Convex MPC and Thrust Allocation with Deadband for Spacecraft Rendezvous

Pedro Taborda, Hugo Matias, Daniel Silvestre, Pedro Lourenço

Abstract— **This paper delves into a rendezvous scenario involving a chaser and a target spacecraft, focusing on the application of Model Predictive Control (MPC) to design a controller capable of guiding the chaser toward the target. The operational principle of spacecraft thrusters, requiring a minimum activation time that leads to the existence of a control deadband, introduces mixed-integer constraints into the optimization, posing a considerable computational challenge due to the exponential complexity on the number of integer constraints. We address this complexity by presenting two solver algorithms that efficiently approximate the optimal solution in significantly less time than standard solvers, making them well-suited for real-time applications.**

Index Terms— **Spacecraft Rendezvous, Model Predictive Control, Thrust Allocation, Convex Optimization**

I. INTRODUCTION

SPACECRAFT rendezvous is an essential task in space exploration, required for many missions, including docking, Active Debris Removal (ADR) [1], and On-Orbit Servicing, Assembly, and Manufacturing (OSAM) operations [2]. Given the safety standards and high cost of these missions, the space industry tends to be very conservative in nature, favoring the use of outdated tools and theoretical methods with a significant focus on safety, robustness, and trust over performance. Solutions often rely on open-loop control, where errors accumulate and are only rectified during planned correction maneuvers [3]. The thrust allocation is also typically handled by a distinct unit that determines firing intervals for the thrusters [4], [5].

In this context, Model Predictive Control (MPC) has gained considerable traction in recent decades [6]–[10]. MPC provides continuous error correction, which is a critical feature for scenarios involving repetitive rendezvous phases, such as in-orbit assembly of large structures or servicing missions [11]. In addition, MPC can implicitly handle the allocation problem

P. Taborda is with the Instituto Superior Técnico, University of Lisbon, Portugal (email: pedrotaborda04@tecnico.ulisboa.pt).

H. Matias is with Institute for Systems and Robotics, Instituto Superior Técnico, University of Lisbon (email: hugomatias@tecnico.ulisboa.pt).

D. Silvestre is with the School of Science and Technology from the NOVA University of Lisbon, with COPELABS from Lusófona University, and also with the Institute for Systems and Robotics, Instituto Superior Técnico, University of Lisbon (email: dsilvestre@isr.tecnico.ulisboa.pt).

P. Lourenço is with GMV, Alameda dos Oceanos no. 115, 1990-392 Lisbon, Portugal (email: palourenco@gmv.com).

and minimize fuel consumption. This is particularly significant as the far-range stages of the rendezvous process substantially contribute to delta-V expenditure.

Nevertheless, the computational complexity of MPC can be a significant challenge in the space flight context, as thrusters usually require a minimum firing duration, which introduces mixed-integer constraints into the optimization [5]. Additionally, the onboard computational power, generally limited by the available power budget and the necessary hardening against space radiation, also restricts the application of MPC [12]. While some research has been successful in reducing the computational burden of MPC for some space-flight applications [13], [14], there is still a gap in addressing this challenge for the particular case of spacecraft rendezvous.

A common and straightforward approach for solving MPC problems involving discrete actuation states (e.g., on or off) is to use binary variables for the actuation states and employ a Mixed-Integer Program (MIP) solver. This is the most direct and standard approach, employed in several applications such as [15]. An MIP is an optimization problem involving discrete and continuous variables that can be solved using algorithms like the branch-and-bound algorithm, employed by the widely-used Gurobi solver [16], and its variants for MPC applications [17]. However, these algorithms are not suitable for real-time applications as they present worst-case exponential complexity, which arises from the nature of solving an MIP, requiring extensive search to identify a feasible solution.

This paper introduces an MPC strategy for spacecraft rendezvous, focusing on orbital control rather than attitude control and docking, which is addressed in more detail in [18]. We address the computational challenges arising from mixed-integer constraints introduced by spacecraft actuators (e.g., thruster deadband due to minimum impulse bit [19, Chapter 3]) by proposing two solver algorithms that efficiently approximate the optimal solution, significantly reducing computation time compared to standard solvers.

The remaining sections are organized as follows. We define the control problem and describe the spacecraft dynamics in Section II. Section III describes the MPC approach and the proposed solver algorithms, and Section IV presents simulation results. Finally, Section V summarizes conclusions.

Notation: We designate the set of integers from i to j ($i \le j$) as $\mathbb{Z}_{[i,j]}$. The set $\mathcal{S}^M$ denotes the M-ary Cartesian power of a set $\mathcal{S}$. Moreover, $\mathbf{0}_{m \times n}$ and $\mathbf{1}_{m \times n}$ denote, respectively, the matrices of zeros and ones with dimension $m \times n$ (if $n = 1$, n is omitted), and $\mathbf{I}_n$ is the identity matrix of size n. The set of positive-definite matrices with size n is designated as $\mathbb{R}^{n \times n}_{\succ 0}$.

II. PRELIMINARIES & PROBLEM STATEMENT

We aim to develop an MPC controller for a chaser spacecraft attempting to rendezvous, using chemical propulsion, with a target spacecraft that is in a circular orbit around the Earth. We assume the gravitational pull caused by the spacecraft to be negligible. The motion of the spacecraft is described in the Earth Centered Inertial (ECI) and Local Vertical Local Horizontal (LVLH) frames depicted in Fig. 1. The ECI frame has its origin at the Earth's center and is nonrotating. Conversely, the LVLH frame has its origin at the target's center, and its axes are defined as follows: the z-axis points from the target to the Earth's center, the x-axis has the target's velocity direction, and the y-axis is perpendicular to the orbital plane.

