AAS 22-083

A CONVEX GUIDANCE APPROACH TO TARGET BALLISTIC CAPTURE CORRIDORS AT MARS

Andrea Carlo Morelli,* Gianmario Merisio,† Christian Hofmann,‡ and Francesco Topputo§

Ballistic capture corridors allow a spacecraft to be temporarily captured about a planet without any thrust firing. They represent a promising approach for future deep-space small-satellites missions, where only limited fuel can be carried onboard. In an effort to enable autonomous interplanetary CubeSats, a guidance algorithm based on convex optimization is exploited to design low-thrust minimum-fuel space trajectories which target ballistic capture corridors at Mars. An Hermite–Legendre–Gauss–Lobatto scheme with nonlinear control interpolation is used to discretize the trajectory. A variable time of flight version of the algorithm is developed and tested in closed-loop guidance simulations, where multiple reference trajectories need to be computed during a simulated interplanetary transfer. The variable time of flight algorithm is of paramount importance when closed-loop guidance is considered to avoid that no feasible solutions are found when the spacecraft is too close to the target celestial body. We show the effectiveness of the variable time of flight algorithm compared to the fixed time of flight one.

INTRODUCTION

Since the first CubeSats were launched in 2003, their application has been limited to Earth-observation and technology-demonstrator missions.[1] Recently, the success of the NASA interplanetary CubeSat MarCO[2] proved that small satellites can also be employed in deep-space. Their usage will further be encouraged in the near future by their reduced production costs, and the number of launches will significantly increase. On the other hand, small satellites are still operated from ground, and thus their operational cost is comparable to those of conventional spacecraft (S/C), hence partially jeopardizing the advantage brought by their lower development costs. Increasing the level of the S/C autonomy would be therefore a desirable goal for future space missions to lower the operations-related costs. Consequently, research in the field of autonomous guidance, navigation, and control (GNC) of interplanetary CubeSats has been gaining attention.[3] In particular, recent works have shown that convex optimization represents a promising approach for real-time guidance.[4–6] In this paper, we exploit a sequential convex programming (SCP) technique[7] based on an

*PhD Student, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milan, Italy, andreacarlo.morelli@polimi.it.

†PhD Student, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milan, Italy, gianmario.merisio@polimi.it.

‡PhD Student, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milan, Italy, christian.hofmann@polimi.it.

§Full Professor, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milan, Italy, francesco.topputo@polimi.it.

Hermite–Legendre–Gauss–Lobatto (HLGL) discretization scheme with nonlinear control interpolation[6] to solve the low-thrust minimum-fuel space trajectory optimization problem. This method has already been proven to be effective to solve the considered problem.[8]

Shifting the current paradigm of how space missions are operated and enabling onboard guidance implies that the spacecraft should be able to repeatedly recompute feasible trajectories in real time, as unmodeled perturbations and off-nominal behaviour of the thruster may cause its trajectory to deviate from the nominal one. Since the process of re-optimizing the spacecraft trajectory and update the required command history has thus far been performed from ground, a new approach must be envisaged in case of autonomous guidance. In particular, recent work has proposed and shown the effectiveness of the so-called closed-loop guidance framework,[9] where the spacecraft trajectory is recomputed at predefined times during its interplanetary phase to avoid that it diverges, with consequent loss of the probe. To simulate off-nominal conditions, the equations of motion are propagated with the obtained controls between two subsequent predefined times and the final spacecraft state is further perturbed. The trajectory computed at a previous step is used as the new initial guess for the subsequent optimization process. In this work, we exploit a similar approach. However, we do not impose a priori the times when the spacecraft trajectory must be recomputed, but rather propagate the equations of motion with the obtained controls perturbed with a Gauss–Markov process,[10,11] and recompute the trajectory only when the propagated state and the nominal one differ more than a certain threshold. During the closed-loop guidance process, when the spacecraft is close to its target, it may be possible that perturbations are such that no feasible solutions exist. It would be therefore convenient to shift the final boundary condition such that feasible trajectories can be computed and followed. This, in turns, means that a variable time of flight algorithm is necessary when autonomous guidance is considered.

One of the major challenges for stand-alone interplanetary CubeSats derives from the fact that small satellites have limited control authority, since the amount of fuel that they can carry is limited due to their strict mass budgets. In space exploration missions to planets, spacecraft have to perform demanding maneuvers in short time to enter a closed orbit about them. Given the characteristics of CubeSats' low-thrust propulsion systems, the classical approach is not feasible. Ballistic capture (BC) allows a spacecraft to approach a planet and enter a temporary orbit about it without requiring maneuvers in between. As part of the low-energy transfers, it is a valuable alternative to Keplerian approaches. It grants several benefits in terms of both cost reduction[12] and mission versatility,[13,14] in general at the cost of longer transfer times.[15,16] The BC mechanism is suited for limited-control platforms, which cannot afford to enter into orbits about a planet because of the lack of proper means. In the past, the BC mechanism was used to rescue Hiten,[17] and to design insertion trajectories in lunar missions like SMART-1[18] and GRAIL.[19] More recently, BC orbits were proposed for BepiColombo,[20] Lunette,[21] and ESMO[22] missions.

BC is an event that occurs in extremely rare occasions and requires acquiring a proper state (position and velocity) far away from the target planet.[14] Massive numerical simulations are required to find the specific initial conditions (ICs) that support capture.[23] The union of those points defines the capture set, which in turn is used to develop a ballistic capture corridor (BCC).[24] BCCs are time-varying manifolds that allow temporary capture about the target body independently on the insertion epoch of the S/C inside the corridors themselves. Consequently, there is no reason to define the guidance time of flight a priori, and hence a variable time of flight guidance algorithm is desirable. In Reference 25, the equations of motions were rewritten in polar coordinates with a change of the independent variable to solve the minimum-time problem. We extend the work in

Reference 25 and build a three-dimensional low-thrust minimum-fuel space trajectory optimization algorithm with variable time of flight. In this work, we use the time as independent variable and linearize the equations of motion with respect to it.

In this work, we develop a variable time of flight version of the convex optimization algorithm able to effectively target trajectories rather than points and, consequently, ballistic capture corridors. We show the efficacy of our approach in closed-loop simulations to target BCCs and compare the results with the ones obtained using the fixed time of flight version.

BALLISTIC CAPTURE CORRIDORS

BC trajectories perform p revolutions about the central body in forward time neither impacting or escaping,[26,27] while they escape the target when integrated backward. When searching for BC trajectories, most of the solutions are spurious, meaning that they are highly unstable trajectories.[27] On the contrary, ideal solutions having regular, quasi-stable post-capture orbits are inferred exploiting the regularity index* S and regularity coefficient $\Delta S_{\%}$.[28] Numerical experiments showed that high-quality post-capture orbits are identified by small regularity index and coefficient.[27–30] On the other hand, the capture ratio $\mathcal{R}_{\mathcal{C}}$ is used to quantitatively measure capture occurrence.[30] A capture set is defined as $\mathcal{C}^p_{-1} := \mathcal{W}^p \cap \mathcal{X}_{-1}$, where $\mathcal{W}_p$ is the weakly stable set containing ICs that perform p complete revolutions around the target without escaping or impacting on it or its moons in forward time. Differently, $\mathcal{X}_{-1}$ is the unstable set containing ICs that escape from the target before completing the first revolution in backward time.[27] $\mathcal{W}^p$ and $\mathcal{X}_{-1}$ are defined according to the algorithmic definition of weak stability boundary (WSB).[23] A BCC is a moving manifold that supports capture. A corridor is obtained by backward propagating ICs belonging to a capture sets $\mathcal{C}^p_{-1}$.[24] Similarly to how capture sets are symbolically designated, a corridor developed from a capture set $\mathcal{C}^p_{-1}$ is indicated as $\check{\mathcal{B}}^p_{-1}$.

Dynamical model

Following the nomenclature in Reference 27, a *target* and a *primary* are defined. The target is the body about which the ballistic capture is studied. The primary is the main body about which the target revolves. The target and primary masses are m_t and m_p, respectively. This work focuses on ballistic capture having Mars as target and the Sun as primary.

Ephemerides. The following reference frames are used: J2000, ECLIPJ2000, RTN@t_i, and RPF. Details about them can be found in Reference 24. In this work, the precise states of the Sun and the major planets are retrieved from the Jet Propulsion Laboratory (JPL)'s planetary ephemerides `de440s.bsp`† (or DE440s).[31] Additionally, the ephemerides `mars097.bsp` of Mars (the target) and its moons are employed‡. The following generic leap seconds kernel (LSK) and planetary constants kernel (PCK) are used: `naif0012.tls`, `pck00010.tpc`, and `gm_de440.tpc`§.

*The same nomenclature introduced in Reference 28 is used in this work.

†Data publicly available at: `https://naif.jpl.nasa.gov/pub/naif/generic_kernels/spk/planets/de440s.bsp` [retrieved Jan 20, 2022].

‡`~/spk/satellites/mars097.bsp` [retrieved Jan 20, 2022].

§Data publicly available at: `https://naif.jpl.nasa.gov/pub/naif/generic_kernels/lsk/naif0012.tls`, and `~/generic_kernels/pck/pck00010.tpc` [retrieved Jan 20, 2022]. The `gm_de440.tpc` PCK kernel consistent with the ephemerides DE440s is custom made. In fact, it has not been released yet.

Table 1. Spacecraft Parameters for SRP Evaluation.

Parameter	Unit	Value
Mass–SRP area ratio m/A	$\mathrm{kg/m^2}$	75
Coefficient of reflectivity C_r	-	1.3

Equations of motion. The equations of motion of the restricted n-body problem are considered. The gravitational attractions of the Sun, Mercury, Venus, Earth (B*), Mars (central body), Jupiter (B), Saturn (B), Uranus (B), and Neptune (B) are taken into account. Additionally, solar radiation pressure (SRP), Mars' non-spherical gravity (NSG), and relativistic corrections[32] (Schwarzschild solution, geodesic precession, and Lense-Thirring precession) are also included in the model. The assumed spacecraft parameters needed to evaluate the SRP perturbation are collected in Table 1. They are compatible with the parameters of a 12U deep-space CubeSat.[33] The terms of the infinite series modeling the NSG are considered up to degree $n_{\mathrm{deg}} = 20$ and order $n_{\mathrm{ord}} = 20$. The coefficients to evaluate the NSG perturbation are retrieved from the MRO120F gravity field model of Mars. Data are publicly available in the file `jgmro_120f_sha.tab`, archived in the Geosciences Node of NASA's Planetary Data System †. Far from Mars, when not in a temporary capture condition, the NSG perturbation is neglected. Equations of motion are integrated in the J2000 inertial frame.

