Morante, David; Sanjurjo Rivo, Manuel; Soler, Manuel. (2019). Multi-Objective Low-Thrust Interplanetary Trajectory Optimization Based on Generalized Logarithmic Spirals. *Journal of Guidance, Control, and Dynamics*, 42(3), pp.: 476-490.

DOI: https://doi.org/10.2514/1.G003702

Multiobjective Low-Thrust Interplanetary Trajectory Optimization Based on Generalized Logarithmic Spirals

David Morante *, Manuel Sanjurjo Rivo† and Manuel Soler‡
Universidad Carlos III de Madrid, 28911, Leganés, Spain

The multi-objective optimal design of low-thrust multigravity-assist trajectories is formulated within the Hybrid Optimal Control framework. A new automated solution strategy for this problem is proposed in this work based on a two-step algorithm. In the first step, the trajectory is assumed to be a Generalized Logarithmic Spiral. A heuristic global search algorithm combined with nonlinear programming are in charge of optimizing the set of parameters defining the spirals, as well as the number, sequence and configuration of the gravity assists. In the second step, candidate solutions are regarded as initial guesses for a direct collocation method, where the problem is transcribed into a Nonlinear Programming Problem by discretization, considering the full dynamics and the complete set of constraints. The presented approach is tested on a rendezvous mission to the asteroid Ceres, allowing a Mars flyby, and on a flyby mission to Jupiter, allowing multiple flybys on different bodies. Pareto-optimal solutions in terms of time of flight and propellant mass consumed are obtained for both cases. Results outperform those found in the literature in terms of optimality while showing the effectiveness of the proposed methodology to generate quick performance estimates for preliminary studies and as accurate solutions for the detailed design.

I. Introduction

Low-thrust propulsion and gravity assists maneuvers are both well known to provide significant benefits in terms of required propellant mass. The former has been successfully demonstrated in remarkable missions such as NASA Deep Space 1 [1] or ESA SMART-1 [2], whereas the latter has been repeatedly used in interplanetary missions, perhaps most notably in the NASA Voyager 2 [3]. The coupling of high-specific-impulse low-thrust propulsion with gravity assists is a natural next step in the development of trajectory design techniques, opening new opportunities for the exploration of the solar system and beyond [4].

However, the propellant reduction achieved with low-thrust engines, when compared to their chemical counterparts, often comes at the cost of a higher transfer time. Therefore, the design of low-thrust multi-gravity assist trajectories

*PhD candidate, Bioengineering and Aerospace Engineering Department, Universidad Carlos III de Madrid, 28911, Leganés, Spain.
†Assistant Professor, Bioengineering and Aerospace Engineering Department, Universidad Carlos III de Madrid, 28911, Leganés, Spain.
‡Assistant Professor, Bioengineering and Aerospace Engineering Department, Universidad Carlos III de Madrid, 28911, Leganés, Spain.

is typically treated as a multi-objective optimization problem. The goal is then to efficiently determine the set of Pareto-Optimal trajectories for an interplanetary mission, regarding flight time and propellant mass consumed. The spacecraft will benefit from flybys of other planets, as well as from the continuous thrust provided by an electric engine.

The aforementioned problem, considering the general formulation based on the multi-body dynamics, can be potentially solved as an Optimal Control Problem (OCP). Authors like Whiffen [5] and Olympio [6] applied optimal control techniques to optimize trajectories in the multi-body environment automatically finding gravity assists. Although this approach provide high fidelity trajectories, it has been proven in practice to be a difficult and time-consuming task, even with robust optimizers, due to the different length and time scales in the dynamics.

In order to reduce the dynamical complexity of the problem, the patched conic approximation is often assumed, splitting the trajectory into a sequence of two-body problems. Additionally, flybys can be considered as instantaneous changes in the heliocentric velocity of the spacecraft [7]. Consequently, the dynamics of the spacecraft is defined by a sequence of discrete events, defining the flybys, and continuous dynamics, describing the heliocentric trajectory. Note that this sequence is not known a priori and it has to be obtained as part of the solution. This interconnection between discrete and continuous dynamics allows to formally pose the optimization problem as a Hybrid Optimal Control Problem (HOCP). General frameworks for the description of HOCPs and its corresponding mathematical formalism are presented by Branicky [8] and Buss [9]. Particular frameworks for space mission planning have been proposed by Chilan and Conway [10] and Ross and D'Souza [11].

A simplified class of hybrid problems, called multiphase optimal control problems (MOCP), ~~have~~ has been pursued quite vigorously since the 1960s. In this formulation, the sequence of planetary encounters is known a priori. These problems can be regarded as OCPs with interior point constraints and therefore optimal control techniques can be also applied. For instance, Coverstone et al. [12] used a multi-objective Genetic Algorithm (GA) to choose initial guesses for the Solar Electric Propulsion Trajectory Optimization Program (SEPTOP) developed by Sauer [13], which implements an indirect transcription technique with a NLP solver. Later, Vavrina and Howell [14] used a GA to choose initial values for their Gravity Assisted Low-Thrust Local Optimization Program (GALLOP) [15], which combines the Sims-Flanagan direct transcription method [16] along with a gradient-based NLP solver. Most recently, Yam, Di Lorenzo, and Izzo [17] used monotonic basin hopping (MBH) along with the Sims-Flanagan transcription to quickly and reliably solve continuous-thrust problems. Although all three of these methods have been demonstrated on problems involving multiple flybys as well as continuous-thrust propulsion, they do not optimize the flyby sequence.

The problem of determining the optimal sequence of flybys together with the optimal low-thrust steering law is a formidable challenge due to the intrinsic combinatorial complexity in addition to the nonlinearity and non-convexity of the continuous dynamics. This implies that the problem exhibits many local optima, which makes the detection of the global ones more complicated. A straightforward approach to find globally optimal trajectories is to solve all possible combinations of available flybys as different MOCPs and to identify the best ones therefrom. Although intuition on

the part of the mission planner could reduce the size of the search space, the computational time is prohibitive. Thus, pre-pruning with respect to some criteria can be applied to reduce the discrete search space; hence, only the remaining mission sequences are considered of interest. However, pre-pruning may not account for solutions that are nonintuitive. Therefore, an algorithm that intelligently searches among all possibilities, such as a heuristic method, is desired instead.

Based on the previous consideration, several researchers have developed automated tools that solve the trajectory optimization problem with no a-priori information about the flyby sequence. Most of them have focused on missions involving only impulsive maneuvers, such as Gad and Abdelkhalik [18] and Englander et al. [19]. An automated solution for low-thrust propulsion has been addressed by Englander and Conway [20] for single-objective problems. In their approach they combine two nested optimization algorithms. The outer loop uses a GA to select the flyby sequence while the inner loop solves the corresponding MOCP using MBH global search algorithm with the Sims-Flanagan transcription method resulting in a medium-fidelity algorithm, ~~albeit computationally expensive~~. This methodology was later extended to include multi-objective problems by Englander et. al [21] and spacecraft system optimization by Vavrina et. al [22] .

~~Consequently, all of the preceding approaches are unsuitable for the conceptual assessment of a large number of solutions, such as those usually required in the preliminary design phases. Alternatively, mission designers seek for near-optimal and low-fidelity solutions that could serve a twofold purpose: they provide rapid, broad overviews of the candidate trajectories design space and they provide a sufficiently good seed for more sophisticated optimizers~~.

Previous approaches using Sims-Flanagan transcription are used regularly for conceptual assessment in preliminary design. Computational times for such tools range from several hours to days [20, 21], depending on the complexity of the problem. Alternatively, faster assessments at lower fidelity are desirable. They could serve a twofold purpose: they provide rapid, broad overviews of the candidate trajectories design space and they provide a sufficiently good seed for more sophisticated optimizers. The traditional method is to use a global search algorithm to find optimal low-fidelity approximations of the low-thrust trajectory, such as ballistic arcs or shape-based representations.

Ballistic trajectories do not guarantee to represent the best low-thrust trajectory, whereas shape-based approaches have been traditionally applied to efficiently generate near-optimal solutions by several researchers. Shape-based models assume an analytical expression for the trajectory and the required thrust profile to generate such trajectory is obtained therefrom. Examples include Bacon [23] and Tsu [24], who introduced the logarithmic spiral; Petropoulos and Longuski [25], who used the exponential sinusoid; Wall and Conway [26], who modeled the trajectory as an inverse polynomial; De Pascale and Vasile [27] and Novak and Vasile [28] , who created three-dimensional shape-based models incorporating pseudoequinoctial elements and spherical coordinates respectively and Taheri and Abdelkhalik [29] used a finite Fourier series to parametrized the trajectory. Similarly, Roa et al. [30] found an entire new family of Generalized Logarithmic Spirals based on the thrust profile of the logarithmic spirals, whose versatility was later improved by adding an additional degree of freedom in the solution [31].

In [25] Petropoulos and Longuski apply a broad search algorithm with pruning criteria along with exponential sinusoids to generate candidate trajectories for the NLP-based GALLOP [15]. However, the cost estimated by exponential sinusoid methods does not properly estimate the optimal value as neither coasting nor rendezvous phases can be included in the model. Vasile et al. [32] study the optimality of the exponential sinusoid and concludes that this model is far from satisfying the necessary condition of optimality. Recently, Roa et al. opted in [33] to use his shaped-based method together with a branch and prune algorithm for the direct exploration of the search space to generate as many candidate trajectories as possible as well as initial guesses for the Sims-Flanagan transcription method.

~~The goal of this paper is to contribute to the field of research on preliminary design methodologies by providing an approach that overcomes some of the limits of the previous approaches. More precisely, the aim is to develop a fast and robust optimization algorithm that can, without a priori knowledge, compute the number of gravity-assists and the planets to flyby in addition to the low-thrust steering law that are optimal for a given interplanetary mission.~~

The goal of this paper is to contribute to the field of research on preliminary design methodologies of low-thrust, gravity-assist trajectories. More precisely, the aim is to reduce the computational times of the existing medium-fidelity algorithms [21, 22] by incorporating a low-fidelity model as in [25, 33], albeit maintaining the multi-objective character of the problem and the automatic selection of flyby bodies .

In this work we propose a sequential two-step algorithm termed MOLTO-IT (Multi-Objective Low-Thrust Optimizer for Interplanetary Trajectories). ~~In Step 1, we parametrize the trajectory from one planet to another, imposing a predefined thrust-coast sequence where the thrusting arcs follow a Controlled Generalized Logarithmic Spiral.~~ In Step 1, simplifications such as coplanar bodies and no enforced propulsion constraints along with a shape-based parameterization of the low-thrust trajectory [31] are assumed. Two nested optimization loops are combined: the outer loop implements a multi-objective heuristic algorithm to search over the flyby sequences and the launch and flyby dates; the inner loop solves a NLP for each interplanetary leg between planets, obtaining the optimal spiral * parameters as well as the optimal flybys configuration variables. This approach allows for a quick preliminary sub-optimal representation of the solution with an implicit satisfaction of the dynamics and boundary conditions on states. In Step 2, non-dominated solutions from the previous step are locally optimized by solving the resulting MOCPs with a direct transcription approach along with a NLP solver, improving fidelity and incorporating constraints . The main contribution of this work lies on the combination of the logarithmic spirals for approximating the low-thrust arcs along with a global-local multi-objective hybrid scheme algorithm which clearly becomes a robust and flexible tool able to efficiently provide candidate missions with better performances than those found in the literature.

The paper is organized as follows. Firstly, the interplanetary trajectory optimization problem with unknown flybys is formally stated as a HOCP in Section II. Secondly, in Section III we introduce the selected models for both the

*In this work the family of Controlled Generalized Logarithmic Spirals will be termed as logarithmic spirals or simply spirals

low-thrust trajectory and the gravity assists maneuvers used for the Step1 and Step 2. Then, in Section IV the two-step solution methodology is explained in detail. Later, in Section V the algorithm capabilities are tested in two different mission scenarios: a rendezvous mission to Ceres and a flyby mission to Jupiter. Finally, some conclusions and perspectives are discussed in Section VI.

II. Mathematical Framework

The trajectory of a spacecraft in the Solar System can be determined by the solution of the resulting multi-body problem. This approach is useful to take into account the effect of escape and capture phases as well as the high-fidelity modeling of flybys and low-energy transfers. Nevertheless, this results in a hypersensitive dynamical system, extremely difficult and time-consuming to tackle with current optimization tools.

When multi-body effects are not significant, sufficiently accurate solutions can be obtained by applying the patched conic approximation, which decomposes the problem into a series of restricted two-body problems; i.e. the trajectory changes from being heliocentric to planetocentric when the spacecraft enters the sphere of influence of a particular planetary body. An additional approximation assumes that the radius of this sphere is infinitesimal and consequently the gravity assist maneuver occurs instantaneously, resulting in a much more tractable problem. As a first approach this is not unreasonable because the flyby certainly does occur rapidly in comparison to the many years required for the complete interplanetary trajectory and the radius of the sphere of influence is negligible compared to the total distance travelled by the spacecraft [34]. Therefore, the flyby can be modeled as a discontinuous change in the heliocentric velocity when the spacecraft encounters a planet.