A. Full Dynamics Model

Since the goal is controlling the chaser spacecraft's relative position to the target, it is useful to describe the motion of the spacecraft in the LVLH frame, as the target remains stationary in this frame. At each time instant t, the chaser's position in the LVLH frame, $\mathbf{r}_C^L(t) \in \mathbb{R}^3$, is given by

$$\mathbf{r}_C^L(t) = \mathbf{R}^\top(t)(\mathbf{r}_C^E(t) - \mathbf{r}_T^E(t)), \qquad (1)$$

where $\mathbf{r}_C^E(t)$ and $\mathbf{r}_T^E(t)$ denote the positions of the chaser and target spacecraft in the ECI frame, and $\mathbf{R}(t) \in \mathrm{SO}(3)$ is the rotation matrix from the LVLH frame to the ECI frame. Taking now the second-order derivative of (1) with respect to time, it can be shown that the chaser dynamics in the LVLH frame are described by

$$\ddot{\mathbf{r}}_C^L(t) = -2\boldsymbol{\omega} \times \dot{\mathbf{r}}_C^L(t) - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}_C^L(t)) \\ + \mathbf{R}^\top(t)(\ddot{\mathbf{r}}_C^E(t) - \ddot{\mathbf{r}}_T^E(t)), \qquad (2)$$

where $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity of the LVLH frame with respect to the ECI frame. Also, by Newton's Second Law and Law of Universal Gravitation, $\ddot{\mathbf{r}}_C^E(t)$ and $\ddot{\mathbf{r}}_C^E(t)$ are given by

$$\ddot{\mathbf{r}}_C^E(t) = -\frac{Gm_E}{\|\mathbf{r}_C^E(t)\|^3}\mathbf{r}_C^E(t) + \frac{1}{m_C}\mathbf{R}(t)\mathbf{u}(t), \\ \ddot{\mathbf{r}}_T^E(t) = -\frac{Gm_E}{\|\mathbf{r}_T^E(t)\|^3}\mathbf{r}_T^E(t), \qquad (3)$$

where G is the gravitational constant, m_E is the Earth's mass, m_C is the chaser's mass, and $\mathbf{u}(t) \in \mathbb{R}^3$ denotes the forces acting on the chaser spacecraft due to the actuation, expressed in the LVLH frame. In addition, we highlight that $\mathbf{r}_T^E(t)$, $\mathbf{R}(t)$ and $\boldsymbol{\omega}$ can be specified *a priori* based on the target's circular orbit, which is assumed to be known.
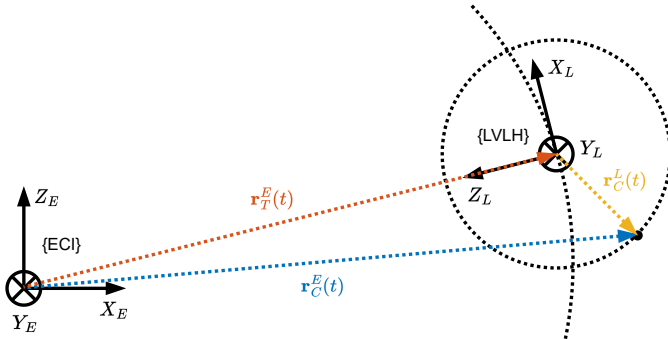

Fig. 1. ECI and LVLH reference frames.

B. Simplified Model

Given the highly nonlinear nature of the model described by (2) and (3), it is beneficial for the MPC controller to consider a simplified model for enhanced efficiency. By noting that the distance between the two spacecraft is much smaller than the orbital radius of the target and then linearizing the full relative dynamics in the LVLH frame around the circular orbit, the chaser's motion can be approximately described by the well-known Clohessy-Wiltshire (CW) equations [3, Appendix A]. These can be expressed in linear state-space form as

$$\dot{\tilde{\mathbf{x}}}(t) = \mathbf{A}\tilde{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t), \qquad (4)$$

where $\mathbf{A}$ and $\mathbf{B}$ are given by

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2\omega \\ 0 & -\omega^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3\omega^2 & -2\omega & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \frac{1}{m_C}\begin{bmatrix} \mathbf{0}_{3\times3} \\ \mathbf{I}_{3\times3} \end{bmatrix}, \qquad (5)$$

and with state $\tilde{\mathbf{x}}(t) = [\tilde{\mathbf{r}}_C^L(t)^\top \ \dot{\tilde{\mathbf{r}}}_C^L(t)^\top]^\top$, where tilde notation distinguishes the state of the approximate model from the state $\mathbf{x}(t) = [\mathbf{r}_C^L(t)^\top \ \dot{\mathbf{r}}_C^L(t)^\top]^\top$ of the full dynamics model.

C. Actuation and Model Discretization

Spacecraft propulsion systems vary in complexity and capabilities. Such systems encompass chemical and electric propulsion, achieving high and low thrust, respectively [19]. Within chemical propulsion, necessary for rendezvous, thrusters operate on a shared principle: a propellant is heated and expelled through a nozzle, generating a fixed and constant thrust. In addition, thrusters have a minimum operational time that they should be active for (minimum impulse bit [19]).