The equations of motion, written in a non-rotating Mars-centered reference frame are[27,32,34]

$$\begin{aligned}\ddot{\mathbf{r}} = &-\frac{\mu_t}{r^3}\mathbf{r} - \sum_{i\in\mathbb{P}}\mu_i\left(\frac{\mathbf{r}_i}{r_i^3} + \frac{\mathbf{r}-\mathbf{r}_i}{\|\mathbf{r}-\mathbf{r}_i\|^3}\right) + \frac{QA}{m}\frac{\mathbf{r}-\mathbf{r}_\odot}{\|\mathbf{r}-\mathbf{r}_\odot\|^3} - \mathcal{R}\frac{\mu_t}{r^2}\left(\Lambda\frac{\mathcal{R}^\top\mathbf{r}}{r} - \begin{bmatrix} J \\ K \\ H \end{bmatrix}\right) + \\ &+ \frac{\mu_t}{c^2 r^3}\left[\left(4\frac{\mu_t}{r} - v^2\right)\mathbf{r} + 4\left(\mathbf{r}\cdot\mathbf{v}\right)\mathbf{v}\right] + 2\left(\Omega\times\mathbf{v}\right) + 2\frac{\mu_t}{c^2 r^3}\left[\frac{3}{r^2}\left(\mathbf{r}\times\mathbf{v}\right)\left(\mathbf{r}\cdot\mathbf{J}\right) + \left(\mathbf{v}\times\mathbf{J}\right)\right]\end{aligned} \tag{1}$$

where μ_t is the gravitational parameter of the target body (Mars in this work); $\mathbf{r}$ and $\mathbf{v}$ are the position and velocity vectors of the spacecraft with respect to the target, respectively, being r and v their magnitudes; $\mathbb{P}$ is a set of $n-2$ indexes (where n refers to the n-body problem) each referring to the perturbing bodies; μ_i and $\mathbf{r}_i$ are the gravitational parameter and position vector of the i-th body with respect to the target, respectively; A is the Sun-projected area on the spacecraft for SRP evaluation; m is the spacecraft mass; $\mathbf{r}_\odot$ is the position vector of the Sun with respect to the target; $\mathcal{R}$ is the time-dependent matrix transforming vector components from the Mars-fixed frame to the non-rotating frame in which the equations of motion are written; Λ, J, K, and H are defined as in Reference 35; $c = 299\,792\,458\,\mathrm{m\,s^{-1}}$ (from SPICE[36,37]) is the speed of light in vacuum; $\mathbf{J}$ is the rotating central body's angular momentum per unit mass in the J2000 frame. Then, Q is such that

$$Q = \frac{LC_r}{4\pi c} \tag{2}$$

where C_r is the spacecraft coefficient of reflectivity, and $L = S_\odot 4\pi d_{\mathrm{AU}}^2$ is the luminosity of the Sun. The latter is computed from the solar constant‡ $S_\odot = 1367.5\,\mathrm{W\,m^{-2}}$ evaluated at $d_{\mathrm{AU}} = 1\,\mathrm{AU}$.

*Here B stands for barycenter.

†Data publicly available at: `https://pds-geosciences.wustl.edu/mro/mro-m-rss-5-sdp-v1/mrors_1xxx/data/shadr/` [retrieved Jan 20, 2022].

‡`https://extapps.ksc.nasa.gov/Reliability/Documents/Preferred_Practices/2301.pdf` [last accessed Jan 20, 2022].

Table 2. Physical Quantities for BCC Section.

Physical Quantity	Unit	Normalization Factor	Value
Mass	kg	$\mathrm{MU} = M_{\male}$	6.417120×10^{23}
Length	km	$\mathrm{LU} = R_{\male}$	3.396000×10^{3}
Time †	s	$\mathrm{TU} = \left[\mathrm{LU}^3/(G\,\mathrm{MU})\right]^{1/2}$	9.562814×10^{2}
Velocity	km/s	VU = LU/TU	3.551256

† Gravitational constant $G = 6.674080 \times 10^{-20}\,\mathrm{km}^3/\mathrm{kg}/\mathrm{s}^2$.[41]

Lastly, Ω in Eq. 1 is

$$\Omega = \frac{3}{2}\mathbf{v}_{\odot/t} \times \left(-\frac{\mu_{\odot}\mathbf{r}_{\odot/t}}{c^2 r_{\odot/t}^3} \right) \tag{3}$$

where $\mu_{\odot}$ is the gravitational parameter of the Sun; $\mathbf{r}_{\odot/t}$ and $\mathbf{v}_{\odot/t}$ are the position and velocity vectors, respectively, of the target body with respect to the Sun, being $r_{\odot/t}$ and $v_{\odot/t}$ their magnitudes.

Numerical integration of the equations of motion. The equations of motion in Eq. (1) are integrated with the GRavity TIdal Slide (GRATIS) tool[38] in their nondimensional form to avoid ill-conditioning.[27] The physical quantities specifically used to write the dimensionless equations are reported in Table 2. Numerical integration is carried out with the Dormand-Prince 8th-order embedded Runge-Kutta method propagation scheme.[39] That is an adaptive step 8th-order RK integrator with 7th-order error control. The coefficients were derived by Prince and Dormand.[40] The dynamics is propagated with relative and absolute tolerances both set to 10^{-12}.[27]

Corridor development

In this work, ten $\mathcal{C}_{-1}^{6}$ capture sets are constructed following the methodology discussed in Reference 27. The grid of ICs is chosen to maximize the capture ratio $\mathcal{R}_c$ based on the analysis reported in Reference 30. Ten capture epochs t_{CAP} uniformly distributed from January 1, 2030 at 12:00:00.000 (UTC) to February 20, 2032 at 10:32:39.144 (UTC) are chosen, covering a complete Earth–Mars synodic period of ≈ 780 days*. The inclination i_0 and the right ascension of the ascending node Ω_0 of the orbital plane of the initial osculating orbits are taken both equal to 0.2π rad, maximizing the capture ratio for Mars (see Figure 10 in Reference 30). The initial osculating eccentricity e_0 is fixed equal to 0.99.[14] Finally, a discretization of pericenter radius r_0 and argument of pericenter ω_0 is assumed to build the grid of ICs. r_0 is discretized in the interval $[R_t + \varepsilon, 10R_t]$ with step $\Delta r_0 = 50\,\mathrm{km}$ ($\varepsilon = 100\,\mathrm{km}$). ω_0 is discretized in the interval $[0, 2\pi)$ with step $\Delta\omega = 1\,\mathrm{deg}$. The grid is made of $N_{r_0} = 678$ and $N_{\omega_0} = 360$ evenly spaced points of r_0 and ω_0, respectively. Each IC is assumed to be the pericenter of an osculating orbit, and therefore having true anomaly $\theta_0 = 0$. Finally, a corridor is developed from the capture set showing the best compromise in terms of both the regularity index and capture ratio.[24]

Figure 1 summarizes the quantitative and qualitative information about all the ten $\mathcal{C}_{-1}^{6}$ capture sets computed. This is used to select the capture set from which the corridor to target has been developed. The capture set selected is the one having capture epoch t_{CAP} at November 25, 2031 at 18:40:49.054 (UTC). It shows a large capture ratio and it seems very regular. In fact, both the

*The data set supporting the findings of this study is available on Zenodo at `https://zenodo.org/` [last accessed Jan 20, 2022] with the identifier `https://doi.org/10.5281/zenodo.5931462`.

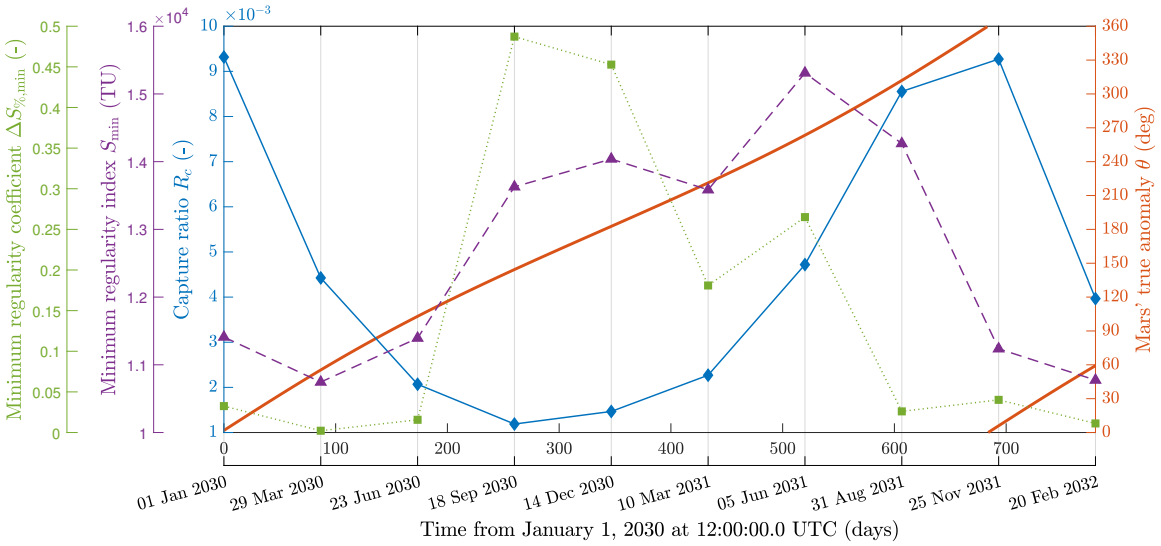


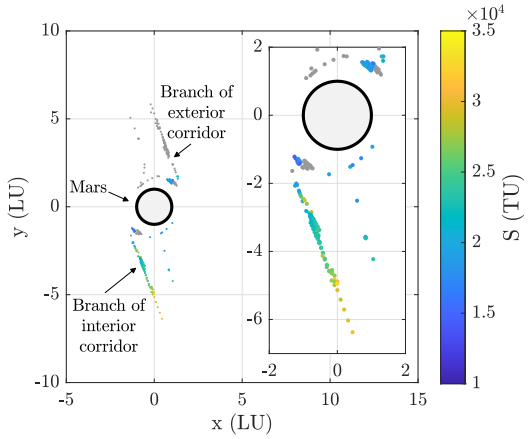
Figure 1. Qualitative and quantitative indexes of the ten $\mathcal{C}^6_{-1}$ capture sets as a function of the time past January 1, 2030 at 12:00:00.0 (UTC). Capture epochs of capture sets also shown. On the left y-axes, capture ratio R_c (thick, blue, filled diamonds line), minimum regularity index $\check{S}$ (dashed, purple, filled triangles line), and minimum regularity coefficient $\Delta S_{\%}$ (dotted, green, filled squares line). On the right y-axis, Mars' true anomaly θ (thick, orange line).

minimum regularity index and coefficient are small. For similar reasons, another valid option would have been the capture set with capture epoch t_{CAP} occurring at January 1, 2030 at 12:00:00.000 (UTC). The selected capture set is compared against a poor one in Figure 2. The plot in Figure 2(a) shows how the capture set is fragmented, unstructured, and not very regular when compared to the selected one. Since only the most regular portion of the corridor is of interest for targeting purposes, just a subset of the capture set's ICs is used to built it. This results in constructing a subcorridor $\check{\mathcal{S}}^p_{-1}$, which is defined as a subset of the corridor $\check{\mathcal{B}}^p_{-1}$. In this paper, the subcorridor $\check{\mathcal{S}}^6_{-1}$ is developed starting from a circular domain with radius 500 km located in the most regular region of the capture set and opportunistically chosen by visual inspection. The magnification of Figure 2(b) shows the selected domain, which is placed at the top of the cluster of ICs developing the interior corridor and spans the cluster width almost completely. Finally, the envelope $\partial\check{\mathcal{S}}^6_{-1}$ of the subcorridor is represented in Figure 3.

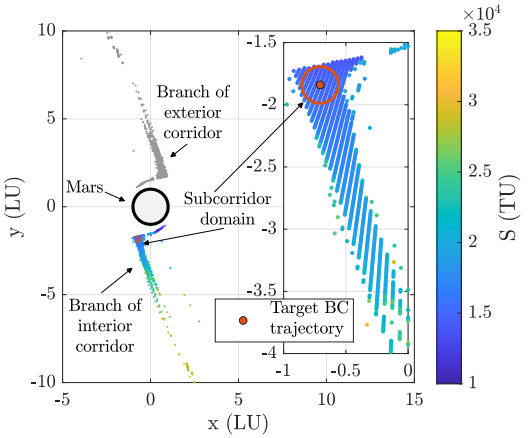
In this work, instead of the whole subcorridor, a single trajectory belonging inside $\check{\mathcal{S}}^6_{-1}$ is targeted by the guidance algorithm. The IC of such trajectory corresponds to the center of the subcorridor's initial circular domain, which is the red dot shown in the magnification of Figure 2(b). A portion of the complete trajectory is highlighted in Figure 3(b). Throughout this paper, it is referred to indistinctly as target trajectory or target BC trajectory.

