minimum firing times together with minimum time between firings (time can be defined by time units or number of revolutions)
- ATOX constraint: Turn off thruster near periapsis under a threshold altitude.
- Thrust level restrictions: Nominal engine level and switch to secondary level after a specific period; or nominal level, cut-off during eclipses, restart with secondary level until recover nominal level after a defined duration.
- Time after eclipses to recover thrust.
- Sun illumination constraint: minimum allowed angle between Sun-satellite vector and thrust vector.

4. Common Functionalities

Sections 2 and 3 describe the basics of the optimization approach behind IGEOR and OPEPOR, but there are some relevant common features that will be commented in this section.

4.1. Target Definition

Target elements can be defined in keplerian or equinoctial elements given the possibility to use them as osculating or mean elements. In addition, x_6 can be defined as true anomaly, argument of latitude or argument of longitude for any set target orbital elements.

4.2. Moving Targets

The optimization problem does not always require targeting a fixed value. The target values may vary depending on the final time when the satellite arrives at that orbit. Therefore, the target can be defined either as a fixed value, based on a linear evolution with respect to a reference date for Ω and x_6, or also based on a previously computed orbit.

4.3. Optimization Segments

Both programs can consider sequential optimization processes (segments) where each segment defines the initial state condition based on the final state of the previous segment. This allows to stablish various objectives along the orbital transfer (e.g. orbit raising followed by a fine position phase for insertion into final orbit).

4.4. Re-optimization

A re-optimization of the trajectory is always supported, but both programmes have implemented a strategy to efficiently provide an initial guess when a state vector later than the initial state vector of the trajectory is provided. Section 3.2 explains the method used by OPEPOR for this re-optimization cases. On the other hand, IGEOR will compute via interpolation the new starting costate vector of the trajectory to re-optimize (in IGEOR only the initial costate vector is part of the optimization variables).

This strategy proves particularly advantageous in satellite operations due to periodic manoeuvre commanding, as it enables subsequent optimizations to begin with a more accurate initial estimate and to avoid initializing from scratch.

4.5. Propagation Methods

Propagation can be performed with OA or CI. The averaging of the orbital dynamics allows a quicker propagation as integration step can be much higher than CI, and its contribution to the smoothness of the state dynamics improves the converge optimization rate. However, the information of the fast variable, x_6, and dependant variables is lost in the process although the application of (28) can help to mitigate this problem.

The averaged dynamics is defined by averaging the original dynamics over one orbital period (T) [4]. Note that equation (26) is only relevant in the IGEOR case, as OPEPOR still assumes a piecewise linear distribution of the costate of the averaged system.

$$\dot{\bar{\hat{\boldsymbol{x}}}} = \frac{d\bar{\hat{\boldsymbol{x}}}}{d\hat{t}} = \frac{1}{T}\int_0^{2\pi} \frac{d\hat{\boldsymbol{x}}}{d\hat{t}}\frac{d\hat{t}}{d\hat{x}_6} d\hat{x}_6 \tag{25}$$

$$\dot{\boldsymbol{\lambda}} = -\frac{1}{T}\int_0^{2\pi} \boldsymbol{\lambda}\frac{\delta\dot{\hat{\boldsymbol{x}}}}{\delta\hat{\boldsymbol{x}}}\frac{d\hat{t}}{d\hat{x}_6} d\hat{x}_6 + \frac{1}{T}\int_0^{2\pi} \frac{\delta\hat{x}_6}{\delta\hat{\boldsymbol{x}}}\boldsymbol{\lambda}\frac{d\hat{\boldsymbol{x}}}{d\hat{t}}\left(\frac{d\hat{t}}{d\hat{x}_6}\right)^2 d\hat{x}_6 - \frac{1}{T}\int_0^{2\pi} \frac{\delta\hat{x}_6}{\delta\hat{\boldsymbol{x}}}\left(\frac{d\hat{t}}{d\hat{x}_6}\right)^2 d\hat{x}_6 \times \boldsymbol{\lambda}\dot{\bar{\hat{\boldsymbol{x}}}} \tag{26}$$

$$T = \int_0^{2\pi} \frac{d\hat{t}}{d\hat{x}_6} d\hat{x}_6 \tag{27}$$

$$\bar{\hat{x}}_6 \approx \omega + \Omega + M \tag{28}$$

5. Workflow

Even though IGEOR can work as stand-alone software which does not require further iteration, it only covers a limited number of considerations. On the other hand, OPEPOR can provide accurate results for a higher number of scenarios, although it has the risk to converge to local optimums if proper boundaries for the total time of flight are not defined. According to GMV's experience, the schema in Fig. 2 is the most robust workflow for an optimization problem that starts from scratch. Note that the propagation method in IGEOR is not fixed. Usually, an execution with