At each sampling step k, each thruster can either generate a single force pulse from the beginning of the sampling interval or remain inactive. Therefore, the control signal produced by a single thruster i, $\mathbf{u}_i(t)$, is piecewise constant and given by

$$\mathbf{u}_i(t) = \begin{cases} \mathbf{f}_i, & \text{if } t \in [kh, kh + s_i[k]), \\ \mathbf{0}_3, & \text{if } t \in [kh + s_i[k], (k+1)h), \end{cases} \qquad (6)$$

for all $k \in \mathbb{Z}_0^+$, where $\mathbf{f}_i \in \mathbb{R}^3 \setminus \{\mathbf{0}_3\}$ denotes the constant thrust force generated by the thruster, h denotes the sampling period, and $s_i[k] \in \{0\} \cup [h_{\min}, h]$ is the pulse duration, which must have a minimum value of $h_{\min} \in (0, h]$ when the thruster is activated. Hence, by noting that the overall control signal is given by $\mathbf{u}(t) = \sum_{i=1}^M \mathbf{u}_i(t)$, the linear continuous-time model in (4) can be exactly discretized as

$$\tilde{\mathbf{x}}[k+1] = e^{\mathbf{A}h}\tilde{\mathbf{x}}[k] + \sum_{i=1}^M \int_{kh}^{(k+1)h} e^{\mathbf{A}((k+1)h-\tau)}\mathbf{B}\mathbf{u}_i(\tau)\,d\tau$$

$$= \boldsymbol{\Phi}\tilde{\mathbf{x}}[k] + \boldsymbol{\Phi}\sum_{i=1}^M \int_0^{s_i[k]} e^{-\mathbf{A}\tau}\,d\tau\mathbf{B}\mathbf{f}_i$$

$$= \boldsymbol{\Phi}\tilde{\mathbf{x}}[k] + \boldsymbol{\Phi}\sum_{i=1}^M \mathbf{G}(s_i[k])\mathbf{B}\mathbf{f}_i, \qquad (7)$$

with $\boldsymbol{\Phi} := e^{\mathbf{A}h}$ and $\mathbf{G}(s) := \int_0^s e^{-\mathbf{A}\tau}\,d\tau$, $\forall s \in \mathbb{R}$.

D. Linearization of the Actuator Dynamics

For the discrete-time model in (7), the state transition matrix $e^{\mathbf{A}h}$ is given by

$$e^{\mathbf{A}h} = \begin{bmatrix} 1 & 0 & 6(\omega h - \sin(\omega h)) & \frac{4}{\omega}\sin(\omega h) - 3h & 0 & \frac{2}{\omega}(1 - \cos(\omega h)) \\ 0 & \cos(\omega h) & 0 & 0 & \frac{1}{\omega}\sin(\omega h) & 0 \\ 0 & 0 & 4 - 3\cos(\omega h) & \frac{2}{\omega}(\cos(\omega h) - 1) & 0 & \frac{1}{\omega}\sin(\omega h) \\ 0 & 0 & 6\omega(1 - \cos(\omega h)) & 4\cos(\omega h) - 3 & 0 & 2\sin(\omega h) \\ 0 & -\omega\sin(\omega h) & 0 & 0 & \cos(\omega h) & 0 \\ 0 & 0 & 3\omega\sin(\omega h) & -2\sin(\omega h) & 0 & \cos(\omega h) \end{bmatrix}.$$

$$(8)$$

As a result, $\mathbf{G}$ is a nonlinear function, rendering the model in (7) nonlinear with respect to the control variables $s_i[k]$. As we aim to build an MPC constrained by these dynamics at each time step, it is beneficial to linearize $\mathbf{G}$ so that linear solvers can be used to solve the optimization problem. Consequently, linearizing $\mathbf{G}$ around a point s_0 leads to

$$\mathbf{G}(s) \simeq \mathbf{G}(s_0) + e^{-\mathbf{A}s_0}(s - s_0), \qquad (9)$$

resulting in the affine discrete-time model

$$\tilde{\mathbf{x}}[k+1] = \mathbf{\Phi}\tilde{\mathbf{x}}[k] + \mathbf{\Gamma}\mathbf{s}[k] + \mathbf{d}, \qquad (10)$$

where $\mathbf{s}[k] := [s_1[k] \dots s_M[k]]^\top$, $\mathbf{\Gamma} = e^{\mathbf{A}(h-s_0)}[\mathbf{Bf}_1 \dots \mathbf{Bf}_M]$ and $\mathbf{d} = e^{\mathbf{A}h}(\mathbf{G}(s_0) - e^{-\mathbf{A}h}s_0)\mathbf{\Gamma}\mathbf{1}_M$.

III. MODEL PREDICTIVE CONTROL WITH MIXED-INTEGER CONSTRAINTS

Considering the setup described in the preceding section, at every discrete-time instant k, for a given initial state $\mathbf{x}[k]$ of the chaser spacecraft, the MPC controller relies on the following optimization problem for a given horizon N:

$$\begin{aligned} \underset{\hat{\mathbf{x}}[\cdot], \hat{\mathbf{s}}[\cdot]}{\text{minimize}} \quad & J(\hat{\mathbf{x}}[\cdot], \hat{\mathbf{s}}[\cdot]) \\ \text{subject to} \quad & \hat{\mathbf{x}}[0] = \mathbf{x}[k], \\ & \hat{\mathbf{x}}[n+1] = \mathbf{\Phi}\hat{\mathbf{x}}[n] + \mathbf{\Gamma}\hat{\mathbf{s}}[n] + \mathbf{d}, \; \forall n \in \mathbb{Z}_{[0,N-1]}, \\ & \hat{\mathbf{s}}[n] \in (\{0\} \cup [h_{\min}, h])^M, \; \forall n \in \mathbb{Z}_{[0,N-1]}, \end{aligned}$$

$$(11)$$

where $\hat{\mathbf{x}}[\cdot]$ and $\hat{\mathbf{s}}[\cdot]$ are the optimization variables matching the predicted state and control sequences over the horizon, and the cost function J should translate the mission objectives. For the problem at hand, a possible definition of the cost function is

$$J(\hat{\mathbf{x}}[\cdot], \hat{\mathbf{s}}[\cdot]) = \hat{\mathbf{x}}[N]^\top \mathbf{Q}\hat{\mathbf{x}}[N] + \sum_{n=0}^{N-1} \mathbf{1}_M^\top \hat{\mathbf{s}}[n], \qquad (12)$$

where $\mathbf{Q} \in \mathbb{R}_{\succ 0}^{6 \times 6}$. The first component of (12) expresses the objective of reaching the target spacecraft, located at the origin of the LVLH frame. The second penalizes the control effort, obtained by summing the activation times of all thrusters over the prediction horizon.

