Acta Astronautica 225 (2024) 689–706

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Research paper

Robust Model Predictive Control for an Ion Beam Shepherd in a large-debris removal mission

Javier Urrios [a,*], Rafael Vazquez [a], Francisco Gavilan [a], Ignacio Alvarado [b]

[a] Department of Aerospace Engineering, Universidad de Sevilla, Sevilla, 41092, Spain
[b] Department of Systems Engineering and Automation, Universidad de Sevilla, Sevilla, 41092, Spain

ARTICLE INFO

Keywords:
Active debris removal
Ion Beam Shepherd
Model Predictive Control
Robust control
Ray-marching

ABSTRACT

The increasing accumulation of space debris poses significant risks to spacecraft, making the development of effective debris mitigation technologies necessary. This paper explores the Ion Beam Shepherd (IBS) method as a potential contactless solution for deorbiting large debris objects. The IBS system concept involves a spacecraft equipped with an ion thruster to direct a controlled ion beam at the debris, generating a small force that gradually lowers its orbit. A proposed configuration of the chaser's actuator system integrates radial and out-of-plane cold-gas thrusters along with in-track ion thrusters to enhance control and safety while maintaining low mission costs. A robust Model Predictive Control (MPC) strategy is implemented, using the theory of MPC for Tracking to ensure accurate positioning and effective deorbiting. This theoretical approach addresses uncertainties and perturbations to robustly guarantee safe distances between the chaser and the debris. Additionally, a new ray-marching-based algorithm is introduced to estimate the force and torque exerted by the ion beam on the target, considered as a 6 degrees of freedom object, improving simulation accuracy and control performance assessment. A comprehensive simulation of the deorbit of a large debris object is performed, demonstrating the potential of the IBS technology for future large-debris removal missions. This research advances the conceptual framework and control techniques for the IBS technology, advancing towards its future implementation in space debris mitigation.

1. Introduction

The proliferation of space debris has become a critical issue in contemporary space exploration and satellite operations. According to [1], there are more than 30 000 tracked debris objects orbiting Earth, with millions of smaller fragments posing additional risks. These debris objects, traveling at high velocities, can cause significant damage to operational spacecraft, endangering space missions and the lives of astronauts aboard the International Space Station. The accumulation of debris also raises concerns about the potential onset of Kessler syndrome [2], a scenario where cascading collisions among space objects lead to a self-sustaining chain reaction of debris generation, further aggravating the problem.

To mitigate this risk, the space industry has proposed various debris removal strategies [3], which can be broadly categorized as contact and non-contact methods. Contact methods often require precise maneuvering and docking capabilities, whereas contactless methods can operate at a distance, but may require high power or advanced technology to achieve the desired effects. Examples of contact methods are net captures, harpoons, or robotic arms, while the main proposed technologies for contactless removal are laser ablation and ion beams. Ongoing efforts are focused on translating these strategies into practical applications. For example, Astroscale, a Japanese company, recently demonstrated its capabilities by capturing images of a rocket upper stage from a distance of several hundred meters using the ADRAS-J satellite. This mission aims to remove the debris using a robotic arm. Moreover, NASA's Office of Technology, Policy, and Strategy recently published an analysis on the costs and benefits of contact methods for space debris remediation [4]. While contact methods are effective, they require direct contact with the debris, making them less suitable for faster spinning objects.

The Ion Beam Shepherd (IBS) technology (which was nearly simultaneously proposed by [5–7]) represents a promising alternative for space debris mitigation. The IBS system uses a spacecraft equipped with an ion thruster to project a controlled stream of charged particles (ions) towards a target debris object. This ion beam generates a small force, allowing the debris to be gradually deorbited into Earth's atmosphere for disintegration. Usually, a target with an almost circular orbit is chosen (although some studies consider higher eccentricities, such

* Corresponding author.
E-mail addresses: furrios@us.es (J. Urrios), rvazquez1@us.es (R. Vazquez), fgavilan@us.es (F. Gavilan), ialvarado@us.es (I. Alvarado).

https://doi.org/10.1016/j.actaastro.2024.09.054
Received 1 August 2024; Received in revised form 17 September 2024; Accepted 22 September 2024
Available online 26 September 2024

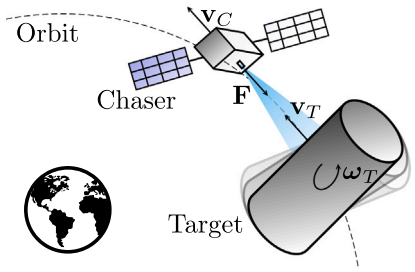

Fig. 1. Ion Beam Shepherd concept.

as [8] or [9]), and the shepherd (or chaser) is set to travel the same orbit but a bit ahead. The chaser then points its ion beam backwards to slow down the debris (see Fig. 1). This technology offers several advantages: (i) it avoids direct contact with debris, increasing the safety of operations; (ii) it works independently of the target shape, rotational state or material; (iii) it is reusable until fuel depletion, potentially enabling multi-target Active Debris Removal (ADR) missions; (iv) taking into account that the momentum exchange is performed remotely, it is relatively efficient given its high specific impulse and that ions are concentrated in a narrow cone; and (v) it is based on mature and space-tested electric propulsion technology. Moreover, the IBS is well suited to tackle large space debris objects, which are of special interest. Indeed, according to [4], the deorbiting of large debris objects is highly beneficial, because it not only eliminates the necessity of spacecraft to constantly maneuver to avoid collisions, but also mitigates the generation of smaller debris resulting from surface degradation. Additionally, the deorbiting of these large objects effectively neutralizes their high potential for further debris creation. However, the ion beam also poses demanding challenges; the relative position between the chaser and the target must be accurately kept for the method to be effective, and formation flying with the target at such a small distance requires a robust control that guarantees no collision and deals with uncertainty and perturbations.

For the IBS method to be effective, the ion beam should be accurately and efficiently characterized, in order to properly compute its interaction with the debris. Efforts towards these tasks were made in [10–12], and even some in-house software was developed to study such interaction [13]. The next task was to build a subsystem capable of generating such a beam and optimize it to improve the beam convergence [12] and minimize power consumption [14]. Guidelines and constraints to be taken into account in the preliminary design of an IBS mission were outlined in [15]. Some algorithms for on-board force computation (although not torque) have also been developed [16,17], the latter leveraging deep learning techniques. Furthermore, additional research has investigated more advanced consequences of the ion beam–target interaction such as surface sputtering and backscattering contamination, through numerical simulations [18] and experiments [12,19], with the aim of further improving the models.

In terms of the relative positioning and control of the chaser with respect to the target, different configurations have been developed. Conventionally, the shepherd satellite consists of the main ion beam thruster, usually called the Impulse Transfer Thruster (ITT), which is the one that points to the space debris object (target), and another Impulse Compensation Thruster (ICT) in the opposite direction, to counteract the reaction force generated by the ITT [7]. In order to extract energy as efficiently as possible, the ion beam should act in the opposite direction of the target's velocity, that is, it should be positioned in front of the debris. Initially, as part of a preliminary study, in [20] it was assumed that the chaser had control thrusters capable of continuous control in all directions, assumed no measurement errors and a spherical target (and thus no considerations in terms of attitude). Then, [21] extended the formulation to include state errors while preserving robustness, and made use of Pulse-Width Modulated hydrazine thrusters, while maintaining the spherical form hypothesis. Later, the attitude evolution of the target was thoroughly studied [22,23], along with possible strategies to control it [9,24].

The efficiency of the relative position control presented in [20] can be further enhanced by considering alternate control strategies. For instance, control in the in-track direction could be achieved by modulating the force exerted by the ICT. This approach was investigated in [25], where the problem was simplified to a planar case, and the system was controlled solely through ICT modulation, without any control in the radial direction. This control strategy demonstrated the potential for reduced fuel consumption. To achieve this, and H_∞ based controller was implemented for a time-varying and parametric uncertain plant. However, the chaser remained subject to periodic relative position errors due to the low controllability of the system and the small but non-zero eccentricity of the orbits. Later, more eccentric orbits were considered in [8], where a periodic Linear Quadratic Regulator (LQR) with a periodic reference input was implemented to minimize periodic errors due to the time-varying plant. These relative position errors were reduced to around 1 m for a LEO orbit with 0.1 eccentricity. Moreover, the work in [26] extended the formulation in [25] to the three-dimensional case, and discovered that it was also feasible to control the system in the out-of-plane direction by only performing yaw attitude deviations, which reduces fuel consumption since there is no need for an additional thruster. More recently, a new work proposes multiple ITTs to control the relative position of the chaser and the attitude of the target simultaneously, while deorbiting the target faster [27].

Although the developments mentioned above are interesting and crucial, some of them may not be suitable for implementation in the near future given the low Technology Readiness Level (TRL) of the IBS technology. For example, even if the yaw control in [26] works in simulations, it might not be safe to send such a satellite without a highly responsive controller in the out-of-plane direction. Similarly, although it is technically possible to control in-plane dynamics just with modulation of the ICT [25], adding an actuator in the radial direction is a reasonable way to reduce the probability of impact between the chaser and the target. Lastly, as powerful as the idea of using multiple beams for simultaneous attitude and relative position control [27] sounds, for an initial mission, it is advisable to aim for a minimal number of ion beams to mitigate the risk of component or system malfunction. By reducing the complexity of the propulsion system, the overall reliability of the spacecraft can be enhanced, thereby increasing the likelihood of mission success.

This paper proposes a new configuration of the chaser actuator system, where the ITT and ICT in the in-track directions are maintained, but on–off cold-gas thrusters are added in the radial and out-of-plane directions (see Fig. 2). Cold-gas thrusters are simple and reliable actuators, especially the ones that operate in a binary on–off mode, and the addition of them in the radial and out-of-plane directions ensures controllability and security, while not increasing the mission budget too much. For the control, a Robust Model Predicitive Controller for Tracking is implemented.

Model Predictive Control (MPC) [28] is an advanced control strategy used in a variety of industries, including aerospace. Unlike traditional control methods, MPC looks into the future by predicting the system's behavior over a specified time horizon and optimizing control actions accordingly. It utilizes a dynamic model of the system, which is usually Linear Time-Invariant (LTI), to simulate future behavior and selects control inputs that minimize a defined cost function, accounting for constraints on states and inputs. MPC has already been used in similar use cases, such as rendezvous [29], also in the presence of on–off thrusters [30].

There are important aspects that must be taken into account in the design of an MPC controller, mainly feasibility, stability, and robustness [31]. Feasibility refers to the ability of the MPC controller to find

a solution that satisfies all the constraints imposed on the system while achieving the control objectives. Stability is the ability of the closed-loop system to remain bounded and converge to a desired operating point or trajectory over time. Lastly, robustness indicates the ability of the controller to maintain satisfactory performance in the presence of uncertainties, disturbances, and model mismatches. There are several formulations in which these aspects can be tackled. In this work, a Model Predictive Controller based on the formulations developed in [32,33] is implemented, which combines MPC for Tracking and Robust MPC. Robust MPC has been merged with MPC for Tracking before (see [34]), employing the notion of tubes to ensure robustness. Here, however, to ensure robustness, the constraint tightening approach in [32] is taken instead, and thus the final controller is different.

MPC for Tracking [33] was initially developed to enable systems to follow piece-wise changing references without losing the feasibility of the controller. This is not of much use in this work, since the chaser should regulate itself to a nominal nonchanging relative position with respect to the target. However, another consequence of implementing the MPC for Tracking scheme is that the region of attraction of the controller (region in which the controller successfully operates) is expanded, enhancing its capabilities and, in this case, further reducing fuel consumption.

Robust MPC (for example, the formulation in [32]) was developed to account for uncertainties in the dynamic model given to the MPC. These uncertainties could be caused by perturbations, linearization errors, or others. Assuming the uncertainties are unknown but bounded, the goal of the Robust MPC scheme is to find a solution that minimizes the cost function while ensuring the constraints are met for any value of the uncertainties, and guaranteeing recursive feasibility and stability. This formulation is extremely useful for the IBS deorbiting scenario, since robust constraints on the position of the chaser can be set so that the ion beam always impacts the target efficiently for any target's attitude state.

The contribution of this work lies in the conceptual design of a robust control system for a realistic IBS mission scenario, utilizing a new actuator configuration. Following the literature [25,27], the deorbiting of a large debris object has been considered. However, it must be acknowledged that, once in very low orbits, the process of reentry from such large objects would be random with associated unacceptable risks on ground and thus a secondary mechanism to ensure a safe reentry would be required, which is beyond the scope of this paper. The mission aims to deorbit the current most statistically concerning debris object according to [35], the Zenit-2 second stage rocket body (NORAD ID: 28353), down to an orbit of 340 km, although in principle this framework can be used to deorbit many other debris objects. Perturbations and unmodeled dynamic effects are bounded and fed to the controller to ensure the chaser maintains a safe distance from the target, while preserving a fairly large region of attraction and not impacting fuel consumption. This proposed configuration incorporates cold-gas thrusters in the radial and out-of-plane directions, alongside the existing in-track ion thrusters (ITT and ICT). This design concept aims to enhance controllability and safety without significantly increasing the mission budget. Additionally, this paper introduces a novel method for computing the force and torque exerted by the ion beam on the target using a ray-marching algorithm that is integrated into the mission simulator. In this way, the performance of the controller can be measured more accurately. By addressing these aspects, this work advances the theoretical understanding and control algorithms of the IBS technology for large space debris mitigation, providing a potential solution for future implementation.

This paper is organized as follows. In Section 2, the problem statement is presented and the equations of motion are introduced, together with the modeled perturbations. Section 3 includes an exhaustive explanation of how the ion beam is modeled, introducing a new algorithm based on ray-marching to compute the force and torque exerted on the target. Next, Section 4 tackles all aspects related to control, introducing a linear model of the system, the constraints to meet by the controller, and the controller itself. Disturbance bounding is also performed in this section, and considerations regarding recursive feasibility, convergence, and the tuning of the controller are present. In Section 5, a simulation of the whole deorbiting mission is performed, and relevant results are shown and discussed. Lastly, the conclusions are summarized in Section 6, along with possible future work.

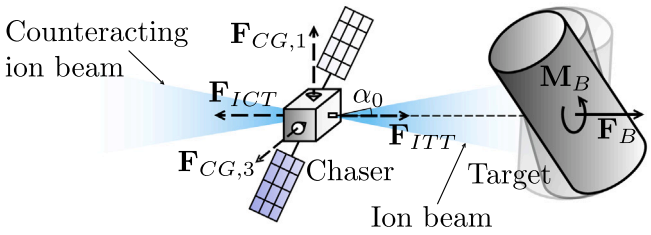

Fig. 2. Proposed IBS configuration.

2. Problem statement and equations of motion

The deorbiting of a large space debris object using IBS technology is considered in this paper. Depending on the dimensions of the target body, and leaving security margins to account for differences between the target's Center of mass (COM) and geometric center, the distance between the chaser's COM and the ion beam source, and some extra space in case of malfunctioning, the chaser will be positioned a certain distance d_{CT} in front of the target.[1] The chaser has to maintain this distant position accurately, because due to the separation between the two bodies, small deviations in the chaser position means the beam hits the target only partially, notably reducing efficiency.

To perform the deorbit, the chaser must maintain a precise relative position with the target to correctly transfer the momentum of the ion beam. Thus, the spacecraft must have the proper actuators, and a suitable controller must be designed. The spacecraft is equipped with an ITT that maintains its thrust constant with a value of F_{ITT} and an ICT in the opposite direction that can vary its thrust F_{ICT} for control purposes. In the perpendicular directions, the chaser is equipped with cold-gas thrusters.

It is assumed that the chaser possesses an attitude control system capable of always orienting the ITT in the negative in-track direction, which in perfect circular orbits coincides exactly with the target direction. The chaser is assumed to have sensors capable of estimating the relative state of the target and also some orbit parameters used by the controller such as the mean motion n of the orbit. These parameters are estimated with a certain error that is bounded.

In the subsequent sections, the following reference frames are used:

- Earth-Centered-Inertial (ECI) $O_I x_I y_I z_I$, which is an inertial reference frame centered on the Earth, coincident with J2000.
- Local-Vertical Local-Horizontal frame (LVLH) $O_L x_L y_L z_L$. Although this frame is not used by itself, it helps us introduce later frames. Generically, an LVLH frame is a frame centered on an object that orbits a certain body, and their axes are defined as follows. The x_L axis points towards the object position vector as measured from the orbited body, z_L points towards the angular momentum vector of the object, and y_L completes the Cartesian right-handed coordinate system.
- Chaser-centered LVLH frame (CLVLH) $O_C x_C y_C z_C$. It is the LVLH frame centered on the chaser's COM and orbiting the Earth (see Fig. 3).