OA is enough to feed OPEPOR with an initial guess (as it has better convergence rate and performance). Nonetheless, complex cases could require the use of CI in IGEOR. Moreover, an intermediate OPEPOR optimization using OA with IPOPT can be helpful for complex cases too. On the other hand, if the user is only interested in rough outputs such as total time of flight or mass expenditure, then the CI execution is not usually required.

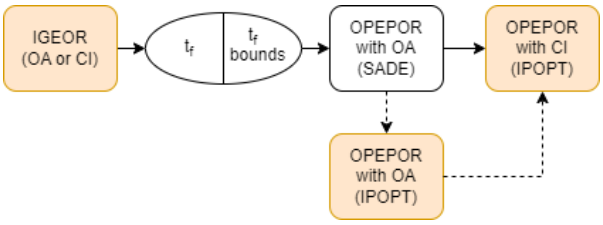

Fig. 2: Robust workflow

Once the user has a draft of the optimal trajectory, the process can directly start with OPEPOR (IPOPT) using that previous solution as initial guess. Additionally, for simpler cases such as relocation problems, IGEOR execution can be skipped without any impact.

It is important to mention that IGEOR and OPEPOR will lead to slightly different results under the same conditions as they do not share the same dynamics. Additionally, the use of alternative propagation methods (OA or CI) can also create small discrepancies.

6. Injection Ω Optimization

In addition to traditional orbit optimization, OPEPOR can be used as an optimizer of the initial longitude of the ascending node, Ω_0, provided by the launcher to reduce the fuel consumption dedicated to correct perturbations effects during the transfer (mainly Earth oblateness effect). It is also able to provide analysis of the impact of changes in the injection date.

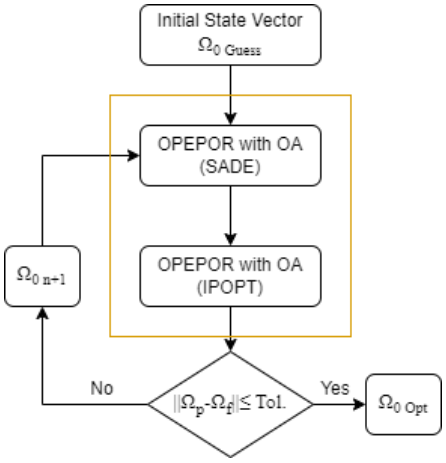

Fig. 3: Injection Ω_0 Optimization

Fig. 3 describes the optimization loop that is automatically performed: an injection Ω_0 is assumed and an optimization is performed with a free target Ω_f. After the optimization, the error Ω_f-Ω_p is computed. Based on that error, a new Ω_0 is defined based on (29) and the loop is restarted until convergence is reached ($\Omega_p \approx \Omega_f$ being Ω_f a free optimization parameter). This analysis can be done to fine the injection Ω_0 or the epoch at which that injection Ω_0 is achieved (assuming the application of a coasting arc from an initial state vector to the optimized epoch).

Once the injection Ω_0 is optimized, the software can perform two additional optimizations, moving the epoch of the start of the transfer before and after the nominal one. This additional optimization must target the final Ω_f to assess the penalties, in time and fuel, caused by changing the starting epoch of the transfer.

$$\Omega_{0_{n+1}} = \Omega_{0_n} + \left(\Omega_f - \Omega_{p_n}\right) \tag{29}$$

7. Results for Operational Cases

To demonstrate the flexibility and potential of IGEOR and OPEPOR, six different cases will be analysed: a GTO to GEO transfer, a LEO disposal, a MEO orbit raising, a LEO orbit raising, a MEO relocation and an injection Ω_0 optimization. IGEOR results are always computed with OA as its purpose is to feed OPEPOR with an estimate of the time of flight, whereas OPEPOR results are finally obtained with CI except for case 6 as only an estimate of the injection Ω is required.