Owing to the constraints on each actuator's activation time $\hat{s}_i[n]$, the optimization problem (11) becomes a Mixed-Integer Linear Program (MILP). The most straightforward approach to address this problem involves using a **Standard** MILP solver, such as Gurobi, which guarantees the optimal solution. However, employing **Standard** MILP solvers becomes impractical for real-time applications as they present worst-case exponential complexity on the number of integer constraints ($N \times M$). In this letter, we propose two alternative, more computationally efficient methods capable of producing solutions similar to the optimal one.

A. Projected Algorithm

The first proposed solver algorithm is denoted as **Projected**. Consider the auxiliary variables $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \{0,1\}^M$ such that, if $\alpha_i = 1$, then $\hat{s}_i[0] \in [h_{\min}, h]$, and if $\beta_i = 1$, then $\hat{s}_i[0] = 0$. Also, α_i and β_i are such that $\alpha_i\beta_i = 0$, $\forall i \in \mathbb{Z}_{[1,M]}$. The algorithm consists of sequentially solving the relaxed problem

$$\begin{aligned} \underset{\hat{\mathbf{x}}[\cdot], \hat{\mathbf{s}}[\cdot]}{\text{minimize}} \quad & J(\hat{\mathbf{x}}[\cdot], \hat{\mathbf{s}}[\cdot]) \\ \text{subject to} \quad & \hat{\mathbf{x}}[0] = \mathbf{x}[k], \\ & \hat{\mathbf{x}}[n+1] = \mathbf{\Phi}\hat{\mathbf{x}}[n] + \mathbf{\Gamma}\hat{\mathbf{s}}[n] + \mathbf{d}, \; \forall n \in \mathbb{Z}_{[0,N-1]}, \\ & \hat{\mathbf{s}}[0] \in \bigtimes_{i=1}^{M}[\alpha_i h_{\min}, (1 - \beta_i)h], \\ & \hat{\mathbf{s}}[n] \in [0, h]^M, \; \forall n \in \mathbb{Z}_{[1,N-1]}, \end{aligned}$$

$$(13)$$

to determine an approximate solution for the first control $\hat{\mathbf{s}}[0]$.

The vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are initialized as $\mathbf{0}_M$ to fully ignore the infeasible set for the pulse duration, $(0, h_{\min})$, in the first iteration of the algorithm. Then, at each iteration, the algorithm solves (13) using the current $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ values and checks if the first control of the solution, $\hat{\mathbf{s}}[0]$, is feasible for the original problem (11). If it does, the algorithm outputs $\hat{\mathbf{s}}[0]$. Otherwise, the algorithm computes the projection of $\hat{\mathbf{s}}[0]$ onto the feasible set $(\{0\} \cup [h_{\min}, h])^M$, denoted as $\bar{\mathbf{s}}$. For each activation time $\hat{s}_i[0]$ that is infeasible for (11) (i.e., $\hat{s}_i[0] \in (0, h_{\min})$), the projected component $\bar{s}_i$ is either $\bar{s}_i = 0$ or $\bar{s}_i = h_{\min}$. Thus, the algorithm locks $\hat{s}_i[0]$ to either $\{0\}$ if $\bar{s}_i = 0$ or to the interval $[h_{\min}, h]$ if $\bar{s}_i = h_{\min}$, by updating α_i and β_i accordingly. On the other hand, for each feasible activation time $\hat{s}_i[0]$, we have that $\hat{s}_i[0] = \bar{s}_i$, and the corresponding α_i and β_i values remain unchanged. This process repeats until a feasible solution $\hat{\mathbf{s}}[0]$ is found, terminating after at most $M+1$ iterations, which would match the case in which only one pair (α_i, β_i) is updated per iteration. This process is further detailed in Algorithm 1.

B. Relaxed Algorithm

The second algorithm, denoted as **Relaxed**, solves (13) just once with $\boldsymbol{\alpha} = \boldsymbol{\beta} = \mathbf{0}_M$, fully ignoring the infeasible set for the pulse duration. The first control of the solution, $\hat{\mathbf{s}}[0]$, is then projected onto the set $(\{0\} \cup [h_{\min}, h])^M$ to return a feasible solution for (11). Algorithm 2 details the **Relaxed** solver.

Algorithm 1 Projected Solver

1: **Input**: $\mathbf{x}[k]$
2: **Output**: $\hat{\mathbf{s}}[0]$
3: **Set**: $\boldsymbol{\alpha} \leftarrow \mathbf{0}_M$, $\boldsymbol{\beta} \leftarrow \mathbf{0}_M$
4: **For** iter $\in \mathbb{Z}_{[0,M]}$:
5:     Solve (13) and set $\hat{\mathbf{s}}[0]$ to the solution
6:     **If** $\hat{\mathbf{s}}[0] \in (\{0\} \cup [h_{\min}, h])^M$:
7:         **Return** $\hat{\mathbf{s}}[0]$
8:     **EndIf**
9:     $\bar{\mathbf{s}} \leftarrow \underset{\mathbf{s} \in (\{0\} \cup [h_{\min}, h])^M}{\arg\min} \|\hat{\mathbf{s}}[0] - \mathbf{s}\|$
10:     **For** $i \in \{j \in \mathbb{Z}_{[1,M]} : \hat{s}_j[0] \in (0, h_{\min})\}$:
11:         $\alpha_i \leftarrow \frac{\bar{s}_i}{h_{\min}}$, $\beta_i \leftarrow |1 - \alpha_i|$ // $\bar{s}_i \in \{0, h_{\min}\}$
12:     **EndFor**
13: **EndFor**