[1] See Nomenclature at end of paper

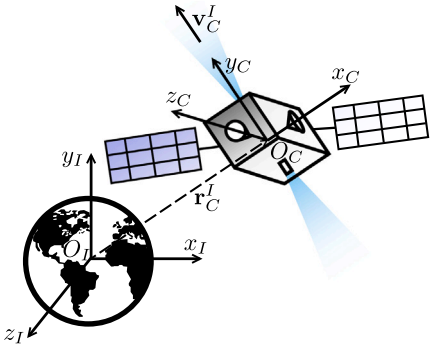

Fig. 3. Chaser-centered LVLH frame definition.

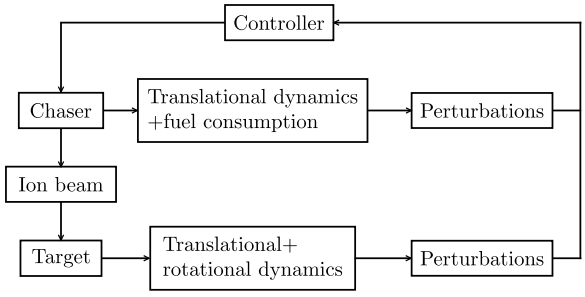

Fig. 4. Block diagram of the problem.

- Target-centered LVLH frame (TLVLH) $O_T x_T y_T z_T$. Analogous to the CLVLH, but the origin is now the target's COM, which is assumed to coincide with its geometric center.
- Target Body Axes (TBA) $O_B x_B y_B z_B$. With its origin at the target's COM, it is fixed to the target, with its z_B axis coinciding with the cylinder revolution axis.

The state variables of the problem are the position and velocity of both the chaser and the target in the ECI frame, $\mathbf{r}_C^I$, $\mathbf{v}_C^I$, $\mathbf{r}_T^I$, $\mathbf{v}_T^I$; the attitude quaternion of the target with respect to ECI $\mathbf{q}_I^B$; the angular velocity of the target in the body axes $\boldsymbol{\omega}_T^B$; and the mass of the chaser m_C. A schematic view of the system is included in Fig. 4.

2.1. Equations of translational motion

The target and chaser dynamics are given by the restricted two-body problem with added perturbations. More specifically, the equations of motion of the COMs of the chaser and target in the ECI frame are

$$\ddot{\mathbf{r}}_C^I = -\frac{\mu_E}{r_C^3}\mathbf{r}_C^I + \mathbf{a}_{C,B}^I + \mathbf{a}_{C,C}^I + \mathbf{a}_{C,D}^I + \mathbf{a}_{C,J2}^I \tag{1}$$

$$\ddot{\mathbf{r}}_T^I = -\frac{\mu_E}{r_T^3}\mathbf{r}_T^I + \mathbf{a}_{T,B}^I + \mathbf{a}_{T,D}^I + \mathbf{a}_{T,J2}^I\,, \tag{2}$$

where $\mu_E = 3.9860044 \cdot 10^5$ km^3/s^2 is the geocentric gravitational constant, and the subscripts C, T represent the chaser and target respectively. The first term of both equations represents the unperturbed dynamics of the two-body problem, while the rest are accelerations due to the ion beam ($\mathbf{a}_{C,B}^I$ and $\mathbf{a}_{T,B}^I$, B subscript from *beam*), control ($\mathbf{a}_{C,C}^I$, C subscript from *control*), drag ($\mathbf{a}_{C,D}^I$ and $\mathbf{a}_{T,D}^I$, D subscript from *drag*) and J2 perturbation ($\mathbf{a}_{C,J2}^I$ and $\mathbf{a}_{T,J2}^I$). Drag and J2 perturbations are chosen because they are the most prominent for a satellite in a Low Earth Orbit (LEO).

2.1.1. Ion beam acceleration

The force $\mathbf{F}_B^C$ exerted by the ion beam on the target is calculated using the method explained in Section 3, and it is expressed in the CLVLH frame. Therefore, the acceleration is obtained as

$$\mathbf{a}_{T,B}^I = R_C^I \frac{\mathbf{F}_B^C}{m_T}, \tag{3}$$

where R_C^I is the DCM that transforms from CLVLH to ECI. The equilibrium state of the system is chosen so that the ion beam fully impacts the target, maximizing the force transmitted to it ($F_B^{C,max} = F_{ITT}$) and thus imparting a maximum acceleration of $\mathbf{a}_{B,T}^{I,max}$. For the system to remain in this equilibrium (ignoring perturbations and unmodeled dynamics), the nominal value of the ICT must be such that the acceleration of the chaser and the target due to the beam are equal

$$\mathbf{a}_{C,B}^I = \mathbf{a}_{T,B}^{I,max}. \tag{4}$$

Note that during the mission, the ion beam may impact the target only partially. The assumption in (4) is only considered to linearize the system along that configuration.

2.1.2. Control acceleration

There is a component of the acceleration that the chaser experiments that comes from the action of controlling the system, $\mathbf{a}_{C,C}^I$. The design of the controller as well as the actuators' behavior are explained in Section 4. Logically, there is no such acceleration term for the target.

2.1.3. Drag perturbation

To model drag, the next simple model is used

$$\mathbf{a}_{i,D} = -\frac{1}{2}\rho_a(h_i) v_{ri} \mathbf{v}_{ri} \frac{A_{f,i}}{m_i} c_{D,i}, \tag{5}$$

where the subindex i stands for chaser C and target T respectively; $\rho_a(h_i)$ is the atmosphere density, which is computed using the International Standard Atmosphere (ISA) model knowing the height h_i; $\mathbf{v}_{ri}$ is the relative velocity between the object and the air (the latter is assumed to rotate in solidarity with the Earth); $A_{f,i}$ is the frontal area; and $c_{D,i}$ is the drag coefficient. The chaser frontal area is assumed to be constant since it always points in the same direction, while the target frontal area depends on its attitude. Knowing the attitude with respect to the TLVLH frame R_T^B, the relative velocity in that same frame $\mathbf{v}_{rT}^T$, and that the cylinder symmetry axis goes along z_B, the frontal area can be computed as

$$\delta = \min\left(\arccos\left(\begin{bmatrix}0 & 0 & 1\end{bmatrix} R_T^B \frac{\mathbf{v}_{rT}^T}{v_{rT}}\right), \arccos\left(\begin{bmatrix}0 & 0 & -1\end{bmatrix} R_T^B \frac{\mathbf{v}_{rT}^T}{v_{rT}}\right)\right), \tag{6}$$

$$A_{f,T} = 2R_c H_c \sin\delta + \pi R_c^2 \cos\delta, \tag{7}$$

where R_c is the cylinder radius, H_c its height, and δ is the angle between the cylinder axis and the in-track direction (which is between 0 and 90 degrees).

2.1.4. J2 perturbation

J2 perturbation, with its origin in Earth's oblateness, is also included. This kind of harmonic perturbations, which appear due to the fact that the Earth is not a perfect sphere, are computed in the Earth-Centered Earth-Fixed frame (ECEF) for obvious reasons. However, taking into account that the z axes of both ECI and ECEF are almost coincident, and that J2 only depends on the latitude of the satellite (and not longitude), the J2 perturbation can be expressed directly in the ECI frame almost without losing accuracy. The acceleration due to J2 is

$$\mathbf{a}_{i,J2}^I = \frac{3}{2}\mu_E J_2 \frac{\mathbf{r}_i^I}{r_i^3}\left(\frac{R_E}{r_i}\right)^2 \circ \left(5\left(\frac{z_i^I}{r_i}\right)^2 \begin{bmatrix}1\\1\\1\end{bmatrix} - \begin{bmatrix}1\\1\\3\end{bmatrix}\right), \tag{8}$$

where again the subscript i stands for chaser C and target T respectively, $J_2 = 0.0010826269$ and $R_E = 6378.1366$ km is the radius of the Earth. The operator $\circ$ denotes the Hadamard product or element-wise product of two vectors.

2.2. Equations of rotational motion

When the beam impacts the space debris object, it exerts not only a force but also a torque, which is also computed by the algorithm presented in Section 3. This torque changes the angular velocity of the debris, and it could potentially accelerate it indefinitely, which is not desirable since phenomena such as centrifugal fragmentation could occur. To ensure this does not happen, and to simulate a more realistic scenario (the frontal area of the debris changes depending on its attitude), the rotational dynamics are also included in the simulation:

$$\dot{\mathbf{q}}_I^B = \frac{1}{2}\mathbf{q}_I^B * \begin{bmatrix} 0 \\ \boldsymbol{\omega}_T^B \end{bmatrix}, \tag{9}$$

$$\dot{\boldsymbol{\omega}}_T^B = \mathcal{I}^{-1}\left(\mathbf{M}_B^B - \boldsymbol{\omega}_T^B \times \mathcal{I}\boldsymbol{\omega}_T^B\right), \tag{10}$$

where $\mathbf{q}_I^B$ is the scalar-first attitude quaternion that represents the attitude of the target with respect to ECI, $*$ represents quaternion multiplication, $\boldsymbol{\omega}_T^B$ is the target angular velocity, $\mathbf{M}_B^B$ is the torque exerted by the beam, and $\mathcal{I}$ is the inertia tensor of the target, with the latter three quantities expressed in the body frame. In this work, the target is modeled as a cylinder with constant density, which means that the COM and the geometric center coincide. With these assumptions, the inertia tensor is computed as

$$\mathcal{I}_x = \mathcal{I}_y = \frac{1}{12}m_T H_c^2 + \frac{1}{4}m_T R_c^2, \tag{11}$$

$$\mathcal{I}_z = \frac{1}{2}m_T R_c^2, \tag{12}$$

$$\mathcal{I}_{xy} = \mathcal{I}_{yz} = \mathcal{I}_{xz} = 0, \tag{13}$$

where m_T is the target mass, and R_c and H_c are the cylinder radius and height respectively. Note that the only torque source for the rotational dynamics is assumed to be the ion beam. There could be more, such as gravity gradient or drag, but they are small and slow, and have been neglected for simplicity since they would have little impact on rotational motion.

2.3. Fuel consumption

As the chaser is controlled with both cold-gas thrusters and ion beams, it uses fuel and thus loses mass. This mass variation is important, as it changes the force that the actuators need to exert to produce the same acceleration. Note that the rules that are commanded to the actuators are computed at the sampling time, even if they are applied some time later. The forces actuating on the chaser are the ITT and the ICT in the in-track direction, and the cold-gas thrusters in the radial and out-of-plane directions. Recall from Section 2.1.1 that ignoring perturbations and unmodeled dynamics, and assuming that the beam completely impacts the target, the ICT force must be such that the system remains in equilibrium with no control ($\mathbf{a}_{C,C}^I = 0$, $\mathbf{a}_{C,C}^I$ is now referred to as $\tilde{\mathbf{u}}$ for clarity). In that scenario, at a generic time instant, the in-track accelerations of the target and the chaser are $a_{T,it} = -F_{ITT}/m_T$ and $a_{C,it} = (F_{ITT} - F_{ICT})/m_C^-$ respectively, where m_C^- represents the chaser mass at the most recent sampling time. Therefore, for a generic acceleration $\tilde{u}_y$ commanded in the in-track direction, the ICT must be such that $\tilde{u}_y = a_{it,C} - a_{it,T}$, that is

$$F_{ICT} = F_{ITT}\left(1 + \frac{m_C^-}{m_T}\right) - m_C^-\tilde{u}_y. \tag{14}$$

Thus, the complete fuel consumption equation is given by

$$\dot{m}_C = -\frac{\left(2 + \frac{m_C^-}{m_T}\right)F_{ITT} - m_C^-\tilde{u}_y}{g_0 I_{sp,IB}} - \frac{m_C^-\left(|\tilde{u}_x| + |\tilde{u}_z|\right)}{g_0 I_{sp,CG}}, \tag{15}$$

where $I_{sp,IB}$ and $I_{sp,CG}$ are the specific impulses of the ion beam and the cold-gas thruster respectively, while $g_0 = 9.80665\ \mathrm{m/s^2}$ is the standard gravity.

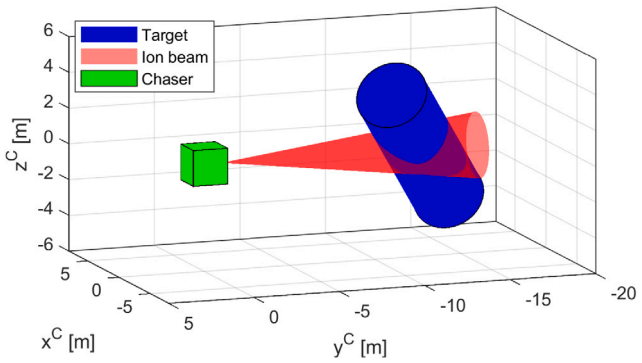

Fig. 5. Depiction of the impact of the ion beam in the target, with a generic relative position and attitude of the later.

2.4. System uncertainties

Within the system, uncertainties related to sensoring and actuation are included. These uncertainties are modeled as uniform distributions so that they have a bounded maximum value. This is necessary to ensure the robustness of the controller in Section 4. If uncertainties were to be modeled as normal distributions or similar, a chance-based approach can be used, in which it is possible to give a probability (which is now not 100%) that the constraints will be met (see, for example, [29]).

The spacecraft has sensors that estimate the relative position and velocity of the target with respect to the chaser, and these measurements are assumed to have some error. If some advanced sensoring system based on lidar/radar is considered, bounding values of the relative position errors are of the order of ±1 cm for the relative position and ±5 cm/min for the relative velocity. Moreover, the ion beam force is assumed to be not completely stationary, slightly changing the divergence angle of the beam. This effect is modeled by considering that the force exerted on the target can vary as much as ±5% of the nominal value.

3. Ion beam modeling. Force and torque computation

Modeling the ion beam is crucial for an accurate simulation of the controlled system. However, computing the force and torque that the beam exerts on the target is not trivial. With an arbitrary position and attitude of the target, perhaps only part of the beam impacts the object and the rest does not (see Fig. 5), and the points where it impacts are crucial for force and torque calculations. To solve this task, some groups discretize the target surface as a triangle mesh and locally compute the force and torque on each one [22]. Others use properties of the assumed ion beam model to project the force and compute it in two dimensions [16]. Another group has even developed internal software for this task [13], based on finite element methods and a shadowing algorithm. In this work, another approach based on ray-marching is chosen, and it is explained in this section.

3.1. Ray-marching basics

Ray-marching is a rendering technique used in computer graphics to generate images by casting rays into a scene and incrementally advancing along each ray to find intersections with objects [36]. In this paper, no images are generated using this technique, but rather the technique is used to compute the distance from a point to the first intersecting object in some given direction, that is, from the ion beam cone vertex to the target in some given direction. The idea is that one can find the distance by iteratively advancing along the ray a distance equal to the shortest distance to the closest object, as Fig. 6 shows.

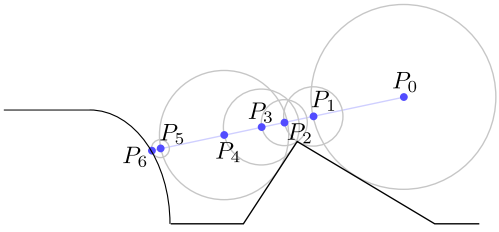

Fig. 6. Ray-marching depiction.

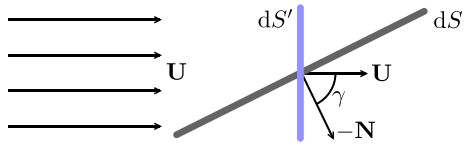

Fig. 7. Relationship between a target differential surface element $\mathrm{d}S$ and a beam differential surface element $\mathrm{d}S'$.

If that shortest-distance computation is cheap (which is the case for simple geometrical shapes, such as a cylinder, as shown in Section 3.3), then the algorithm converges rapidly to the desired distance. Ray-marching may not be suited for more complex geometries, but to the authors' knowledge, no complex geometry for the space debris object has been used yet for the study of a potentially viable IBS deorbit mission, and even then, more complex shapes generated from Boolean operations applied on simple mathematical shapes are also suited for ray-marching.

