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

IAC-24,D1,IPB,1,x85603

An overview of EXTREMA project Pillar II: validating autonomous guidance algorithms via HIL testing

Alessandra Mannocchi[a,*], Davide Perico[a], Gonçalo Oliveira Pinho[a], Gianfranco Di Domenico[a], Alessandro Morselli[a], Francesco Topputo[a]

[a] *Dept. of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy*, alessandra.mannocchi@polimi.it

Over the past twenty years, CubeSats, shoebox-sized satellites, have transformed the way we explore space, opening low-Earth orbit to a growing number of participants. This trend is expected to extend to interplanetary missions, as demonstrated by NASA's deployment of two MarCO CubeSats near Mars in 2018, and by two ESA-funded missions M-ARGO, aiming at rendezvous with an asteroid, and SATIS, for close approach with Apophis asteroid. However, a significant obstacle to further exploration of deep space lies in the ground segment: according to the state of the art, operating, navigating, and controlling interplanetary CubeSats requires the expertise of flight dynamics specialists and the exploitation of limited ground assets, as traditional and larger spacecraft.
In 2020, the European Research Council funded EXTREMA (Engineering Extremely Rare Events in Astrodynamics for Deep-Space Missions in Autonomy), a project aiming towards a paradigm shift of guidance, navigation, and control (GNC) operations from ground to onboard CubeSats. The goal is to enable autonomous miniaturized spacecraft, capable of traveling in deep space without requiring any interaction from the ground. The project is planned to last until 2025, and it is based on three fundamental pillars: Pillar I: Autonomous Navigation, Pillar II: Autonomous Guidance and Control, and Pillar III: Autonomous Ballistic Capture. The outcome of each pillar will be integrated into a series of experiments and brought together in the EXTREMA Simulation Hub (ESH): a hardware-in-the-loop (HIL) testing facility that will allow testing integrated GNC.
This work presents the state of the art of Pillar II at the beginning of the fourth year of the project. Two autonomous guidance algorithms, one direct-based and one indirect-based, have been developed and tested, and are here presented. Their deployment on representative onboard processors is discussed, along with the challenges that this process requires. The integration of the algorithms in the ESH is discussed, focusing on the interfaces with ETHILE, the Extrema Thruster in The Loop Experiment, and the facility modeling the real actuation of low-thrust propulsion systems. ETHILE measures the produced thrust and feds measurements to a high-fidelity numerical propagator. This enables simulations of whole mission profiles to be performed in an accelerated framework. Some preliminary simulations and their corresponding results will be presented thereafter, showcasing how autonomous guidance algorithms can be validated with the proposed HIL approach. To conclude, an overview of the future steps and developments are discussed.

Keywords: EXTREMA, Autonomous Guidance; HIL tests

1 Introduction

The number of nanosatellite launches doubled in 2021 and 2022 with respect to the 2020 ones. For 2024 almost 700 launches are planned, confirming the involvement of a rising number of players in the space sector [4] (see Fig. 1 for the detailed trend). The last years have thus seen the space sector experiencing thriving growth. A considerable role in this growth is represented by the significant decrease in access costs to space. Investments in the field and technological advances led to lower and lower budgets needed to carry on a space mission: in particular, CubeSats have enabled parties not backed up by huge capitals (such as universities or smaller private companies) to space thanks to their low design, manufacturing, and launch costs [1, 2]. Furthermore,

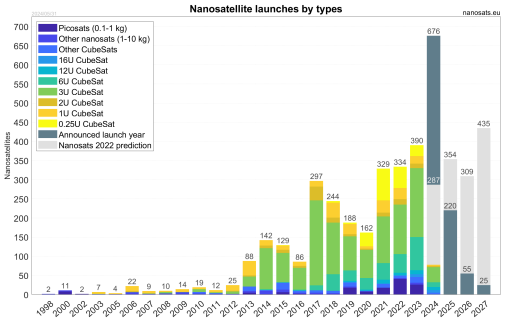

Fig. 1: Nanosatellites launches per year.[§]

by relying on off-the-shelf components, CubeSats also need fewer testing and validation procedures, cutting down costs and time-to-flight even more. It should be noted, however, that the space sector expansion was biased towards that part of space closer to the Earth.

[4] www.nanosats.eu Last access on 7th September 2024

Only a tiny fraction of the total space launches targeted interplanetary orbits. As longer duration characterizes deep-space missions, the great advantages brought by CubeSats technology are hindered by the extensive resources - both in terms of budget and human personnel - required to sustain the human-in-the-loop ground operations during multiple months or years. Moreover, no matter the available budget: ground slots for communications are scarce and are expected to saturate soon. In this framework, deep space is - and will be - a prerogative of a few more prominent stakeholders and agencies. The EXTREMA project (Engineering Extremely Rare Events in Astrodynamics for Deep-Space Missions in Autonomy) [3] aims to steer from such a future scenario by triggering a paradigm shift, enabling deep-space spacecraft with autonomous guidance, navigation, and control capabilities. The project awarded a five-year grant from the European Research Council, is planned to last until 2025, and it is based on three fundamental pillars:

- **Pillar I: Autonomous Navigation**: focuses on the development of navigation algorithms to enable CubeSats to locate themselves in deep space in complete autonomy by exploiting information in the surrounding environment.
- **Pillar II: Autonomous Guidance and Control**: aims to directly shift the current guidance paradigm. As of today, trajectory planning is performed on the ground due to the limited computational resources available on board. Correction maneuvers have to be planned from the ground too, employing a great amount of resources in terms of time and human personnel. EXTREMA aims to develop lightweight and robust closed-loop low-thrust guidance algorithms, exploiting the knowledge of the spacecraft position to compute a new trajectory to achieve mission objectives with complete autonomy.
- **Pillar III: Autonomous Ballistic Capture**: the limited resources characterizing CubeSats systems represent a bottleneck in achieving specific mission objectives as, for instance, expensive orbit insertion maneuvers. Because of this, EXTREMA aims to further develop the autonomy of deep-space probes by engineering ballistic capture, exploiting the multi-body dynamics of the Solar System to remain in the proximity of the target body for a prolonged period of time.

The outcome of each pillar will be integrated into a series of experiments and, brought together in the EXTREMA Simulation Hub (ESH) [4]: a hardware-in-the-loop (HIL) testing facility that will allow testing integrated GNC (Fig. 2). The facility under construction in the DART Laboratory [5] will integrate three different facilities under a comprehensive HIL simulation framework:

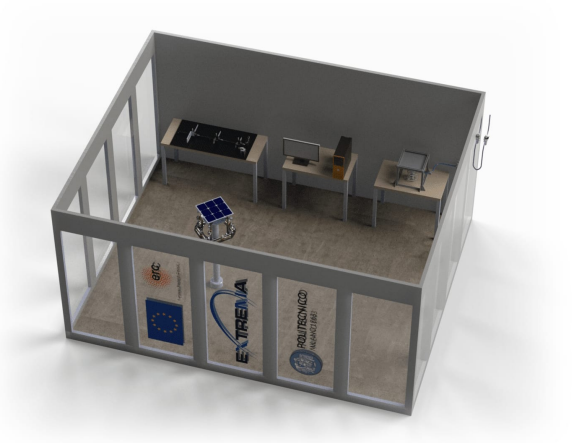

Fig. 2: ESH 3D rendering.

