Image-based Meta-Reinforcement Learning for Autonomous Guidance of an Asteroid Impactor

Lorenzo Federici*
Sapienza University of Rome, Via Eudossiana 18, 00184, Rome, Italy

Andrea Scorsoglio†, Luca Ghilardi‡
University of Arizona, 1127 E. James E. Rogers Way, 85721, Tucson, AZ (USA)

Andrea D'Ambrosio§
Sapienza University of Rome, Via Salaria 851, 00138, Rome, Italy

Boris Benedikter¶, Alessandro Zavoli ‖
Sapienza University of Rome, Via Eudossiana 18, 00184, Rome, Italy

Roberto Furfaro**
University of Arizona, 1127 E. James E. Rogers Way, 85721, Tucson, AZ (USA)
University of Arizona, 1130 N. Mountain Ave., 85721, Tucson, AZ (USA)

This paper focuses on the use of meta-reinforcement learning for the autonomous guidance of a spacecraft during the terminal phase of an impact mission towards a binary asteroid system. The control policy is replaced by a convolutional-recurrent neural network, which is used to map optical observations collected by the onboard camera to the control thrust and thrusting times. The network is trained by a proximal policy optimization algorithm, a family of reinforcement learning methods. The final phase of NASA's Double Asteroid Redirection Test (DART) mission is used as a test case. The objective is to maneuver the spacecraft to impact the smaller object, Dimorphos, in the Didymos binary system. The spacecraft dynamics are described using the bi-elliptic restricted four-body problem with solar radiation pressure. The initial conditions are randomly scattered according to the actual specifications of the DART mission. A random error on the orbital position of Dimorphos is also considered to reflect uncertainty on the binary system's characteristics and dynamics. The control system aims at minimizing the error on the final spacecraft position. Numerical results show that the guidance system can correctly drive the spacecraft towards the final impact point in more than 98% of the 500 test scenarios.

*PhD candidate, Department of Mechanical and Aerospace Engineering; lorenzo.federici@uniroma1.it
†PhD student, Department of Systems & Industrial Engineering, andreascorsoglio@email.arizona.edu
‡PhD student, Department of Systems & Industrial Engineering, lucaghilardi@email.arizona.edu
§PhD candidate, School of Aerospace Engineering; andrea.dambrosio@uniroma1.it
¶PhD candidate, Department of Mechanical and Aerospace Engineering, boris.benedikter@uniroma1.it
‖Researcher, Department of Mechanical and Aerospace Engineering, alessandro.zavoli@uniroma1.it
**Professor, Department of Systems & Industrial Engineering, Department of Aerospace and Mechanical Engineering, robertof@email.arizona.edu
Presented at the AIAA SciTech 2022 Forum, January 3-7, 2022, San Diego (CA) & virtual, paper number AIAA 2022-2270.

Nomenclature

A^{π}	=	advantage function
$\boldsymbol{d}$	=	vector connecting the primary asteroid to the secondary asteroid
$\text{diag}(\boldsymbol{A})$	=	matrix composed of the diagonal elements of matrix $\boldsymbol{A}$
$\mathop{\text{E}}_{\tau}[v]$	=	expected value of variable $v(\tau)$ with respect to random variable τ
$\boldsymbol{f}$	=	spacecraft dynamical model
J	=	merit index
$\hat{\boldsymbol{l}}$	=	camera line-of-sight vector
m	=	spacecraft mass, kg
M	=	mean anomaly, rad
H	=	number of time-steps per episode
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	=	Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\boldsymbol{y}$	=	observation vector
R	=	reward
$\boldsymbol{r}$	=	position vector, km
$\boldsymbol{x}$	=	state vector
K	=	total number of training steps
t	=	time, s
t_f	=	maximum duration of the terminal guidance phase, s
$\boldsymbol{T}$	=	thrust, kN
$\boldsymbol{u}$	=	control vector
$\mathcal{U}(a, b)$	=	uniform distribution in interval $[a, b]$
V^{π}	=	value function
$\boldsymbol{v}$	=	velocity vector, km/s
θ	=	neural network's parameters
μ	=	gravitational parameter, km^3/s^2
π	=	control policy
σ	=	cumulative nondimensional gravitational parameter of the primaries
σ_x	=	standard deviation of Gaussian random variable x
τ	=	trajectory
φ, ψ	=	in-plane and out-of-plane angles, rad
φ_S	=	solar phase angle, rad

Subscripts

h	=	value at h-th time-step
I	=	value at impact
p_i	=	value referred to body p_i, $i = 1, \ldots, 4$
$p_i p_j$	=	vector connecting body p_i to body p_j, $i = 1, \ldots 4$, $j \neq i = 1, \ldots, 4$
max	=	maximum value

Superscripts

$\star$	=	optimal value
$\hat{}$	=	estimate
$\bar{}$	=	reference value
(j)	=	j-th component of a vector
$\top$	=	transpose

I. Introduction

Over the recent years, space missions to asteroids have gained increasing interest from the scientific community. This is mainly due to the valuable information that these small bodies can reveal about our Solar system, including planetary formation and evolution, and the origin of life on Earth. Indeed, several space missions have already been sent towards asteroids, such as the NEAR (Near Earth Asteroid Rendezvous) Shoemaker [1], Dawn [2], Hayabusa 1 and 2 [3, 4], and OSIRIS-REx (Origins, Spectral Interpretation, Resource Identification, Security, Regolith Explorer) [5]. Furthermore, a new generation of asteroid exploration missions are being designed, or about to be launched in the next few years, from different space agencies, including The Near-Earth Asteroid Scout [6], Lucy [7], Hera and DART (Double Asteroid Redirection Test) [8]. In particular, the latter mission has been launched in November 2021 and is directed to a synchronous binary asteroid system called 65803 Didymos. The main body of the system, simply known as Didymos or Didymain, is a near-Earth sub-kilometer (780 m) asteroid, classified as a potentially hazardous object within the Apollo and Amor groups. Its small 170-meter minor-planet moon, discovered in 2003 and named Dimorphos in 2020 (but often referred to as Didymoon) is just 1.2-kilometer away from the primary asteroid. The origin of binary systems such as 65803 Didymos is still under investigation by scientists, and future explorations missions to such systems can shed light on their formation. In the case of the DART mission, having a binary system is also crucial to achieving the main goal of the mission, which is to measure the deviation of Dimorphos's trajectory around Didymos after a kinetic impact with the spacecraft itself. The outcome of the mission will provide the scientific community with fundamental clues for planetary defense purposes, such as understanding if the impact with a spacecraft could be a viable option to counteract possible threats by hazardous near-Earth objects in the future.

The terminal guidance of an asteroid impactor as DART is a complex task. Indeed, after a long interplanetary cruise, the spacecraft is required to hit a body whose size is much smaller than the distances involved, and with a precise impact angle, in order to maximize the target body's trajectory deflection. In a binary asteroid system, the task is further complicated by the complex dynamical environment, characterized by the gravitational influence of two small irregularly-shaped bodies, which cause significant orbital perturbations. Additionally, the Sun's effect on the spacecraft motion cannot be neglected within this low-gravity environment, both because of its gravitational influence and for the acceleration caused by the Solar radiation pressure (SRP). Many works in the literature have already dealt with the study of the dynamics and control of a spacecraft around binary asteroid systems. For example, periodic orbits of a solar sail around a binary system have been analyzed by considering different dynamical models, such as the Hill four-body problem plus SRP and the bi-circular four-body problem plus SRP [9]. Solar sails have also been considered to study the control effort required to maintain a hovering orbit about binary systems [10], taking into account the irregular shape of both the asteroids, or to perform station-keeping around the Lagrangian point L_4, within the framework of the elliptic restricted three-body problem (ER3BP) plus SRP [11]. In addition, ballistic landing trajectories have been studied to deploy science packages on binary systems [12]. In this case, the circular restricted three-body problem (CR3BP) has been employed. The same dynamical model has also been used to study stable regions and periodic orbits around L_4 and L_5 of the 1999 KW4 binary system [13]. A modified CR3BP, called Shape-based CR3BP (SCR3BP), has been taken into account to obtain bounded orbits near Didymos, by considering a realistic shape for the primary and an ellipsoidal shape for the secondary [14].

All of these studies agree that considering a high-fidelity dynamical model is essential for the correct design of a mission around a binary asteroid system. This is especially true for a kinetic impactor mission, as also the smallest deviation from the nominal trajectory can cause the spacecraft to miss the target and, consequently, lead to a failure of the mission. For this reason, the presence onboard of a very accurate navigation and control system, with an enhanced robustness against possible maneuvering errors and/or model uncertainties, is a crucial requirement of the mission. Furthermore, a guidance system capable of autonomously deciding the corrective control on the basis of real-time measurements would be preferable to a more traditional ground-controlled system. This is due to the short time-length of the terminal approach maneuver, which, in most of the cases, can be of the same order of magnitude as the communication delay with the Earth.

Consequently, the DART spacecraft is equipped with SMARTNav, acronym of Small-body Maneuvering Autonomous Real-Time Navigation, an autonomous navigation system designed to control and keep the spacecraft on its path towards Dimorphos entirely on its own, without the need for any human intervention [15]. In particular, SMARTNav starts operating roughly four hours before the designated impact time, when the spacecraft is about 90 000 km away from the target, and controls DART's trajectory until roughly 2 minutes from the goal, when a ballistic flight will naturally drive the spacecraft towards a head-on collision with the asteroid. SMARTNav just relies on the high-resolution images

provided by the onboard camera DRACO (Didymos Reconnaissance and Asteroid Camera for OpNav) [16], the only payload of DART. Such images are used to locate in real-time the two asteroids in space, identify the target asteroid, and estimate the trajectory corrections and commanding maneuvers necessary to hit Dimorphos. SMARTNav might open a broad range of new possibilities in the context of deep-space guidance, by proving for the first time that is possible to locate and fly towards an (eventually unknown) target in complete autonomy and with unprecedented accuracy.

It is worth mentioning that different works in the literature already studied the possibility of using an optical-only guidance, navigation, and control (GNC) system during an asteroid impact mission. More precisely, an autonomous GNC strategy, composed of a image processing and filtering followed by a targeting algorithm based on zero-effort errors, has been recently investigated and applied to a simulated impact mission towards asteroid Bennu [17]. A GPU-based optical navigation, which uses the light intensity of the image pixels to determine the line-of-sight vector, has been coupled with traditional guidance strategies, such as proportional or predictive guidance laws, to control a spacecraft during a simulated impact mission towards a scaled version of asteroid 433 Eros [18].

In this paper, we investigate the preliminary design of an alternative system for the autonomous guidance and navigation of a kinetic impactor towards a binary asteroid system. The proposed approach is based on deep meta-reinforcement learning (meta-RL). Guidance and navigation systems based on the use of deep neural networks (DNN) and reinforcement learning (RL) are becoming increasingly popular also in research works on deep-space applications, other than in the robotics [19], automotive [20], and video-game fields [21]. Such systems exploit the low computational times and the high accuracy of DNNs as universal function approximators to compute in real-time a closed-loop control law to be deployed on the onboard hardware, based on measurements, or observations, collected by the navigation filter [22]. The DNN is trained offline (i.e., on the ground) to solve an optimal control problem (OCP), by leveraging training data collected during repeated simulations, or rollouts, of the considered mission scenario. A numerical reward, provided by the environment, is associated with each observation-control tuple as a performance measure during the network training. After each policy rollout, the network's parameters are updated to maximize the average cumulative reward over a single trajectory. The lack of direct dependence of the optimal control policy on the exact mathematical expression of the dynamical and observation model, which can be also arbitrarily complex black-box functions, makes RL a perfect candidate to design robust guidance systems, able to cope with any kind of dynamical uncertainties, noisy observations, and control errors.

For these reasons, several research papers have already dealt with the use of RL for the closed-loop guidance and control of spacecraft. In particular, RL has been employed to study many different mission scenarios, ranging from interplanetary [23–25] cislunar [26–28] and LEO-GEO [29, 30] trajectory design, to rendezvous missions [31–33], path planning for asteroid hopping rovers [34], formation flying [35], and planetary landing [36, 37].

Furthermore, the combination of recurrent neural networks (RNNs) with RL brought to the definition of what is commonly known as deep meta-reinforcement learning (meta-RL) [38], or learning to learn, which is the optimization

framework employed in this work. RNNs are particular types of DNNs that can keep track of the temporal variation of the observations collected over training in internal network states, thanks to feedback connections. The presence of the internal states, which contain information on the evolution of the input data, allows the network to better specialize the control outputs referred to different instances of the considered environment. This capability significantly boosts the average performance achieved by the policy network in complex scenarios, such as non-Markov, multiple-task, uncertain or partially-observable environments. Meta-RL versatility is confirmed by a number of works that used it to study asteroid close-proximity operations and landing [39], body-fixed hovering over unmapped asteroids [40], multi-target interplanetary missions [41] or image-based lunar landing [42, 43].

The guidance system designed in this work is supposed to exploit the same kind and amount of information provided to a device like SMARTNav, i.e., optical images. During simulations, the images are generated and realistically rendered in real-time using the open-source computer graphics software Blender [44], which can be easily interfaced with a python programming environment. The initial spacecraft state is derived from the conditions at impact provided by mission design, by using simplified dynamics (two-body). A three-dimensional bi-elliptic restricted four-body problem (BER4BP), which also includes the presence of the Solar radiation pressure, is used as dynamical model to take into account the influence of both the asteroids and of the Sun on the spacecraft motion. A random error on the initial position of the minor asteroid along its orbit about the primary is considered too. Proximal policy optimization (PPO) [45] is used as optimization algorithm to teach a deep convolutional neural network, with a recurrent layer, how to control the spacecraft along its impact trajectory and compensate for the deviations caused by the unmodeled dynamical perturbations and uncertainties. The entire framework is then applied to the terminal guidance of DART, used as a test scenario for a fair comparison with SMARTNav.

This paper is organized as follows. Section II introduces the dynamical model, together with the mathematical approach employed to generate feasible initial conditions for the spacecraft. Section III formulates the simulation environment as a partially observable Markov decision process, and Section IV introduces the meta-RL framework used to solve the control problem. Numerical results are provided in Section V, and, eventually, concluding remarks are given in the last section.

II. Dynamical Model

This section presents the spacecraft dynamics about the binary system barycenter, together with the simplified model used to determine the initial spacecraft state starting from known mission data. The final conditions of the terminal guidance phase are also provided.

A. Uncontrolled Dynamics

A restricted four-body problem (R4BP) is introduced to take into account the effect of the primary body p_1 (Sun), and two secondary bodies p_2 and p_3 (binary asteroid system) on the motion of a point-mass spacecraft (S/C) p_4. The motion of the barycenter of the asteroid system (b) around the Sun is considered to be elliptic, out of the ecliptic plane. The motion of p_3 and p_2 around b is instead supposed to be circular and in a retrograde orbital plane. In this way, a 3-dimensional R4BP with a non-coplanar motion of the primaries is derived, which is named as bi-elliptic restricted four-body problem (BER4BP) [46].