3.2. Beam-focused discretization of the force

Under some typical simplifying assumptions [20], the differential force that the ion beam exerts in a differential surface element of the target is given by

$$\mathrm{d}\mathbf{F}_B = m_I n_I \mathbf{U}\,(-\mathbf{N}\cdot\mathbf{U})\,\mathrm{d}S, \tag{16}$$

where m_I is the ion mass, n_I is the number density of ions, $\mathbf{U}$ is the ion velocity vector, and $\mathbf{N}$ is the external unit normal vector to the elementary area $\mathrm{d}S$. Under the same assumptions, a self-similar model of the fluid can be developed to determine the ion number density and velocity at any point far from the beam source, which is the region of interest. This model assumes the beam expands in a conical shape, disregarding factors like internal ambipolar electric fields [10]. However, this simplification is common in similar studies [16,20,25,27], as it has limited impact on the overall behavior of the beam.

For the ray-marching-based algorithm implementation, however, it is more interesting to discretize the force in terms of a differential surface element of the beam, not the target. To do this, observe Fig. 7, where a generic differential surface element of the target $\mathrm{d}S$ is shown along with a differential surface element of the beam $\mathrm{d}S'$. Note that, since the ions move radially away from the source, $\mathbf{U}$ and $\mathrm{d}S'$ are always perpendicular. Furthermore, since $\mathrm{d}S$ and $\mathrm{d}S'$ are arbitrarily small, they can be considered flat surfaces, and $\mathbf{U}$ is assumed constant throughout their span. Note that Fig. 7 is not a 2D simplification of the 3D case, it represents the actual 3D flat differential surfaces seen from a convenient point of view: perpendicular to the plane that contains the normal vectors of both $\mathrm{d}S$ and $\mathrm{d}S'$.

Looking at Fig. 7, it is not difficult to see that since $-\mathbf{N}\cdot\mathbf{U} = U\cos\gamma$, and $\cos\gamma\,\mathrm{d}S = \mathrm{d}S'$, the differential force can be expressed in terms of a differential surface element of the beam as

$$\mathrm{d}\mathbf{F}_B = m_I n_I \mathbf{U} U\,\mathrm{d}S'. \tag{17}$$

Now, spherical coordinates as in Fig. 8 are defined, where $O_C x_C y_C z_C$ represents the CLVLH frame and d_V is the distance from the COM of the chaser to the cone vertex of the ion beam. The value of α ranges between 0 and α_0, which is the angle of divergence of the beam, and $\rho \in [0,\infty)$, $\beta \in [0, 2\pi)$. In these coordinates, a differential surface element of the beam, the ion number density, and the ion velocity are given by

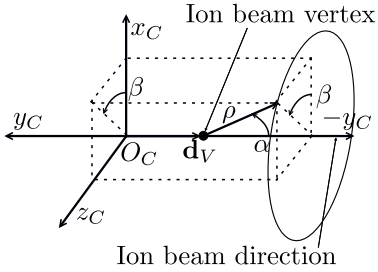

Fig. 8. Definition of spherical coordinates for the ion beam computation.

Table 1
Ion beam characteristics.

R_0 [m]	m_I [kg]	n_0 [m^{-3}]	U_0 [m/s]	α_0 [deg]
0.1894	$2.18\cdot10^{-25}$	$4.13\cdot10^{15}$	71,580	7

$$\mathrm{d}S' = \rho^2 \sin\alpha\,\mathrm{d}\alpha\mathrm{d}\beta, \tag{18}$$

$$n_I = \frac{R_0^2 n_0}{\rho^2 \tan^2\alpha_0 \cos^2\alpha} \exp\left(-C\frac{\tan^2\alpha}{2\tan^2\alpha_0}\right), \tag{19}$$

$$\mathbf{U} = \frac{U_0}{\cos\alpha}\hat{\boldsymbol{\rho}}, \tag{20}$$

where $\hat{\boldsymbol{\rho}}$ is the radial unit vector, R_0 is the radius at a reference plane of the plasma tube that contains 95% of the ion beam mass flow, n_0 is the density at the center of that reference plane, $C \approx 6$ is an integration constant related to that 95%, and U_0 is the ion velocity magnitude at the cone axis (see [16]). Therefore, the differential force can be computed as

$$\mathrm{d}\mathbf{F}_B = \frac{m_I n_0 R_0^2 U_0^2 \tan\alpha}{\tan^2\alpha_0 \cos^3\alpha} \exp\left(-C\frac{\tan^2\alpha}{2\tan^2\alpha_0}\right)\hat{\boldsymbol{\rho}}\,\mathrm{d}\alpha\mathrm{d}\beta. \tag{21}$$

Note that this expression does not depend on the distance between the ion beam impact point and the source point. However, it is still necessary to compute the distance for two reasons. The first is that, even if the distance itself does not appear, it must be known if the beam in that specific direction (given by α and β) intersects the target, because if it does not, then $\mathrm{d}\mathbf{F}_B$ is zero. The second reason is that even if the distance is not needed for the force computation, it is needed for the torque, and that is probably why algorithms in [16,17] do not compute it. For the actual computation, taking into account that the chosen ion thruster is Xenon-based, the same value as in [16,17] is used for the ion mass. The ion density at the center of the reference plane, the ion velocity magnitude at the cone axis, and the divergence angle are also taken from those studies. What is modified, then, is R_0, which is adapted so that the force that the ion beam exerts in an object when it impacts entirely is $F_{ITT} = 164.8$ mN (the value in Table 6). The characteristics of the beam are included in Table 1.

3.3. Ray-marching to compute the distance

For a cylinder of radius R_c and height H_c centered at the origin and with its symmetry axis along z, the distance $\rho(\alpha,\beta)$ from an arbitrary point $\mathbf{Q}^B$ in the target body axes to the cylinder can be computed with just one formula as

$$\rho = \left\| \begin{pmatrix} \max\left(\sqrt{Q_x^{B\,2} + Q_y^{B\,2}} - R_c, 0\right) \\ \max\left(|Q_z^B| - H_c/2, 0\right) \end{pmatrix} \right\|, \tag{22}$$

and, with the cylinder center in a certain position with respect to the CLVLH frame $\mathbf{r}^C_{OCyl}$ and an arbitrary attitude R^B_C, $\mathbf{Q}^B$ can be easily obtained as

$$\mathbf{Q}^B = -R^B_C\left(\mathbf{r}^C_{OCyl} - \mathbf{d}^C_V\right), \tag{23}$$

where $\mathbf{d}^C_V$ is the position vector of the ion beam cone vertex in the CLVLH frame. If the ion beam is always pointed towards $-y_C$, then $\mathbf{d}^C_V = [0, -d_V, 0]^T$.

3.4. Discretizing the beam

To discretize the beam, simple linear spacings in both α and β are taken. In Section 3.5 it is shown that $N_\alpha = 100$ and $N_\beta = 20$ points for α and β respectively are enough to approximate the beam force to an accuracy of around 0.15% of relative error, while still being fairly fast. For each α_i, β_j, one computes

$$\hat{\rho}^C_{ij} = \begin{pmatrix} \cos\beta_j \sin\alpha_i \\ -\cos\alpha_i \\ \sin\beta_j \sin\alpha_i \end{pmatrix}, \tag{24}$$

$$\Delta\mathbf{F}^C_{B,ij} = \frac{m_I n_0 R_0^2 U_0^2 \tan\alpha_i}{\tan\alpha_0^2 \cos^3\alpha_i} \exp\left(-C\frac{\tan^2\alpha_i}{2\tan^2\alpha_0}\right)\hat{\rho}^C_{ij}\Delta\alpha\Delta\beta b_{ij}, \tag{25}$$

where b_{ij} is a boolean variable that is 0 if the beam does not intersect the cylinder and 1 if it does. The force in the CLVLH frame, and torque in the TBA frame, which are the outputs of the algorithm, can be retrieved as

$$\mathbf{F}^C_B = \sum_{i=1}^{N_\alpha}\sum_{j=1}^{N_\beta}\Delta\mathbf{F}^C_{B,ij}, \tag{26}$$

$$\mathbf{M}^B_B = R^B_C\sum_{i=1}^{N_\alpha}\sum_{j=1}^{N_\beta}\mathbf{r}^C_{OCyl,P_{ij}} \times \Delta\mathbf{F}^C_{B,ij}, \tag{27}$$

where $\mathbf{r}^C_{OCyl,P_{ij}}$ is the vector that goes from the cylinder center to the impact point of the beam, and can be computed as

$$\mathbf{r}^C_{OCyl,P_{ij}} = \rho_{ij}\hat{\rho}^C_{ij} - \left(\mathbf{r}^C_{OCyl} - \mathbf{d}^C_V\right). \tag{28}$$

3.5. Sensitivity analysis and results validation

After applying the discretization to the model, it is left to choose the appropriate values of N_α and N_β. Logically, for $N_\alpha \to \infty, N_\beta \to \infty$, the discretized solution converges to the real one. However, as the number of points increases, so does the computation time, and, given that this function is called multiple times by the integrator, it is mandatory that it takes very few to execute. Because of this, even if the programming environment used for the development of this paper is MATLAB, a C++ MEX function has been programmed precisely for this computation of force and torque.

To choose N_α and N_β, a large value is first set for both $N_\alpha = N_\beta = 10{,}000$ and the force and torque are calculated with them, results that can be assumed to be very close to the real solution and thus are denoted $\mathbf{F}_{B,hp}$ and $\mathbf{M}_{B,hp}$, with the subscript hp denoting high precision. Then a series of values lower than 10,000 are tested for both N_α and N_β, and the force $\mathbf{F}_B$ and torque $\mathbf{M}_B$ are computed for each of them. Lastly, the relative error between each obtained result and the one corresponding to the high precision computation is calculated and shown in Table 2 for the force and Table 3 for the torque, to calculate the quality of the results. For these computations the relative position of the target with respect to the chaser in a chaser-centered LVLH frame is $\mathbf{r}^C_{OCyl} = [1, -d_{CT}, 0]^T$ (with $d_{CT} = 12$ m), and its attitude, described in Euler angles, is $\theta_1 = -45°, \theta_2 = -30°, \theta_3 = 0$ for the YXZ sequence. They are chosen so that the beam hits the target in a rather arbitrary attitude and just partially so that torque is generated (otherwise, torque relative errors would just be computer numeric noise), but the same trends are found in other relative positions and attitudes.

Table 2
Values of $100\frac{\|\mathbf{F}_B - \mathbf{F}_{B,hp}\|}{\|\mathbf{F}_{B,hp}\|}$ [%], where $\mathbf{F}_B$ is computed for each N_α, N_β, and $\mathbf{F}_{B,hp}$ is the value for $N_\alpha = N_\beta = 10000$.

$N_\alpha \backslash N_\beta$	10	20	50	100	1000	10,000
10	0.8638	0.8637	0.7932	0.7931	0.7614	0.7618
20	0.5330	0.5330	0.5426	0.5426	0.5319	0.5336
50	0.3418	0.2981	0.2621	0.2621	0.2648	0.2643
100	0.1955	0.1513	0.1394	0.1394	0.1358	0.1357
1000	0.0635	0.0213	0.0133	0.0133	0.0122	0.0124
10,000	0.0521	0.0097	0.0014	0.0010	0.0000	0

Table 3
Values of $100\frac{\|\mathbf{M}_B - \mathbf{M}_{B,hp}\|}{\|\mathbf{M}_{B,hp}\|}$ [%], where $\mathbf{M}_B$ is computed for each N_α, N_β, and $\mathbf{M}_{B,hp}$ is the value for $N_\alpha = N_\beta = 10000$.

$N_\alpha \backslash N_\beta$	10	20	50	100	1000	10,000
10	0.6860	0.7484	0.6410	0.6583	0.6112	0.6132
20	0.3350	0.4071	0.4377	0.4526	0.4368	0.4399
50	0.3450	0.2878	0.2309	0.2323	0.2377	0.2369
100	0.2049	0.1386	0.1244	0.1270	0.1212	0.1212
1000	0.1133	0.0234	0.0105	0.0123	0.0105	0.0109
10,000	0.1112	0.0205	0.0031	0.0022	0.0000	0

Table 4
Computation time [ms] computed for each N_α, N_β.

$N_\alpha \backslash N_\beta$	10	20	50	100	1000	10,000
10	0.049	0.058	0.124	0.233	2.005	18.43
20	0.054	0.079	0.190	0.373	3.691	34.53
50	0.110	0.191	0.460	0.882	8.871	85.78
100	0.202	0.375	0.904	1.776	17.01	168.7
1000	1.806	3.606	8.932	17.86	168.7	1659
10,000	17.61	36.04	87.61	173.2	1699	16,653

It can be seen that the model is relatively precise even for a small number of points, achieving a less than 1% error with only $N_\alpha = N_\beta = 10$. The computation times for each combination are included in Table 4. The calculations were performed using a Windows 11 laptop equipped with an Intel Core i7-12700H processor and 32 GB of RAM. Noting that the accuracy is much more sensitive to N_α than to N_β, and taking a look at the computational times, it seems reasonable to choose $N_\alpha = 100, N_\beta = 20$ as the values with which to perform the simulations.

Now, to validate the results of the model with $N_\alpha = 100, N_\beta = 20$, one can check the force it gives compared to the results obtained in [16] using both direct integration and central projection. For this task the cylinder is modified to have the same measurements, which are a radius of 1.1 m and a height of 2.6 m; and the beam characteristics are changed to coincide with those in [16]. The forces obtained for the three methods and the 15 cases they present (which are depicted in Fig. 9) are included in Table 5, concluding that the mean relative error is around 0.15%, as expected. However, this beam marching algorithm is also able to compute torque while still being efficient.

Note that in Table 5, cases 1, 4, 7, 10 and 13 give the same exact force result. This is no coincidence. With the hypotheses made in Section 3.2, in which the ion beam expands as a cone, and the force magnitude does not depend on the distance, the force can be computed by projecting the debris surface impacted by the ion beam on a certain perpendicular plane (that is the base of how they compute the force in [16]). Therefore, if the ion beam completely impacts the target, the projected shape is a circle regardless of the attitude, and the force is the same. Thus, as long as the ion beam completely impacts the target, the target's attitude does not change its acceleration. If the impact is partial, attitude will influence the result.

4. Control design

To control the relative position of the chaser with respect to the target, a Robust MPC for Tracking controller is designed, based on

Table 5
Comparison of force results obtained for both methods 1 and 2 in [16] vs using beam marching.

Case	Direct integration			Central projection			Beam marching		
	$F_{B,x}$ [N]	$F_{B,y}$ [N]	$F_{B,z}$ [N]	$F_{B,x}$ [N]	$F_{B,y}$ [N]	$F_{B,z}$ [N]	$F_{B,x}$ [N]	$F_{B,y}$ [N]	$F_{B,z}$ [N]
1	0.000	0.000	2.986e−2	0.000	0.000	2.975e−2	0.000	0.000	2.979e−2
2	0.000	3.431e−5	2.943e−2	0.000	3.459e−5	2.943e−2	0.000	3.451e−5	2.947e−2
3	0.000	5.332e−4	1.764e−2	0.000	5.327e−4	1.766e−2	0.000	5.336e−4	1.768e−2
4	0.000	0.000	2.974e−2	0.000	0.000	2.975e−2	0.000	0.000	2.979e−2
5	−7.700e−6	8.587e−5	2.888e−2	−7.487e−6	8.636e−5	2.887e−2	−7.681e−6	8.671e−5	2.891e−2
6	−7.496e−6	5.313e−4	1.834e−2	−7.120e−6	5.318e−4	1.833e−2	−3.458e−6	5.359e−4	1.825e−2
7	0.000	0.000	2.975e−2	0.000	0.000	2.975e−2	0.000	0.000	2.979e−2
8	6.939e−6	4.858e−6	2.967e−2	6.720e−6	5.464e−6	2.968e−2	7.354e−6	6.098e−6	2.971e−2
9	1.314e−4	1.352e−4	2.747e−2	1.303e−4	1.374e−4	2.745e−2	1.297e−4	1.368e−4	2.752e−2
10	0.000	0.000	2.975e−2	0.000	0.000	2.975e−2	0.000	0.000	2.979e−2
11	6.939e−6	4.858e−6	2.967e−2	6.720e−6	5.464e−6	2.968e−2	7.354e−6	6.098e−6	2.971e−2
12	1.314e−4	1.352e−4	2.747e−2	1.303e−4	1.374e−4	2.745e−2	1.297e−4	1.368e−4	2.752e−2
13	0.000	0.000	2.975e−2	0.000	0.000	2.975e−2	0.000	0.000	2.979e−2
14	1.311e−5	1.321e−5	2.959e−2	1.297e−5	1.297e−5	2.959e−2	1.243e−5	1.243e−5	2.964e−2
15	1.575e−4	2.259e−4	2.578e−2	1.566e−4	2.261e−4	2.578e−2	1.529e−4	2.281e−4	2.583e−2

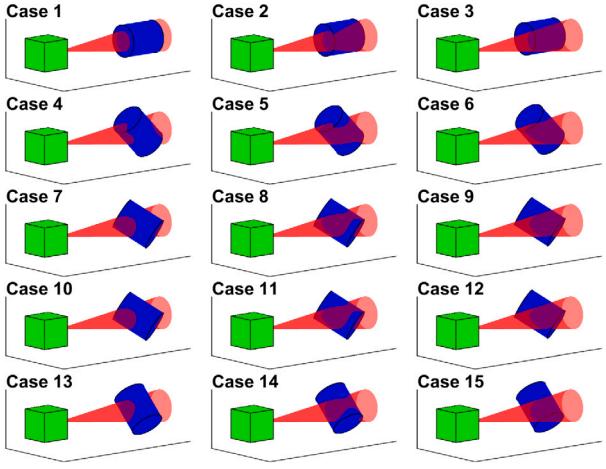

Fig. 9. The 15 different configurations in which the force results are compared to those in [16].

the formulation in [34]. As typical in MPC, the system model for the controller is a linear approximation of the real system. Errors due to these approximations and from other sources are bounded and fed to the controller to make it robust. In addition, one must take into account that the controller output is continuous and is applied to the system via a zero-order-hold (ZOH). However, the cold-gas thrusters operate in a binary on–off mode, regulating the time they are actuating. To convert from one type of actuation to the other, an on–off filter is also included. This section is organized as follows: the on–off filter is explained in Section 4.1, the linear model approximation of the system is presented in Section 4.2, the system constraints are introduced in Section 4.3, the set that bounds the disturbances is computed in Section 4.4, and finally, Section 4.5 includes the controller design.