- **RETINA:** the Realistic Experimental facility for vision-based NAvigation [5] is an optical facility that will simulate the light pattern as received by the spacecraft optical camera through a set of lenses and screens. The output image will be employed to test and validate optical navigation algorithms based on image processing of deep-space starfields.
- **ETHILE:** the EXTREMA THruster In the Loop Experiment [6] is a cold-gas thrust test bench that will mimic the thruster in the spacecraft. In order to allow the simulation of multiple types of thrusters, a scaling framework based on dynamic similarity is employed to map the physical parameters of ETHILE to the ones of the target thruster.
- **STASIS:** the SpacecrafT Attitude SImulation System [7] is an air-bearing platform used to simulate the attitude evolution of a spacecraft in deep space. STASIS will also host the board representing the onboard computer of the spacecraft and the set of attitude sensors and actuators to be employed on the spacecraft. It features a set of moving masses, a wireless power generation system, and a set of additional attitude actuators to compensate for the difference in inertial properties between the platform and the spacecraft.

The aim of the ESH is to perform HIL simulations of interplanetary transfers under an accelerated framework. To do so, the onboard computer on STASIS platform runs the autonomous guidance algorithm and sends the computed thrusting profile to ETHILE.

This work presents the state of the art of Pillar II at the beginning of the fourth year of the project. Two autonomous guidance algorithms, one convex-based and one indirect-based, have been developed and tested,

[5] dart.polimi.it Last access on 7th September 2024

and are here presented. Their deployment on representative onboard processors is discussed, along with the challenges that this process requires. The integration of the algorithms in the ESH is discussed, focusing on the interfaces with ETHILE and the facility modeling the real actuation of low-thrust propulsion systems. Some preliminary simulations and their corresponding results will be presented thereafter, showcasing how autonomous guidance algorithms can be validated with the proposed HIL approach.

The paper is structured as follows. Section 2 provides an overview of the two algorithms developed within EXTREMA. Section 3 describes the facilities, the setup of the simulations, and the scenarios used in the simulations. Section 4 details the results obtained within the facility. Finally, Section 5 provides an overview of the future steps and a discussion on further developments.

2 Autonomous Guidance Algorithms

Computing the thrusting profile for a deep-space CubeSat implies the solution of a low-thrust trajectory optimization problem. No analytic solutions involving an acceptable level of assumptions exist for this kind of problem, even under the two-body dynamics. For this reason, several numerical techniques have been developed. State-of-the-art divides these guidance techniques into indirect and direct approaches [8, 9]. Direct methods [10–13] transform the continuous optimal control problem into a parameter optimization problem discretizing the time domain and then transcribing the dynamics and the other constraints in a set of equality constraints. Indirect methods [14–16] employ the necessary conditions equations obtained from the calculus of variation, requiring an explicit computation of them. Due to the nature of these equations, indirect methods lead to the exact solution of the optimal control problem. They are therefore characterized by high precision, but they have a very small convergence domain, meaning that if they are fed with a poor initial guess, they hardly converge.

Solving the low-thrust space trajectory optimization problem on board poses several mandatory challenges. The requirements for a suitable real-time guidance algorithm include:

- **reliability**: the capability of converging even when poor initial guesses are provided,
- **sustainability**: on board, due to the limited resources, especially for CubeSats,
- **optimality**: the ability to minimize the fuel consumption.

Having a wider converge region, direct methods are more reliable than indirect ones, even if to fully capture the dynamics of the problem they usually require tons of variables, and thus they not sustainable for onboard applications. However, among direct methods, convex optimization [17, 18] represents an interesting approach as it assures robustness. For this reason, it is usually selected for autonomous algorithms [19–25], none of which involve the deep-space scenario though. However, this comes at the cost of an approximated solution: they are not always able to catch a perfect bang-bang thrusting profile for a fuel-optimal problem, meaning that they usually catch a sub-optimal solution to the problem. Moreover, as with most of the direct methods, convex suffers from memory issues when several nodes are considered. On the other hand, indirect methods provide the correct solution to the problem, but historically there are two major difficulties slowing down their development for nowadays problems [8]. Namely, these are the difficult derivation of the necessary conditions for optimality and the necessity of a good initial guess for the great sensitivity of the problem to it.

In EXTREMA, two different algorithms have been developed and deployed: one convex-based, and one indirect-based. The two allow the testing of different guidance approaches for deep-space scenarios.

2.1 Convex-based Algorithm

Consider the problem of determining the trajectory that minimizes the propellant consumption of a spacecraft in motion around the Sun and equipped with a low-thrust engine. If Cartesian coordinates are used, the equations of motion of such spacecraft are written as

$$\begin{bmatrix} \dot{\mathbf{r}}(t) \\ \dot{\mathbf{v}}(t) \\ \dot{m}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{v}(t) \\ -\mu \frac{\mathbf{r}(t)}{\|\mathbf{r}(t)\|_2^3} + \frac{\mathbf{T}(t)}{m(t)} \\ -\frac{\|\mathbf{T}(t)\|_2}{I_{\mathrm{sp}} g_0} \end{bmatrix} \tag{1}$$

where $\mathbf{r}$, $\mathbf{v}$, and m are the position, velocity, and mass variables, respectively. The gravitational parameter of the Sun is indicated as μ, $\mathbf{T}$ is the thrust vector, I_{sp} is the specific impulse, and g_0 is the gravitational acceleration of the Earth at sea level. The objective function, in the considered case, is

$$J_f = -m(t_f) \tag{2}$$

where t_f is the final time. Additionally, the initial and final boundary conditions (BCs) considered are

$$\begin{aligned} &\mathbf{r}(t_0) = \mathbf{r}_0, \mathbf{v}(t_0) = \mathbf{v}_0, m(t_0) = m_0 \\ &\mathbf{r}(t_f) = \mathbf{r}_f, \mathbf{v}(t_f) = \mathbf{v}_f \end{aligned} \tag{3}$$

where t_0 is the initial time. Upper and lower variable bounds can be added as

$$\mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u, \quad \mathbf{T}_l \leq \mathbf{T} \leq \mathbf{T}_u \tag{4}$$

In general, operational constraints can also be accounted and they can include, for example: exclusion or inclusion regions and duty cycles constraints [26].

The low-thrust trajectory optimization problem can therefore be formulated as

$$\begin{aligned} \underset{\mathbf{T}(t)}{\text{minimize}} \quad & J_f \\ \text{subject to} \quad & \text{Dynamics of Eq.(1)} \\ & \text{BCs of Eq.(3)} \\ & \text{Eq.(4)} \\ & \text{Operational constraints} \end{aligned} \tag{5}$$

In general, the problem can be solved with either fixed or free final time. The problem in Eq. (5) is non-convex. The convexified low-thrust trajectory optimization problem with free final time can be found in the literature [27]. The convex-based algorithm consists of a three-step process, here summarized [26].