PROBLEM FORMULATION

Figure 4 represents the transfer problem considered in this paper. The transfer is divided into two parts: the guidance part and the ballistic one. We want to find the fuel-optimal trajectory of a S/C from an initial point $\mathbf{x}_0$ on the Earth's orbit to Mars, exploiting the target ballistic capture trajectory

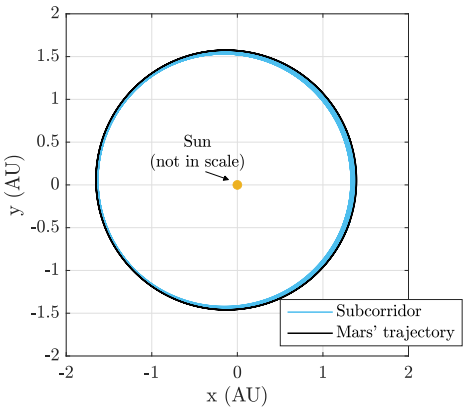


(a) Capture set $\mathcal{C}^6_{-1}$ at t_{CAP} = September 18, 2030 at 11:30:18.396 (UTC). In the magnification, details of the capture set close to Mars which is fragmented, unstructured, and not very regular. Clusters close to Mars' surface present ICs developing both interior and exterior BCCs.

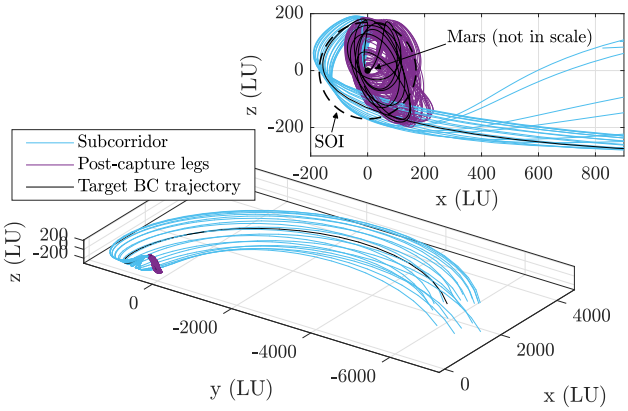


(b) Capture set $\mathcal{C}^6_{-1}$ at t_{CAP} = November 25, 2031 at 18:40:49.054 (UTC). In the magnification, location of the circular domain opportunistically selected to build the subcorridor. The domain is located at the top of the cluster of ICs developing the interior corridor because that is a very regular region of the capture set. The diameter of the circular domain spans the cluster width almost completely.

Figure 2. Capture sets $\mathcal{C}^6_{-1}$. Branches developing exterior BBCs are colored in gray. Regularity index of ICs belonging to branches developing interior BCCs. ICs are represented in nondimensional quantities in the Mars-centered RTN@t_{CAP} frame. Mars is the gray circle with black surround.



(a) Subcorridor envelope, in light blue, in the Sun-centered ECLIPJ2000 frame compared to Mars's orbit, in black. Envelope backward propagated for 2000 days.



(b) Three-dimensional view of the subcorridor envelope in the Mars-centered ECLIPJ2000 frame. In the magnification, subcorridor envelope close to Mars' sphere of influence, xz-plane. Post-capture legs and target BC trajectory shown in purple and black, respectively. Envelope backward propagated for 600 days.

Figure 3. Representations in the physical space of the subcorridor envelope $\partial\check{\mathcal{S}}^6_{-1}$ developed from capture set with t_{CAP} occurring at November 25, 2031 at 18:40:49.054 (UTC), in light blue, built from the red circular domain border shown in Figure 2(b).

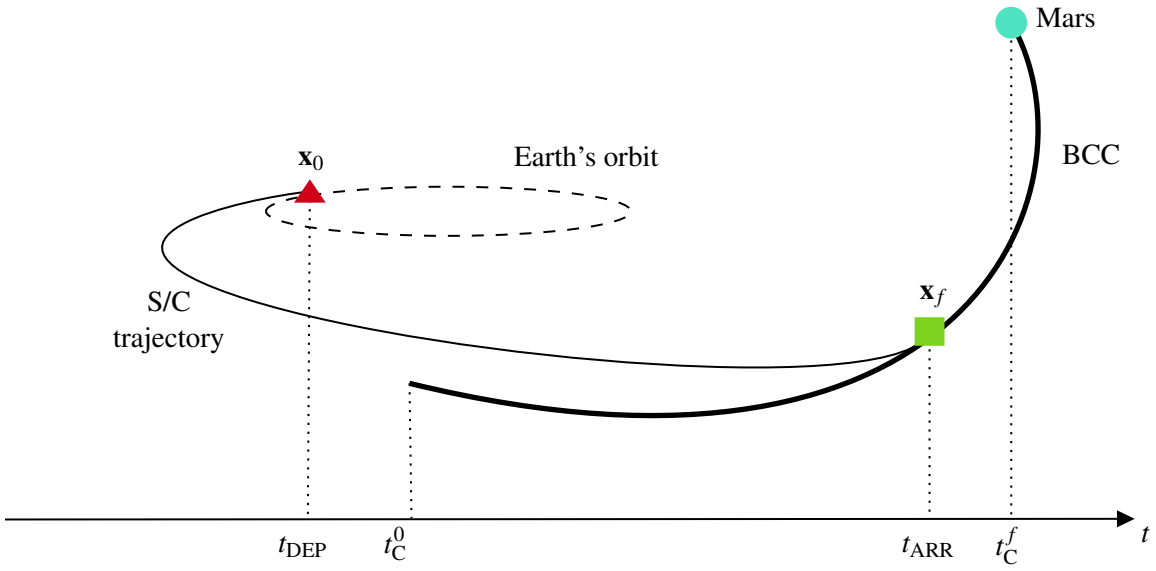


Figure 4. Representation of the Problem.

defined between times $t_{\mathrm{C}}^0 > t_{\mathrm{DEP}}$ and $t_{\mathrm{C}}^f < 0.75 t_{CAP}$. The margin on t_{CAP} is considered to avoid entering the corridor after having crossed the Mars' SOI. The spacecraft leaves Earth's orbit at time t_{DEP} and arrives at the BCC at time t_{ARR}. The time of flight of the guidance part of the trajectory is therefore defined as

$$t_f = t_{\mathrm{ARR}} - t_{\mathrm{DEP}} \tag{4}$$

where $t_{\mathrm{C}}^0 \leq t_{\mathrm{ARR}} < t_{\mathrm{C}}^f$. The total time of flight TOF of the S/C is the sum of t_f and the ballistic time of flight, defined as

$$\Delta t_{\mathrm{b}} = t_{\mathrm{C}}^f - t_{\mathrm{ARR}}. \tag{5}$$

Therefore we have that

$$\mathrm{TOF} = t_f + \Delta t_{\mathrm{b}}. \tag{6}$$

Note that TOF is constant, as the departure and capture dates are fixed. In this work, we develop a guidance algorithm for which, in general, t_f can vary. Therefore, the sum of t_f and Δt_{b} is constant, but the single terms may vary.

When referring to the guidance part of the trajectory, we consider the departure time as $t_0 = 0$ and the final time as t_f. We seek to minimize the fuel consumption of the spacecraft during the guidance part of the trajectory, i.e. to maximize the spacecraft mass at time t_f, or

$$\underset{T}{\text{minimize}} \quad -m(t_f) \tag{7}$$

where T is the thrust magnitude. For the guidance algorithm, we consider two-body dynamics only. The motion of a spacecraft during the interplanetary phase is thus described by

$$\begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ -\mu \mathbf{r}/r^3 + \mathbf{T}/m \\ -T/(I_{\mathrm{sp}} g_0) \end{bmatrix} \tag{8}$$

where $\mathbf{r} = [r_x, r_y, r_z]^\top$, $\mathbf{v} = [v_x, v_y, v_z]^\top$, and m are the position vector, the velocity vector, and the mass of the spacecraft, respectively. μ is the standard gravitational parameter of the primary body (in our case, the Sun), $\mathbf{T} = [T_x, T_y, T_z]^\top$, and $T = \|\mathbf{T}\|_2$. I_{sp} and g_0 are the specific impulse and gravitational acceleration, respectively. In Eq. 8, states and controls are coupled through the term $\mathbf{T}/m$; moreover, the equations of motion are nonconvex. Consequently, they must be convexified in order for the problem to be solved by means of convex optimization techniques. By applying a change of variables,[4,25] Eq. 8 can be rewritten as

$$\dot{\mathbf{x}} = \mathbf{f}_f(\mathbf{x}, \mathbf{u}) + \mathbf{B}\mathbf{u} \tag{9}$$

where

$$\mathbf{f}_f = [v_x, v_y, v_z, -r_x/r^3, -r_y/r^3, -r_z/r^3, 0]^\top. \tag{10}$$

Moreover,

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{3\times 4} \\ \mathbf{I}_{3\times 3} \quad \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} \quad b \end{bmatrix} \tag{11}$$

with $b = -1/(I_{sp} g_0)$. The new vector of controls is $\mathbf{u} = [\tau_x, \tau_y, \tau_z, \Gamma]^\top$, with $\tau_i = T_i/m$, $i = x, y, z$ and $\Gamma = T/m$. The mass m has been substituted with the variable $w = \ln m$, and hence the new state vector is $\mathbf{x} = [\mathbf{r}^\top, \mathbf{v}^\top, w]^\top$. After the change of variables, the following two convex inequality constraints arise:

$$\tau_x^2 + \tau_y^2 + \tau_z^2 \leq \tau^2 \tag{12a}$$

$$0 \leq \tau \leq T_{\max} e^{-w^*}[1 - (w - w^*)] + \chi. \tag{12b}$$

where the slack variable $\chi \geq 0$ has been introduced to avoid artificial infeasibility. Equation 9 can finally be convexified through linearization to obtain

$$\dot{\mathbf{x}} = \mathbf{f}_f(\mathbf{x}^*) + \mathbf{A}(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + \mathbf{B}\mathbf{u} + \nu \tag{13}$$

where the superscript $(\cdot)^*$ indicates the reference trajectory, $\mathbf{A} = \partial \mathbf{f}_f / \partial \mathbf{x}$ is the Jacobian matrix of the free spacecraft dynamics, and ν is another slack variable. To ensure that the linearization is valid, a trust-region constraint with radius R needs to be enforced as

$$\|\mathbf{x} - \mathbf{x}^*\|_1 \leq R. \tag{14}$$

In general, upper and lower bounds on the state and control variables must be imposed too, and thus

$$\mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u \tag{15a}$$

$$\mathbf{u}_l \leq \mathbf{u} \leq \mathbf{u}_u. \tag{15b}$$

Moreover, initial boundary conditions are imposed as

$$\mathbf{x}(t_0) = \mathbf{x}_0. \tag{16}$$

Usually, final boundary conditions are imposed similarly, although the final spacecraft mass is free. In this paper, however, we aim to target a trajectory inside a ballistic capture corridor rather than a specific point in the state space. For this reason, the final boundary conditions can be written as

$$\mathbf{r}(t_f) = \mathbf{r}_{\text{BC}}(t_{\text{ARR}}), \quad \mathbf{v}(t_f) = \mathbf{v}_{\text{BC}}(t_{\text{ARR}}) \tag{17}$$

where $\mathbf{r}_{\mathrm{BC}}(t_{\mathrm{ARR}})$ and $\mathbf{v}_{\mathrm{BC}}(t_{\mathrm{ARR}})$ are the ephemerides of the target BC trajectory, i.e., the position and velocity that the spacecraft must have to be inserted into the corridor as a function of the time t_{ARR}. We approximate $\mathbf{r}_{\mathrm{BC}}(t_{\mathrm{ARR}})$ and $\mathbf{v}_{\mathrm{BC}}(t_{\mathrm{ARR}})$ by means of polynomials. Therefore, they are in general nonconvex. We convexify them by linearization, such that

$$\mathbf{r}(t_f) = \mathbf{r}_{\mathrm{BC}}(t^*_{\mathrm{ARR}}) + \left.\frac{\mathrm{d}\mathbf{r}_{\mathrm{BC}}}{\mathrm{d}t_{\mathrm{ARR}}}\right|_{t_{\mathrm{ARR}}=t^*_{\mathrm{ARR}}} (t_{\mathrm{ARR}} - t^*_{\mathrm{ARR}}) + \boldsymbol{\gamma} \tag{18a}$$