4.1. On–off filter for the cold-gas thrusters

The control output as computed by the MPC lies in the reals and is assumed to be applied as a ZOH. Indeed, this is possible in the in-track direction, where the force of the ITT can be modulated as needed. However, one must remember that in the radial and out-of-plane directions the actuators operate in a binary on–off mode, which means they apply only either zero or maximum thrust, and are able to correctly actuate on the system by conveniently adjusting the times they are switched on and off. To convert the MPC control output into a feasible actuation, a centered equal-area filter is applied, so that the real control commanded to the actuators is

$$\tilde{u}_i(t) = \begin{cases} 0 & 0 \le t < T_i^{on} \\ u_{max,CG} \cdot \mathrm{sgn}(u_i) & T_i^{on} \le t \le T_i^{off} \\ 0 & T_i^{off} < t \le T_c, \end{cases} \tag{29}$$

where $T_i^{on} = \frac{T_c}{2}\left(1 - \frac{|u_i|}{u_{max,CG}}\right)$, $T_i^{off} = \frac{T_c}{2}\left(1 + \frac{|u_i|}{u_{max,CG}}\right)$; $i = \{x, z\}$, that is, this filter is applied only in the radial and out-of-plane directions; and $u_{max,CG} = F_{CG}/m_C^-$ is the maximum acceleration the cold-gas thrusters can produce at the control sampling time. The control is assumed to be recomputed every T_c seconds. Note that this filter centers the actuation in the middle of the control interval and adjusts the control so that "control areas" are preserved (that is, $\int_0^{T_c} \tilde{u}_i \mathrm{d}t = \int_0^{T_c} u_i$), which forces both controls to have a "similar" impact on the system.

There is one last feature of this filter. If the control action $|u_i|$ commanded by the MPC is so small that $T_i^{off} - T_i^{on} < \Delta T_{lim}$ (where ΔT_{lim} is a small value), the filter is ordered not to act at all. This behavior is added to deal with the fact that cold-gas thrusters have a minimum-width pulse that they can perform. More details on this are given in Section 4.4.1.

4.2. Linear model

Sensors on-board the chaser estimate its absolute state $\mathbf{r}_C^I, \mathbf{v}_C^I$ and the relative state of the target with respect to the chaser-centered LVLH frame $\mathbf{r}_T^C, \mathbf{v}_T^C$. The target is to be kept at a distance d_{CT} behind the chaser in the negative in-track direction, so $\mathbf{r}_{T,ref}^C = [0, -d_{CT}, 0]^T$, $\mathbf{v}_{T,ref}^C = [0,0,0]^T$. Therefore, for the controller, the virtual state $\mathbf{x} = (\mathbf{r}_V^C, \mathbf{v}_V^C)$, is defined, where $\mathbf{r}_V^C = \mathbf{r}_{T,ref}^C - \mathbf{r}_T^C, \mathbf{v}_V^C = \mathbf{v}_{T,ref}^C - \mathbf{v}_T^C$. The task of the controller is to regulate the virtual state $\mathbf{x}$ to the origin. The dynamics of the virtual state can be approximated by a modified version of the Hill–Clohessy–Wiltshire (HCW) equations:

$$\dot{\mathbf{x}} = A_c\mathbf{x} + B_c\mathbf{u} + \mathbf{w}_c, \tag{30}$$

where

$$A_c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 + c & 0 & 0 & 0 & 2n & 0 \\ 0 & -2c & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 + c & 0 & 0 & 0 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{31}$$

and $\mathbf{w}_c$ is an unknown but bounded state disturbance. In A_c, n is the mean motion of the chaser, which can be obtained from $\mathbf{r}_C^I, \mathbf{v}_C^I$, and c is a new term that accounts for the gradient effect the ion beam has on the target when the chaser and the target are not properly

aligned (see [20]). This term c is computed assuming that the target is perfectly spherical (the so-called cannonball model), which is logically not the case, but allows the controller to have some kind of information about the effect of the ion beam on the system, enhancing performance. Coming from a cylinder, setting the radius of the apparent sphere is not straight-forward. In this work, it is chosen to be the mean between the cylinder radius and its semiheight, $R_{as} = (R_c + H_c/2)/2$. This is a good compromise value for its simplicity and because, for cylindrical targets with height to radius ratios between 3.5 and 16.3, it lies between the equivalent sphere radius obtained by computing the mean frontal area of the cylinder ($\overline{R}_A = \sqrt{2R_c/\pi\left(R_c/\pi + 2H_c\right)}$), and the equivalent radius of a sphere with the same volume ($\overline{R}_V = \sqrt[3]{3R_c^2 H_c/4}$). Another way to compute this equivalent sphere radius may be convenient for other H_c/R_c proportions.

The system in (30), with $n_x = 6$ states and $n_u = 3$ control inputs, is an LTI continuous system, while the controller acts discretely. The discrete version of the system is

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{w}_k, \tag{32}$$

where, again, $\mathbf{w}_k$ is an unknown but bounded state disturbance. Assuming that the controller acts every T_c seconds, matrices A and B can be computed using the State Transition Matrix (STM) $\Phi(t)$ as

$$\dot{\Phi} = A_c\Phi, \qquad \Phi(0) = \mathbf{I}_6, \tag{33}$$

$$A = \Phi(T_c), \tag{34}$$

$$B = \int_0^{T_c} \Phi(\tau)\mathrm{d}\tau\, B_c, \tag{35}$$

where $\mathbf{I}_6$ is the 6×6 identity matrix. Although there are simple analytic solutions for A, B in the classical HCW model, these solutions become much more cumbersome with the addition of the c term, and their numerical calculation becomes a better choice. However, (35) is not an efficient way of computing B, since it performs a quadrature of a function that performs a numerical integration each time it is evaluated. Instead, differentiating (35), applying the Fundamental Theorem of Calculus, and reordering the equations, one obtains the next system of Ordinary Differential Equations (ODEs):

$$\dot{\Phi} = A_c\Phi, \qquad \Phi(0) = \mathbf{I}_6, \tag{36}$$

$$\dot{\Pi} = A_c\Pi, \qquad \Pi(0) = \mathbf{0}_{6\times 3}, \tag{37}$$

$$\dot{\Psi} = \Pi, \qquad \Psi(0) = B_c, \tag{38}$$

in which $\Phi(T_c) = A$, $\Psi(T_c) = B$. Note that, since the system is LTI, these matrices are calculated once and then stored; they do not need to be recomputed at each sampling time, except when the controller is re-tuned (since the chaser and target orbits slowly change, after some time the mean motion n must be updated and the controller re-tuned).

4.3. System constraints

In accordance with the MPC formulation, the constraints that the system must meet at each time instant are expressed as polytopes. In the case of the state, a certain polytope that ensures full impact of the ion beam is defined for position, and maximum values are set for velocity. For the control, there are maximum values that the actuators can input into the system.

4.3.1. State constraints

In terms of position, the chaser should not deviate significantly from the virtual reference, since that would imply only partially hitting the target (or potentially not hitting it at all), which in turn means a longer mission time and more fuel consumption, since the ion thrusters would be more time activated. Note, however, that it may not be convenient to impose very strict constraints in position either. The chaser has some clearance to move while still completely hitting the target (or, more precisely, the cone that represents the 95% of the ion beam contribution completely hits the target). So, if the chaser is in that vicinity, it can be left unactuated while not draining mission performance; quite the contrary, it would increase efficiency because less fuel would be used.

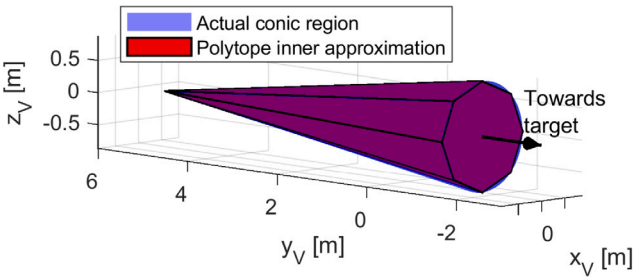

Fig. 10. Visualization of the position constraints polytope.

To compute the clearance region, it is assumed that within that region, the ion beam completely hits the target regardless of the target's attitude. This is equivalent to saying that the ion beam must hit the sphere inscribed in the cylinder, that is, a sphere with radius R_c. With a divergence angle α_0, recalling that in the nominal relative position the COMs of the target and chaser are separated by d_{CT}, and that the distance between the chaser's COM and the ion beam source is d_V, the maximum displacement in the in-track direction while still entirely hitting the target can be computed to be

$$\Delta y_{max} = \frac{R_c}{\sin\alpha_0} - \left(d_{CT} - d_V\right). \tag{39}$$

From there, if one moves towards the negative in-track direction with an angle less than α_0, the target would still be completely hit, so this region is actually a cone. Another limit must be imposed on the other side, since the chaser should not get too close to the target. Assuming that in the worst attitude case the ion beam source should be at $d_{min} = 3$ m of the target's closest point and not less, the maximum movement in the negative in-track direction is

$$\Delta y_{min} = -\left(d_{CT} - d_V - \sqrt{R_c^2 + \left(\frac{H_c}{2}\right)^2} - d_{min}\right). \tag{40}$$

For the simulation case studied in Section 5, this truncated cone is shown in Fig. 10, along with an inner approximation made with an eight-plane polytope, which is the actual region input into the controller. To build the inner approximation, the next approach was followed:

- The cone vertex is denoted as $P_0 = \{0, \Delta y_{max}, 0\}$.
- Eight points are defined at the minimum y distance, inscribing an octagon inside the cone intersection circle: $P_i = \{(\Delta y_{max} - \Delta y_{min})\tan\alpha_0 \cos\beta_i, \Delta y_{min}, (\Delta y_{max} - \Delta y_{min})\tan\alpha_0 \sin\beta_i\}$, where $\beta_i = 0, \frac{\pi}{4}, \frac{\pi}{2}, \ldots, \frac{7\pi}{4}$ for $i = 1, \ldots, 8$, and $P_9 := P_1$.
- Eight planes are defined with the three-point definition: $P_0P_iP_{i+1}$, for $i = 1, \ldots, 8$.
- Eight plane inequalities are set using those planes, such that all of them contain the origin.
- The last inequality is given by $y > \Delta y_{min}$.

Position constraints are therefore given by $A_r\mathbf{r} \le \mathbf{b}_r$, where A_r and b_r are defined according to the polytope in Fig. 10.

In terms of velocity, a maximum velocity $v_{max} = 1$ m/min is imposed, to increase safety in operations. Thus, velocity restrictions are expressed as $A_v\mathbf{x} \le \mathbf{b}_v$, where

$$A_v = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & -\mathbf{I}_3 \end{bmatrix}, \qquad b_v = v_{max}\mathbf{1}_{6\times 1}. \tag{41}$$

Therefore, the state admissible set is

$$\mathcal{X} = \left\{ \mathbf{x} \in \mathbb{R}^6 : A_x \mathbf{x} \leq \mathbf{b}_x \right\}, \tag{42}$$

where $A_x = \begin{bmatrix} A_r \\ A_v \end{bmatrix}$, $\mathbf{b}_x = \begin{bmatrix} \mathbf{b}_r \\ \mathbf{b}_v \end{bmatrix}$.

4.3.2. Control constraints

Actuators have a maximum force that they can provide, which is F_{CG} and $F_{ICT,max}$ for the cold-gas thruster and ICT respectively. This force is transformed into acceleration (the actual control input in (32)) by dividing it by the chaser mass. Thus, in the radial and out-of-plane directions, the maximum acceleration is given by

$$u_{max,CG} = \frac{F_{CG}}{\overline{m}_C}, \tag{43}$$

where $\overline{m}_C$ denotes the mass of the chaser at the tuning of the controller. Note that there is no constraint on the minimum value of $|u_x|$, $|u_z|$, since the filter introduced in Section 4.1 already handles these cases. The discrepancy in the state due to the difference between what the controller commands and what the cold-gas thrusters perform is deduced and bounded in Section 4.4.1.

However, in the in-track direction, it is a little different. Recalling (14), one can obtain the maximum control value $u_{max,ICT}$ (that is, the maximum acceleration in the positive in-track direction) by setting F_{ICT} to zero, and the minimum control value $u_{min,ICT}$ (the maximum acceleration in the negative in-track direction) by setting $F_{ICT} = F_{ICT,max}$, thus obtaining

$$u_{max,ICT} = F_{ITT}\left(\frac{1}{m_T} + \frac{1}{\overline{m}_C}\right), \tag{44}$$

$$u_{min,ICT} = -\frac{F_{ICT,max} - F_{ITT}}{\overline{m}_C} + \frac{F_{ITT}}{m_T}. \tag{45}$$

It is appropriate to note that the force F_{ITT} that affects the chaser does not depend on the ion beam impacting the target or not. Rather, m_T appears because it determines the acceleration of the target, and recall that, as explained in Section 2.1.1, in the equilibrium configuration the acceleration of the chaser must be equal to that of the target. The control admissible set ends up being

$$\mathcal{U} = \left\{ \mathbf{u} \in \mathbb{R}^3, : A_u \mathbf{u} \leq \mathbf{b}_u \right\}, \tag{46}$$

where $A_u = \begin{bmatrix} \mathbf{I}_3 \\ -\mathbf{I}_3 \end{bmatrix}$, $\mathbf{b}_u = \left[u_{max,CG},\ u_{max,ICT},\ u_{max,CG},\ u_{max,CG},\ -u_{min,ICT},\ u_{max,CG}\right]^T$.

Note that mass $\overline{m}_C$ was defined as the mass that the chaser has when the tuning of the controller is performed, even if mass actually decreases during operations. The controller has been implemented in this way because otherwise the admissible sets would change continuously and a continuous re-tuning of the controller (which, of course, is not viable) would be needed. Note, however, that these limits are a conservative estimation of the actual limits of the actuators, since when m_C decreases, $u_{max,CG}$, $u_{max,ICT}$ increase and $u_{min,ICT}$ decreases, so this approach may have an impact on the performance of the controller (which has been tested to be virtually nonexistent for the chosen re-tuning frequency), but not on its feasibility.

4.4. Disturbances bounding

There are certain behaviors of the real system that the linear model does not include. One example of them is the presence of the on–off filter, which of course modifies the trajectory of the body with respect to what the controller expects. Another such example is unmodeled dynamics, such as J2 and drag perturbations, or the linearization of the equations itself. While these cannot be precisely included in the linear model, they can be bounded into an uncertainty polytope $\mathcal{W}$. If that uncertainty polytope is small enough, then the MPC controller is able to compute a non-empty region in which it can control the system no matter the value of the disturbances $\mathbf{w}$ provided that they are contained in $\mathcal{W}$ (more details in Section 4.5). This section is devoted to the bounding of such disturbances.