In the first layer, a problem is solved by computing the optimal trajectory from the current spacecraft state as estimated by the navigation algorithm with either fixed or free final time [26]. The second step consists of refining the solution found in the first step for the next duty cycle. In this phase, the fidelity of the dynamical model can be increased and operational constraints such as duty cycles can be imposed. These are an alternation of a thrust arc with a maximum duration of n days and an m-day-long coast arc to allow for autonomous navigation. Finally, the third step transforms the thrust commands found by the algorithm such that the thrust angles are expressed as single arbitrary-order polynomials in each thrust arc and the switch on and off times are exactly defined.

First Step: Global Optimization Within the EXTREMA project, a closed-loop guidance approach is adopted [28]. This means that the spacecraft trajectory is recomputed periodically during the interplanetary cruise in order to account for deviations from the nominal trajectory due to missed thrust events, non-nominal thruster behavior, etc, after each orbit determination process. Hence, the commands after the first duty cycle(s) are not executed, as an updated trajectory is available. For this reason, the approach of the convex-based algorithm considers a simpler dynamical model in the first step to have a reference trajectory to track from the current state to the target celestial body. The simple dynamics increase the reliability of the optimization, hence the spacecraft will have at least a trajectory to follow. Nominally, this step considers a fixed final time optimization. However, if disturbances are high or if the spacecraft engine has not executed the correct commands for several days, a free-final-time optimization is performed [27].

Second Step: Refined Optimization Once the first step has been performed and a reference path to follow to reach the target is available onboard, the commands can be refined to account, for e.g., for more complex dynamics and operational constraints such as duty cycles or a variable thrust depending on the distance to the Sun. Fig. 3 shows a representation of a duty cycle.

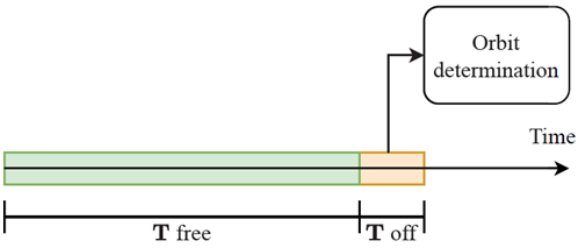

Fig. 3: Duty cycle concept [26].

During an interplanetary cruise, the orbit determination process has to be performed periodically. During the process, the thrusters have to be off to allow the satellite to have the attitude required by the navigation subsystem and to avoid interference. Currently, navigation is performed by communicating with the spacecraft through ground stations and engineers have to intervene in the process. The working week is therefore usually considered as the duration of the duty cycle, with approximately n = 6 days of thrust and m = 1 day of forced coast arc. Having an operational-compliant trajectory is fundamental to properly design and execute commands that actually lead the spacecraft toward the final target. If they are not considered the planned and executed trajectories may differ significantly, making the overall required propellant higher.

These first two steps can be summarized as per Fig. 4. After the first step, the time of flight for Step 2 is defined as ToF2 = K(n + m). The factor K is 1 if the trajectory is to be refined for the next duty cycle only. In general, K can be greater than 1 if commands for more duty cycles are required to be computed. The final boundary condition for the second optimization is obtained by evaluating the trajectory computed in Step 1 at the time ToF2 (the Waypoint in Fig. 4). Note that the optimization for Step 2 always considers a fixed final time scenario.

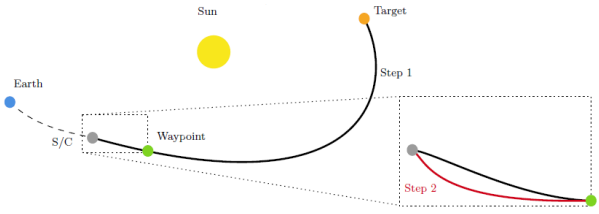

Fig. 4: Logic of first and second steps of the convex-based algorithm [26].

Third Step: Thrust Regularization

Once the first two steps have been executed, the thrust commands should be regularized before being fed to the onboard computer. The output of the convex-based guidance algorithm may be unsuitable to be directly fed to the onboard computer because of three reasons [29]:

- the switch on and off times may not be pre-

cisely captured, and therefore the engine may be switched on or off too soon or too late,

- the profile may not be exactly bang-bang,
- physical constraints on the thrust variables may be violated outside of the collocation points.

To overcome these issues, a strategy to regularize the output of the convex-based algorithm has been developed [29].

2.2 Indirect-based Algorithm

Indirect-based algorithms do not transcribe the dynamics and the constraints through a collocation scheme, but they rather find the solution to the necessary conditions of optimality of the problem. To derive them, consider the dynamics of a spacecraft similar to the one of Eq. (1)

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\alpha}, u) \Rightarrow \begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = -\frac{\mu}{r^3}\mathbf{r} + u\frac{T_{max}}{m}\boldsymbol{\alpha} \\ \dot{m} = -u\frac{T_{max}}{I_{sp}g_0} \end{cases} \tag{6}$$

where $\boldsymbol{\alpha}$ is the unit vector indicating the thrusting direction, and $u \in [0, 1]$ is the throttle factor, determining if the engine is thrusting to the maximum, 1, or is switched off, 0. The boundary conditions of the problem are still those of Eq. (3), but no bounds constraints on the variable are imposed. The final time is fixed and as the optimal profile has been said to be bang-bang, and thus discontinuous, the objective function is changed in

$$J_\epsilon = \frac{T_{max}}{I_{sp}} \int_{t_i}^{t_f} [u - \epsilon u(1-u)]\, \mathrm{d}t \tag{7}$$

Note that $\epsilon = 1$ corresponds to the energy-optimal (EO) problem, while $\epsilon = 0$ is the fuel-optimal (FO) problem. The discontinuity is then overcome using a continuation, i.e. gradually reducing ϵ and iteratively solving the problem. The necessary conditions for optimality are derived by introducing the costates, $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_\mathbf{r}; \boldsymbol{\lambda}_\mathbf{v}; \lambda_m]$ associated with the states. Considering the Hamiltonian function

$$H_\epsilon = \boldsymbol{\lambda}_r \cdot \mathbf{v} - \frac{\mu}{r^3}\mathbf{r} \cdot \boldsymbol{\lambda}_v + u\frac{T_{max}}{I_{sp}g_0}(S_\epsilon - \epsilon + \epsilon u) \tag{8}$$