Note that orbital elements that are not defined still need to be targeted by IGEOR (ex: ω in circular orbits). As explained in Section 2.4, these cases establish a value of 0 for the corresponding orbital elements without generating any optimality and convergence problem. As this element are not defined there are omitted from the results as the error respect to the target has lack of meaning.

7.1. Case 1: Orbit Raising GTO-GEO

This case represents a GTO to GEO transfer (see Table 1) with J2 perturbation [2]. The spacecraft has a mass of 2000 kg, a thrust of 0.35 N and an Isp of 2000 s. In addition, a second execution of OPEPOR is performed with a more realistic perturbation model ($J_{32,32}$, solar radiation pressure and solar and lunar third body perturbations). The solar radiation coefficient is set to 2.2 and the radiation area to 25 m^2.

Table 1: Initial, target and results for a GTO-GEO case

	$\boldsymbol{x_0}$	$\boldsymbol{x_f}$	IGEOR $\boldsymbol{x_f - x_p}$	OPEPOR J_2 propagation model $\boldsymbol{x_f - x_p}$	OPEPOR Complex propagation model $\boldsymbol{x_f - x_p}$
a (km)	24505.9	42165.0	$-5.97 \cdot 10^{-8}$	$-2.46 \cdot 10^{-4}$	$5.78 \cdot 10^{-3}$
e	0.725	0.0	$0.57 \cdot 10^{-12}$	$1.00 \cdot 10^{-4}$	$1.69 \cdot 10^{-4}$
i (deg)	7.0	0.0	$2.10 \cdot 10^{-10}$	$1.00 \cdot 10^{-2}$	$1.57 \cdot 10^{-2}$
Ω (deg)	0.0	-	-	-	-
ω (deg)	0.0	-	-	-	-
L (deg)	0.0	-	-	-	-

IGEOR solution led to a transfer of 137.75 days (mass consumption of 212.57 kg). On the other hand, OPEPOR (J_2 propagation model) reached a final solution of 137.55 days (212.26 kg), and OPEPOR (complex propagation model) found a solution of 137.65 days (212.42 kg).

7.2. Case 2: LEO Disposal

This case represents a LEO disposal transfer (see Table 2) with J2 and aerodynamic drag perturbation. The spacecraft has a mass of 2000 kg, a thrust of 0.08 N, an Isp of 1500 s, a C_D of 3.0 and a constant drag surface area of 10 m^2. As this case aims for deorbiting, only reducing a and maintaining a circular orbit is needed.

Table 2: Initial, target and results for a LEO disposal case

	$\boldsymbol{x_0}$	$\boldsymbol{x_f}$	IGEOR $\boldsymbol{x_f - x_p}$	OPEPOR $\boldsymbol{x_f - x_p}$
a (km)	7078.14	6778.14	$4.02 \cdot 10^{-6}$	$-3.28 \cdot 10^{-3}$
e	0.0013	0.0	$4.50 \cdot 10^{-8}$	$7.54 \cdot 10^{-5}$
i (deg)	98.19	98.19 / -	$5.48 \cdot 10^{-11}$	-

Ω (deg)	0.0	0.0 / -	$1.47 \cdot 10^{-11}$	-
ω (deg)	90.0	-	-	-
L (deg)	0.0	-	-	-

IGEOR found an optimal time of flight of 47.43 days (mass consumption of 22.31kg), but also targeting i_f and Ω_f. Instead, OPEPOR reached a solution of 46.44 days (21.84 kg).

This case is impacted by two drawbacks due to IGEOR formulation. Firstly, IGEOR cannot target a reduced set of orbital elements (in this case the effects can be considered negligible as the minimum solution only under J2 effects does not generate considerable changes i and Ω). Secondly, IGEOR does not admit the use of aerodynamical perturbation and only gives an estimate of the optimal time of flight. As a result, the optimal time of flight is lower in OPEPOR (because it can consider drag perturbation which helps to deorbiting).