Let us consider a reference frame $P = (b; \hat{\boldsymbol{x}}_P, \hat{\boldsymbol{y}}_P, \hat{\boldsymbol{z}}_P)$, which corresponds to the perifocal frame referred to the orbit of p_3 around p_2, and let $\boldsymbol{r}$ denote the spacecraft position with respect to b. Assuming a Newtonian gravity model, the equations of motion for the 4-th body p_4, in frame P, are

$$\ddot{\boldsymbol{r}} = -\sum_{i=1}^{3} \mu_{p_i} \frac{\boldsymbol{r}_{p_i p_4}}{\left\|\boldsymbol{r}_{p_i p_4}\right\|^3} + \ddot{\boldsymbol{r}}_b \tag{1}$$

where $\ddot{\boldsymbol{r}}_b$ is the acceleration of the barycenter b with respect to an inertial frame, written in frame P, μ_{p_i} is the gravitational constant of body p_i, and $\boldsymbol{r}_{p_i p_j} = \boldsymbol{r}_{p_j} - \boldsymbol{r}_{p_i}$ indicates a vector connecting p_i to p_j, with $\boldsymbol{r}_{p_i}$ the position vector of p_i with respect to b. In particular, by referring to Fig. 1

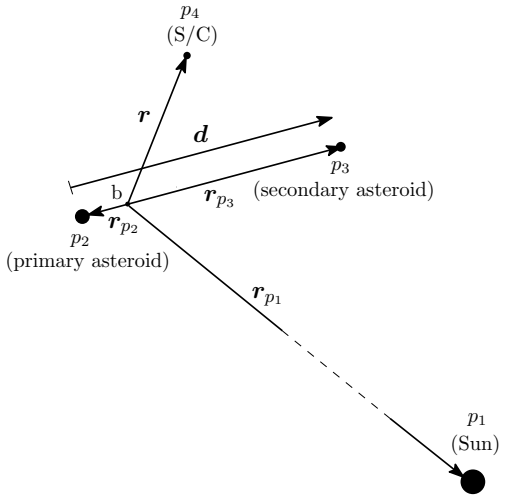


Fig. 1 Schematic of the relative position among the four bodies.

$$
\begin{aligned}
\boldsymbol{d} &= \boldsymbol{r}_{p_3} - \boldsymbol{r}_{p_2} \\
\boldsymbol{r}_{p_1p_2} &= -\boldsymbol{r}_{p_1} - \mu\boldsymbol{d} \\
\boldsymbol{r}_{p_1p_3} &= -\boldsymbol{r}_{p_1} + (1-\mu)\boldsymbol{d} \\
\boldsymbol{r}_{p_1p_4} &= -\boldsymbol{r}_{p_1} + \boldsymbol{r} \\
\boldsymbol{r}_{p_2p_4} &= \boldsymbol{r} + \mu\boldsymbol{d} \\
\boldsymbol{r}_{p_3p_4} &= \boldsymbol{r} - (1-\mu)\boldsymbol{d}
\end{aligned}
\tag{2}
$$

where $\mu = \frac{\mu_{p_3}}{\mu_{p_2}+\mu_{p_3}}$ is the mass ratio of the secondary bodies.

The acceleration of the binary system barycenter b is, according to the two-body dynamics

$$
\ddot{\boldsymbol{r}}_b = \mu_{p_1}\left[(1-\mu)\frac{-\boldsymbol{r}_{p_1} - \mu\boldsymbol{d}}{\left\|\boldsymbol{r}_{p_1p_2}\right\|^3} + \mu\frac{-\boldsymbol{r}_{p_1} + (1-\mu)\boldsymbol{d}}{\left\|\boldsymbol{r}_{p_1p_3}\right\|^3}\right] \tag{3}
$$

By combining Eqs. (1)–(3), it is possible to rewrite the spacecraft acceleration in P as

$$
\begin{aligned}
\ddot{\boldsymbol{r}} =& \mu_{p_1}\left[(1-\mu)\frac{-\boldsymbol{r}_{p_1} - \mu\boldsymbol{d}}{\left\|\boldsymbol{r}_{p_1p_2}\right\|^3} + \mu\frac{-\boldsymbol{r}_{p_1} + (1-\mu)\boldsymbol{d}}{\left\|\boldsymbol{r}_{p_1p_3}\right\|^3} - \frac{-\boldsymbol{r}_{p_1} + \boldsymbol{r}}{\left\|\boldsymbol{r}_{p_1p_4}\right\|^3}\right] + \\
&- \mu_{p_2}\frac{\boldsymbol{r} + \mu\boldsymbol{d}}{\left\|\boldsymbol{r}_{p_2p_4}\right\|^3} - \mu_{p_3}\frac{\boldsymbol{r} - (1-\mu)\boldsymbol{d}}{\left\|\boldsymbol{r}_{p_3p_4}\right\|^3}
\end{aligned}
\tag{4}
$$

Let us assume that body p_3 moves along a circular Keplerian orbit about p_2, with classical orbital parameters $\{a_{p_3} = d, i_{p_3}, \Omega_{p_3}, M_{p_3}, \omega_{p_3} = 0, M_{p_3} = 0\}$, where a indicates the semi-major axis, i the orbital inclination, Ω the right ascension of the ascending node, ω the argument of the pericenter and M the mean anomaly at current time. Similarly, the binary system is supposed to move along an elliptic orbit about the Sun p_1, with classical orbital parameters $\{a_b, e_b, i_b, \Omega_b, \omega_b, M_b\}$. Physical quantities are made non-dimensional through the reference units

$$
\bar{r} = d, \qquad \bar{\mu} = \mu_{p_2} + \mu_{p_3}, \qquad \bar{t} = \sqrt{\frac{d^3}{\bar{\mu}}}, \qquad \bar{v} = \frac{\bar{r}}{\bar{t}}, \qquad \bar{a} = \frac{\bar{v}}{\bar{t}} \tag{5}
$$

that is, the semi-major axis of the orbit of p_3 about p_2, the cumulative gravitational constant of the binary system and the orbital period of p_3 about p_2, divided by 2π. In this way, relevant quantities are in a small range around unity, reducing the errors related to the finite precision of the computer.

By passing from reference frame P to a rotating frame $N = (b; \hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$, which corresponds to the radial-transverse-normal (RTN) frame associated to the orbital motion of p_3 around p_2, the non-dimensional equations of motion of the

spacecraft in frame N can be written as

$$\dot{\boldsymbol{r}} = \boldsymbol{v} \tag{6}$$

$$\dot{\boldsymbol{v}} = -2\,\hat{\boldsymbol{z}} \times \boldsymbol{v} - \hat{\boldsymbol{z}} \times (\hat{\boldsymbol{z}} \times \boldsymbol{r}) + (\sigma - 1)\left[(1-\mu)\frac{-\boldsymbol{r}_{p_1} - \mu\hat{\boldsymbol{x}}}{\left\|\boldsymbol{r}_{p_1 p_2}\right\|^3} + \right.$$
$$\left. + \mu\frac{-\boldsymbol{r}_{p_1} + (1-\mu)\hat{\boldsymbol{x}}}{\left\|\boldsymbol{r}_{p_1 p_3}\right\|^3} - \frac{-\boldsymbol{r}_{p_1} + \boldsymbol{r}}{\left\|\boldsymbol{r}_{p_1 p_4}\right\|^3}\right] + (1-\mu)\frac{\boldsymbol{r} + \mu\hat{\boldsymbol{x}}}{\left\|\boldsymbol{r}_{p_2 p_4}\right\|^3} - \mu\frac{\boldsymbol{r} - (1-\mu)\hat{\boldsymbol{x}}}{\left\|\boldsymbol{r}_{p_3 p_4}\right\|^3} \tag{7}$$

where, from now on, all vector quantities are supposed to be written in frame N, unless otherwise specified with proper subscripts. In particular, $\boldsymbol{d} = \hat{\boldsymbol{x}}, \hat{\boldsymbol{z}} = [0\ 0\ 1]^\top$ is the angular velocity of frame N with respect to frame P, and $\sigma = \frac{\mu_{p_1}}{\bar{\mu}} + 1$ the cumulative gravitational parameter of the primaries. The Sun position $\boldsymbol{r}_{p_1}$ at any time is computed by propagating its initial state $\boldsymbol{r}_{p_1,0}$, $\boldsymbol{v}_{p_1,0}$ forward in time through a Keplerian dynamics. So, it is just dependent on the current time t: $\boldsymbol{r}_{p_1} = \boldsymbol{r}_{p_1}(t)$.

The solar radiation pressure (SRP) acting on the spacecraft can be evaluated as

$$p_\odot = \frac{\phi_\odot}{c_0\ \left\|\boldsymbol{r}_{p_1 p_4,\mathrm{AU}}\right\|^2} \tag{8}$$

where $\phi_\odot = 1371\,\mathrm{W/m^2}$ is the Sun's irradiance at 1 AU, c_0 is the speed of light in vacuum and $\left\|\boldsymbol{r}_{p_1 p_4,\mathrm{AU}}\right\|$ the spacecraft-Sun distance expressed in AU.

So, the net perturbing acceleration due to the SRP is, in non-dimensional unit,

$$\boldsymbol{a}_\odot = \frac{1}{m}\frac{p_\odot A}{\bar{m}\bar{a}}\left(\hat{\boldsymbol{l}} \cdot \hat{\boldsymbol{r}}_{p_1 p_4}\right)\hat{\boldsymbol{r}}_{p_1 p_4} \tag{9}$$

where m indicates the non-dimensional mass of the spacecraft, $\bar{m} = 1000\,\mathrm{kg}$ a reference mass value, A is the total area of the spacecraft's solar panels in $\mathrm{m^2}$ and $\hat{\boldsymbol{l}}$ denotes a vector directed in the opposite direction to the normal to the solar panels. By supposing that the spacecraft is axial symmetric and that the onboard camera in mounted on the opposite side with respect to the solar panels, $\hat{\boldsymbol{l}}$ coincides with the camera line-of-sight. For simplicity, $\hat{\boldsymbol{l}}$ is here supposed to be known at any time t, and the spacecraft's attitude dynamics is neglected.

So, the final form of the ballistic equations of motion of the spacecraft is

$$\dot{\boldsymbol{r}} = \boldsymbol{v} \tag{10}$$

$$\begin{aligned}\dot{\boldsymbol{v}} = &-2\,\hat{\boldsymbol{z}} \times \boldsymbol{v} - \hat{\boldsymbol{z}} \times (\hat{\boldsymbol{z}} \times \boldsymbol{r}) + (\sigma - 1)\left[(1-\mu)\frac{-\boldsymbol{r}_{p_1} - \mu\hat{\boldsymbol{x}}}{\left\|\boldsymbol{r}_{p_1p_2}\right\|^3} + \right.\\ &\left. + \mu\frac{-\boldsymbol{r}_{p_1} + (1-\mu)\hat{\boldsymbol{x}}}{\left\|\boldsymbol{r}_{p_1p_3}\right\|^3} - \frac{-\boldsymbol{r}_{p_1} + \boldsymbol{r}}{\left\|\boldsymbol{r}_{p_1p_4}\right\|^3}\right] + (1-\mu)\frac{\boldsymbol{r} + \mu\hat{\boldsymbol{x}}}{\left\|\boldsymbol{r}_{p_2p_4}\right\|^3} + \\ &- \mu\frac{\boldsymbol{r} - (1-\mu)\hat{\boldsymbol{x}}}{\left\|\boldsymbol{r}_{p_3p_4}\right\|^3} + \boldsymbol{a}_{\odot}\end{aligned} \tag{11}$$

B. Controlled Dynamics

In order to actively control its approach trajectory to the binary asteroid system, the spacecraft makes use of a low-thrust engine with maximum thrust $T_{\max}$ and effective exhaust velocity c. So, the control thrust at any time can be expressed in frame N as

$$\boldsymbol{T} = T_x\hat{\boldsymbol{x}} + T_y\hat{\boldsymbol{y}} + T_z\hat{\boldsymbol{z}} \tag{12}$$

The following condition on the maximum thrust must hold along the whole spacecraft trajectory

$$\|\boldsymbol{T}\| \leqslant T_{\max} \tag{13}$$

Eventually, the controlled dynamics of the spacecraft is governed by the equations of motion

$$\dot{\boldsymbol{r}} = \boldsymbol{v} \tag{14}$$

$$\begin{aligned}\dot{\boldsymbol{v}} = &-2\,\hat{\boldsymbol{z}} \times \boldsymbol{v} - \hat{\boldsymbol{z}} \times (\hat{\boldsymbol{z}} \times \boldsymbol{r}) + (\sigma - 1)\left[(1-\mu)\frac{-\boldsymbol{r}_{p_1} - \mu\hat{\boldsymbol{x}}}{\left\|\boldsymbol{r}_{p_1p_2}\right\|^3} + \right.\\ &\left. + \mu\frac{-\boldsymbol{r}_{p_1} + (1-\mu)\hat{\boldsymbol{x}}}{\left\|\boldsymbol{r}_{p_1p_3}\right\|^3} - \frac{-\boldsymbol{r}_{p_1} + \boldsymbol{r}}{\left\|\boldsymbol{r}_{p_1p_4}\right\|^3}\right] + (1-\mu)\frac{\boldsymbol{r} + \mu\hat{\boldsymbol{x}}}{\left\|\boldsymbol{r}_{p_2p_4}\right\|^3} + \\ &- \mu\frac{\boldsymbol{r} - (1-\mu)\hat{\boldsymbol{x}}}{\left\|\boldsymbol{r}_{p_3p_4}\right\|^3} + \boldsymbol{a}_{\odot} + \frac{\boldsymbol{T}}{m}\end{aligned} \tag{15}$$

$$\dot{m} = -\frac{\|\boldsymbol{T}\|}{c} \tag{16}$$

$$\dot{t} = 1 \tag{17}$$

By defining the spacecraft state as $\boldsymbol{x} = [\boldsymbol{r}^\top \ \boldsymbol{v}^\top \ m \ t]^\top$, Eqs. (14)-(17) can be rewritten in compact form as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{T}) \tag{18}$$

Note that the independent variable time t, with derivative $\dot{t} = 1$, has been included in the state vector to recast the problem as an autonomous one. This transformation is necessary for the problem to be formulated as a Markov decision process, as will be described in the next section.

C. Boundary Conditions

1. Initial Conditions

The terminal guidance of the spacecraft is supposed to start a few hours before the collision occurs. Let's denote with t_f the final time of this mission phase. We want to determine a nominal spacecraft state at the beginning of the terminal guidance, that is, at time $t_0 = 0$, starting from the target value of some quantities at impact time provided by mission design. These quantities are the spacecraft's impact velocity v_I, the impact in-plane angle φ_I, the impact out-of-plane angle ψ_I and the impact solar phase angle φ_S, as well as, of course, the impact date t_I. They play a central role on the outcome of the mission. Indeed, the impact velocity and impact angles determine the amount of momentum transferred to the target asteroid, and, with it, the deflection imparted on its orbit; on the other hand, the solar phase angle affects the lighting conditions of the impact site and the ability of the spacecraft to autonomously navigate to it by just using visual images.