It can be proved [30, 31] that the optimal thrust direction $\boldsymbol{\alpha}^*$ and the optimal throttle factor u^* are such that H_ϵ is minimized at any time instant by virtue of the Pontryagin minimum principle (PMP), meaning

$$\boldsymbol{\alpha}^* = -\frac{\boldsymbol{\lambda}_v}{\lambda_v} \quad \text{if} \quad \lambda_v \neq 0 \tag{9}$$

$$u^* = \begin{cases} 0 & \text{if} \quad S_\epsilon > \epsilon \\ \frac{\epsilon - S_\epsilon}{2\epsilon} & \text{if} \quad |S_\epsilon| \leq \epsilon \\ 1 & \text{if} \quad S_\epsilon < -\epsilon \end{cases} \tag{10}$$

(10)

where S_ϵ is the optimal switching function

$$S_\epsilon = 1 - \lambda_m - \lambda_v \frac{I_{sp}g_0}{m} \tag{11}$$

As $\boldsymbol{\alpha}^*$ and u^* depend on the state and costate $\mathbf{y} = [\mathbf{x}; \boldsymbol{\lambda}]$, to find the optimal control they have to be integrated for the whole trajectory, with the following augmented equations of the dynamics

$$\dot{\mathbf{y}} = \mathbf{F}(\mathbf{y}) \Rightarrow \begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = -\frac{\mu}{r^3}\mathbf{r} + u\frac{T_{max}}{m}\boldsymbol{\alpha} \\ \dot{m} = -u\frac{T_{max}}{I_{sp}g_0} \\ \dot{\boldsymbol{\lambda}}_r = -\frac{3\mu}{r^5}(\mathbf{r} \cdot \boldsymbol{\lambda}_v)\mathbf{r} + \frac{\mu}{r^3}\boldsymbol{\lambda}_v \\ \dot{\boldsymbol{\lambda}}_v = -\boldsymbol{\lambda}_r \\ \dot{\lambda}_m = -\frac{u\lambda_v T_{max}}{m^2} \end{cases} \tag{12}$$

Thus, the indirect method solves a two-point-boundary-value problem: being $\mathbf{y}(t) = \boldsymbol{\varphi}_\epsilon(\mathbf{y}_i, t_i, t)$ the solution flow for a specific ϵ value of Eq. (12) integrated from the initial time t_i to a generic time instant t, with $\boldsymbol{\alpha}^*$ and u^* provided by Eq. (9) and (10), respectively, the EO and FO shooting problems aim to find $\boldsymbol{\lambda}_i^* = \boldsymbol{\lambda}^*(t_i)$ such that $\mathbf{y}(t_f) = \boldsymbol{\varphi}_\epsilon([\mathbf{x}_i, \boldsymbol{\lambda}_i^*], t_i, t_f)$ satisfies the final boundary conditions of Eq. (3) plus $\lambda_m(t_f) = 0$.

In operational-compliant trajectory design, the objective is to switch off the engine according to the duty cycle scheme imposed. In order to satisfy the necessary conditions of optimality, this event is treated in the indirect-based guidance algorithm as an interior-point constraint [32]. We define $S_{DC} = S_{DC}(t)$ as the duty cycle switching function, depending only on time, and we properly model it to impose that the engine switches off every time the function crosses a certain value ϵ_{DC}. Accordingly, the optimal throttle factor turns into

$$u^* = \begin{cases} 0 & \text{if} \quad S_\epsilon > \epsilon \quad \text{or} \quad S_{DC} < \epsilon_{DC} \\ \frac{\epsilon - S_\epsilon}{2\epsilon} & \text{if} \quad |S_\epsilon| \leq \epsilon \quad \text{and} \quad S_{DC} \geq \epsilon_{DC} \\ 1 & \text{if} \quad S_\epsilon < -\epsilon \quad \text{and} \quad S_{DC} \geq \epsilon_{DC} \end{cases} \tag{13}$$

It can be proved [33] that being dependent only on time, this constraint does not impose any jump to consider during the integration of $\mathbf{y}(t_f) = \boldsymbol{\varphi}_\epsilon([\mathbf{x}_i, \boldsymbol{\lambda}_i^*], t_i, t_f)$. However, introducing several discontinuities in the throttle factor, a three-layer continuation scheme has been ideated to enhance the convergence.

EO to FO continuation The starting point of this continuation scheme requires the solution of the EO problem without the duty cycles imposition. Then, the FO solution is computed gradually decreasing ϵ to zero. Usually the number of optimal switching imposed by the optimality is fewer than the one required by an operational-compliant trajectory. As the switchings create an ill-conditioned state transition matrix (STM), used to provide analytical derivatives to the two-point

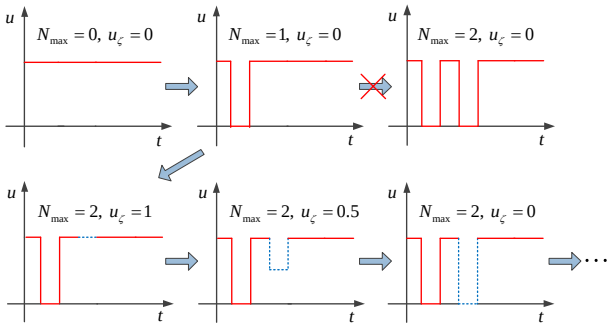

Fig. 5: N_{max}-continuation scheme.

boundary value problem, the EO to FO continuation is the first performed. In this way, usually, ill-conditioned STMs are encountered only at the final steps of the integration.

N_{max}-continuation After, the approach called N_{max}-continuation is employed. It gradually turns inactive coasting arcs into active ones, i.e., the maximum number of forced coasting arcs N_{max} increases at each iteration of the continuation scheme. This continuation can be seen in Fig. 5.

The variable u_ζ is defined as the throttle factor in the N_{max}-th coasting arc to be imposed, and it is the third parameter on which a continuation is performed. Starting from the FO solution (i.e., $N_{max}=0$ and $u_\zeta=0$), the number of active coasting arcs N_{max} is increased from 0 to 1. The FO solution is used as the initial guess for this auxiliary problem (i.e., $N_{max}=1$ and $u_\zeta=0$). Supposing the solution of this problem is obtained, N_{max} is increased from 1 to 2, and the new auxiliary problem is solved (i.e., $N_{max}=2$ and $u_\zeta=0$). Suppose that instead, in this case, the solver fails. Then, the problem u_ζ is imposed as 1, and the new problem is solved (i.e., $N_{max}=2$ and $u_\zeta=1$). The value of u_ζ is gradually reduced until $u_\zeta=0$ is obtained. Starting from this solution as an initial guess, a new problem with $N_{max}=3$ and $u_\zeta=0$ is solved, and so on until the trajectory is OC.