Algorithm 2 Relaxed Solver

1: **Input**: $\mathbf{x}[k]$
2: **Output**: $\hat{\mathbf{s}}[0]$
3: Solve (13) with $\boldsymbol{\alpha} = \boldsymbol{\beta} = \mathbf{0}_M$, set $\hat{\mathbf{s}}[0]$ to the solution
4: $\hat{\mathbf{s}}[0] \leftarrow \underset{\mathbf{s} \in (\{0\} \cup [h_{\min}, h])^M}{\arg\min} \|\hat{\mathbf{s}}[0] - \mathbf{s}\|$
5: **Return** $\hat{\mathbf{s}}[0]$

The **Relaxed** algorithm represents the simplest approach, considered both because it is computationally the least expensive and also a natural conceptual extension of the **Projected** algorithm. Furthermore, in [14], a comparable convexification technique has been shown to produce the optimal solution for a class of optimal control problems with similar mixed-integer constraints. However, the problem addressed in our paper has a different structure from the problems considered in [14] and the actuation constraints do not verify some of the conditions that ensure the optimality of the **Relaxed** solution with respect to the original problem. In this paper, we introduce a similar convexification approach for the class of problems (11), with the addition of the projection onto the feasible set to ensure a solution that satisfies the constraints of the original problem.

IV. SIMULATION RESULTS

In this section, the efficacy of the MPC controller is assessed for the different solvers through simulation examples obtained in a MATLAB environment. For simplicity, and without loss of generality, we assume the target's orbital plane is aligned with the zx-plane of the ECI frame. Consequently, the motion of the target and the rotation matrix are described by

$$\mathbf{r}_T^E(t) = \begin{bmatrix} R_T \cos(\omega t) \\ 0 \\ R_T \sin(\omega t) \end{bmatrix}, \quad \mathbf{R}(t) = \begin{bmatrix} -\sin(\omega t) & 0 & -\cos(\omega t) \\ 0 & 1 & 0 \\ \cos(\omega t) & 0 & -\sin(\omega t) \end{bmatrix}, \quad (14)$$

with $\boldsymbol{\omega} = [0\ 0\ \omega]^\top$ and $\omega = \sqrt{Gm_E R_T^{-3}}$. The parameters used in the simulations are presented in Table I.

The simulation results presented in this section were obtained using the Gurobi [16] solver as the **Standard** algorithm. All computations were executed on a single desktop computer equipped with an AMD Ryzen 5 3600 @ 3.60 GHz processor and 32 GB of RAM. The numerical integration of the complete dynamics was conducted using the Runge–Kutta method through the *ode45* implementation in MATLAB.

TABLE I
DEFAULT SIMULATION PARAMETERS

Parameter	Value	Description
G	6.674×10^{-11} m^3 kg^{-1} s^{-2}	Gravitational constant
m_E	5.972×10^{24} kg	Earth's mass
R_T	7171 km	Target's orbital radius
m_C	2000 kg	Chaser's mass
M	6	Number of thrusters
$\mathbf{f}_i$	$1000\mathbf{w}_i$ [N] *	Thrust forces
$\mathbf{Q}$	$\mathrm{diag}(\mathbf{1}_6)$	State error weight matrix
$h_{\min}$	5 s	Minimum activation time
h	10 s	Sampling period
T_{sim}	3600 s	Simulation time
s_0	$h/2$	Linearization point
$\mathbf{x}(0)$	$[0\ 0\ 100\ \mathrm{km}\ 0\ 0\ 0]^\top$	Chaser's initial state

* $\mathbf{w}_1 = -\mathbf{w}_4 = [1\ 0\ 0]^\top$, $\mathbf{w}_2 = -\mathbf{w}_5 = [0\ 1\ 0]^\top$, $\mathbf{w}_3 = -\mathbf{w}_6 = [0\ 0\ 1]^\top$.

Assessing quantitative metrics is useful for drawing conclusions about the MPC controller's performance. In this scenario, we consider two evaluation metrics:

1) **Mission Time:** the mission time is the time T_{mission} such that the distance between the chaser and the target does not exceed $1\ \mathrm{km}$ for all $t \geq T_{\mathrm{mission}}$.

2) **Fuel Consumption**: the fuel consumption refers to the total amount of fuel spent by the chaser spacecraft. Since we assume identical thrusters, except for their direction, the fuel consumed by each thruster is given by the total duration of its activation. The overall fuel consumption is the sum of the individual fuel consumption of each thruster, measured in seconds.

A. Variation of the Minimum Activation Time

We begin by briefly assessing the influence of the minimum activation time on the trajectories. In Fig. 2, we present the resulting trajectories and activation profiles obtained with the **Standard** algorithm for $h_{\min} = 0\ \mathrm{s}$, $h_{\min} = 2\ \mathrm{s}$, and $h_{\min} = 4\ \mathrm{s}$. Since the trajectories obtained with the other two algorithms are indistinguishable from those in Fig. 2, we present only those obtained with the **Standard** algorithm. The trajectories are displayed in zx-plane of the LVLH frame, showcasing only the activation profiles of the thrusters along this plane. Additionally, Table II presents the fuel consumption, mission time, and the accumulated MPC solve time for each simulation.