Specifically, the impact in-plane and out-of-plane angles are defined as the angles between the spacecraft's impact velocity $\hat{\boldsymbol{v}}_I$ and the secondary asteroid's velocity $\hat{\boldsymbol{v}}_{p_3}$ measured into and out of the binary system's orbital plane, respectively

$$\varphi_I = \text{arctan2}(\hat{\boldsymbol{v}}_I \cdot \hat{\boldsymbol{d}}, \hat{\boldsymbol{v}}_I \cdot \hat{\boldsymbol{v}}_{p_3}) \tag{19}$$

$$\psi_I = \sin^{-1}\left(-\hat{\boldsymbol{v}}_I \cdot \hat{\boldsymbol{h}}_{p_3}\right) \tag{20}$$

where $\hat{\boldsymbol{h}}_{p_3} = \hat{\boldsymbol{d}} \times \hat{\boldsymbol{v}}_{p_3} = \hat{\boldsymbol{z}}_P = \hat{\boldsymbol{z}}$ is the angular momentum of the secondary asteroid. The solar phase angle, instead, denotes the angle, measured in the binary system's orbital plane, between the spacecraft's impact velocity and the asteroid-Sun direction at impact time

$$\varphi_S = \cos^{-1}\left(\hat{\boldsymbol{v}}_{I,\text{ip}} \cdot \hat{\boldsymbol{r}}_{p_1p_3,\text{ip}}\right) \tag{21}$$

with $\hat{\boldsymbol{v}}_{I,\text{ip}} = (\hat{\boldsymbol{v}}_I \cdot \hat{\boldsymbol{x}}_P)\,\hat{\boldsymbol{x}}_P + (\hat{\boldsymbol{v}}_I \cdot \hat{\boldsymbol{y}}_P)\,\hat{\boldsymbol{y}}_P$ and $\hat{\boldsymbol{r}}_{p_1p_3,\text{ip}} = (\hat{\boldsymbol{r}}_{p_1p_3} \cdot \hat{\boldsymbol{x}}_P)\,\hat{\boldsymbol{x}}_P + (\hat{\boldsymbol{r}}_{p_1p_3} \cdot \hat{\boldsymbol{y}}_P)\,\hat{\boldsymbol{y}}_P$ the in-plane (i.e., in the binary system's orbital plane) components of the impact velocity and asteroid-Sun direction.

A sufficiently accurate value for the initial position $\boldsymbol{r}_0$ and velocity $\boldsymbol{v}_0$ that the spacecraft should have to meet those impact conditions can be computed using a two-body dynamical model, obtained by neglecting the Sun's influence and supposing the total mass of the binary system as concentrated in the primary asteroid. First, let us compute the mean anomaly $M_{p_3,f}$ of the minor asteroid at impact time. It can be easily derived starting from the Sun's angular position

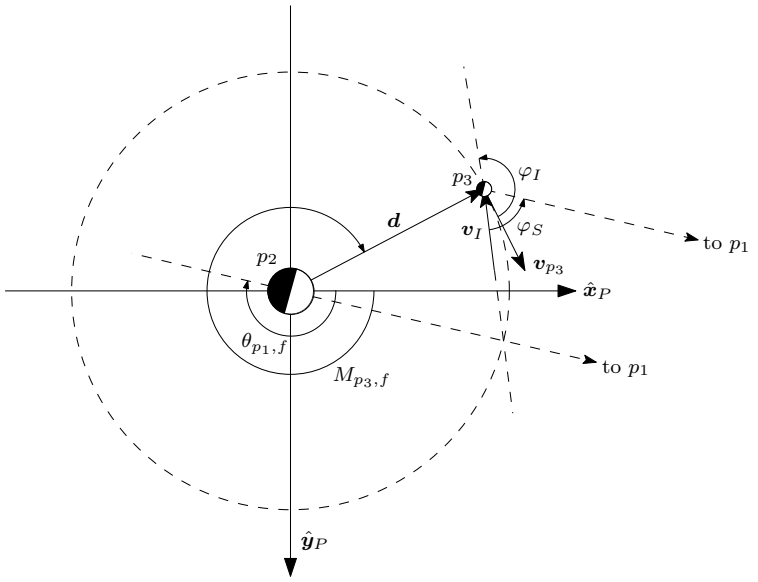


Fig. 2 View of the impact geometry on the binary system's orbital plane.

$\theta_{p_1,f}$ in frame P at the impact date t_I (computed from the binary system's ephemeris) and the value of the angles φ_I, ψ_I, and φ_S. Indeed, with reference to Fig. 2, by supposing that the Sun is sufficiently far to be approximated as a light source at infinity, we have that

$$M_{p_3,f} = -\frac{\pi}{2} + \varphi_S + \varphi_I + \theta_{p_1,f} \tag{22}$$

Figures 3a and 3b show a sketch of the hyperbolic approach of the spacecraft till the impact with the minor body in a two-body model. The spacecraft hyperbolic excess velocity is $v_\infty = \sqrt{v_I^2 - 2\frac{\bar{\mu}}{d}}$. Because of the small mass of the binary system, and the large incoming velocity of the spacecraft, its hyperbolic trajectory can be safely approximated as a straight line. So, a fairly-accurate value of the initial distance of the spacecraft from the system is given by $r_0 = v_\infty t_f$.

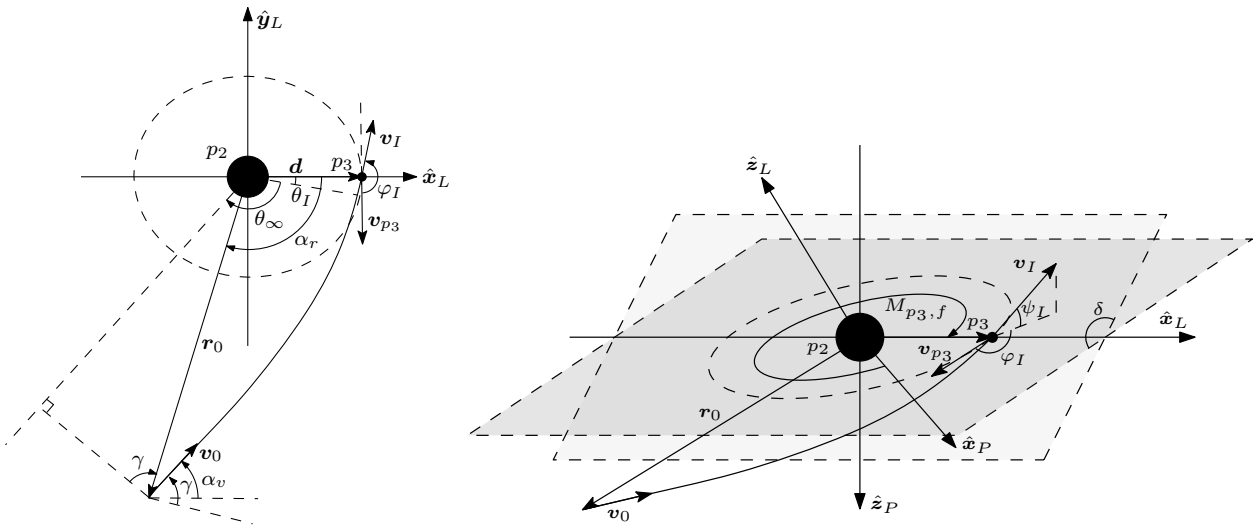
The conservation law of the specific angular momentum h_{hyp} of the spacecraft along the hyperbola can be written as

$$h_{\text{hyp}} = r_0 v_\infty \cos\gamma = -d\, v_I \cos\varphi_I \tag{23}$$

from which the initial flight path angle γ can be derived as

$$\gamma = \arccos\left(-\frac{d\, v_I \cos\varphi_I}{r_0 v_\infty}\right) \tag{24}$$

The parameters describing the hyperbola, that is, the semi-latus rectum p_{hyp}, the eccentricity e_{hyp}, the argument of



(a) In-plane view. **(b) Out-of-plane view.**

Fig. 3 The hyperbolic approach of the spacecraft.

periapsis θ_I, and the true anomaly at infinite distance θ_∞, can be derived from the conics equations:

$$p_{\text{hyp}} = \frac{h_{\text{hyp}}^2}{\bar{\mu}} = \frac{d^2 v_I^2 \cos {\varphi_I}^2}{\bar{\mu}} \tag{25}$$

$$e_{\text{hyp}} = \sqrt{\frac{p_{\text{hyp}} v_\infty^2}{\bar{\mu}} + 1} \tag{26}$$

$$\theta_I = \text{sign}(\pi - \varphi_I) \arccos\left(\frac{\frac{p_{\text{hyp}}}{d} - 1}{e_{\text{hyp}}}\right) \tag{27}$$

$$\theta_\infty = \arccos\left(-\frac{1}{e_{\text{hyp}}}\right) \tag{28}$$

Let us define a local frame $L = (p_2; \hat{\boldsymbol{x}}_L, \hat{\boldsymbol{y}}_L, \hat{\boldsymbol{z}}_L)$, with $\hat{\boldsymbol{x}}_L$ pointing from the primary asteroid to the secondary asteroid's position at impact time, $\hat{\boldsymbol{z}}_L$ directed as the angular momentum of the spacecraft, and $\hat{\boldsymbol{y}}_L$ defined to create a right-handed system (see Fig. 3a).

The angles between $\hat{\boldsymbol{x}}_L$ and $\boldsymbol{r}_0$ and $\hat{\boldsymbol{x}}_L$ and $\boldsymbol{v}_0$, measured in the plane of the spacecraft trajectory, are

$$\alpha_r = -\theta_I - \theta_\infty - \gamma + \frac{\pi}{2} \tag{29}$$

$$\alpha_v = \alpha_r + \gamma + \frac{\pi}{2} \tag{30}$$

So, the components of the initial spacecraft position and velocity in L are:

$$[\boldsymbol{r}_0]_L = r_0 \cos\alpha_r \, \hat{\boldsymbol{x}}_L + r_0 \sin\alpha_r \, \hat{\boldsymbol{y}}_L \tag{31}$$

$$[\boldsymbol{v}_0]_L = v_\infty \cos\alpha_v \, \hat{\boldsymbol{x}}_L + v_\infty \sin\alpha_v \, \hat{\boldsymbol{y}}_L \tag{32}$$

The angle δ between the spacecraft trajectory plane and the binary system's orbital plane is

$$\delta = \text{atan2}\left(\sin\psi_I, \frac{\cos\psi_I}{\cos\varphi_I}\right) \tag{33}$$

So, eventually, the components of $\boldsymbol{r}_0$ and $\boldsymbol{v}_0$ in the perifocal frame P can be obtained through a rotation around axis $\hat{\boldsymbol{x}}_L$ of an angle δ, and a rotation around axis $\hat{\boldsymbol{z}}_P$ of an angle $-M_{p_3,f}$

$$[\boldsymbol{r}_0]_P = \boldsymbol{C}_3(-M_{p_3,f})^\top \boldsymbol{C}_1(\delta)^\top [\boldsymbol{r}_0]_L \tag{34}$$

$$[\boldsymbol{v}_0]_P = \boldsymbol{C}_3(-M_{p_3,f})^\top \boldsymbol{C}_1(\delta)^\top [\boldsymbol{v}_0]_L \tag{35}$$

being

$$\boldsymbol{C}_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{36}$$

$$\boldsymbol{C}_3(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{37}$$

the rotation matrices referred to a counter-clockwise rotation of θ around the first and third axis of a right-handed triad, respectively.

In order to account for an imperfect knowledge of the binary system's characteristics, a random error has been added to the nominal value $\overline{M}_{p_3,0}$ of the initial mean anomaly of the secondary asteroid, simply computed by propagating its position at impact backward for a time t_f. The actual value of the mean anomaly is sampled from a uniform distribution centered about $\overline{M}_{p_3,0}$ at the beginning of each simulation:

$$M_{p_3,0} \sim \mathcal{U}\left(\overline{M}_{p_3,0} - \delta M, \overline{M}_{p_3,0} + \delta M\right) \tag{38}$$

where δM is the maximum angular error.

2. Terminal Conditions

Concerning the final conditions, any simulation is terminated either at time t_f or when the spacecraft reaches the minimum distance from the secondary asteroid along its trajectory. This latter condition occurs when the spacecraft incoming velocity has a null component along the line connecting the minor asteroid to the spacecraft. The condition can be expressed mathematically as

$$\chi(t) = \boldsymbol{v}(t) \cdot (\boldsymbol{r}(t) - \boldsymbol{r}_{p_3}(t)) = 0 \tag{39}$$

III. Problem Statement

In this section, the image-based terminal guidance of the spacecraft is mathematically posed as a partially observable Markov decision process. The optical sensor model used to generate the camera images is also described.

A. Problem Formulation as a Markov Decision Process

The terminal guidance problem can be formulated as a discrete-time Markov decision process (MDP). The system evolution in an MDP is determined by a finite number of interaction events or time steps $h = 0, 1, \ldots, H$, spaced between the initial time $t_0 = 0$ and the final time t_f:

$$t_0 < t_1 < \ldots < t_H \leqslant t_f \tag{40}$$

At each time step h a decision maker (referred to as the agent) chooses a control action $\boldsymbol{u}_h$ among the admissible ones, on the basis of its knowledge of the current system state $\boldsymbol{x}_h$. As a consequence of this action, the environment transitions to a new state $\boldsymbol{x}_{h+1}$ and returns a scalar reward $R_h = R(\boldsymbol{x}_h, \boldsymbol{x}_{h+1})$, which can be intended as a measure of the goodness of the last decision maker's choice. MDPs satisfy the Markov property, that is, at any time the system state depends only on the previous state and the agent's control.

In this application, the control $\boldsymbol{u}_h$ returned by the agent at step h is the output of a closed-loop control policy π, taking as input an observation $\boldsymbol{y}_h$ of the current state $\boldsymbol{x}_h$

$$\boldsymbol{u}_h = \pi(\boldsymbol{y}_h) \tag{41}$$

$$\boldsymbol{y}_h = \boldsymbol{\eta}(\boldsymbol{x}_h) \tag{42}$$

To give the spacecraft an additional degree of freedom, and let it decide on its own when and for how long to thrust, the

control $\boldsymbol{u}_h \in [-1,1]^5$ defines both the thrust $\boldsymbol{T}_h$ and the thrusting time Δt_h:

$$\boldsymbol{T}_h = \boldsymbol{\Gamma}(\boldsymbol{u}_h) = \frac{T_{\max}}{\left\| [u_h^{(2)} \; u_h^{(3)} \; u_h^{(4)}] \right\|} \frac{u_h^{(1)} + 1}{2} \left[u_h^{(2)} \, \hat{\boldsymbol{l}} + u_h^{(3)} \, \hat{\boldsymbol{v}} + u_h^{(4)} \, \hat{\boldsymbol{n}} \right] \tag{43}$$

$$\Delta t_h = \frac{u_h^{(5)} + 1}{2} (t_f - t_h) \tag{44}$$

where $V = (p_4; \hat{\boldsymbol{l}}, \hat{\boldsymbol{v}}, \hat{\boldsymbol{n}})$ is a reference frame attached to the velocity of the spacecraft

$$\hat{\boldsymbol{l}} = \frac{\hat{\boldsymbol{v}}_h}{v_{h,\text{ip}}} \times \hat{\boldsymbol{z}} \tag{45}$$

$$\hat{\boldsymbol{v}} = \hat{\boldsymbol{v}}_h \tag{46}$$

$$\hat{\boldsymbol{n}} = \hat{\boldsymbol{l}} \times \hat{\boldsymbol{v}} \tag{47}$$

This definition of the control has been selected as it inherently meets the constraint on the maximum value of the thrust modulus (Eq. (13)). Furthermore, RL performs better when the actions are sampled from intervals centered about zero. To avoid having a number of time-steps H which is either too low or too high, the time-length of each step Δt_h is further limited between 1 s and 1 h. The minimum step-size has been selected in accordance to the actual update frequency of SMARTNav [15]. A time horizon $H_{\max}$ has also been set.

