74nd International Astronautical Congress (IAC) – 2-6 October 2023.

IAC-23,C1,5,3,x80085

The EXTREMA Autonomous Guidance Algorithm for Low-Thrust Interplanetary Spacecraft

Andrea Carlo Morelli[1], **Alessandro Morselli**[2], **Davide Perico**[3], **Francesco Topputo**[4]

This work presents the architecture of the autonomous guidance algorithm that is being tested in hardware-in-the-loop simulations in the context of the ERC-funded EXTREMA project at Politecnico di Milano. The optimal spacecraft trajectory is computed by means of a three-step process: first, a problem is solved which computes the optimal trajectory from the current spacecraft state as estimated by the navigation algorithm with either fixed or free final time. The second step consists of refining the solution found at the first step for the next duty cycle. In this phase, the duty cycle constraints (a thrust arc maximum duration of n days and a m-day-long coast arc to allow for autonomous navigation) are also imposed and the time of flight is fixed to $n + m$ days. Finally, the third step transforms the thrust commands found by the convex optimization algorithm such that the thrust angles are expressed as single arbitrary-order polynomials in each thrust arc. This last step allows for the transformation of the algorithm output into actual executable commands by the thruster and the attitude control system. The algorithm is run on a Raspberry Pi and in closed-loop simulations to test whether the performance is suitable for onboard use. The computed commands are then executed by the thrust test bench ETHILE, a facility developed within the project EXTREMA.

1 Introduction

In the past two decades, the space sector has witnessed disruptive changes. On top of the national space agencies, new and smaller actors have had the possibility of integrating, launching, and operating their own satellites. The development of small spacecraft such as CubeSats has significantly contributed to this paradigm shift as these platforms have considerably lower production and integration costs with respect to larger spacecraft [1]. However, the cost of their operations is comparable to the one of standard missions. Autonomous Guidance, Navigation, and Control (GNC) capabilities onboard miniaturized spacecraft would enable low-cost operations for low-cost platforms, which could possible lead to a new space era: a multitude of miniaturized spacecraft exploring our Solar System and provide precious information about it. Although CubeSats have almost only been used for missions around the Earth so far, few interplanetary CubeSats missions have recently been proposed as well [2, 3].

[1]PhD Candidate, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milano, Italy, andreacarlo.morelli@polimi.it
[2]Assisant Professor, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milano, Italy, alessandro.morselli@polimi.it
[3]PhD Candidate, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milano, Italy, davide.perico@polimi.it
[4]Full Professor, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34, 20156, Milano, Italy, francesco.topputo@polimi.it

Autonomous guidance is the task of computing the spacecraft trajectory directly onboard [4]. If low-thrust engines are considered, this translates to solving an optimal control problem (OPC). The most used methods that solve OPCs are usually classified in indirect and direct ones [5]. The former exploit the calculus of variations to find the necessary conditions for optimality and solve the resulting two-point boundary value problem; the latter discretize the OPC and solve the associated (non)linear program. However, both the approaches lack of robustness or require too many computational resources to be run on spacecraft hardware [6], which is usually of reduced capabilities due to the harsh space environment (e.g., radiations). Convex optimization is a direct method that relies on highly-efficient solvers to find a solutions to problems that are written in convex form [7] and it has solid theoretical guarantees [8]. Most of real-life problems are inherently nonconvex, and therefore they are convexfied and a sequential convex programming (SCP) technique is usually used to solve them [9]. This technique has shown to be robust for a number of different aerospace-related problems [10], including powered descent and landing [11], entry [12], and low-thrust trajectory optimization [13].

The EXTREMA project [14], which has received funding from the European Research Council, aims at enabling autonomous GNC capabilities for deep-space CubeSats. It builds upon three Pillars, one of them being related with autonomous guidance and control. In this work, the autonomous guidance algorithm that is being used within the project is described. The EXTREMA thrust

test bench named ETHILE [15] that emulates the performance of a real low-thrust engine is used to execute the guidance commands found by the guidance algorithm. The latter is run on a Raspberry Pi to simulate reduced memory and performance of actual spacecraft hardware. The algorithm consists of a three-steps process. In the first layer, a problem is solved which computes the optimal trajectory from the current spacecraft state as estimated by the navigation algorithm with either fixed or free final time [16]. The second step consists of refining the solution found at the first step for the next duty cycle. In this phase, the fidelity of the dynamical model can be increased and operational constraints can be added. In this work, we consider duty cycle constraints (a thrust arc maximum duration of n days and a m-day-long coast arc to allow for autonomous navigation) are imposed and the time of flight is fixed to $n+m$ days. 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.

The remainder of the paper is organized as follows. Section 2 introduces the low-thrust trajectory optimization problem. Section 3 presents the architecture of the EXTREMA autonomous guidance algorithm. Section 4 shows the simulations that have been performed. Finally, Section 5 concludes the work.