4.4.1. Bounding the on–off filter mismatch

Taking a look at (31), it is clearly seen that the in-plane dynamics (in the xy plane) and the out-of-plane dynamics are decoupled. This fact is also true for the equivalent discrete system given by A, B. It is therefore convenient, for simplicity, to decouple the dynamics into in-plane dynamics, governed by A^{xy}, B^{xy}; and out-of-plane dynamics, governed by A^z, B^z. The matrix A^{xy} is constructed by taking rows and columns 1,2,4,5 of A, while B^{xy} is built taking rows 1,2,4,5 and columns 1,2 of B. Similarly, to build A^z, take rows and columns 3,6 of A, and for B^z take rows 3,6 and column 3 of B. Moreover, B_x^{xy} denotes the first column of B^{xy}, and B_y^{xy} the second. Denoting the commanded control by the MPC as $\mathbf{u} = \left[u_x, u_y, u_z\right]^T$, the evolution of the linear system, divided by in-plane and out-of-plane dynamics, if there was no on–off filter is described by

$$\mathbf{x}_{k+1}^{xy} = A^{xy}\left(T_c\right)\mathbf{x}_k^{xy} + B_x^{xy}\left(T_c\right)u_x + B_y^{xy}\left(T_c\right)u_y, \tag{47}$$

$$\mathbf{x}_{k+1}^{z} = A^{z}\left(T_c\right)\mathbf{x}_k^{z} + B^{z}\left(T_c\right)u_z, \tag{48}$$

where $\mathbf{x}^{xy} = \left[x, y, v_x, v_y\right]^T$, $\mathbf{x}^z = \left[z, v_z\right]^T$, and the T_c inside the parenthesis means that equations in (36)–(38) are integrated from 0 to T_c. There is, however, an on–off filter that alters this behavior, and it can be shown that, in terms of matrices A and B, the response of the system is then

$$\begin{aligned}\tilde{\mathbf{x}}_{k+1}^{xy} = {} & A^{xy}\left(T_c\right)\mathbf{x}_k^{xy} + A^{xy}\left(\frac{T_c}{2}\left(1 - \frac{|u_x|}{u_{max,CG}}\right)\right) \\ & \times B_x^{xy}\left(T_c\frac{|u_x|}{u_{max,CG}}\right)u_{max,CG}\,\mathrm{sgn}\left(u_x\right) + B_y^{xy}(T_c)u_y,\end{aligned} \tag{49}$$

$$\begin{aligned}\tilde{\mathbf{x}}_{k+1}^{z} = {} & A^{z}\left(T_c\right)\mathbf{x}_k^{z} + A^{z}\left(\frac{T_c}{2}\left(1 - \frac{|u_z|}{u_{max,CG}}\right)\right) \\ & \times B^{z}\left(T_c\frac{|u_z|}{u_{max,CG}}\right)u_{max,CG}\,\mathrm{sgn}\left(u_z\right).\end{aligned} \tag{50}$$

To bound the error due to the on–off filter, the corresponding states are now subtracted

$$\begin{aligned}\Delta\mathbf{x}^{xy} = {} & B_x^{xy}\left(T_c\right)u_x - A^{xy}\left(\frac{T_c}{2}\left(1 - \frac{|u_x|}{u_{max,CG}}\right)\right) \\ & \times B_x^{xy}\left(T_c\frac{|u_x|}{u_{max,CG}}\right)u_{max,CG}\,\mathrm{sgn}\left(u_x\right),\end{aligned} \tag{51}$$

$$\begin{aligned}\Delta\mathbf{x}^{z} = {} & B^{z}\left(T_c\right)u_z - A^{z}\left(\frac{T_c}{2}\left(1 - \frac{|u_z|}{u_{max,CG}}\right)\right) \\ & \times B^{z}\left(T_c\frac{|u_z|}{u_{max,CG}}\right)u_{max,CG}\,\mathrm{sgn}\left(u_z\right),\end{aligned} \tag{52}$$

where $\Delta\mathbf{x}^{xy} = \mathbf{x}^{xy} - \tilde{\mathbf{x}}^{xy}$ and $\Delta\mathbf{x}^{z} = \mathbf{x}^{z} - \tilde{\mathbf{x}}^{z}$, where the subscripts k and $k+1$ are omitted since the value of the state errors is not dependent on the prior state. Note that these state errors are odd with respect to u_x and u_z respectively. Thus, they are guaranteed to be bounded by $-\Delta\mathbf{x}_{max}^{xy} \leq \Delta\mathbf{x}^{xy} \leq \Delta\mathbf{x}_{max}^{xy}$ and $-\Delta\mathbf{x}_{max}^{z} \leq \Delta\mathbf{x}^{z} \leq \Delta\mathbf{x}_{max}^{z}$, in which each element of $\Delta\mathbf{x}_{max}^{xy}$ and $\Delta\mathbf{x}_{max}^{z}$ is obtained as:

$$\Delta\mathbf{x}_{max,i}^{xy} = \left|\max_{u_x\in[0,u_{max,CG}]}\Delta\mathbf{x}_i^{xy}\left(u_x\right)\right|, \qquad i = 1,2,3,4, \tag{53}$$

$$\Delta\mathbf{x}_{max,i}^{z} = \left|\max_{u_z\in[0,u_{max,CG}]}\Delta\mathbf{x}_i^{z}\left(u_z\right)\right|, \qquad i = 1,2, \tag{54}$$

where the subscript i denotes the ith row of the corresponding vector. Finally, concatenating the previous vectors as $\Delta\mathbf{x}_{fil,max} = \left[\Delta\mathbf{x}_{max,1}^{xy}, \Delta\mathbf{x}_{max,2}^{xy}, \Delta\mathbf{x}_{max,1}^{z}, \Delta\mathbf{x}_{max,3}^{xy}, \Delta\mathbf{x}_{max,4}^{xy}, \Delta\mathbf{x}_{max,2}^{z}\right]^T$, one has

$$-\Delta\mathbf{x}_{fil,max} \leq \Delta\mathbf{x} \leq \Delta\mathbf{x}_{fil,max}, \qquad \forall\mathbf{u} \in \mathcal{U}. \tag{55}$$

There is one more aspect that should be taken into account. Until now, no limitation has been imposed on how small the difference in the activation and deactivation instants $T_i^{on} - T_i^{off}$ of the cold-gas thrusters can be. However, in reality, there is a minimum duration due

to the transient behavior of the actuator, which is around 10 ms in state-of-the-art cold-gas thrusters. This could mean that, if the MPC commanded a control input u_i very close to zero, the actuator may not be able to perform it. Fortunately, this does not pose a problem in this formulation. In fact, imagine that the MPC does not command control in the radial direction, then $\Delta\mathbf{x}^{xy}\left(u_x=0\right)=\mathbf{x}^{xy}(0)-\tilde{\mathbf{x}}^{xy}(0)=0$, which is obviously within the limits of $\Delta\mathbf{x}^{xy}_{max}$. Assume now that a small control is commanded by the MPC but not performed by the actuator; the state error is thus

$$\Delta\mathbf{x}^{xy}\left(u_x\right)=\mathbf{x}^{xy}\left(u_x\right)-\tilde{\mathbf{x}}^{xy}(0)=B_x^{xy}\left(T_c\right)u_x. \tag{56}$$

From this equation, it is clear that there must be some maximum limit value $u_{x,lim}>0$ for which $B_x^{xy}\left(T_c\right)u_x$ is still bounded by $\Delta\mathbf{x}^{xy}_{max}$. Therefore, if the control commanded by the MPC is less than $u_{x,lim}$, the cold-gas thruster can simply not act, and the error would still be bounded by the same previously computed polytope. Similar reasoning can be used in the out-of-plane dynamics to obtain a value for $u_{z,lim}$. For this specific problem, this ends up meaning that when less than 0.15 s are commanded to the cold-gas thrusters, they do not need to act, since the controller is able to handle the mismatch. Since this value is much higher than the minimum possible impulse width in state-of-the-art cold-gas thrusters, this limitation does not pose any problems.

4.4.2. Bounding J2 perturbation

This perturbation, although small, given sufficient time, can have a noticeable effect on the orbit. Note, however, that since the chaser and target are really close, the acceleration due to J2 (which does not depend on the properties of the objects) is expected to be practically the same for both. To demonstrate it, one can compute the gradient of the perturbation. Indeed, let $\mathbf{r}_C^I,\mathbf{r}_T^I$ be the position of the chaser and target in the ECI frame respectively, and $\Delta\mathbf{r}^I=\mathbf{r}_C^I-\mathbf{r}_T^I$ the relative position. If $\Delta\mathbf{r}^I\ll\mathbf{r}_T^I$, then

$$\mathbf{a}_{J2}\left(\mathbf{r}_T^I+\Delta\mathbf{r}^I\right)\approx\mathbf{a}_{J2}\left(\mathbf{r}_T^I\right)+\nabla\mathbf{a}_{J2}\big|_{\mathbf{r}_T^I}\Delta\mathbf{r}^I, \tag{57}$$

where $\nabla\mathbf{a}_{J2}\big|_{\mathbf{r}}$ is the gradient matrix of $\mathbf{a}_{J2}$ evaluated at a generic point $\mathbf{r}$. Therefore, the difference in the accelerations due to the chaser and target amounts to

$$\Delta\mathbf{a}^I_{J2}=\mathbf{a}_{J2}\left(\mathbf{r}_C^I\right)-\mathbf{a}_{J2}\left(\mathbf{r}_T^I\right)\approx\nabla\mathbf{a}_{J2}\big|_{\mathbf{r}_T^I}\Delta\mathbf{r}^I. \tag{58}$$

The gradient matrix can be computed as $\nabla\mathbf{a}_{J2}\big|_{\mathbf{r}}=\frac{3}{2}\mu_E J_2 R_E^2\left(5M_1(\mathbf{r})-M_2(\mathbf{r})\right)$, where

$$M_1(\mathbf{r})=\frac{1}{r^9}\begin{bmatrix}z^2\left(r^2-7x^2\right) & -7xyz^2 & xz\left(2r^2-7z^2\right)\\ -7xyz^2 & z^2\left(r^2-7y^2\right) & yz\left(2r^2-7z^2\right)\\ -7xz^3 & -7yz^3 & z^2\left(3r^2-7z^2\right)\end{bmatrix} \tag{59}$$

$$M_2(\mathbf{r})=\frac{1}{r^7}\begin{bmatrix}r^2-5x^2 & -5xy & -5xz\\ -5xy & r^2-5y^2 & -5yz\\ -15xz & -15yz & 3r^2-15z^2\end{bmatrix}. \tag{60}$$

Taking into account that J2 is more notable closer to Earth, and that one seeks to find an upper bound of the error, assume that the target is orbiting at an Earth radius distance, for example, $\mathbf{r}_T^I=R_E\frac{1}{\sqrt{3}}[1,1,1]^T$, and that the separation between chaser and target is $\Delta\mathbf{r}^I=[20,20,20]^T$ m (these values will actually always be smaller). Then, the norm of the relative acceleration obtained is $\Delta a_{J2,max}\approx 3.3\cdot10^{-7}$ m/s^2. Assuming that this acceleration difference remains constant between control intervals and that the control interval is around 120 s, it generates a velocity error of $\Delta v_{J2,max}\approx\Delta a_{J2,max}T_c\approx 4\cdot10^{-5}$ m/s and a position error of $\Delta r_{J2,max}\approx\Delta a_{J2,max}T_c^2/2\approx 2\cdot10^{-3}$ m. Thus, the state error in J2 is bounded by those values.

4.4.3. Bounding drag perturbation

For the drag perturbation, a similar procedure is taken. However, analytical expressions of the gradient are now much more cumbersome and numerical methods are preferred. Assuming that the target cylinder mass is $m_T=9000$ kg with a height and radius of $H_c=12$ m and $R_c=2$ m respectively, its area–mass ratio can change between $(A_f/m)_T\in[1.4,5.4]\cdot10^{-3}$ m^2/kg, depending on whether the frontal area is minimum (cylinder axis and velocity vector aligned) or maximum (cylinder axis and velocity vector perpendicular). The area–mass ratio of the chaser, assuming its frontal area is of 4 m^2, is around $(A_f/m)_C\in[8,11]\cdot10^{-3}$ m^2/kg, since it increases when it consumes fuel. For this calculation, a maximum limit of 140 kg of fuel consumption was considered. Therefore, to have the maximum possible difference, the values of $(A_f/m)_T=1.4\cdot10^{-3}$ m^2/kg and $(A_f/m)_C=11\cdot10^{-3}$ m^2/kg are chosen.

For these calculations, the altitude of the orbit is assumed to be 340 km. This is a worst-case scenario, since at that point the mission is considered finished, but at the same time is the point where density and velocity are maximum. The drag coefficient is set to a typical value of 2.2 for both the chaser and target. After performing some calculations assuming a maximum relative distance between the chaser and the target of $\Delta\mathbf{r}^I=[20,20,20]^T$ (as done for the J2 perturbation), the maximum acceleration difference obtained due to drag is $\Delta a_{D,max}\approx 3\cdot10^{-6}$ m/s^2, meaning that errors in relative velocity and position are $\Delta v_{D,max}\approx\Delta a_{D,max}T_c\approx 4\cdot10^{-4}$ m/s, $\Delta r_{D,max}\approx\Delta a_{D,max}T_c^2/2\approx 2\cdot10^{-2}$ m; values considerably higher compared with those obtained for the J2 perturbation.

4.4.4. Bounding position and velocity sensoring errors

The measurement error committed by the system that estimates the relative state of the target must be taken into account in the robust formulation of the controller. Thus, since, as noted in Section 2.4, this error is as high as 0.05 m in position and 0.06 m/min in velocity, the values $\Delta r_{s,max}=0.05$ m, $\Delta v_{s,max}=0.06$ m/min are added to the uncertainty polytope, where s stands for sensors.

4.4.5. Bounding linearization and mismatch errors

There are more subtle errors that arise in the linear model that the controller has of the system. The main action of linearizing introduces an approximation and, moreover, the controller assumes the mean motion parameter n remains constant during operation, and is only updated when the controller is re-tuned, as remarked in Section 4.2. This means that the re-tuning policy has an impact on this error, which has been estimated to be small, and thus the controller has been configured to be re-tuned each 100 km of mean altitude (altitude of an equivalent circular orbit with the same semi-major axis). Assuming that the target starts at an altitude of 840 km, the controller should therefore be re-tuned at the mean altitudes of 740, 640, 540 and 440 kilometers.

If the mean motion n varied only depending explicitly on time (and not control), it would be a time-varying parameter, and specific bounding techniques for Linear Time-Varying systems could be employed. However, here n is affected by past control actions, so this approach is not possible, and a numeric non-explicit bounding strategy is implemented instead. Both chaser and target orbits are propagated during a control time interval using two different models. First, the restricted two-body problem equations (equations in (1), (2) with no perturbations, since they have already been bounded), and second, the linear model with the "worst possible" mean motion n. That is, if satellites start from a height of 740 km, use an n corresponding to 840 km, and so on. In addition, eccentricity is considered to be 0.01 (for both the chaser and target) for the propagation of the restricted two-body problem equations to account for the possibly not perfectly circular orbits. The results given by these two models are then compared, and the differences are bounded. Then the maximum differences are retrieved. This results approximately in

$$\Delta r_{lin,max}\approx 9\cdot10^{-3}\text{ m},\qquad \Delta v_{lin,max}\approx 3\cdot10^{-5}\text{ m/s}. \tag{61}$$

There is another error due to the mismatch that could be considered. As stated in 4.1, the controller model predicts that the acceleration

produced by the cold-gas thrusters is given by the cold-gas thruster force divided by the chaser mass at the control sampling time, F_{CG}/m_C^-. However, when the actions are commanded, the mass of the chaser is actually a bit smaller since it is applied later and it has already burned fuel. There is therefore a discrepancy in the acceleration. Estimating that the spacecraft burns a mean amount of around 0.01 kg/orbit, the discrepancy in acceleration is on the order of 10^{-10} m/s^2, so, even if it exists, it is not taken into account due to its negligible impact.

4.4.6. Bounding discrepancies in the beam model

As remarked in Section 4.2, a linear term c is added to the model so that the controller has some kind of knowledge about how the off-centered beam affects the movement of the target. Logically, this model is not the same as the one used in simulation and explained in Section 3. Some discrepancy error exists between them and must be bounded. However, note that since one of the constraints that the controller ensures is that the chaser stays within the region $\mathcal{X}$ (see Section 4.3.1), the beam completely impacts the target. Therefore, there is no discrepancy in the force estimation, since the force exerted towards the target is just the maximum, pointing towards the negative in-track direction, as noted in Section 3.5. There would actually be a mismatch in terms of the torque produced on the target, but it is not relevant in this case. It is important to remark that this force difference vanishes because, as a consequence of the hypotheses stated in Section 3.2, the force does not depend on the distance. In a real scenario, attitude may influence the force even if the ion beam fully hits the target, but this effect would nevertheless be small because it arises from higher-order terms. More in-depth analysis of the discrepancies between the ion beam model used by the controller and a real ion beam are left for future work, as higher-fidelity models would need to be implemented to correctly test the results.