Additionally, a spacecraft equipped with an electric engine ~~have~~ has two different working conditions: thrusting and coasting, which corresponds to the discrete description of the system. The spacecraft can switch between both modes of operation due to power system requirements or due to a control decision. Therefore, the spacecraft dynamics has to be modeled as a Hybrid Dynamical System, as it combines both continuous and discrete time evolution and both continuous and discrete state variables. The continuous dynamics determines the heliocentric interplanetary trajectory whereas the discrete dynamics characterizes the gravity assists and the engine on/off switchings. In this section, we present a general mathematical framework for Hybrid Optimal Control problems based on the one proposed by Buss in [9]. In Section III, this framework will be particularized for the optimization of low-thrust interplanetary trajectories

A. Hybrid Dynamical System

The state of a hybrid dynamical system is determined by the continuous state vector $\boldsymbol{x}(t) \in \mathcal{X} \subset \mathbb{R}^{n_x}$, which is constrained to be in the set $\mathcal{X}$ of permissible continuous states, and by the discrete state vector $\boldsymbol{q}(t)^{\dagger} \in \mathcal{Q} \subset \mathbb{N}^{n_q}$, which is constrained to be in the set $\mathcal{Q}$ of permissible discrete states. The system can be controlled by a continuous input

[†]The discrete state vector can represent also a categorical variable

vector $\boldsymbol{u}(t) \in \mathcal{U} \subset \mathbb{R}^{n_u}$, which belongs to the set $\mathcal{U}$ of permissible continuous controls, and by a discrete input vector $\boldsymbol{v}(t) \in \mathcal{V} \subset \mathbb{N}^{n_v}$, which belongs to the set $\mathcal{V}$ of permissible discrete controls. Both input vectors [‡] can be dynamical variables or static parameters depending on whether they are time-varying or time-independent respectively. Therefore, the evolution of the state vector with respect to the independent time variable $t \in \mathbb{R}$ is given by its hybrid dynamics as follows [§]:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t) \qquad \text{if} \quad s_j(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t) \neq 0, \qquad j = 1, \ldots, n_s \tag{1}$$

$$[\boldsymbol{x}(t_i^+), \boldsymbol{q}(t_i^+)] = \phi_j(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t_i^-) \qquad \text{if} \quad s_j(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t_i^-) = 0, \qquad j \in \{1, \ldots, n_s\} \tag{2}$$

The continuous behavior of the hybrid dynamical system is described by the set of differentiable equation $f : \mathcal{X} \times \mathcal{Q} \times \mathcal{U} \times \mathcal{V} \times \mathbb{R} \longrightarrow \mathbb{R}^{n_x}$, whereas the discontinuous behavior is characterized by the set of discrete event functions, which includes the n_s discontinuity surfaces $s_j : \mathcal{X} \times \mathcal{Q} \times \mathcal{U} \times \mathcal{V} \times \mathbb{R} \longrightarrow \mathbb{R}$ and transition map functions $\phi_j : \mathcal{X} \times \mathcal{Q} \times \mathcal{U} \times \mathcal{V} \times \mathbb{R} \longrightarrow \mathcal{X} \times \mathcal{Q}$ for $j = 1, \ldots, n_s$. ~~The discrete events ensue when the state or controls intersects any of the discontinuity surfaces~~. Discontinuity surfaces pose the conditions that both state and controls must satisfy for a discrete event to ensue. The times t_i at which these events occur, are called event transition times. In case the discontinuous surface depends only on the state vector, it represents an autonomous event, whereas if it depends uniquely on the controls, it defines a controlled event. The successor states $\boldsymbol{x}(t_i^+)$ and $\boldsymbol{q}(t_i^+)$ just after a discrete event is given by the transition map functions. As long as all discontinuity surfaces $s_j(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t) \neq 0$ for $j = 1, \ldots, n_s$, the system trajectory evolves continuously according to Eq.(1). As an illustration, the discontinuity surface associated to a discrete flyby event represents the condition of intercepting the planet to flyby, whereas the transition map function determines the change in the heliocentric velocity during the flyby maneuver.

Let us defined the event sequence σ as the time-ordered sequence of N active discrete events, as follows:

$$\sigma = [(t_0, s_0), (t_1, s_1), \ldots, (t_N, s_N)] \tag{3}$$

where the pair (t_i, s_i) indicates that at the event transition time t_i the discrete event function s_i is intersected. Note that, in general, the number N of active discrete events will be different from the number n_s of available discrete event functions.

[‡] Input vectors can be also termed as control vectors, control inputs, control variables, controls or decision variables

[§] Explicit time-dependency of state and control variables will be usually omitted in this work for purpose of better readability of the text. Hence it holds that $\boldsymbol{x}(t) = x$, $\boldsymbol{q}(t) = q$ and $\boldsymbol{u}(t) = u$

B. Hybrid Optimal Control Problem

The HOCP is to find the continuous $\boldsymbol{u}(t)$ and discrete $\boldsymbol{v}(t)$ control inputs such that a multi-objective function $\boldsymbol{J}(\boldsymbol{u}, \boldsymbol{v}, t)$, typically a vector function of the hybrid system state and control, is minimized:

$$\min \boldsymbol{J}(\boldsymbol{u}, \boldsymbol{v}, t) = \Theta\left[\boldsymbol{x}(t_0^-), \boldsymbol{x}(t_0^+), \ldots, \boldsymbol{x}(t_N^-), \boldsymbol{x}(t_N^+); t_0, \ldots t_N\right] + \int_{t_0}^{t_f} \psi(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t) dt \tag{4}$$

subject to

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t) \quad \text{if} \quad s_j(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t) \neq 0, \quad j = 1, \ldots, n_s \tag{5}$$

$$[\boldsymbol{x}(t_i^+), \boldsymbol{q}(t_i^+)] = \phi_j(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t_i^-) \quad \text{if} \quad s_j(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t_i^-) = 0, \quad j \in \{1, \ldots, n_s\} \tag{6}$$

$$\boldsymbol{u}(t) \in \mathcal{U} \subset \mathbb{R}^{n_u}, \quad \boldsymbol{v}(t) \in \mathcal{V} \subset \mathbb{N}^{n_v}, \quad \forall t \in \left[t_0, t_f\right] \tag{7}$$

$$\boldsymbol{x}(t) \in \mathcal{X} \subset \mathbb{R}^{n_x}, \quad \boldsymbol{q}(t) \in \mathcal{Q} \subset \mathbb{N}^{n_q}, \quad \forall t \in \left[t_0, t_f\right] \tag{8}$$

$$0 \leq g(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t), \quad t \in \left[t_0, t_f\right] \tag{9}$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t_0), \qquad \boldsymbol{q}(t_0) = \boldsymbol{q}_0(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t_0) \tag{10}$$

$$\boldsymbol{x}(t_f) = \boldsymbol{x}_f(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t_f), \qquad \boldsymbol{q}(t_f) = \boldsymbol{q}_f(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{u}, \boldsymbol{v}, t_f) \tag{11}$$

The objective function J involves n_o single objective functions given as $J(\boldsymbol{u}, \boldsymbol{v}, t) = [J_1(\boldsymbol{u}, \boldsymbol{v}, t), \ldots, J_{n_o}(\boldsymbol{u}, \boldsymbol{v}, t)]$. In the above, the integrand term $\psi : \mathcal{X} \times \mathcal{U} \times \mathcal{V} \times \mathbb{R} \longrightarrow \mathbb{R}^{n_o}$ of the multi-objective function $\boldsymbol{J}$ is a vector real-valued function of the state and control variables and of time. The Mayer type part $\Theta : \mathcal{X} \times \mathbb{R} \longrightarrow \mathbb{R}^{n_o}$ is a general vector function of the event transition times t_i for $i = 0, \ldots, N$ and of the continuous states $\boldsymbol{x}(t_i^+)$, $\boldsymbol{x}(t_i^-)$ just before and just after the discrete events. Here, t_0 and $t_N = t_f$ are the beginning and final times, which are associated to an initial and final event function respectively, whereas the remaining $N - 1$ transition times are related to interior event functions.

The minimization of the multi-objective function in Eq.(4) is subject to initial and terminal conditions on the state vector (10)-(11), admissible values for the continuous and discrete control and state variables (7)-(8) and further inequality constraints (9) given by the function $g : \mathcal{X} \times \mathcal{U} \times \mathcal{V} \times \mathbb{R} \longrightarrow \mathbb{R}^{n_g}$. Obviously, valid hybrid optimal trajectories must obey both the continuous and discrete dynamics (5)-(6).

The key challenge, when solving HOCP, is that the optimal event sequence σ is not known and therefore it has to be determined when solving the problem. Additionally, when facing multi-objective problems, instead of searching for a unique optimal law for the continuos and discrete control inputs as in single objective optimization, the aim is to obtain a whole set of different solutions that are equally optimal in terms of Pareto efficiency.

III. Modelling

In this section we describe the models required to formally pose the problem under consideration, i.e. the optimal design of low-thrust interplanetary trajectories as a Hybrid Optimal Control Problem. We present the set of continuous differential equations along with the discrete event functions and transition maps governing the evolution of the continuous and discrete state vector of the spacecraft. The interconnection between them is depicted in the following diagram in Figure 1.

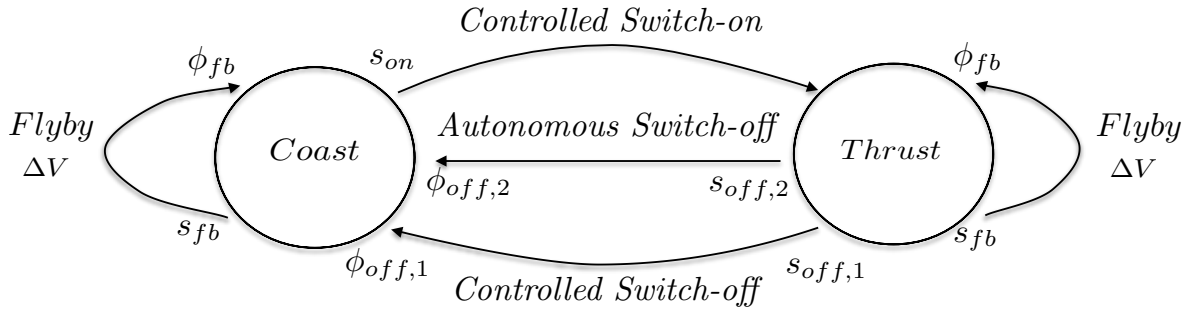

Figure 1 Diagram for a Spacecraft Hybrid Dynamical System

The solution to the problem will involve searching over a wide range of parameters, including mission departure date, selection of the appropriate combination of flybys, arrival dates at each of the bodies, and even the continuous and discrete control time-profile. ~~The general form of the problem is very complex to handle and hardly any solution can be found or, at least, in a reasonable amount of computation time~~ The general high-fidelity form including all the operational constraints of the problem is very complex to handle in a reasonable amount of computation time. Hence, simplifications in the dynamics and/or in the constraints are normally assumed to tackle the problem. In this work a simpler ~~and planar~~ representation of the problem will be deduced from the complete models. In particular, the motion will be assumed planar and the trajectory will be parametrized as a Generalized Logarithmic Spiral. Note that the planar assumption is a good first approximation for trajectories within the solar system planets (with the exception of Pluto), where the inclination change is not significant. However, it may not be valid for trajectories to highly inclined targets. The use of Generalized Logarithmic Spirals reduces the complexity of determining the trajectory at the cost of not restricting the thrust to a specific propulsion system. Therefore, the trajectory may be or may not be realizable by the propulsion system under consideration.