2 Problem Formulation

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 [17]

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

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 OPC required to find the optimal spacecraft trajectory is composed of the following elements.

- The objective function, which in our case is

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

  where t indicates the time and the pedex $(\cdot)_f$ indicates final quantities.

- The dynamics of the spacecraft (e.g., the one in Eq. (1)). In general, the fidelity of the model can be increased by considering perturbations of other celestial bodies and solar radiation pressure.
- Initial and final boundary conditions (BCs), namely

$$\mathbf{r}(t_0) = \mathbf{r}_0,\ \mathbf{v}(t_0) = \mathbf{v}_0,\ m(t_0) = m_0 \tag{3a}$$

$$\mathbf{r}(t_f) = \mathbf{r}_f,\ \mathbf{v}(t_f) = \mathbf{v}_f \tag{3b}$$

  where the pedex $(\cdot)_0$ indicates initial quantities.
- Upper and lower variables bounds, i.e.,

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

- In general, operational constraints can also be accounted for. They can include, for example: exclusion or inclusion constraints (e.g., the Earth shall always be visible from the spacecraft), and duty cycles constrains (i.e., the a priori imposition of no-thrust periods in order to perform other tasks, such as orbit determination).

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

$$\begin{aligned} \underset{\mathbf{T}(t)}{\text{minimize}} \quad & \text{Eq. (2)} \\ \text{subject to:} \quad & \text{The dynamics in Eq. (1)} \\ & \text{BCs in Eqs. (3a) and (3b)} \\ & \text{Bounds in Eq. (4)} \\ & \text{(Operational constraints)} \end{aligned} \tag{5}$$

In general, the problem can be solved with either fixed or free final time. If the latter case is considered and the target is a celestial body, the final boundary conditions must be modified accordingly.

The problem in Eq. (5) is nonconvex. In fact, the dynamics in Eq. (1) are nonconvex, and they have to be modified for the problem to be expressed in a convex form. The convexified low-thrust trajectory optimization problem with free final time can be found in literature [16].

3 Three Steps Guidance Algorithm

The objective of the guidance algorithm is to provide the onboard computer with the commands to be executed by the thruster. In general, the EXTREMA guidance algorithm should take as inputs at least the initial and final boundary conditions, solve the convexified version of the problem in Eq. (5) by means of SCP, and provide as output the switch on and off times and the time history of the attitude. While finding the output, the algorithm should possess the following three fundamental characteristics [17]:

C1. *computational affordability*, meaning that it should be compliant with onboard execution in terms of memory and computational time required;

C2. *reliability*, meaning that the algorithm should provide a solution at any time instant, if triggered;

C3. *optimality*, meaning that the solution should minimize the fuel consumption.

In the light of the characteristics C1-C3, a three-steps guidance process has been envisaged.

3.1 First Step: Global Optimization

Within the EXTREMA project, a closed-loop guidance approach is adopted [18]. 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 behaviour, etc. Due to the fact that the spacecraft trajectory is recomputed after each orbit determination process, planning a trajectory that considers high-fidelity dynamics, operational constraints, variable thrust and specific impulse may be an overshoot. In particular, characteristics C1 and C2 may be compromised, as a more complex optimization problem is solved each time. Still, the commands after the first duty cycle(s) are not even executed, as a updated trajectory is available. Previous work has shown that in case high-fidelity models are accounted for in the spacecraft dynamics, reliability and computational time must be traded off [19]. Therefore, our approach 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 increases the reliability and the computational affordability of the optimization, hence the spacecraft will have at least have 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, the fixed final time optimization may be infeasible. In such case, a free-final-time optimization is performed [16]. In this work, the widely-used Hermite–Simpson discretization method [20] is used to solve the convexified version of the problem in Eq. (5).

3.2 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., more complex dynamics and operational constraints such as duty cycles or a variable thrust depending from the distance to the Sun.

In this work, the second step considers the same orbital dynamics as Step 1. The duty cycles are taken into account instead. Figure 1 shows what we mean with the expression *duty cycle*. During an interplanetary cruise, the orbit determination process has to be performed periodically. During the process, the thrusters have to be off because of two reasons: first, to allow the satellite to have the attitude required by the navigation subsystem; second, to avoid interference. The alternation of a period of time when the engines can be on or off depending on the optimization and a period when they shall be off is defined as *duty cycle* in this work. Clearly, a complete interplanetary trajectory is made up of several duty cycles. 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 duration of the duty cycle, with approximately $n = 6$ days of free thrust and $m = 1$ day of forced coast arc. When onboard GNC is envisaged instead, n and m can change at each duty cycle depending on several factors for convenience. This is the first reason why duty cycle constraints are not considered in the first step: the majority of the thrust profile computed at Step 1 is not executed, and therefore all the information after the first duty cycle(s) may be re-computed at the next optimization. Designing the whole trajectory by imposing duty cycle constraints may be an overshoot and can worsen the performance of the algorithm in terms of characteristics C1 and C2. However, 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, especially in autonomous GNC scenarios, the planned and executed trajectories may differ significantly, making the overall required propellant higher.