Examining Fig. 2 and Table II reveals that the minimum activation time has a minimal impact on the overall performance of the controller. The central part of the maneuver, requiring certain thrusters to remain continuously on for extended periods, remains unaffected. Notably, for $h_{\min} = 0\ \mathrm{s}$, the algorithm addresses a linear problem without integer constraints, leading to significantly lower computation times. These findings suggest the feasibility of using a convex relaxation, which the **Projected** and **Relaxed** algorithms are designed around.
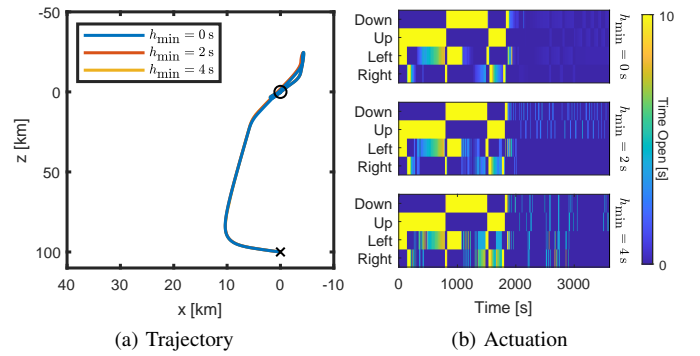

(a) Trajectory (b) Actuation

Fig. 2. Resulting trajectories and activation profiles obtained using the **Standard** algorithm for different values of the minimum activation time. Chaser's initial position ×, target position ○.

TABLE II
MISSION TIMES AND FUEL CONSUMPTION OBTAINED FOR DIFFERENT VALUES OF THE MINIMUM ACTIVATION TIME WITH $N = 10$

Min Activation Time	Fuel Cons.	Mission Time	Acc. Solve Time
$h_{\min} = 0\ \mathrm{s}$	3070.49 s	1930 s	4.50 s
$h_{\min} = 2\ \mathrm{s}$	2931.49 s	1890 s	27.19 s
$h_{\min} = 4\ \mathrm{s}$	3068.53 s	1890 s	21.62 s

B. Variation of the Horizon Length

In order to evaluate the influence of the prediction horizon on the algorithms, the three solver algorithms were applied to the rendezvous scenario employing different horizon lengths. Fig. 3 displays the resulting trajectories and activation profiles for $N = 5$, $N = 10$, $N = 15$, and $N = 100$. Additionally, Table III presents the fuel consumption, mission time, and the accumulated MPC solve time for each simulation.
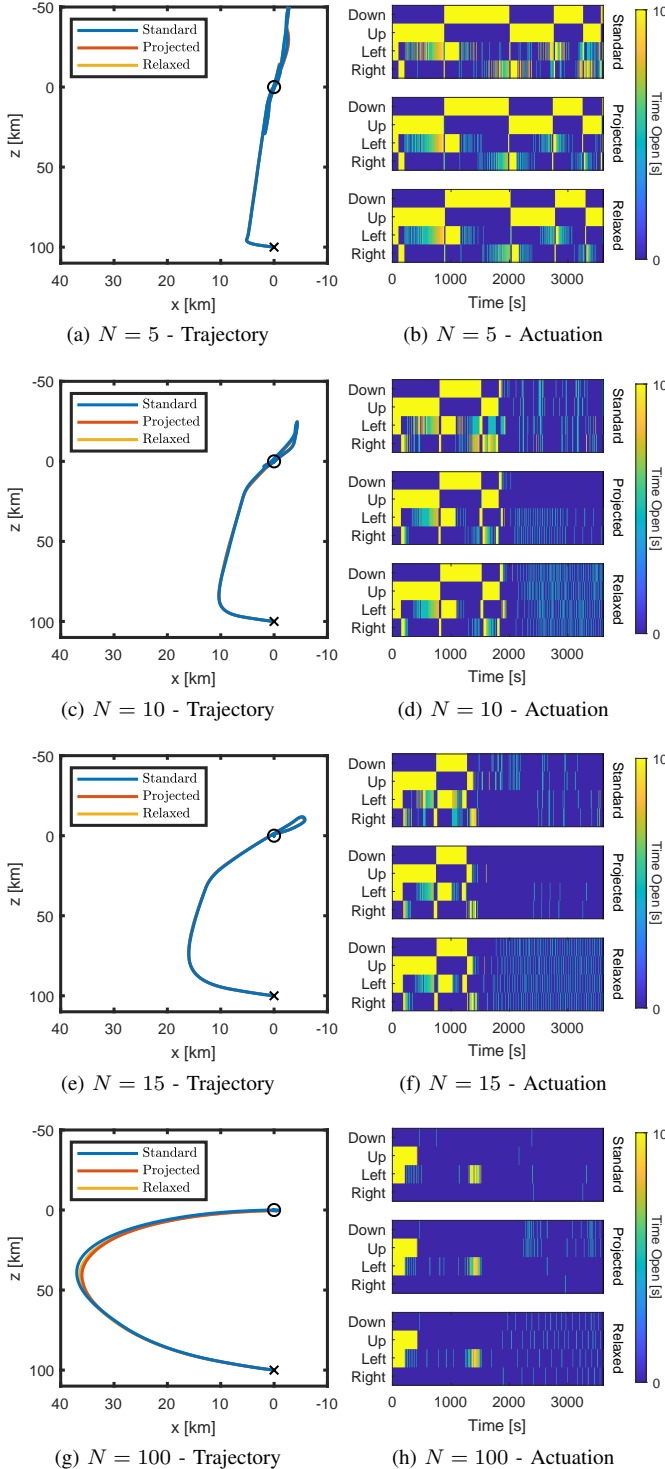

(a) $N = 5$ - Trajectory

(b) $N = 5$ - Actuation

(c) $N = 10$ - Trajectory

(d) $N = 10$ - Actuation

(e) $N = 15$ - Trajectory

(f) $N = 15$ - Actuation

(g) $N = 100$ - Trajectory

(h) $N = 100$ - Actuation

Fig. 3. Resulting trajectories and activation profiles obtained using each algorithm for different horizon lengths. Chaser's initial position ×, target position ○.