A. Continuous and Discrete State and Control Variables

The continuous state $\boldsymbol{x} = [\mathbf{r}, \mathbf{v}, m]$ of the spacecraft is determined by its position $\mathbf{r}(t) \in \mathbb{R}^3$ and velocity $\mathbf{v}(t) \in \mathbb{R}^3$ vectors with respect to an inertial reference frame, which, without loss of generality, is centered in the Sun. Besides, the evolution of the spacecraft mass $m(t)$ is needed to fully determine its dynamics. The categorical state variable $q(t) \in \{thrust, coast\}$ determines the working condition of the electric engine. When thrusting, the engine can only

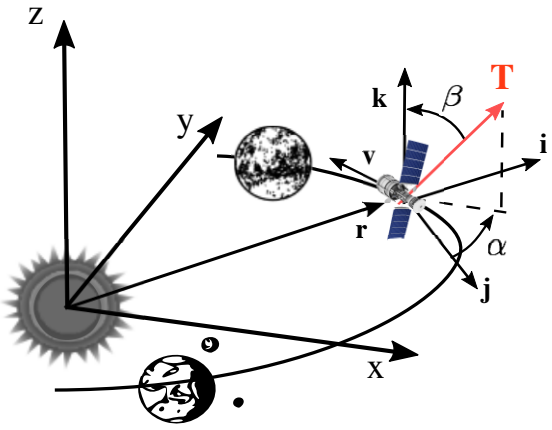

Figure 2 Interplanetary Transfer

operate at its maximum available thrust level T and its orientation can be controlled by the dynamical continuous input vector $\boldsymbol{u} = [\alpha, \beta]$, where α and β represent the azimuth and declination angles respectively. The steering law is measured with respect to the orbital reference frame defined by the following unitary vectors:

$$\mathbf{i} = \frac{\mathbf{r}}{\mathrm{r}}, \qquad \mathbf{k} = \frac{\mathbf{r} \times \mathbf{v}}{\mathrm{rv}}, \qquad \mathbf{j} = \mathbf{k} \times \mathbf{i} \tag{12}$$

where $\mathrm{r} = ||\mathbf{r}||$ is the radial distance and $\mathrm{v} = ||\mathbf{v}||$ is the magnitude of the velocity. Figure 2 depicts the geometry of the problem. Additionally, the switch between modes of operation can be managed by the binary control input $v(t) \in \{0, 1\}$. The coasting state is required when $\mathrm{v} = 0$, while the thrusting state is required with $\mathrm{v} = 1$ as long as the power system requirements are satisfied.

B. Continuous Dynamics

The evolution of the continuous state of the spacecraft is modeled as a particle moving in a central gravity field under the action of a perturbing acceleration $\boldsymbol{a}_p$. The set of differential equations are described as follows:

$$\boldsymbol{f}: \begin{cases} \mathbf{v} = -\dfrac{\mu_s}{\mathrm{r}^3}\mathbf{r} + \boldsymbol{a}_p(\boldsymbol{x}, q, \boldsymbol{u}, t) \\ \mathbf{r} = \mathbf{v} \\ \dot{m} = \dot{m}(\boldsymbol{x}, q, \boldsymbol{u}, t) \end{cases} \tag{13}$$

where μ_s represents the gravitational constant of the Sun. The perturbing acceleration $\boldsymbol{a}_p$ and the mass flow rate $\dot{m}$ have different expressions depending on the discrete state of the spacecraft. Assuming that the thrust is the only external

force:

$$\boldsymbol{a}_p, \dot{m} : \begin{cases} \boldsymbol{a}_p = \dfrac{T}{m}\,\mathbf{d}, & \dot{m} = -\dfrac{T}{g_0 I_{sp}} \quad \text{if} \quad q = thrust \\ \boldsymbol{a}_p = 0, & \dot{m} = 0 \quad \text{if} \quad q = coast \end{cases} \tag{14}$$

where I_{sp} is the specific impulse and g_0 is the gravitational acceleration at sea level. The unitary vector $\mathbf{d}$ points toward the direction of the thrust vector determined by $\boldsymbol{u}$. The magnitude T of the thrust generated by an electric engine and its specific impulse I_{sp} can be modeled as constants or can incorporate polynomials functions based on the power available to the spacecraft, which depends on the type of electric propulsion system under consideration. Note that, when coasting, the solution to the differential system in Eq.13 corresponds to a Keplerian arc.

C. Discrete Dynamics

The discrete dynamics of the spacecraft allows to characterize the effect of both performing flyby maneuvers and turning the engine on and off. In the following lines we define the set of discrete event functions, including discontinuity surfaces and transition map functions, necessary to include such effects in the model.

1. Flybys

Let us define the continuous state vector of a planet b_j as $\boldsymbol{x}_{b,j}(t) = [\mathbf{r}_{b,j}, \mathbf{v}_{b,j}]$, where $\mathbf{r}_{b,j}(t) \in \mathbb{R}^3$ and $\mathbf{v}_{b,j}(t) \in \mathbb{R}^3$ represent its position and velocity heliocentric vectors respectively. Flybys are assumed to produce an instantaneous change in the heliocentric velocity of the spacecraft, given by the transition map $\phi_{fb,j}$ and occurring when the position vector of the spacecraft intersects the discontinuity surface s_{fb}:

$$\text{Flybys} : \begin{cases} \phi_{fb,j} : \mathbf{v}(t_i^+) = \boldsymbol{\Delta}_j(\mathbf{v}(t_i^-), \boldsymbol{p}_j, \mathbf{v}_{b,j}(t_i^-), \mu_{b,j}, t_i^-), \quad q(t_i^+) = q(t_i^-), \quad \boldsymbol{r}(t_i^+) = \boldsymbol{r}(t_i^-), \quad m(t_i^+) = m(t_i^-) \\ s_{fb,j} : ||\mathbf{r} - \mathbf{r}_{b,j}(t)|| = 0, \quad j \in \{1, \ldots, n_s\} \end{cases} \tag{15}$$

where $\mu_{b,j}$ is the gravitational constant of the corresponding planet b_j. The spacecraft heliocentric position vector and mass are assumed not to change during the flyby maneuver as powered flybys are not considered. This is a reasonable assumption as powered flybys have never been implemented on real missions.

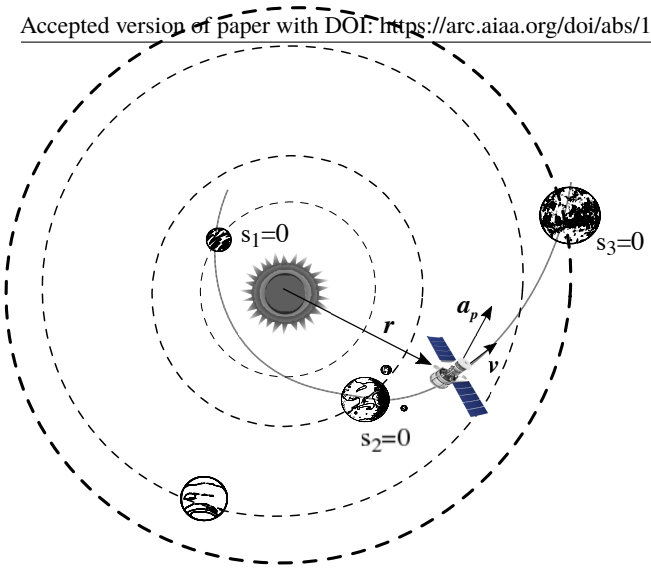

Figure 3 Interplanetary Transfer

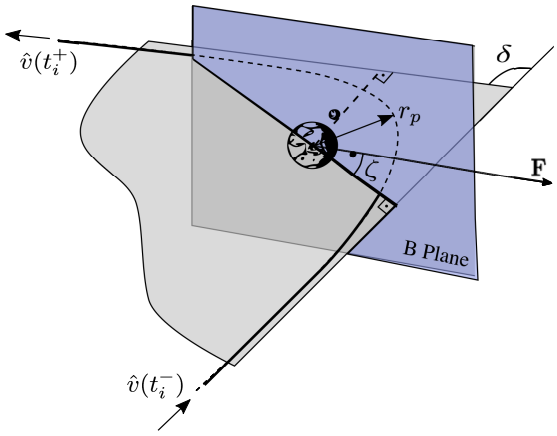

Figure 4 Flyby geometry

Following the aforementioned approach, if a planet b_j is encountered at t_i, the heliocentric post-flyby velocity $\mathbf{v}(t_i^+)$ can be obtained assuming a hyperbolic trajectory around the planet, which is a function of the pre-flyby velocity $\mathbf{v}(t_i^-)$, the planet heliocentric velocity $\mathbf{v}_{b,j}(t_i^-)$ and additional static control variables or parameters $\boldsymbol{p}_j = [r_{p,j}, \zeta_j]$, which are subject to optimization. The additional parameters are the minimum distance of approach $r_{p,j}$ and the B-Plane angle ζ.

$$\boldsymbol{\Delta}_j(\mathbf{v}(t_i^-), \boldsymbol{p}_j, \mathbf{v}_{b,j}(t_i^-), \mu_{b,j}, t_i^-) : \begin{cases} \hat{\mathbf{v}}(t_i^-) = \mathbf{v}(t_i^-) - \mathbf{v}_{b,j}(t_i^-) \\ \delta = 2\arcsin(1/(1 + r_{p,j}\hat{\mathrm{v}}^2(\mathrm{t_i^-})/\mu_{\mathrm{b,j}})) \\ \hat{\mathbf{v}}(t_i^+) = \cos\delta\hat{\mathbf{i}} + \cos\zeta\sin\delta\hat{\mathbf{j}} + \sin\zeta\sin\delta\hat{\mathbf{k}} \\ \mathbf{v}(t_i^+) = \mathbf{v}_{b,j}(t_i) + \hat{\mathbf{v}}(t_i^+) \end{cases} \tag{16}$$

where the unit vectors are

$$\hat{\mathbf{i}} = \frac{\hat{\mathbf{v}}(t_i^-)}{\hat{\mathrm{v}}(\mathrm{t_i^-})}, \quad \hat{\mathbf{j}} = \frac{\hat{\mathbf{i}} \times \mathbf{v}_{b,j}(t_i^-)}{||\hat{\mathbf{i}} \times \mathbf{v}_{b,j}(t_i^-)||}, \quad \hat{\mathbf{k}} = \hat{\mathbf{i}} \times \hat{\mathbf{j}} \tag{17}$$

Here $\hat{\mathbf{v}}(t_i^-)$ is the spacecraft incoming relative velocity to the planet, which rotates an angle δ on the orbital plane of the hyperbola to scape as $\hat{\mathbf{v}}(t_i^+)$. The orbital plane is determined by the angle ζ measured on the B-plane with respect to the vector **F**, which is contained on the B-plane and is parallel to ecliptic plane (see Fig. 4). The B-plane is defined as the plane passing through the center of the planet and normal to the arrival asymptote $\hat{\mathbf{v}}(t_i^-)$.

Let us define $\sigma_{fb,i}$ as the time-ordered sequence of flyby events:

$$\sigma_{fb} = [\ldots, (t_i, s_{fb,i}), \ldots] \tag{18}$$

that has to be determined as part of the solution.

2. Engine on-off switching

The switch between the thrust/coast modes of operation can be described by a controlled event or by an autonomous event. The former occurs as a consequence of a controlled decision, for propellant savings reasons, whereas the latter occurs as a consequence of a constraint deriving from other subsystems requirements, e.g. when there is not enough power available for the engine to operate. Both are summarized in the following functions:

$$\text{Switching-on}: \begin{cases} \phi_{on}: q(t_i^+) = 1, \quad \mathbf{v}(t_i^+) = \mathbf{v}(t_i^-), \quad \mathbf{r}(t_i^+) = \mathbf{r}(t_i^-), \quad m(t_i^+) = m(t_i^-) \\ s_{on}: q(t_i^-) = 0, \quad v(t_i^-) = 1, \quad 0 \leq g(\boldsymbol{x}, q, \boldsymbol{u}, \boldsymbol{v}, t_i^-) \end{cases} \tag{19}$$

$$\text{Switching-off}: \begin{cases} \phi_{off}: q(t_i^+) = 0, \quad \mathbf{v}(t_i^+) = \mathbf{v}(t_i^-), \quad \mathbf{r}(t_i^+) = \mathbf{r}(t_i^-), \quad m(t_i^+) = m(t_i^-) \\ s_{off,1}: q(t_i^-) = 1, \quad v(t_i^-) = 0 \\ s_{off,2}: q(t_i^-) = 1, \quad 0 \geq g(\boldsymbol{x}, q, \boldsymbol{u}, \boldsymbol{v}, t_i^-) \end{cases} \tag{20}$$

Here the event surface s_{on} refers to the controlled switching-on whereas $s_{off,1}$ and $s_{off,2}$ represents the event surface for the controlled and autonomous switching-off respectively. Also, the function g imposes the constraint related to the power system, i.e. when $g > 0$ there is not enough power available and the thruster cannot operate. Let us define σ_{sw} as the time-ordered sequence of the on-off switchings of the engine:

$$\sigma_{sw} = [\ldots, (t_i, s_{on/off}), \ldots] \tag{21}$$

that has to be determined as part of the solution.