Thus, the relevant aspect regarding ion beam discrepancies is the noise of the beam itself. As noted in Section 2.4, this error can be as high as 5% of the nominal thrust, which in the case of the ICT corresponds to 8.24 mN. This causes an acceleration perturbation of $\approx 10^{-6}$ m/s^2, which corresponds to

$$\Delta r_{B,max} \approx 6 \cdot 10^{-3} \text{ m}, \qquad \Delta v_{B,max} \approx 10^{-4} \text{ m/s}, \tag{62}$$

where the subscript B stands for beam.

4.5. Formulation of the MPC controller

An MPC for Tracking controller that guarantees robust constraint satisfaction is developed in this section, combining the constraints tightening techniques in [32] for robustness along with the tracking formulation in [33]. With this approach, the controller computes the control action by solving a unique Quadratic Programming (QP) problem, which can be efficiently done using specialized algorithms.

Ideally, the MPC must be able to reach any feasible equilibrium reference state $(\mathbf{x}_r, \mathbf{u}_r)$ (it is supposed that all state variables can be measured). However, in a conventional MPC formulation, the QP feasibility is not guaranteed for any given reference $\mathbf{x}_r$, to be reached in N steps and departing from an arbitrary initial condition $\mathbf{x}_0$. To avoid this, as suggested in [33], the tracking formulation adds an artificial reference $(\overline{\mathbf{x}}_s, \overline{\mathbf{u}}_s)$ to be followed, and deviations between this artificial and the real reference are penalized in the cost function, so that the artificial ends up converging to the real. In this way, it is ensured that feasibility is not lost despite changes in the reference. In this precise problem, there are no reference changes, but this implementation is beneficial regardless. Indeed, it expands the feasibility region of the controller, since a certain initial $\mathbf{x}_0$ may seem infeasible to a conventional MPC controller, but can be steered to the correct reference $(\mathbf{x}_r, \mathbf{u}_r)$ by gradually modifying the artificial reference each sampling time.

To reduce the number of decision variables, as in [33], the artificial reference is characterized by $\overline{\boldsymbol{\theta}}$ (of size n_θ), and the state and control inputs are computed as $\left[\overline{\mathbf{x}}_s^T, \overline{\mathbf{u}}_s^T\right]^T = M_\theta \overline{\boldsymbol{\theta}}$, where $M_\theta = \ker\left(\left[A - \mathbf{I}_6, B\right]\right)$. The performance index is then identical to that in [33] (particularized to the case in which $\mathbf{x}_r = \mathbf{u}_r = 0$), where penalization is given to discrepancies between the actual states and inputs and the ones of the artificial reference, and discrepancies between the artificial reference and the real reference are also penalized. The cost function therefore is

$$V_N\left(\mathbf{x}; \overline{\mathbf{u}}_F, \overline{\boldsymbol{\theta}}\right) = \sum_{i=0}^{N-1} \left(\left\|\overline{\mathbf{x}}(i) - \overline{\mathbf{x}}_s\right\|_Q^2 + \left\|\overline{\mathbf{u}}(i) - \overline{\mathbf{u}}_s\right\|_R^2 \right) + \left\|\overline{\mathbf{x}}(N) - \overline{\mathbf{x}}_s\right\|_P^2 + \left\|\overline{\mathbf{x}}_s\right\|_T^2, \tag{63}$$

where $\overline{\mathbf{x}}(i+1) = A\overline{\mathbf{x}}(i) + B\overline{\mathbf{u}}(i)$ is the nominal system evolution (that is, ignoring disturbances), $\overline{\mathbf{u}}_F = \left[\overline{\mathbf{u}}(0)^T, \overline{\mathbf{u}}(1)^T, \dots \overline{\mathbf{u}}(N-1)^T\right]^T$ are the control input vectors at each sampling time concatenated, and $\|\mathbf{v}\|_M^2 = \mathbf{v}^T M \mathbf{v}$ is the weighted euclidean norm. The current state $\mathbf{x}$ is a parameter, while the sequence of control inputs $\overline{\mathbf{u}}_F$ and the artificial reference $\overline{\boldsymbol{\theta}}$ are decision variables. The optimization problem $\mathcal{P}_N(\mathbf{x})$ that the MPC must be able to solve, taking now into account the effect of the disturbances, is

$$\min_{\overline{\mathbf{u}}_F, \overline{\boldsymbol{\theta}}} \quad V_N\left(\mathbf{x}; \overline{\mathbf{u}}_F, \overline{\boldsymbol{\theta}}\right) \tag{64}$$

$$s.t. \quad \overline{\mathbf{x}}(0) = \mathbf{x}, \tag{65}$$

$$\overline{\mathbf{x}}(i+1) = A\overline{\mathbf{x}}(i) + B\overline{\mathbf{u}}(i), \qquad i = 0, \dots, N-1 \tag{66}$$

$$\begin{bmatrix} \overline{\mathbf{x}}_s \\ \overline{\mathbf{u}}_s \end{bmatrix} = M_\theta \overline{\boldsymbol{\theta}}, \tag{67}$$

$$\overline{\mathbf{x}}(i) \in \mathcal{X} \ominus \mathcal{H}(i), \qquad i = 1, \dots, N-1 \tag{68}$$

$$\overline{\mathbf{u}}(i) \in \mathcal{U} \ominus K_c \mathcal{H}(i), \qquad i = 0, \dots, N-1 \tag{69}$$

$$\begin{bmatrix} \overline{\mathbf{x}}(N) \\ \overline{\boldsymbol{\theta}} \end{bmatrix} \in \Omega_a \ominus \mathcal{L}_a(N), \tag{70}$$

where $\mathcal{X}, \mathcal{U}$ are the state and control admissible sets (recall Section 4.3) and Ω_a is the robust positive invariant set for tracking of the system given a terminal linear feedback controller $\overline{\mathbf{u}} = K_t\left(\overline{\mathbf{x}} - \overline{\mathbf{x}}_s\right) + \overline{\mathbf{u}}_s = K_t\overline{\mathbf{x}} + L\overline{\boldsymbol{\theta}}$ (where $L = \left[-K_t, \mathbf{I}_{n_\theta}\right] M_\theta$). The set $\mathcal{L}_a(i)$ is the set of $\mathcal{L}(i)$ extended to the tracking formulation (so that the projection of $\mathcal{L}_a(i)$ over the $\overline{\boldsymbol{\theta}}$ space is the null set), and the sets $\mathcal{H}(i), \mathcal{L}(i)$ are defined as

$$\mathcal{H}(i) = \bigoplus_{j=0}^{i-1} A_{K_c}^j \mathcal{W} \tag{71}$$

$$\mathcal{L}(i) = A_{K_c}^{i-1} \mathcal{W}, \tag{72}$$

where $A_{K_c} = A + BK_c$ is the matrix that governs the response of the nominal system when a linear feedback controller $\mathbf{u} = K_c(\mathbf{x} - \overline{\mathbf{x}}) + \overline{\mathbf{u}}$ is used. The operators $\oplus$ and $\ominus$ represent the Minkowski sum and the Pontryagin set difference, respectively, while $\bigoplus$ is the equivalent of summation for sets applying the Minkowski sum.

First, note that the tightened constraints imposed in (68) and (69) ensure that the state and control are feasible at every time instant for any realization of the disturbances $\mathbf{w}(i) \in \mathcal{W}$. Note also that these reduced constraints depend on the gain K_c of the terminal linear controller, and that it can be chosen to reduce the sets $\mathcal{H}(i)$, which is desirable since it corresponds to an enlargement of the tightened sets. The calculation of K_c and the reduced sets $\mathcal{X} \ominus \mathcal{H}(i)$, $\mathcal{U} \ominus K_c\mathcal{H}(i)$ has been carried out following the methods in [32], involving Linear Matrix Inequalities (LMIs).

Note also that the gain K_c of the linear controller used to tighten the constraints is different from the gain K_t used for the computation of the robust positive invariant set, similar to what happens in [34]. However, the computation of the robust positive invariant set differs, since to guarantee robustness, [34] uses the notion of tube of trajectories, instead of the constraints tightening approach in [32]. The method for computing this robust positive invariant set is explained in Section 4.5.1. It is important to note that, to avoid numerical problems,

it was necessary to perform all MPC set-related calculations using meters and minutes as the length and time units, respectively.

Having solved the problem $\mathcal{P}(\mathbf{x})$, the MPC controller $\kappa_N(\mathbf{x})$ applies the first input vector of $\overline{\mathbf{u}}_F$ (that is, $\overline{\mathbf{u}}(0)$) to the system. Then, at the next control instant, the optimization problem is solved again.

4.5.1. Robust positive invariant set characterization

The robust invariant set for tracking $\Omega_a \in \mathbb{R}^{n_x+n_\theta}$ can be characterized in terms of the properties it must have to ensure recursive feasibility. In the first place, it must meet, for all $\overline{\mathbf{x}}_a = \left[\overline{\mathbf{x}}^T, \overline{\theta}^T\right]^T \in \Omega_a$, that

$$\begin{bmatrix}\mathbf{I}_{n_x} & \mathbf{0}_{n_x,n_\theta}\end{bmatrix}\overline{\mathbf{x}}_a \in \mathcal{X} \ominus \mathcal{H}(N), \tag{73}$$

$$\begin{bmatrix}K_t & L\end{bmatrix}\overline{\mathbf{x}}_a \in \mathcal{U} \ominus K_c\mathcal{H}(N-1), \tag{74}$$

so that state and control constraints are robustly met. Moreover, the artificial reference must be admissible:

$$\begin{bmatrix}\mathbf{0}_{n_x} & M_\theta\end{bmatrix}\overline{\mathbf{x}}_a \in \lambda_t\left(\mathcal{X} \ominus \mathcal{H}(N)\right) \times \left(\mathcal{U} \ominus K_c\mathcal{H}(N-1)\right), \tag{75}$$

where $\lambda_t = 0.99$ is a factor added to shrink the set, as otherwise Ω_a may not be finitely determined (see [34]). Lastly, the terminal linear feedback controller $\overline{\mathbf{u}} = K_t\left(\overline{\mathbf{x}} - \overline{\mathbf{x}}_s\right) + \overline{\mathbf{u}}_s$ must be able to steer the system into the equilibrium characterized by $\overline{\theta}$ regardless of perturbations, so Ω_a must also meet

$$\begin{bmatrix}A + BK_t & BL \\ 0 & \mathbf{I}_{n_\theta}\end{bmatrix}\Omega_a \oplus \mathcal{L}_a(N) \subseteq \Omega_a, \tag{76}$$

where $\mathcal{L}_a(i)$ can be interpreted as the set of possible deviations at the time instant i between the actual and the nominal model that a disturbance $\mathbf{w} \in \mathcal{W}$ at the initial time instant $i = 0$ can cause if the system is controlled according to $\mathbf{u} = K_c\left(\mathbf{x} - \overline{\mathbf{x}}\right) + \overline{\mathbf{u}}$. Therefore, since the MPC has a control horizon N, the impact an initial perturbation has on the final state predicted by the controller is bounded by $\mathcal{L}_a(N)$.

The robust positive invariant set is thus the largest set that meets Eqs. (73)–(76).

4.5.2. Recursive feasibility and convergence

To prove recursive feasibility, one must show that if the problem $\mathcal{P}_N(\mathbf{x}(k))$ has a solution, so does $\mathcal{P}_N(\mathbf{x}(k+1))$, where $\mathbf{x}(k+1) = A\mathbf{x}(k) + B\kappa_N(\mathbf{x}(k)) + \mathbf{w}(k)$. To do this, it is enough to find some $\overline{\mathbf{u}}_F(k+1)$, $\overline{\theta}(k+1)$ that meet the constraints of $\mathcal{P}(\mathbf{x}(k))$, but they do not have to be optimal. Let $\overline{\mathbf{u}}_F^*(k)$, $\overline{\theta}^*(k)$ denote the optimal solution of $\mathcal{P}_N(\mathbf{x}(k))$, and let us assume that $\overline{\theta}(k+1) = \overline{\theta}^*(k)$ (that is, it remains constant); then $\overline{\mathbf{x}}_s$ and $\overline{\mathbf{u}}_s$ are known, and one can introduce the change of variables $\Delta\overline{\mathbf{x}}(i) = \overline{\mathbf{x}}(i) - \overline{\mathbf{x}}_s$, $\Delta\overline{\mathbf{u}}(i) = \overline{\mathbf{u}}(i) - \overline{\mathbf{u}}_s$. Taking into account this change of variables and noting that $\|\overline{\mathbf{x}}_s\|_T^2$ is now a constant, one can set up the next cost function, which is equivalent to that in (63):

$$V_N'\left(\mathbf{x}; \overline{\mathbf{u}}_F\right) = \sum_{i=0}^{N-1}\left(\|\Delta\overline{\mathbf{x}}(i)\|_Q^2 + \|\Delta\overline{\mathbf{u}}(i)\|_R^2\right) + \|\Delta\overline{\mathbf{x}}(N)\|_P^2. \tag{77}$$

Moreover, regarding constraints, (67) is no longer necessary since $\overline{\theta}$ is not a decision variable. Constraint (65) becomes

$$\Delta\overline{\mathbf{x}}(0) = \mathbf{x}(0) - \overline{\mathbf{x}}_s, \tag{78}$$

and (66) is now

$$\Delta\overline{\mathbf{x}}(i+1) = A\Delta\overline{\mathbf{x}}(i) + B\Delta\overline{\mathbf{u}}(i), \qquad i = 0, \dots, N-1. \tag{79}$$

Constraints (68) and (69) become

$$\Delta\overline{\mathbf{x}}(i) \in \left(\mathcal{X} \oplus (-\overline{\mathbf{x}}_s)\right) \ominus \mathcal{H}(i), \qquad i = 1, \dots, N-1 \tag{80}$$

$$\Delta\overline{\mathbf{u}}(i) \in \left(\mathcal{U} \oplus (-\overline{\mathbf{u}}_s)\right) \ominus K_c\mathcal{H}(i), \qquad i = 0, \dots, N-1. \tag{81}$$

Lastly, the terminal constraint (70) is now

$$\Delta\overline{\mathbf{x}}(N) \in \Omega \ominus \mathcal{L}(N), \tag{82}$$

where Ω, which is the terminal invariant set now in $\mathbb{R}^{n_x}$ (since $\overline{\theta}^*(k)$ is fixed). To construct Ω, recall the definition of Ω_a. Condition (75) is automatically satisfied, since $\overline{\theta}^*(k)$ was a solution of the problem in the previous time step. For any $\Delta\overline{\mathbf{x}} \in \Omega$, conditions (73), (74) and (76) must be met. In this new notation, constraints (73) and (74) become

$$\Delta\overline{\mathbf{x}} \in \left(\mathcal{X} \oplus (-\overline{\mathbf{x}}_s)\right) \ominus \mathcal{H}(N), \tag{83}$$

$$K_t\Delta\overline{\mathbf{x}} \in \left(\mathcal{U} \oplus (-\overline{\mathbf{u}}_s)\right) \ominus K_c\mathcal{H}(N-1). \tag{84}$$

Lastly, regarding constraint (76), the last block row is not a condition anymore, and the first block row now reads

$$\left(A + BK_t\right)\Omega \oplus \mathcal{L}(N) \subseteq \Omega. \tag{85}$$

Finally, note that the QP problem given by the cost function (77) and the constraints from (78) to (82) is the same as the one in [32], for which recursive feasibility is demonstrated. There is only one subtle difference, and that is that in [32] Eq. (80) is also imposed for $i = 0$. However, this is not an issue because what we want to demonstrate is that if $\mathcal{P}_N(\mathbf{x}(k))$ has a solution, $\mathcal{P}_N(\mathbf{x}(k+1))$ does too. So the task is to find a solution for $\mathcal{P}_N(\mathbf{x}(k+1))$, and even if $\mathbf{x}(k)$ is not in $\mathcal{X}$, $\mathbf{x}(k+1)$ is. Thus, the problem returns a solution $\Delta\overline{\mathbf{u}}_F'^*(k+1)$, and thus $\overline{\mathbf{u}}_F'^*(k+1)$, $\overline{\theta}^*(k)$ is a solution to $\mathcal{P}_N(\mathbf{x}(k+1))$, proving recursive feasibility.