$$\mathbf{v}(t_f) = \mathbf{v}_{\mathrm{BC}}(t^*_{\mathrm{ARR}}) + \left.\frac{\mathrm{d}\mathbf{v}_{\mathrm{BC}}}{\mathrm{d}t_{\mathrm{ARR}}}\right|_{t_{\mathrm{ARR}}=t^*_{\mathrm{ARR}}} (t_{\mathrm{ARR}} - t^*_{\mathrm{ARR}}) + \boldsymbol{\kappa}. \tag{18b}$$

where $\boldsymbol{\gamma}$ and $\boldsymbol{\kappa}$ are two slack variables. To ensure that the linearization of the final boundary conditions is valid, the following additional trust-region constraint with radius R_{h} must be considered

$$|t_f - t^*_f| \leq R_{\mathrm{h}}. \tag{19}$$

Upper and lower bounds on the time of flight are imposed as

$$t_{f,l} \leq t_f \leq t_{f,u}. \tag{20}$$

The convex low-thrust minimum-fuel space trajectory optimization problem can finally be written as

$$\underset{T}{\text{minimize}} \quad -w(t_f) + \lambda \left\|\nu\right\|_1 + \lambda \left\|\gamma\right\|_1 + \lambda \left\|\kappa\right\|_1 + \lambda \max(0,\chi) \tag{21a}$$

$$\text{subject to:} \quad \dot{\mathbf{x}} = \mathbf{f}_f(\mathbf{x}^*) + \mathbf{A}(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + \mathbf{B}\mathbf{u} + \nu \tag{21b}$$

$$\tau_x^2 + \tau_y^2 + \tau_z^2 \leq \tau^2 \tag{21c}$$

$$0 \leq \tau \leq T_{\max} e^{-w^*}[1 - (w - w^*)] + \chi \tag{21d}$$

$$\left\|\mathbf{x} - \mathbf{x}^*\right\|_1 \leq R \tag{21e}$$

$$\mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u \tag{21f}$$

$$\mathbf{u}_l \leq \mathbf{u} \leq \mathbf{u}_u \tag{21g}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \tag{21h}$$

$$\mathbf{r}(t_f) = \mathbf{r}_{\mathrm{BC}}(t^*_{\mathrm{ARR}}) + \left.\frac{\mathrm{d}\mathbf{r}_{\mathrm{BC}}}{\mathrm{d}t_{\mathrm{ARR}}}\right|_{t_{\mathrm{ARR}}=t^*_{\mathrm{ARR}}} (t_{\mathrm{ARR}} - t^*_{\mathrm{ARR}}) + \boldsymbol{\gamma} \tag{21i}$$

$$\mathbf{v}(t_f) = \mathbf{v}_{\mathrm{BC}}(t^*_{\mathrm{ARR}}) + \left.\frac{\mathrm{d}\mathbf{v}_{\mathrm{BC}}}{\mathrm{d}t_{\mathrm{ARR}}}\right|_{t_{\mathrm{ARR}}=t^*_{\mathrm{ARR}}} (t_{\mathrm{ARR}} - t^*_{\mathrm{ARR}}) + \boldsymbol{\kappa} \tag{21j}$$

$$|t_f - t^*_f| \leq R_{\mathrm{h}} \tag{21k}$$

$$t_{f,l} \leq t_f \leq t_{f,u}. \tag{21l}$$

The objective function has been augmented with the slack variables weighted with a sufficiently large parameter $\lambda > 0$, as they must be zero at the end of the optimization process.

CONVEX GUIDANCE ALGORITHM

Solving problem in Eqs. 21a-21l does not mean solving the original nonconvex space trajectory optimization problem. In fact, the SCP technique[7] is required. In this context, the convex problem is repeatedly solved until the original nonconvex constraints are satisfied. This section details the discretization method used to solve each of the convex subproblems, and explains how the variable time of flight algorithm has been integrated inside the sequential convex programming.

Arbitrary-order Hermite–Legendre–Gauss–Lobatto Discretization

The arbitrary-order Hermite–Legendre–Gauss–Lobatto discretization method[6] consists of approximating the state variables by means of arbitrary-order polynomials using an Hermite interpolation and the Legendre–Gauss–Lobatto rules to define nodal and collocation points.[42] The total time of flight is divided into K segments. Each segment $[t_i, t_{i+1}]$ is mapped into the interval $[-1,1]$ through the transformation

$$t \rightarrow \frac{h}{2}\xi + \frac{t_{i+1}+t_i}{2}, \quad i = 1,\ldots,K-1 \tag{22}$$

where $\xi \in [-1,1]$ and $h = t_{k+1} - t_k$ is the time step. In this work, nodes and collocation points are defined inside the interval $[-1,1]$ as the roots of the derivative of the $(n-1)$th order Legendre polynomial,[42] where n is the order of the method.

The state $\mathbf{x}^{(k)}(\xi) \in \mathbb{R}^{n_x \times 1}$ ($n_x = 7$) is approximated inside the kth segment as

$$\mathbf{x}^{(k)}(\xi) \approx \mathbf{a}_0^{(k)} + \mathbf{a}_1^{(k)}\xi + \cdots + \mathbf{a}_n^{(k)}\xi^n, \quad k = 1,\ldots,K \tag{23}$$

while the control $\mathbf{u}^{(k)}(\xi) \in \mathbb{R}^{n_u \times 1}$ ($n_u = 4$) as

$$\mathbf{u}^{(k)}(\xi) \approx \mathbf{a}_{u,0}^{(k)} + \mathbf{a}_{u,1}^{(k)}\xi + \cdots + \mathbf{a}_{u,n}^{(k)}\xi^{n_p-1}, \quad k = 1,\ldots,K \tag{24}$$

where $n_p = (n+1)/2$ is the number of nodes in each segment. According to the HLGL method, when $h = const.$ the right-hand side of Eq.9 is discretized at the points ξ as

$$\mathbf{f}_{l,hf}(\mathbf{x}(\xi),\mathbf{u}(\xi)) = \frac{h}{2}[\mathbf{f}_f(\mathbf{x}^*(\xi)) + \mathbf{A}(\mathbf{x}^*(\xi))(\mathbf{x}(\xi) - \mathbf{x}^*(\xi)) + \mathbf{B}\mathbf{u}(\xi)] \tag{25}$$

where the superscript $(\cdot)^{(k)}$ has been dropped for simplicity. Note that $h = const.$ corresponds to constant time of flight t_f. In fact, we have that $t_f = hK$. To extend the method and allow the time of flight to vary, we further linearize Eq. 9 with respect to h, yielding

$$\mathbf{f}_{l,hv}(\mathbf{x}(\xi),\mathbf{u}(\xi)) = \frac{h^*}{2}[\mathbf{A}(\mathbf{x}^*(\xi))(\mathbf{x}(\xi) - \mathbf{x}^*(\xi)) + \mathbf{B}\mathbf{u}(\xi)] + \frac{h}{2}[\mathbf{f}_f(\mathbf{x}^*(\xi)) + \mathbf{B}\mathbf{u}^*(\xi)]. \tag{26}$$

Note that the subscripts $(\cdot)_{l,hf}$ and $(\cdot)_{l,hv}$ indicate the linearized dynamics with fixed and variable time step, respectively.

At this point, the column vectors of coefficients $\mathbf{a}_q^{(k)} \in \mathbb{R}^{n_x \times 1}$, $q = 0,\ldots,n$ in Eq. 23 are found by solving the following linear system:

$$\underbrace{\begin{bmatrix} \mathbf{I}_{n_x} & \theta_1\mathbf{I}_{n_x} & \theta_1^2\mathbf{I}_{n_x} & \ldots & \theta_1^n\mathbf{I}_{n_x} \\ \mathbf{I}_{n_x} & \theta_2\mathbf{I}_{n_x} & \theta_2^2\mathbf{I}_{n_x} & \ldots & \theta_2^n\mathbf{I}_{n_x} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{I}_{n_x} & \theta_{n_p}\mathbf{I}_{n_x} & \theta_{n_p}^2\mathbf{I}_{n_x} & \ldots & \theta_{n_p}^n\mathbf{I}_{n_x} \\ \mathbf{0}_{n_x} & \mathbf{I}_{n_x} & 2\theta_1\mathbf{I}_{n_x} & \ldots & n\theta_1^{n-1}\mathbf{I}_{n_x} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{n_x} & \mathbf{I}_{n_x} & 2\theta_{n_p}\mathbf{I}_{n_x} & \ldots & n\theta_{n_p}^{n-1}\mathbf{I}_{n_x} \end{bmatrix}}_{\theta} \underbrace{\begin{bmatrix} \mathbf{a}_0^{(k)} \\ \mathbf{a}_1^{(k)} \\ \vdots \\ \mathbf{a}_{n_p}^{(k)} \\ \vdots \\ \mathbf{a}_{n-1}^{(k)} \\ \mathbf{a}_n^{(k)} \end{bmatrix}}_{\mathbf{a}^{(k)}} = \underbrace{\begin{bmatrix} \mathbf{x}^{(k)}(\theta_1) \\ \mathbf{x}^{(k)}(\theta_2) \\ \vdots \\ \mathbf{x}^{(k)}(\theta_{n_p}) \\ \mathbf{f}_{l,hv}^{(k)}(\theta_1) \\ \vdots \\ \mathbf{f}_{l,hv}^{(k)}(\theta_{n_p}) \end{bmatrix}}_{\mathbf{b}^{(k)}}. \tag{27}$$

In Eq. (27), θ_j ($j = 1,\ldots,n_p$) are the positions of the nodal points, $\mathbf{I}_{n_x}$ is the $n_x \times n_x$ identity matrix, and $\mathbf{0}_{n_x}$ the $n_x \times n_x$ null matrix. Moreover, $f^{(k)}_{l,hv}(\theta_j) = f^{(k)}_{l,hv}(\mathbf{x}(\theta_j),\mathbf{u}(\theta_j))$. Once the coefficients $\mathbf{a}^{(k)}_q$ have been determined as $\mathbf{a}^{(k)} = \theta^{-1}\mathbf{b}^{(k)}$, Eq. 23 can be used to define the state and its derivative at the collocation points:

$$\begin{aligned}
\mathbf{x}^{(k)}(\zeta) &= \underbrace{\begin{bmatrix} \mathbf{1}_{n_x} & \zeta_1\mathbf{1}_{n_x} & \ldots & \zeta_1^n\mathbf{1}_{n_x} \\ \mathbf{1}_{n_x} & \zeta_2\mathbf{1}_{n_x} & \ldots & \zeta_2^n\mathbf{1}_{n_x} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{1}_{n_x} & \zeta_{n_c}\mathbf{1}_{n_x} & \ldots & \zeta_{n_c}^n\mathbf{1}_{n_x} \end{bmatrix}}_{\zeta} \underbrace{\begin{bmatrix} \mathbf{a}^{(k)}_0 \\ \mathbf{a}^{(k)}_1 \\ \vdots \\ \mathbf{a}^{(k)}_n \end{bmatrix}}_{\mathbf{a}^{(k)}} = \zeta\theta^{-1}\mathbf{b}^{(k)} = \phi\mathbf{b}^{(k)} \\
\frac{\mathrm{d}\mathbf{x}^{(k)}(\zeta)}{\mathrm{d}\xi} &= \underbrace{\begin{bmatrix} \mathbf{0}_{n_x} & \mathbf{1}_{n_x} & \ldots & n\zeta_1^{n-1}\mathbf{1}_{n_x} \\ \mathbf{0}_{n_x} & \mathbf{1}_{n_x} & \ldots & n\zeta_2^{n-1}\mathbf{1}_{n_x} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0}_{n_x} & \mathbf{1}_{n_x} & \ldots & n\zeta_{n_c}^{n-1}\mathbf{1}_{n_x} \end{bmatrix}}_{\zeta'} \underbrace{\begin{bmatrix} \mathbf{a}^{(k)}_0 \\ \mathbf{a}^{(k)}_1 \\ \vdots \\ \mathbf{a}^{(k)}_n \end{bmatrix}}_{\mathbf{a}^{(k)}} = \zeta'\theta^{-1}\mathbf{b}^{(k)} = \phi'\mathbf{b}^{(k)}
\end{aligned} \tag{28}$$

where ζ_j ($j = 1,\ldots,n_c = (n-1)/2$) are the positions of the collocation points. The control at the collocation points is obtained similarly. Note, however, that no information on the control dynamics is available and thus only the first n_p rows of the system in Eq. 27 can be considered. For this reason, the control is approximated by means of a polynomial of order $n_p - 1$. Once the matrices and vectors corresponding to all the trajectory segments are constructed, the dynamical constraints can be written as

$$\Delta = \Phi'\hat{\mathbf{b}} - \frac{h^*}{2}[\hat{\mathbf{A}}(\Phi\hat{\mathbf{b}} - \Phi\hat{\mathbf{b}}^*) + \hat{\mathbf{B}}\Phi_u\hat{\mathbf{b}}_u] - \frac{h}{2}[\hat{\mathbf{f}}_f + \hat{\mathbf{B}}\Phi_u\hat{\mathbf{b}}^*_u] = \mathbf{0} \tag{29}$$

where the capital letters Φ and Φ_u and the symbol $\hat{(\cdot)}$ over the variables indicate the assembled quantities. For a detailed explanation of the method with fixed time step, the interested reader is referred to literature.[6]