7.3. Case 3: Orbit Raising MEO

This case represents a MEO orbit raising (see Table 3) with J2 perturbations. The spacecraft has a mass of 2000 kg, a primary engine level of 0.35 N of thrust and an Isp of 2000 s and a secondary engine level, which is applied during eclipses, of 0.20 N of thrust and 1500 s of Isp.

Target Ω_f and L_f in Table 3 are defined at the initial epoch. This means that their values evolve with time. This evolution is defined as a linear evolution from the reference epoch (initial epoch for this case). The rate of this evolution can be estimated, considering the effect of the J2 perturbation and the dynamics of a circular orbit, to be -0.02588 º/day for Ω and 613.72 º/day for L.

Usually, in real operations, targeting L in an orbit raising is done in two steps. First, a segment is defined to reach the target orbit reducing a_f some kilometres below final target (10 km for this particular case) and skipping the targeting of L_f. And latter, a second segment dedicated to fine positioning is defined to reach the final orbital slot (this final relocation requires a small fraction of the full orbit raising). This is done only by OPEPOR as IGEOR is execute with OA and fine positioning manoeuvres don't need a time estimate due to its reduced time of flight and simplicity respect to other scenarios.

Table 3: Initial, target and results for a MEO orbit raising

	$\boldsymbol{x_0}$	$\boldsymbol{x_f}$	IGEOR $\boldsymbol{x_f - x_p}$	OPEPOR $\boldsymbol{x_f - x_p}$	
				1º Segment	2º Segment
a (km)	12000.0	29600.0	$5.83 \cdot 10^{-1}$	$-3.07 \cdot 10^{-3}$	$-5.09 \cdot 10^{-3}$
e	0.0	0.0	$2.93 \cdot 10^{-6}$	$1.27 \cdot 10^{-4}$	$1.00 \cdot 10^{-4}$
i (deg)	56.0	56.0	$-1.21 \cdot 10^{-4}$	$9.97 \cdot 10^{-3}$	$9.58 \cdot 10^{-3}$
Ω (deg)	354.05	330.0	$-6.68 \cdot 10^{-3}$	$-1.00 \cdot 10^{-2}$	$-4.37 \cdot 10^{-3}$
ω (deg)	0.0	-	-	-	-
L (deg)	0.0	-/150.0	-	-	$6.97 \cdot 10^{-3}$

IGEOR solution led to a transfer of 135.00 days (mass consumption of 201.53 kg). On the contrary, OPEPOR found an optimal solution of 134.96 days (205,30 kg) for the first segment, and a time of flight of 16.21 days (25.01 kg) for the second, being the total time of flight 151.17 days (230.31 kg).

Comparing the first segment of OPEPOR with IGEOR, OPEPOR converges to a solution with a lower time of flight, but higher mass. The explanation is the amount of time that the spacecraft's engine is set at each engine level. OA considers eclipses when evaluating (26), which implies that eclipse time is also averaged and some discrepancies are introduced.