TABLE III
MISSION TIMES AND FUEL CONSUMPTION OBTAINED USING EACH
ALGORITHM FOR DIFFERENT HORIZON LENGTHS

Horizon	Algorithm	Fuel Cons.	Mission Time	Acc. Solve Time
	Standard	5667.33 s	3580 s	4.25 s
$N = 5$	Projected	5186.60 s	3570 s	3.36 s
	Relaxed	5229.32 s	3590 s	2.01 s
	Standard	3286.42 s	1860 s	14.39 s
$N = 10$	Projected	2925.65 s	1890 s	3.84 s
	Relaxed	2885.57 s	1880 s	2.89 s
	Standard	2470.85 s	1420 s	27.86 s
$N = 15$	Projected	2252.42 s	1420 s	4.64 s
	Relaxed	2299.30 s	1430 s	3.10 s
	Standard	777.13 s	1400 s	3791.84 s
$N = 100$	Projected	791.25 s	1430 s	30.69 s
	Relaxed	808.94 s	1430 s	23.08 s

The results suggest that exploring the mixed-integer solution space does not yield substantial benefits in such scenarios, as both the **Projected** and **Relaxed** algorithms produce trajectories that are very similar to the optimal solution. In addition, Table III reinforces this observation, as the fuel consumption and mission time across different values of N are very similar for all three algorithms. Remarkably, the **Projected** algorithm, which partially explores the mixed-integer solution space, fails to generate significantly better trajectories than the **Relaxed** algorithm, which does not explore the mixed-integer solution space at all. With a sufficiently large prediction horizon, such as $N = 100$, the results resemble a Hohmann transfer orbit - the (two impulse) maneuver using the lowest possible amount of energy to transfer a spacecraft between orbits.

The solutions produced by these algorithms closely resemble the optimal solution, as observed in the activation profiles and consequently in the trajectories themselves. The substantial accumulated computation time of the **Standard** algorithm makes it impractical not only for real-time applications but also for simulations employing extended prediction horizons. Conversely, the **Projected** and **Relaxed** algorithms are much more computationally efficient, rendering them a more suitable option for real-time applications.

C. Computation Times

For a deeper analysis of the computational cost of each algorithm, we conducted 100 simulations using each algorithm for $N = 5$, $N = 10$, and $N = 15$. Fig. 4 depicts the resulting histograms and the mean solve times over time. Additionally, Table IV provides the values of the mean computation times, as well as the 95 % and 99 % percentiles of these times.

As presented in Fig. 4 and Table IV, the **Standard** algorithm exhibits a significantly greater average computation time than the other two algorithms across all values of N. The computation times of the **Standard** algorithm also exhibit a greater dispersion, as demonstrated by their wider distributions. On the other hand, the **Projected** and **Relaxed** algorithms differ by less than an order of magnitude in terms of average computation time, with the **Projected** algorithm taking slightly longer than the **Relaxed** algorithm. This discrepancy is anticipated, as the **Projected** algorithm solves multiple optimization problems

(a) $N = 5$ - Histogram

(b) $N = 5$ - Profile

(c) $N = 10$ - Histogram

(d) $N = 10$ - Profile

(e) $N = 15$ - Histogram
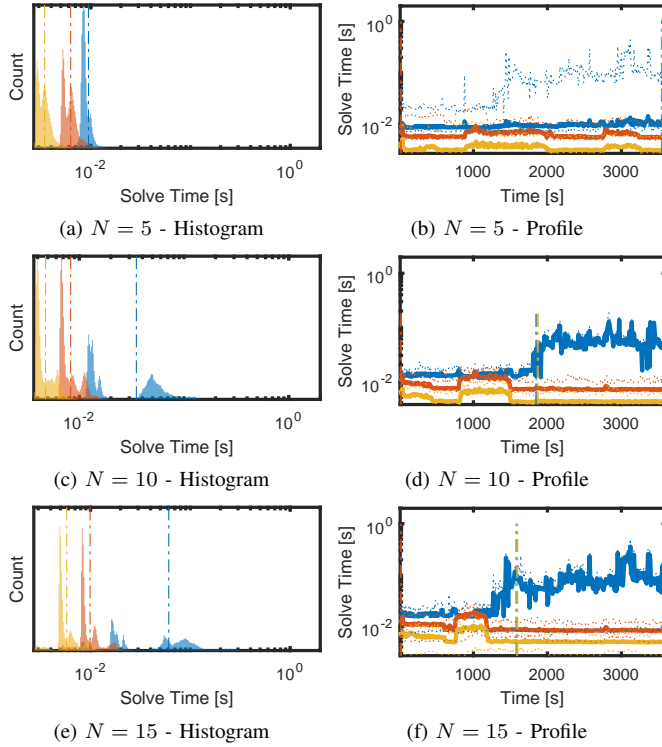
(f) $N = 15$ - Profile

Fig. 4. Computation times obtained using each algorithm for different horizons - **Standard** (blue), **Projected** (red), **Relaxed** (yellow). Mean solve time (dash-dotted, histograms), mission time (dash-dotted, profiles), maximum solve time (dotted, profiles).

per MPC iteration, while the **Relaxed** algorithm solves only one per iteration. The **Projected** and **Relaxed** algorithms also display similar and narrower distributions of the solve times.