Variable Time of Flight SCP Algorithm

The original nonconvex low-thrust space trajectory optimization problem is solved by iteratively considering a sequence of convex subproblems. Each of these is first discretized using the HLGL method and then solved by means of a convex optimization solver. The idea of the SCP algorithm is to satisfy the nonlinear constraints using a trust-region-based method. In fact, the solution of each of the convex subproblems is kept close to the reference by means of a trust region. During the process, the trust-region radius is increased or decreased depending on the quality of the solution found at the current SCP iteration. The process continues until the violation of the nonlinear, nonconvex constraints is below a certain threshold ε_c. Detailed information of the SCP algorithm can be found in literature.[7] In this paper, we exploit the aforementioned mechanism to build the variable time of flight convex low-thrust space trajectory optimization problem.

The variable time of flight part of the SCP algorithm is summarized in Fig. 5. At the very first SCP iteration (see Fig. 5(a)), an initial guess is provided for both the trajectory and the time of flight, i.e., for the final boundary condition $\mathbf{x}^{(1)}_f$. In particular, the latter has been obtained by evaluating the interpolating polynomials $\mathbf{r}_{\mathrm{BC}}(t_{\mathrm{ARR}})$ and $\mathbf{v}_{\mathrm{BC}}(t_{\mathrm{ARR}})$. In fact, the final boundary condition

changes according to the insertion epoch of the S/C into the BCC. When the first convex optimization problem is considered, the final boundary condition is linearized and, in general, the optimized solution will have a different, unfeasible final boundary condition $\mathbf{x}_{f,LIN}^{(1)}$. The actual final boundary condition correspondent to the optimized time of flight is $\mathbf{x}_f^{(2)}$, found by evaluating the polynomials $\mathbf{r}_{\mathrm{BC}}$ and $\mathbf{v}_{\mathrm{BC}}$ at the new optimized time t_{ARR}. The difference

$$\Delta\mathbf{x}_f^{(1)} = \mathbf{x}_{f,LIN}^{(1)} - \mathbf{x}_f^{(2)} \tag{30}$$

defines how far the linearized solution is from the nonlinear one. The second iteration (see Fig. 5(b)), considers $\mathbf{x}_f^{(2)}$ as new reference final boundary condition. The process continues until the norm $||\Delta\mathbf{x}_f^{(k)}||_\infty$ is below the threshold ε_c (see Fig. 5(c)). The fixed time of flight version of the algorithm can be obtained by simply considering the trust-region radius R_{h} inside Eq. 21k such that $R_{\mathrm{h}} = 0$.

Note that the SCP algorithm only stops when all the nonlinear constraints are below the given threshold. Here, we reported the reasoning for the final boundary conditions constraint only.

Closed-Loop Guidance

The current paradigm of how space missions are operated foresees that engineers often check from ground the spacecraft status[43] and quickly react to any deviation from nominal conditions by sending new commands to be executed. On the contrary, an effective autonomous guidance system should be able to provide the onboard spacecraft computer with a new reference trajectory and control commands in real-time without any interaction with ground. It is paramount to ensure that the actual spacecraft trajectory does not differ too much from the nominal one to avoid potentially catastrophic consequences on the mission. The so-called closed-loop guidance approach has been proposed for the purpose. In particular, recent work[9] developed a closed-loop guidance algorithm in which the spacecraft trajectory is recomputed at predefined times during its interplanetary trajectory. In this paper, we build a closed-loop guidance algorithm that recomputes the trajectory whenever it deviates over a certain threshold with respect to the nominal one. This is supposed to be a more realistic and flexible approach.

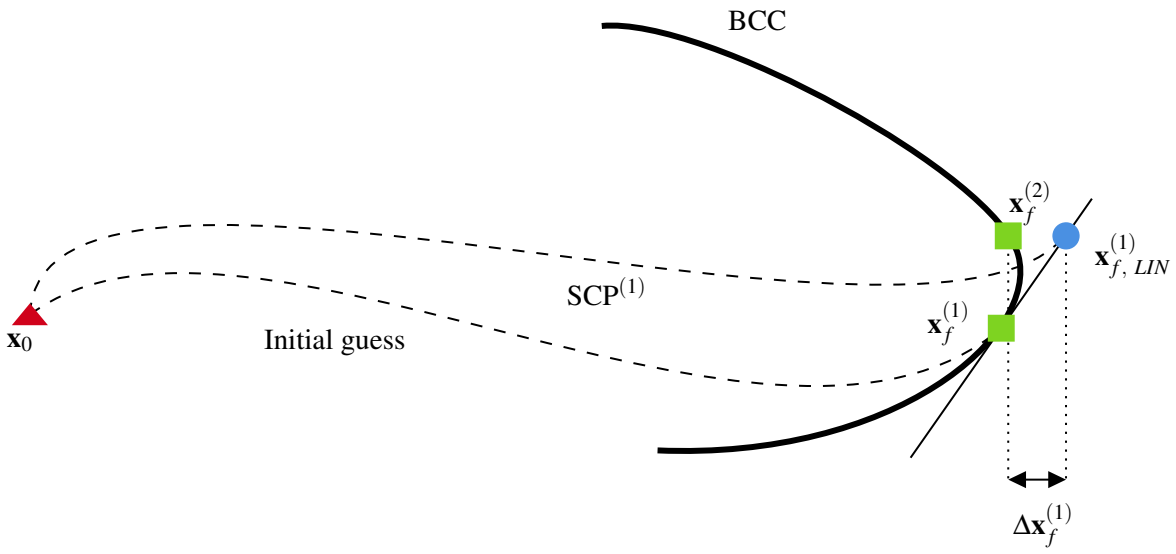
Figure 6 shows the complete structure of the algorithm. We provide a simple cubic-based initial guess[44] to compute the first reference trajectory. Subsequently, we simulate the actual trajectory followed by the spacecraft by propagating the equations of motion (EoM), with two-body dynamics and correlated process noises for the acceleration variables, modeled as Gauss–Markov processes.[11,45] In particular, we define the perturbed acceleration variable τ to be used for the propagation as

$$\tau(t) = \sigma\eta(t) + \bar{\tau} \tag{31}$$

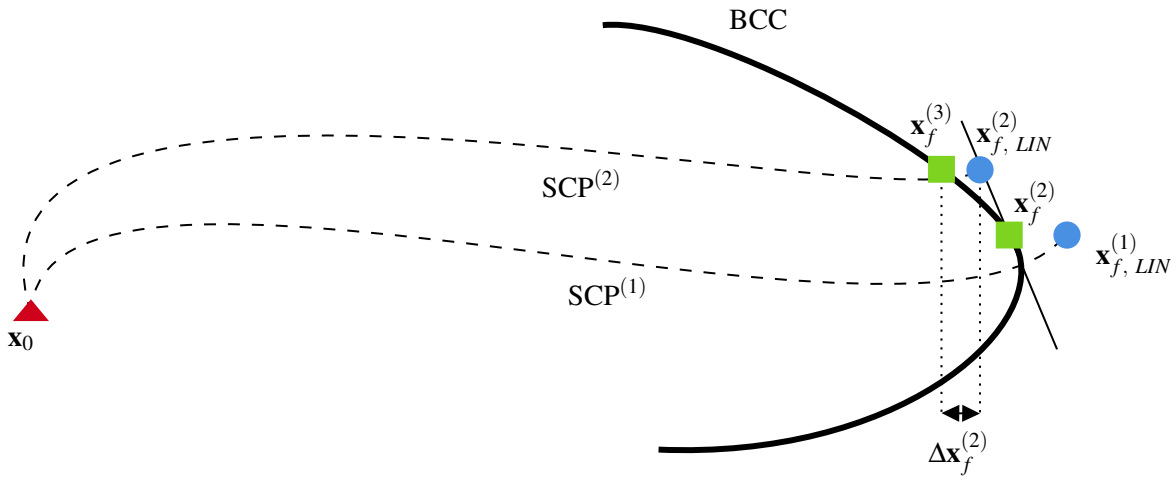
where

$$\eta(t) = \eta_k e^{-\delta(t-t_k)} + \sqrt{1-e^{-2\delta(t-t_k)}}\omega_k, \quad t \in [t_k, t_{k+1}]. \tag{32}$$

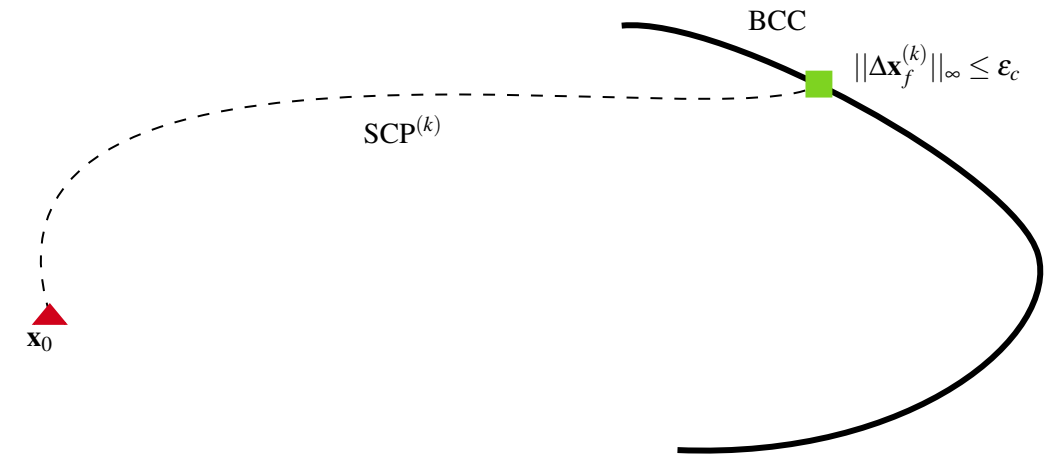
$\bar{\tau}$ is the unperturbed acceleration variable, σ is the variance of the process, δ is the inverse of the correlation time, and ω is the white noise, which is considered constant between the integration times t_k and t_{k+1}. Both η_k and ω_k are elements of random vectors defined before the propagation starts. Note that the acceleration variables τ_i, $i = x, y, z$ are defined accordingly.



(a) First Iteration of the Variable Time of Flight SCP Algorithm.



(b) Second Iteration of the Variable Time of Flight SCP Algorithm.



(c) kth Iteration of the Variable Time of Flight SCP Algorithm (Converged).

Figure 5. Variable Time of Flight Algorithm Explanation.