To ensure that the terminal condition in Eq. (39) is not missed, the time t_{h+1} at the next step is computed as

$$t_{h+1} = \tau(t_h, \boldsymbol{u}_h) = \begin{cases} t_r \; : \; \chi(t_r) = 0 & \chi(t_h) > 0 \wedge \chi(t_h + \Delta t_h) < 0 \\ t_h + \Delta t_h & \text{otherwise} \end{cases} \tag{48}$$

Thus, Eq. (39) is checked $\forall t \in [t_h, t_{h+1}]$ by looking for a change of sign of function χ with a root-finding method as the secant method. The corresponding root t_r, or, if none is found, time $t_h + \Delta t_h$, is returned as next time.

The new state $\boldsymbol{x}_{h+1}$ is obtained through the numerical integration of Eqs. (18) starting from the previous state $\boldsymbol{x}_h$, and by assuming a constant thrust $\boldsymbol{T}_h$ during the whole time-step. Hence, the state transition function is

$$\boldsymbol{x}_{h+1} = \boldsymbol{\phi}(\boldsymbol{x}_h, \boldsymbol{u}_h) = \boldsymbol{x}_h + \int_{t_h}^{\tau(t_h, \boldsymbol{u}_h)} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\Gamma}(\boldsymbol{u}_h)) \, \mathrm{d}t \tag{49}$$

Note that the transition function in Eq. (49) satisfies the Markov property, thus justifying why the time t has been considered as an additional state variable. The episode is terminated if the stopping (or done) condition is met, defined as

$$d(\boldsymbol{x}_h) = \big((t_h = t_f) \vee (\chi(t_h) = 0) \big) = 1 \tag{50}$$

The observation $\boldsymbol{y}_h$ received by the agent at step h is made up of the current time t_h and a gray-scale image $\boldsymbol{I}_h$ taken by the onboard camera, which corresponds to a matrix of dimension $i_h \times i_w$, being i_h and i_w the number of pixels along height and width, respectively. Each entry of $\boldsymbol{I}_h$ is an integer number in $[0, 255]$, specifying the brightness of the corresponding pixel. For simplicity, it is assumed that, as in SMARTNav, an onboard system is able to recognize the secondary asteroid in the image frame and align the camera bore-sight with it at each time-step; so the camera line-of-sight is $\hat{\boldsymbol{l}} = -\hat{\boldsymbol{r}}_{p_3 p_4}$. Moreover, it is assumed that the satellite is capable of producing thrust in each direction independently from the onboard camera pointing. Since the agent does not have direct access to the full spacecraft state $\boldsymbol{x}$ to decide the next control, but only to an observation $\boldsymbol{y}$ dependent on it, the MDP is defined as partially observable (POMDP).

The initial state $\boldsymbol{x}_0$ of the spacecraft is determined with the procedure described in Sec. II.C. Actually, the interplanetary trajectory design provides a nominal range of variation for the impact quantities, whose precise value will depend on the actual departure date from Earth within the launch window. Since the terminal guidance algorithm should be designed to correctly face every possible impact condition within the nominal range, the actual value of the generic quantity $q \in \{t_I, v_I, \varphi_I, \psi_I, \varphi_S\}$ at impact is sampled at the beginning of each episode according to a uniform distribution

$$q = q_l + p(q_u - q_l) \tag{51}$$

where $p \in \mathcal{U}(0, 1)$ is sampled once for all the quantities, thus simulating that a launch date has been fixed. q_l, q_u denote the lower and upper bound for quantity q provided by mission design. The combined effect of the random impact condition and the uncertainty δM on the initial mean anomaly of the target asteroid can be modeled as a probability distribution $\mathcal{X}_0$ for the initial state $\boldsymbol{x}_0$.

The goal of the mission is to hit the secondary asteroid within the maximum time t_f, in spite of the dynamical perturbations not modeled during mission design (i.e., during derivation of the initial conditions) and the uncertainty on the minor asteroid position. To this aim, a delayed reward definition has been used in this study:

$$R_h = \begin{cases} -\left\|\boldsymbol{r}_{p_3 p_4, h+1}\right\| & \text{if } d(\boldsymbol{x}_{h+1}) \\ 0 & \text{otherwise} \end{cases} \tag{52}$$

The goal of the agent is to find the control policy $\pi^\star$ that maximizes the expected sum of rewards (or return) $G(\tau)$ collected along a trajectory $\tau = \{(\boldsymbol{x}_0, \boldsymbol{u}_0), \ldots, (\boldsymbol{x}_{H-1}, \boldsymbol{u}_{H-1}), \boldsymbol{x}_H\}$

$$J(\pi) = \mathop{\mathrm{E}}_{\tau \sim \pi}\left[\sum_{h=0}^{H-1} R_h\right] = \mathop{\mathrm{E}}_{\tau \sim \pi}\left[G(\tau)\right] \tag{53}$$

Overall, the terminal guidance problem can be formulated as a POMDP as follows:

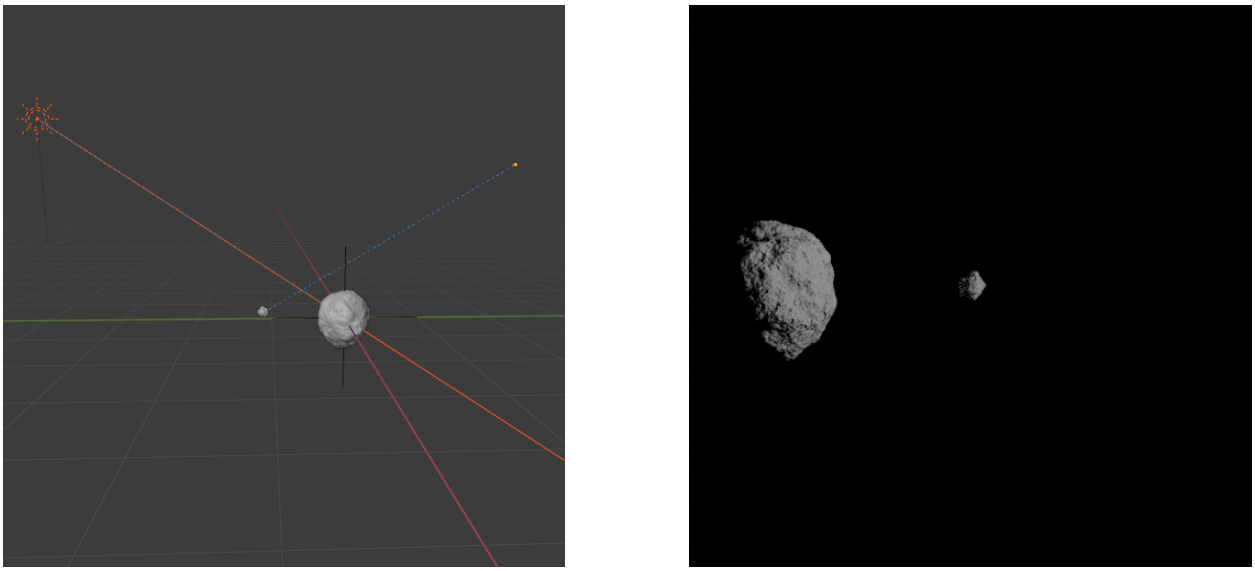
$$\mathcal{M}: \begin{cases} \max\limits_{\pi} J(\pi) & \\ \text{s.t.: } \boldsymbol{x}_{h+1} = \boldsymbol{\phi}(\boldsymbol{x}_h, \boldsymbol{u}_h), & h = 0, \dots, H-1 \\ \quad \boldsymbol{u}_h = \pi(\boldsymbol{y}_h), & h = 0, \dots, H-1 \\ \quad \boldsymbol{y}_h = \{t_h, \boldsymbol{I}_h\} = \boldsymbol{\eta}(\boldsymbol{x}_h), & h = 0, \dots, H-1 \\ \quad \boldsymbol{x}_0 \sim \mathcal{X}_0 & \\ \quad d(\boldsymbol{x}_H) = 0 & \end{cases} \tag{54}$$

B. Optical Sensor Model

The simulation environment is created using a custom class derived from the OpenAI Gym formalization [47]. The simulated images taken from the onboard camera are generated using VisualEnv [48], a tool for real-time rendering based on the open-source software Blender [44]. With this tool, it is possible to create realistic visual environments leveraging physically based materials, light sources, and cameras. Moreover, the tool seamlessly integrates with python through an API which allows it to run within the environment itself. The scene is created using the built-in modeling tools available in Blender. The two asteroids of the binary system are created starting from spheres, which are then modified using a procedurally generated random texture to create surface roughness and displacements. A second random texture is used to create craters. Sunlight is simulated using a light source at infinite distance generating parallel rays. The camera field-of-view is instead supposed to be always centered on the target asteroid, for simplicity. The position of all the objects in the scene (i.e. the spacecraft, the two asteroids, and the Sun) is updated at each step and the scene seen by the camera is rendered and returned as a matrix $\boldsymbol{I}_h$. So, $\boldsymbol{I}_h$ can be evaluated just starting from the spacecraft position $\boldsymbol{r}_h$ and the celestial body position (dependent only on the current time t_h). Function $\boldsymbol{\eta}$ is thus a black-box function that models the rendering engine of Blender. Figure 4 shows a 3D view of the scene configuration inside Blender software exactly two minutes before the impact, along one of the obtained solutions (Fig. 4a), and the corresponding rendered image as seen by the camera onboard (Fig. 4b).

IV. Reinforcement Learning

This section describes the optimization framework used in this study, based on reinforcement learning. Specifically, after the architecture chosen for the control policy network has been introduced, the optimization algorithm is explained in detail. Then, the role of the recurrent layer in the policy network and the deep meta-reinforcement learning framework adopted in this study are presented.



(a) 3D view of the scene. **(b) Rendered image taken from the camera.**

Fig. 4 System configuration in Blender two minutes before the impact along one sample solution.

A. Policy Network

Solving a control problem via deep reinforcement learning means approximating the exact control policy π by a deep neural network (DNN) with parameters θ, π_θ, which is trained by trial-and-error to maximize the expected trajectory return $J(\pi) = J(\theta)$. So, the objective becomes determining the optimal set of network parameters $\theta^\star$ (i.e., weights of the neuron-to-neuron connections and neurons' biases).

A diagonal multi-variate Gaussian policy is used in this study to achieve a good balance between exploration and exploitation during training. Hence, at each time-step t_h, the network π_θ receives the current observation $\boldsymbol{y}_h$ as input, and returns the mean value of the control $\boldsymbol{\mu}_h$ and the corresponding standard deviation $\boldsymbol{\sigma}_h$. Being the policy stochastic, in current RL notation the symbol $\pi_\theta(\cdot|\boldsymbol{y}_h)$ is typically used to indicate the probability that a given control $\boldsymbol{u}_h$ is returned, given the observation $\boldsymbol{y}$. This notation has been adopted also in this paper for the sake of consistency with RL terminology, although, practically, the policy π_θ returns the parameters of the probability distribution and not directly the probability value.

To allow a wide exploration of the solution space, during training the actual control is sampled according to the Gaussian distribution

$$\boldsymbol{u}_h \sim \pi_\theta(\cdot|\boldsymbol{y}_h) = \mathcal{N}(\boldsymbol{\mu}_h, \Sigma_h) \tag{55}$$

with $\Sigma_h = \text{diag}(\boldsymbol{\sigma}_h\, \boldsymbol{\sigma}_h^\top)$ the covariance matrix. To ensure that the control $\boldsymbol{u}$ always lies in the definition interval, the probability that each of its components falls outside of interval $[-1, 1]$ (that is, the tails of the Gaussian) is clipped to zero. During the final policy deployment or evaluation, instead, the exploration is turned off and the mean control is

returned for a given observation: $\boldsymbol{u}_h = \boldsymbol{\mu}_h$.

The network architecture used in this work is schematized in Fig. 5. The network is composed of a sequence of layers, each one carrying out a specific task. The image $\boldsymbol{I}_h$ is first fed to a sequence of four convolutional layers that extract spatial information by simultaneously increasing the depth of the image and decreasing the height and width dimensions. The kernel size and number of layers have been selected to transform the 256×256 input matrix into an output vector of length 128. The vector output of the convolutional block is then concatenated with the time t_h and fed to a fully connected block (MLP) with two hidden layers, to increase the non-linearity in the network. The width of the two layers is intermediate between the number of inputs and the number of outputs of the MLP block. Indeed, a similar architecture, with progressively thinner layers going from inputs to outputs, has reconstructed the optimal solution of several space trajectory optimization problems with satisfying accuracy [23, 31, 37]. Lastly, the previous control $\boldsymbol{u}_{h-1}$ is appended to the output of the MLP block and fed to a long short-term memory (LSTM) block, that is a recurrent layer capable of understanding the temporal relationship between the observations making up the input sequences. The LSTM has been preferred to other RNN alternatives because of its increased capability of capturing and storing information about long-term temporal dependencies in the input data. The role of the LSTM layer in the whole network will be further discussed in Sections IV.C and IV.D.