3. Launch

The launch event is characterized by an instantaneous change in the initial heliocentric velocity that is performed by the launcher at the departure epoch.

$$\text{Launch}: \begin{cases} \phi_{la}: \mathbf{v}(t_0^+) = \Delta_{la}(\boldsymbol{x}_{b,1}, t_0^-), \quad \mathbf{r}(t_0^+) = \mathbf{r}(t_0^-), \quad m(t_0^+) = m(t_0^-), \quad q(t_0^+) = q(t_0^-) \\ s_{la}: t_0 - t_0^* = 0 \end{cases} \tag{22}$$

The velocity of the spacecraft after launch is given by:

$$\Delta_{la} = \mathbf{v}_{b,1}(t_0^-) + \mathbf{v}_{\infty,0} \tag{23}$$

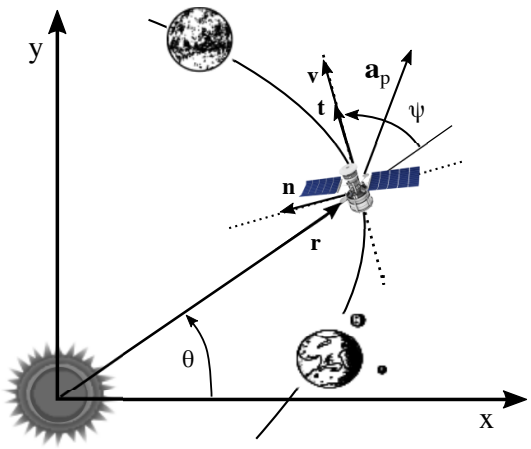

Figure 5 Geometry of the problem

where $\boldsymbol{v}_{\infty,0}$ is the departure hyperbolic excess velocity. Thus, the spacecraft matches the position of the departure planet and its initial velocity is the sum of the planet's velocity and the relative velocity at the defined initial epoch t_0^*.

D. Parametric Dynamical Model: Controlled Generalized Logarithmic Spirals

Although the solution to the HOCP using the model previously described can potentially lead to the optimal solution, it is hard and time-consuming to tackle. Providing that the flyby sequence is not obvious and that the problem is not highly constrained, the most challenging aspect relies on the correct identification of the optimal flyby sequences σ_{fb}. Hence, we propose a parametric model for the continuous and discrete control profiles. Additionally, trajectories for both the spacecraft and planets are assumed to be planar and lie on the same plane as well as no constraints on the power available to the spacecraft are considered.

When thrusting the spacecraft acceleration and orientation is assumed to follow the one required for a generalized logarithmic spiral. They have been thoroughly studied and described in previous works [30, 31], being the result to the continuous dynamics in Eq.(13) when the perturbing acceleration $\mathbf{a}_p$ is assumed to have the following form:

$$\mathbf{a}_p = \frac{\mu_s}{r^2}(\xi \cos\psi\ \mathbf{t} + (1-2\xi)\sin\psi\ \mathbf{n}) \tag{24}$$

where ξ is a control parameter and ψ is the flight path angle. The control parameter ξ can be adjusted as required, and it affects both the magnitude and direction of the thrust vector. The acceleration is projected onto an intrinsic reference frame, which is defined by the following unitary vectors:

$$\mathbf{t} = \frac{\mathbf{v}}{\mathrm{v}}, \qquad \mathbf{b} = \frac{\mathbf{r}\times\mathbf{v}}{\mathrm{rv}}, \qquad \mathbf{n} = \mathbf{b}\times\mathbf{t} \tag{25}$$

where the vectors $\mathbf{t}$ and $\mathbf{n}$ are directed along the tangential and normal direction respectively and are contained in the

Table 1 Trajectory parametrization summary

Family	Trajectory
Elliptic	$r(\theta) = \dfrac{1}{c_1 + c_2 \cosh \beta(\theta)}$
Parabolic	$r(\theta) = c_3 e^{(c_4\theta - 1)}$
Hyperbolic I	$r(\theta) = \dfrac{1}{\sinh \dfrac{\beta(\theta)}{2}(c_5 \sinh \dfrac{\beta(\theta)}{2} + c_6 \cosh \dfrac{\beta(\theta)}{2})}$
Hyperbolic II	$r(\theta) = \dfrac{1}{c_7 + c_8 \cos \beta(\theta)}$

orbital plane. Figure (5) depicts the geometry of the problem referred to the inertial reference frame.

The continuous dynamical equations (13) can be reformulated using a set of intrinsic coordinates $(\mathrm{r}, \mathrm{v}, \psi, \theta)$, which allow to derive two constants of motion K_1 and K_2. They are extensions of the laws of conservation of energy and angular momentum. In particular, they can be solved from the initial conditions $(r_0, \mathrm{v}_0, \psi_0)$:

$$K_1 = \mathrm{v}_0^2 - \frac{2\mu}{r_0}(1 - \xi) \tag{26}$$

$$K_2 = r_0 \mathrm{v}_0^2 \sin \psi_0 \tag{27}$$

Making use of K_1, K_2 and ξ, Roa [31] obtained closed-form analytical solutions for the trajectory and time, as a function of the polar angle θ, avoiding the need to explicitly or implicitly impose the constraints related to the continuous equation of motion. Depending on the sign value of the constant K_1, three families of solutions are obtained: elliptic ($K_1 < 0$), parabolic ($K_1 = 0$), and hyperbolic ($K_1 > 0$). There are two types of hyperbolic spirals: spirals of type I correspond to $K_2 < (2(1 - \xi))$, whereas spirals of type II satisfy $K_2 > (2(1 - \xi))$. The analytical solutions for the trajectory are summarized in Table 1 as a function of the constants $c_1, c_2, c_3, c_4, c_5, c_6, c_7$ and c_8 [31], which can be derived from the initial conditions and the control parameter ξ, and of the $\beta(\theta)$ that represents the spiral anomaly.

The state of the spacecraft can be redefined as $\boldsymbol{x} = [r, v, \psi, t]$ and expressed as a function of the polar angle θ given the control parameter ξ and the initial state $\boldsymbol{x}_0 = [r_0, v_0, \psi_0, t_0]$ at θ_0 as follows:

$$\boldsymbol{x}(\theta) = \boldsymbol{X}_T(\boldsymbol{x}_0(\theta_0), \xi; \theta) \tag{28}$$

where X_T is a closed-form analytical representation of the trajectory when thrusting. The complete analytical repre-

sentation can be found in [31]. Additionally, if the spacecraft is coasting it will follow a Keplerian arc and its state can be easily determined as a function of the polar angle given the state $\boldsymbol{x}_0$ when the coasting begins as follows:

$$\boldsymbol{x}(\theta) = \boldsymbol{X}_C(\boldsymbol{x}_0(\theta_0); \theta) \tag{29}$$

where X_c is a closed-form analytical representation of the trajectory when coasting. An estimate of the propellant consumption is made by assuming a constant specific impulse I_{sp} for the low-thrust engines. This simplification permits the required propellant mass to be expressed as a fraction of the initial spacecraft mass, based on the time integral of the thrust acceleration and the rocket equation.

In this work we predefine a thrust-coast sequence for an interplanetary leg where the target is to flyby the next planet. We assume that the spacecraft first traverses a generalized logarithmic spiral from θ_0 to an intermediate point θ_A, and then describes a Keplerian orbit from θ_A to θ_F, (see Fig. 6). Nevertheless, if the target is to rendezvous a planet the transfer leg will be decomposed in three arcs instead of just two (see Fig. 7): a generalized logarithmic spiral from θ_0 to θ_A, a coast arc from θ_A to θ_B, and a second spiral segment from θ_B to θ_F. In the case of multi-revolution trajectories, the proposed sequence will be repeated for each revolution. However, multi-revolution problems will not be considered in this work.

Regarding the flyby maneuvers, as the incoming and outgoing relative velocity before and after the flyby are contained in the same plane the flyby B-plane angle ζ_i can only take two values: -180 deg and 180 deg. Note that, in such case, the spacecraft heliocentric velocity will rotate an angle given by $+\delta$ or $-\delta$. Therefore, instead of determining the values $r_{p,j}$ and ζ_i the flyby ca be described by providing only the rotation angle $\delta \in (-\delta_{lim}, +\delta_{lim})$, where δ_{lim} is defined as the maximum deflection angle that will be obtained at the minimum $r_{p,j}$.

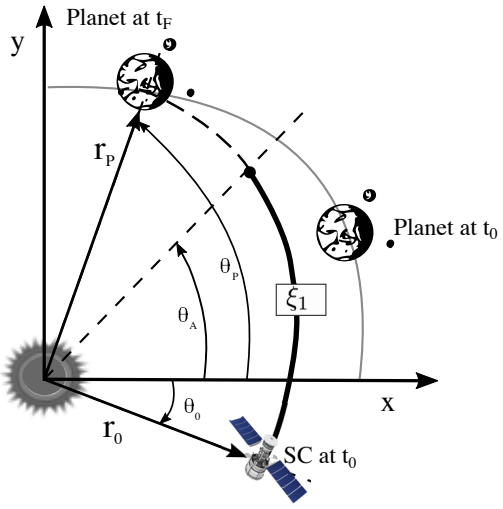

Figure 6 Flyby leg configuration

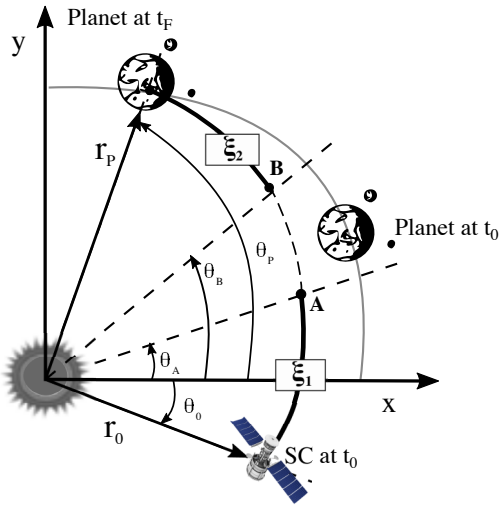

Figure 7 Rendezvous leg configuration

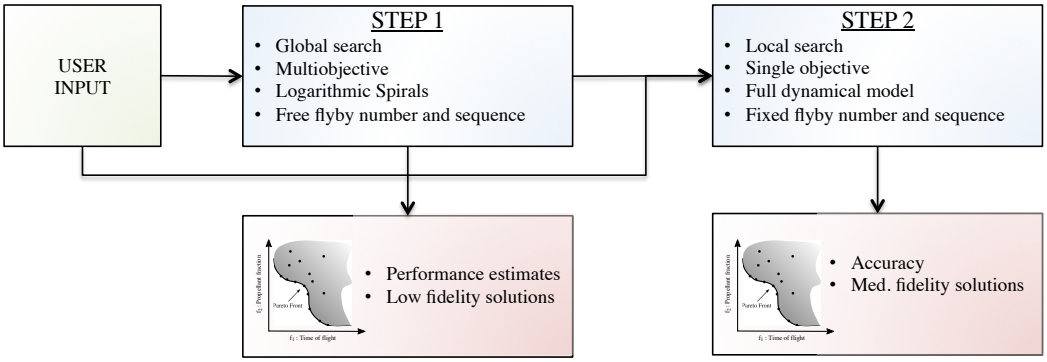

Figure 8 MOLTO-IT general scheme

IV. Solution Approach

In this work we present MOLTO-IT, which is a two.step solution approach for the problem under consideration. The algorithm is schematically depicted in Figure 8. In Step 1 we incorporate the parametric model of the trajectory, pursuing the goal of developing a flexible and robust algorithm able to rapidly find solutions, which would approximate the optimal performances as well as the mission design variables, needing minimum information from the user. In Step 2 we include the complete model of the trajectory seeking for accuracy and robustness in addition to the possibility of including complex constraints with ease. In the following lines we describe both steps in detail.