The duty cycle switching function $S_{DC}=S_{DC}(t)$, depending only on time, has to be properly modeled to impose the operational compliant trajectory shape required, i.e., the correct duration of each forced coasting arc. This is performed by requiring that the engine switches off every time the function crosses a certain value ϵ_{DC}. To model the $S_{DC}(t)$, a pulse wave function is selected. In particular, due to its regularity and smoothness with respect to the normal pulse wave function, an expansion based on the sinc function is employed as

$$S_{DC}(t)=\frac{A\tau}{T}\left(1+2\sum_{n=1}^{\infty}(\mathrm{sinc}(n\frac{\tau}{T})\cos(2\pi nft))\right) \tag{14}$$

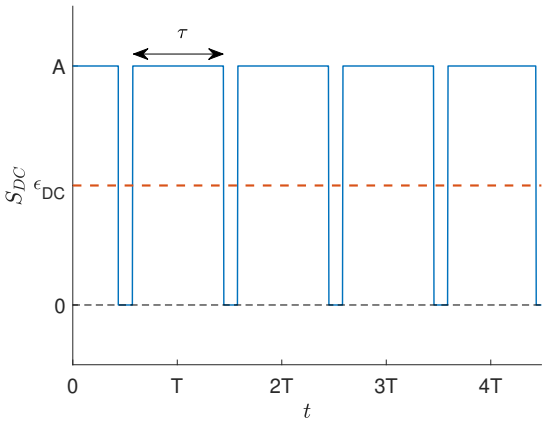

Fig. 6: Pulse wave modeling the duty cycles.

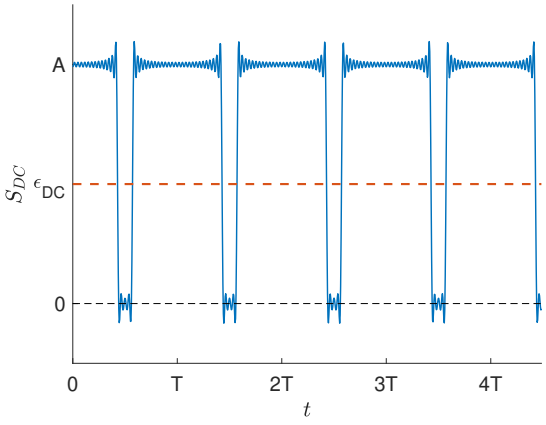

Fig. 7: Pulse wave modeled with sinc expansion.

where A is the amplitude, T is the period, τ is the pulse length, and f = 1/T is the frequency, of the pulse wave. These values are customized in order to enforce the right duration of the coasting arcs, e.g., to have duty cycles corresponding to a working week with 6 days of thrusting and 1 day of coasting, T is imposed to be 7 days, and τ to be 6 days. See Fig. 6 and 7 for reference.

3 HIL Simulations

3.1 EXTREMA Guidance Simulation Framework

For the purpose of this paper, considering the ESH previously presented, only the ETHILE thrust bench has been employed to validate the guidance algorithms.

Additionally, two other ESH units are introduced into the simulation framework, namely The SPace Environment SImulator (SPESI) and EXTREMA fLAtSat Processing SystEm (ELAPSE), described next.

SPESI: aims to forward-propagate the spacecraft trajectory by feedback on its highly accurate environment models by real-time measurements from the spacecraft sensors [34]. Specifically, it receives a measurement of the probe's attitude and a measurement of the thruster

force.

ELAPSE: combines software-hardware components resulting in a highly modular autonomous GNC flight software coded in C++, residing on flight-representative embedded hardware [35]. The system integrates the guidance and navigation algorithms, supports their computations, and provides an execution policy. Moreover, the system is capable of commanding the thruster and the ADCS to follow the guidance solution generated.

The integration of the algorithms within the main software is a crucial process that first involves the automatic C/C++ code generation from MATLAB and the verification of the software requirements. Once the resulting code is available in form of a plug and play library, it can be integrated into the host software of ELAPSE. Its modularity and abstraction level allow for the agile integration of each of the two algorithms. Supporting additional methods are then coded to complete the process and the Guidance Processing Unit is available to be tested on the embedded processor by applying a Processor-In-the-Loop (PIL) approach.
Additionally, a HIL simulation pipeline is employed to further validate the algorithms in a more operative condition.

3.2 Open-Loop Simulation

To validate the solutions provided by the two algorithms onboard during the interplanetary cruise, they were deployed within the ESH to perform HIL tests (see Fig. 8). Specifically, the Guidance Processing Unit within the main flight software in ELAPSE were deployed on a Raspberry Pi Model 4 Model B with 4GB RAM with a Broadcom 2711, 64-bit quad-core Cortex-A72 processor[1]. The guidance solution is output in the form of a sequence of thrust epochs and thrust direction parametrizations. In this framework, while the times are commanded to ETHILE, the direction is sent to SPESI as if they were the real one, bypassing STASIS. In order to represent the system expected behavior, the information was perturbed by considering a reasonable pointing error.

Figure 8 presents a schematic of what just described, giving visual information on how the simulation loop works.

3.3 Scenarios

We test our approach against three representative scenarios concerning a spacecraft departing from Earth and aiming to reach a target state. Table 1 presents the data about the scenarios, normalized according to quantities as in [36].

[1]https://www.raspberrypi.com/products/raspberry-pi-4-model-b/ [last visited Sep 18, 2024]

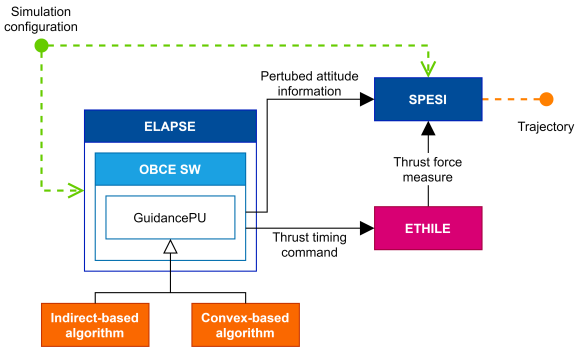

Fig. 8: ESH units involved into the HIL guidance loop simulations with their interactions.

Scenario	Earth to Venus	Earth to Mars	Earth to Dionysus
$\mathbf{r}_0$ [LU]	$[0.9368, -0.3627, 0.0002]^\top$	$[0.9368, -0.3627, 0.0002]^\top$	$[0.9431, -0.3667, -0.0001]^\top$
$\mathbf{v}_0$ [VU]	$[0.3392, 0.9309, 0]^\top$	$[0.3392, 0.9309, 0]^\top$	$[0.3460, 0.9288, 0]^\top$
$\mathbf{m_0}$ [kg]	22.6	1500	4000
$\mathbf{r}_f$ [AU]	$[0.6863, 0.2272, -0.0363]^\top$	$[1.1283, 0.9066, -0.0085]^\top$	$[0.0758, 3.2065, 0.0876]^\top$
$\mathbf{v}_f$ [VU]	$[-0.3795, 1.1092, 0.0371]^\top$	$[-0.4797, 0.7031, 0.0265]^\top$	$[-0.3735, 0.1387, 0.0938]^\top$
ToF [days]	1000	1360	3700
$T_{\max}$ [N]	0.33	2.25×10^{-3}	0.32
I_{sp} [s]	3800	3067	3000

Tab. 1: Parameters of the considered scenarios on EclipJ2000 reference frame.