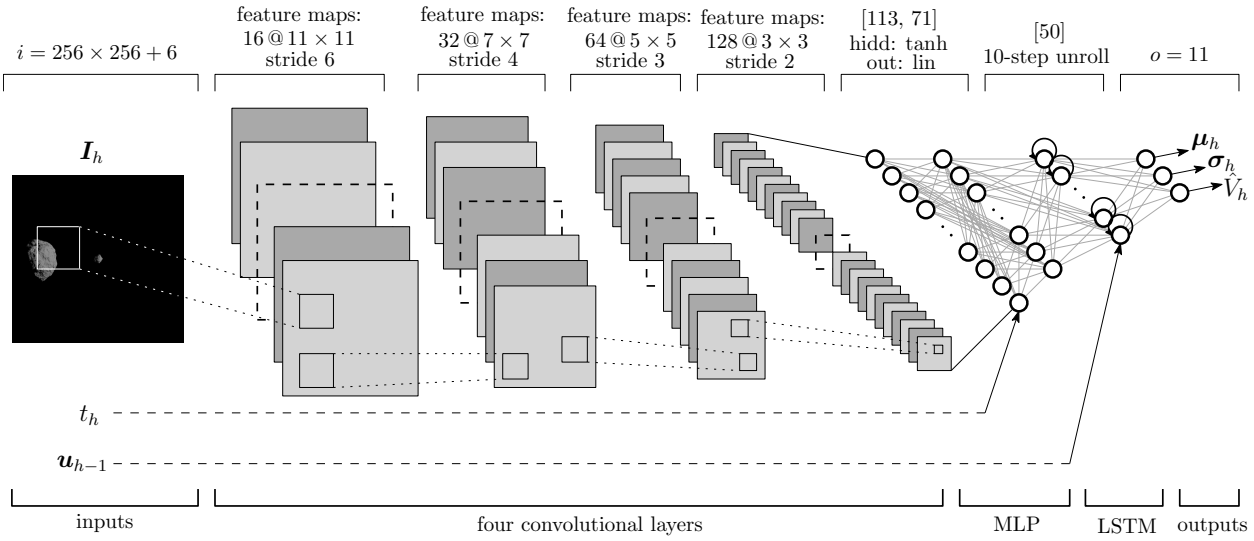


Fig. 5 Neural network architecture.

B. Proximal Policy Optimization

The optimization algorithm used in this work for the network training is proximal policy optimization (PPO) [45], which is a family of model-free policy-gradient reinforcement learning methods developed in 2017 by Schulman et al. at OpenAI. PPO exploits a first-order unconstrained optimization, thus representing an easier, but equally-performing, alternative to trust region policy optimization (TRPO) [49]. While both the methods try to limit the distance between

the new and the previous policy at each update to avoid a performance collapse, the basic difference between the two methods resides in the mathematical expression of the objective function.

Indeed, TRPO tackles the problem by maximizing a surrogate objective function subject to a KL-divergence constraint D_{KL} [50] to limit the distance δ between policies at successive iterations. Mathematically speaking, at iteration k of the training procedure, TRPO computes the new parameters $\theta_{(k+1)}$ by solving the following constrained optimization problem:

$$\begin{aligned} &\max_{\theta} \underset{\substack{\tau\sim\pi_{\theta_{(k)}} \\ h=0,\dots,H-1}}{\mathrm{E}} \left[\tilde{r}_h(\theta,\theta_{(k)}) A^{\pi_{\theta_{(k)}}}(\boldsymbol{y}_h,\boldsymbol{u}_h)\right] \\ &\text{s.t.:} \underset{\substack{\tau\sim\pi_{\theta_{(k)}} \\ h=0,\dots,H-1}}{\mathrm{E}} \left[D_{\mathrm{KL}}\left(\pi_\theta(\cdot|\boldsymbol{y}_h)\|\pi_{\theta_{(k)}}(\cdot|\boldsymbol{y}_h)\right)\right] \leqslant \delta \end{aligned} \tag{56}$$

where $\tilde{r}_h$ is the probability ratio between the updated and the previous policy at time-step h

$$\tilde{r}_h(\theta,\theta_{(k)}) = \frac{\pi_\theta(\boldsymbol{u}_h|\boldsymbol{y}_h)}{\pi_{\theta_{(k)}}(\boldsymbol{u}_h|\boldsymbol{y}_h)} \tag{57}$$

and $A^{\pi_\theta}(\boldsymbol{y}_h,\boldsymbol{u}_h)$ corresponds to the advantage function at time-step h, and it is a measure of the return improvement obtained by taking the specific action $\boldsymbol{u}_h$ after receiving observation $\boldsymbol{y}_h$, instead of randomly selecting the action according to $\pi(\cdot|\boldsymbol{y}_h)$.

However, the advantage function cannot be computed exactly and its value has to be approximated. The generalized advantage estimator (GAE) $\hat{A}_h$ [51] is used to this aim. Let $\hat{V}_h = \hat{V}(\boldsymbol{y}_h)$ be an estimation of the value function at time-step h, which represents the expected return obtained until the end of the episode by receiving the observation $\boldsymbol{y}_h$ and then acting according to policy π_θ. Then, the GAE is defined as

$$\hat{A}_h = \sum_{h'=h}^{H-1} \lambda^{h'-h} \delta_{h'}^{\hat{V}} \tag{58}$$

where

$$\delta_h^{\hat{V}} = R_h + \hat{V}_{h+1} - \hat{V}_h \tag{59}$$

λ is defined GAE factor. In particular, the value function estimate $\hat{V}_h$ is either the output of a second and independent DNN, named critic, which usually receives, just like the policy network (also known as actor), the current observation $\boldsymbol{y}_h$ as input, or is an additional output of the policy network itself, which plays both the actor and critic role (as done in this work, see Fig. 5). Hence, the set of parameters θ includes the weights and biases of both networks (actor and critic), which must be computed and updated accordingly.

As opposed to TRPO, PPO avoids using the KL-divergence constraint by introducing a so-called clipped policy

objective function J^{clip}, that is expressed as

$$J^{\text{clip}}(\theta) = \mathop{\mathrm{E}}_{\substack{\tau\sim\pi_{\theta_{(k)}} \\ h=0,\dots,H-1}} \left[\min\left\{\tilde{r}_h \hat{A}_h, \text{clip}\left(\tilde{r}_h, 1-\epsilon, 1+\epsilon\right)\hat{A}_h\right\}\right] \tag{60}$$

As it can be observed, J^{clip} tries to force the policy π_θ to stay within a small range, named clip range and defined by the tolerance $\epsilon \in [0, 1]$, around its previous value $\pi_{\theta_{(k)}}$. The complete surrogate objective function used in PPO is defined as

$$J^{\text{ppo}}(\theta) = J^{\text{clip}}(\theta) - c_v L^{\text{v}}(\theta) \tag{61}$$

where c_v is a hyperparameter known as value function coefficient and L^{v} is a mean-squared-error between the current value function estimation $\hat{V}_h$ and the obtained reward-to-go, required to correctly update also the critic network's parameters

$$L^{\text{v}}(\theta) = \mathop{\mathrm{E}}_{\substack{\tau\sim\pi_{\theta_{(k)}} \\ h=0,\dots,H-1}} \left[\left(\hat{V}_h - \sum_{h'=h}^{H} R_{h'}\right)^2\right] \tag{62}$$

A graphical overview of the training process is given in Fig. 6. The network training via PPO consists of alternating a rollout and an update phase at every iteration. In the rollout phase of iteration k, a set $\mathcal{D}_{(k)} = \{\tau_i\}$ of trajectories, a set $\mathcal{R}_{(k)} = \{\{R_h\}_i\}$ of corresponding rewards and a set $\mathcal{V}_{(k)} = \{\{\hat{V}_h\}_i\}$ of value functions are collected by n_w worker agents that run in parallel the most up-to-date policy $\pi_{\theta_{(k)}}$ in as many independent realizations of the environment, for n_s training steps each. In the update phase, the trajectories in $\mathcal{D}_{(k)}$ are randomly shuffled and divided into mini-batches, each with n_b steps. Then, the network's parameters θ are updated by performing, sequentially, n_{sga} stochastic gradient ascent (SGA) iterations on each mini-batch, with a learning rate $\alpha_{(k)}$

$$\theta_{(k+1)} = \theta_{(k)} + \alpha_{(k)} \left.\nabla_\theta \hat{J}^{\text{ppo}}(\theta)\right|_{\theta_{(k)}} \tag{63}$$

where $\hat{J}^{\text{ppo}}$ indicates an empirical approximation of the objective function J^{ppo} obtained by substituting the expected values in Eq. (60) and (62) with the average over the finite set of trajectories in $\mathcal{D}_{(k)}$ and a time average over the H time steps in a single trajectory

$$\hat{J}^{\text{ppo}}(\theta) = \hat{J}^{\text{clip}}(\theta) - c_v \hat{L}^{\text{v}}(\theta) \tag{64}$$

$$\hat{J}^{\text{clip}}(\theta) = \frac{1}{|\mathcal{D}_{(k)}|H} \sum_{\tau\in\mathcal{D}_{(k)}} \sum_{h=0}^{H-1} \min\left\{\tilde{r}_h \hat{A}_h, \text{clip}\left(\tilde{r}_h, 1-\epsilon, 1+\epsilon\right)\hat{A}_h\right\} \tag{65}$$

$$\hat{L}^{\text{v}}(\theta) = \frac{1}{|\mathcal{D}_{(k)}|H} \sum_{\tau\in\mathcal{D}_{(k)}} \sum_{h=0}^{H-1} \left(\hat{V}_h - \sum_{h'=h}^{H} R_{h'}\right)^2 \tag{66}$$

The training process is stopped when a maximum number of iterations K is reached.

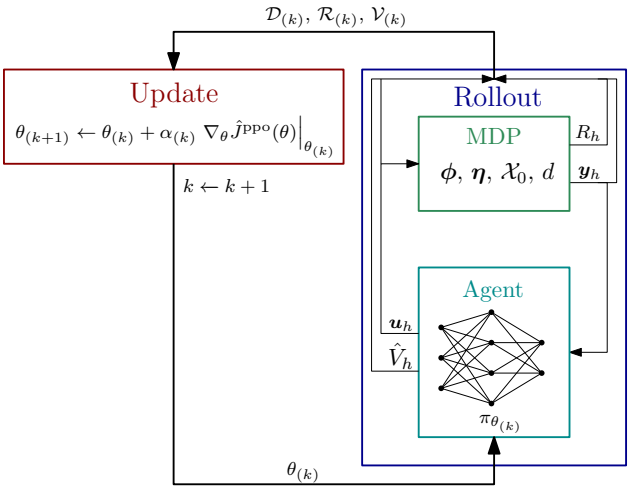


Fig. 6 Schematic of the training process by PPO.

C. Recurrent Neural Network

Image-based environments, as the one considered in this work, come with the downside of not providing the agent with all the information it would need to make thoughtful decisions about the next control action to choose. In fact, a single image as observation is usually not enough to decode the whole system state at a given step, as, for example, it does not provide any clue about the system's velocity and/or the motion of the other objects making up the scene. For this reason, policy-gradient RL usually struggles to cope with these partial-observable scenarios using just a standard fully-connected network as a control policy. A possible workaround, often used in Atari problems [52], consists in stacking a number of images (typically 3 or 4) in a 3D tensor before feeding them to the network. This solution gives the network some information about the evolution of the state of the environment over time.

A second approach relies on the use of a recurrent neural network (RNN) within the policy network, that is, a network with a feedback connection capable of keeping track of temporal dependencies in sequential input data. In this study, a particular RNN architecture called long-short term memory (Figure 7) has been used. An LSTM cell is composed of three internal gates (an input gate, a forget gate, and an output gate) that control the flow of data within the cell. The feedback connection of the LSTM unit has been unrolled in Fig. 7 to better clarify how it works. The three gates allows the network to store in an internal cell state $\boldsymbol{c}_h$ relevant information about the temporal evolution of the input observations $\boldsymbol{y}$ received so far, in order to form a belief (or an estimate) of the present system state $\boldsymbol{x}_h$, thus playing the role of the navigation system. When using an LSTM, the different time points in a single trajectory are provided to it one by one, and the value of the output at step h is computed by combining, through the net weights, the current input $\boldsymbol{y}_h$, the previous output $\boldsymbol{h}_{h-1}$ and the cell state $\boldsymbol{c}_h$. The current cell state $\boldsymbol{c}_h$, in turn, is computed by properly dropping and/or updating elements of the previous state $\boldsymbol{c}_{h-1}$ on the base of $\boldsymbol{y}_h$ and $\boldsymbol{h}_{h-1}$. At the beginning of each episode, all

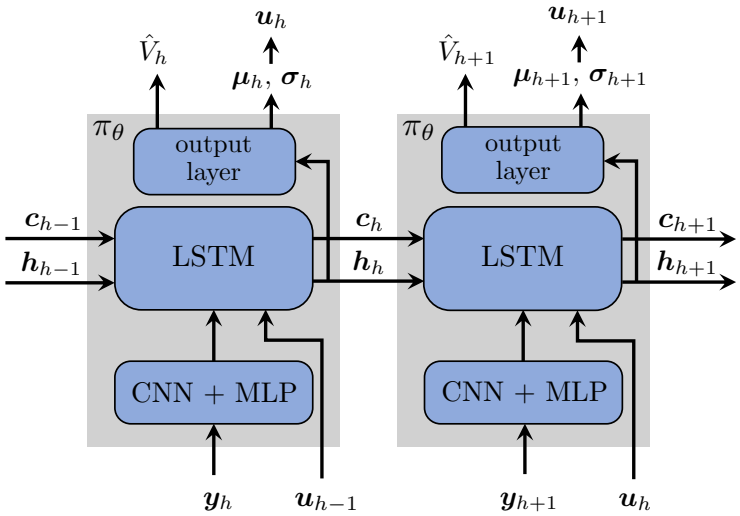


Fig. 7 Unrolling of the policy network with an LSTM unit.

the components of the network output $\boldsymbol{h}_0$ and state $\boldsymbol{c}_0$ are re-initialized to 0. More detailed information about LSTM units can be found in Ref. [53]. In this application, the sequences fed to the network have at most 10 consecutive time steps, as the little improvement in performance noted by using longer observation sequences did not justify the, much more consistent, increase in training time.

D. Meta-Reinforcement Learning

Using a recurrent neural network is one of the most common approaches to implement a meta-learning [54]. The term meta-learning, or "learning to learn", can be in general applied to any machine learning procedure, from reinforcement learning to standard supervised learning. The most common view of meta-learning is to learn a flexible learning algorithm that generalizes well across different tasks sampled from a distribution $p(\mathcal{T})$, where each task $\mathcal{T}$ is identified by a dataset $\mathcal{D}_\mathcal{T}$ and an objective function $J_\mathcal{T}$. Typically, the goal of meta-RL is to exploit the data coming from previously-seen tasks to better and/or quickly learn a new task from the same distribution. Focusing just on the RL version (meta-reinforcement learning or meta-RL), each task may represent a different realization of a single stochastic environment (as in the present application), or a slightly different environment sampled from a multi-environment distribution. The key idea in meta-RL is to train a neural network with parameters θ to represent a function $\phi_\mathcal{T} = f_\theta(\mathcal{T})$ of the specific task $\mathcal{T}$, so that $\pi(\boldsymbol{y}; \phi_\mathcal{T})$ is the optimal control policy for that task. The optimal network parameters are thus the parameters that maximize the expected return obtained in the different tasks

$$\max_\theta \mathop{\mathrm{E}}_{\mathcal{T}\sim p(\mathcal{T})} \left[J_\mathcal{T}(f_\theta(\mathcal{T})) \right] \tag{67}$$

A variety of approaches have been used to date to implement the meta-learning paradigm [55], including the use of

recurrent networks. When using a recurrent cell within the policy network, the model embeds the dataset $\mathcal{D}_{\mathcal{T}}$ (i.e., the evolution of the received observations) of the specific task into the internal network state $\boldsymbol{c}$, which is used for action predictions together with the last observation. So, when a recurrent network is used, the task-specific function $\phi_{\mathcal{T}}$ consists of the task-dependent internal network state $\boldsymbol{c}(\mathcal{T})$. The general network parameters θ, instead, are still learned by standard gradient ascent on Eq. (67) across trajectories from different tasks.