A. Step 1: Global Multi-objective Search

Making use of the Controlled Generalized Logarithmic Spirals and of the predefined thrusting sequence, the infinite dimensional Hybrid Optimal Control Problem is reduced to a mixed-integer parameter optimization problem. The real and integer nature of the variables and the requisite of evaluating many different scenarios simultaneously, make population-based heuristic algorithms, such as genetic algorithms or particle swarm optimization, the most adequate techniques to solve it. However, they are not well suited for handling the nonlinear constraints arising from both the flyby and rendezvous conditions. This kind of constraints can be tackled much more efficiently with gradient-based solvers. Consequently, two nested optimization algorithms combining an heuristic algorithm with a gradient-based solver will be used as it is schematically represented in Figure 9

The outer loop heuristic algorithm has two main assignments: on the one hand, it is in charge of optimizing the discrete design variables associated to the flyby sequence; on the other hand, it runs as an automatic initial guess generator for the gradient based solver in the inner loop. In particular, it provides initial guesses for the Launch Date and the transfer times for each interplanetary leg. The decision vector consists on the variables summarized in Table 2. Note that the number of variables is different depending on the number of flybys for each specific candidate trajectory. Thus, we devised a method using null values, which is called the 'null-gene' technique as introduced in [17] and [18]. When the population-based algorithm parses a decision vector, it will skip over the null values and construct a

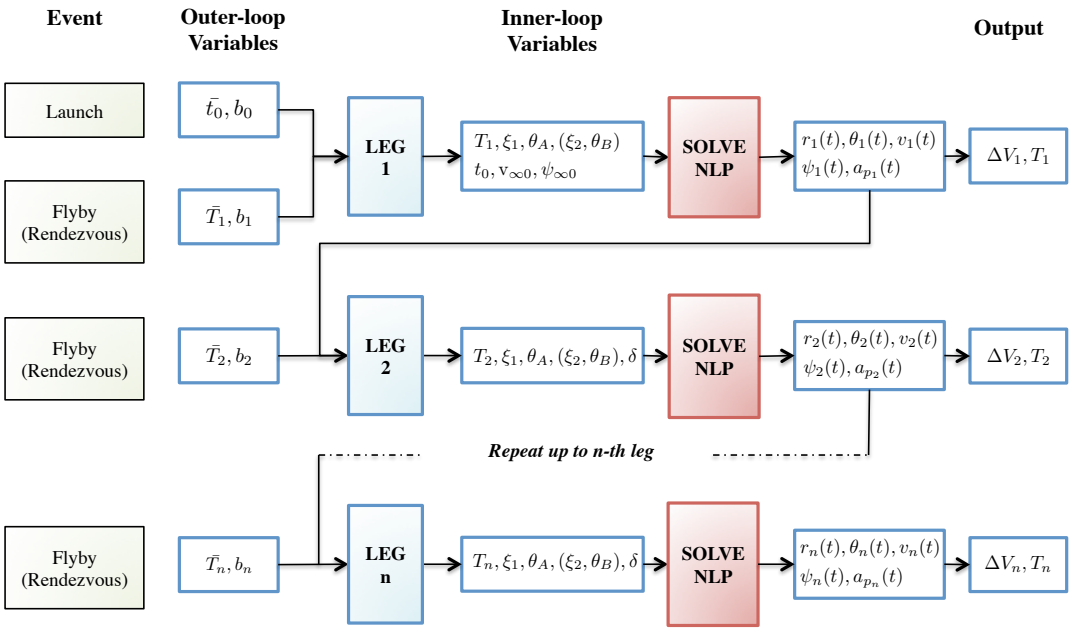

Figure 9 MOLTO-IT Step 1 algorithm scheme

Table 2 Heuristic Algorithm Variables

Variable	Meaning	Lower Bound	Upper Bound
$\tilde{t}_0$	Departure Epoch Guess	t_{0min}	t_{0max}
$\tilde{T}_i$	Leg Transfer Time Guess	T_{imin}	T_{imax}
b_i	Flyby Body	-	-

trajectory only from the values that represent planets. The maximum dimension of the decision vector is determined by the maximum number of allowed flybys.

Every population in the outer loop defines a different inner loop problem. The inner loop problem consists on a sequence of $n + 1$ interplanetary legs from departure to the target body, where n is the number of flybys. Each interplanetary leg is formulated as a real parameter optimization problem with nonlinear constraints (NLP) and solved with a gradient based method. The number of variables for the NLP and the number of constraints depends on the transfer leg under consideration. Let us define the initial and final event of a leg as *Event-1* and *Event-2* respectively. The NLP variables and its bounds are summarized in Table 4 as a function of the type of event at each of the end points: Launch (L), Flyby (F), Rendezvous (R) or All (-).

The Launch epoch and the Transfer leg Time are subjected to a lower and upper bounds centered in the value provided by the outer loop. The widths of such intervals,i.e. $2T_{lim}$ for the transfer time and $2t_{lim}$ for the launch date, are provided by the user. They have a twofold purpose: it improves the convergence of the solver and it maintains the diversity in the solutions, helping to produce a well-spread Pareto front. A good value for T_{lim} can be determined as a 10% of the expected transfer time. For instance, if the outer loop select a transfer time of 500 days, the inner loop will

Table 3 User Defined Parameters

Description	Variable
Departure Body	b_o
Arrival Body	b_f
Mission type	Flyby/Rendezvous
Launch Window	$[t_{0,min}, t_{0,max}]$
Max/Min Flyby num.	$[n_{fb,max}, n_{fb,min}]$
Flyby bodies	$[b_1, \ldots, b_n]$
Minimum Flyby radius	$r_{fb,min}$
Time of Flight	$[T_{i,min}, T_{i,max}]$
Max/Min Launch v_∞	$[v_{\infty 0,\min}, v_{\infty 0,\max}]$

search within 450 and 550 days. Similarly, a good value for t_{lim} consists on a 10% of the Launch window span. The larger the widths, the less spread the Pareto front, although each population has more chances to converge to a feasible solution.

Note that the angular variables θ_A and θ_B are defined as a fraction of the total travelled angle for each leg, in order to avoid variable limits. The value of the initial guess for $\psi_{\infty 0}$ is selected depending on the semi-major axis of the next body after launch., i.e. $\pi/2$ for outer planets, $-\pi/2$ for inner planets and 0 for resonant flybys. The initial guesses for the parameters defining the spiral arc consists on providing an almost coasting trajectory leg.

The general form of the NLP algorithm, regardless of the type of trajectory leg, is described in Alg. 1. Regarding the computation of the state of the different bodies at different epochs, constant orbital elements are assumed for each leg, given that accurate ephemerides are provided in tabular data which may not be differentiable and lead to non-convergence. The osculating orbital elements of each body are computed at the starting epoch of each leg using JPL NAIF-SPICE ephemerides and are deemed as constants during each leg. As the trajectories considered in this model are planar, the position and velocity of the target bodies are projected onto the ecliptic plane.

If all the interplanetary legs computed by the inner loop are feasible, the resulting cost, that is total ΔV and total Transfer Time ToF is returned to the outer loop. On the contrary, if an unfeasible trajectory is encountered, the inner loop stops, returning a penalty cost to the outer loop. The general form of the Step 1 algorithm is summarized in the Algorithm 2. For running the Step 1, the user only need to provide the data outline in Table 3. ~~Note that solutions obtained with this model are not associated to any specific propulsion system and they are expected to approximate the performances of a wide range of low-thrust engines. This fact is indeed useful for the preliminary design phase. Once the features of the spacecraft are fully determined the mission can be re-optimized in the next step.~~ Additionally, it should be noted that the global search algorithm is biased towards the solutions that the shape-based parameterization represents and they may not be realizable with the selected propulsion subsystem or due to operational constraints that are not considered at this Step.

Algorithm 1 Step 1 inner-loop Fitness and Constraint function for leg i

Input : $(t_0, v_{\infty 0}, \psi_{\infty 0}), \{\delta\}, T_i, \xi_1, \theta_A(\xi_2, \theta_B)$
Ouput : ΔV

▷ Obtain Initial Transfer State
if *Event-1* = Launch **then**
  Compute state of planet b_1: $\boldsymbol{x}_{b,1}(t_0)$
  Add launcher Impulse: $\boldsymbol{x}(\theta_0) = \Delta_{la}(\boldsymbol{x}_{b,1}, \mathrm{v}_{\infty 0}, \psi_{\infty 0})$ [Eq. 23]
else
  Obtain initial state $\boldsymbol{x}(\theta_0)$ from leg $i-1$
  Compute state of planet b_1: $\boldsymbol{x}_{b,1}(t_0)$
  Compute state after Flyby: $\boldsymbol{x}(\theta_0) = \Delta_{fb}(\boldsymbol{x}(\theta_0), \boldsymbol{x}_{b,1}, \delta)$ [Eq. 16]
end if

Compute final time: $t_F = t_0 + T_i$
Compute state of planet b_2: $\boldsymbol{x}_{b,2}(t_F)$
▷ Obtain Final Transfer State
if *Event-2* = Flyby **then**
  Compute thrust arc from θ_0 to θ_A: $\boldsymbol{x}(\theta) = \boldsymbol{X}_T(\boldsymbol{x}(\theta_0), \xi_1; \theta)$
  Compute coast arc from θ_A to θ_F: $\boldsymbol{x}(\theta) = \boldsymbol{X}_C(\boldsymbol{x}(\theta_A); \theta)$
else
  Compute thrust arc from θ_0 to θ_A: $\boldsymbol{x}(\theta) = \boldsymbol{X}_T((\boldsymbol{x}(\theta_0), \xi_1; \theta)$
  Compute coast arc from θ_A to θ_B: $\boldsymbol{x}(\theta) = \boldsymbol{X}_C((\boldsymbol{x}(\theta_A); \theta)$
  Compute thrust arc from θ_B to θ_F: $\boldsymbol{x}(\theta) = \boldsymbol{X}_T((\boldsymbol{x}(\theta_B), \xi_2; \theta)$
end if

▷ Compute Constraints
if *Event-2* = Flyby **then**
  $c_1 : r_{b,2}(t_F) - r(\theta_F) = 0$
  $c_2 : T_i - t(\theta_F) = 0$
else
  $c_1 : r_{b,2}(t_F) - r(\theta_F) = 0$
  $c_2 : T_i - t(\theta_F) = 0$
  $c_3 : \mathrm{v}_{\mathrm{b},2}(\mathrm{t_F}) - \mathrm{v}(\theta_\mathrm{F}) = 0$
  $c_4 : \psi_{b,2}(t_F) - \psi(\theta_F) = 0$
end if

▷ Compute Cost
Compute ΔV from spiral arc

Algorithm 2 Step 1 outer-loop Fitness Function

Input : $\bar{t_0}, \bar{b_0}, \bar{T_1}, \bar{b_1}, \ldots, \bar{T_n}, \bar{b_n}$ ▷ GA variables
Ouput : $\Delta V, ToF$ ▷ Objective function

Determine number of flybys n_{fb}
for $i = 0 : n_{fb} + 1$ **do** ▷ Solve i-th leg
  Departure Body = b_i
  Arrival Body = b_{i+1}
  Determine departure and arrival event ▷ Launch, Flyby or Rendezvous
  Solve NLP Problem (Algorithm 1)
  if *unfeasible* **then**
    break
  end if
end for
Compute Cost Function

Table 4 NLP Problem Variables

Variable	Meaning	Lower Bound	Upper Bound	Initial Guess	Event1	Event2
t_0	Departure Epoch	$\bar{t}_0 - t_{lim}$	$\bar{t}_0 + t_{lim}$	$\bar{t}_0$	L	-
$v_{\infty 0}$	Initial Excess Velocity Module	$v_{\infty 0,\min}$	$v_{\infty 0,\max}$	$v_{\infty 0,\max}$	L	-
$\psi_{\infty 0}$	Initial Excess Velocity Angle	$-\pi$	π	$\frac{\pi}{2}, -\frac{\pi}{2}, 0$	L	-
δ	Flyby Deflection Angle	-1	1	0	F	-
T	Leg Transfer Time	$\bar{T}_i - T_{lim}$	$\bar{T}_i + T_{lim}$	$\bar{T}_i$	-	-
ξ_1	Spiral Control Parameter	0	1	0.4	-	-
θ_A	Thrust to Coast switching angle	0	1	0.01	-	-
ξ_2	Spiral Control Parameter	0	1	0.4	-	R
θ_B	Coast to Thrust switching angle	0	1	0.99	-	R

B. Step2: Local Optimization

In this second step, we optimize the candidate trajectories obtained in the first step. The three-dimensional problem is then reduced to a multiphase optimal control problem and can be solved by using a direct collocation scheme. The scheme converts the trajectory optimization problem to a parameter optimization problem using direct transcription and then uses a nonlinear programming (NLP) problem solver. For that purpose, the Hermite-Simpson collocation scheme has been chosen, as it is one of the most popular and effective choices [35].

In this step, ~~the time of the mission is fixed and the objective is the minimization of the propellant consumed~~ only a single-objective function is allowed. Typically, either the time of flight or the propellant consumed. Further constraints can be added to the problem, such as limiting the spacecraft angular velocity or the direction of the launch $\mathbf{v}_{\infty,0}$ (because in a real launch the $\mathbf{v}_{\infty,0}$ is constrained by the inclination and ascending node of the low Earth parking orbit, the injection point, and the launch date), albeit they are not considered for the examples presented in this paper.

V. Results

The main goal of this section is to investigate how well the new MOLTO-IT algorithm presented in this work sequentially approximate the performances and the design variables of the optimal mission in terms of accuracy and computational cost. As representative missions to evaluate its capabilities, we present a rendezvous mission from Earth (E) to the asteroid Ceres (C), and a flyby mission from E to Jupiter (J).