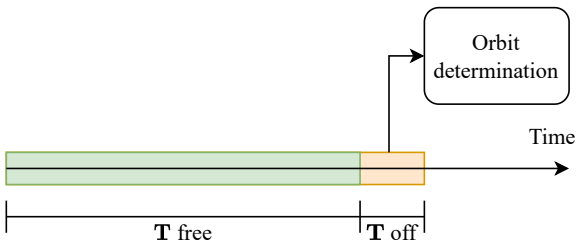

Fig. 1: Representation of a duty cycle.

The first two steps can be summarized as per Fig. 2. Consider the spacecraft is undergoing its interplanetary trajectory towards the target celestial body. Orbit determination has just been performed and the onboard computer needs the trajectory to follow to be updated, according to the closed-loop guidance scheme. The idea

is to

1. first, compute the nominal trajectory from the current spacecraft state to the target celestial body;
2. then, depending on the current values of the parameters n and m, the time of flight for Step 2 is defined as $\mathrm{ToF}_2 = K(n+m)$. 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.
3. finally, the solution obtained at the first step is re-optimized for the next duty cycle(s). The final boundary condition for the second optimization is obtained by evaluating the trajectory computed in Step 1 at the time ToF_2 (the Waypoint in Fig. 2). Note that the optimization for Step 2 always considers a fixed final time scenario.

Due to the fact that Step 2 imposes a coast period when Step 1 does not, the thrust level for the two steps cannot be the same. In fact, given n, m, and a safety factor $\xi < 1$, we impose

$$T_{\max}^{Step1} = \xi \frac{n}{n+m} T_{\max}, \tag{6}$$

i.e., we decrease the maximum available thrust in Step 1. This technique also serves as a safety margin for unexpected non-nominal behaviour during the cruise, such as the loss of the spacecraft for some period of time [21]. In fact, it is likely that a safety margin would be required regardless of the presence of Step 2. An alternative to this approach would be to directly impose the operational constraints and consider an high-fidelity model in Step 1 but for only the first K duty cycles, and using an adaptive discretization mesh. Different adaptive meshes have been developed in literature for different purposes, e.g., to better capture the bang-bang structure of the thrust profile [22, 23]. Although incorporating Step 2 in Step 1 is a valuable alternative, a one-step approach would be less safe from a convergence point of view. In fact, if Step 2 of our approach does not converge, Step 1 can still be used as a reference for the next duty cycle.

3.3 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. In fact, convex optimization is a direct method. This means that the problem in Eq. (5) is first convexified and then discretized and solved. The output of the convex-based guidance algorithm may be unsuitable to be directly fed to the onboard computer because of three reasons [24]:

1. the switch on and off times may not be precisely captured, and therefore the engine may be switched on or off too soon or too late. This could result in a more or less significant deviation of the spacecraft state to the desired one, in turn causing an higher propellant consumption.
2. The profile may not be exactly bang-bang. Direct collocation methods usually do not impose a bang-bang structure of the thrust profile, and therefore some points that are neither 0 nor $T_{\max}$ can be present.
3. Physical constraints on the thrust variables may be violated outside of the collocation points. In turn, this may lead to thruster execution errors.

To overcome the aforementioned issues, we have developed a strategy to regularize the output of the convex optimization layer such that the switch on and off times are exactly defined and the thrust angles are expressed as arbitrary-order polynomials inside each thrust arc [24].

4 Simulations

We test our approach using the database of scenarios presented in Ref. [25]. A batch of 10 scenarios is considered. Table 1 presents data about the scenarios, and Fig. 3 shows the nominal associated trajectories computed with the algorithm. Both the table and the figure show that the trajectories consist of multi-revolutions transfers with long times of flight and challenging initial and final boundary conditions. Note that the fourth scenario converged with variable time of flight. We perform two analyses. First, the algorithm is run on a Raspberry Pi in hardware-in-the-loop simulations to show how ETHILE and the guidance algorithm interface; moreover, closed-loop simulations are run to show the effectiveness of the three steps guidance scheme. In all simulations, the parameters of the SCP algorithm are taken from the literature [17] and the Embedded COnic Solver (ECOS) is used [26]. All simulations are run considering $T_{\max} = 2mN$ and $I_{\mathrm{sp}} = 2000s$.