As for the convergence of this MPC scheme, it can be shown in a similar way to how it is done in [33]. The key idea is to show that the optimal cost is a Lyapunov function. That, together with the definite positiveness of the optimal cost, implies that the system is steered towards a vicinity of the reference.

4.5.3. Tuning of the proposed controller

There are a series of tuning parameters in the above controller formulation. The most important ones are probably the refreshing rate and the control horizon. Regarding the refreshing rate, ideally, one would want a fast enough refreshing rate so that the system does not deviate too much from the desired configuration due to perturbation drifts. However, a higher refreshing rate is also more computationally demanding, and, especially with on–off thrusters, it may end up consuming more propellant. A value of $T_c = 120$ s was found to be a good compromise value between both objectives.

In terms of control horizon, it must be such that the controller can look "long enough" into the future so that the MPC can take advantage of the linear dynamic model it uses to predict the states of the system. In this scenario, that would be on the order of $2\pi/n$, which is the period of the matrix A_c. However, a higher control horizon also means more computational burden, and, especially in this formulation, it might mean not being able to build the controller (indeed, since state and control constraints are tightened each prediction step into the future, one could end up with an empty terminal invariant set). A control horizon of $N = 10$ (i.e., being able to predict one fifth of the initial orbit into the future, and more as the mission progresses) was found to be a good compromise.

Matrices Q and R are the typical weighting matrices that penalize deviations in the states and control inputs, respectively. These are chosen to be

$$Q = \mathbf{I}_6, \tag{86}$$

$$R = \mathrm{diag}\left([100, 10, 100]\right). \tag{87}$$

In the units used by the controller, which are meters in length and minutes in time, the relative velocity is generally an order of magnitude lower compared to the relative position; thus, Q gives priority to minimizing position error. In these units, the control acceleration is generally an order of magnitude larger than the position error, so the R chosen focuses on reducing fuel consumption, especially in the directions of cold-gas thrusters, since they have a lower specific impulse and thus consume more propellant. All the previously mentioned errors are with respect to the virtual reference. Since, as long as state and

control variables remain within their polytopes, their exact values are not that important, a not-so-high matrix T is chosen to penalize the deviation between the artificial reference and the real equilibrium (the origin),

$$T = \mathbf{I}_6. \tag{88}$$

The values of K_t and P were obtained by tuning a discrete LQR for the matrices A and B in Section 4.2 with weighting matrices Q and R. Lastly, the value of K_c was obtained following the procedure based on LMI in [32].

5. Simulation and results

The object considered for deorbit is the Zenit-2 second stage rocket body (NORAD ID: 28353), which, according to [35], is the statistically most concerning object in the space environment, at least at that time. Not only that, this rocket body (approximated as a cylinder) is also chosen because it is a challenging object to deorbit: it is only 2 m in radius, while being 12 m long. This means that, assuming erratic rotational motion, the chaser should be at least $\sqrt{6^2 + 2^2} = 6.32$ m away from it (with the hypothesis that the COM and the geometric center coincide). This number in reality would be more around $d_{CT} = 12$ m, to account for the possible difference between the geometric center and the COM, the distance between the chaser's COM and the ion beam source, and to have some margin in case of some malfunctioning. A maximum value of the COM and geometric center offset can be estimated to be around 2 m (assume that structure asymmetries account for as much as 1 m, and that remaining fuel pushed into one of the extremes displaces it another meter).

The Zenit-2 second stage rocket body has an apogee of 849 km and a perigee of 839 km in a 71 deg inclined orbit [37], which corresponds to a practically circular orbit. The goal is to reduce its altitude to 340 km as in [25]. A typical value of the drag coefficient $c_{D,i} = 2.2$ is chosen for both the target and the chaser, and the chaser frontal area is set to $A_{f,C} = 4$ m^2.

The system is simulated in MATLAB with the aid of a custom C++ MEX function for the calculation of the force and torque of the ion beam. The characteristics of the rocket body as well as those of the chaser are presented in Table 6. The characteristics of the cold-gas thrusters in the chaser were gathered from typical values in [38], while the impulse transfer thruster is based on the XIPS 25 cm thruster (see [39]). The initial values of the state variables are shown in Table 7, where the chaser state is given in the TLVLH frame instead of ECI for clarity. It is assumed that the target starts at its perigee and, without loss of generality, that such perigee happens precisely in the x direction of the ECI frame. The chaser is assumed to start at the nominal relative position. An arbitrary initial attitude quaternion is set for the target, and its initial angular velocity is set to 0.2 deg/s with a 10 deg wobble angle, which are typical values for rocket bodies.

In terms of translational dynamics, both satellites are subject to the dynamic equations of the restricted two-body problem, with the addition of J2 and drag perturbations, as stated in Section 2.1. Moreover, the chaser motion is also affected by its actuators (cold-gas thrusters in the radial and out-of-plane directions and ion beams in the in-track direction), while the target orbit is slowly altered by the counterdirectional ion beam. Regarding rotational dynamics, they are not considered for the chaser, as it is assumed to have an attitude control system capable of keeping it aligned with its LVLH frame. In the case of the target, its attitude dynamics are altered by the action of the ion beam. This ion beam interaction is modeled with the ray-marching-based algorithm explained in Section 3.

To control the system, an MPC is used to ensure that the chaser remains in the correct region. Perturbations, unmodeled dynamics, and uncertainties are bounded to ensure that state and control constraints are always satisfied robustly. Moreover, a tracking formulation was adopted in the controller to enhance its performance. A re-tuning of the controller is done each 100 km to update its information regarding the mean motion of the orbit, and the chaser mass (which decreases due to fuel consumption).

Table 6
Properties of the target and chaser: $m_{C,0}$, initial mass of the chaser; m_T, mass of the target; $A_{f,C}$, frontal area of the chaser; R_c, H_c, radius and height of the target (assumed cylinder) respectively; d_V, distance between chaser COM and ITT source, F_{ITT}, constant force of the ITT; $F_{ICT,max}$, maximum force of the ICT; F_{CG}, force of the cold-gas thrusters; $I_{sp,IB}$, $I_{sp,CG}$, specific impulses of the ion beams and cold-gas thrusters, respectively; d_{CT}, distance between chaser and target COMs.

$m_{C,0}$ [kg]	m_T [kg]	$A_{f,C}$ [m^2]	R_c [m]	H_c [m]	d_V [m]
500	9000	4	2	12	1
F_{ITT} [mN]	$F_{ICT,max}$ [mN]	F_{CG} [mN]	$I_{sp,IB}$ [s]	$I_{sp,CG}$ [s]	d_{CT} [m]
164.8	2 F_{ITT}	50	3613	100	12

Table 7
Initial conditions of the state variables.

Variable	Value
$\mathbf{r}^I_{T,0}$ [km]	$[7217.14, 0, 0]^T$
$\mathbf{v}^I_{T,0}$ [km/s]	$[0, 2.420, 7.029]^T$
$\mathbf{r}^T_{C,0}$ [km]	$[0, 12, 0]^T$
$\mathbf{v}^T_{C,0}$ [km/s]	$[0, 0, 0]^T$
$\mathbf{q}^B_{I,0}$ [-]	$1/2\,[1, 1, 1, 1]^T$
$\boldsymbol{\omega}^B_{T,0}$ [deg/s]	$[0.0347, 0, 0.1970]^T$
$m_{C,0}$ [kg]	500

Note that, as long as the chaser remains within the correct position polytope (see Section 4.3.1), it will completely hit the target, and thus the force (ignoring noise) will just be the maximum, as remarked at the end of Section 3.5. The force therefore is independent of the attitude as long as the controller maintains the chaser in the correct region. Notice that this is a consequence of the assumed ion beam model and the position polytope set for the chaser, not a simulation simplification. Note also that the torque depends on the target's attitude even when completely hit by the beam.

The total duration of the mission is around 165 days. One may think that this number is big, especially comparing it to the numbers in, for example, [25], but it is important to recall that the target being deorbited here is around six times heavier, and its orbit is 200 km higher in altitude. This is the reason why no comparisons with other publications are given in this section. Mainly because there are no studies that deal with such massive objects in such high-altitude orbits. This case scenario is nevertheless extremely relevant, as it provides a viable alternative to remove the current statistically most concerning object in the space environment.

The evolution of the height of the target and the force that the ion beam exerts on the chaser are shown in Fig. 11. The height can be seen to have a periodic component, which is mostly due to the initial eccentricity of the target orbit. It is interesting to note that its evolution is almost linear, although the slope decreases a little as the mission progresses. The force of the ion beam, almost constant and around the maximum value, means that the chaser is indeed staying where it should, maximizing the efficiency of the mission. No zoom is made in the force evolution plot, since no relevant information can be retrieved from it; the chaser is constrained to stay inside the region where the beam fully impacts the target, and thus the observed force variations are only caused by the noise introduced in Section 2.4.

The relative position of the chaser with respect to the reference in the radial and in-track directions of its LVLH frame is included in Fig. 12. The points shown are those that belong to the control instants (that is, each 120 s), so as not to overload the plot. Here, although not much can be seen due to the high number of points (consequence of the long duration of the mission), it can be inferred that the chaser stays well within the imposed state polytope by the

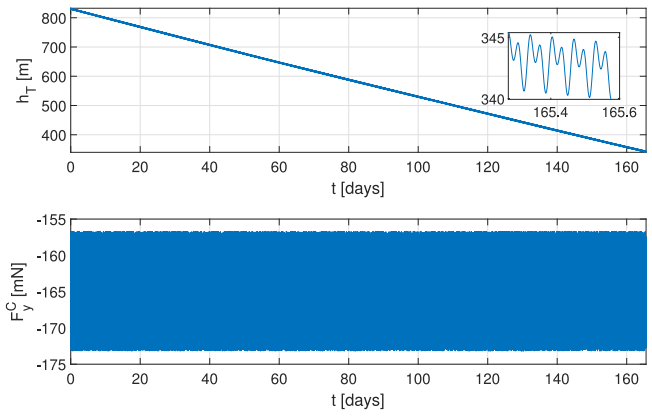

Fig. 11. Evolution of the height of the target (top) and the force in the in-track direction exerted by the ion beam on the target (bottom) along the whole deorbiting mission.

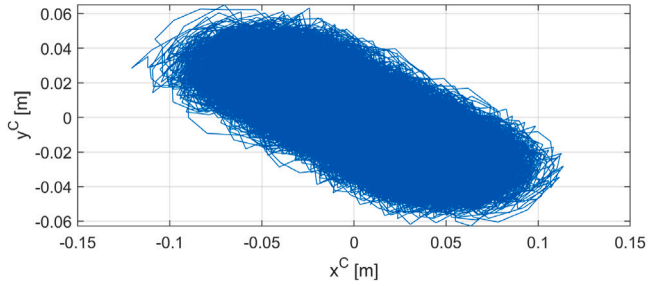

Fig. 12. Evolution of the relative state of the chaser with respect to the virtual reference in the radial and in-track directions along the whole deorbiting mission.

controller. Moreover, note that the distribution of points is more or less symmetric with respect to the radial axis, which helps to impart torque to the target "symmetrically" so that its angular momentum does not grow indefinitely. In the out-of-plane direction there is also some movement due to J2, but it is periodic and so small that cold-gas thrusters in this direction rarely fire during the mission (less than 20 times), as they do not pass the minimum control threshold explained in Section 4.4.1.

This not-firing behavior is shown at the bottom of Fig. 13, which shows the control actions in the radial, in-track and out-of-plane directions. They are shown for a time interval of 500 min instead of during the whole mission for the sake of clarity, and the interval is purposely centered so that one of the few out-of-plane cold-gas thrusters activations is included. In the radial and out-of-plane directions, since the actuators are cold-gas thrusters, what is shown is the time they fire. However, in the in-track direction, where the ion beams act, the actual acceleration commanded by the controller is plotted. As can be observed, the minimum control threshold helps the system to act only when necessary, reducing the fatigue of the thrusters and potentially minimizing fuel consumption.

The evolution of the angular velocity of the target in the body axes is shown in Fig. 14. It can be seen that, thanks to the more or less symmetric distribution previously remarked in Fig. 12, the angular velocity of the target does not increase too much during the mission, preventing potentially dangerous events such as fragmentation.

Fig. 15 shows the fuel consumption divided between the amount used by the ion beams (both ITT and ICT), the CGTs in the radial direction and the CGTs in the out-of-plane direction (in the chaser-centered LVLH frame). The cold-gas thruster propellant is seen to be in fact incomparably low to that of the ion beams (see Table 8. This fact justifies this new actuator configuration, with cold-gas thrusters in the

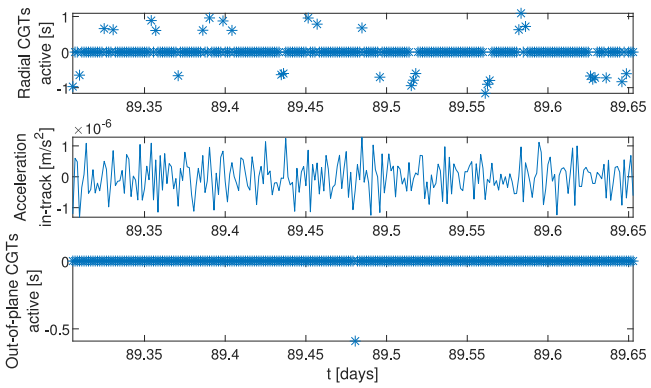

Fig. 13. Evolution of control variables during 500 min. The time interval was centered so that activation of the out-of-plane CGTs is seen. Top and bottom: time that the cold-gas thrusters are turned on during each control interval in the radial and out-of-plane direction, respectively. Middle: relative acceleration commanded in the in-track direction.

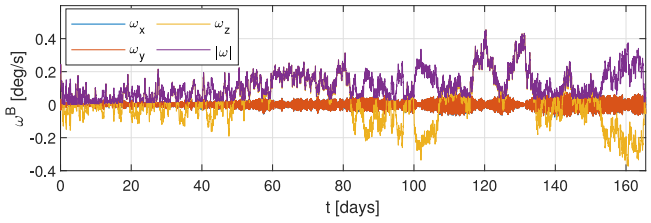

Fig. 14. Angular velocity evolution in target body axes along the whole deorbiting mission.

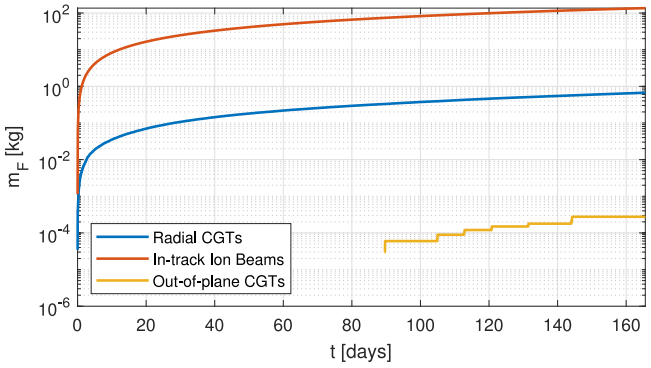

Fig. 15. Fuel consumption along the whole deorbiting mission, divided by how much fuel is consumed by the radial cold-gas thrusters, the ion beams (both ITT and ICT), and the out-of-plane cold-gas thrusters.

radial and out-of-plane directions, since they enhance controllability and ensure robustness while hardly impacting fuel consumption.

Lastly, to test the versatility of the controller, a set of simulations is conducted for different relative initial states of the chaser. Specifically, the chaser is set to start from a series of different points of a circle which is at a distance of 1.5 meters from the virtual reference. Two different circles are considered, one in the XY-plane and the other in the YZ-plane of the CLVLH frame; and the results are shown in Fig. 16, where the red-shaded region represents the position constraints polytope imposed by the controller. Simulations were performed for one orbit, enough time for the controller to stabilize the chaser around the virtual reference. As seen, the controller converges for the majority of the points, excluding the ones that are farthest from the position constraints polytope (marked in red), since the controller enforces that in just one iteration the relative state must be within that polytope. In fact, this is clearly seen in the plot, since asterisks show when the

Table 8
Total fuel consumption by radial CGTs, ion beams and out-of-plane CGTs.