The distributions shown in Fig. 4 exhibit a bimodal pattern, which becomes more pronounced as N increases, particularly for the **Standard** algorithm. This behavior can be understood by observing the temporal profiles of the mean solver times in Fig. 4. One can observe that the solve times for each algorithm are not constant throughout the simulation; instead, they vary significantly, especially for the **Standard** algorithm. Interestingly, as the simulation progresses and the chaser approaches the target, the solve times for the **Standard** algorithm increase. In contrast, the solve times for the other two algorithms exhibit less variation throughout the simulation and, in fact, tend to decrease as the simulation advances.

TABLE IV

COMPUTATION TIMES OBTAINED USING EACH ALGORITHM
FOR DIFFERENT HORIZON LENGTHS

Horizon	Algorithm	Mean	95% Percentile	99% Percentile
$N = 5$	Standard	9.46 ms	10.67 ms	17.79 ms
	Projected	6.27 ms	8.59 ms	10.79 ms
	Relaxed	3.44 ms	4.53 ms	5.74 ms
$N = 10$	Standard	35.07 ms	84.22 ms	124.13 ms
	Projected	8.27 ms	12.30 ms	14.88 ms
	Relaxed	4.76 ms	6.75 ms	7.80 ms
$N = 15$	Standard	60.80 ms	163.96 ms	280.52 ms
	Projected	9.91 ms	17.31 ms	19.55 ms
	Relaxed	5.79 ms	9.58 ms	10.91 ms

V. CONCLUSION & FUTURE RESEARCH

This paper focuses on the application of MPC to spacecraft rendezvous, addressing the computational challenges arising from mixed-integer constraints on the actuation. We propose two solver algorithms that efficiently approximate the optimal solution in significantly less time than standard MIP solvers, rendering them suitable for real-time applications. Extensions may include addressing elliptical orbits and safety concerns.

REFERENCES

[1] C. Bonnal, J.-M. Ruault, and M.-C. Desjean, "Active debris removal: Recent progress and current trends," *Acta Astronautica*, vol. 85, pp. 51–60, Apr. 2013.

[2] D. Arney, R. Sutherland, J. Mulvaney, D. Steinkoenig, C. Stockdale, and M. Farley, "On-orbit Servicing, Assembly, and Manufacturing (OSAM) State of Play, 2021 Edition," White Paper 20210022660, OSAM National Initiative, Oct. 2021.

[3] W. Fehse, *Automated Rendezvous and Docking of Spacecraft*. Cambridge University Press, Nov. 2003.

[4] J. A. Bezerra and D. A. Santos, "Optimal exact control allocation for under-actuated multirotor aerial vehicles," *IEEE Control Systems Letters*, vol. 6, pp. 1448–1453, 2021.

[5] F. Ankersen, S.-F. Wu, A. Aleshin, A. Vankov, and V. Volochinov, "Optimization of spacecraft thruster management function," *Journal of guidance, control, and dynamics*, vol. 28, no. 6, pp. 1283–1290, 2005.

[6] E. N. Hartley, "A tutorial on model predictive control for spacecraft rendezvous," in *2015 European Control Conference (ECC)*, pp. 1355–1361, July 2015.

[7] A. Weiss, M. Baldwin, R. S. Erwin, and I. Kolmanovsky, "Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies," *IEEE Transactions on Control Systems Technology*, vol. 23, pp. 1638–1647, July 2015.

[8] D. Silvestre and G. Ramos, "Model Predictive Control With Collision Avoidance for Unknown Environment," *IEEE Control Systems Letters*, vol. 7, pp. 2821–2826, 2023.

[9] S. Zhu, R. Sun, J. Wang, J. Wang, and X. Shao, "Robust model predictive control for multi-step short range spacecraft rendezvous," *Advances in Space Research*, vol. 62, no. 1, pp. 111–126, 2018.

[10] M. Leomanni, G. Bianchini, A. Garulli, and R. Quartullo, "Sum-of-Norms Periodic Model Predictive Control for Space Rendezvous," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 3, pp. 1311–1318, 2021.

[11] J. Briz, N. Paulino, P. Lourenço, P. Cachim, E. Forgues-Mayet, L. Ferreira, A. Groth, E. Papadopoulos, G. Rekleitis, K. Nanos, and V. Preda, "Guidance, Navigation, and Control of In-Orbit Assembly of Large Antennas – technologies and approach for IOANT," in *Proceedings of the 41st ESA Antenna Workshop on Large Deployable Antennas*, (Noordwijk, The Netherlands), ESA, Sept. 2023.

[12] T. M. Lovelly, T. W. Wise, S. H. Holtzman, and A. D. George, "Benchmarking Analysis of Space-Grade Central Processing Units and Field-Programmable Gate Arrays," *Journal of Aerospace Information Systems*, vol. 15, pp. 518–529, Aug. 2018.

[13] B. Açıkmeşe and L. Blackmore, "Lossless convexification of a class of optimal control problems with non-convex control constraints," *Automatica*, vol. 47, no. 2, pp. 341–347, 2011.

[14] D. Malyuta and B. Açıkmeşe, "Lossless convexification of optimal control problems with semi-continuous inputs," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6843–6850, 2020.

[15] P. Hespanhol, R. Quirynen, and S. Di Cairano, "A structure exploiting branch-and-bound algorithm for mixed-integer model predictive control," in *2019 18th European Control Conference (ECC)*, pp. 2763–2768, IEEE, 2019.

[16] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2022.

[17] A. Bemporad and V. V. Naik, "A numerically robust mixed-integer quadratic programming solver for embedded hybrid model predictive control," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 412–417, 2018.

[18] Y. Yang, "Coupled orbital and attitude control in spacecraft rendezvous and soft docking," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 233, p. 3109–3119, Aug. 2018.

[19] C. D. Brown, *Spacecraft Propulsion*. Washington DC: American Institute of Aeronautics and Astronautics, Jan. 1996.