4.1 Hardware-In-The-Loop Simulations

We run Steps 1 and 3 of the algorithm (i.e., we compute the regularized thrust commands for the whole duration of the interplanetary cruise) on a Raspberry Pi Model 4 Model B with 4GB RAM with a Broadcom 2711, 64-bit quad-core Cortex-A72 processor and execute the guidance solution with ETHILE. This analysis serves to show the way the algorithm is run and its solution executed within the EXTREMA project. The algorithm has been written in MATLAB®, and the MATLAB® Coder is

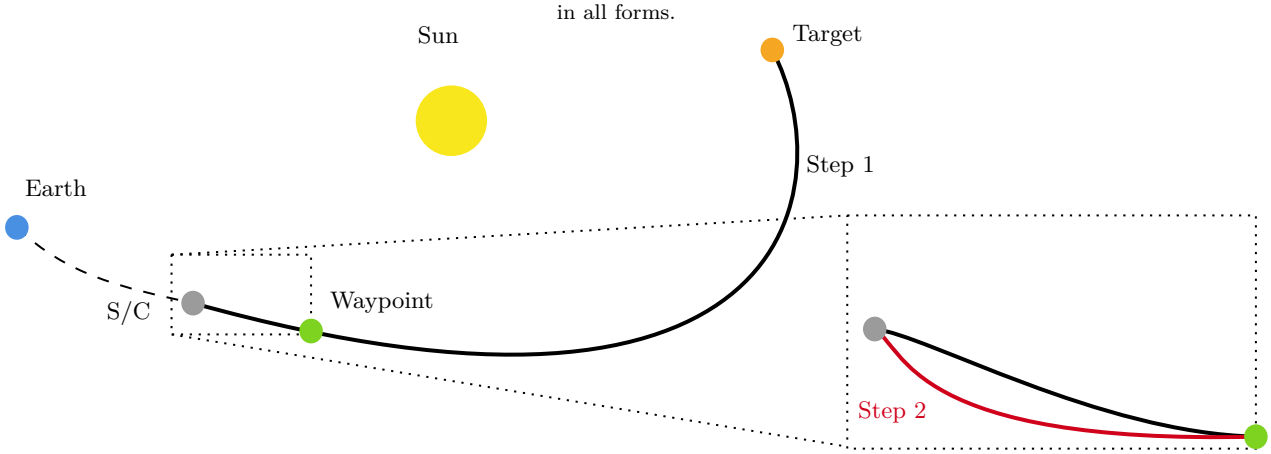

Fig. 2: Representation of the logic of Steps 1 and 2.

Tab. 1: Data of the considered scenarios [25].

# Scenario	ID	m_0 [kg]	$\|\mathbf{r}_0\|$[AU]	$\|\mathbf{v}_0\|$ [VU]	$\|\mathbf{r}_f\|$ [AU]	$\|\mathbf{v}_f\|$ [VU]	ToF [days]
1	esh_ic_2023-08-29_14-51-40	18.87	1.30	0.83	2.45	0.61	1850
2	esh_ic_2023-08-29_15-38-04	21.05	1.36	0.80	1.62	0.80	1840
3	esh_ic_2023-08-29_15-59-49	23.62	1.16	0.90	2.10	0.68	1990
4	esh_ic_2023-08-29_16-18-38	16.75	1.25	0.86	1.94	0.73	1400
5	esh_ic_2023-08-29_16-55-03	19.74	1.13	0.92	1.70	0.79	1640
6	esh_ic_2023-08-29_17-36-57	23.72	1.41	0.78	2.01	0.70	1900
7	esh_ic_2023-08-29_18-36-59	13.10	1.05	1.01	2.38	0.60	1090
8	esh_ic_2023-08-29_19-17-19	15.69	1.10	0.96	2.24	0.70	1710
9	esh_ic_2023-08-29_19-57-43	24.98	1.04	1.00	2.26	0.72	2070
10	esh_ic_2023-08-29_20-57-42	15.46	1.07	0.96	1.82	0.71	1630

used to generate executable C code. Step 3 is performed with the MATLAB ® function *fmincon* with the *ode45* integrator. Although a relatively high number of nodes $N = 250$ is used, the computational time of the whole procedure (Steps 1 and 3) takes less than 2 minutes for each generated trajectory. Figure 4 shows four of the thrust profiles that have been calculated with the algorithm on the Raspberry Pi and consequently executed by ETHILE. The noise of the thrust is under the 10% of the nominal level. Usually, standard low-thrust engines have execution errors in the order of 2-3% [27]. However, propulsion systems for CubeSats like M-ARGO [3] are still under development and may have higher execution errors. Therefore, the entity of the noise ETHILE has is compatible with such engines.

4.2 Closed-Loop Simulations

The whole three steps guidance process is instead tested in closed-loop simulations performed in MATLAB ® version R2022b on an Intel Core i7-10700@2.90 GHz desktop computer with 16 GB of RAM. We select Scenarios #2 and #7 of the scenarios from Ref. [25] used for the hardware-in-the-loop simulations and run closed-loop simulations for each of them. After every trajectory re-computation, the state of the spacecraft after $K = 1$ duty cycles is randomly perturbed to simulate the deviation from the nominal trajectory, and the guidance algorithm is run again. The process stops if either the algorithm is not able to converge to a solution or when the residual time of flight to reach the target is less than the duration of a duty cycle. At each trajectory recomputation, the solution obtained at the previous duty cycle is used as initial guess of the new SCP optimization. Table 2 summarizes the main parameters that have been used to run the closed-loop simulations. The state at the beginning of each duty cycle is perturbed with decreasing values from the beginning of the transfer to the end. This is to simulate the increase of performance of the autonomous navigation algorithm [28].