4 Results

This section provides an overview of the results obtained by the two algorithms and their interactions with the facilities of the ESH.

4.1 Convex-based Algorithm Results

In this section, the results of the Convex-based algorithm are shown. Figs. 9 to 11 show the nominal solution for each of the scenarios presented in Table 1. These are computed using Steps 1 and 3 of the Convex-based algorithm, therefore not considering any duty cycles.

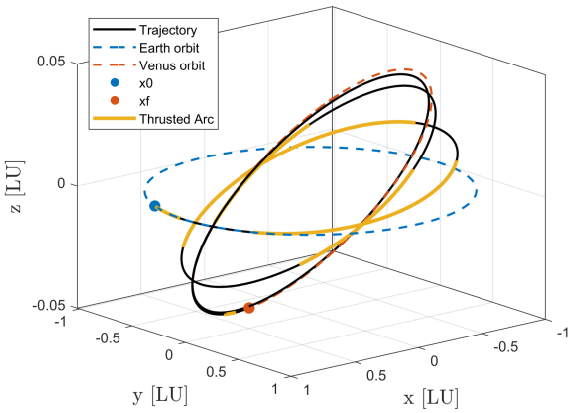

Fig. 9: Trajectory in EclipJ2000 reference frame of the Earth to Venus scenario with the convex-based algorithm.

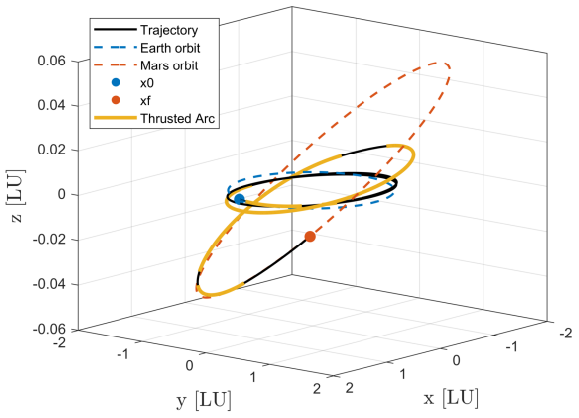

Fig. 10: Trajectory in EclipJ2000 reference frame in the Earth to Mars scenario using the Convex-based algorithm.

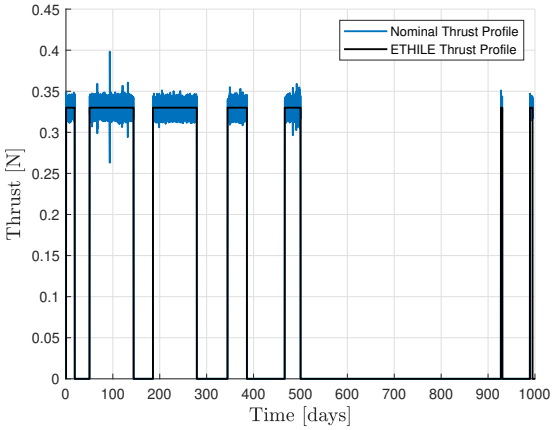

Fig. 12: Thrust profile from nominal solution against thrust profile measured by SPESI for the Earth to Venus scenario.

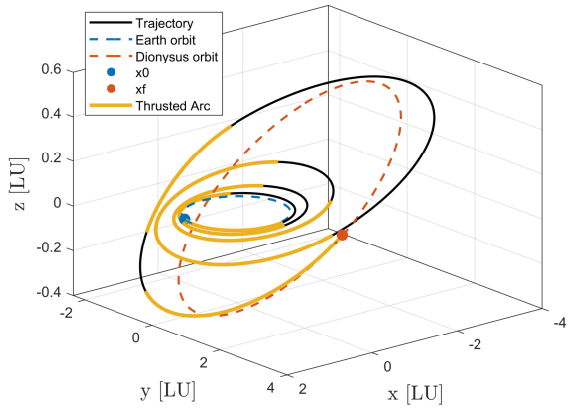

Fig. 11: Trajectory in EclipJ2000 reference frame in the Earth to Dionysus scenario using the Convex-based algorithm.

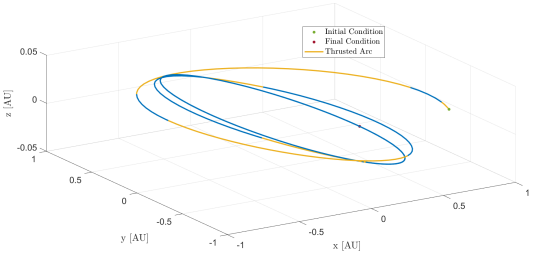

Fig. 13: Trajectory in EclipJ2000 reference frame in the Earth to Venus scenario.

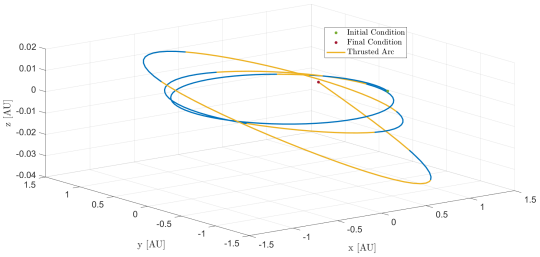

Fig. 14: Trajectory in EclipJ2000 reference frame in the Earth to Mars scenario.

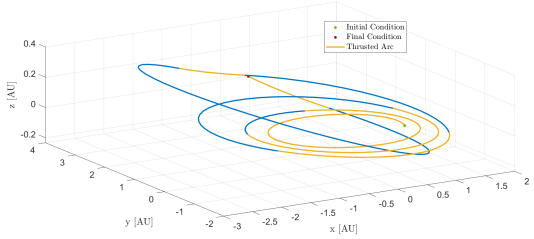

Fig. 15: Trajectory in EclipJ2000 reference frame in the Earth to Dionysus scenario.

Thrust History of ETHILE Fig. 12 highlights the thrust profile of the nominal solution with the thrust profile measured by ETHILE. The latter one introduces perturbations as it mimics the physicality of the spacecraft engine. Consequently, these perturbed thrust commands would lead to a trajectory different from the nominal one shown in Fig. 9.

4.2 Indirect-based Algorithm Results

In this section, the results of the indirect-based algorithm are shown. Firstly, the same scenarios of the convex-based algorithm have been run, only with the EO to FO continuation, to validate the optimality of the fuel optimal solutions found. The trajectories solution are shown in Figs. 13 to 15.

In the following Figs. from. 16 to 18 instead, the throttle factor and the optimal switching functions are reported. It can be noted that, apart from some noises due to the direct collocation in the convex algorithm, the solutions match.