After having propagated the trajectory for the first of the K trajectory segments, the algorithm checks whether the normalized difference err_p between the propagated and reference states overcomes the selected threshold ε_p. We define err_p as

$$\text{err}_\text{p} = \frac{||\mathbf{x}(t_\text{p}) - \mathbf{x}^*(t_\text{p})||_2}{||\mathbf{x}^*(t_\text{p})||_2} \tag{33}$$

where t_p is the propagation time, $\mathbf{x}(t_\text{p})$ is the propagated state, and $\mathbf{x}^*(t_\text{p})$ the reference state. If err_p is under the threshold ε_p, the process is iterated over all other trajectory segments until either the threshold is overcame or the final target is reached. In the first case, $\mathbf{x}(t_\text{p})$ is then further perturbed to simulate the error introduced by the uncertainty of the navigation procedure. We introduce a random perturbation in the range of $[-10^3, 10^3]$ km on the position variables and in the range of $[-10^{-3}, 10^{-3}]$ km/s on the velocity variables, which are conservative values taken from literature on optical-based autonomous navigation systems.[46,47]

Since the time of flight changes after each time the trajectory is recomputed, we adjust the number of intervals K to be used for the next optimization such that the function $K(t_f)$ varies linearly with respect to the time of flight. In particular, we define

$$K(t_f) = K^{(0)} \frac{t_f}{t_f^{(0)}} \tag{34}$$

where $K^{(0)}$ is the number of trajectory segments correspondent to the time of flight $t_f^{(0)}$, associated with the first trajectory optimization process.

Finally, we update the new initial boundary condition for the next optimization process, as well as the new initial guess, which is considered as the trajectory computed at the previous optimization loop.

In Fig. 6 and in the rest of the paper, the term *assembled* is used to indicate the complete trajectory obtained by collecting the different optimized trajectories correspondent to a single iteration of the closed-loop guidance algorithm.

RESULTS

We compare the performance of the closed-loop guidance algorithm with fixed and variable time of flight in terms of final spacecraft mass, number of trajectory reoptimizations, and convergence rate. We use the Embedded Conic Solver (ECOS)[48] to solve the convex low-thrust space trajectory optimization problem. All simulations are performed in MATLAB© on an Intel Core i7-10510U CPU 2.30 GHz laptop with 16GB of RAM. Table 3 presents the SCP algorithm parameters; more information on them can be found in previous works.[5,6] Note that the parameters R_0 and $R_{h,0}$ have two values each. This is because they refer to the first and the following closed-loop optimization iterations, respectively. In particular, R_0 becomes much smaller after the first optimization process because the computed trajectory at previous iteration is used as initial guess for the next one. $R_{h,0}$ is instead set to 0 for the first optimization process to indicate that the fixed time of flight algorithm must be used. This is to ensure a fair comparison of the two versions of the algorithm. Table 5 shows the physical quantities that have been used to write the dimensionless version of the problem. Table 4 presents the closed-loop guidance algorithm parameters. The magnitude of Gauss–Markov variance σ has been considered as the 3% of the maximum engine thrust T_max. We have used the 7th-order HLGL discretization, as it proved effective to solve the low-thrust minimum-fuel space

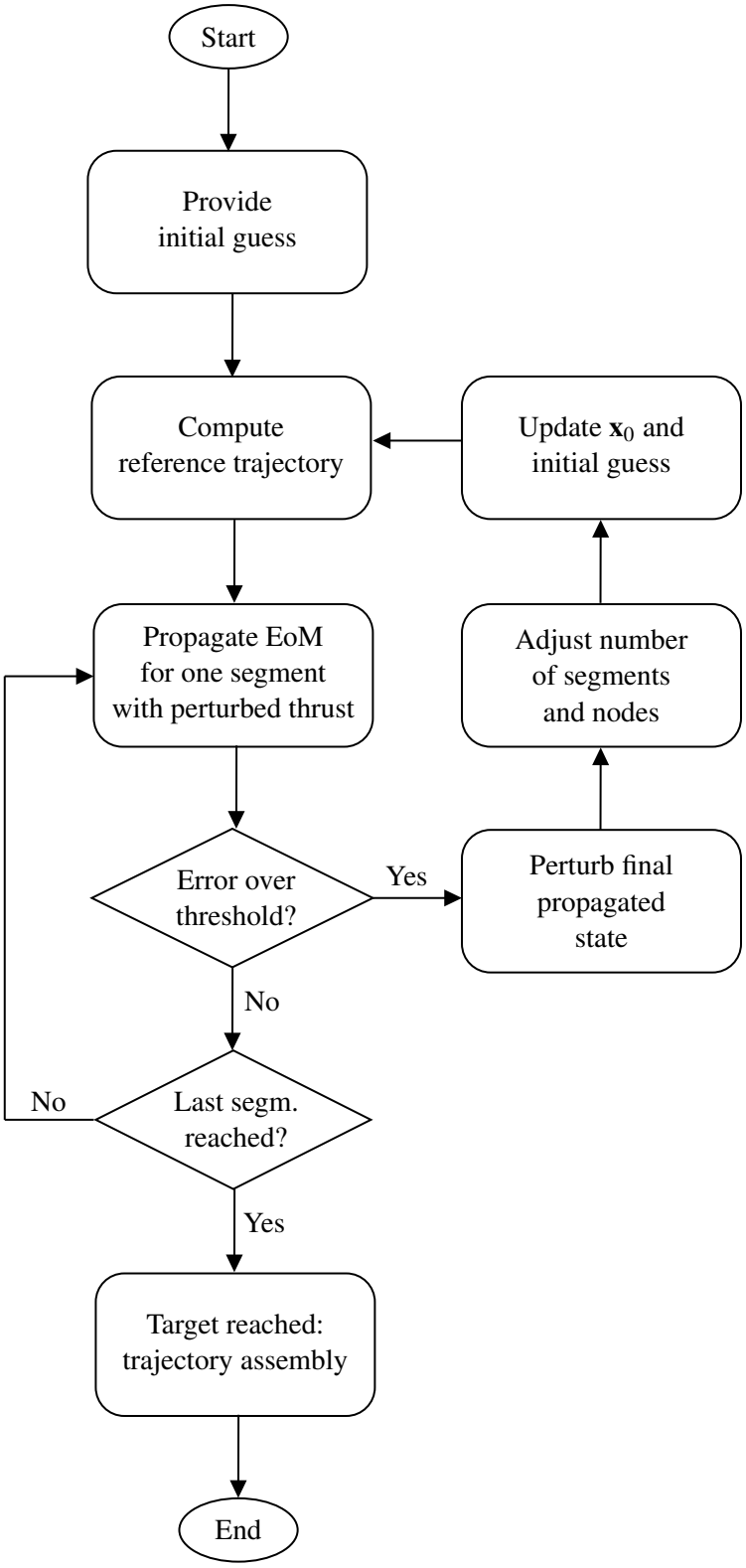


Figure 6. Structure of the Closed-Loop Guidance Algorithm.

Table 3. Parameters of the SCP Algorithm.

Parameter	Value
Penalty weight λ	10.0
Initial trust region R_0	100.0, 1.0
Initial trust region $R_{\mathrm{h},0}$	0.0, 1.0
ρ_0, ρ_1, ρ_2	0.01, 0.2, 0.85
α	1.5
β	1.5
ε_c, ε_ϕ, ε_x	10^{-6}, 10^{-3}, 10^{-7}
Max. iterations	250

Table 4. Parameters of the Closed-Loop Guidance Algorithm.

Parameter	Value
Gauss–Markov variance σ	6.7557×10^{-5} m/s^2
Inverse of corr. time δ	5.8132×10^{1} 1/s
Initial nr. of intervals $K^{(0)}$	83
Threshold ε_p	1×10^{-4}

trajectory optimization problem in previous works.[6,8] This results in a total of 250 trajectory nodes when a number of intervals $K^{(0)} = 83$ is chosen. Note however that both the number of intervals and nodes change for each of the reoptimization processes, according to Eq. 34. This, in turn, results in an uniformly discretized assembled trajectory and thrust profile.

Table 5. Physical Quantities for the Results Section.

Physical Quantity	Unit	Normalization Factor	Value
Gravitational parameter	km^3/s^2	μ	1.3271244×10^{11}
Gravitational accel.	km/s^2	g_0	9.80665×10^{-3}
Length	km	LU = AU	$1.495978707 \times 10^{8}$
Velocity	km/s	VU	$\sqrt{\mu/\mathrm{LU}}$
Time	s	TU	LU/VU
Acceleration	km/s^2	ACU	VU/TU
Mass	kg	MU $= m_0$	22.6
Max. thrust	N	$T_{\max}$	2.2519×10^{-3}
Specific impulse	s	I_{sp}	3067

Closed-Loop Guidance Simulations

Table 6 presents the transfer data that have been used to perform the simulations. All data are in the J2000 reference frame. We select 5 different nominal time of flights $t_f^{(0)}$ for the guidance part of the transfer, uniformly distributed between 1150 and 2150 days. These values are typical for CubeSats' low-thrust transfers involving ballistic capture arcs at Mars.[34] The departure time t_{DEP} has been selected such that $t_{\mathrm{C}}^0 - t_{\mathrm{DEP}} > 1150$ days. $\mathbf{r}_0$ and $\mathbf{v}_0$ correspond to the position and velocity of the Earth at departure date. Figure 7 represents the trajectory extracted from the selected ballistic capture corridor, together with the 5 nominal final boundary conditions for the transfer. It can be observed that they are equally distributed and roughly cover the whole length of the trajectory. The initial and final points of the ballistic capture corridor are also indicated.

We perform a robustness analysis for both the fixed and variable time of flight algorithms. We run the closed-loop guidance algorithm 100 times for each of the 5 nominal $t_f^{(0)}$, resulting in a total

Table 6. Data of the Different Transfers. $t_f^{(0)}$ is in Days.

Param. \ $t_f^{(0)}$	1150	1400	1650	1900	2150
Dep. date	June 1, 2023 at 00:00:00.0 (TDB)				
$\mathbf{r}_0$ (LU)	$[-0.347456, -0.873913, -0.378831]^\top$				
$\mathbf{v}_0$ (VU)	$[0.923319, -0.317902, -0.137843]^\top$				
$\mathbf{r}_f^{(0)}$ (LU)	$[1.247875, 0.582523, 0.237669]^\top$	$[-1.477285, 0.645577, -0.328939]^\top$	$[0.182593, 1.281716, -0.590050]^\top$	$[0.469847, 1.294358, 0.580878]^\top$	$[-1.604446, -0.237482, -0.071788]^\top$
$\mathbf{v}_f^{(0)}$ (VU)	$[-0.340328, 0.729903, -0.341432]^\top$	$[-0.332357, 0.605561, -0.269182]^\top$	$[0.841101, 0.169986, -0.058427]^\top$	$[-0.749805, 0.294968, 0.152009]^\top$	$[0.152322, -0.668784, -0.309186]^\top$

of 1000 simulations. Table 7 presents the obtained results. The two versions of the algorithm are compared in terms of average final spacecraft mass, number of interplanetary trajectory segments *S*, and convergence rate *Cv*. The number of interplanetary segments *S* corresponds to the number of times the trajectory has been recomputed. The last column of the table indicates the increase (or, in general, the decrease) of the time of flight associated with the variable time of flight algorithm. It can be noted that the algorithm does not influence either the value of the final spacecraft mass or the number of trajectory segments. On the contrary, the variable time of flight algorithm outperformed the fixed t_f version in terms of converged cases, the former resulting in double the convergence rate of the latter. On average, the increase of the time of flight for the variable t_f algorithm was 25 days. The average final S/C mass is around $17.7 - 17.8$ kg for both versions of the algorithm, in accordance with the values found in literature.[34]

Figure 8 shows the comparison of typical transfer trajectories obtained with the fixed and variable time of flight algorithms for $t_f^{(0)} = 1150$ and 1400 days. It can be noted that the final boundary conditions and the assembled trajectories look similar for the two algorithms when $t_f^{(0)} = 1150$ days because the average Δt_f is $+4.1$ days only. On the contrary, they are significantly different in the case of $t_f^{(0)} = 1400$ since $\Delta t_f = +59.0$ days.

Figure 9 shows the comparison of the nominal (i.e., when no closed-loop guidance is considered) thrust profile and the typical ones obtained with the closed-loop guidance algorithm with fixed and variable time of flight for the case $t_f^{(0)} = 1400$. The variable time of flight thrust profile clearly lasts longer than the other two.