Duration [days]	Fuel by radial CGTs [kg]	Fuel by ion beams [kg]	Fuel by out-of-plane CGTs [kg]
165.6	0.67	136.3	0.0003

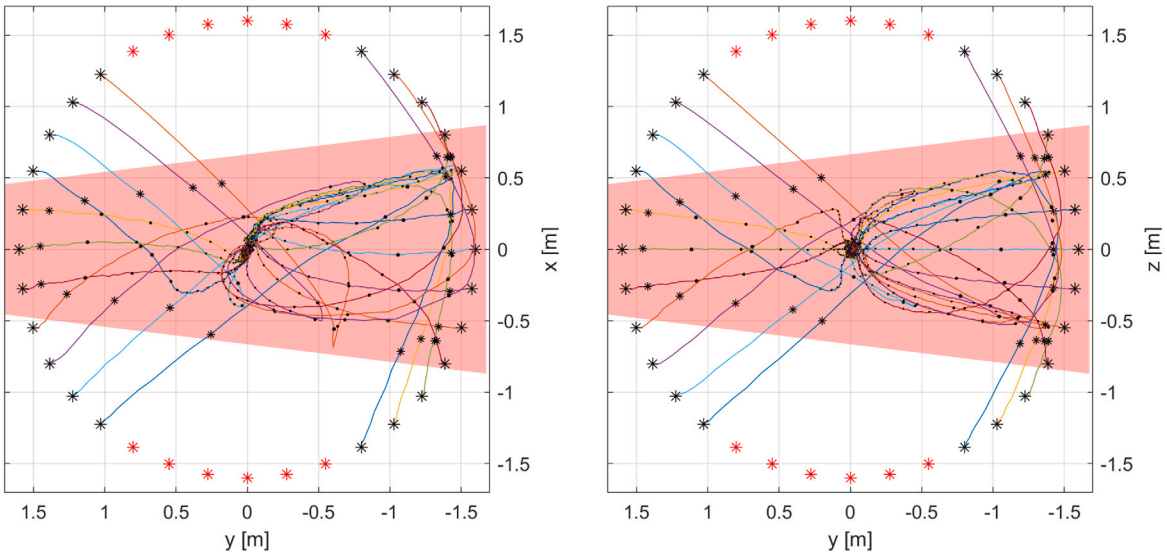

Fig. 16. Performance of the controller for different initial relative states of the chaser. The chaser starts from different positions of a 1.5 m in radius circle in the CLVLH XY-plane (left) and YZ-plane (right). Red region denotes the position constraint polytope. Only the first orbit revolution is shown. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

control is recomputed (and thus sensor measurements are used), and excluding the departing asterisks, all the rest lie inside the red-shaded region, which also validates the boundings performed for the controller. As expected, symmetric behavior is seen in the YZ-plane, excluding small differences due to sensor and force noise and hitting the target from slightly different positions.

6. Conclusions

The proposed Ion Beam Shepherd system with an enhanced actuator configuration, incorporating cold-gas thrusters alongside the typical Impulse Transfer and Impulse Compensation Thrusters, has proven to be a robust and viable solution for the deorbiting of large space debris such as the Zenit-2 second stage rocket body. Simulation results demonstrate that the system can successfully deorbit this statistically concerning 9000 kg object in approximately 165.6 days, with a fuel consumption of around 137 kg, less than 1% of which is attributed to the cold-gas thrusters.

The inclusion of cold-gas thrusters together with a novel Model Predictive Controller ensure robustness without significant fuel penalties, making the system practical for near-future missions. The development of a fast and accurate ray-marching-based ion beam force and torque computation algorithm speeds up simulation times.

While the results confirm the feasibility of using IBS for large debris deorbiting, further investigation is required on aspects such as higher-fidelity ion beam models, ion beam sputtering effects on the chaser, long-term reliability of constant-thrust ion beams, or the use of state observers to loosen the sensoring requirements. Nevertheless, the study indicates that the IBS system is a promising candidate for future space debris removal missions.

CRediT authorship contribution statement

Javier Urrios: Software, Methodology, Conceptualization. **Rafael Vazquez:** Software, Methodology, Conceptualization. **Francisco Gavilan:** Software, Methodology, Conceptualization. **Ignacio Alvarado:** Software, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors gratefully acknowledge support by grant TED2021-132099B-C33 funded by MICIU/AEI/10.13039/501100011033 and by "European Union NextGenerationEU/PRTR".

Appendix A. Abbreviations and nomenclature

Vector quantities are represented as bold symbols. Except for Section 4.4.1, if a vector has a superscript, it indicates the reference frame in which it is measured, while the subscript indicates the object it is related to (C for chaser and T for target); i.e., $\mathbf{a}_C^I$ denotes the acceleration of the chaser measured from the ECI frame. The modulus of a vector is simply denoted as the vector itself unbold. For clarity, in the notation list, superscripts of vector quantities are omitted.

ADR	Active Debris Removal
CGT	Cold-Gas Thruster
CLVLH	Chaser-centered LVLH frame
COM	Center Of Mass
DCM	Direction Cosine Matrix
ECEF	Earth-Centered Earth-Fixed frame
ECI	Earth-Centered-Inertial frame
HCW	Hill-Clohessy-Wiltshire
IBS	Ion Beam Shepherd
ICT	Impulse Compensation Thruster
ISA	International Standard Atmosphere
ITT	Impulse Transfer Thruster
LEO	Low Earth Orbit
LMI	Linear Matrix Inequality

LQR	Linear Quadratic Regulator
LTI	Linear Time-Invariant
LVLH	Local-Vertical Local-Horizontal frame
MPC	Model Predictive Control
ODE	Ordinary Differential Equation
QP	Quadratic Programming
STM	State Transition Matrix
TBA	Target Body Axes frame
TLVLH	Target-centered LVLH frame
TRL	Technology Readiness Level
ZOH	Zero-Order-Hold

$\mathbf{0}_{i\times j}$	Zero matrix of size $i \times j$
$\mathbf{1}_{i\times j}$	Ones matrix of size $i \times j$
$A_{f,i}$	Frontal area of object i
$\mathbf{a}_{i,j}$	Acceleration of object i due to factor j
$c_{D,i}$	Drag coefficient of object i
d_{CT}	Distance between chaser and target COMs
d_V	Distance between chaser COM and ITT
$\mathbf{F}_B$	Force exerted by the ICT on the target
F_{ICT}	Force magnitude of ICT
F_{ITT}	Force magnitude of ITT
H_c	Target cylinder height
h_i	Height of object i
$\mathbf{I}_i$	Identity matrix of ith order
$I_{sp,CG}$	Specific impulse of CGTs
$I_{sp,IB}$	Specific impulse of ion beams
$\mathbf{M}_B$	Torque exerted by the ICT on the target
m_i	Mass of object i
N	MPC control horizon
n	Instantaneous orbit mean motion
n_0	Ion number density at reference plane center
$\mathbf{q}_A^B$	Attitude quaternion from A to B frame
R_0	Radius of the 95% ion mass flow tube at the reference plane
R_A^B	DCM from A to B frame
R_c	Target cylinder radius
$\mathbf{r}_i$	Position of object i
T_c	Control refreshing period
$\mathbf{u}$	Control variables vector
$\mathbf{v}_i$	Velocity of object i
$\mathbf{v}_{ri}$	Relative velocity between object i and air
$\mathbf{w}_k$	State disturbance at instant k
$\mathbf{x}$	Virtual state of the chaser
α_0	Ion beam divergence angle
$\boldsymbol{\omega}_T$	Target angular velocity
$\mathcal{I}$	Target's inertia matrix
$\mathcal{U}$	Control constraints polytope
$\mathcal{X}$	State constraints polytope

References

[1] ESA Space Debris Office, ESA's Annual Space Environment Report, 2023, https://www.sdo.esoc.esa.int/environment_report/Space_Environment_Report_latest.pdf, (Accessed 27 September 2024).

[2] D.J. Kessler, B.G. Cour-Palais, Collision frequency of artificial satellites: The creation of a debris belt, J. Geophys. Res. Space Phys. 83 (A6) (1978) 2637–2646, http://dx.doi.org/10.1029/JA083iA06p02637.

[3] C.P. Mark, S. Kamath, Review of active space debris removal methods, Space Policy 47 (2019) 194–206, http://dx.doi.org/10.1016/j.spacepol.2018.12.005.

[4] J. Locke, T.J. Colvin, L. Ratliff, A. Abdul-Hamid, C. Samples, Cost and benefit analysis of mitigating, tracking, and remediating orbital debris, Cost Benefit Anal. Mitig. Track. Remediat. Orbital Debris (2024).

[5] S. Kitamura, Large space debris reorbiter using ion beam irradiation, in: 61 St International Astronautical Congress, Prague, Czech Republic, 2010.

[6] C. Bonnal, J. Ruault, P. Bultel, M. Desjean, High level requirements for an operational space debris deorbiter, in: Proc. NASA/DARPA International Conference on Orbital Debris Removal, Chantilly, VA, 2009.

[7] C. Bombardelli, J. Peláez, Ion beam shepherd for contactless space debris removal, J. Guid. Control Dyn. 34 (3) (2011) 916–920, http://dx.doi.org/10.2514/1.51832.

[8] S. Khoroshylov, Relative control of an ion beam shepherd satellite in eccentric orbits, Acta Astronaut. 176 (2020) 89–98, http://dx.doi.org/10.1016/j.actaastro.2020.06.027.

[9] V. Aslanov, A. Ledkov, Detumbling of axisymmetric space debris during transportation by ion beam shepherd in 3D case, Adv. Space Res. 69 (1) (2022) 570–580, http://dx.doi.org/10.1016/j.asr.2021.10.002.

[10] M. Merino Martínez, E. Ahedo Galilea, C. Bombardelli, H. Urrutxua Cereijo, J. Peláez Álvarez, Hypersonic Plasma Plume Expansion in Space, 32nd International Electric Propulsion Conference, 2011.

[11] M. Merino, F. Cichocki, E. Ahedo, A collisionless plasma thruster plume expansion model, Plasma Sources Sci. Technol. 24 (3) (2015) 035006, http://dx.doi.org/10.1088/0963-0252/24/3/035006.

[12] S. Kitamura, Y. Hayakawa, S. Kawamoto, A reorbiter for large GEO debris objects using ion beam irradiation, Acta Astronaut. 94 (2) (2014) 725–735, http://dx.doi.org/10.1016/j.actaastro.2013.07.037.

[13] M. Merino, E. Ahedo, C. Bombardelli, H. Urrutxua, J. Peláez, Ion beam shepherd satellite for space debris removal, Prog. Propuls. Phys. 4 (2013) 789–802, http://dx.doi.org/10.1051/eucass/201304789.

[14] F. Cichocki, M. Merino, E. Ahedo, M. Smirnova, A. Mingo, M. Dobkevicius, Electric propulsion subsystem optimization for "ion beam shepherd" missions, J. Propuls. Power 33 (2) (2017) 370–378, http://dx.doi.org/10.2514/1.B36105.

[15] H. Urrutxua, C. Bombardelli, J.M. Hedo, A preliminary design procedure for an ion-beam shepherd mission, Aerosp. Sci. Technol. 88 (2019) 421–435, http://dx.doi.org/10.1016/j.ast.2019.03.038.

[16] A. Alpatov, F. Cichocki, A. Fokov, S. Khoroshylov, M. Merino, A. Zakrzhevskii, Determination of the force transmitted by an ion thruster plasma plume to an orbital object, Acta Astronaut. 119 (2016) 241–251, http://dx.doi.org/10.1016/j.actaastro.2015.11.020.

[17] M. Redka, S. Khoroshylov, Determination of the force impact of an ion thruster plume on an orbital object via deep learning, Space Sci. Technol. 28 (5) (2022) 15–26, http://dx.doi.org/10.15407/knit2022.05.015.

[18] F. Cichocki, M. Merino, E. Ahedo, Spacecraft-plasma-debris interaction in an ion beam shepherd mission, Acta Astronaut. 146 (2018) 216–227, http://dx.doi.org/10.1016/j.actaastro.2018.02.030.

[19] A. Melnikov, V. Abgaryan, A. Mogulkin, O. Peysakhovich, V. Svotina, Experimental study of ion beam interaction with target surface aimed at developing a contactless method for space debris removal by an ion beam, Acta Astronaut. 216 (2024) 120–128, http://dx.doi.org/10.1016/j.actaastro.2024.01.004.

[20] C. Bombardelli, H. Urrutxua, M. Merino Martínez, E. Ahedo Galilea, J. Peláez Álvarez, Relative Dynamics and Control of an Ion Beam Shepherd Satellite, vol. 143, Advances in the Astronautical Sciences, 2012.

[21] S. Khoroshylov, Relative motion control system of spacecraft for contactless space debris removal, Sci. Innov 14 (4) (2018) 5–16, http://dx.doi.org/10.15407/scine14.04.005.

[22] V.S. Aslanov, A.S. Ledkov, et al., Attitude motion of cylindrical space debris during its removal by ion beam, Math. Probl. Eng. 2017 (2017) http://dx.doi.org/10.1155/2017/1986374.

[23] A. Ledkov, V. Aslanov, Attitude motion of space debris during its removal by ion beam taking into account atmospheric disturbance, in: J. Phys. Conf. Ser., 1050, IOP Publishing, 2018, 012041, http://dx.doi.org/10.1088/1742-6596/1050/1/012041.

[24] V.S. Aslanov, A.S. Ledkov, Space debris attitude control during contactless transportation in planar case, J. Guid. Control Dyn. 43 (3) (2020) 451–461, http://dx.doi.org/10.2514/1.G004686.

[25] A. Alpatov, S. Khoroshylov, C. Bombardelli, Relative control of an ion beam shepherd satellite using the impulse compensation thruster, Acta Astronaut. 151 (2018) 543–554, http://dx.doi.org/10.1016/j.actaastro.2018.06.056.

[26] S. Khoroshylov, Out-of-plane relative control of an ion beam shepherd satellite using yaw attitude deviations, Acta Astronaut. 164 (2019) 254–261, http://dx.doi.org/10.1016/j.actaastro.2019.08.016.

[27] A.S. Ledkov, V.S. Aslanov, Active space debris removal by ion multi-beam shepherd spacecraft, Acta Astronaut. 205 (2023) 247–257, http://dx.doi.org/10.1016/j.actaastro.2023.02.003.

[28] E. Camacho, C. Alba, Model Predictive Control, Springer, 2013.

[29] F. Gavilan, R. Vazquez, E.F. Camacho, Chance-constrained model predictive control for spacecraft rendezvous with disturbance estimation, Control Eng. Pract. 20 (2) (2012) 111–122, http://dx.doi.org/10.1016/j.conengprac.2011.09.006.

[30] R. Vazquez, F. Gavilan, E.F. Camacho, Model Predictive Control for Spacecraft Rendezvous in Elliptical Orbits with On/Off Thrusters, IFAC-PapersOnLine 48 (9) (2015) 251–256, http://dx.doi.org/10.1016/j.ifacol.2015.08.092.

[31] M. Schwenzer, M. Ay, T. Bergs, D. Abel, Review on model predictive control: An engineering perspective, Int. J. Adv. Manuf. Technol. 117 (5) (2021) 1327–1349, http://dx.doi.org/10.1007/s00170-021-07682-3.

[32] I. Alvarado, P. Krupa, D. Limon, T. Alamo, Tractable robust MPC design based on nominal predictions, J. Process Control 111 (2022) 75–85, http://dx.doi.org/10.1016/j.jprocont.2022.01.006.

[33] D. Limón, I. Alvarado, T. Alamo, E.F. Camacho, MPC for tracking piecewise constant references for constrained linear systems, Automatica 44 (9) (2008) 2382–2387, http://dx.doi.org/10.1016/j.automatica.2008.01.023.
[34] D. Limón, I. Alvarado, T. Alamo, E.F. Camacho, Robust tube-based MPC for tracking of constrained linear systems with additive disturbances, J. Process Control 20 (3) (2010) 248–260, http://dx.doi.org/10.1016/j.jprocont.2009.11.007.
[35] G. Borelli, M. Trisolini, M. Massari, C. Colombo, et al., A comprehensive ranking framework for active debris removal missions candidates, in: 8th European Conference on Space Debris, ESA/ESOC, ESA, 2021, pp. 1–10.
[36] J. Wong, Ray marching and signed distance functions, 2016, https://jamie-wong.com/2016/07/15/ray-marching-signed-distance-functions/, (Accessed 27 September 2024).
[37] Space-Track.org, Orbital elements data, 2024, https://www.space-track.org, (Accessed 27 September 2024).
[38] K. Anflo, B. Crowe, In-space demonstration of an ADN-based propulsion system, in: 47th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, 2011, p. 5832, http://dx.doi.org/10.2514/6.2011-5832.
[39] W. Tighe, K.-R. Chien, E. Solis, P. Rebello, D. Goebel, J. Snyder, Performance evaluation of the XIPS 25-cm thruster for application to NASA discovery missions, in: 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, 2006, p. 4666, http://dx.doi.org/10.2514/6.2006-4666.