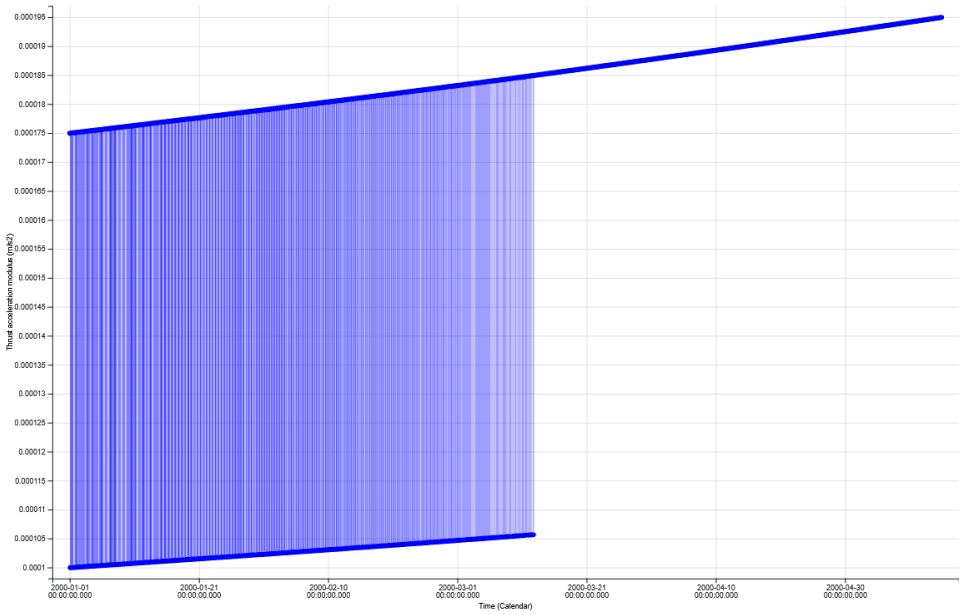

Fig. 4: Thrust magnitude evolution along the transfer (thrust off during eclipses)

7.4. Case 4: Orbit Raising LEO

This case represents a LEO orbit raising to a target orbit defined in mean keplerian elements (see Table 4) with complex perturbation model ($J_{8,8}$, aerodynamic drag, solar radiation pressure and solar and lunar third body perturbation). The spacecraft has a mass of 180 kg, a thrust of 0.015 N, an Isp of 2000 s, a C_D of 3.0, a constant drag surface area of 1 m^2, a solar radiation coefficient of 1.2 and a radiation area of 1.5 m^2. Additionally, a firing constraint in terms of revolutions has been considered. The spacecraft thrust in cycle of two revolutions, one revolution with manoeuvres and the next without manoeuvres.

In this case, OPEPOR has been used as a standalone software, i.e., no initial guess of the time of flight has been computed. The definition of the time of flight boundaries has been done based on external results of similar missions, demonstrating that the initial guess does not need to be precise or come mandatory from IGEOR.

Table 4: Initial, target and results for a LEO orbit raising

	$\boldsymbol{x_0}$	$\boldsymbol{x_f}$ (mean elements)	OPEPOR $\boldsymbol{x_f - x_p}$ (mean elements)
a (km)	7019.11	7578.0	$-1.00 \cdot 10^{-2}$
e_x	$4.52 \cdot 10^{-4}$	0.0	$2.22 \cdot 10^{-5}$
e_y	$-1.26 \cdot 10^{-5}$	0.001	$-4.11 \cdot 10^{-5}$
i_x	-1.20	-	-
i_y	$-1.55 \cdot 10^{-3}$	-	-
L (deg)	-1.60	-	-

OPEPOR found an optimal time of flight of 79.27 days (mass consumption of 2.63 kg). Fig. 6 shows how the spacecraft turns on and off the thruster according to the specifications of thrusting one revolution and performing a coasting arc the next revolution.