We specifically chose these scenarios because they have been previously addressed in the literature, in particular by Petropoulos and Longuski [25], who also propose a two-step approach. Firstly, they use the Satellite Tour Design Program - Low Thrust, Gravity Assist (STOUR-LGTA) which models the trajectory as a exponential sinusoid and conducts a broad search of the design space to select good candidates. Then, previous solutions are provided as initial guess for the direct optimization based program GALLOP [36]. Thus, we compare and verify results from our Step 1 with solutions from STOUR-LGTA and solutions from our Step 2 with those ones obtained from GALLOP. Even

though Petropoulos and Longuski [25] optimize over a predefined sequence of gravity assists, in this work we allow the algorithm to automatically search for the optimal bodies to flyby. In order to facilitate proper comparisons we use in our MOLTO-IT Step 2 simulations the same model than Petropoulos and Longuski [25] used for GALLOP:

$$T = c_{T0} + c_{T1} P_a \tag{30}$$

$$\dot{m} = c_{m0} + c_{m1} P_a \tag{31}$$

$$P_a = \frac{P_0}{\mathrm{r}^2} \left(\frac{\gamma_0 + \gamma_1/\mathrm{r} + \gamma_2/\mathrm{r}^2}{1 + \gamma_3 \mathrm{r}} \right) \tag{32}$$

In particular: $c_{T0} = -1.9137$ N, $c_{T1} = 36.242$ N/kW, $c_{m0} = 0.47556$ kg s^{-1}, $c_{m1} = 0.90209$ kg s^{-1}/kW , $\gamma_0 = 1.1063$, $\gamma_1 = 0.1495$ AU, $\gamma_2 = -0.299$ AU^2, $\gamma_3 = -0.0432$ AU^{-1}. Additionally, the reference power P_0 is 10kW and the thruster is assumed to need at least 0.649kW to operate, while not being able to use power in excess of 2.6kW. These values were first introduced by Williams and Coverstone-Carroll [4] for recreating the propulsion system used on Deep Space 1. For the Step 1, a constant value of 3000s is chosen as specific impulse as Petropoulos and Longuski [25] in STOUR-LGTA

MOLTO-IT is fully coded in MATLAB and all examples were run in a Intel Core i7 (2,5GHz) computing system. In Step 1, we selected the NonDominated Sorting Genetic Algorithm NSGA-II [37] running in parallel as heuristic solver and the Sequential Quadratic Problem algorithm included in the Optimization Toolbox of MATLAB [38] as a NLP solver. In Step 2, we use the Interior Point solver Ipopt [39] to solve the resulting NLP problem after direct transcription of the corresponding MOCP. In both cases, forward numerical derivatives were applied to compute gradients of the objective and constraint functions of the pertinent optimization problem.

A. Earth-Ceres Rendezvous Mission

The first case is a low-thrust rendezvous mission to the asteroid Ceres (C), allowing both, direct and via Mars flyby trajectories. A search for optimal trajectories over the year 2003 with a mission duration between 200 and 1400 days is performed with our algorithm. Definition of the mission scenario is outlined in Table 5. This is a compelling example due to the challenge of matching the final state of the body to rendezvous when compared to the less restricted flyby case.

The problem is first solved applying the Step 1 algorithm and the genetic algorithm parameters are set to the values summarized in Table 6. Several runs of the algorithm were tested and equally Pareto-optimal solutions in terms of non-domination were obtained in all cases. All trajectories in the population by the 10^{th} generation were feasible. At 20^{th} generation most of the population lies along a distinct non-dominated front, whereas from generation 20 to 30 the front is progressively shifted to lower time of flights and propellant masses. After 30 generations members of

Table 5 EC Problem Definition

Description	Value
Launch Window	Jan 01, 2003 - Dec 31, 2003
Mission type	rendezvous
Launch v_∞	1.6 km/s
Flyby num.	[0,1]
Flyby Bodys	Mars
$r_{fb,min}$	200 km
ToF	[200 ,1400] days
Initial mass m_0	568 kg
I_{sp}	3000 s

Table 6 Genetic Algorithm Parameters

Population size	100
Max. Generations	50
Mutation Fraction	0.3
Crossover Fraction	0.3

the population were uniformly distributed along the optimal front as no improvement was observed thereafter. As an example, the final Pareto front obtained from one of the tests is presented in Figure 10. The average evaluation time for each individual was 0.46 s and for each generation was 6.5 s. The maximum number of generations was achieved after 5.4 min computing time (see Table 10).

Note that a well spread Pareto front is obtained covering most of the allowable mission time interval (solutions longer than 2.7 years are out of the Pareto front), leading to propellant mass fractions that range from 55% to 22% and comprising both direct and via Mars flybys trajectories. Thus, the user is provided a wide variety of trajectories and the possibility of exploring various criteria at once. Results reveal promising launch dates in early July 2003 for both direct and flyby transfer. The optimal Mars flyby date remains the same for all solutions and the Ceres time of encounter varies consequently with the mission flight time.

Although no constraint has been applied to the maximum thrust produced by the generalized spirals, the trajectories obtained in this example exhibit typical accelerations of low-thrust engines, except for trajectories with very short transfer arcs.

Table 7 EC Computational Time

Variable	Avg. CPU Time
Population	0.46 s
Generation	6.5 s
Total	5.4 min

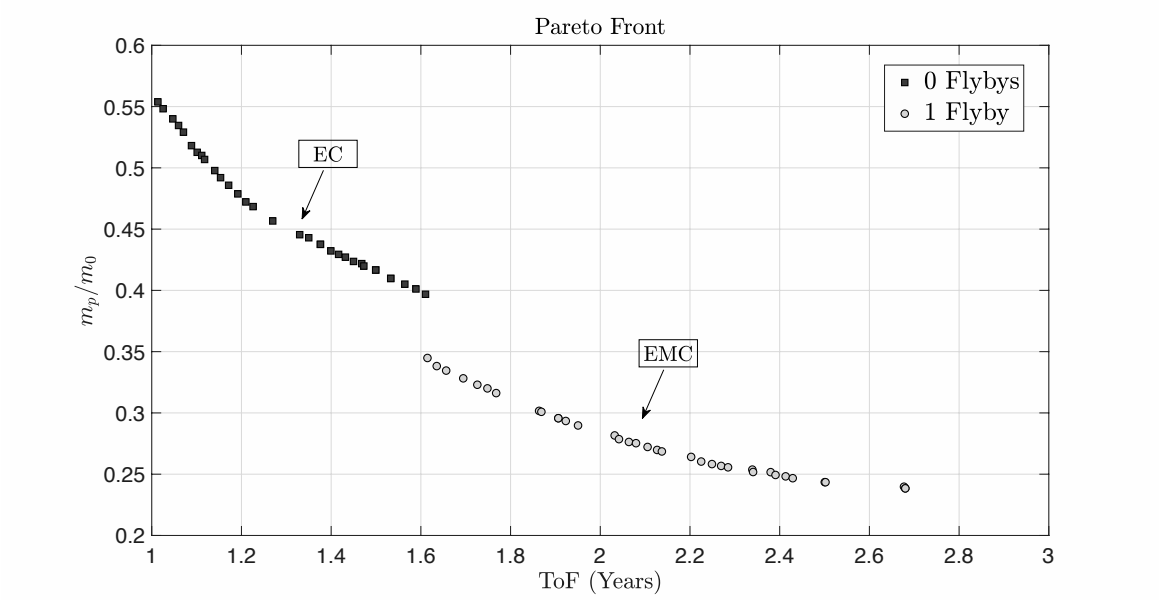

Figure 10 EC Pareto Front

The geometry of the minimum propellant trajectory provided by the Step 1 is shown in Fig. 11 where the dashed lines represent the coast arcs and the continuos lines are spirals arcs. This trajectory and its control profile were used as an initial estimate to run the Step 2 algorithm. A grid of 50 nodes uniformly distributed per leg were used, reaching convergence with a tolerance of 10^{-6} in 50 NLP-iterations after 3.2 min. Both results from Step 1 and Step 2 are compared in Table 8 with those ones obtained by Petropoulos and Longuski [25] using STOUR-LGTA and GALLOP.

It can be seen that the solution from MOLTO-IT Step 1 is better than the trajectory from STOUR-LGTA exhibiting a lower propellant mass fraction for a 143 days shorter transfer mission. Additionally, the STOUR-LGTA solution is dominated by the solutions from MOLTO-IT Step1 shown in the Pareto front in Fig.10. Note that the solution obtained with the exponential sinusoid does not fulfill the rendezvous condition and it would require an additional maneuver at arrival, whereas the solution with the generalized spirals does satisfy it. A similar situation occurs when comparing Step 2 and GALLOP, except for the flyby radius that reaches the minimum for both cases. Note that the propellant consumed in Step 2 is higher than in Step 1, which is attributable to the moderate inclination of the Asteroid Ceres (10 deg) that has to be achieved and is not taken into account with the generalized logarithmic spirals as they describe a planar motion.

B. Earth-Jupiter Flyby Mission

~~In this case we test the ability of the algorithm to search over long sequences of flybys~~. In this case, a search for optimal trajectories for a flyby mission to Jupiter during the launch years 2029 and 2030 is performed, allowing up to 3 flybys. The available planets to flyby are restricted to Mars, Venus and Earth. Table 9 contains the definition of this

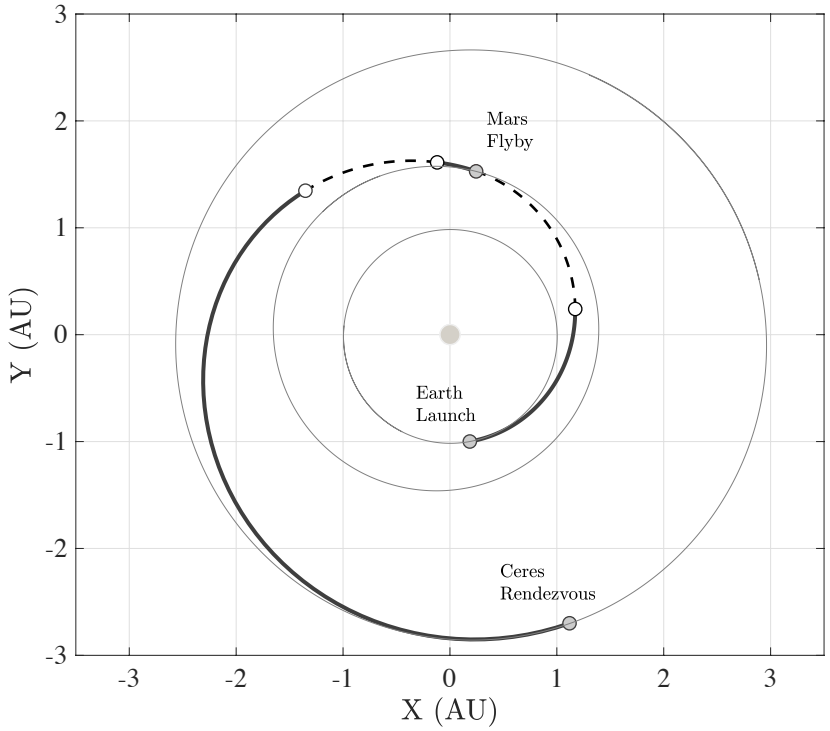

Figure 11 EMC Minimum Fuel trajectory

Table 8 EMC Minimum propellant solution

		MOLTO-IT		Petropoulos and Longuski [25]	
Parameter	Units	Step 1	Step 2	STOUR-LGTA	GALLOP
Launch Date	-	Jul 2, 2003	May 13, 2003	May 6, 2003	May 6, 2003
Launch v_∞	km/s	1.6	1.6	1.6	1.6
Mars Flyby Date	-	Feb 03, 2004	Dec 31, 2003	Feb 01, 2004	Feb 01, 2004
Mars Flyby v_∞	km/s	2.40	2.16	1.43	1.92
Mars B-Plane angle	deg	55.0	10.0	2.3	82.3
Mars Flyby Altitude	km	200	200	5432	200
Arrival Date	-	Mar 18, 2006	Jan 26, 2006	Jun 12, 2006	Feb 09, 2006
Arrival v_∞	km/s	0	0	0.237	0
Propellant mass fraction	-	0.224	0.229	0.256	0.233
TOF EM	days	216	232	271	271
TOF MC	days	774	757	862	739
TOF Total	days	990	990	1133	1010

Table 9 EJ Problem Definition

Variable	Value
Launch Window	Jan 01, 2029 - Dec 31, 2030
Mission type	flyby
Launch v_∞	2 km/s
Flyby num.	[0,3]
Flyby Bodys	Mars, Venus, Earth
$r_{fb,min}$	200 km
ToF	[100 ,1500] days
Initial mass m_0	360 kg
I_{sp}	3000 s

Table 10 EC Computational Time

Variable	Avg. CPU Time
Population	0.16 s
Generation	13 s
Total	11 min

mission.

As before, the problem was first solved applying the Step 1 algorithm using the same parameters for the Genetic algorithm, except for the population number that was increased up to 150 individuals to account for the greater dimension of the design space. If a smaller population size were chosen, there would be insufficient diversity in the initial population and the algorithm may not efficiently explore the solution space. A larger population size would make the total time for the solver unnecessarily large. The mean execution time per generation was 13 s leading a total computational time of 11 min (see Table 10). Figure 12 illustrates the population of Pareto optimal trajectories obtained therefrom. It can be seen that the algorithm converged to three different flyby configurations: EVJ, EVEJ and EVEMJ for an increasing time of flight respectively. Results reveal a promising launch date in October 2029, regardless of the mission time and transfer type. The date of every flyby occurs at the same epoch, whereas the time of encounter with Jupiter varies for every trajectory consistent with the total time of flight.