Figure 10 shows the average CPU time required to solve the convex low-thrust space trajectory optimization problem as a function of the reoptimization number for both the fixed and the variable time of flight algorithms. Both of them follow the same trend: the CPU time is approximately 60 seconds for the first optimization, and decreases by several orders of magnitudes at the end of the interplanetary trajectory. This is because of two reasons: first, the time of flight gradually decreases as the S/C approaches the BCC, so the number of discretization points is lower. Moreover, after the first optimization process, a very accurate initial guess is provided to the algorithm for the next trajectory segment. This significantly decreases the CPU time required to reach convergence.

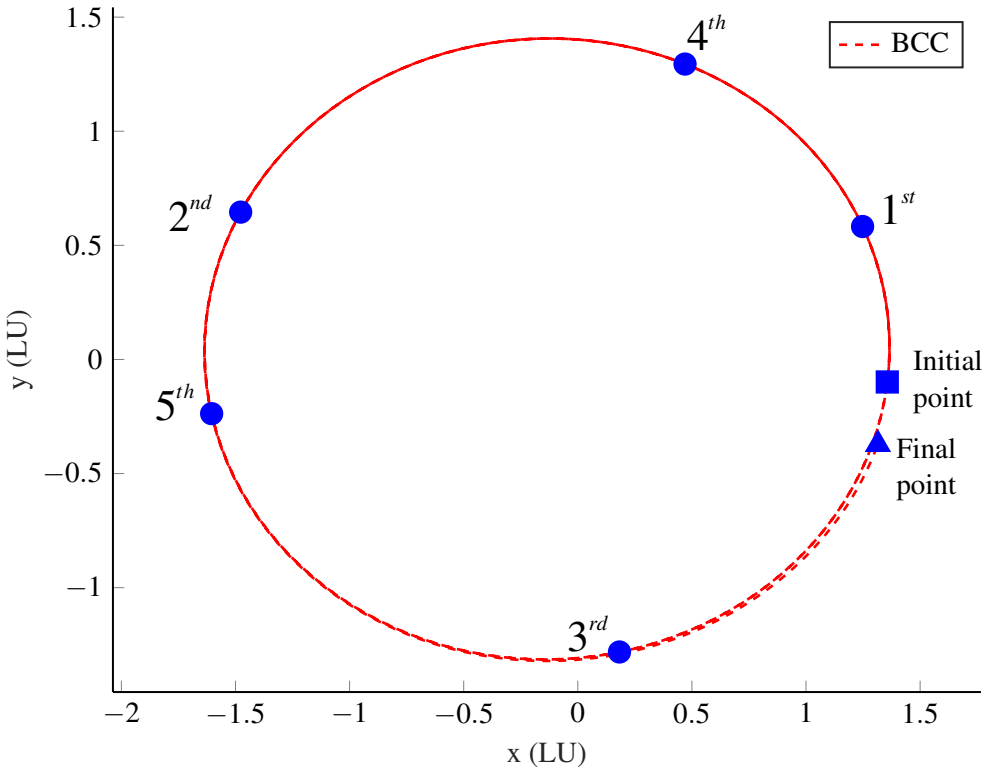


Figure 7. BCC Trajectory With Nominal Target Points Indicated With Ordinal Numbers (in the J2000 reference frame).

Finally, it is also interesting to note that the variable time of flight algorithm requires a slightly higher CPU time.

CONCLUSION

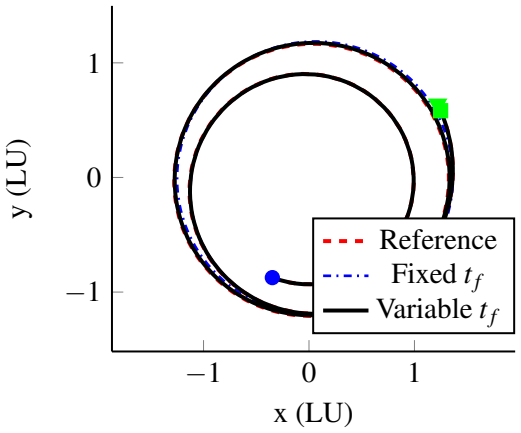
In this paper, a convex guidance approach to target ballistic capture corridors was presented. It collocates in the context of autonomous guidance, navigation, and control of interplanetary CubeSats. We developed a convex guidance algorithm based on the Hermite–Legendre–Gauss–Lobatto discretization scheme, able to solve variable time of flight low-thrust trajectory optimization problems. Our algorithm shows superior performance in terms of converged cases with respect to classical fixed time of flight convex optimization strategies to solve the same problem. For this reason, our algorithm represents an ideal choice for onboard low-thrust trajectory optimization to ballistic capture corridors at Mars, since the target final boundary condition is a trajectory rather than a celestial body.

ACKNOWLEDGMENT

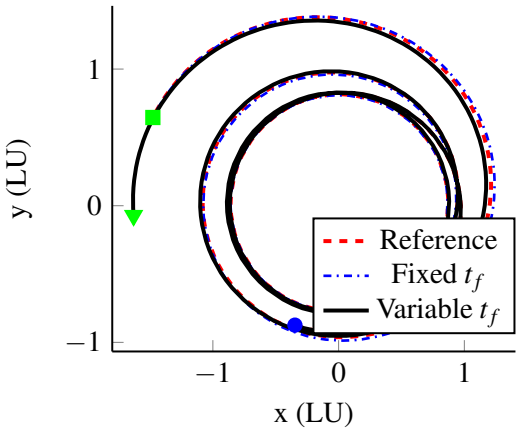
This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 864697).

Table 7. Comparison of Fixed and Variable Time of Flight Algorithms.

Parameter	**Fixed t_f**			**Variable t_f**			
$t_f^{(0)}$ (days)	m_f (kg)	S	$Cv.$ (%)	m_f (kg)	S	$Cv.$ (%)	Δt_f (days)
1150	18.4	23.3	7.0	18.4	21.9	39.0	+4.1
1400	16.1	30.8	59.0	16.2	32.1	52.0	+59.0
1650	18.0	36.3	3.0	18.1	34.9	43.0	+25.5
1900	18.8	31	24.0	18.8	31.3	27.0	+3.1
2150	17.3	39.6	11.0	17.3	39.9	55.0	+33.4
Average	17.7	32.2	20.8	17.8	32.0	43.2	+25.0



(a) Comparison of reference trajectory, fixed, and variable time of flight assembled trajectories for $t_f^{(0)} = 1150$ days.



(b) Comparison of reference trajectory, fixed, and variable time of flight assembled trajectories for $t_f^{(0)} = 1400$ days.

Figure 8. Comparison of Trajectories Obtained With the Closed-Loop Guidance Algorithm (in the J2000 Reference Frame).

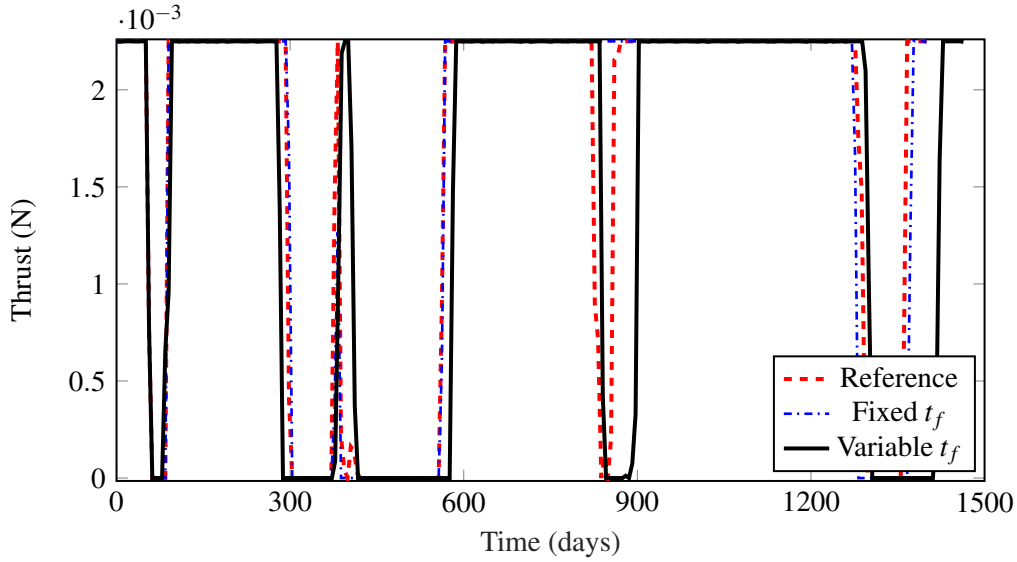
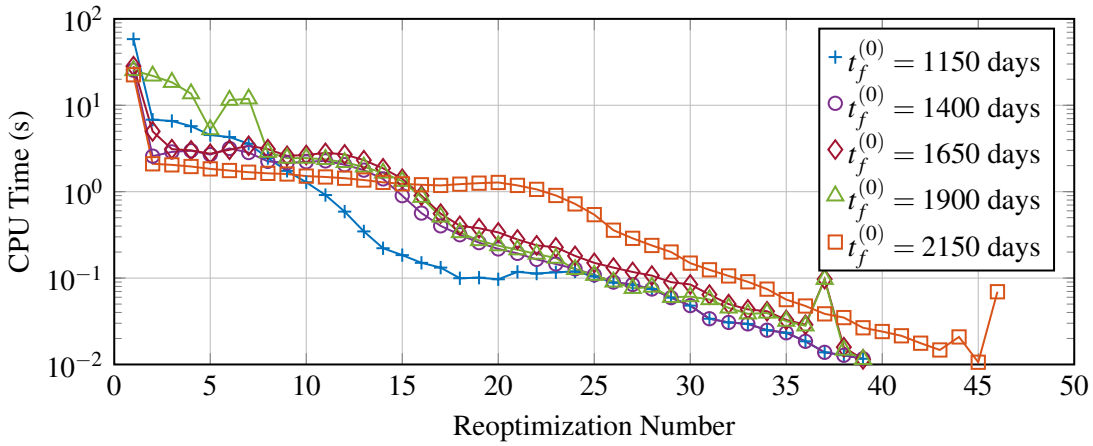
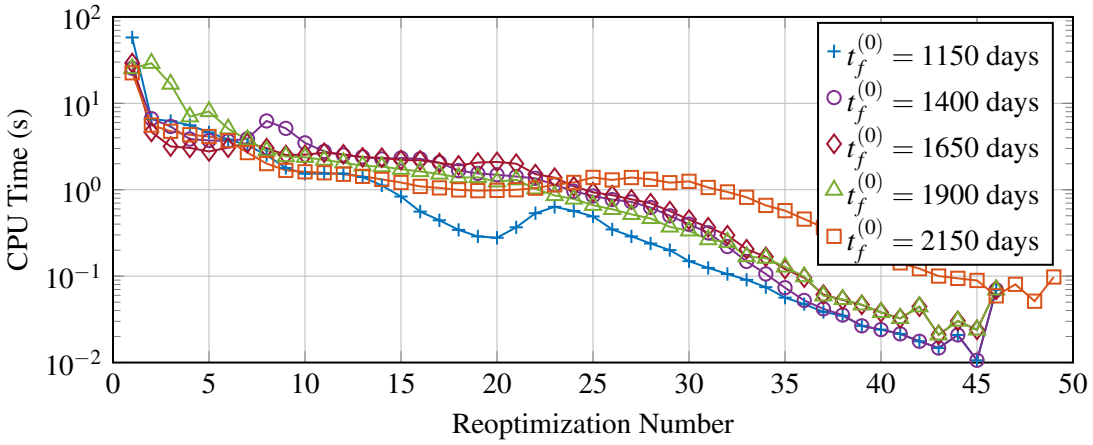


Figure 9. Comparison of Reference Thrust Profile, Fixed and Variable Time of Flight Assembled Thrust Profiles for $t_f^{(0)} = 1400$ Days.



(a) Fixed Time of Flight Algorithm.



(b) Variable Time of Flight Algorithm.