The adaptability of the recurrent model to different tasks can be further enhanced by providing the recurrent block also with the output $\boldsymbol{u}_{h-1}$ of the previous observation (and, when available, the reward R_{h-1}), in addition to the last observation $\boldsymbol{y}_h$ (or output of the preceding layer) [38]. In this way, the internal network state keeps track also of specific information of the task distribution, as the evolution of the controls and rewards, and the network output will be specifically tuned on the structure and statistics of the problem at hand. A previous study by the authors already demonstrated that an LSTM-based meta-RL approach is able to consistently improve the network performance when dealing with an environment with scattered initial conditions [41], as in the present application.

V. Numerical Results

This section presents the numerical results obtained on the mission scenario selected as a study case, that is, the terminal guidance of the DART spacecraft towards Dimorphos.

A. Mission Scenario

The terminal phase of the DART mission is assumed to start about 4 hours before the impact with Dimorphos. In fact, this is when, according to the current mission schedule [15], SMARTNav starts operating and guiding the spacecraft in autonomy. So, a maximum mission time equal to $t_f = 4\,\text{h}$ has been used in all simulations. Furthermore, as for mission specifications [15], the spacecraft's engines must be turned off a couple of minutes before the impact, to avoid creating blurry images of Dimorphos because of vibrations induced to the solar panels. So, for the last 120 s, the thrust $\boldsymbol{T}$ returned by the control policy has been set to zero.

The orbital parameters of the two asteroids of the 65803 Didymos system, together with their gravitational parameter μ and mean radius ρ, are listed in Table 1. The spacecraft and camera characteristics used in this work are instead reported in Table 2. Specifically, $i_h \times i_w$ defines the camera resolution in pixels, FOV indicates the camera field of view, while W is the camera sensor size, which determines the sensor area sensitive to light. It is worthwhile noticing that the actual resolution of DRACO's images processed by SMARTNav is 1024×1024 pixels [56]. In this study, the images are directly rendered in Blender at a lower resolution (256×256) both to speed up the network training process and to test the guidance algorithm performance in a more penalizing scenario.

The range of variation of the nominal values of the quantities at impact time, as described in Section II.C, is given in Table 3. A maximum angular deviation equal to $\delta M = 10\,\text{deg}$ has been used to express the uncertainty on the initial

orbital position of Dimorphos.

Table 1 65803 Didymos orbital parameters in ICRF⋆ and physical data [57].

Asteroid	a	e	i, deg	Ω, deg	ω, deg	M, deg	μ, km^3/s^2	ρ, m
p_2	1.644 AU	0.384	3.408	73.199	319.319	136.650	3.567×10^{-8}	390
p_3	1.190 km	0	160	149	-	-	3.693×10^{-10}	85

⋆ The parameters are referred to the epoch: 2021 Jul 01 at 00:00:00 UTC.

Table 2 Spacecraft and camera characteristics [16, 56, 58].

Parameter	Value
m_0, kg	560
$T_{\max}$, N	0.137
c, km/s	30.33
A, m^2	22
$i_h \times i_w$, px	256×256
FOV, deg	0.29
W, mm	13.27

Table 3 Ranges of the nominal quantities at impact [58, 59].

t_I, UTC	v_I, km/s	φ_I, deg	ψ_I, deg	φ_S, deg
[25 Sep 2022, 11pm, 1 Oct 2022, 11pm]	[6.12, 6.76]	[170, 180]	[-33.5, -6.9]	[58.3, 59.9]

B. Training Behavior

The results presented in this section have been obtained by using pyRLprob*, an in-house python library for training, evaluation, and postprocessing of OpenAI-Gym environments through the open-source library Ray [60], which includes both a multiple-CPU and GPU implementation of PPO. The value of the hyperparameters used for PPO is listed in Table 4. The learning rate $\alpha_{(k)}$ decreases with a linear law along training. These specific values have been selected after an extensive trial-and-error procedure carried out on a fully-observable version of the problem, that is, by using the full spacecraft state as input to the control policy, in order to reduce the overall computational burden. This preliminary analysis has been run on a computer with an 8-core Intel Core i7-9700K CPU @3.60 GHz.

The final network training by PPO, including the image rendering, has been realized on a cpu-only workstation with a 56-core Intel Xeon E5-2680 @2.40 GHz. The overall training process took about 20 h on this hardware. During training, a deterministic version of the policy (i.e., with the exploration switched off) is concurrently deployed into an evaluation environment for 600 steps to evaluate its performance. The best policy in terms of the average value of the trajectory return in the evaluation environment is the one returned by the training procedure as putative optimal policy.

*https://github.com/LorenzoFederici/pyrlprob

Table 4 PPO hyperparameters.

Hyperparameter	Symbol	Value
Time horizon	$H_{\max}$	100
Training iterations	K	250
Clip range	ϵ	0.05
Value function coefficient	c_v	0.5
Training workers	n_w	15
Steps per worker	n_s	200
SGA iterations per update	n_{sga}	30
Steps per mini-batch	n_b	600
Initial learning rate	$\alpha_{(0)}$	1×10^{-4}
Final learning rate	$\alpha_{(K)}$	1×10^{-6}

(a) Episode return.

(b) Final miss distance.

(c) Mean episode length.

Fig. 8 Performance trend of the RL policy in the training and evaluation environments.

Figure 8 presents the evolution during the net training of the average value (solid curve) and interquartile range (shaded region) of the episode return G (Fig. 8a) and terminal miss distance $d_{p_3p_4}$ (Fig. 8b), and the mean value of the episode length H (Fig. 8c) in both the training and evaluation environments. The miss distance $d_{p_3p_4}$ is defined as the

distance of the spacecraft from the surface of (a spherically-shaped) Dimorphos at the end of the mission

$$d_{p_3 p_4} = \max\left(0, \left\|\boldsymbol{r}_{p_3 p_4}\right\| - \rho_{p_3}\right) \tag{68}$$

Figure 8a shows that learning proceeded smoothly throughout the training session, with average performance always increasing monotonously until the very end of the optimization. The non-zero slope of the curve near the final iteration suggests that a further increase in performance might still have been possible. A similar trend is noticeable in Fig. 8b for the miss distance. The difference between the stochastic and deterministic policy behavior becomes significant in the last part of the training when the latter rapidly reaches values very close to zero.

An attentive reader can notice a peculiar shape in both the miss distance and episode return trends when the policy is in evaluation mode, characterized by a large initial improvement, an intermediate plateau, and a final, slower, improvement. Indeed, in the first part of the training, the network rapidly learns that is better to almost completely switch off the control thrust to get closer to the target, being the initial spacecraft conditions designed to let it hit Dimorphos with a completely ballistic flight in a rather similar dynamical model. At that point, the only way the network has to further reduce the average distance from the asteroid is to slowly add control points in specific parts of the trajectory to progressively eliminate the residual error due to the dynamical perturbations. This behavior is confirmed by the number of time steps H in each episode, which shows a slowly-increasing trend in the second half of the training (Fig. 8c). It can be also noted a consistent difference (of about 3 to 4 steps) in the mean number of steps per episode between training and evaluation. This difference is mainly an effect of the intrinsic randomness of the policy in the training rollouts. In fact, in the final part of the trajectory, where the control points are very close to each other, it is sufficient to have a slightly longer value of the thrusting time to prematurely meet one of the terminal conditions of the episode. In the evaluation phase, when a deterministic version of the policy is used, this effect is no longer present.

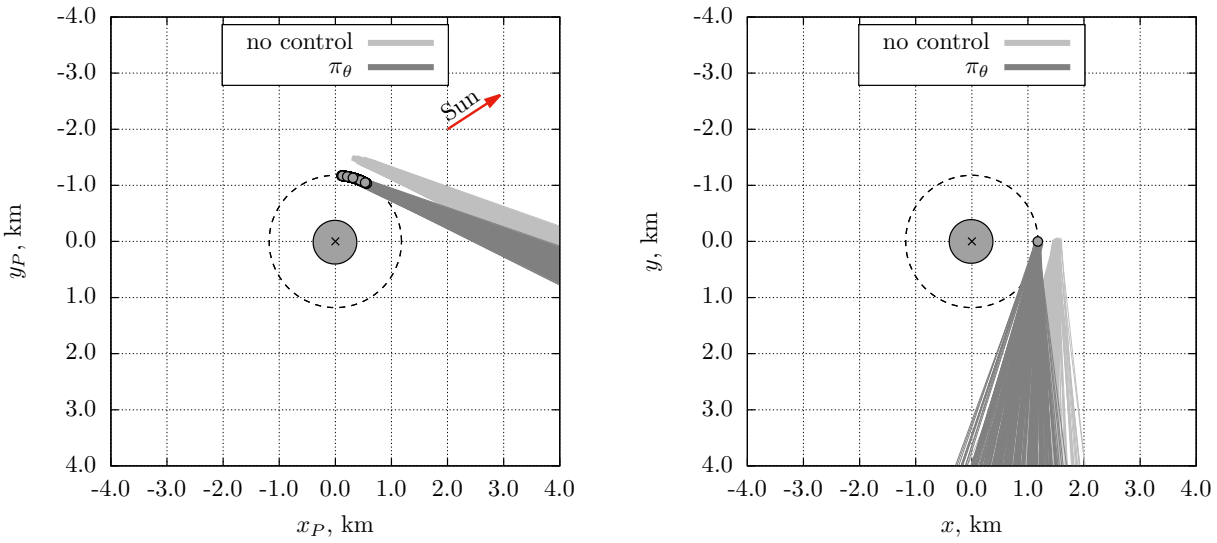
C. Monte Carlo Analysis

The effectiveness of the trained network as a control policy has been verified by means of a Monte Carlo analysis, involving the deployment of the optimal policy in 500 evaluation episodes. The results in terms of minimum, mean and maximum value of the miss distance and overall success rate (i.e., the fraction of trajectories that hit Dimorphos) are summarized in Table 5. For comparison, the results obtained in the same mission scenarios without any control thrust have been reported too. In this case, increasingly complex dynamical models have been considered to better highlight, in absence of any control action, the contribution on the final miss distance of each perturbative effect considered in this study. The terminal part of the spacecraft trajectories in the orbital plane of the asteroids (both in frame P and N) are reported in Fig. 9 for both the controlled and uncontrolled case in the full dynamical model. In these plots, the central gray circle represents Didymos, the black dashed circle Dimorphos' orbit and the black circle markers the position of

Dimorphos at impact time in the different test scenarios. All objects are plotted to scale.

Table 5 Monte Carlo campaigns w/ and w/o the control thrust with different dynamics.

Control policy	Dynamical model	$d_{p_3p_4}$, m			SR, %
		min	mean	max	
none	2BP	0	0.07	13.5	99.0
	4BP	163.3	238.8	316.6	0
	4BP + SRP	172.9	247.3	325.8	0
	4BP + SRP + δM	173.6	259.0	345.4	0
π_θ	4BP + SRP + δM	0	0.24	34.4	98.4



(a) Perifocal frame P. **(b) RTN frame N.**

Fig. 9 Spacecraft trajectory on the binary system's orbital plane.

It is clear that a pure ballistic flight would drive the spacecraft toward Dimorphos in any test case in a two-body dynamical model (2BP), unless for minor deviations due to the patched-conics approximation. Indeed, this is the model that was used in Sec. II.C to derive the nominal initial state of the spacecraft. The major perturbation is due to the 4th-body effect (4BP), i.e. Sun's gravity, which tends to deviate the spacecraft trajectory outwards (see Fig. 9), causing it to miss the target in all scenarios with an average error around 240 m. The solar radiation pressure (SRP) and the uncertain initial position of Dimorphos (δM) give rise to an additional 10-m error each, for a total average miss distance of about 260 m (that is, more than 3 times the mean radius of the asteroid), and a maximum distance of up to 345 m. The control policy π_θ is able to almost completely compensate for the considered dynamical perturbations and guide the spacecraft to collide with Dimorphos in 492 scenarios out of 500 (with just 5 trajectories missing the spherical

approximation of the target asteroid by more than 10 m).

Figure 10 shows the distribution of the impact points on plane x-z of frame N for both the controlled and uncontrolled full-dynamics scenarios. The corresponding error ellipses at 75%, 95%, and 99% confidence levels are also included. As mentioned previously, without any control action the spacecraft tends to move outwards with respect to the binary system as an effect of the perturbations not considered in the trajectory design process. Moreover, the impact points feature a greater dispersion than in the controlled case, as clearly highlighted by the dimension of the error ellipses. When using the optimal control policy π_θ, almost all the impact points fall within the mean outline of Dimorphos (reported as a black dashed circle in the figure). Furthermore, they are concentrated near the central region of the asteroid, thus also showing some level of robustness against possible non-spherical shapes of Dimorphos.

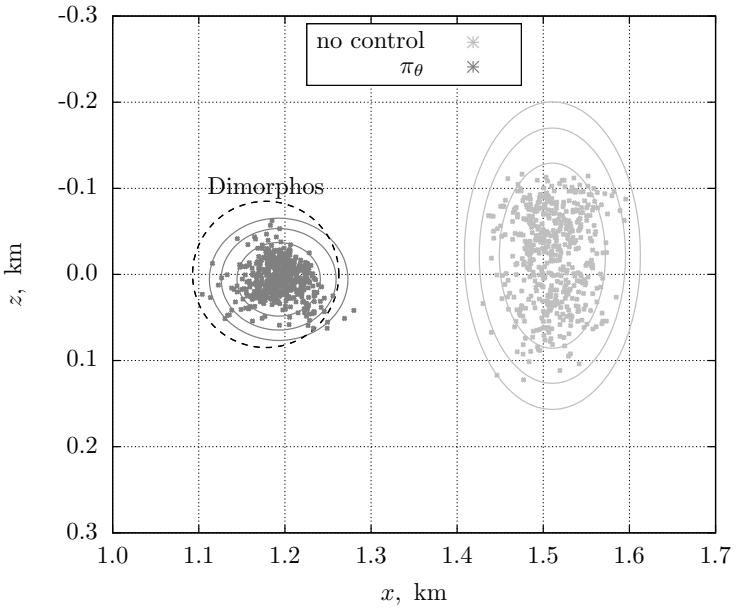


Fig. 10 Impact points w/ and w/o control in-plane x-z.

The thrust magnitude along the Monte Carlo trajectories is shown in Fig. 11. The corresponding thrust components in frame N are instead shown in Fig. 12. It is interesting to note that, in all cases, the spacecraft realizes a pure ballistic flight until 1 hour before the designated impact time. At that point, the spacecraft performs a long burn lasting more than half an hour, followed by a series of other burns whose norm varies greatly among the different trajectories and whose time-length tends to decrease while approaching the impact time. To be noted that the thrust modulus is always significantly lower than the maximum value, represented as a black dashed line. The corresponding propellant consumption is always below 10 g, and, for this reason, it has not been included in the merit index.