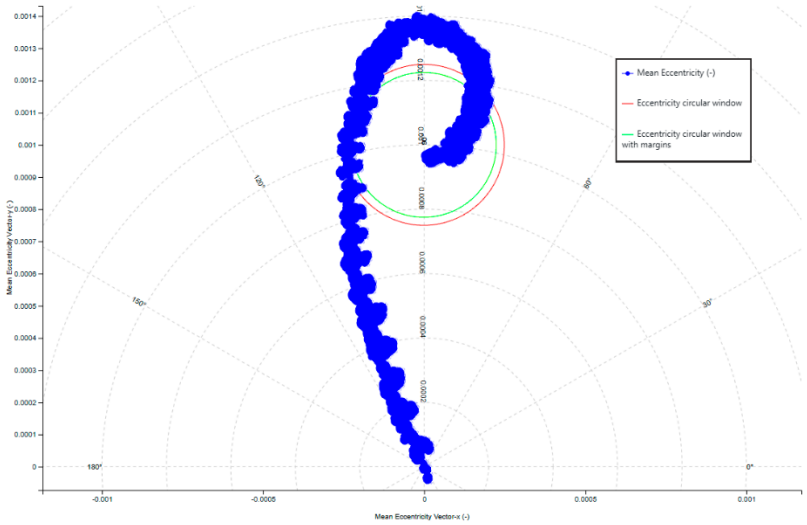

Fig. 5: Mean eccentricity vector (e_x, e_y) evolution along the transfer

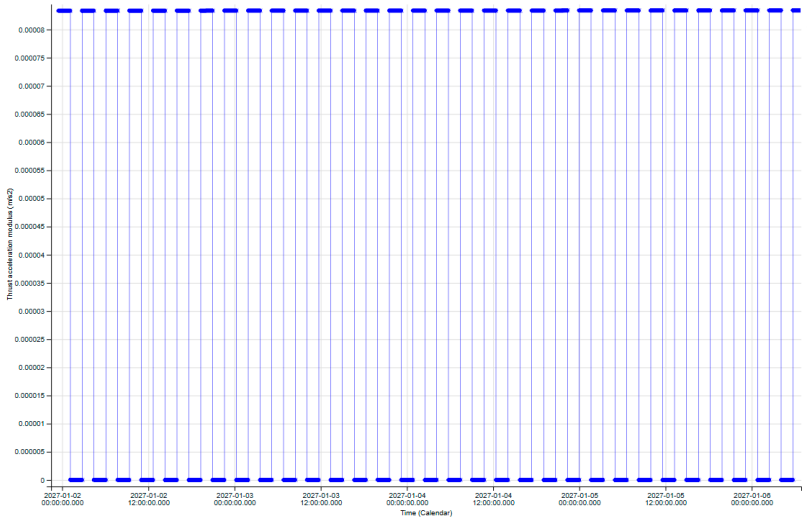

Fig. 6: Thrust acceleration modulus evolution at the beginning of the transfer

7.5. Case 5: Relocation MEO

This case represents a relocation manoeuvre in a MEO orbit (see Table 5) with J2 perturbations and a Sun illumination imposition of 90º (it means that if the thrust direction is fixed in body frame at -X direction, the satellite face X is not illuminated by the Sun). The spacecraft has a mass of 2000 kg, a primary engine level of 0.18 N of thrust and an Isp of 1750 s.

This scenario has been designed to be the most constrained case regarding the Sun illumination requirement. A Sun elevation angle of 0º, combined with a minimum angle of 90º creates a geometry where the optimal thrust direction needs to be modified the whole orbit to comply with the requirement, one half to comply with the requirement itself and another to impose symmetry in the thrust law.

As in case 3, target Ω_f and L_f in Table 5 are defined at the initial epoch and a linear evolution from that reference epoch is assumed. The rates of these parameters are the same than in case 3.

Table 5: Initial, target and results for a MEO relocation manoeuvre

	$\boldsymbol{x_0}$	$\boldsymbol{x_f}$	OPEPOR with imposition $\boldsymbol{x_f - x_p}$	OPEPOR without imposition $\boldsymbol{x_f - x_p}$
a (km)	29600.0	29600.0	$8.31 \cdot 10^{-4}$	$-4.88 \cdot 10^{-3}$
e	0.0	0.0	$8.87 \cdot 10^{-5}$	$3.56 \cdot 10^{-5}$
i (deg)	56.0	56.0	$6.44 \cdot 10^{-3}$	$5.93 \cdot 10^{-3}$
Ω (deg)	298.04	298.04	$-4.93 \cdot 10^{-3}$	$-5.74 \cdot 10^{-3}$
ω (deg)	0.0	-	-	-
L (deg)	0.0	40.0	$-3.26 \cdot 10^{-4}$	$2.03 \cdot 10^{-3}$

OPEPOR solution with Sun illumination imposition led to a transfer of 8.13 days (mass consumption of 7.37 kg). While the case without the imposition found a solution of 6.38 days (5.78 kg). Fig. 7 and Fig. 8 show that in-plane components move around 180º during the first half (when lowering the semi-major axis) and around 0º during the second (when increasing the semi-major axis). However, the case with the Sun illumination imposition, Fig. 7, shows a bigger oscillation and with a linear trend. This behaviour is due to the fact that the whole orbit is restricted by the Sun illumination imposition.