Figures 5 and 6 show the envelop of the trajectories computed at each duty cycle, together with the points in space where the re-computation happened (red circles), for the considered scenarios. Due to the large perturbations, the trajectories differ significantly from one another. The fixed final time algorithm always converged

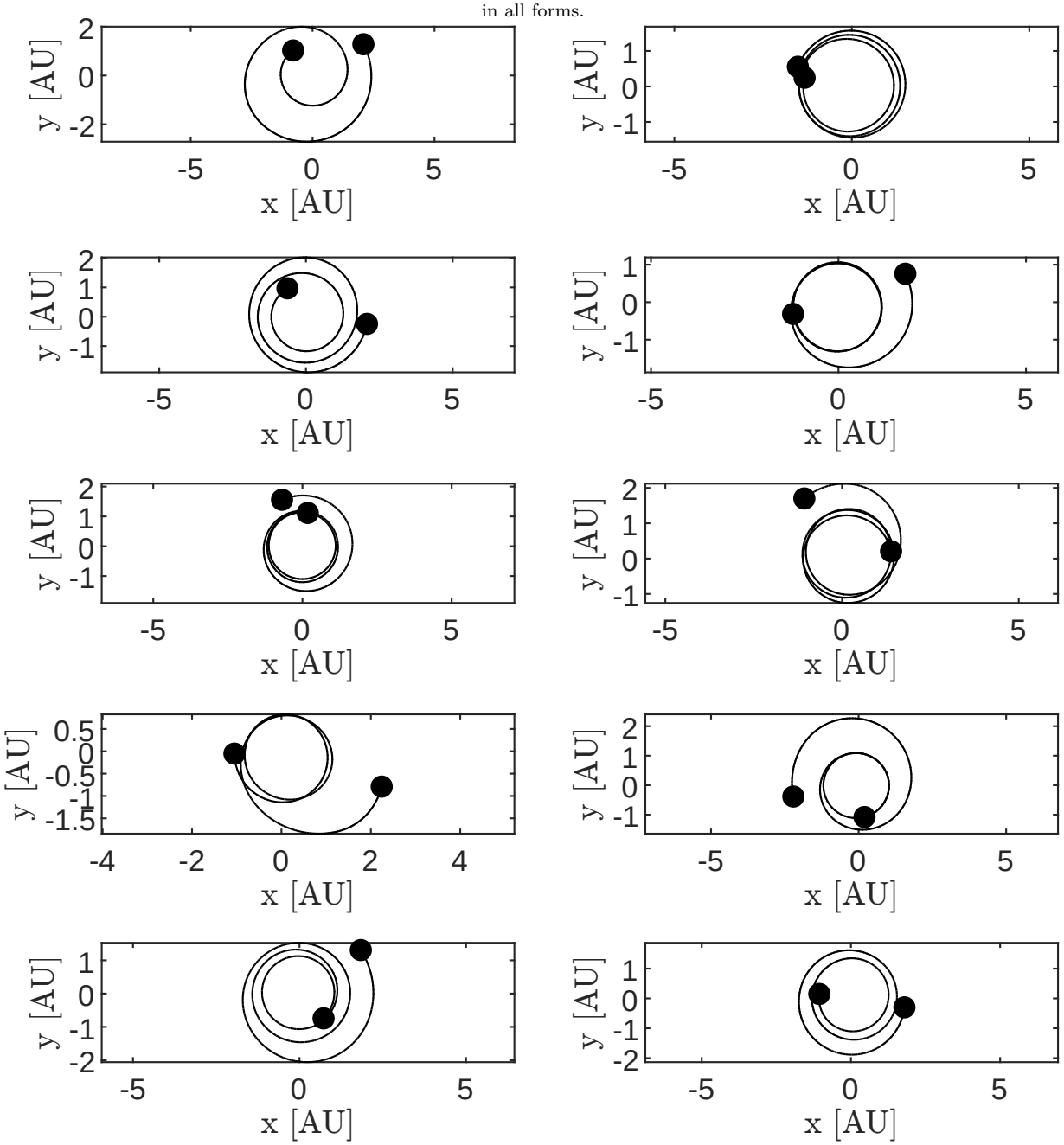

Fig. 3: Nominal trajectories associated with the 10 selected scenarios.

Tab. 2: Duty cycles parameters.