In order to corroborate that the algorithm converge to the globally optimal sequences of flybys, the same mission was solved again fixing the number of flybys. Thus, four additional cases with 0, 1, 2 and 3 flybys were solved using a population of 50 individuals for each. Results are displayed in Figure 12 and it can be observed that the free flyby solution is the optimal envelope of the fixed flyby cases. The average computational time for performing a generation was 6 s and for the genetic algorithm to be completed was approximately 5 min. The total time needed to compute all the fixed flyby solutions is 20 min. Hence, the automatic algorithm is able to find the same solution in almost half of the computational time than the fixed flyby case.

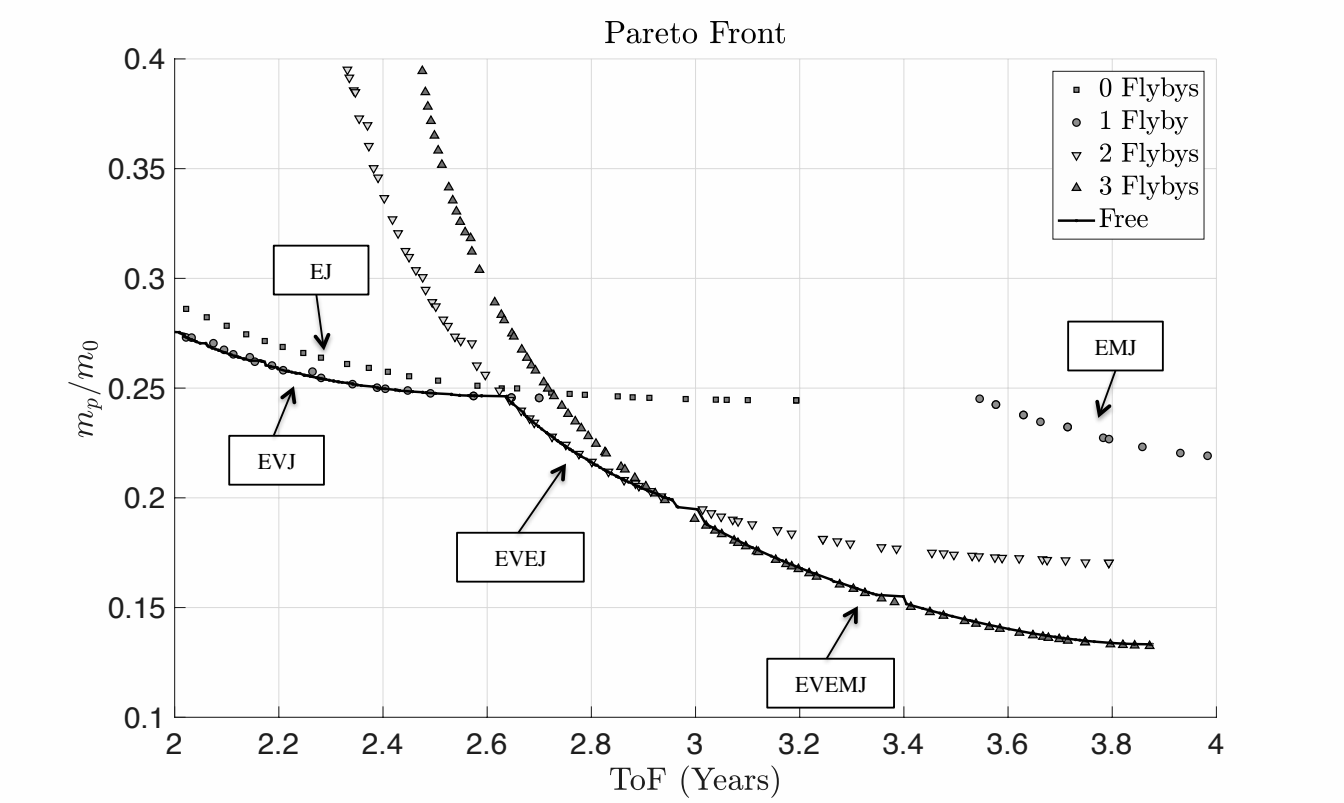

Figure 12 EJ Pareto front for fixed and free number of flybys

The next experiment follows the goal of evaluating how close the Pareto front from Step 2 is to the true optimal Pareto solution. Firstly, the minimum time solution previously obtained for each flyby configuration in Step 1 was used as initial guess to solve the resulting minimum time MOCP in Step 2. Then, the time interval ranging from the computed minimum time of flight to the maximum allowed mission time was divided into a 0.1 year sub-intervals. For each mission time determined by the grid and for each flyby configuration, a different minimum fuel problem were solved using as initial guess the closest candidate trajectory from Step 1 in terms of ToF. Although not optimal after Step 1, the direct transfer scenario was also included in this procedure. Finally, only the dominating solutions were selected and plotted in Figure 13 for comparisons against the Pareto front obtained in Step 1.

It is clear that the Step 2 Pareto is mainly below the one obtained in Step 1, because in this case the control is not constrained to follow any specific time-profile. The reduction in propellant mass achieved after Step 2 is around 40-20% relative to the one consumed by the spirals. Additionally, it is worth noting that the switching points from one flyby sequence to the other are well predicted in Step 1. However, the region corresponding to one flyby is not so well predicted by the spirals, because the shape-based result demands more thrust attainable by typical low-thrust, arriving faster at Venus, in particular, one period of Venus earlier. This result suggests that a productive direction of future research for our approach is to include constraints on the acceleration produced by the spirals.

It is clear from the previous comparisons that the shape-based solution provides a good starting point for the direct optimizer without any a-priori knowledge of the optimal solution and that performances obtained are close to the true optimal trajectories. This approximation works better for long flight times than for short flight times.

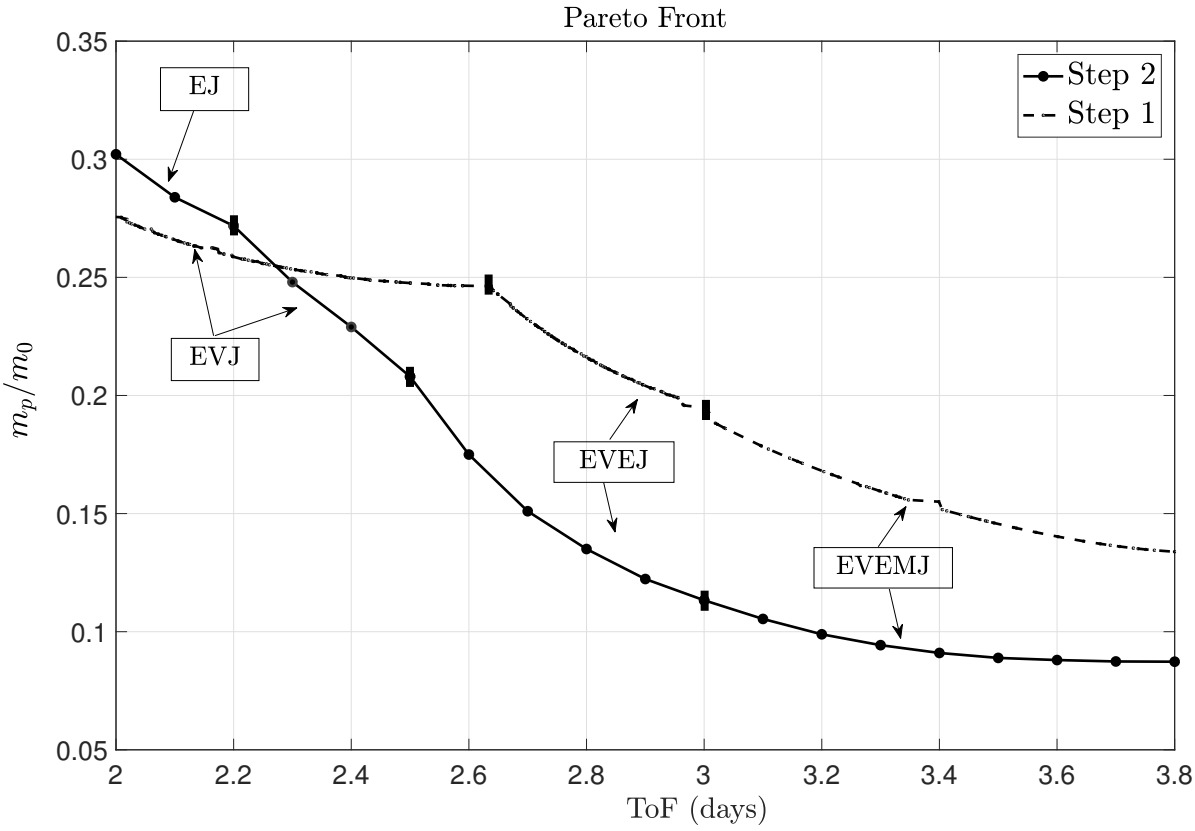

Figure 13 EJ paretos obtained after Step 1 and after Step 2

The optimal trajectory corresponding to the minimum fuel solution provided by the Step 1 is shown in Fig. 14 where the dashed lines represent the coast arcs and the continuos lines are spirals arcs. This trajectory and its control profile were used as an initial estimate to run the Step 2 algorithm. A grid of 50 nodes uniformly distributed per leg were used, reaching convergence with a tolerance of 10^{-6} in 110 NLP-iterations after 15 min. Both results from Step 1 and Step 2 are listed and compared in Table 11 with those ones obtained by Petropoulos and Longuski [25] using STOUR-LGTA and GALLOP, considering the same flyby sequence.

When the Step 1 and Step 2 solutions in Table 11 are compared, it can be noted that the B-Plane angles, flyby velocities and flybys altitudes are different, significantly in the case of Venus, meaning that Step 2 was able yo change these to account for the effect of the the out-of-plane. Additionally, It can be seen that the solution from MOLTO-IT Step 1 exhibits a lower propellant mass than STOUR-LGTA for a 552 days shorter transfer mission. Note that the average thrust acceleration for each leg predicted by the spirals is much lower than the one predicted by the exponential sinusoids

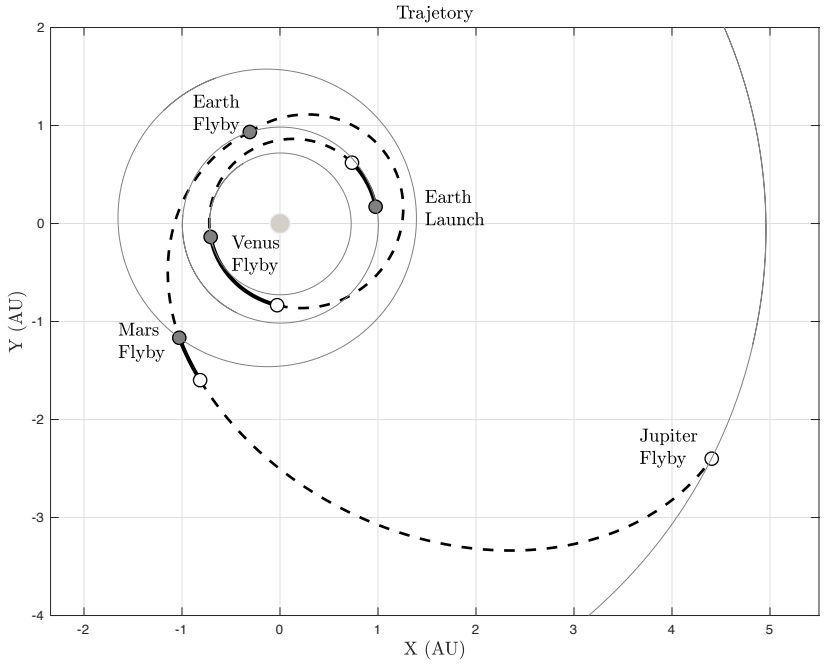

Figure 14 EMC Minimum Fuel trajectory

and that it includes a purely coasting leg from Venus to Mars. The optimized solutions from Step 2 presents also a lower propellant mass than the optimized solution from GALLOP. It is clear that the solution found by STOUR-LGTA correspond to a local minimum and therefore GALLOP could only converge in the neighborhood of this trajectory.

VI. Conclusions

In this work, the problem of interest is the automated solution of the Hybrid Optimal Control Problem resulting from the design of low-thrust multigravity assists interplanetary trajectories. We describe our tool MOLTO-IT (Multi-Objective Low-Thrust Optimizer for Interplanetary Trajectories) which is a fully automated algorithm based on a two-step procedure able to design interplanetary trajectories with low- thrust propulsion and complex flyby sequences, while not requiring any a priori information about the optimal sequence or trajectory.