Figure 10. CPU Times as Function of the Reoptimization Number.

REFERENCES

[1] K. Woellert, P. Ehrenfreund, A. Ricco, and H. Hertzfeld, "Cubesats: Cost-effective science and technology platforms for emerging and developing nations," *Advances in Aerospace Research*, Vol. 47, No. 4, 2011, pp. 663–684, 10.1016/j.asr.2010.10.009.

[2] A. Klesh and J. Krajewski, "MarCO: Mars Cube One – Lessons Learned from Readying the First Interplanetary Cubesats for Flight," *49th Lunar and Planetary Science Conference*, 2018.

[3] G. Di Domenico, E. Andreis, A. C. Morelli, G. Merisio, V. Franzese, C. Giordano, A. Morselli, P. Panicucci, F. Ferrari, and F. Topputo, "Toward Self-Driving Interplanetary CubeSats: the ERC-Funded Project EXTREMA," *72nd International Astronautical Congress (IAC 2021)*, 2021, pp. 1–11.

[4] Z. Wang and M. J. Grant, "Minimum-Fuel Low-Thrust Transfers for Spacecraft: A Convex Approach," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54, No. 5, 2018, pp. 2274–2290, 10.1109/TAES.2018.2812558.

[5] C. Hofmann and F. Topputo, "Rapid Low-Thrust Trajectory Optimization in Deep Space Based On Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 7, 2021, pp. 1379–1388. 10.2514/1.G005839.

[6] A. C. Morelli, C. Hofmann, and F. Topputo, "Robust Low-Thrust Trajectory Optimization Using Convex Programming and a Homotopic Approach," *IEEE Transactions on Aerospace and Electronic Systems*, 2021. Available online, 10.1109/TAES.2021.3128869.

[7] Y. Mao, M. Szmuk, X. Xu, and B. Açıkmeşe, "Successive Convexification: A Superlinearly Convergent Algorithm for Non-convex Optimal Control Problems," `https://arxiv.org/abs/1804.06539`, Preprint, submitted February 2019.

[8] C. Hofmann, A. C. Morelli, and F. Topputo, "On the Performance of Discretization and Trust-Region Methods for On-Board Convex Low-Thrust Trajectory Optimization," *AIAA SCITECH 2022 Forum*, 2022, p. 1892.

[9] C. Hofmann and F. Topputo, "Closed-Loop Guidance for Low-Thrust Interplanetary Trajectories Using Convex Programming," *11th International ESA Conference on Guidance, Navigation & Control Systems*, 2021, pp. 1–15.

[10] B. Schutz, B. Tapley, and G. H. Born, *Statistical orbit determination*. Elsevier, 2004.

[11] D. A. Dei Tos, M. Rasotto, F. Renk, and F. Topputo, "LISA Pathfinder mission extension: A feasibility analysis," *Advances in Space Research*, Vol. 63, No. 12, 2019, pp. 3863–3883.

[12] E. A. Belbruno and J. K. Miller, "Sun-perturbed Earth-to-moon transfers with ballistic capture," *Journal of Guidance, Control, and Dynamics*, Vol. 16, jul 1993, pp. 770–775. 10.2514/3.21079.

[13] E. Belbruno and J. Carrico, "Calculation of weak stability boundary ballistic lunar transfer trajectories," *Astrodynamics Specialist Conference*, 2000, p. 4142. 10.2514/6.2000-4142.

[14] F. Topputo and E. Belbruno, "Earth–Mars transfers with ballistic capture," *Celestial Mechanics and Dynamical Astronomy*, Vol. 121, No. 4, 2015, pp. 329–346. 10.1007/s10569-015-9605-8.

[15] C. Circi and P. Teofilatto, "On the dynamics of weak stability boundary lunar transfers," *Celestial Mechanics and Dynamical Astronomy*, Vol. 79, No. 1, 2001, pp. 41–72. 10.1023/A:1011153610564.

[16] V. V. Ivashkin, "On trajectories of Earth-Moon flight of a particle with its temporary capture by the Moon," *Doklady Physics*, Vol. 47, Springer, 2002, pp. 825–827. 10.1134/1.1526433.

[17] E. A. Belbruno and J. Miller, "A ballistic lunar capture trajectory for the Japanese spacecraft hiten," tech. rep., 1990. IOM 312/904-1731-EAB.

[18] G. D. Racca, A. Marini, L. Stagnaro, J. Van Dooren, L. Di Napoli, B. H. Foing, R. Lumb, J. Volp, J. Brinkmann, R. Grünagel, *et al.*, "SMART-1 mission description and development status," *Planetary and space science*, Vol. 50, No. 14-15, 2002, pp. 1323–1337. 10.1016/S0032-0633(02)00123-X.

[19] M. J. Chung, S. J. Hatch, J. A. Kangas, S. M. Long, R. B. Roncoli, and T. H. Sweetser, "Trans-lunar cruise trajectory design of GRAIL (Gravity Recovery and Interior Laboratory) mission," *AIAA/AAS Astrodynamics Specialist Conference*, 2010, p. 8384. 10.2514/6.2010-8384.

[20] R. Jehn, S. Campagnola, D. Garcia, and S. Kemble, "Low-thrust approach and gravitational capture at Mercury," *18th International Symposium on Space Flight Dynamics*, Vol. 548, 2004, p. 487.

[21] J. Elliott and L. Alkalai, "Lunette: A network of lunar landers for in-situ geophysical science," *Acta Astronautica*, Vol. 68, No. 7-8, 2011, pp. 1201–1207. 10.1016/j.actaastro.2010.10.024.

[22] M. Vetrisano, W. V. d. Weg, and M. Vasile, "Navigating to the Moon along low-energy transfers," *Celestial Mechanics and Dynamical Astronomy*, Vol. 114, No. 1, 2012, pp. 25–53. 10.1007/s10569-012-9436-9.

[23] F. Topputo and E. Belbruno, "Computation of weak stability boundaries: Sun–Jupiter system," *Celestial Mechanics and Dynamical Astronomy*, Vol. 105, No. 1-3, 2009, p. 3. 10.1007/s10569-009-9222-5.

[24] G. Merisio and F. Topputo, "Characterization of ballistic capture corridors aiming at autonomous ballistic capture at Mars," *2021 AAS/AIAA Astrodynamics Specialist Conference*, 2021, pp. 1–21.

[25] W. Zhenbo and J. M. Grant, "Optimization of Minimum-Time Low-Thrust Transfers Using Convex Programming," *Journal of Spacecraft and Rockets*, Vol. 55, No. 1, 2017, pp. 1–13, 10.2514/1.A33995.

[26] N. Hyeraci and F. Topputo, "Method to design ballistic capture in the elliptic restricted three-body problem," *Journal of guidance, control, and dynamics*, Vol. 33, No. 6, 2010, pp. 1814–1823. 10.2514/1.49263.

[27] Z.-F. Luo, F. Topputo, F. Bernelli Zazzera, and G. J. Tang, "Constructing ballistic capture orbits in the real Solar System model," *Celestial Mechanics and Dynamical Astronomy*, Vol. 120, No. 4, 2014, pp. 433–450. 10.1007/s10569-014-9580-5.

[28] D. A. Dei Tos, R. P. Russell, and F. Topputo, "Survey of Mars ballistic capture trajectories using periodic orbits as generating mechanisms," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 6, 2018, pp. 1227–1242. 10.2514/1.g003158.

[29] Z. F. Luo and F. Topputo, "Capability of satellite-aided ballistic capture," *Communications in Nonlinear Science and Numerical Simulation*, Vol. 48, 2017, pp. 211–223. 10.1016/j.cnsns.2016.12.021.

[30] Z.-F. Luo and F. Topputo, "Analysis of ballistic capture in Sun–planet models," *Advances in Space Research*, Vol. 56, No. 6, 2015, pp. 1030–1041. 10.1016/j.asr.2015.05.042.

[31] R. S. Park, W. M. Folkner, J. G. Williams, and D. H. Boggs, "The JPL Planetary and Lunar Ephemerides DE440 and DE441," *The Astronomical Journal*, Vol. 161, No. 3, 2021, p. 105. 10.3847/1538-3881/abd414.

[32] C. Huang, J. C. Ries, B. D. Tapley, and M. M. Watkins, "Relativistic effects for near-earth satellite orbit determination," *Celestial Mechanics and Dynamical Astronomy*, Vol. 48, No. 2, 1990, pp. 167–185, 10.1007/BF00049512.

[33] F. Topputo, Y. Wang, C. Giordano, V. Franzese, H. Goldberg, F. Perez-Lissi, and R. Walker, "Envelop of reachable asteroids by M-ARGO CubeSat," *Advances in Space Research*, Vol. 67, No. 12, 2021, pp. 4193–4221. 10.1016/j.asr.2021.02.031.

[34] G. Aguiar and F. Topputo, "A Technique for Designing Earth-Mars Low-Thrust Transfers Culminating in Ballistic Capture," *7th International Conference on Astrodynamics Tools and Techniques (ICATT)*, 2018, pp. 1–8.

[35] R. G. Gottlieb, "Fast gravity, gravity partials, normalized gravity, gravity gradient torque and magnetic field: Derivation, code and data," tech. rep., 1993. 188243, prepared for Lyndon B. Johnson Space Center under contract NAS9-17885.

[36] C. H. Acton Jr, "Ancillary data services of NASA's navigation and ancillary information facility," *Planetary and Space Science*, Vol. 44, No. 1, 1996, pp. 65–70. 10.1016/0032-0633(95)00107-7.

[37] C. Acton, N. Bachman, B. Semenov, and E. Wright, "A look towards the future in the handling of space science mission geometry," *Planetary and Space Science*, Vol. 150, 2018, pp. 9–12. 10.1016/j.pss.2017.02.013.

[38] F. Topputo, D. A. Dei Tos, K. V. Mani, S. Ceccherini, C. Giordano, V. Franzese, and Y. Wang, "Trajectory design in high-fidelity models," *7th International Conference on Astrodynamics Tools and Techniques (ICATT)*, 2018, pp. 1–9.

[39] O. Montenbruck and E. Gill, *Satellite Orbits Models, Methods and Applications*. Springer, 2000. 10.1007/978-3-642-58351-3.

[40] P. J. Prince and J. R. Dormand, "High order embedded Runge-Kutta formulae," *Journal of computational and applied mathematics*, Vol. 7, No. 1, 1981, pp. 67–75. 10.1016/0771-050x(81)90010-3.

[41] P. J. Mohr, D. B. Newell, and B. N. Taylor, "CODATA recommended values of the fundamental physical constants: 2014," *Journal of Physical and Chemical Reference Data*, Vol. 45, No. 4, 2015, p. 043102, 10.6028/nist.sp.961r2015.

[42] P. Williams, "Hermite–Legendre–Gauss–Lobatto Direct Transcription in Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 4, 2009, pp. 1392–1395, 10.2514/1.42731.

[43] C. Steiger, E. Montagnon, A. Accomazzo, and P. Ferri, "BepiColombo mission to Mercury: First year of flight," *Acta Astronautica*, Vol. 170, 2020, pp. 472–479.

[44] E. Taheri and O. Abdelkhalik, "Initial Three-Dimensional Low-Thrust Trajectory Design," *Advances in Space Research*, Vol. 57, No. 3, 2016, pp. 889 – 903, 10.1016/j.asr.2015.11.034.

[45] C. Giordano, *Analysis, Design, and Optimization of Robust Trajectories for Limited-Capability Small Satellites*. PhD thesis, Politecnico di Milano, 2021.

[46] E. Andreis, V. Franzese, and F. Topputo, "An Overview of Autonomous Optical Navigation for Deep-Space CubeSats," *72nd International Astronautical Congress (IAC 2021)*, 2021, pp. 1–11.

[47] E. Andreis, V. Franzese, and F. Topputo, "On-board Orbit Determination for Deep-Space CubeSats," *Journal of Guidance Control and Dynamics*, Manuscript under review.

[48] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP Solver for Embedded Systems," *European Control Conference*, Jul. 2013, pp. 3071–3076.