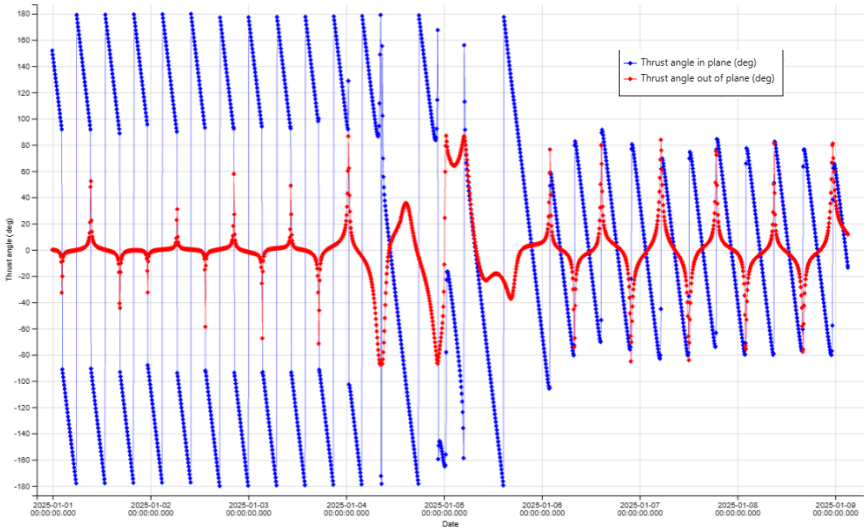

Fig. 7: Thrust in and out of plane angles for relocation with Sun illumination imposition

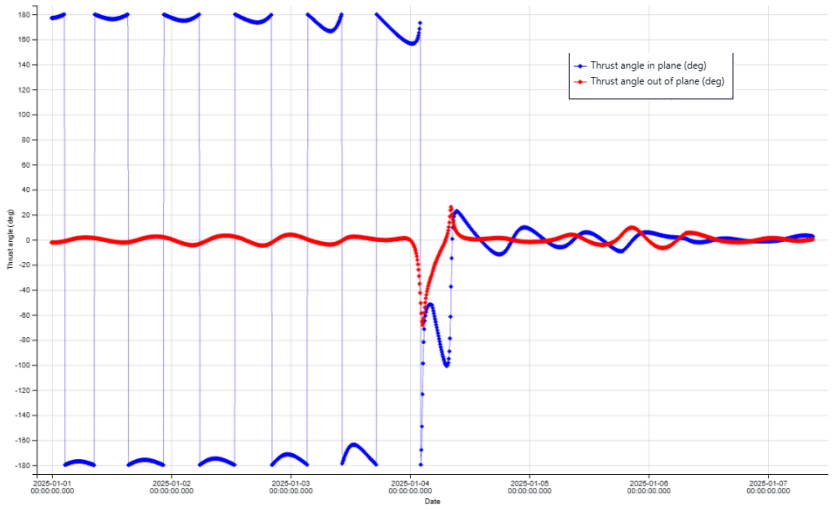

Fig. 8: Thrust in and out of plane angles for relocation without Sun illumination imposition

Fig. 9 shows a pyramid shape instead of creating a coasting arc in the middle. It is caused by the minimum time problem formulation, where the solution implies thrusting all the time. However, if mission requirements require a coasting arc, this can be forced in the relocation manoeuvre and optimize the trajectory and coasting time required to reach the new slot.