Parameter	Value
Duty cycle parameter n	8
Duty cycle parameter m	1
Safety factor ξ	0.95

for Scenario #2, and therefore the final boundary conditions and the time of flight did not change throughout the interplanetary cruise. For Scenario #7, on the other hand, the fixed final time algorithm failed to converge in some cases due to the large perturbations and therefore the final boundary conditions were changed. The free final time algorithm assumes that the final target moves in a two-body motion around the Sun. Figures 7 and 8 present the closed-loop thrust profile (black solid line) computed using Steps 1 to 3 of the algorithm, where the duty cycle constraints are imposed in Step 2. Moreover, the figures show the nominal thrust profile (red solid line) obtained using Steps 1 and 2. Due to the large

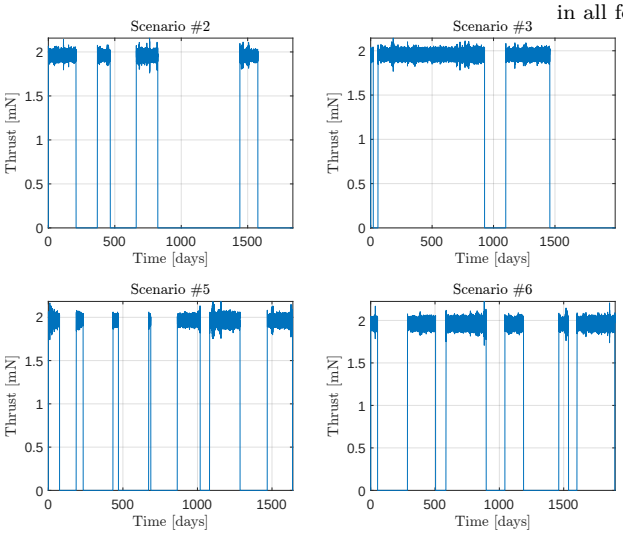

Fig. 4: Four of the thrust profiles executed by ETHILE.

perturbations, the two profiles are considerably different. However, the closed-loop guidance approach allows to re-optimize the trajectory after every duty cycle, and therefore the fuel mass consumption of the closed-loop profile is not significantly different from the one associated with the nominal thrust profile. For Scenario #7, note that because the variable time of flight algorithm was used, the nominal profile terminates before the closed-loop one.

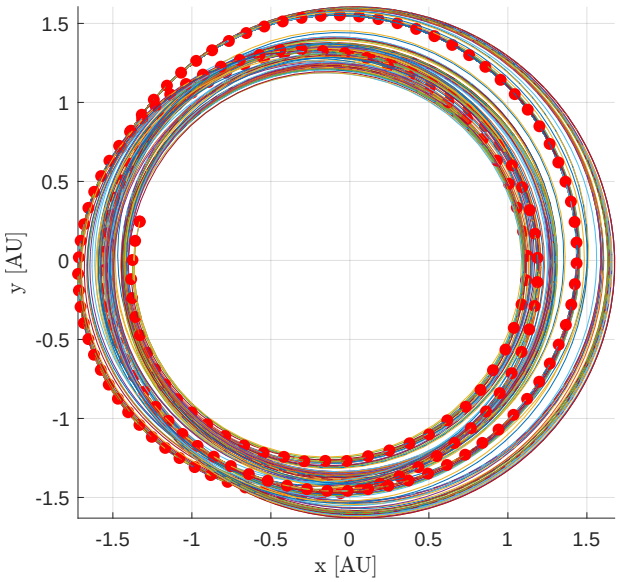

Fig. 5: Envelop of closed-loop trajectories associated with Scenario #2.

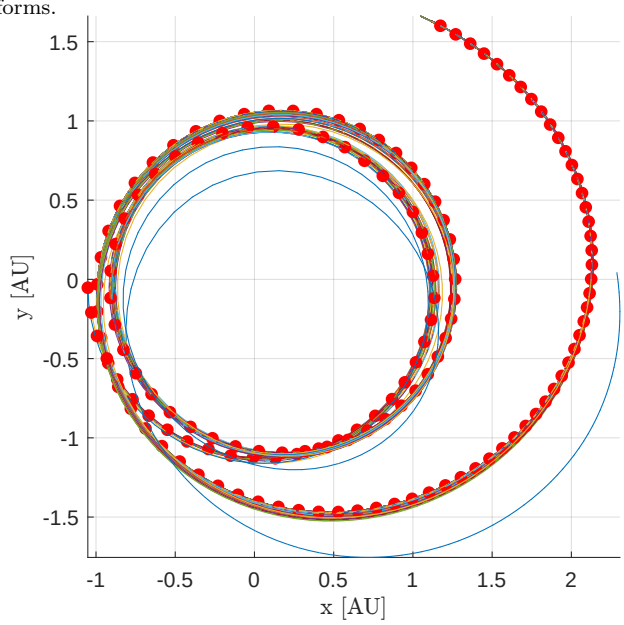

Fig. 6: Envelop of closed-loop trajectories associated with Scenario #7.

5 Conclusions

In this work, the architecture of the autonomous guidance algorithm that is being used within the EXTREMA project at Politecnico di Milano has been described. The algorithm is based on convex optimization and it exploits a regularization process to make the thrust commands. The algorithm has shown to be effective when run on a Raspberry Pi and its computed commands were correctly executed by the in-house built facility ETHILE. Moreover, duty cycle constraints were introduced thanks to a three-steps process to avoid a degradation of the algorithm performance in terms of reliability and computational burden. Further work will consist of enhancing the second step with high-fidelity dynamics and other relevant operational constraints.