An example of application of the N_{max}-continuation to obtain an operational compliant trajectory with duty cycles is here reported. The selected trajectory is an in-

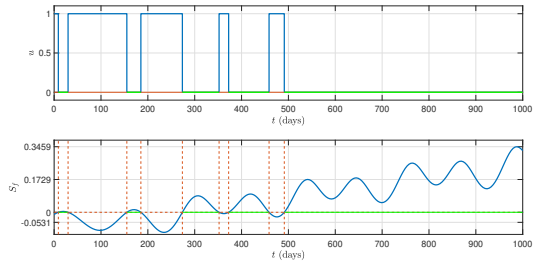

Fig. 16: Throttle factor and optimal switching function for the Earth to Venus scenario.

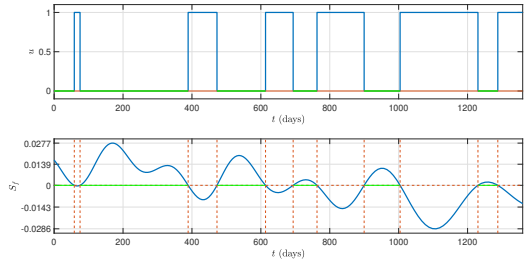

Fig. 17: Throttle factor and optimal switching function for the Earth to Mars scenario.

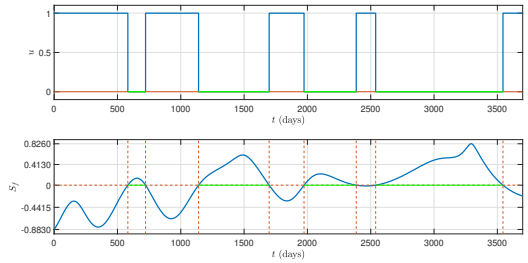

Fig. 18: Throttle factor and optimal switching function for the Earth to Dionysus scenario.

terplanetary transfer of the CubeSat M-ARGO toward an asteroid [33]. The duty cycle chosen scheme is $T = 7$ days and $\tau = 6$ days. In the Figs. from. 19 to 20 the results are shown. The starting point is the Sun–Earth Lagrangian point 2.

4.3 Computational evaluation

During the PIL testing phase introduced in Section 3, a benchmark of the Indirect and Convex-based functions was performed to identify the impact of the lower performances of the embedded processor on the computational time. Specifically, here the CPU time measurements for each of the three representative scenarios of Table 1 are reported in Table 2.

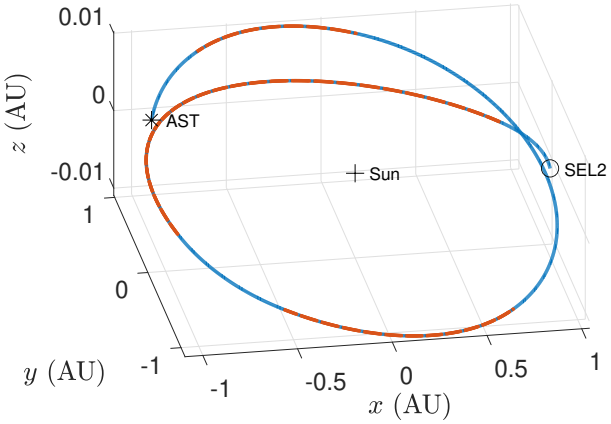

Fig. 19: Trajectory in the EclipJ2000 reference frame.

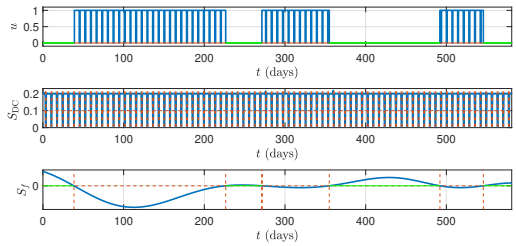

Fig. 20: Throttle factor, optimal switching function, and duty cycle switching function profiles.

	Convex-based	**Indirect-based**
Earth-Venus	21.33 (4.43)	49.22 (5.11)
Earth-Mars	19.82 (3.89)	53.14 (11.59)
Earth-Dionysus	40.60 (10.53)	160.23 (72.62)

Tab. 2: Mean CPU time in seconds for each scenario when algorithms are run on the Raspberry Pi4B (MATLAB CPU time in parenthesis).

An important worsening of the CPU time was recorded as expected. Moreover, one can see that the worst case is represented by Dionysus, where the optimization is more challenging as the time of flight is much longer than the first two cases.

5 Conclusions

In this work, the architecture of the two autonomous guidance algorithms under development and testing within the EXTREMA project at DART Lab have been described.

The first algorithm is based on convex optimization and it exploits a regularization process to make the thrust commands. The second algorithm is indirect-based and imposes the duty cycles as interior-point constraints. An overview of the facilities within the ESH is proposed, as well as the scheme for the HIL simulations performed in the Hub. Some preliminary results are shown. The algorithms have shown to be effective

when run on representative hardware, and they compute commands correctly executed by the in-house built thrust-facility ETHILE.

The agility characterizing the deployment of the algorithms not only on a flight representative software-hardware unit but also within a complete HIL facility is paramount in improving the algorithm's reliability, and robustness and continuously growing their flight-readiness.

Acknowledgments

This research is part of EXTREMA, a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 864697).

A.M. would like to thank the Agenzia Spaziale Italiana (ASI) and the Space Generation Advisory Council (SGAC) for the financial support granted to attend the 72nd Space Generation Congress (SGC) and the 75th International Astronautical Congress (IAC).

A.M. and F.T. would also like to acknowledge the funding received under ESA Contracts No. 4000136010/21/NL/GLC/my.

Bibliography

[1] Roger Walker, David Binns, Cristina Bramanti, Massimo Casasco, Paolo Concari, Dario Izzo, Davar Feili, Pablo Fernandez, Jesus Gil Fernandez, Philipp Hager, D. Koschny, V. Pesquita, N. Wallace, I. Carnelli, M. Khan, M. Scoubeau, and D. Taubert. Deep-space CubeSats: thinking inside the box. *Astronomy & Geophysics*, 59(5):5–24, 2018.

[2] Woellert, Kirk and Ehrenfreund, Pascale and Ricco, Antonio J and Hertzfeld, Henry. Cubesats: Cost-effective science and technology platforms for emerging and developing nations. *Advances in Aerospace Research*, 47(4):663–684, 2011.

[3] Gianfranco Di Domenico, Eleonora Andreis, Andrea Carlo Morelli, Gianmario Merisio, Vittorio Franzese, Carmine Giordano, Alessandro Morselli, Paolo Panicucci, Fabio Ferrari, and Francesco Topputo. The ERC-Funded EXTREMA Project: Achieving Self-Driving Interplanetary CubeSats. In *Modeling and Optimization in Space Engineering: New Concepts and Approaches*, pages 167–199. Springer, 2022.

[4] Alessandro Morselli, Gianfranco Di Domenico, Eleonora Andreis, Andrea Carlo Morelli, Gianmario Merisio, Vittorio Franzese, Carmine Giordano, Paolo Panicucci, Fabio Ferrari, Francesco Topputo, et al. The EXTREMA orbital simulation hub: A facility for GNC testing of autonomous interplanetary CubeSat. In *4S Symposium*, pages 1–13, 2022.