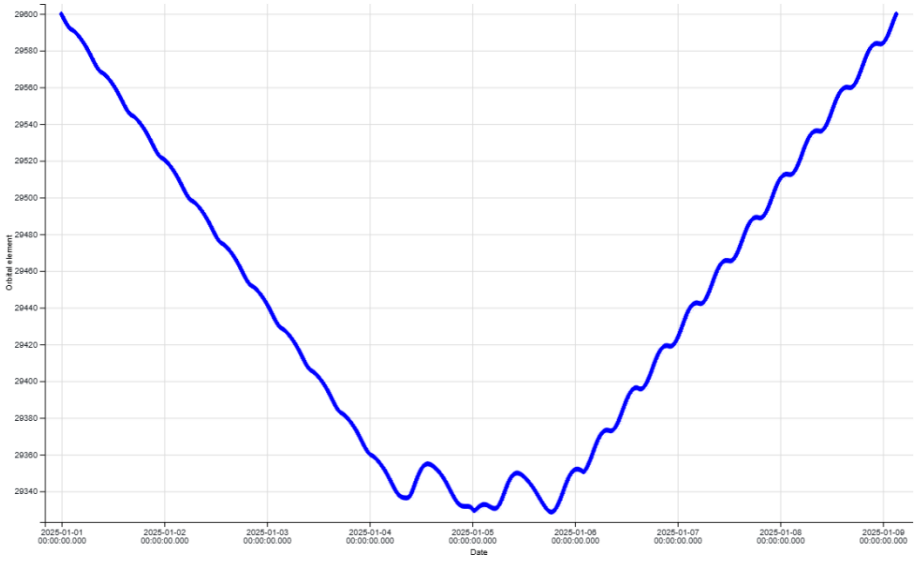

Fig. 9: Semi-major axis evolution along the relocation with Sun Illumination imposition

7.6. Case 6: Injection Ω Optimization

This case represents also a MEO orbit raising, but its goal is to find the optimal injection Ω_0 (see Table 6) with J2 perturbation. The spacecraft has a mass of 2000 kg, a thrust of 0.35 N, and Isp of 2000 s. Target Ω_f is defined following the same procedure than case 3.

Table 6: Initial, target and results for an injection Ω case

	x_0	x_f	OPEPOR $x_f - x_p$
a (km)	12000.0	29600.0	$-1.06 \cdot 10^{-2}$
e	0.0	0.0	$1.40 \cdot 10^{-5}$
i (deg)	56.0	56.0	$1.50 \cdot 10^{-4}$
Ω (deg)	To be optimized	330.0	$-1.00 \cdot 10^{-3}$
ω (deg)	0.0	-	-
L (deg)	0.0	40.0	-

The optimization loop process started from a non-optimal injection Ω_{0_0} of 150.0º. This first iteration reached a final Ω_p of 122.55º in 131.33 days, not reaching the target Ω_f by 155.95º. However, in the next iteration after applying equation (29), the $\Omega_f - \Omega_p$ is equal to $-1.00 \cdot 10^{-3}$º. So, the algorithm only needed two iterations to reach convergence, being the resulting optimal injection Ω_0 354.05º with a transfer time of 131.33 days (mass consumption of 202.67 kg).

Additionally, an analysis of advancing and delaying the injection epoch one week was performed, resulting in transfers of 133.07 days (mass consumption of 205.35 kg) and 133.02 days (205.28 kg), respectively.

8. Conclusions and Further Work

IGEOR and OPEPOR are both well-tested software components which allow for low-thrust manoeuvres optimizations in a wide range of scenarios. Each tool has its own characteristics, being IGEOR a simpler program that guarantees the global optimum, and OPEPOR a more flexible and powerful tool which requires an initial guess of the total time of flight to avoid local minimums. In deep, OPEPOR can accommodate the addition of new considerations to the optimal problem without major modifications to the algorithm. Moreover, OPEPOR can be easily re-used in more advanced analysis such as optimizing the launcher orbit, just by inserting it on a simple loop, showing a robust convergence rate.

In future work, GMV will continue working in improving the performance of this software components and the possibility to feed OPEPOR not only with boundaries for the time of flight, but also with costate information reducing the computational time and processes to robustly converge to the optimum.

References

[1] J. M. Sánchez Pérez and A. Campa, "Automation of multi-revolution low-thrust transfer optimization via differential evolution," *Advances in the Astronautical Sciences,* pp. vol. 152, p. 1579-1598, 2014.

[2] S. Geffroy and E. Richard, "Optimal low-thrust transfers with constraints---generalization of averaging techniques," *Acta astronautica,* pp. vol. 41, no 3, p. 133-149, 1997.

[3] Z. P. Olikara, "Framework for optimizing many-revolution low-thrust transfers," *AAS/AIAA Astrodynamics Specialist Conference, Paper 18–332,* 2018.

[4] T. Dargent, "Averaging technique in T-3D an integrated tool for continuous thrust optimal control in orbit transfers," in *Space Flight Mechanics Conference, Paper AAS-14-158,* Santa Fe, New Mexico, USA, 2014.

[5] B. A. Conway, "Spacecraft Trajectory Optimization," 2010.