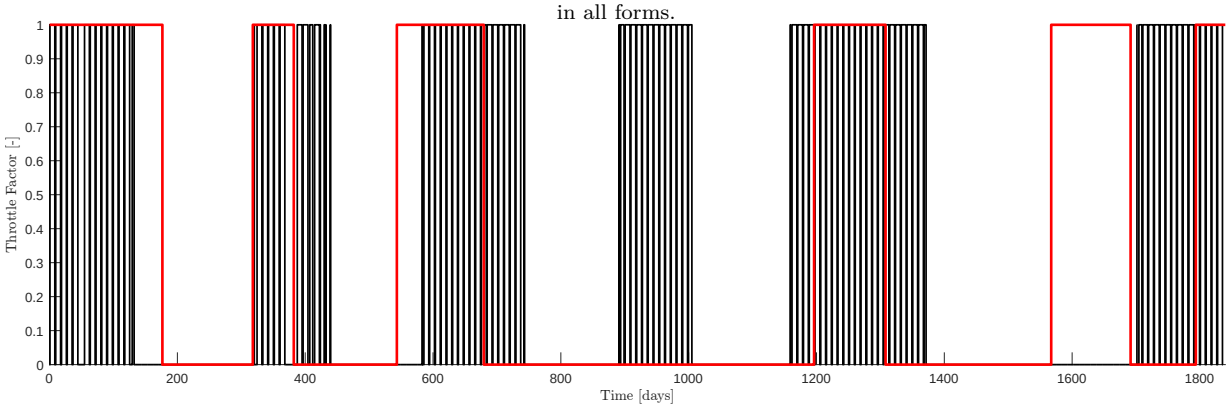

Fig. 7: Nominal and closed-loop thrust profile associated with Scenario #2.

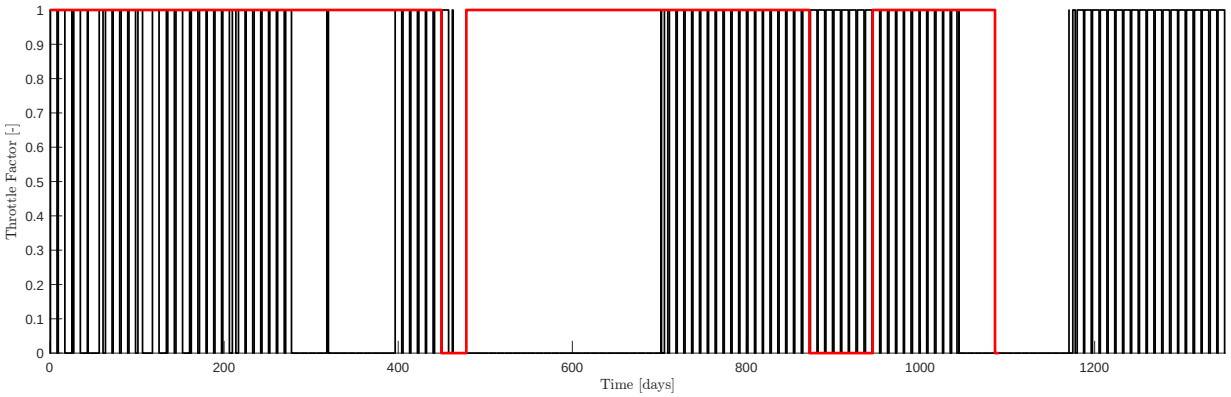

Fig. 8: Nominal and closed-loop thrust profile associated with Scenario #7.

Acknowledgments

A.C.M. would like to thank the Agenzia Spaziale Italiana (ASI) and the Space Generation Advisory Council (SGAC) for the finantial support granted to attend the 71st Space Generation Congress (SGC) and the 74th International Astronautical Congress (IAC). 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 programme (Grant Agreement No. 864697).

Bibliography

[1] R. Walker, D. Binns, C. Bramanti, M. Casasco, P. Concari, D. Izzo, D. Feili, P. Fernandez, J. G. Fernandez, P. Hager *et al.*, "Deep-space CubeSats: thinking inside the box," *Astronomy & Geophysics*, vol. 59, no. 5, pp. 24–30, 2018.

[2] E. Dotto, V. Della Corte, M. Amoroso, I. Bertini, J. Brucato, A. Capannolo, B. Cotugno, G. Cremonese, V. Di Tana, I. Gai, S. Ieva, G. Impresario, S. Ivanovski, M. Lavagna, A. Lucchetti, E. Mazzotta Epifani, A. Meneghin, F. Miglioretti, D. Modenini, M. Pajola, P. Palumbo, D. Perna, S. Pirrotta, G. Poggiali, A. Rossi, E. Simioni, S. Simonetti, P. Tortora, M. Zannoni, G. Zanotti, A. Zinzi, A. Cheng, A. Rivkin, E. Adams, E. Reynolds, and K. Fretz, "LICIACube - The Light Italian Cubesat for Imaging of Asteroids In support of the NASA DART mission towards asteroid (65803) Didymos," *Planetary and Space Science*, vol. 199, p. 105185, 2021.