Our algorithm is not only a valuable tool for preliminary mission planning but also for detailed mission. The Step 1 allows for a preliminary analysis as it provides the user in one run an after several minutes a set of solutions that closely estimates the optimal performances and mission design parameters. The outer-loop provides an intelligent search tool to find optimal flyby sequences whereas the inner-loop permits to quickly find feasible trajectories with low propellant masses. The Step 2 allows for a refinement of the trajectory, improving optimality and allowing for the inclusion of further constraints. As tested on this paper, the algorithm is able to successfully and efficiently solve challenging problems in astrodynamics such as rendezvous missions and missions involving long sequence of flybys, outperforming other methodologies available.

Table 11 EVEMJ Minimum propellant solution

		MOLTO-IT		Petropoulos and Longuski [25]	
Parameter	Units	Step 1	Step 2	STOUR-LGTA	GALLOP
Launch Date	-	Oct 1, 2029	Sept 28, 2029	Sept 3, 2029	Sept 3, 2029
Launch v_∞ (km/s)	km/s	2	2	2	2
Initial mass	kg	N/A	N/A	360	360
Venus Flyby Date	-	Feb 22, 2030	Mar 19, 2030	Feb 15, 2030	Feb 15, 2030
Venus Flyby v_∞	km/s	2.59	5.15	3.64	3.77
Venus Flyby Altitude	km	119,985	15,720	6,533	30,000
Venus B-Plane angle	deg	180.0	-69.6	178.3	60.7
Earth Flyby Date	-	Jan 04, 2031	Jan 11, 2031	Jan 15, 2031	Dec 30, 2030
Earth Flyby v_∞	km/s	8.33	8.32	6.50	5.18
Earth Flyby Altitude	km	201	200	655	1,035
Earth B-Plane angle	deg	180.0	-175.5	-180.0	-176.5
Mars Flyby Date	-	May 09, 2031	May 13, 2031	May 26, 2031	May 26, 2031
Mars Flyby v_∞	km/s	15.21	15.43	13.70	11.26
Mars Flyby Altitude	km	200	200	200	200
Mars B-Plane angle	deg	0.0	-9.0	-1.8	-5.5
Arrival Date	-	Aug 14, 2033	Aug 21, 2033	Jan 20, 2035	Jan 20, 2035
Arrival v_∞ (km/s)	km/s	5.65	5.62	5.85	6.25
Average thrust acceleration, EV	mm/s^2	0.03	0.10	0.12	N/A
Average thrust acceleration, VE	mm/s^2	0.15	0.01	0.16	N/A
Average thrust acceleration, VM	mm/s^2	0.00	0 .00	0.10	N/A
Average thrust acceleration, MJ	mm/s^2	0.09	0.02	0.14	N/A
Propellant mass fraction	-	0.132	0.088	0.294	0.256
TOF EV	days	144	172	165	165
TOF VE	days	316	298	334	318
TOF VM	days	125	122	131	147
TOF MJ	days	828	831	1,335	1,335
TOF Total	days	1,413	1413	1,965	1,965

It can be seen in the rendezvous mission to Ceres, that the solution from our Step 1 exhibits a 12% lower propellant mass fraction than STOUR-LGTA (Satellite Tour Design Program - Low Thrust, Gravity Assist) for a 143 days shorter transfer mission whereas Step 2 reduces 1.7% the propellant mass compared to GALLOP (Gravity Assisted Low-Thrust Local Optimization Program) for a 20 days shorter mission. Similarly, in the mission to Jupiter, solutions from MOLTO-IT Step 1 exhibits a 55% lower propellant mass than STOUR-LGTA for a 552 days shorter transfer mission whereas Step 2 reduces 65% the propellant mass of GALLOP. Although no data are available to compare the computing times, it is clear that our algorithm produces solutions in a reasonable short time of the order of 10 min. For both cases, a well spread Pareto front is obtained at once, where only the optimal sequences are considered.

Acknowledgments

This work has been partially supported by Spanish Government through grant TRA2014−58413−C2−2−R. The project has been funded under RD&I actions of Programa Estatal de Investigación, Desarrollo e Innovación Orientada a los Retos de la Sociedad (call 2014). The authors thank Javier Roa and Jesús Peláez for value support for using Generalized Logarithmic Spirals.

References

[1] Rayman, M. D., Varghese, P., Lehman, D. H., and Livesay, L. L., "Results from the deep space 1 technology validation mission," *Acta Astronautica*, Vol. 47, No. 2-9, 2000, pp. 475–487.

[2] Racca, G., Marini, A., Stagnaro, L., Van Dooren, J., Di Napoli, L., Foing, B., Lumb, R., Volp, J., Brinkmann, J., Grünagel, R., et al., "SMART-1 mission description and development status," *Planetary and space science*, Vol. 50, No. 14-15, 2002, pp. 1323–1337.

[3] Murrill, M. B., "The grandest tour-Voyager," *Mercury*, Vol. 22, 1993, pp. 66–77.

[4] Williams, S. N., and Coverstone-Carroll, V., "Benefits of solar electric propulsion for the next generation of planetary exploration missions," *Journal of the Astronautical Sciences*, Vol. 45, No. 2, 1997, pp. 143–159.

[5] Whiffen, G., "Mystic: Implementation of the Static Dynamic Optimal Control Algorithm for High-Fidelity, Low-Thrust Trajectory Design," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, American Institute of Aeronautics and Astronautics, 2006. doi:10.2514/6.2006-6741.

[6] Olympio, J., "Algorithm for Low-Thrust Optimal Interplanetary Transfers with Escape and Capture Phases," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, American Institute of Aeronautics and Astronautics, 2008. doi:10.2514/6.2008-7363.

[7] Olympio, J. T., "Optimisation and Optimal Control Methods for Planet Sequence Design of Low-Thrust Interplanetary Transfer Problems with Gravity Assists," Ph.D. thesis, Ecole des Mines de Paris, October 2008.

[8] Branicky, M. S., Borkar, V. S., and Mitter, S. K., "A unified framework for hybrid control: Model and optimal control theory," *IEEE transactions on automatic control*, Vol. 43, No. 1, 1998, pp. 31–45. doi:10.1109/9.654885.

[9] Buss, M., Glocker, M., Hardt, M., von Stryk, O., Bulirsch, R., and Schmidt, G., *Nonlinear Hybrid Dynamical Systems: Modeling, Optimal Control, and Applications*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 311–335. doi: 10.1007/3-540-45426-8_18.

[10] Chilan, C. M., and Conway, B. A., "Automated Design of Multiphase Space Missions Using Hybrid Optimal Control," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 5, 2013, pp. 1410–1424. doi:10.2514/1.58766.

[11] Ross, I. M., and D'Souza, C. N., "Hybrid Optimal Control Framework for Mission Planning," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005, pp. 686–697. doi:10.2514/1.8285.

[12] Coverstone-Carroll, V., Hartmann, J., and Mason, W., "Optimal multi-objective low-thrust spacecraft trajectories," *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, No. 2, 2000, pp. 387 – 402. doi:10.1016/S0045-7825(99)00393-X.

[13] Sauer, J., C. G., "Optimization of multiple target electric propulsion trajectories," *11th Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, Washington, D.C., 1973, p. 11.

[14] Vavrina, M., and Howell, K., "Multiobjective Optimization of Low-Thrust Trajectories Using a Genetic Algorithm Hybrid," *AAS/AIAA Space Flight Mechanics Meeting*, American Astronautical Society Paper 09- 151, Savannah, GA, 2009.

[15] McConaghy, T. T., Debban, T. J., Petropoulos, A. E., and Longuski, J. M., "Design and Optimization of Low-Thrust Trajectories with Gravity Assists," *Journal of Spacecraft and Rockets*, Vol. 40, No. 3, 2003, pp. 380–387. doi:10.2514/2.3973.

[16] Sims, J. A., and Flanagan, S. N., "Preliminary Design of Low-Thrust Interplanetary Missions," *Astrodynamics Specialist Conference*, American Astronautical Society Paper 99-338, Girdwood, Alaska, 1999.

[17] Yam, C. H., Lorenzo, D. D., and Izzo, D., "Low-thrust trajectory design as a constrained global optimization problem," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 225, No. 11, 2011, pp. 1243–1251. doi:10.1177/0954410011401686.

[18] Gad, A., and Abdelkhalik, O., "Hidden Genes Genetic Algorithm for Multi-Gravity-Assist Trajectories Optimization," *Journal of Spacecraft and Rockets*, Vol. 48, No. 4, 2011, pp. 629–641. doi:10.2514/1.52642.

[19] Englander, J. A., Conway, B. A., and Williams, T., "Automated Mission Planning via Evolutionary Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 6, 2012, pp. 1878–1887. doi:10.2514/1.54101.

[20] Englander, J. A., and Conway, B. A., "Automated Solution of the Low-Thrust Interplanetary Trajectory Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 1, 2016, pp. 15–27. doi:10.2514/1.G002124.

[21] Englander, J. A., Vavrina, M. A., and Ghosh, A. R., "Multi-Objective Hybrid Optimal Control for Multiple-Flyby Low-Thrust Mission Design," *AAS/AIAA Space Flight Mechanics Meeting,*, AAS Paper 15-227, 2015.

[22] Vavrina, M. A., Englander, J. A., and Ghosh, A. R., "Coupled Low-thrust Trajectory and System Optimization via Multi-Objective Hybrid Optimal Control," *AAS/AIAA Space Flight Mechanics Meeting*, AAS paper 15-397, 2015.

[23] Bacon, R. H., "Logarithmic spiral: an ideal trajectory for an interplanetary vehicle with engines of low sustained thrust," *American Journal of Physics*, Vol. 27, 1959, pp. 12–18. doi:10.1119/1.1934788.

[24] Tsu, T. C., "Interplanetary Travel by Solar Sail," *ARS Journal*, Vol. 29, No. 6, 1959, pp. 422–427. doi:10.2514/8.4791.

[25] Petropoulos, A. E., and Longuski, J. M., "Shape-based Algorithm for Automated Design of Low-Thrust, Gravity-Assist Trajectories," *Journal of Spacecraft and Rockets*, Vol. 41, No. 5, 2004, pp. 787–796. doi:10.2514/1.13095.

[26] Wall, B. J., and Conway, B. A., "Shape-Based Approach to Low-Thrust Rendezvous Trajectory Design," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 95–101. doi:10.2514/1.36848.

[27] Pascale, P. D., and Vasile, M., "Preliminary Design of Low-Thrust Multiple Gravity-Assist Trajectories," *Journal of Spacecraft and Rockets*, Vol. 43, No. 5, 2006, pp. 1065–1076. doi:10.2514/1.19646.

[28] Novak, D. M., and Vasile, M., "Improved Shaping Approach to the Preliminary Design of Low-Thrust Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 1, 2011, pp. 128–147. doi:10.2514/1.50434.

[29] Abdelkhalik, O., and Taheri, E., "Shape Based Approximation of Constrained Low-Thrust Space Trajectories using Fourier Series," *Journal of Spacecraft and Rockets*, Vol. 49, No. 3, 2012, pp. 535–546. doi:10.2514/1.58789.

[30] Roa, J., Peláez, J., and Senent, J., "New Analytic Solution with Continuous Thrust: Generalized Logarithmic Spirals," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 10, 2016, pp. 2336–2351. doi:10.2514/1.G000341.

[31] Roa, J., and Peláez, J., "Introducing a degree of freedom in the family of generalized logarithmic spirals," *26th Spaceflight Mechanics Meeting*, American Astronautical Society Paper 16-317, Springfield, VA, 2015.

[32] Vasile, M., Schütze, O., and Junge, O., "Spiral trajectories in global optimization of interplanetary and orbital transfers," Tech. Rep. ARIADNA AO05/4106, ESA, 2005.

[33] Roa, J., Petropoulos, A. E., and Park, R. S., "Semi-Analytic Preliminary Design of Low-Thrust Missions," *Astrodynamics Specialist Conference*, American Astronautical Society Paper 17-623, Stevenson, WA, 2017.

[34] Conway, B. A., *Spacecraft trajectory optimization*, Vol. 29, Cambridge University Press, 2010.

[35] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207. doi:10.2514/2.4231.

[36] McConaghy, T. T., Debban, T. J., Petropoulos, A. E., and Longuski, J. M., "Design and Optimization of Low-Thrust Trajectories with Gravity Assists," *Journal of Spacecraft and Rockets*, Vol. 40, No. 3, 2003, pp. 380–387. doi:10.2514/2.3973.

[37] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 2002, pp. 182–197. doi:10.1109/4235.996017.

[38] Coleman, T., Ann Branch, M., and Grace, A., "Optimization Toolbox for Use with MATLAB, User's Guide, version 3," 1999.

[39] Wächter, A., and Biegler, L. T., "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, Vol. 106, No. 1, 2006, pp. 25–57.