This trend of the thrust law can be explained by looking at Fig. 13, which shows the images fed as observations to the network in four key moments along a sample trajectory (bottom half of the figure). For comparison, a high-definition version of the same frames, with the same resolution as the real DRACO camera, is reported in the top half. An hour

before the impact (Fig. 13a), the camera starts seeing the binary system as a handful of white pixels. So, the spacecraft becomes able to determine its relative position, velocity, and approach direction with respect to the system and start maneuvering accordingly. At the end of a long first burn (Fig. 13b), the camera can distinguish the two asteroids from each other, as Dimorphos is seen as a separate dot of light. So, the guidance network, thanks to the recurrent unit, can start predicting what will be the final position of the asteroid's moon at arrival. This information will become clearer the closer the spacecraft gets to the system (Fig. 13c), so it can start controlling its trajectory with more precision with a sequence of shorter-duration burns. The controlled phase is forced to terminate 2 minutes before the impact (Fig. 13d) when the last coasting drives the spacecraft towards the target.

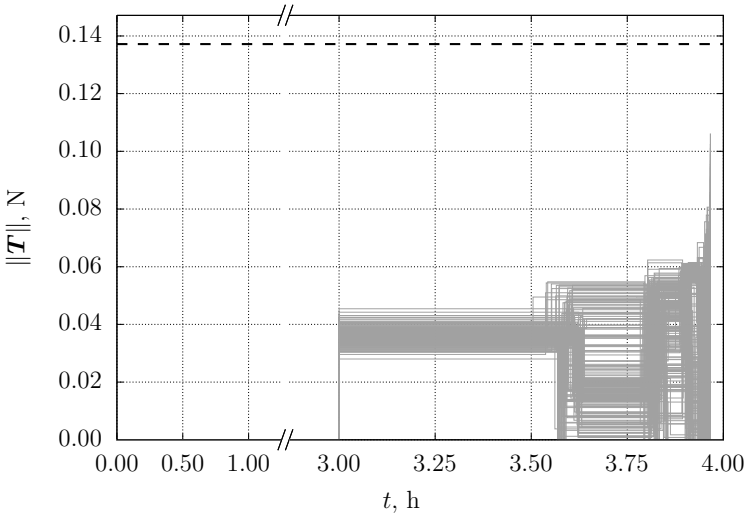
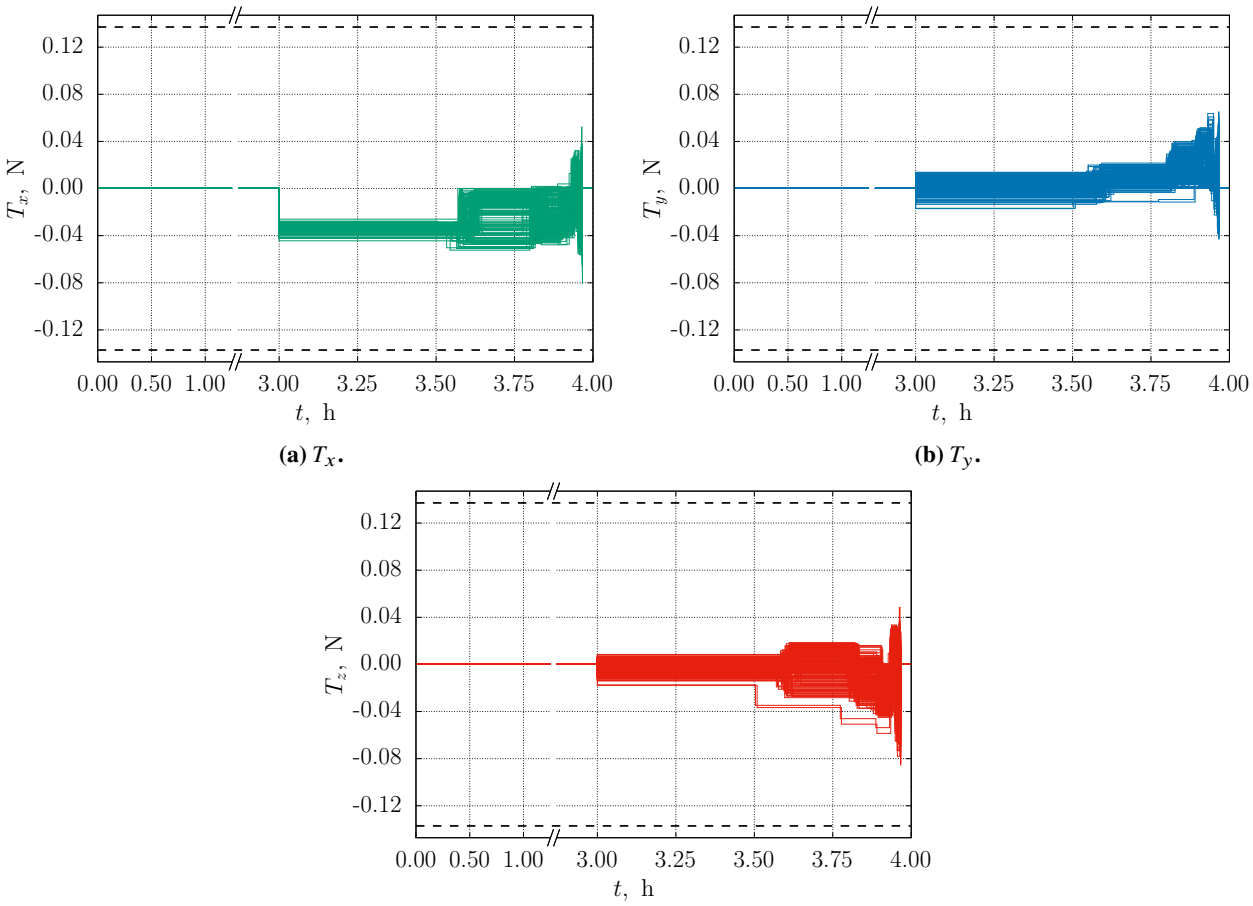


Fig. 11 Thrust magnitude.

Figure 13 shows also that, with a higher-resolution camera, the spacecraft would have been able to distinguish the two asteroids much sooner (see Fig. 13a). So, it is plausible that, in this case, the maneuvering sequence would have been different from the one presented above. Further analysis with the real camera resolution are thus due to better understand the dependence of the control law on the image size. Anyway, as clear from Fig. 13d, the differences between the two cameras' resolutions become less and less important as the spacecraft approaches the asteroids.

It is worth underlining that the control policy has been left free to autonomously determine the time-length of each simulation step with the aim of not biasing the obtained solutions with design choices taken externally, such as a fixed time-step size or a lower guidance frequency in the first part of the simulation. Indeed, in principle, the optimization algorithm could have determined that a non-zero control was also necessary during the early phases of the mission, when the network receives completely dark images. Conversely, numerical results showed that properly controlling the spacecraft just in the last hour of the mission is sufficient to impact the target asteroid in (almost) all considered scenarios. So, a posteriori, one can safely say that, with the camera resolution considered in this study, similar performance could be probably achieved in a shorter training time by starting the image-based guidance just an hour before the impact.



(a) T_x**.**

(b) T_y**.**

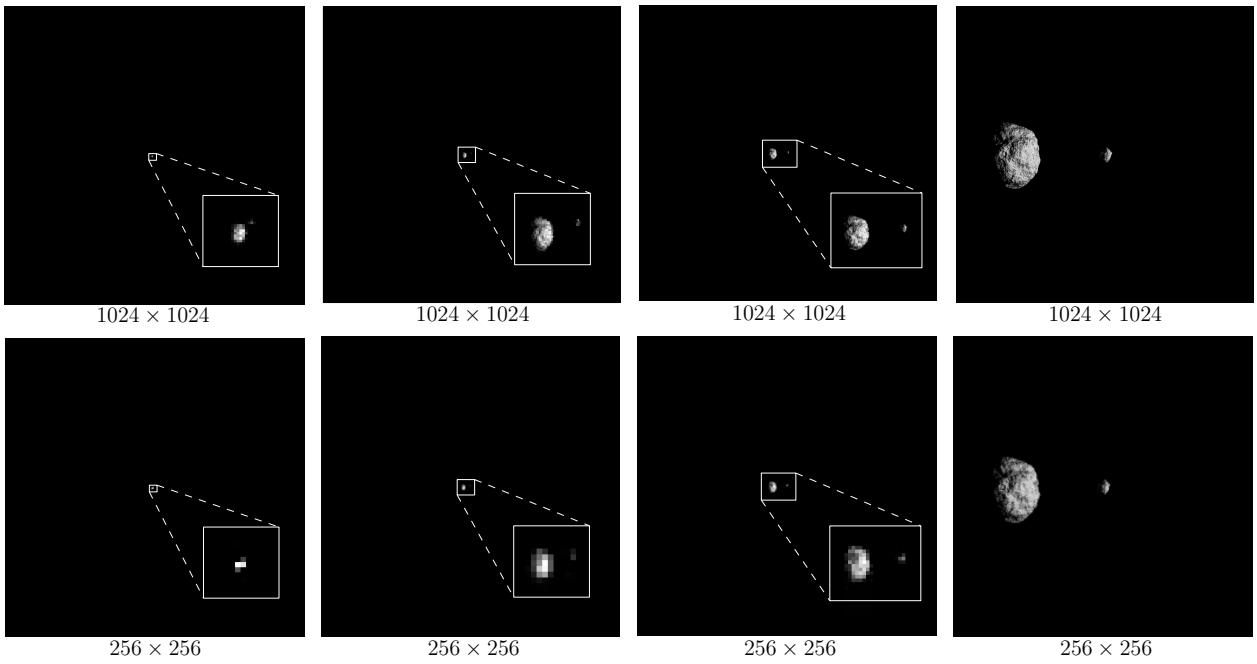
(c) T_z**.**

Fig. 12 Components of the thrust.

A final note concerns the computational effort required to computing the control when the trained network is deployed on the flight hardware, which is one of the main advantages of the proposed neural-network-based guidance approach. On a standard pc equipped with an NVIDIA GeForce GTX 1050 Ti GPU, the average computational time required for computing a single action through the network is 3.8 ms, that is, much lower than the minimum step-size considered in this application (1 s). A computational time of the same order of magnitude is expected on a typical onboard architecture (for example, an NVIDIA Jetson Nano) with roughly half the clock speed, but which features more optimized deep learning libraries.

VI. Conclusion

This paper presented a guidance algorithm based on meta-reinforcement learning to accomplish, in autonomy, the terminal mission phase of an asteroid impactor in a binary system. The guidance system consists of a convolutional-recurrent neural network, which takes as input the current mission time and the images collected in real-time by the onboard camera, and returns as output the corresponding control thrust and firing time. Specifically, the neural network



(a) $t = 3\,\mathrm{h}$, $r_{34} = 47\,351\,\mathrm{km}$. **Start of first burn.** **(b)** $t = 3\,\mathrm{h}\,35'$, $r_{34} = 9672\,\mathrm{km}$. **End of first burn.** **(c)** $t = 3\,\mathrm{h}\,50'$, $r_{34} = 3905\,\mathrm{km}$. **End of second burn.** **(d)** $t = 3\,\mathrm{h}\,58'$, $r_{34} = 783\,\mathrm{km}$. **Start of coasting phase.**

Fig. 13 Example image sequence. Bottom: low-resolution images used for training; top: high-resolution version.

is composed of four convolutional layers, a fully-connected layer, and a final long short-term memory block. The policy-gradient method proximal policy optimization (PPO) is used as a training algorithm.

The performance of the guidance system is tested in a mission scenario that simulates the final phase of the DART mission to the minor asteroid (Dimorphos) of the 65803 Didymos system. The spacecraft state at the beginning of the considered scenario, that is, four hours before the impact, is derived starting from the conditions at impact time provided by mission specifications by using simplified dynamics (two-body). A bi-elliptic restricted four-body problem, with the solar radiation pressure perturbation, is considered as real dynamical model to propagate the spacecraft motion along its approach trajectory. A random uncertainty on the initial mean anomaly of Dimorphos is also taken into account to reflect a non-perfect knowledge of the binary system dynamics. A final two-minute coasting is also enforced to meet DART mission requirements. The presence of the perturbative accelerations not modeled during mission design, together with the random error on Dimorphos' position, causes the spacecraft to miss the target in absence of control actions. So, the objective of the guidance system was to still enable the spacecraft to hit the asteroid in all the possible environment realizations.

The final Monte Carlo simulations demonstrated that a image-based guidance network trained by PPO is perfectly able to compensate for unmodeled dynamics, scattered initial conditions and uncertainties on the target asteroid position, bringing the spacecraft back on the right collision path in more than 98% of the analyzed scenarios. The solutions found, where the control effort is concentrated in the last quarter of the trajectory, are a direct consequence of the lower

camera resolution than the real one mounted on DART spacecraft. Anyway, even with poorer information about the spacecraft state, the guidance network managed to achieve the mission goals by adapting at its best to the surrounding environment. The use of a recurrent layer inside the policy network was paramount to reconstruct the system state at any time, simply receiving the last captured image as input, thus carrying out the task of the navigation system. The presence of the recurrent layer also provided the network with increased adaptivity to variations in the environment definition caused by uncertain initial conditions (meta-reinforcement learning).

As a final remark, the aim of this preliminary study was to understand meta-RL ability to cope with the asteroid-impact problem when considering biased and scattered initial conditions, as well as an uncertain knowledge of the target position. Anyway, thanks to its model-independence, the current approach can be easily extended to take into account also other types of uncertainties, such as control errors, dynamical perturbations or uncertain model parameters, and noisy observations. The spacecraft attitude can be considered as well as part of the controllable dynamics without any specific issue nor modifications to the RL framework. In principle, the generalization capability of neural networks may also allow the control policy to meet the mission objectives when deployed in a dynamical model with higher fidelity than the training one, provided that the observation space is similar between the two scenarios.

References

[1] Prockter, L., Murchie, S., Cheng, A., Krimigis, S., Farquhar, R., Santo, A., and Trombka, J., "The NEAR shoemaker mission to asteroid 433 eros," *Acta Astronautica*, Vol. 51, No. 1-9, 2002, pp. 491–500. https://doi.org/10.1016/S0094-5765(02)00098-X.

[2] Russell, C., Barucci, M., Binzel, R., Capria, M., Christensen, U., Coradini, A., De Sanctis, M., Feldman, W., Jaumann, R., Keller, H., et al., "Exploring the asteroid belt with ion propulsion: Dawn mission history, status and plans," *Advances in Space Research*, Vol. 40, No. 2, 2007, pp. 193–201. https://doi.org/10.1016/j.asr.2007.05.083.

[3] Kawaguchi, J., "The Hayabusa mission - Its seven years flight," *2011 Symposium on VLSI Circuits - Digest of Technical Papers*, 2011, pp. 2–5.