[5] Paolo Panicucci and Francesco Topputo. The TinyV3RSE Hardware-in-the-Loop Vision-Based Navigation Facility. *Sensors*, 22(23):9333, 2022.

[6] Alessandro Morselli, Andrea Carlo Morelli, Francesco Topputo, et al. ETHILE: A Thruster-In-The-Loop Facility to Enable Autonomous Guidance and Control of Autonomous Interplanetary CubeSat. In *73rd International Astronautical Congress*, pages 1–10, 2022.

[7] Gianfranco Di Domenico, Francesco Topputo, et al. STASIS: an Attitude Testbed for Hardware-in-the-Loop Simulations of Autonomous Guidance, Navigation, and Control Systems. In *73rd International Astronautical Congress*, pages 1–20, 2022.

[8] Jhon T. Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998.

[9] Oskar Von Stryk and Roland Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37(1):357–373, 1992.

[10] Paul J Enright and Bruce A Conway. Discrete approximations to optimal trajectories using direct transcription and nonlinear programming. *Journal of Guidance, Control, and Dynamics*, 15(4):994–1002, 1992.

[11] Charles R Hargraves and Stephen W Paris. Direct trajectory optimization using nonlinear programming and collocation. *Journal of guidance, control, and dynamics*, 10(4):338–342, 1987.

[12] Francesco Topputo and Chen Zhang. Survey of direct transcription for low-thrust space trajectory optimization with applications. *Abstract and Applied Analysis*, 2014, 2014.

[13] Alessandra Mannocchi, Carmine Giordano, and Francesco Topputo. A homotopic direct collocation approach for operational-compliant trajectory design. *The Journal of the Astronautical Sciences*, 69(6):1649–1665, 2022.

[14] Jean Albert Kechichian. Optimal low-earth-orbit-geostationary-earth-orbit intermediate acceleration orbit transfer. *Journal of guidance, control, and dynamics*, 20(4):803–811, 1997.

[15] Christopher L Ranieri and Cesar A Ocampo. Indirect optimization of three-dimensional finite-burning interplanetary transfers including spiral

dynamics. *Journal of guidance, control, and dynamics*, 32(2):445–455, 2009.

[16] Yang Wang and Francesco Topputo. Indirect optimization of power-limited asteroid rendezvous trajectories. *Journal of Guidance, Control, and Dynamics*, 45(5):962–971, 2022.

[17] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[18] Christian Hofmann and Francesco Topputo. Rapid low-thrust trajectory optimization in deep space based on convex programming. *Journal of Guidance, Control, and Dynamics*, 44(7):1379–1388, 2021.

[19] Scott A Striepe, Chirold D Epp, and Edward A Robertson. Autonomous precision landing and hazard avoidance technology (ALHAT) project status as of May 2010. In *International Planetary Probe Workshop 2010 (IPPW-7)*, number NF1676L-10317, 2010.

[20] Qiong Wang and Jizhong Liu. A Chang'e-4 mission concept and vision of future Chinese lunar exploration activities. *Acta astronautica*, 127:678–683, 2016.

[21] Changyi Zhou, Yingzhuo Jia, Jianzhong Liu, Huijun Li, Yu Fan, Zhanlan Zhang, Yang Liu, Yuanyuan Jiang, Bin Zhou, Zhiping He, et al. Scientific objectives and payloads of the lunar sample return mission—Chang'E-5. *Advances in Space Research*, 69(1):823–836, 2022.

[22] Yuichi Tsuda, Takanao Saiki, Fuyuto Terui, Satoru Nakazawa, Makoto Yoshikawa, Sei-ichiro Watanabe, Hayabusa2 Project Team, et al. Hayabusa2 mission status: landing, roving and cratering on asteroid Ryugu. *Acta Astronautica*, 171:42–54, 2020.

[23] Sven Stappert, Jascha Wilken, Leonid Bussler, and Martin Sippel. A systematic comparison of reusable first stage return options. In *8th European Conference For Aeronautics and Space Sciences*, 2019.

[24] Simone D'Amico, J-S Ardaens, and Robin Larsson. Spaceborne autonomous formation-flying experiment on the PRISMA mission. *Journal of Guidance, Control, and Dynamics*, 35(3):834–850, 2012.

[25] Didier Pinard, Stéphane Reynaud, Patrick Delpy, and Stein E Strandmoe. Accurate and autonomous navigation for the ATV. *Aerospace Science and Technology*, 11(6):490–498, 2007.

[26] A. C. Morelli, A. Morselli, D. Perico, and F. Topputo. The extrema autonomous guidance algorithm for low-thrust interplanetary spacecraft. In *74th International Astronautical Congress (IAC 2023)*, pages 1–10, 2023.

[27] Andrea Carlo Morelli, Gianmario Merisio, Christian Hofmann, and Francesco Topputo. A convex guidance approach to target ballistic capture corridors at mars. In Matt Sandnas and David B. Spencer, editors, *Proceedings of the 44th Annual American Astronautical Society Guidance, Navigation, and Control Conference, 2022*, pages 731–754, Cham, 2024. Springer International Publishing.

[28] Christian Hofmann and Francesco Topputo. Closed-loop guidance for low-thrust interplanetary trajectories using convex programming. In *11th International ESA Conference on Guidance, Navigation & Control Systems, ESA GNC 2021*, pages 1–15, 2021.

[29] Andrea C. Morelli, Alessandro Morselli, Carmine Giordano, and Francesco Topputo. Convex trajectory optimization using thrust regularization. *Journal of Guidance, Control, and Dynamics*, 47(2):339–346, 2024.

[30] J. M. Longuski, J. J. Guzman, and J. E. Prussing. *Optimal Control with Aerospace Applications*. Space Technology Library, 2014.

[31] Bruce A Conway. *Spacecraft trajectory optimization*. Cambridge University Press, Cambridge, UK, 2010.

[32] AE Bryson and Yu-Chi Ho. *Applied optimal control*. Taylor & Francis, London, 1975.

[33] Alessandra Mannocchi, Carmine Giordano, and Francesco Topputo. An indirect formulation of operational compliant low-thrust trajectories. In *AIAA Scitech 2024 Forum*, page 0631, 2024.

[34] Carmine Giordano, Francesco Topputo, et al. Spesi: A real-time space environment simulator for the extrema project. In *33rd AAS/AIAA Space Flight Mechanics Meeting*, pages 1–13, 2023.

[35] Davide Perico, Gianfranco Di Domenico, Gianmario Merisio, Francesco Topputo, et al. Elapse: a flatsat software and processing unit for deep-space autonomous gnc systems testing. In *46th AAS Guidance, Navigation and Control Conference*, pages 1–17, 2024.

[36] Andrea Carlo Morelli, Christian Hofmann, and Francesco Topputo. Robust low-thrust trajectory optimization using convex programming and a homotopic approach. *IEEE Transactions on Aerospace and Electronic Systems*, 58(3):2103–2116, 2021.