[3] F. Topputo, Y. Wang, G. 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, pp. 4193–4221, 2021.

[4] P. Lu, "Introducing computational guidance and control," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 193–193, 2017.

[5] J. T. Betts, "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193 – 207, 1998.

[6] 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, pp. 1379–1388, 2021.

[7] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004, pp. 21–187.

[8] R. Bonalli, A. Cauligi, A. Bylard, and M. Pavone, "GuSTO: Guaranteed Sequential Trajectory optimization via Sequential Convex Programming," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6741–6747.

[9] Y. Mao, M. Szmuk, and B. Açıkmeşe, "Successive convexification of non-convex optimal control problems and its convergence properties," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 3636–3641.

[10] X. Liu, P. Lu, and B. Pan, "Survey of convex optimization for aerospace applications," *Astrodynamics*, vol. 1, no. 1, pp. 23–40, Sep. 2017.

[11] M. Sagliano, "Generalized *hp* pseudospectral-convex programming for powered descent and landing," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 7, pp. 1562–1570, Apr. 2019.

[12] X. Liu, Z. Shen, and P. Lu, "Entry trajectory optimization by second-order cone programming," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 2, pp. 227–241, Aug. 2016.

[13] 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, pp. 2274–2290, Mar. 2018.

[14] G. Di Domenico, E. Andreis, A. C. Morelli, G. Merisio, V. Franzese, C. Giordano, A. Morselli, P. Panicucci, F. Ferrari, and F. Topputo, "The ERC-Funded EXTREMA Project: Achieving Self-Driving Interplanetary CubeSats," in *Modeling and Optimization in Space Engineering: New Concepts and Approaches*. Springer, 2022, pp. 167–199.

[15] A. Morselli, A. C. Morelli, and F. Topputo, "Ethile: A thruster-in-the-loop facility to enable autonomous guidance and control of autonomous interplanetary cubesat," in *73rd International Astronautical Congress (IAC 2022)*, 2022, pp. 1–10.

[16] A. C. Morelli, G. Merisio, C. Hofmann, and F. Topputo, "A Convex Guidance Approach to Target Ballistic Capture Corridors at Mars," in *44th AAS Guidance, Navigation and Control Conference*, 2022, pp. 1–24.

[17] C. Hofmann, A. C. Morelli, and F. Topputo, "Performance Assessment of Convex Low-Thrust Trajectory Optimization Methods," *Journal of Spacecraft and Rockets*, vol. 60, no. 1, 2023.

[18] C. Hofmann and F. 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*, 2021, pp. 1–15.

[19] C. Hofmann, "Computational guidance for low-thrust spacecraft in deep space based on convex optimization," Ph.D. dissertation, 2023.

[20] 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*, vol. 58, no. 3, pp. 2103–2116, 2021.

[21] D. Oh, D. Landau, T. Randolph, P. Timmerman, J. Chase, J. Sims, and T. Kowalkowski, "Analysis of system margins on deep space missions using solar electric propulsion," in *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, 2008, p. 5286.

[22] N. Kumagai and K. Oguri, "Adaptive-Mesh Sequential Convex Programming for Space Trajectory Optimization," in *AAS/AIAA Astrodynamics Specialist Conference, Big Sky, MT, AAS*, 2023.

[23] X. Zhou, R.-Z. He, H.-B. Zhang, G.-J. Tang, and W.-M. Bao, "Sequential convex programming method using adaptive mesh refinement for entry trajectory planning problem," *Aerospace Science and Technology*, vol. 109, p. 106374, 2021.

[24] A. C. Morelli, A. Morselli, C. Giordano, and F. Topputo, "Convex Trajectory Optimization using Thrust Regularization," 2023, under review.

[25] G. Merisio, "EXTREMA: Random autonomous interplanetary mission scenarios for EXTREMA Simulation Hub (ESH) hardware-in-the-loop simulations," Aug. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.8300807

[26] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP Solver for Embedded Systems," in *European*

Control Conference, Zurich, Switzerland, 2013, pp. 3071–3076.

[27] V. Franzese, F. Topputo, F. Ankersen, and R. Walker, "Deep-space optical navigation for M-ARGO mission," *The Journal of the Astronautical Sciences*, vol. 68, no. 4, pp. 1034–1055, 2021.

[28] E. Andreis, V. Franzese, and F. Topputo, "On-board Orbit Determination for Deep-Space CubeSats," *Journal of Guidance, Control, and Dynamics*, pp. 1466–1479, 2022.