[4] Watanabe, S.-i., Tsuda, Y., Yoshikawa, M., Tanaka, S., Saiki, T., and Nakazawa, S., "Hayabusa2 mission overview," *Space Science Reviews*, Vol. 208, No. 1-4, 2017, pp. 3–16. https://doi.org/10.1007/s11214-017-0377-1.

[5] Lauretta, D., Balram-Knutson, S., Beshore, E., Boynton, W. V., d'Aubigny, C. D., DellaGiustina, D., Enos, H., Golish, D., Hergenrother, C., Howell, E., et al., "OSIRIS-REx: sample return from asteroid (101955) Bennu," *Space Science Reviews*, Vol. 212, No. 1-2, 2017, pp. 925–984. https://doi.org/10.1007/s11214-017-0405-1.

[6] Pezent, J. B., Sood, R., and Heaton, A., "Near Earth Asteroid (NEA) Scout Solar Sail Contingency Trajectory Design and Analysis," *2018 Space Flight Mechanics Meeting*, 2018. https://doi.org/10.2514/6.2018-0199.

[7] Stanbridge, D., Williams, K., Williams, B., Jackman, C., Weaver, H., Berry, K., Sutter, B., and Englander, J., "Lucy: Navigating a Jupiter Trojan tour," *Advances in the Astronautical Sciences*, Vol. 162, 2018, pp. 3781–3798.

[8] Cheng, A. F., Rivkin, A. S., Michel, P., Atchison, J., Barnouin, O., Benner, L., Chabot, N. L., Ernst, C., Fahnestock, E. G., Kueppers, M., et al., "AIDA DART asteroid deflection test: Planetary defense and science objectives," *Planetary and Space Science*, Vol. 157, 2018, pp. 104–115. https://doi.org/10.1016/j.pss.2018.02.015.

[9] Heiligers, J., and Scheeres, D. J., "Solar-sail orbital motion about asteroids and binary asteroid systems," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 9, 2018, pp. 1947–1962. https://doi.org/10.2514/1.G003235.

[10] D'Ambrosio, A., Circi, C., and Zeng, X., "Solar-photon sail hovering orbits about single and binary asteroids," *Advances in Space Research*, Vol. 63, No. 11, 2019, pp. 3691–3705. https://doi.org/10.1016/J.ASR.2019.02.021.

[11] Guzzetti, D., Sood, R., Chappaz, L., and Baoyin, H., "Stationkeeping Analysis for Solar Sailing the L4 Region of Binary Asteroid Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 6, 2019, pp. 1306–1318. https://doi.org/10.2514/1.G003994.

[12] Tardivel, S., and Scheeres, D. J., "Ballistic deployment of science packages on binary asteroids," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 3, 2013, pp. 700–709. https://doi.org/10.2514/1.59106.

[13] Liang, Y., Gómez, G., Masdemont, J. J., and Xu, M., "Stable Regions of Motion Around a Binary Asteroid System," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 11, 2019, pp. 2521–2531. https://doi.org/10.2514/1.G004217.

[14] Capannolo, A., Ferrari, F., and Lavagna, M., "Families of bounded orbits near binary asteroid 65803 Didymos," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 1, 2019, pp. 189–198. https://doi.org/10.2514/1.G003437.

[15] Chen, M., Atchison, J., Carrelli, D., Ericksen, P., Fletcher, Z., Haque, M., Jenkins, S., Jensenius, M., Mehta, N., Miller, T., O'Shaughnessy, D., Sawyer, C., Superfin, E., Tschiegg, R., and Reed, C., "Small-body maneuvering autonomous real-time navigation (smart nav): Guiding a spacecraft to didymos for nasa's double asteroid redirection test (DART)," *Advances in the Astronautical Sciences*, Vol. 164, 2018, pp. 347–359.

[16] Fletcher, Z. J., Ryan, K. J., Maas, B. J., Dickman, J. R., Hammond, R. P., Bekker, D. L., Nelson, T. W., Mize, J. M., Greenberg, J. M., Hunt, W. M., Smee, S. A., Chabot, N. L., and Cheng, A. F., "Design of the Didymos Reconnaissance and Asteroid Camera for OpNav (DRACO) on the double asteroid redirection test (DART)," *Space Telescopes and Instrumentation 2018: Optical, Infrared, and Millimeter Wave*, Vol. 10698, edited by M. Lystrup, H. A. MacEwen, G. G. Fazio, N. Batalha, N. Siegler, and E. C. Tong, International Society for Optics and Photonics, SPIE, 2018, pp. 602 – 612. https://doi.org/10.1117/12.2310136.

[17] Purpura, G., and Di Lizia, P., "Autonomous GNC strategy for an asteroid impactor mission," *CEAS Space Journal*, Vol. 13, No. 1, 2021, pp. 65–81. https://doi.org/10.1007/s12567-020-00325-5.

[18] Lyzhoft, J. R., Kaplinger, B. D., Wie, B., and Hawkins, M. J., "GPU-Based Optical Navigation and Guidance for a Hypervelocity Asteroid Intercept Vehicle (HAIV)," *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013. https://doi.org/10.2514/6.2013-4966.

[19] Pierson, H. A., and Gashler, M. S., "Deep learning in robotics: a review of recent research," *Advanced Robotics*, Vol. 31, No. 16, 2017, pp. 821–835. https://doi.org/10.1080/01691864.2017.1365009.

[20] Sallab, A. E., Abdou, M., Perot, E., and Yogamani, S., "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, Vol. 2017, No. 19, 2017, pp. 70–76. https://doi.org/10.2352/ISSN.2470-1173.2017.19.AVM-023.

[21] Lample, G., and Chaplot, D. S., "Playing FPS Games with Deep Reinforcement Learning," *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI Press, 2017, p. 2140–2146. https://doi.org/10.5555/3298483.3298548.

[22] Izzo, D., Märtens, M., and Pan, B., "A survey on artificial intelligence trends in spacecraft guidance dynamics and control," *Astrodynamics*, Vol. 3, No. 4, 2019, pp. 287–299. https://doi.org/10.1007/s42064-018-0053-6.

[23] Zavoli, A., and Federici, L., "Reinforcement Learning for Robust Trajectory Design of Interplanetary Missions," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 8, 2021, pp. 1440–1453. https://doi.org/10.2514/1.G005794.

[24] Miller, D., Englander, J. A., and Linares, R., "Interplanetary Low-Thrust Design Using Proximal Policy Optimization," *Advances in the Astronautical Sciences*, Vol. 171, 2020, pp. 1575–1592.

[25] Rubinsztejn, A., Bryan, K., Sood, R., and Laipert, F., "Using Reinforcement Learning to Design Missed Thrust Resilient Trajectories," *2020 AAS/AIAA Astrodynamics Specialist Conference*, Lake Tahoe, virtual, 2020.

[26] Federici, L., Scorsoglio, A., Zavoli, A., and Furfaro, R., "Autonomous Guidance for Cislunar Orbit Transfers via Reinforcement Learning," *2021 AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, virtual, 2021.

[27] LaFarge, N. B., Miller, D., Howell, K. C., and Linares, R., "Autonomous closed-loop guidance using reinforcement learning in a low-thrust, multi-body dynamical environment," *Acta Astronautica*, Vol. 186, 2021, pp. 1–23. https://doi.org/10.1016/j.actaastro.2021.05.014.

[28] Sullivan, C. J., and Bosanac, N., "Using Reinforcement Learning to Design a Low-Thrust Approach into a Periodic Orbit in a Multi-Body System," *AIAA Scitech 2020 Forum*, 2020. https://doi.org/10.2514/6.2020-1914.

[29] Holt, H., Armellin, R., Scorsoglio, A., and Furfaro, R., "Low-thrust Trajectory Design using Closed-loop Feedback-driven Control Laws and State-dependent Parameters," *AIAA Scitech 2020 Forum*, 2020. https://doi.org/10.2514/6.2020-1694.

[30] Arora, L., and Dutta, A., "Reinforcement Learning for Sequential Low-Thrust Orbit Raising Problem," *AIAA Scitech 2020 Forum*, 2020. https://doi.org/10.2514/6.2020-2186.

[31] Federici, L., Benedikter, B., and Zavoli, A., "Deep Learning Techniques for Autonomous Spacecraft Guidance During Proximity Operations," *Journal of Spacecraft and Rockets*, Vol. 58, No. 6, 2021, pp. 1774–1785. https://doi.org/10.2514/1.A35076.

[32] Broida, J., and Linares, R., "Spacecraft rendezvous guidance in cluttered environments via reinforcement learning," *Advances in the Astronautical Sciences*, Vol. 168, 2019, pp. 1777–1788.

[33] Scorsoglio, A., Furfaro, R., Linares, R., and Massari, M., "Actor-critic reinforcement learning approach to relative motion guidance in near-rectilinear orbit," *Advances in the Astronautical Sciences*, Vol. 168, 2019, pp. 1737–1756.

[34] Jiang, J., Zeng, X., Guzzetti, D., and You, Y., "Path planning for asteroid hopping rovers with pre-trained deep reinforcement learning architectures," *Acta Astronautica*, Vol. 171, 2020, pp. 265–279. https://doi.org/10.1016/j.actaastro.2020.03.007.

[35] Silvestrini, S., and Lavagna, M. R., "Spacecraft Formation Relative Trajectories Identification for Collision-Free Maneuvers using Neural-Reconstructed Dynamics," *AIAA Scitech 2020 Forum*, 2020. https://doi.org/10.2514/6.2020-1918.

[36] Furfaro, R., Scorsoglio, A., Linares, R., and Massari, M., "Adaptive generalized ZEM-ZEV feedback guidance for planetary landing via a deep reinforcement learning approach," *Acta Astronautica*, Vol. 171, 2020, pp. 156–171. https://doi.org/10.1016/j.actaastro.2020.02.051.

[37] Gaudet, B., Linares, R., and Furfaro, R., "Deep reinforcement learning for six degree-of-freedom planetary landing," *Advances in Space Research*, Vol. 65, No. 7, 2020, pp. 1723–1741. https://doi.org/10.1016/j.asr.2019.12.030.

[38] Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M., "Learning to reinforcement learn," *arXiv preprint arXiv:1611.05763*, 2016.

[39] Gaudet, B., Linares, R., and Furfaro, R., "Terminal adaptive guidance via reinforcement meta-learning: Applications to autonomous asteroid close-proximity operations," *Acta Astronautica*, Vol. 171, 2020, pp. 1–13. https://doi.org/10.1016/j.actaastro.2020.02.036.

[40] Gaudet, B., Linares, R., and Furfaro, R., "Six degree-of-freedom body-fixed hovering over unmapped asteroids via LIDAR altimetry and reinforcement meta-learning," *Acta Astronautica*, Vol. 172, 2020, pp. 90–99. https://doi.org/10.1016/j.actaastro.2020.03.026.

[41] Federici, L., Scorsoglio, A., Zavoli, A., and Furfaro, R., "Meta-Reinforcement Learning for Adaptive Spacecraft Guidance during Multi-Target Missions," *72nd International Astronautical Congress (IAC)*, Dubai, United Arab Emirates, 2021.

[42] Scorsoglio, A., D'Ambrosio, A., Ghilardi, L., Gaudet, B., Curti, F., and Furfaro, R., "Image-Based Deep Reinforcement Meta-Learning for Autonomous Lunar Landing," *Journal of Spacecraft and Rockets*, 2021, pp. 1–13. https://doi.org/10.2514/1.A35072.

[43] Scorsoglio, A., D'Ambrosio, A., Ghilardi, L., Furfaro, R., Gaudet, B., Linares, R., and Curti, F., "Safe Lunar landing via images: A Reinforcement Meta-Learning application to autonomous hazard avoidance and landing," *2020 AAS/AIAA Astrodynamics Specialist Conference*, Lake Tahoe, virtual, 2020.

[44] Community, B. O., *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL http://www.blender.org.

[45] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017. URL https://www.arXiv:1707.06347.

[46] Assadian, N., and Pourtakdoust, S. H., "On the quasi-equilibria of the BiElliptic four-body problem with non-coplanar motion of primaries," *Acta Astronautica*, Vol. 66, No. 1-2, 2010, pp. 45–58. https://doi.org/10.1016/j.actaastro.2009.05.014.

[47] Brockman, G., et al., "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[48] Scorsoglio, A., and Furfaro, R., "VisualEnv: visual Gym environments with Blender," *arXiv preprint arXiv:2111.08096*, 2021.

[49] Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P., "Trust region policy optimization," *International conference on machine learning*, PMLR, 2015, pp. 1889–1897.

[50] Kullback, S., and Leibler, R. A., "On Information and Sufficiency," *The Annals of Mathematical Statistics*, Vol. 22, No. 1, 1951, pp. 79 – 86. https://doi.org/10.1214/aoms/1177729694.

[51] Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P., "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[52] Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M., "The arcade learning environment: An evaluation platform for general agents," *Journal of Artificial Intelligence Research*, Vol. 47, 2013, pp. 253–279. https://doi.org/10.1613/jair.3912.

[53] Hochreiter, S., and Schmidhuber, J., "Long Short-Term Memory," *Neural Computation*, Vol. 9, No. 8, 1997, pp. 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

[54] Hochreiter, S., Younger, A. S., and Conwell, P. R., "Learning to Learn Using Gradient Descent," *Artificial Neural Networks - ICANN 2001*, Springer Berlin Heidelberg, Berlin, Germany, 2001, pp. 87–94.

[55] Hospedales, T. M., Antoniou, A., Micaelli, P., and Storkey, A. J., "Meta-Learning in Neural Networks: A Survey," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2021. https://doi.org/10.1109/TPAMI.2021.3079209.

[56] Bekker, D., Smith, R., and Tran, M. Q., "Guiding DART to Impact—the FPGA SoC Design of the DRACO Image Processing Pipeline," *2021 IEEE Space Computing Conference (SCC)*, IEEE, 2021, pp. 122–133. https://doi.org/10.1109/SCC49971.2021.00020.

[57] Naidu, S., Benner, L., Brozovic, M., Nolan, M., Ostro, S., Margot, J., Giorgini, J., Hirabayashi, T., Scheeres, D., Pravec, P., Scheirich, P., Magri, C., and Jao, J., "Radar observations and a physical model of binary near-Earth asteroid 65803 Didymos, target of the DART mission," *Icarus*, Vol. 348, 2020. https://doi.org/10.1016/j.icarus.2020.113777.

[58] McQuaide, M., Atchison, J., Bellerose, J., Laipert, F., Mottinger, N., Tarzi, Z., and Velez, D., "Double Asteroid Redirection Test (DART) Phase D Mission Design & Navigation Analysis," *7th IAA Planetary Defense Conference (PDC)*, Wien, Austria, 2021.

[59] Sarli, B. V., Ozimek, M. T., Atchison, J. A., Englander, J. A., and Barbee, B. W., "NASA Double Asteroid Redirection Test (DART) Trajectory Validation and Robustness," *AAS/AIAA Space Flight Mechanics Meeting*, 2017.

[60] Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M. I., et al., "Ray: A distributed framework for emerging AI applications," *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, 2018, pp. 561–577.